PERSONAL HEALTH BENEFITS OF PERVASIVE COMPUTING AND FEEDBACK

by

ERICK N. TSENG

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degrees of
Bachelor of Science in Electrical Engineering and Computer Science
and Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

May 23, 2001 June 2001

Author_______________

Department of Electrical Engineering and Computer Science
May 23, 2001

Certified by___

Dan Ariely
Thesis Supervisor

Accepted by_

Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Table of Contents

1 Abstract

Personal Health Benefits of
Pervasive Computing and Feedback

by

Erick N. Tseng

Submitted to the
Department of Electrical Engineering and Computer Science

May 23, 2001

In Partial Fulfillment of the Requirements for the Degrees of
Bachelor of Science in Electrical Engineering and Computer Science
And Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

The HealthTracker study was conducted over six weeks to examine the effects of a web-based health monitoring system on a group of 67 college-aged users. A six-group design was used to study the specific effects of individual feedback and competitive feedback information, as well as the potential benefits of pervasive computing. The effects and success of the system were determined by evaluating user compliance and health improvements as demonstrated through a daily wellness survey. It was found that users generally appreciated the additional feedback information as it educated them about their personal health and wellness and allowed them to make informed decisions about their daily lifestyle habits. Subjects accessing HealthTracker wirelessly enjoyed the portability and customization of using a wireless PDA. As a result of a strong attachment to their devices, wireless users entrusted HealthTracker with greater personal information and spent more time interacting with the system. These results were consistent with the well-documented successes of biofeedback technology and the ongoing research in Human Computer Interaction.

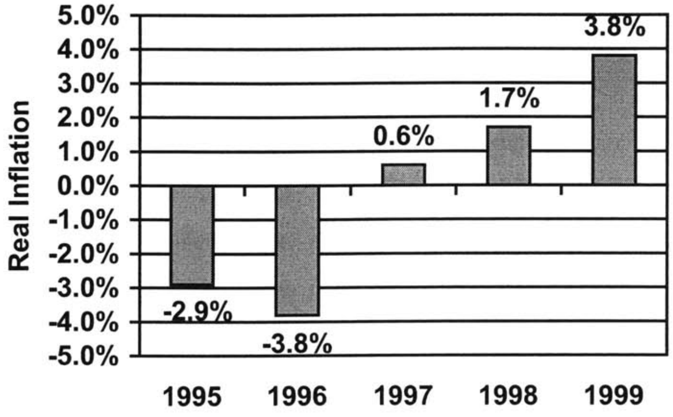
Thesis Supervisor: Dan Ariely
Title: Associate Professor, MIT Sloan School of Management and the MIT Media Laboratory

2 Modern Problems in Healthcare

Healthcare has always been an interactive process. Generally based upon a dynamic relationship between doctor and patient, the processes of medical diagnosis, health monitoring and treatment have always required the input of a patient and the feedback of a doctor. Looking through history, we find that the methods of this input/feedback process have evolved slowly through the years. Organized healthcare arguably began with doctor house calls. Though functional on a very small scale, house visits quickly become very impractical as the technique is scaled to large cities. The development of hospitals and clinics resolved this inefficiency be centralizing health care in large institutions.

Today we are yet again facing an issue of scaling and efficiency in medicine. Many hospitals are now severely lacking in doctors, space and other resources. For example, the average length of a hospital stay is now only 3.54 days, while the average length of stay for intermediate care is just 1.88 days.[1] Meanwhile, increasing demand for quality health care has placed a greater onus on medical institutions and practitioners. The average time a person actually spends with a doctor per visit is now down to only seven minutes.[2] As a result, physicians are finding it increasingly difficult to devote their full attention to individual patients while patients are feeling cheated out of the opportunity to have a complete dialogue with their doctor. The ultimate result of such rising pressures could potentially be a drop in the quality of medical treatment, a consequence we most certainly want to avoid.

The most immediate effect resulting from these problems has been the rising price of healthcare. Currently in the United States, per capita spending on healthcare is approximately $3,000, the highest in the world.[3] Healthcare spending elsewhere in the world, though more economical, is still considerable: approximately $1,800 in Japan, and $1,500 in the United Kingdom.[3] More importantly, these numbers have been steadily climbing over the years. Health-care premiums in the United States, though briefly under control in the early- to mid-1990s, are now outpacing the consumer price index.[4] (See Exhibit 1)



CPI	2.5%	3.3%	1.5%	1.6%	2.7%

Exhibit 1: Real Health-Care Premium Inflation. This graph illustrates the rising cost of healthcare by displaying inflation in healthcare premiums in relation to the Consumer Price Index (CPI). (Source: Health Care Financing Administration, U.S. Bureau of labor Statistics)

These financial problems and quality of service concerns in healthcare are both real and immediate. For example, the high price of healthcare is already discouraging many people with low income from visiting doctors or hospitals when they need to. Other individuals are simply delaying their visits. Both situations pose great medical risks. To make the matter even more complicated, solving these immense problems in healthcare will probably require the combined initiatives of both the public and private sector as well as of a variety of different industries.

3 Personal Health Management

3.1 Why People are Taking Control of Their Own Wellness

In response to the high cost and decreasing quality of healthcare, consumers have started looking for alternative ways to meet their healthcare needs. Personal health management has been steadily increasing in popularity amongst middle and upper class individuals as an effective supplement, if not replacement, to traditional healthcare. Growth in personal health management has been fueled by a number of other factors. First, consumers are responding to real and perceived inadequacies in the current healthcare system.[5] Problems in the current system range from the unavailability of qualified physicians to mediocre medical treatment to the increased awareness of medical errors and hospital infection rates.[5] The Institute of Medicine reported in

November 1999 that medical errors kill 44,000 to 98,000 people a year.[6] In short, frustration with the present managed care system has driven consumers to search for other sources of healthcare.

Changing demographics in the population have been a second stimulus for growth in personal health management. A more educated population, a longer work week and increased income are all elements of this shifting demographic. A more well-informed healthcare consumer translates into a more pro-active consumer. People are spending an increasing percentage of their personal income on personal health and wellness, as they begin to make more health decisions on their own and look for healthcare resources besides their primary care physician. This rise in personal health spending is also being fueled by increased disposable income. With more money, consumers are finding themselves with greater flexibility in spending and many are electing to spend their money on "health preservation," alternative medicine and personal wellness.

A final driving force behind the personal health management movement is the growth of online e-health services. The explosion of the Internet has introduced enormous opportunities to healthcare and medicine, and a whole litany of companies have launched websites targeting the new market of health-conscious consumers. Some websites, many sponsored by medical schools and prominent research hospitals, serve as an informational resource for consumers. Other websites are derivates of more ambitions business plans, hoping to actively integrate themselves into people's lives as an interactive health monitor, a source of medical advice and prescription medicine and even as a virtual doctor. The Web has truly revolutionized the healthcare industry by creating an empowered consumer. "Consumers are spending more time looking online for resources relevant to their daily lives," says David Sze, vice president of strategy for Excite.[7]

3.2 Shifting from Reactive to Preventive Medicine

As part of this movement towards personal health medicine, individuals with higher socio-economic status have also been shifting their focus from crisis-driven health to prevention-based health. Prior to the 1990's, people rarely worried about their health unless they fell ill. In this crisis-driven paradigm, people only dealt with health issues "after the fact." Since then, a new philosophy has come to the fore: preventive medicine. Instead of waiting for an illness to strike, people are now pursuing a regimen of regular exercise and healthy diet to avoid sickness in the first place. These individuals have decided to take their health into their own hands. Instead of relying on doctors and physicians, these people are taking an active role in maintaining their well-being: joining fitness clubs, taking yoga, purchasing dietary supplements, etc.

Exhibit 2 illustrates this trend from reactive to proactive healthcare. As a result of this new movement, the health industry has seen a surge of new health and wellness products and services ranging from food supplements to fitness training to mental relaxation courses. The Hartman Group estimates that this new market of preventive health products and services now accounts for over $60 billion in sales.[8]

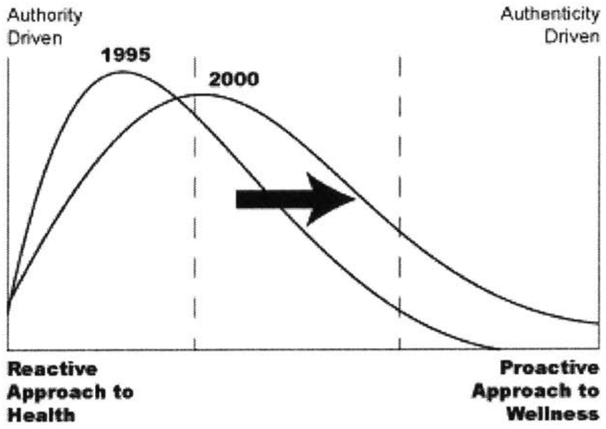


Exhibit 2: Movement from Reactive to Proactive Healthcare. It used to be that individuals would only visit doctors and hospitals if they fell ill. Now, instead of just reacting to medical problems, individuals with higher socio-economic status are shifting to a more proactive approach to wellness. In this manner, people try to carry a healthy, well-balanced lifestyle in order to prevent illness before it begins. (Source: The Hartman Group)

It is important to note that these trends in healthcare are generally limited to people in the middle to upper class. Individuals with lower socio-economic status have yet to benefit from these changes and are most likely the ones suffering the most from the increasing cost of healthcare. Hopefully, as developments in personal health management and preventive medicine continue to unfold, and as Internet technology becomes more pervasive, more and more people will begin to benefit from improved healthcare.

Nevertheless, these trends in personal health management and preventive medicine have already begun to revolutionize the healthcare industry. Hospitals and doctors are no longer monopolizing the healthcare process. Instead, consumers now have multiple options and are empowered to make their own educated decisions regarding their health and well-being. This is the fundamental principle behind HealthTracker's design and the research's goals.

4 Difficult Problems with Personal Health Management

While personal health management can be very effective in handling standard illnesses and medical afflictions, it still cannot completely replace doctors and traditional medical practices. There are multitudes of diseases and conditions that only a doctor ought to handle and even more questions and concerns that only a doctor can answer. There are even medical afflictions that are capable of progressing completely unnoticed unless detected by a doctor's trained skills of observation.

4.1 The Medical Diagnoses Process

A doctor's singular contribution to the process of diagnosing a patient is his or her trained expertise, which enables him or her to perform the following:

1) Ask the right questions that will elicit an accurate portrayal of the problem.
2) Notice physical or mental indications of an ailment.
3) Draw accurate conclusions and offer professional medical advice.

All three tasks build upon the training a doctor receives during medical school and then even more so upon a doctor's work experience. The first contribution is critical, as asking the right questions can often save time wasted in pursuing incorrect avenues and possibly incorrect diagnoses.

The second contribution is significant for very similar reasons. In addition, it could take a doctor's trained eye to notice a physical or mental anomaly that requires attention. For example, an individual can ignore a broken bone for an indefinite amount of time if he or she develops a tolerance for the pain. After a while, the pain becomes second nature, and the individual completely forgets about the injury, resulting in a bone that heals incorrectly and becomes permanently misaligned. In this case, a doctor would notice symptoms of a problem that a patient had developed a tolerance for or learned to work around.

The third and final contribution is clearly important, as this is the step where a patient would then take a diagnosis and embark upon a regimen of treatment. Accuracy in the diagnosis is essential.

4.2 Identifying Gradual Health Changes

Many health problems progress at such a gradual rate that the affected individual learns to adapt to the symptoms over time. For example, many chronic illnesses that afflict the elderly begin with barely noticeable symptoms. One of the initial indicators of Alzheimer's disease is very slight memory loss, a symptom that the afflicted can grow accustomed to as the problem grows worse. Alzheimer's typically begins with simple short-term memory loss, like forgetting where one put one's keys. It is clearly very easy to pass off these problems as simple forgetfulness. Alzheimer's then slowly evolves to more pronounced, intermediate memory loss, like failing to remember whether one had eaten lunch yet. Yet since these problems evolve over many years, they too can be easily overlooked. Oftentimes Alzheimer's isn't caught until it has progressed to the point where the afflicted begins to forget important and very noticeable memories, like relatives and even one's own name.

Multiple Sclerosis is another disease with symptoms that progress extremely slowly. The muscle atrophy of MS sufferers is typically very gradual. As a result, the afflicted learn to adapt to their slowly weakening muscles and effectively train themselves to no longer notice the problem. Parkinson's disease also develops very slowly and begins with the occasional shaking of the hands, an occurrence that can be easily overlooked and ignored. As the disease progresses and the palsy becomes more pronounced, the afflicted can often adapt and delay medical attention and proper treatment. Because of their gradual symptoms and the ease with which the afflicted can adapt to them, these types of medical ailments are often not diagnosed in a timely manner.

4.3 Catching Unnoticed Health Maladies

In these types of situations, the afflicted is completely unaware of his/her condition and if ignored, these types of conditions can often escalate into a serious problem that might seriously threaten a person's health. Attention Deficit Disorder (ADD) is one example of such a condition. A person suffering from ADD may simply not notice his/her inability to pay attention, focus on work or general hyperactivity. To a person with ADD, all of these symptoms would seem completely normal since that individual would have grown up with these symptoms. If left untreated, ADD could severely interfere with a person's ability to function at work or at home.

Depression is another medical affliction that can go completely unnoticed. Having always suffered from the condition, a person might not realize there's something wrong with his/her constantly feeling sad and depressed. Individuals suffering from depression display a wide range of both mental and physical symptoms, ranging from weight loss to thoughts of suicide. All of these symptoms could easily go unnoticed by the afflicted. Without treatment, severe cases of depression could lead to someone getting hurt or even killed.

5 Biofeedback

To catch many of the difficult problems outlined in Section 4, doctors and researchers have looked to biofeedback for answers. Doctors, psychologists and therapists have been using biofeedback in medicine for decades. The simple acts of taking someone's temperature or stepping on a scale are some of the most basic forms of biofeedback. In short, "biofeedback is a treatment technique in which people are trained to improve their health by using signals from their own bodies."[9] Already, biofeedback has been successfully used to help combat a variety of health-related maladies. Stroke victims have been trained to regain movement in paralyzed muscle.[10] Patients with a high risk of heart attacks have been able to lower their blood pressure. Some people have even successfully used biofeedback to lose weight. Biofeedback has also proven to be extremely useful in preventing many of the ailments covered in Section 4.

5.1 Auditory Biofeedback

Over the past thirty years, a wide variety of feedback techniques have been employed to solve medical problems, one of the most common being auditory feedback. For example, in 1974, Spearing and Poppen helped a 23-year old student combat cerebral palsy. Due to his condition, this individual dragged his toes while walking. So the two doctors installed switches in the man's left shoe "to count the number of foot drags, and also to trigger an auditory signal to give him feedback when he did drag his toe."[10] At the outset, the student dragged his foot 71% of the time, as measured against the student's total number of steps. Over a 4-month period of using the biofeedback technique, that percentage dropped to 33%. Withdrawal of the feedback system quickly raised that percentage to 68%, and then the reapplication of the biofeedback reduced the average to less than 10%. The two doctors' conclusion was that this man's palsy could be significantly reduced under a regular regimen of "booster sessions," during which the biofeedback technique would be used.[10]

Auditory biofeedback techniques were also used to alter congenital nystagmus (CN) in albinos, an ocular disease that is characterized by involuntary rapid and repetitive movement of the eyes. In response to auditory cues, patients were able to reduce the amplitude of their nystagmus by

50% or more.[11] Then, when these auditory cues were combined with visual triggers, the patient was able to alter the waveform of the nystagmus in such a manner that progressively larger portions of each cycle of the nystagmus were spent with the eye relatively stationary. As a result of this alteration in the waveform, patients' visual acuity (VA) also improved significantly.

Auditory biofeedback is also being used in conjunction with other techniques like virtual reality machines, kinematic analysis and interactive video conferencing to battle other maladies like Parkinson's disease and Attention Deficit/Hyperactivity Disorder (ADHD).

5.2 Electromyographic (EMG) Biofeedback

Alverto Marinacci was one of the first physicians to ever use another variation of biofeedback to repair impaired muscles. Marinacci's technique was to use employ electromyographic (EMG) signals as feedback. Marinacci "recorded EMG impulses from impaired muscles and then turned the output into auditory signals to provide the patient with information about any muscle activity that could thus be detected."[10] One of Marinacci's patients had been suffering from serious muscle damage and could not move his orbicularis and frontalis. After just one week of EMG treatment, the patient showed about 20% voluntary contraction of those very same muscles. Over the next six months, this patient's motor units had increased from 400 μV to 1500 μV, and muscle mobility rose 40%.[10]

Electromyographic biofeedback training has also been used to increase range of movement in muscle. In 1981, EMG biofeedback was administered to a mentally retarded individual suffering from cerebral palsy that resulted from postnatal meningitis.[12] The patient was a 29 year old woman and the symptoms of her condition included a short attention span, lack of motivation, low frustration tolerance and increased spasticity in her left hand. The individual also had limited extension movement in her fingers and a distinct amount of wrist flexion and wrist ulnar deviation. With the help of EMG biofeedback, the patient underwent repeated physical training on her fingers and wrist, in an effort to restore extension movements in those regions. Through this training, the patient was successful in significantly improving her condition. The individual was able to hyperextend her fingers following finger extension training and there was also a substantial increase in active wrist extension and a marked reduction in wrist ulnar deviation.

EMG biofeedback has also been used to alleviate chronic back pain. In 1979, a 71 year old female suffering from chronic back pain and paravertebral muscle spasm underwent seventeen sessions of EMG biofeedback training. In these sessions, the patient was taught to progressively relax her back muscles. EMG levels were monitored during baseline, treatment and follow-up sessions. As time progressed, there was a demonstrable decrease in both EMG levels and the frequency of the back pain. The improvements were maintained for another 12 weeks after the last EMG biofeedback treatment.[13] EMG biofeedback has also been used to lessen chronic low back pain (CLBP). In these situations, patients were trained to isolate and slowly restore the functional abilities of their trunk extensor muscles using EMG biofeedback. Before long, significant improvement was observed in the strength of patients' lumbar paraspinal muscles and patients reported a substantial reduction in their low back pain.[14]

The range of therapy possible with EMG biofeedback is seemingly endless. The technique has actually been used to improve cases of dysmenorrhea. In one particular study, fifteen dysmenorrhea sufferers were divided into three groups: those that received both relaxation training and EMG biofeedback, those that received just relaxation training and then finally those that received no treatment. On the first day of menstruation, patients receiving biofeedback training were able to maintain a reduced level of EMG activity, while patients receiving just relaxation training demonstrated an elevated level of EMG activity. As time progressed, subjects in the biofeedback group exhibited significant improvement in their dysmenorrhea symptoms, whereas patients receiving just relaxation training and no treatment showed no signs of improvement.[15]

Today, physicians and researchers continue to use EMG feedback. There are even commercially-available products that utilize EMG signals to restore muscle functionality. These devices allow individuals to measure and quantify their muscle movements from the privacy of their own homes. And with the assistance of the audio and graphical feedback provided by these devices, individuals can work towards personal goals set by physical therapists.

Like auditory feedback, EMG signals are also often combined with other forms of biofeedback to help mitigate a variety of medical problems. For example, basal frontal EMG and frontal EMG have been combined with finger temperature to help individuals recover from stress and increase overall job performance.[16]

5.3 Respiratory Sinus Arrhythmia Biofeedback

Respiratory Sinus Arrhythmia (RSA) is the fluctuating pattern of heart beats accompanying the respiratory cycle. The heart beats more slowly when we exhale, thus lengthening the heart beat period in RSA; while the heart beat quickens when we inhale, thus decreasing the RSA period. Using RSA biofeedback, Vaschillo (1984) was able to demonstrate that humans were capable of willingly controlling their heart rhythm structure. By monitoring both heart rhythm and blood pressure, Vaschillo observed that a healthy adult's cardiovascular system operates at resonant frequency ranges of 0.055 – 0.11 Hz and 0.02 – 0.55 Hz.[17] These two resonant frequency ranges appeared to be controlled by the baroreflex, in which the "slow" frequency range (0.055 – 0.11 Hz) was caused by the baroreflex's cardiac component and the "very slow" frequency range (0.02 – 0.55 Hz) was caused by the baroreflex's vascular component.

RSA signals have been used in Russia as a device for helping individuals treat stress-related physical disorders and neurotic conditions.[18] RSA biofeedback has been used to treat neurotic patients, bronchial maladies, heart rate disorders and disorders of the autonomous nervous system. One Russian study used RSA to help reduce the effect of asthma. "The theoretical rationale for this has been outlined by Vaschillo (1984), who proposed that periodically producing voluntary increases in the amplitude of cardiac period oscillations at this frequency may exercise, and thereby strengthen, the baroreceptor reflexes that modulate blood pressure."[18] By directly projecting these baroreceptors to the hypothalamus and limbic system, a decrease in stress-induced sympathetic arousal and/or parasympathetic withdrawal can occur. This would then affect the autonomic perturbations on asthma.

Researchers instructed subjects to breathe in a variety of manners, for example through pursed lips and through a wide-open mouth. The study found that users would attempt to maximize the amplitude of the RSA biofeedback by breathing at a rate within the frequency range of 0.03-0.12 Hz. At this particular frequency range, the period of heart beats is believed to be influenced by both the sympathetic and parasympathetic systems. This equilibrium leads to an ideal blood pressure that is cause for therapeutic effects on asthma. The breathing exercise worked especially well on children.

This research in asthma led Vaschillo to additional findings in respiratory conditions. Using RSA biofeedback, Vaschillo was able to discover that deep breathing led to an improvement in blood pressure and a stretching of the muscles lining the body's airways. The combined effect of these two results was a decrease in bronchoconstriction.[18] The study was not able to conclude however, whether this relaxation of the bronchial region was the sole result of the deep breathing, or also an effect of subjects' relaxation. Though subjects claimed that they were not any more relaxed while processing the RSA biofeedback, these assertions could not be definitively confirmed in the study.

It turns out that RSA biofeedback can be used as a relaxation method. The curative properties of RSA biofeedback on the autonomous nervous system allows it to be effectively used to teach individuals to control their psycho-emotional condition. Vaschillo conducted a study involving 20 healthy wrestlers whose ages ranged from 19 to 22 years old. Half of the group received no biofeedback, while the other half received 20 minutes of RSA biofeedback training daily for 10 consecutive days. At the completion of the experiment, athletes who hadn't received biofeedback demonstrated no noticeable improvement in their balance and inhibitory activities. Wrestlers who had received RSA biofeedback, on the other hand, experienced significant decreases in muscular relaxation disorders.

The obvious significance of one's heart to nearly every function of the body makes heart rhythms an extremely powerful form of biofeedback. Even without the instrumentation needed to measure RSA biofeedback, something as simple as measuring a pulse can serve as sufficient biofeedback to improve a person's health. For example, in 1977, eighteen healthy adults participated in a series of weekly exercise sessions. In each of these sessions, subjects walked on a treadmill five times for 10 minutes each time. The treadmill was set to a speed of 2.5 mph and at a 6% grade. Eight of the study's participants received beat-to-beat heart rate biofeedback and were instructed to try to lower their heart rates. The other ten participants did not receive any feedback. At the end of five weeks, the group receiving feedback had demonstrated a dramatically lower mean heart rate compared to the control group (96.8 versus 108.6 beats per minute). In addition, the experimental group's systolic blood pressure was also much lower (114.0 versus 131.3 mmHg). So, even with biofeedback as simple as one's own pulse, these experimental subjects were able to register significant improvement in their cardiovascular health.

5.4 Controlling Blood Flow

Rheoencephalography (REG) is a method widely used in Russia for the continuous evaluation of brain blood flow. Since the 1960's, REG has been used by many Russian clinics and hospitals as a tool for diagnosis of cerebrovascular disorders because of its ease of use and well-developed procedures for REG waveform recognition. In working closely with REG however, physicians at the Applied Physiology Laboratory, the Institute for Complex Problems of Hygienic and Occupational Diseases, and the Siberian Department of the Russian Academy of Medical Sciences soon noticed that people had the ability to control and change components of the REG waveform through REG biofeedback.

With this realization, doctors and researchers designed a variety of studies to determine the potential of REG feedback. One such study involved 241 workers from a metallurgical rolling plant.[19] All of these individuals were suffering from a minor level of psychosomatic disorders with symptoms such as headaches, sleepiness and periodic blood pressure fluctuations. Through a series of 10 meetings with doctors over the course of two weeks, the subjects were tested on their ability to control the arterial and venous components of their brain's REG waveform. The feedback itself was both auditory and visual. Whenever subjects reached a particular REG threshold set by the physician, both a sound and a green light would turn on.

The subjects were asked to control their REG waveform in three different ways: 1) to increase the arterial component of their REG waveform, 2) to decrease the venous component of their REG waveform, and 3) to increase both the arterial component of their REG and decrease the venous component simultaneously. Participants were then instructed to try to achieve these goals using a variety of strategies, such as breathing, music listening, temperature feelings and imagination.

It turned out that 195 subjects or 80.9% of the all the subjects were successful in controlling any component of their REG waveform. More importantly, these subjects also demonstrated improved psychological conditions and higher work productivity. In fact, 85% of the successful experimental subjects had no performance errors at the plant, where individuals typically cause 0 to 4 errors in a shift. Psychological tests evaluating emotional status, mood and feelings were administered both before and after the biofeedback sessions. Labor productivity was estimated by managers at the metallurgical plant.

The results of this study indicate that not only can people control the blood flow in their brain, but that there are positive benefits to this ability as well. Within the context of occupational medicine, labor productivity and quality of work can be increased. Psychologically, REG biofeedback can help to control emotional states and mental wellness.

In another medical study, the self-regulation of blood flow and biofeedback helped alleviate the symptoms of classic migraines. Eight migraine sufferers were trained to control their own blood volume pulse amplitude (BVPA) using photoelectric plethysmograph feedback. These subjects learned to increase and decrease BVPA at superficial temporal artery (STA) and finger locations.[20] Then while experiencing a migraine headache in a laboratory, these subjects were measured for and received pain measurements. Results of the study showed a significant relationship between subjects' voluntary changes in pulse amplitude and the reports on pain measurement.

5.5 Ocular and Olfactory Feedback

Ocular biofeedback is perhaps one of the earliest biofeedback techniques ever developed and has been used to alleviate a wide variety of medical illnesses and conditions. For example, a mechanical device was developed in 1988 that used visual feedback to improve heat control and symmetrical standing in children suffering from cerebral palsy. The device was a biofeedback seat insert that encouraged an erect sitting posture.[21] Due to their condition, these children exhibited inadequate trunk control and the biofeedback device's purpose was to improve trunk control by developing better posture. The seat contained a momentary-contact pressure switch that was activated when a child exerted pressure on it by extending his/her trunk. The switch could then be wired to control any type of electronic device such as a television or radio. In this study, the switch was connected to a VCR.

The five children in this study quickly made the connection between the pressure switch and the VCR that would turn on and play an entertaining video program. Driven by their desire to watch the video program, the children would voluntarily extend their trunk against the pressure switch. These results demonstrate the effectiveness of ocular biofeedback as a therapeutic training tool in improving the posture of child suffering from cerebral palsy. This improvement in posture then led to restored trunk control.

Ocular feedback has also been used to alleviate a wide variety of other medical conditions. Professor Kenneth J. Ciuffreda of the School of Optometry at the University of New York has reported positive results in using visual biofeedback in correcting such disorders as nystagmus, strabismus and amblyopia.[22]

In 1957, Efron reported a patient who could actually stop the onset of an uncinate seizure by a particularly strong olfactory stimulus. Efron was able to transfer this olfactory stimulus to a visual stimulus of a silver bracelet. These two sensations were essentially paired together in the patient's mind and from then on, the sight of a silver bracelet was enough to thwart the onset of a seizure.

Forester also researched techniques to prevent seizures. By the mid-1960's he was successfully using "anti-feedback" to prevent seizures. Forester found that certain stimuli would actually trigger seizures in some of his patients. So, instead of using Efron's technique of presenting positive stimuli that would prevent seizures, Forester conditioned his patients to become desensitized to negative stimuli that reliably *triggered* seizures. Employing a technique similar to vaccinations, Forester presented these stimuli in small dosages that were innocuous enough not to trigger seizures on their own, but potent enough to eliminate the threat of a similar and more noxious stimuli. Over time, this vaccination process proved to be successful and Efron's patients no longer responded to the harmful stimuli.

5.6 The Future of Biofeedback

With the advent of computer technology in medicine, biofeedback now has the potential of achieving even greater results. Advanced electronic sensors have expanded the range of health problems that could be solved with biofeedback, while innovative computer technology has improved the level of interaction between a patient and the feedback information itself. The MIT Media Laboratory and the d'Arbeloff Laboratory for Information Systems and Technology are just two centers that are researching these health-related sensors and computer technologies.

Michael Hawley of the MIT Media Lab for example, has recently designed Embedded Sensor Packs (ESP) for expeditionary projects. ESP technology is an end-to-end system that embeds biosensors into a wide variety of clothing so that human vital signs can be monitored in real-time and in real-life situations.

> "Consider being able to watch one's vital signs and surroundings in real-life situations. No longer is your medical care a matter of an annual check-up plus trips to the doctor when you're sick. Instead, regular at-home monitoring can give your physician an early warning when you start to ail. Athletes can be analyzed in competitive situations; one example: runners monitored on the track, not the

> treadmill. Explorers in harsh or life-threatening environments can be closely watched from a remote location, receiving real-time instruction to help them remain safe in the field. The examples are limitless. With ESP, the technology is catching up."[23]

Since its development, ESP technology has been successfully outfitted for marathon runners, cyclists, Army rangers and members of an Everest expedition. As the quotation above explains, it is biofeedback technology like ESP that will eventually empower individuals to manage their own health and gain a significant amount of autonomy from the traditional healthcare system.

The d'Arbeloff Laboratory for Information Systems and Technology is working towards a very similar vision. A the d'Arbeloff Laboratory, researchers are also developing biosensors that, when incorporated into wearable computers, can allow individuals and physicians to monitor biofeedback in real-time and real-life.

> "Rather than merely providing Internet access services, the utility of wearable computers will be extended dramatically when they are hooked up to various devices and systems. The ring sensor developed at the d'Arbeloff Laboratory, for example, monitors the physiological status of the wearer and transmits the information to the medical professional over the Internet. This ring sensor allows for continuous, long-term monitoring, which opens up new possibilities of preventive medicine and long-term care."
>
> "Other health monitoring devices, such as the vestibular-ocular test apparatus, the glucose counter, and the insulin delivery system, can all be hooked up to a wearable computer without wiring the patient body. In turn, a medical professional carrying a wearable computer would be able to access all the data, clinical protocol, and operational procedures whenever and wherever needed. Moreover, their services would be recorded at all times, and their task performance would be monitored rigorously to avoid any mistake and malpractice."[24]

Many new doors will open when these new technologies are combined with established biofeedback techniques that have been developed and honed over the past few decades. For example, biofeedback can become a truly integral component of people's lives as sensors and computing technology are directly incorporated into clothing and other apparel. This promise of "wearable health monitoring" will be an enormous development for people suffering from chronic illnesses and require constant monitoring. In effect, the continued development of biofeedback will serve as an extension of the self health care movement, empowering individuals to take control of their own health and well-being and to continue living full, productive lives despite medical illness.

6 How Technology Can Provide a Solution

There clearly are a variety of problems in modern-day healthcare. Financial and resource constraints threaten to lower quality of service while increasing healthcare costs are preventing more and more people from affording healthcare. Personal health management has been one popular response to these troubles, but its current scope is severely limited. Only a percentage of middle to upper class individuals is currently practicing self-health care. Most people have yet to engage in and benefit from the new medical practice. And even current self-health care techniques fail to provide a comprehensive solution. It remains extremely difficult for individuals to detect and properly self-diagnose illnesses and conditions that may be adaptable or even imperceptible.

A solution to many of these problems can be found in technology, specifically by leveraging recent computing developments in such areas as artificial intelligence and Internet technology.

> "In the United States today, 90 million people suffer from chronic medical conditions like diabetes, asthma, and heart disease. Chronic illnesses account for approximately 75% of total healthcare costs in the United States. As a result, healthcare organizations, realizing the importance that successful management of these patients has on overall cost-effectiveness and patient satisfaction, need innovative methods to manage these patients."[25]

Technology can provide these innovative methods. Artificial intelligence and computer automation can help physicians save a large amount of valuable time by assisting them in simple medical procedures like health monitoring and basic medical diagnosis. Internet technology will help to maximize the reach of new healthcare developments. With its immense scalability and increasing ubiquity, the Internet will most certainly serve as the vehicle through which healthcare can service the greatest number of people, regardless of their socio-economic status. The HealthTracker system is also a technology that could prove to be the solution to many of the current healthcare problems. Its functionality will be explained shortly in Section 6.4.

6.1 Automated Diagnoses

Research in automated diagnoses has been expanding very quickly and promises to deliver remarkable solutions to many modern-day problems. For example, researchers are currently using artificial intelligence to automate the process of medical diagnoses. As explained in Section 4.1, the process is currently performed only by medical practitioners and can often require a significant amount of time. The advent of computer smart agents and artificial intelligence however, introduces the possibility that a computer could replace a doctor role in some or all of the medical diagnoses steps.

It will not be long before a computer containing a database of medical knowledge and frequently encountered medical problems could use artificial intelligence to ask appropriate questions, examine a patient's physical and mental feedback, and suggest reasonable diagnoses. Taking this future possibility even one step further, this intelligent software loaded on a wireless device could then enable any individual to receive expert medical advice anytime and anywhere.

It is important to note, of course, that it is questionable if a computer will ever fully replace a doctor. Even with such advanced technology available, a doctor ought to still regularly corroborate a computer's diagnoses, and patients with more serious medical problems should still rely on a human's learned opinion. Nevertheless, a wireless "computer doctor" could still potentially relieve much of the burden currently placed on doctors, and free up their time to handle other important duties, such as critical medical procedures or life-saving operations.

6.2 Web-Based Solutions

The recent surge of the Internet and web-based services has presented itself as an obvious solution to current problems in healthcare. Over the past decade, the Web has blossomed into a wellspring of health and wellness information and services. Rising costs and perceived inadequacies in the current healthcare system along with a more pro-active population have steered consumers to the Internet.

Online, individuals can find information that is both personalized and private. They have access to an extensive library of news, journal articles and reference materials on diseases and conditions. Expert advice can be obtained regarding drugs and other health topics. Virtual communities are available to discuss health-related topics and e-commerce sites make available health-related products and services. The Internet empowers consumers to find health and wellness information anytime, anywhere, often at no cost–a very compelling proposition.

"The United States spends more per capita on healthcare than any other industrialized country, yet some 50 million Americans go without medical coverage. In a wired world, the current health-care system is an expensive anachronism."[2] The growing number of web-based health service offerings promises to shatter this status quo however. Already, the number of people surfing the Web for medical information has increased from 17 million in 1998 to more than 20 million in 2001, according to Cyber Dialogue.[2]

The variety of self health care services offered on the Web is astounding. For example, some websites are already providing consumers with online medical diagnoses and advice, a procedure we had earlier observed as historically monopolized by doctors. One such website is Boston-based Mediconsult.com. For $195, an individual can submit a medical history online to Mediconsult.com's MediXpert advice service and pose health-related questions to actual physicians. While the resulting medical report and treatment suggestions are ultimately generated by a human, the web interface has enabled individuals to access the information from the comfort and privacy of their own home.

Other websites like Drkoop.com, HealthCentral and InteliHealth provide a wide variety of medical news, research and access to virtual communities. WebMD is another popular website that offers a broad range of medical information and services. At WebMD, visitors can browse through an extensive library information that covers a very comprehensive collection of medical illnesses and conditions. Links to additional medical journals and related websites are also provided for those seeking additional research. WebMD also provides users with suggestions for possible treatment and medication. The website can even help individuals find a local doctor.

Perhaps one of WebMD's most powerful features is an interactive space in which users can network with other people who have similar concerns. In a real-time chat format, groups of people can converse and share their experiences and exchange advice. This online community can be extremely effective, since the human factor makes the online experience much more personable and will set people more at ease.

InteliHealth, Harvard Medical School's consumer health information website, offers many of these same resources. One of InteliHealth's more unique features is a free service that will send weekly emails to subscribers. Subscribers can specify the content of these emails, and their options range from general health news updates to specialty emails covering topics like allergies, cancer, drugs and nutrition. InteliHealth also contains a complete searchable version of Merriam-Webster's Medical Dictionary. Here again is an instance of a website trying to educate consumers about their own health.

Not only are these general health and wellness websites a rich source of information, but they can actually enact real change. Microsoft executive Linda Stone was diagnosed with a chronic inflammatory disease on her 40th birthday a few years ago. When she heard the news, Stone turned to the Internet to research her condition and to join relevant online communities. Through these resources, Stone eventually discovered that she had been misdiagnosed and was able to take appropriate action.[2]

There are also an increasing number of websites dedicated to providing online and offline biofeedback resources. One such website, biofeedback.net, offers a wide variety of biofeedback support. The site lists a litany of biofeedback resource centers and societies from around the world and provides links to their homepages. Biofeedback training courses and medical products that employ biofeedback technology are sold as well. The site also provides visitors with a list of additional biofeedback research sites and contact information for biofeedback consultants and laboratories. Another site, biofeedbackzone.com, offers very similar resources. In addition, biofeedbackzone.com also contains a relatively large collection of research papers in biofeedback technology so that consumers can educate themselves about the biofeedback process.

6.3 Health Hero Network

A private company called Health Hero Network has developed and is now marketing a wellness monitoring product called the Health Buddy Appliance. The Health Buddy Appliance is designed to help healthcare providers and patients collaborate and manage chronic illnesses together using a web-based application and an Internet appliance. Using the Internet appliance, customers answer a series of daily questions about their physical and mental health. These responses then get sent back to healthcare providers who monitor the data and can send back feedback or suggestions.

Health Hero Network's central mission is extremely similar to the primary goals of the HealthTracker research project: to empower patients with control of their own health. In achieving this goal, many other positive benefits will follow. Hospitals will free up space, since more patients can be sent home where they can monitor their own health with the help of technology like the Health Buddy Appliance or HealthTracker. Medical costs will decrease, since less money will be spent on expensive hospital and physician resources. And hopefully, the occurrences of avoidable diseases and medical conditions will diminish, as more and more people practice preventive health medicine.

The Health Hero Network recently performed a case study of its Health Buddy Appliance to test the product's effectiveness.[26] The company collected data from three Congestive Heart Failure (CHF) programs that were using the Health Buddy. Together, these three CHF programs had a total of 238 patients. Based on the collected data, it was determined that using the Health Buddy led to a 69% decrease in the utilization of hospitals and emergency rooms for CHF-related causes and reduced costs by $8000 per individual per year. In another case study performed on two different CHF programs, with a total of 213 patients, the same type of positive results were observed. In this second study, inpatient hospitalizations and emergency room visits dropped by 70%, while costs were reduced by $3,304 per member per year.

Users of the Health Buddy also reported an increased satisfaction in their conversations with their health care providers. Compliance amongst users, measured by how many and how promptly the system's daily questionnaires were filled out, was extremely high. About 90% of the daily sessions were completed on the day they were received. These extremely positive results are very promising for the HealthTracker study, given the two projects' overlapping goals and similar technology.

6.4 HealthTracker

The HealthTracker system developed in this research project is an extension of some of the more recent web-based healthcare technology and is deeply rooted in consumers' ongoing push towards personal health management. The system is a web-based health monitoring system that is capable of delivering a variety of feedback information to its users. HealthTracker is also designed to run on both a desktop computer as well as a wireless PDA and can be accessed using any standard web browser. To examine the effects and evaluate the potential benefits of the system, HealthTracker was tested on a group of college students in their twenties and thirties. The research focused around two major areas:

- **Human Computer Interaction**
    - **The trust relationship between users and the application:** How much personal information would users share with the health monitoring system? How attached will users become with the program?
    - **The effects of pervasive computing on personal health management:** Does a wireless device improve the efficacy of a web-based health monitoring system? If so, then by how much? Do users feel more attached to the system when accessing it using a wireless device versus a desktop computer?
- **The effects of feedback information on a person's health:** Can feedback in a health monitoring system affect or control a person's health? If so, what type of feedback is most effective and what are the results?

The results of the HealthTracker study would add to previous research completed in areas of feedback technology and human-computer interaction (HCI) by revealing the efficacy, and ultimately the viability, of monitoring and controlling people's health via the Web. And in the mounting advance towards mobile devices and increased personalization and portability, this research project would also help determine the possible advantages and disadvantages of monitoring people's health wirelessly. Finally, the HealthTracker study will also extend past research in feedback technology, exploring new techniques made possible by the advent of the Internet and determine the value of Web-based feedback.

6.4.1 Expectations

Before the HealthTracker study began, we established a number of hypotheses. First, we expected users accessing the system with a wireless device to develop a closer relationship with the monitoring program. Past research in pervasive computing and HCI demonstrated that when it came to portability and flexibility, individuals logically favored wireless devices over desktop computers. Research also found that wireless devices quickly become integral in people's lives as users learned to carry the devices around wherever they went. Eventually, users become attached to their wireless devices, treating them much like a set of keys or a wallet. For

HealthTracker, our hypothesis was that this progressive attachment to a wireless device would, by association, also translate into an increased attachment to the monitoring software running on the device.

A corollary to this hypothesis was that wireless users would then share more information with the HealthTracker system, having developed a closer relationship with the program. More generally in regards to information-sharing, we projected that a trust relationship would develop between all users, both wireless and desktop, and the system as the study progressed. The rationale here is that as users grew comfortable with HealthTracker in their lives, their reservations about the program would diminish.

Our final hypothesis was that biofeedback would have a visible impact on people's health, and the more feedback a user received, the more their health was affected. Past experiments in biofeedback have demonstrated that humans respond very positively to feedback when it is administered properly. Asking users to access HealthTracker on a daily basis and making feedback graphs readily available were two ways we strove to make the feedback as powerful as possible.

7 HealthTracker Experiment

7.1 The Subjects

67 students and employees from the Massachusetts Institute of Technology participated in the HealthTracker study. The ages of the participants ranged from late teens to early thirties. Subjects were solicited using email and word-of mouth and were offered two incentives conditional upon successful completion of the study: 1) $50 compensation and 2) a chance to win one of several Handspring Platinum Visor PDA's.

All the subjects were first divided into two separate studies: a desktop PC study and a wireless PDA study. The 37 subjects in the desktop computer study were instructed to access the HealthTracker system daily via a desktop computer. The 30 other subjects were each given a Handspring Platinum Visor Personal Digital Assistant (PDA). We equipped these handheld devices with an OmniSky wireless modem, allowing the PDA's to connect to the Web. These wireless users were instructed to access the HealthTracker system through their wireless devices.

Each of the two studies was then subdivided into another three groups for a total of six groups. Each of the six experimental groups was given a different condition. (See Exhibit 3) Each group was also assigned a more descriptive name that reflected its experimental conditions. (See Exhibit 4) Users in groups one through three accessed the HealthTracker system through the desktop computer, while users in groups four through six used the Handspring Visor to access the system. Three groups (the Desktop–No Feedback Group, the Wireless–Use Limitations Group and the Wireless–No Feedback Group) received no graphical feedback. Two groups, the Desktop–Individual Feedback Group in the desktop study and the Wireless–Individual Feedback Group in the wireless study, received graphical feedback that illustrated individual performance over time. The Desktop–Competitive Feedback Group received this same individual feedback, but it was portrayed along with the average performance of all the participants in the study. Finally, the Wireless–Use Limitations Group in the wireless study could not use their wireless PDA's for any purpose but to access the HealthTracker system.

	Platform		Feedback			
	Desktop PC	Wireless PDA	None	Display Individual Performance	Display Individual Performance vs. Study Average	Restrictions on Using the PDA
Group 1	✓		✓			
Group 2	✓			✓		
Group 3	✓				✓	
Group 4		✓	✓			✓
Group 5		✓	✓			
Group 6		✓		✓		

Exhibit 3: Conditions for the Six User Experimental User Groups. A checkmark in this table indicates an applicable test condition to a particular test group. Each test group was assigned a particular platform through with to access the HealthTracker system. Users also received no feedback or one of two types of feedback, depending on their group assignment. Finally, users in the Wireless–No Feedback and Limitations on Use Group were not allowed to use their wireless PDA's except to access the HealthTracker system.

Group 1	↔	The Desktop–No Feedback Group
Group 2	↔	The Desktop–Individual Feedback Group
Group 3	↔	The Desktop–Competitive Feedback Group
Group 4	↔	The Wireless–Use Limitations Group
Group 5	↔	The Wireless–No Feedback Group
Group 6	↔	The Wireless–Individual Feedback Group

Exhibit 4: Descriptive Naming Assignments for the Six Experimental Groups. Each of the names is intended to reflect the experimental conditions of the associated group.

There are very important relationships between these six experimental groups. The conditions of each user group were designed so that we could extract a variety of useful information. It was our hope that comparing the results from certain pairs of groups would reveal valuable information about biofeedback, pervasive computing and human computer interactions. Exhibit 5 depicts some of these key pairings and the findings they might reveal.

By comparing users receiving no feedback and those receiving individual feedback, we hoped to determine the effects of feedback on self health care. Would users interact with the feedback, and use its information to control or even change their daily lifestyle habits? Comparing users receiving individual feedback to those receiving both individual feedback and the study average would then give us insight into "competitive feedback." Would the additional information illustrating one's performance versus the performance of a group of peers affect self health care in any way? Would individuals use this study average as a motivation to change their own habits? Next, evaluating the results of desktop users versus wireless users would give insight into the advantages and disadvantages of wireless computing. Was one platform preferred over another, and for what reason? Finally, comparing wireless users with limitations on use to those with no limitations would allow us to evaluate the type of relationships users will form with their wireless devices. How much will users trust their devices, and how much personal information will they be willing to share with them?

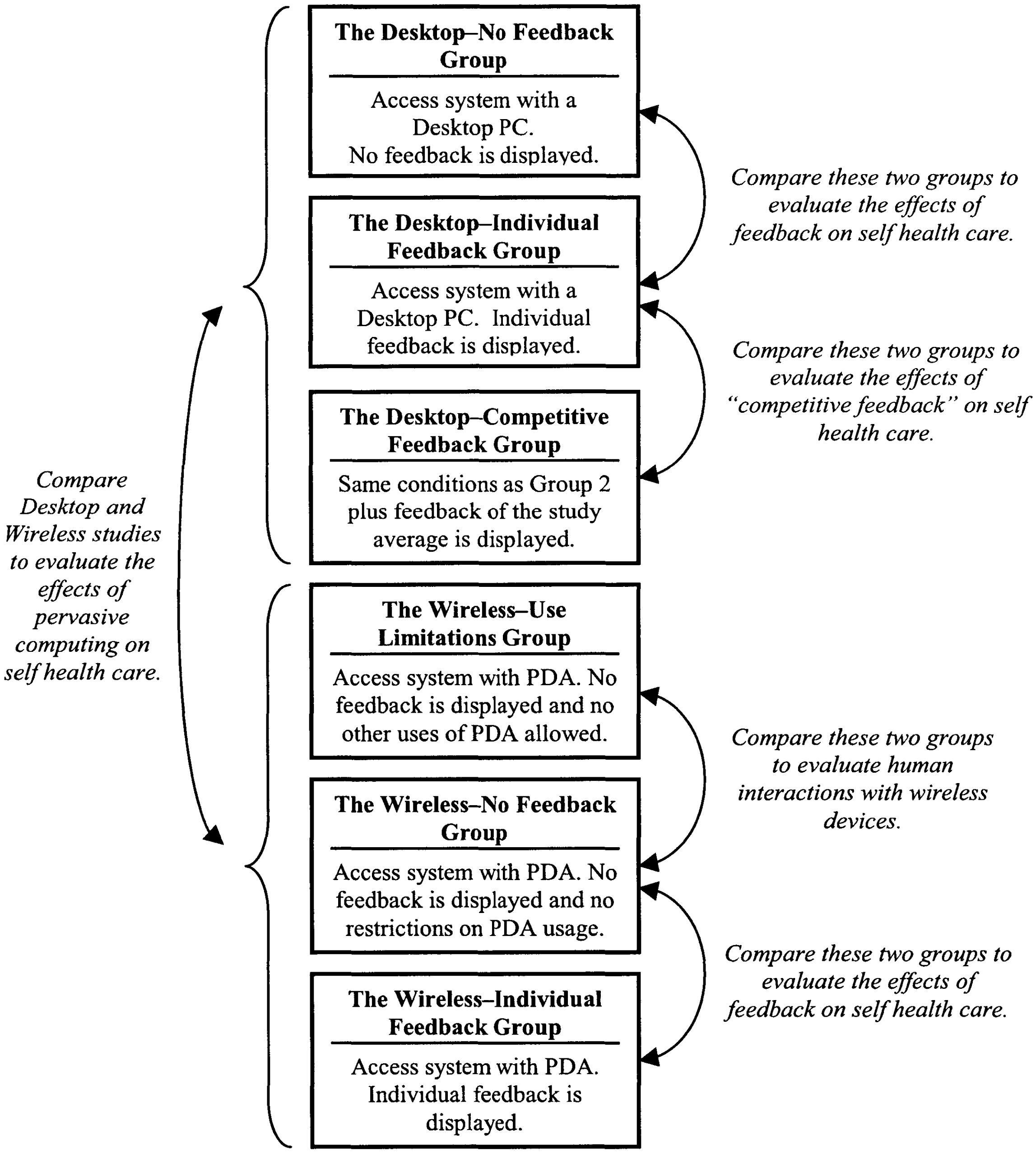

The Desktop–No Feedback Group
Access system with a Desktop PC. No feedback is displayed.
The Desktop–Individual Feedback Group
Access system with a Desktop PC. Individual feedback is displayed.
The Desktop–Competitive Feedback Group
Same conditions as Group 2 plus feedback of the study average is displayed.
The Wireless–Use Limitations Group
Access system with PDA. No feedback is displayed and no other uses of PDA allowed.
The Wireless–No Feedback Group
Access system with PDA. No feedback is displayed and no restrictions on PDA usage.
The Wireless–Individual Feedback Group
Access system with PDA. Individual feedback is displayed.
Compare these two groups to evaluate the effects of feedback on self health care.
Compare these two groups to evaluate the effects of "competitive feedback" on self health care.
Compare these two groups to evaluate human interactions with wireless devices.
Compare these two groups to evaluate the effects of feedback on self health care.
Compare Desktop and Wireless studies to evaluate the effects of pervasive computing on self health care.

Exhibit 5: Key Relationships Amongst the HealthTracker Experimental Groups. Important relationships between each test group are illustrated in this figure. These relationships, as well as others will be further elaboarted upon in Section 10.

7.2 Procedure

The entire study lasted six weeks for desktop users, who started in mid-March 2001, and four weeks for wireless users, who started the experiment in late-March. All subjects completed the experiment on April 30, 2001.

7.2.1 Orientation Meeting

The study began with two weeks of orientation meetings. Orientations for participants in the desktop study ran from March 12, 2001 to March 15, 2001. Orientations for wireless users ran from March 19, 2001 to March 22, 2001. Every participant participated in an orientation meeting specific to his or her user group. At these meetings, participants were presented with a general overview of the study and informed of their specific role in the experiment and only their roles. So for example, participants in the Desktop–No Feedback Group did not know that other subjects were accessing HealthTracker with wireless devices or that other subjects were receiving various forms of graphical feedback. Limiting participants to only what they needed to know for their user group preserved the validity of the study's data.

All participants received instructions on how to access and use the HealthTracker website. Users were encouraged to access the HealthTracker website between the hours of 8:00pm and 11:00pm as it was then that they would have the best opportunity to reflect on the current day and provide complete answers to the HealthTracker questionnaire. Additional information was also provided describing how the $50 compensation would be distributed. Appendix A contains a copy of these instructions for desktop users and Appendices B, C and D present copies of these instructions for the three wireless groups.

Wireless users received additional information pertaining to their devices. Wireless users were shown in detail how to operate their Handspring Visors and wireless modems. To make sure that everybody knew how to operate their devices and access the HealthTracker system, everybody had to successfully log in to the website before leaving the orientation meeting. A chart of Palm's text input interface, Graffiti™, was included as well. Finally, participants in the Wireless–No Feedback Group and the Wireless–Individual Feedback Group received suggestions on other ways they could use their device. These participants were encouraged to experiment with the included docking cradle that connected a desktop computer and allowed for synchronization and Web downloads. The Wireless–No Feedback Group and the Wireless-

Individual Feedback Group users were ultimately encouraged to interact with their devices as much as possible.

The next major component of the orientation meeting was an initial questionnaire. The same questionnaire was given to all six groups and a copy of it can be found in Appendix E. In short, the questionnaire asked general questions about users' health and wellness habits. The survey contained questions on sleep, eating habits, diet, physical activity and mental health. Most questions asked subjects to respond using a scale from 0 to a 100. This 101-point scale was designed to increase the granularity of the data. The questionnaire also inquired whether subjects believed a wireless device could affect their physical or mental health. The survey ended with four short answer questions that asked users about their motivations for exercising, eating and being healthy, and for participating in the HealthTracker study.

Two questions about caffeine and alcohol consumption (questions 6 and 7) asked users to express their answers in terms of units as defined in Exhibit 6. Three questions regarding diet (questions 3, 4 and 5) asked subjects to rate their perceived intake of particular foods on a scale of 0 to 100 where 0 represented none, 50 represented the "recommended amount" and 100 represented "twice as much as needed." Subjects were told to use their own judgment to determine what their personal "recommended amount" was and to then adhere to that standard for all future questions for the duration of the study.

Caffeine Units Conversion			Alcohol Units Conversion		
one 12 oz. can of soda	=	1 unit	one 12 oz. can of beer	=	1 unit
one Vivrin caffeine pill	=	1 unit	one glass of wine	=	1 unit
one 7 oz. cup of coffee	=	2 units	one shot of liquor	=	1 unit
one 7 oz. cup of tea	=	1 unit			

Exhibit 6: Units Definitions for Questions About Caffeine and Alcohol Consumption. In all of the questionnaires, there were questions about caffeine and alcohol consumption. This table was included in all questionnaires so that users could accurately convert their answers into the appropriate units.

Questions 34 through 37 were measured using two medical instruments purchased from the Sharper Image. The first device, the Lifewatch Blood Pressure Wrist Monitor, measured pulse rate and blood pressure. The second device, the Tanita Body Fat Monitor/Scale, measured weight and body fat percentage using electrostatics.

Finally, all participants were asked to sign a consent form that required subjects to satisfactorily complete the study and be in compliance with their group's conditions in order to qualify for the $50 compensation and the raffling off of the free Visors. Appendix F contains an example of this consent form.

7.2.2 Debrief Meeting

At the end of the experiment, each subject was scheduled for a 30-minute meeting with other members from their own study group. These meetings took place from May 1, 2001 to May 4, 2001. To begin the debrief meeting, subjects were given a general questionnaire (Appendix J) that asked questions pertaining to physical health, eating habits, diet and mental wellness. The questionnaire also asked about HealthTracker's effectiveness in making subjects more conscious about their health and well-being, as well as the system's success in actually affecting change in users' daily lifestyle. The survey then concluded with a series of short answer questions that asked users to elaborate on what they liked and disliked about HealthTracker, as well as the system's overall usefulness and user-friendliness.

Following this general questionnaire, subjects were then given a supplementary survey that asked questions specific to their study group. For individuals receiving feedback, this survey asked subjects to comment on the usefulness of the feedback and its effect on them. Subjects were also asked to report any complaints or problems they had with the feedback and provide suggestions for future improvement. Users not receiving any kind of feedback were asked if they would have liked to have feedback and if that additional information would have had any effect on them. All wireless users were also asked about their PDA experience. Wireless-Use Limitations subjects were asked if the restrictions placed upon them had been frustrating and if the limitations had affected their overall HealthTracker experience. The other wireless users who had unlimited use of their PDA's were asked if they had developed an attachment to their devices, and if this relationship had affected their overall HealthTracker experience.

The purpose of all questionnaires administered in the debrief meeting was to supplement the data received from the HealthTracker questionnaires with actual testimonies from the subjects themselves. The hope was that significant correlations would be found between what the subjects perceived and what their data illustrated.

At the conclusion of the debrief meeting, users who had completed at least 75% of the daily questionnaires were awarded their $50 compensation. Subjects who completed between 40% and 75% were awarded $20 compensation. The four winners of the free Handspring Visor were randomly selected by an independent party one week after the debrief meetings.

7.3 The Daily Questionnaire

The central component of the HealthTracker system was an online questionnaire that was to be answered on a daily basis, either via a desktop computer or one of the Visor PDA's. The website could be accessed with any standard web browser.

The first page of the website was a login page. (See Exhibit 7 and Exhibit 8) Each user was assigned a unique login name and password that allowed them to enter the system. Upon successful login, the system would take the user to the daily questionnaire. (See Exhibit 9 and Exhibit 10) For users receiving individual feedback, this questionnaire also included a graph displaying the users' performance over time. (See Exhibit 11 and Exhibit 12) For users receiving competitive feedback, a graph was displayed that charted their performance versus the study average. (See Exhibit 13) The questions in the daily questionnaire remained the same throughout the course of the study and asked users about their current health, eating habits, diet, physical fitness and mental health.

The structure of the questions was also identical to that of the orientation questionnaire. Users were asked to reply to most questions using a numerical scale from 0 to 100 where 0 and 100 represented two extremes. A copy of this questionnaire can be found in Appendix G.

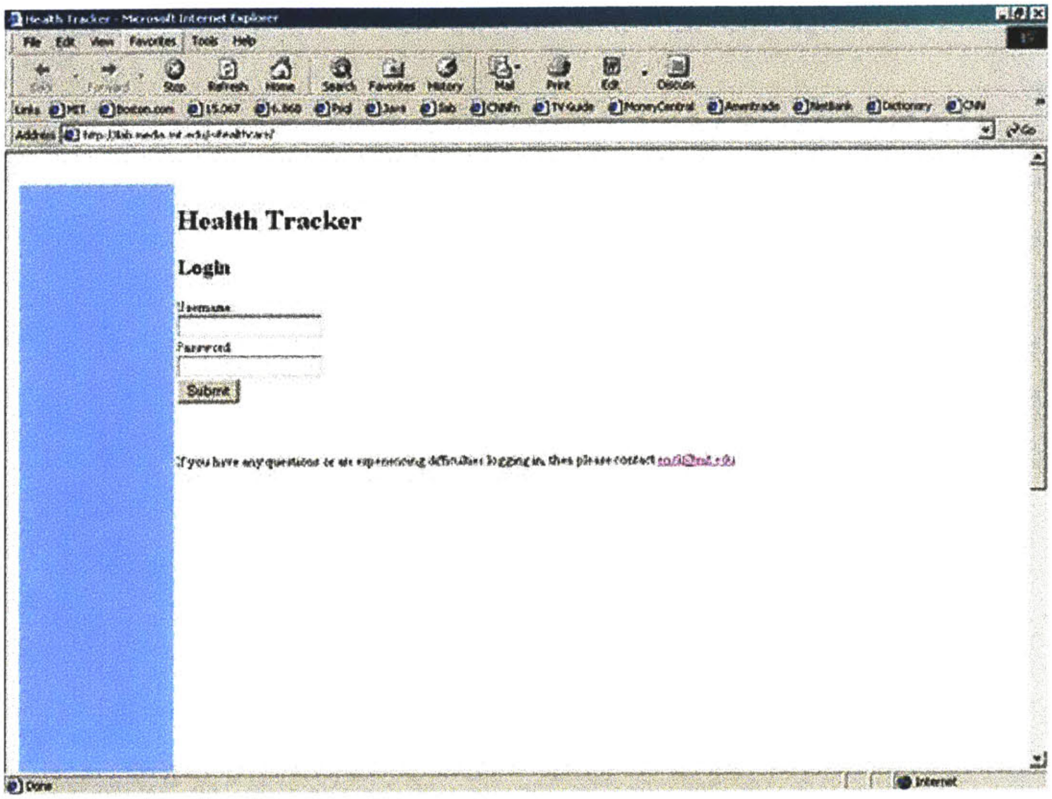
If users were in a study group that received feedback, their feedback graphs were placed adjacent to the questions to encourage subjects to view the graphs as they were filling out their responses to the daily survey. The hope was that the feedback would be useful in helping subjects monitor and control their health and well-being. The graphs were updated every time a user logged in, so they always reflected the most recent set of data.

By default, a subject's responses on the previous day were displayed for each question. There were two purposes to this feature. First, these "previous day defaults" allowed participants to answer the questionnaire as quickly as possible. Some of the questions, like alcohol consumption and number of cigarettes smoked, did not apply to many of our subjects. If this was the case, participants could just skip these questions since all the questions started out with zeros by default. So if a person never smoked or drank alcohol, then that default of zero would be passed on from day to day as the user kept skipping that question. Furthermore, users could periodically skip questions that either changed infrequently, like weight, or required time to measure, like one's resting pulse. Day-to-day fluctuations in these metrics are not as important as longer term changes, so using the same values two or three days in a row was acceptable.

The second benefit of displaying "previous day defaults" is that they in themselves served as a form of feedback. Users were able to look at their responses from the previous day and compare them to their responses for the present day. These comparisons allowed subjects a chance to reflect on possible changes in their health and wellness.

In the event that subjects forgot to answer a questionnaire, subjects could make them up by logging in and changing the month and date fields of the questionnaire as appropriate. Subjects were asked to do this only if they could accurately remember their responses for that day in question however. To keep track of when users really answered questionnaires, a computer timestamp was attached to every questionnaire submission. This timestamp included the current date and time on the web server.

The $50 compensation and chance to win a Handspring Visor served as the incentives for participants to diligently fill out as many questionnaires as possible. Subjects were told that these rewards would only be available if they were in compliance with the study's conditions and answered a significant majority of the daily questionnaires.


Health Tracker - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Links MIT Boston.com 15.067 6.868 Pod Java Sab CNNfn TV Guide MoneyCentral Ameritrade NetBank Dictionary CNN
Address
Go
Done
Internet

Health Tracker

Login

Username

Password

Submit

If you have any questions or are experiencing difficulties logging in, then please contact [illegible]

Exhibit 7: HealthTracker Login Page for Desktop Users. This is the opening page of the HealthTracker system for desktop computers. Users had to submit their assigned username and password to enter the system. Important announcements and notices of server failures were also posted on this page.

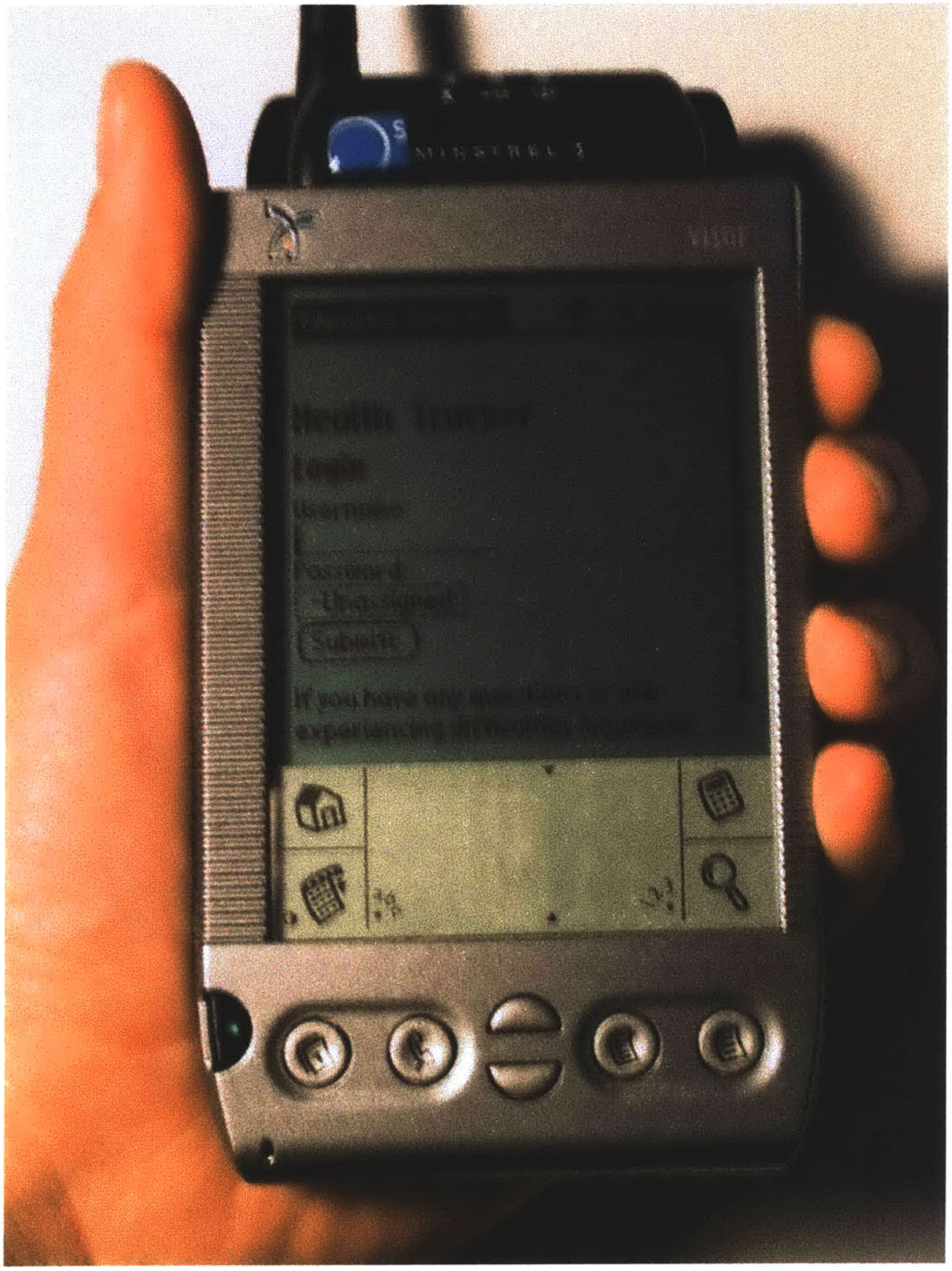
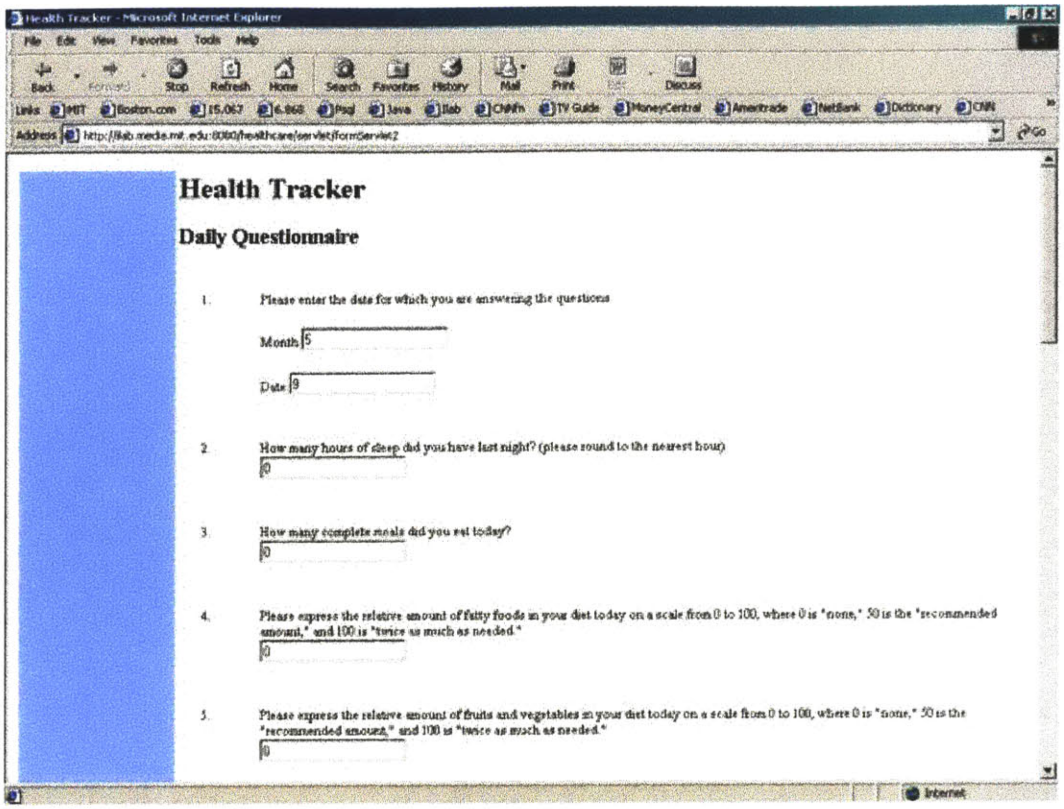


Exhibit 8: HealthTracker Login Page for Wireless Users. This is the opening page of the HealthTracker system for Handspring Visor PDA's. Users had to submit their assigned username and password to enter the system. Important announcements and notices of server failures were also posted on this page.


Health Tracker - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Links MIT Boston.com 15.067 6.868 Psql Java Ilab CNNfn TV Guide MoneyCentral Ameritrade NetBank Dictionary CNN
Address http://ilab.media.mit.edu:8080/healthcare/servlet/formServlet2
Go
Internet

Health Tracker

Daily Questionnaire

1. Please enter the date for which you are answering the questions

   Month 5

   Date 9

2. How many hours of sleep did you have last night? (please round to the nearest hour)

   0

3. How many complete meals did you eat today?

   0

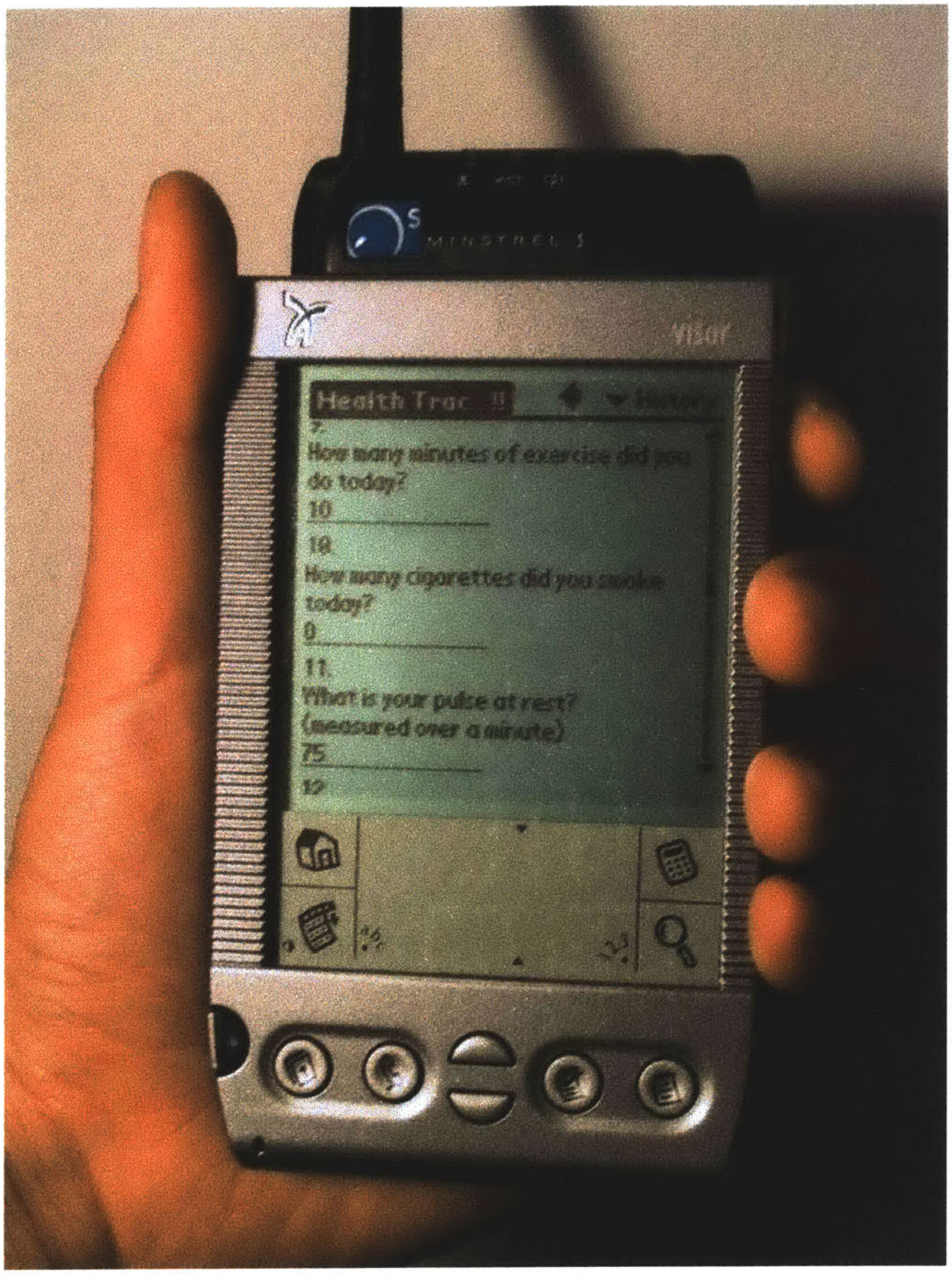
4. Please express the relative amount of fatty foods in your diet today on a scale from 0 to 100, where 0 is "none," 50 is the "recommended amount," and 100 is "twice as much as needed."

   0

5. Please express the relative amount of fruits and vegetables in your diet today on a scale from 0 to 100, where 0 is "none," 50 is the "recommended amount," and 100 is "twice as much as needed."

   0

Exhibit 9: HealthTracker Daily Questionnaire with no Feedback for Desktop Users. This is a screen shot of the HealthTracker Daily Questionnaire for desktop users not receiving feedback. This 20-question survey was to be answered on a daily basis and inquired users about their physical health, eating habits, diet and mental health.



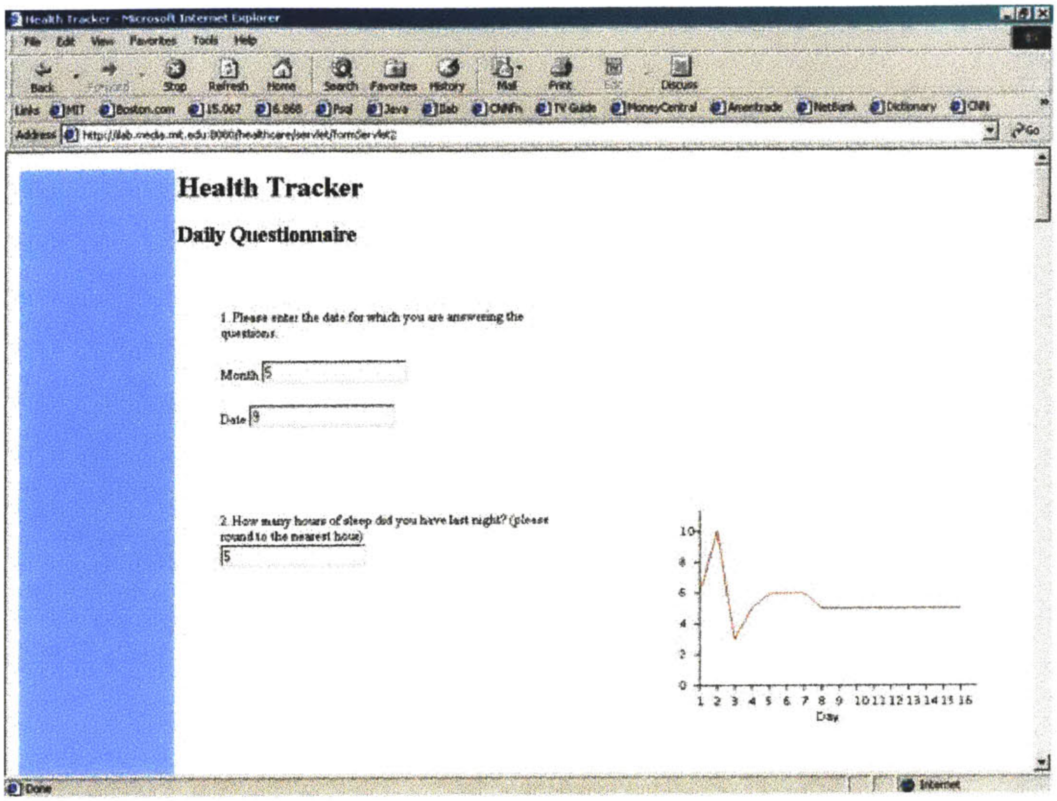
9.
How many minutes of exercise did you do today?
10

10.
How many cigarettes did you smoke today?
0

11.
What is your pulse at rest? (measured over a minute)
75

12

Exhibit 10: HealthTracker Daily Questionnaire with no Feedback for Wireless Users. This is a picture of the HealthTracker Daily Questionnaire for wireless users not receiving feedback. This 20-question survey was to be answered on a daily basis and inquired users about their physical health, eating habits, diet and mental health.



Health Tracker - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Forward
Stop
Refresh
Home
Search
Favorites
History
Mail
Print
Edit
Discuss
Links MIT Boston.com 15.067 6.868 Psql Java llab CNNfn TV Guide MoneyCentral Ameritrade NetBank Dictionary CNN
Address http://llab.media.mit.edu:8080/healthcare/servlet/formServlet2
Go
10
8
6
4
2
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Day
Done
Internet

Health Tracker

Daily Questionnaire

1. Please enter the date for which you are answering the questions.

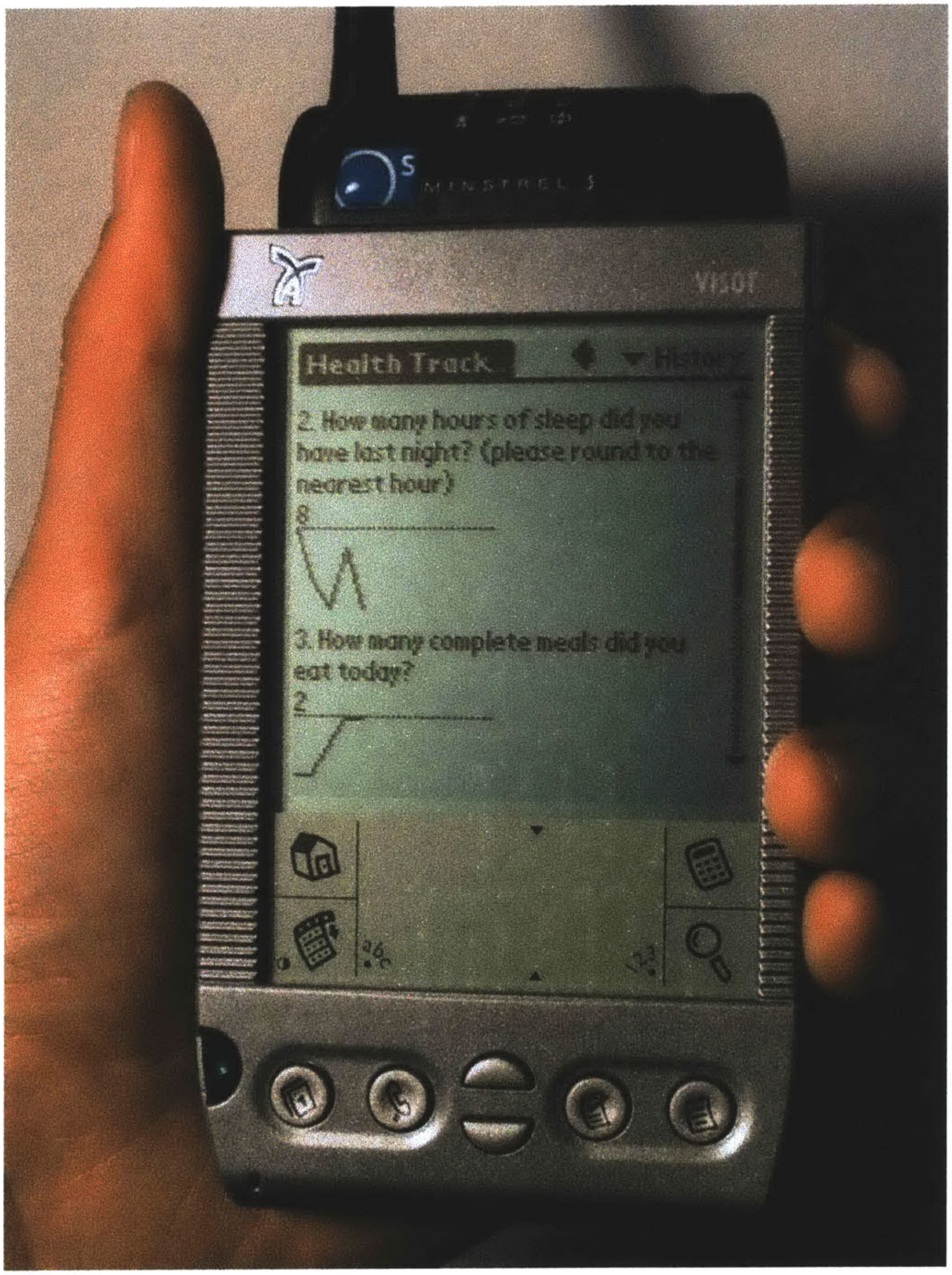
Month 5

Date 9

2. How many hours of sleep did you have last night? (please round to the nearest hour)

5

Exhibit 11: HealthTracker Daily Questionnaire with Individual Performance Feedback for Desktop Users. This is a screen shot of the HealthTracker Daily Questionnaire for desktop users receiving feedback on their individual performance. This 20-question survey was to be answered on a daily basis and inquired users about their physical health, eating habits, diet and mental health. The graph to the right is an example of the graphical feedback users received for each question. The graph illustrates all of the users' responses since the beginning of the study.



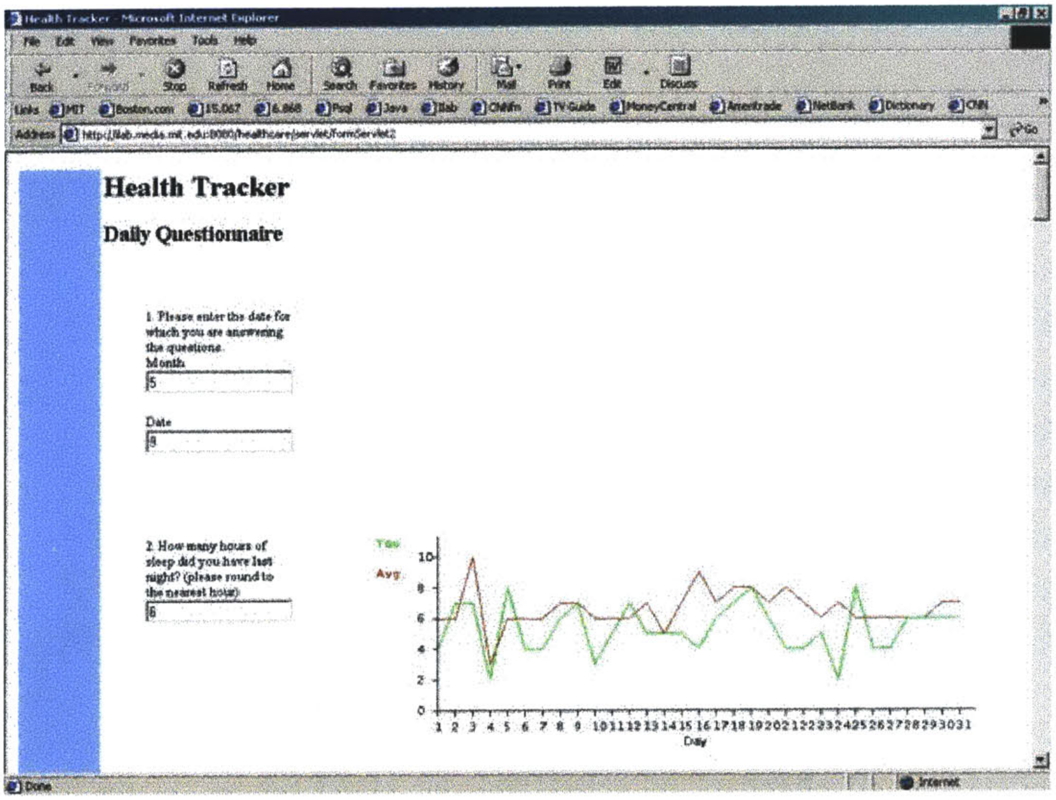
2. How many hours of sleep did you have last night? (please round to the nearest hour)

8

3. How many complete meals did you eat today?

2

Exhibit 12: HealthTracker Daily Questionnaire with Individual Performance Feedback for Wireless Users. This is a picture of the HealthTracker Daily Questionnaire for wireless users receiving feedback on their individual performance. The graphs at the bottom of each question are examples of the graphical feedback users received for each question.


Health Tracker - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Links MIT Boston.com 15.067 6.868 Psql Java Ilab CNNfn TV Guide MoneyCentral Ameritrade NetBank Dictionary CNN
Address http://ilab.media.mit.edu:8080/healthcare/servlet/formServlet2
Go
Done
Internet

Health Tracker

Daily Questionnaire

1. Please enter the date for which you are answering the questions.
Month
5

Date
8

2. How many hours of sleep did you have last night? (please round to the nearest hour)
6

You
Avg
10
8
6
4
2
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Day

Exhibit 13: HealthTracker Daily Questionnaire with Competitive Feedback, Individual Performance Feedback versus the Study Average. This is a screen shot of the HealthTracker Daily Questionnaire for users receiving feedback on their individual performance and the study average. This 20-question survey was to be answered on a daily basis and inquired users about their physical health, eating habits, diet and mental health. The graph to the right is an example of the graphical feedback users received for each question. The graph illustrates all of the users' responses in green and the study average in red.

7.4 The Midterm Questionnaire

On April 18, 2001, about halfway through the course of the experiment, an optional "midterm questionnaire" was integrated into the HealthTracker system. This questionnaire was to be answered only once during the study and was displayed immediately after a user logged in and before the daily questionnaire was displayed. It was made clear at the beginning of the questionnaire that any or all of the questions could be skipped.

The questionnaire was divided up into four tiers, with each successive tier asking more and more personal questions about familial medical history. Specifically, users were asked to report if members of their immediate family (mother, father or siblings) or they themselves suffered from any of several medical conditions. These conditions ranged from things like allergies, arthritis and back pain to more personal ailments such as depression, heart disease and dyslexia, to even more personal conditions like cancer, AIDS, mental illness and herpes. For a copy of the actual midterm questionnaire, please refer to Appendix H.

The primary purpose of this midterm questionnaire was to evaluate how many questions, as well as which ones, users would respond to. The survey was designed so that all users could mark at least one box for every question. If no one in the family suffered from a particular ailment, for example, then the "none" box could be checked. Upon completion of the study, the number of skipped questions, identified as questions with no checked boxes, was then counted. This data would then be used to examine the relationship and level of trust between users and the HealthTracker system.

Besides giving us insight into the current level of information-sharing between subjects and HealthTracker, the midterm questionnaire also served a secondary purpose. Medical monitoring systems like HealthTracker understandably operate better when equipped with background information like medical histories. So the midterm survey also served as a way to extract this type of information informally from the subjects in order to improve the system's services.

7.5 The Final Questionnaire

On April 28, the weekend before the study ended, a final optional questionnaire was incorporated into HealthTracker. Like the midterm questionnaire, this survey was also to be answered only once during the study. If users had already answered the midterm questionnaire, the final survey was to be displayed immediately after login and before the daily questionnaire. Had users not yet answered the midterm survey, the final questionnaire fell between the midterm and daily questionnaire. Exhibit 14 portrays the progression of these pages as a flow chart.

The purpose of the final survey was to again examine how close subjects had become with HealthTracker. The survey asked the user to perform four basic physical exercises: 1) touching one's toes five times, 2) 10 jumping jacks, 3) jogging in place for one minute and 4) 10 sit-ups. Between each exercise, users were asked to measure their pulse and express how difficult they found the exercise. It was again made explicit at the beginning of the survey that subjects had the option of skipping some or all of the questions. For a copy of the final questionnaire, please see Appendix I.

Like the midterm survey, the final questionnaire's secondary purpose was to informally extract medical information from the subjects with the intention of expanding the system's understanding of its users and ultimately improving its service.

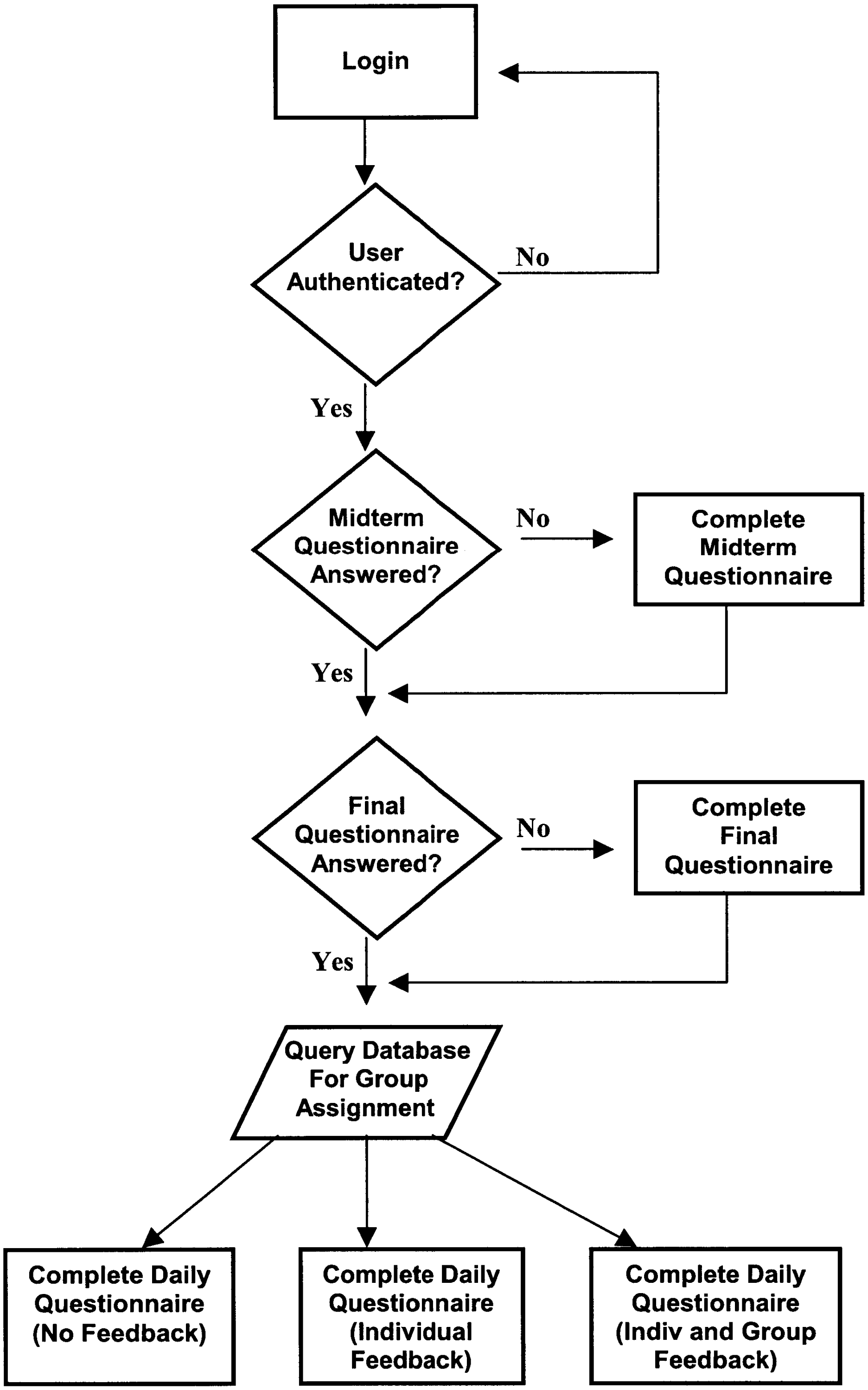

Login
User Authenticated?
No
Yes
Midterm Questionnaire Answered?
No
Complete Midterm Questionnaire
Yes
Final Questionnaire Answered?
No
Complete Final Questionnaire
Yes
Query Database For Group Assignment
Complete Daily Questionnaire (No Feedback)
Complete Daily Questionnaire (Individual Feedback)
Complete Daily Questionnaire (Indiv and Group Feedback)

Exhibit 14: HealthTracker Flow Chart. This diagram illustrates the general processes of the HealthTracker system and the appropriate connections between each questionnaire.

7.6 The HealthTracker Architecture

7.6.1 Client-Server Topology

The HealthTracker system employed a client-server model. A central web server hosted the HealthTracker website and database. The server was a Dell PowerEdge 1300 running on an Intel Pentium III 700Mhz processor with 256K cache. The machine had 64 SDRAM and ran the Linux operating system. The server was shared with another ongoing research project.

Exhibit 15 illustrates this client-server design. The Visors contained a 33Mhz Motorola Dragonball VZ processor, had 8MB of memory and ran on version 3.5.2H of the Palm OS. Each Visor was equipped with an OmniSky™ Minstrel S™ Wireless Modem. The modems accessed the Internet via TCP/IP over the CDPD network and had a maximum data transfer rate of 19.2 kpbs full duplex. The Visors connected to the Web via OmniSky's proprietary web browser and wireless service.

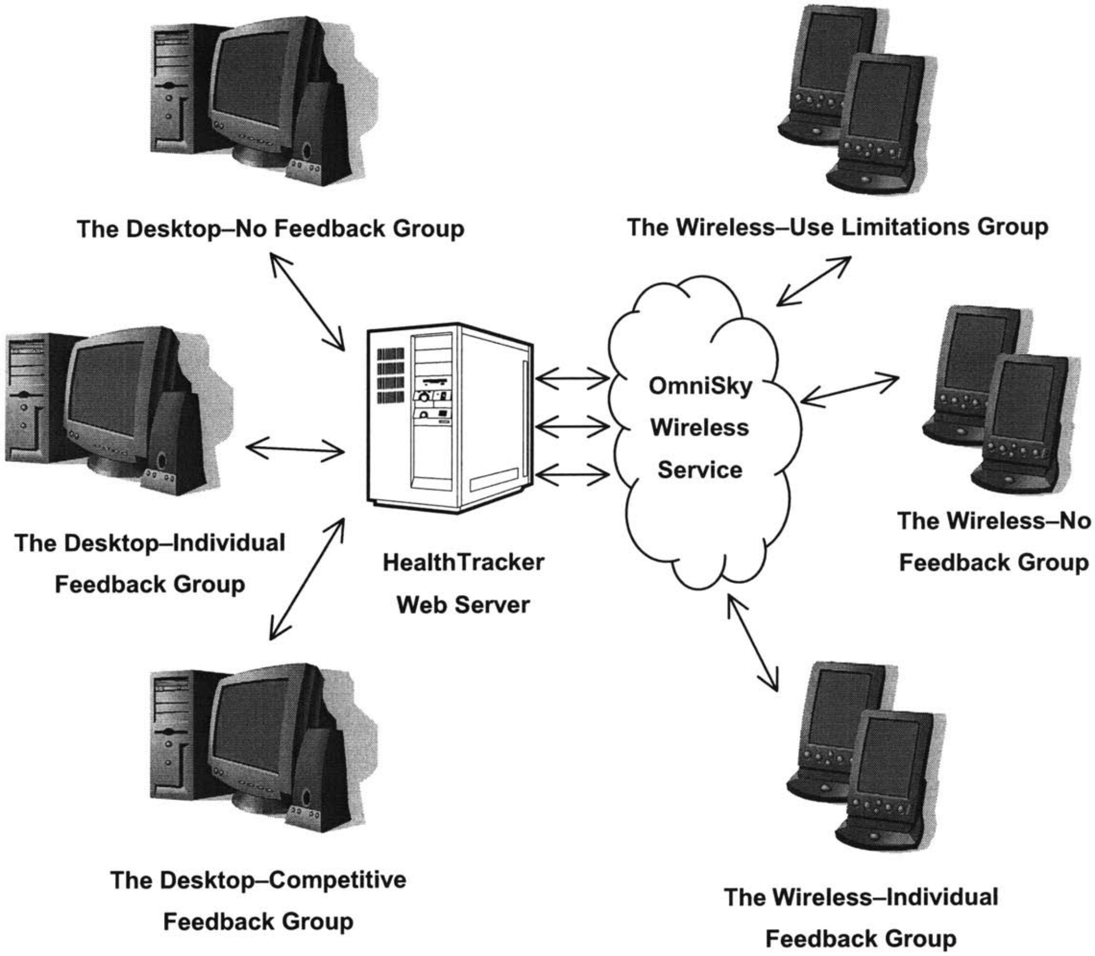


Exhibit 15: HealthTracker Client-Server Architecture. The HealthTracker system employed a client-server model. A central web server hosted the HealthTracker website and database. 30 Handspring Platinum Visors were divided amongst the three wireless user groups and were used to access the HealthTracker system. The Visors accessed the server using OmniSky's wireless service. The 37 other users in the study accessed the system using desktop computers.

7.6.2 The Servlet

The HealthTracker website used an html form interface to administer and collect survey data. The backend of the website was a Java servlet that created dynamic html and processed SQL database queries. The HealthTracker software was written and compiled in Java, using JDK version 1.2.2. The servlet ran on an Apache Tomcat version 3.2.1 engine. JDBC, a database access API for Java, was used to interface between the java servlet and the PostgreSQL database.

A copy of all the core Java source code used in the HealthTracker system can be found in Appendix Q. This collection of code includes source for the servlet, the graphical feedback generation, the database interface and all the online questionnaires.

Building HealthTracker as a servlet had two major advantages. First, the 33 MHz processor and 8MB of memory on board the Visor PDA were not at all sufficient to handle the heavy computing that was required in loading and processing the HealthTracker website. Running the system as a servlet enabled us to relegate all the processor-intensive computing to the server, putting as little burden on the wireless devices as possible. In fact, as far as the Visor was concerned, it was simply loading a basic html file and sending back a simple CGI form. All other processes, including the construction of the feedback graphs, were performed server-side.

The second advantage to running a servlet was security. We strove to make HealthTracker as secure a system as possible. As a servlet, we were able to keep as much of a user's personal data, such as responses to questionnaires and feedback graphs, stored on the server itself. Had we instead built HealthTracker as an applet, for example, we would have been forced to save much more information, like feedback graphs, on local home machines.

The Apache Tomcat servlet engine was used to deliver both the servlet as well as normal html pages, such as the login page. This design allowed us to post announcements of system failures on the login page in the event that the servlet were to crash.

Backing the servlet was a PostgreSQL database. This database stored all user profiles, daily questionnaire submissions and the results of both the midterm and final questionnaires. SQL queries enabled the servlet to pull information from the database for the following purposes:

1) User Authentication – Upon login, participants' username and password are checked against the user profiles for positive verification.
2) Group Assignments – After a successful login, the servlet queries the database for users' group assignments so that it can direct subjects to the appropriate version of the daily questionnaire.
3) Graphical Feedback – For subjects receiving feedback, the servlet requests from the database historical data so that it can draw feedback graphs displaying past performance.

The servlet also updated the database every time a user submitted a questionnaire. This database was backed up regularly on both the HealthTracker server as well as on a remote machine running on a different network. Exhibit 16 illustrates the relevant modules and layers in the HealthTracker architecture.

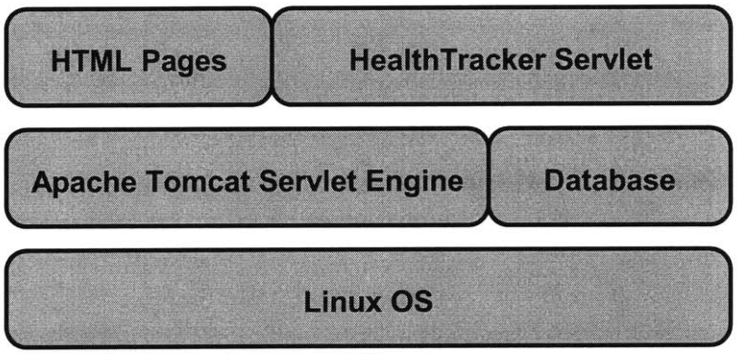


Exhibit 16: Servlet Architecture: The HealthTracker system ran on the Linux platform. An Apache Tomcat Servlet Engine was used to deliver both the HealthTracker servlet as well as normal html pages, such as the login page. The entire system was backed by a PostgreSQL database that stored all user data, login information and questionnaire results.

7.6.3 Generating Graphical Feedback

The Java servlet also generated and displayed the jpeg graphical feedback images. For users who received feedback, the servlets made use of the Grapher and SmallGrapher Java classes (See Appendix Q) to generate jpeg images of feedback graphs. These two classes accept arrays of data representing previous responses made by a given user to a given question. This data is pulled from the database using SQL commands. The historical data is then used to generate a line graph that displays the progress over time of a user for each question on the survey. This image file is saved on the central server and then displayed on the user's browser.

7.6.4 Security

The confidentiality of all the subjects was one of our key concerns and specific features were implemented in the design of the HealthTracker architecture to ensure that user information and questionnaire data remain secure.

The opening login screen prevented unauthorized users from accessing the HealthTracker system. All user information and data was stored in a password-protected database on the HealthTracker server. The server itself also required its own login and password. Finally, the graphs generated for the feedback were stored on the server in yet another password-protected account.

8 Desktop Study Results

8.1 Debrief Questionnaire

Desktop users generally found the HealthTracker system to be useful, as illustrated in Exhibit 17. 70% of the Desktop–No Feedback Group users, 82% of the Desktop–Individual Feedback Group users and 91% of the Desktop–Competitive Feedback Group users found the system to be very or somewhat useful in their lives. What is also very interesting is how the percentage of positive responses increases from the Desktop–No Feedback Group to the Desktop–Individual Feedback Group to the Desktop–Competitive Feedback Group.

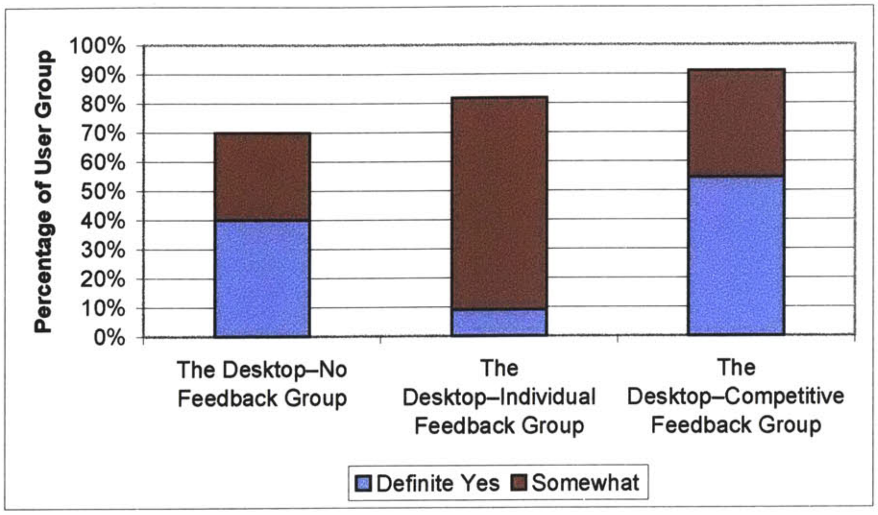


Exhibit 17: Desktop Users' Responses to the Question: "Did you find HealthTracker to be Useful?" Desktop users generally found HealthTracker to be useful. 70% of the Desktop–No Feedback Group users, 82% of the Desktop–Individual Feedback Group users and 91% of the Desktop–Competitive Feedback Group users found the system to be very or somewhat useful in their lives. There is also a distinct positive trend as the graph moves from left to right.

Users commented on how the system increased their awareness of their health habits and wellness lifestyle. Many subjects were pleased with how the system made them consciously monitor important elements like diet and exercise. (See Exhibit 18 and Exhibit 19) Some participants even commented on how they HealthTracker caused them to actively change their diet or go out and exercise more. Said one desktop user, "I was able to check how much time I dedicated to exercise and fitness. If, at the end of the day, I felt as if I hadn't done enough exercise, I could always schedule it for the next day."

A very strong, increasing trend from the Desktop–No Feedback Group to the Desktop–Individual Feedback Group to the Desktop–Competitive Feedback Group in Exhibit 18 also indicates greater effectiveness in the HealthTracker system as more feedback is delivered to the user.

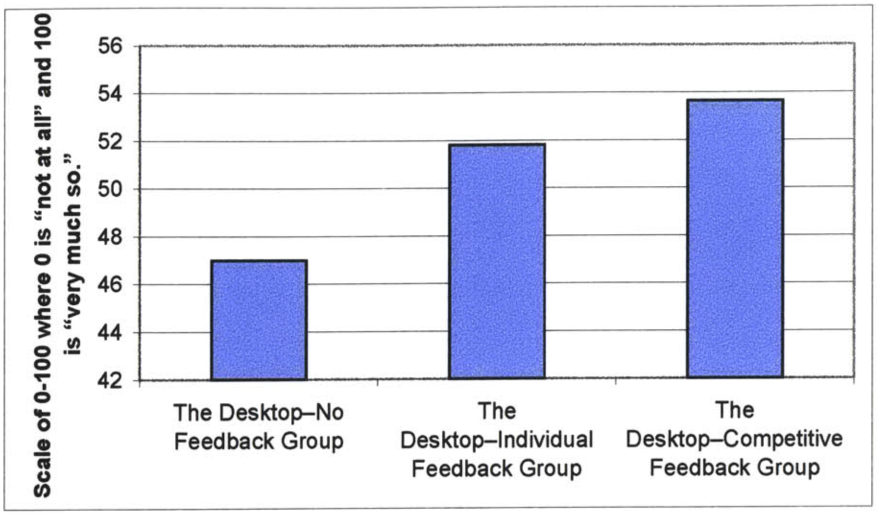


Exhibit 18: How Conscious HealthTracker Made Desktop Users About Their Diet. A strong, increasing trend from the Desktop–No Feedback Group to the Desktop–Individual Feedback Group to the Desktop–Competitive Feedback Group indicates greater effectiveness in the HealthTracker system as more feedback is delivered to the user.

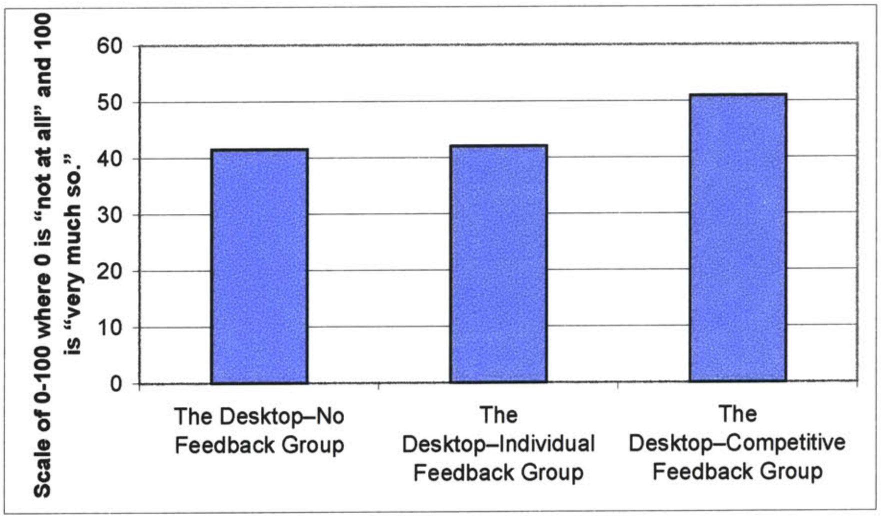


Exhibit 19: How Conscious HealthTracker Made Desktop Users About Their Physical Fitness. Desktop users were generally positive about how the system increased their awareness of their health habits and wellness lifestyle. Many subjects specifically mentioned their satisfaction with how the system made them consciously monitor important elements like diet and exercise.

Another subject described how HealthTracker reduced the amount of snack food she ate at night. After answering the questionnaire in the evening, this individual became very conscious of her eating habits and how much she had already consumed during the day, prompting her to watch her diet by not snacking in the evening.

Many participants receiving feedback mentioned how they enjoyed looking for patterns in their graphs and would act on trends that bothered them, for example increases in fatty food consumption.

Response to HealthTracker's effectiveness at making subjects more conscious about mental health was generally not as strong. As illustrated in Exhibit 20 and Exhibit 21, responses to questions examining HealthTracker's efficacy in general happiness and mental health scored twenty to forty index points lower than diet and exercise across all three groups. Nevertheless, there was once again a strong relationship between the three desktop groups in which the amount of positive reaction rose as the amount of feedback users received increased.

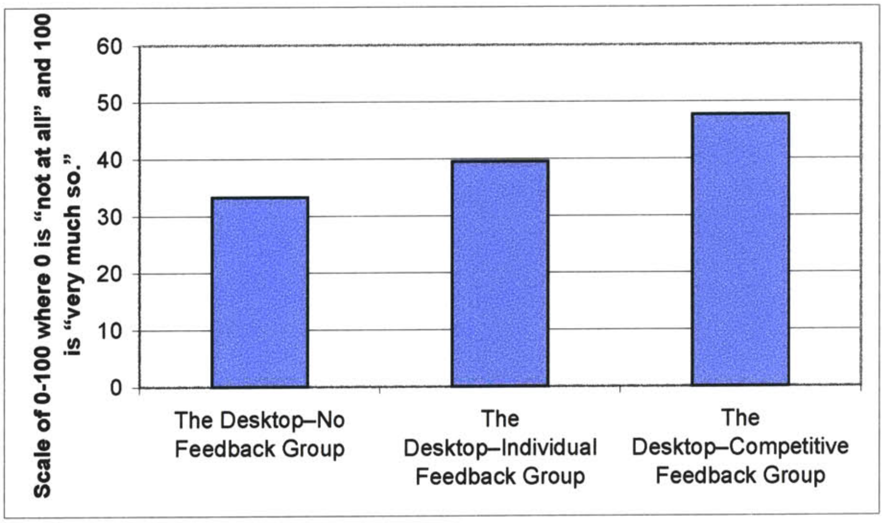


Exhibit 20: How Conscious HealthTracker Made Desktop Users About Their General Happiness. Response to HealthTracker's effectiveness at making subjects more conscious about mental health was generally not very strong. Nevertheless, there was once again a strong relationship between the three desktop groups in which the amount of positive reaction rose as the amount of feedback users received increased.

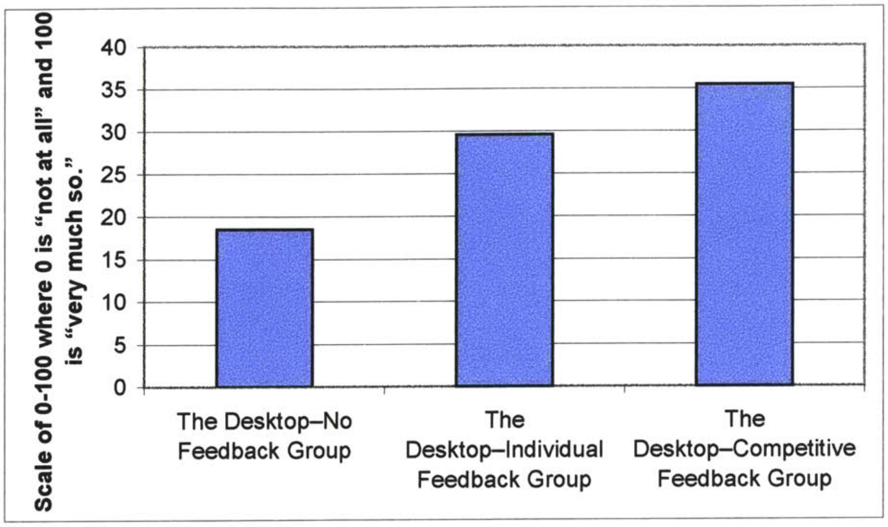


Exhibit 21: How Conscious HealthTracker Made Desktop Users About Their Mental Health. Response to HealthTracker's effectiveness at making subjects more conscious about mental health was generally not very strong. Nevertheless, there was once again a strong relationship between the three desktop groups in which the amount of positive reaction rose as the amount of feedback users received increased.

Users who found HealthTracker to be less than helpful had a number of reasons. Some found the system to be an interesting concept and maybe effective in making them more conscious of their health, but decided the system was ineffective since it didn't actually make them change any of their habits. Some users in the Desktop–No Feedback Group suggested that integrating feedback into HealthTracker in some fashion would have significantly improved the efficacy of the system. Note that these users were not cognizant of the fact that other user groups in the experiment *were* receiving feedback. Many subjects said that lack of time prevented them from really trying to engage with the system.

Finally, a number of users suggested that HealthTracker would be extremely effective if it were set up with personalized goals established with the assistance of a personal physician. HealthTracker's role would be to allow an individual to monitor his or her progress towards these goals and perhaps even offer encouragement or professional advice along the way.

There was general consensus that HealthTracker was user-friendly, meaning that the website was easy to use and the interface was both intuitive and appealing. (See Exhibit 22) Users did disagree on an appropriate length for the questionnaire. Some thought the 20-question survey was just the right length, while others would have preferred more brevity.

The biggest reasons users found HealthTracker to be somewhat difficult to use were technical. Subjects described how they often had difficulty logging in and encountered occasional server errors towards the beginning of the experiment. These technical issues will be discussed shortly in Section 10.5.

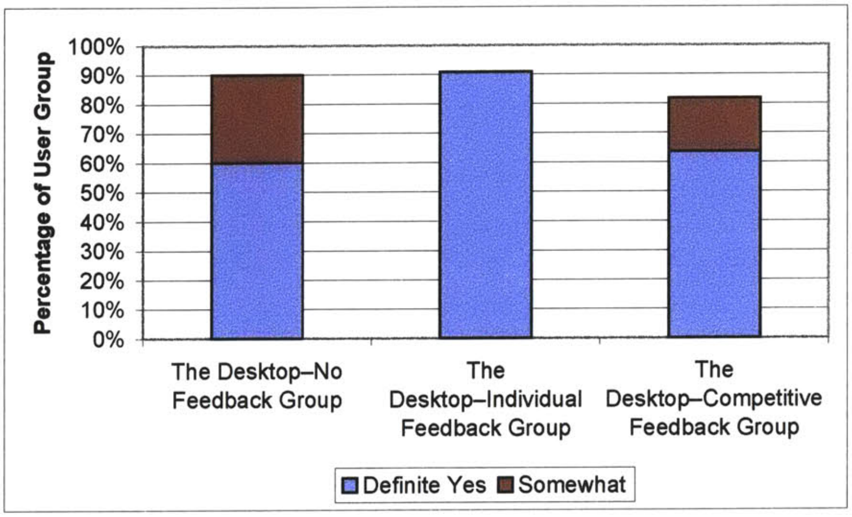


Exhibit 22: Desktop Users' Responses to the Question: "Did you find HealthTracker to be User-Friendly?" There was general consensus amongst desktop subjects that HealthTracker was user-friendly, meaning that the website was easy to use and the interface was both intuitive and appealing.

Responses were more mixed as to whether or not subjects would continue using the HealthTracker system if it were available. It is again interesting to note an increasing trend in the positive responses from the Desktop–No Feedback Group to the Desktop–Individual Feedback Group to the Desktop–Competitive Feedback Group. (See Exhibit 23)

Additional data from the Debrief Questionnaire can be found in Appendix S.

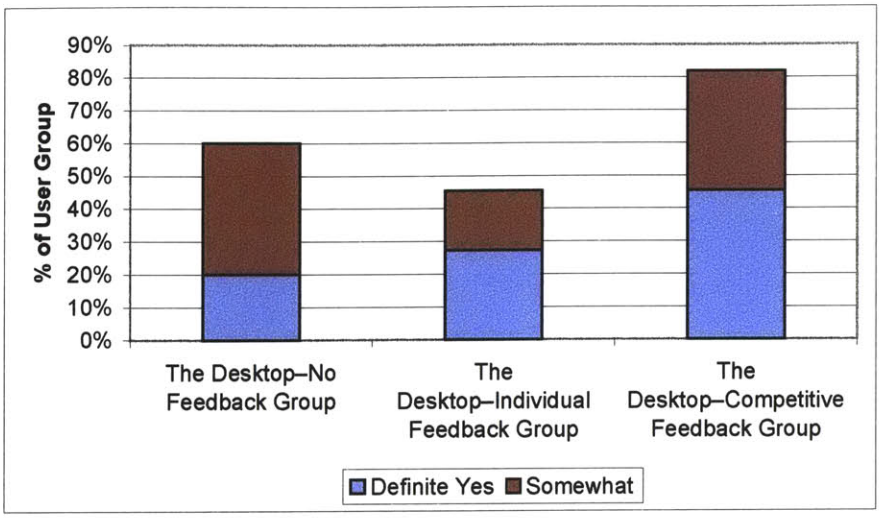


Exhibit 23: Desktop Users' Responses to the Question: "Would you continue using the HealthTracker system?" Though responses to this question were more mixed, it is again interesting to note an increasing trend in the positive responses from the Desktop–No Feedback Group to the Desktop–Individual Feedback Group to the Desktop–Competitive Feedback Group.

8.2 Final Questionnaire

Due to technical and timing problems that will be explained in detail in Section 10.5.4, data from the final questionnaire ended up being incomplete and unusable.

8.3 Midterm Questionnaire

The central purpose of the midterm questionnaire was to evaluate how many questions, as well as which ones, users would respond to. The survey was designed so that all users could enter an answer for every question. After the HealthTracker experiment ended, the number of questions that were skipped and left unanswered was then counted. This process was performed for every tier of questions for every group. Recall that each progressive tier asked increasingly personal questions about users' familial medical history. So while tier one questions inquired about allergies and back pain, tier four questions asked about mental illness and AIDS in the family.

Exhibit 24 presents the percentages of each tier of the midterm survey completed by desktop users. As the graph indicates, the three desktop user groups responded to nearly all questions in all four tiers. No group answered fewer than 89%, on average, of any tier of questions.

Furthermore, there really is no significant distinction in the reply percentages across the four tiers within each group. The standard deviations across the four tiers are 1%, 4% and 3% for the Desktop–No Feedback Group, the Desktop–Individual Feedback Group and the Desktop–Competitive Feedback Group respectively.

There is just a slight downward trend in the graph from the Desktop–No Feedback Group to the Desktop–Individual Feedback Group to the Desktop–Competitive Feedback Group, indicating that the more feedback users received from HealthTracker, the more questions they skipped in the midterm questionnaire. This slope is extremely small, as evidenced by a standard deviation of only 4% across the three groups' averages.

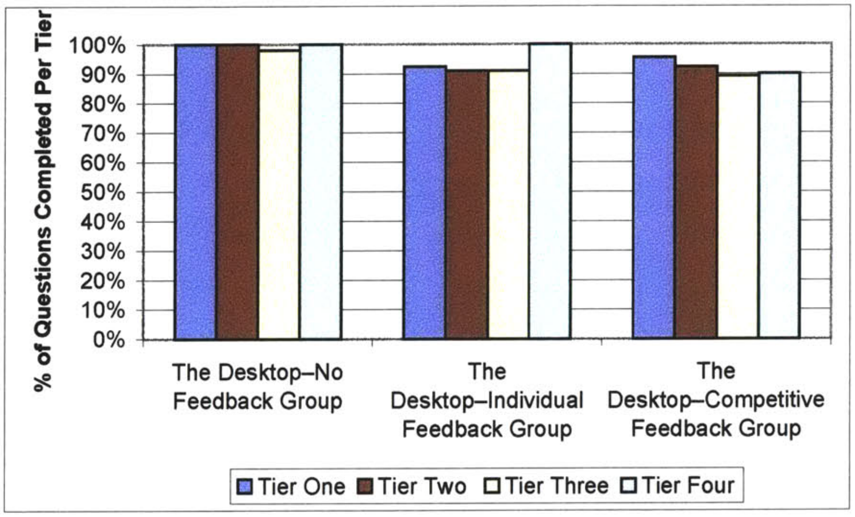


Exhibit 24: Percentage of Midterm Questionnaire Completed by Desktop Users. The three desktop user groups responded to nearly all questions in all four tiers. No group answered fewer than 89%, on average, of any tier of questions. Furthermore, there really is no significant distinction in the reply percentages across the four tiers within each group.

8.4 Daily Questionnaire

8.4.1 Compliance

Desktop users exhibited a relatively high level of compliance, defined as the percentage of completed daily questionnaires. (See Exhibit 25) These group averages reflect the omission of one subject from the Desktop–No Feedback Group and another participant from the Desktop–Competitive Feedback Group who both withdrew from the study in its first week. The Desktop–No Feedback Group users answered, on average, 83.0% of the daily questionnaire. The Desktop–Individual Feedback Group users answered 77.3% of the survey and the Desktop–Competitive Feedback Group users completed 76.0%. There is just a slight decreasing trend in the level of compliance from the Desktop–No Feedback Group to the Desktop–Individual Feedback Group to the Desktop–Competitive Feedback Group.

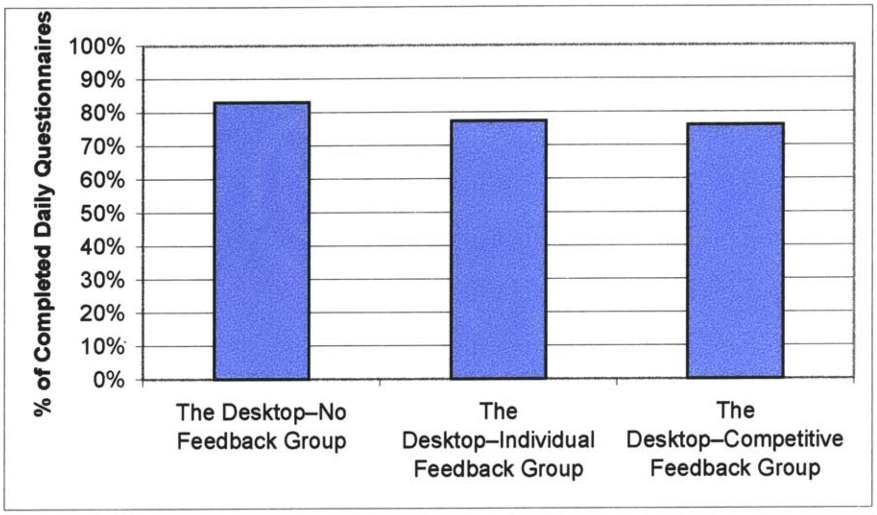


Exhibit 25: Desktop Users Compliance in the Daily Questionnaire. Desktop users exhibited a relatively high level of compliance, defined as the percentage of completed daily questionnaires. There is also just a slight decreasing trend in the level of compliance from the Desktop–No Feedback Group to the Desktop–Individual Feedback Group to the Desktop–Competitive Feedback Group.

While compliance was relatively high however, the frequency with which desktop users logged into HealthTracker was rather low. (See Exhibit 26) All three desktop user groups, on average, logged into HealthTracker for only half of the days of the experiment, or every other day. The Desktop–No Feedback Group users logged in on only 53% of the total days in the study, while the Desktop–Individual Feedback Group and the Desktop–Competitive Feedback Group logged in 52% and 51% respectively.

HealthTracker was designed so that users could "make up" questionnaire submissions they had missed from previous days. All users were cautioned to do this only if the questionnaires they were filling out were for relatively recent days. Nonetheless, it appears from this data that, on average, all desktop users filled out two questionnaires every time they logged into HealthTracker.

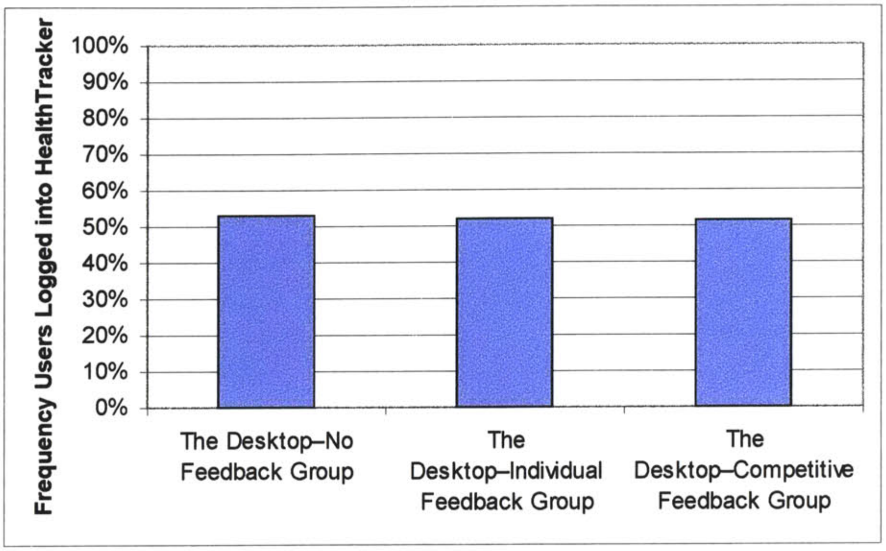


Exhibit 26: How Often Desktop Users Logged Into HealthTracker. While compliance was relatively high however, the frequency with which desktop users logged into HealthTracker was rather low. All three desktop user groups, on average, logged into HealthTracker for only half of the days of the experiment, or every other day. The Desktop–No Feedback Group users logged in on only 53% of the total days in the study, while the Desktop–Individual Feedback Group and the Desktop–Competitive Feedback Group logged in 52% and 51% respectively.

Another useful compliance metric is the number of times users changed their responses from day-to-day. Recall that by default, every question displays a subject's response from the previous day. Therefore, a skipped question can be identified by two identical responses on consecutive response days. Counting these instances of repeated responses can only be an approximate representation of compliance however, since motivations for skipping a question can be varied. Some users might have skipped questions in the interest of time, while others simply had the same answer two days in a row.

Exhibit 27 illustrates the frequency that desktop users changed their responses from day-to-day. To calculate these values, the percentage of times users changed their responses to each individual question was first calculated. These individual percentages were then averaged for each group. Finally, the percentages for all the questions per group were averaged together, producing the data shown in Exhibit 27.

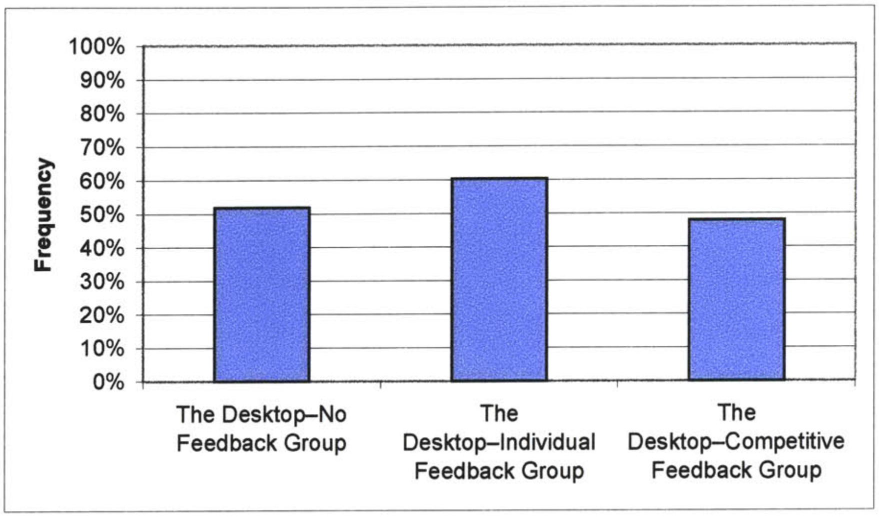


Exhibit 27: Frequency that Desktop Users Changed Responses From Day-to-Day. The Desktop–Individual Feedback Group users made the most day-to-day changes to their questionnaires, modifying 60% of their responses day-to-day on average. The Desktop–No Feedback Group and the Desktop–Competitive Feedback Group users modified their responses 52% and 48% of the time on average. These results indicate that users receiving just individual feedback were more likely to change questionnaire responses, while users receiving both individual and group feedback were least likely to change their data.

The Desktop–Individual Feedback Group users made the most day-to-day changes to their questionnaires, modifying 60% of their responses day-to-day on average. The Desktop–No Feedback Group and the Desktop–Competitive Feedback Group users modified their responses 52% and 48% of the time on average. These results indicate that users receiving just individual feedback were more likely to change questionnaire responses, while users receiving both individual and group feedback were least likely to change their data.

8.4.2 Effects of the Daily Questionnaire

The results of the HealthTracker daily questionnaire revealed a number of distinguishable trends over time. Consumption of fruits and vegetables rose slightly for all desktop users. (See Exhibit 28) The most significant increases were in the Desktop–No Feedback Group and the Desktop–Competitive Feedback Group. For example, the first week average index for the Desktop–No Feedback Group was 39.6, while the group ended the last week with an average index of 44.8. Meanwhile, the Desktop–Individual Feedback Group's consumption of fatty foods had a notable decrease, dropping from an average index of 53.0 over the first week of the experiment to an average index of 47.0 over the last week. (See Exhibit 29)

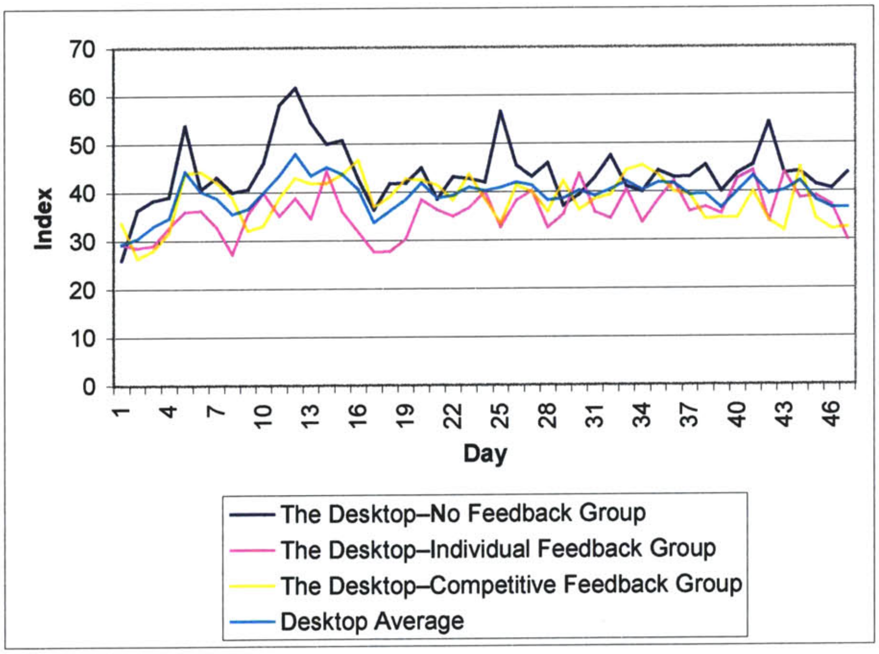


Exhibit 28: Desktop Users' Consumption of Fruits and Vegetables. (Responses were indexed on a scale of 0 to 100, where 0 was "none," 50 was "the recommended amount," and 100 was "twice as much as needed.") Consumption of fruits and vegetables rose slightly for all desktop users. The most significant increases were in the Desktop–No Feedback Group and the Desktop–Competitive Feedback Group.

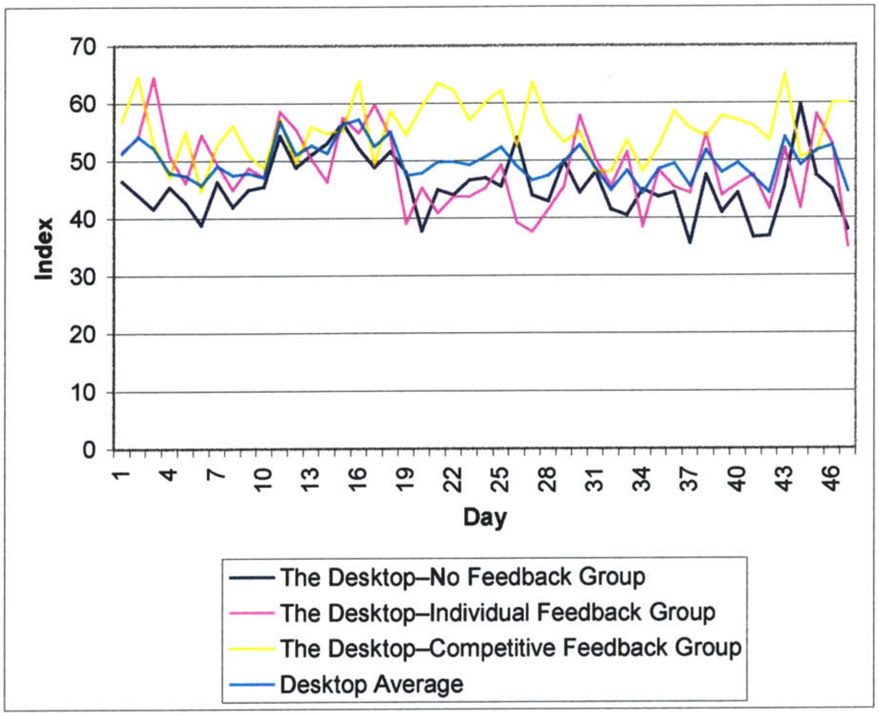


Exhibit 29: Desktop Users' Consumption of Fatty Foods. (Responses were indexed on a scale of 0 to 100, where 0 was "none," 50 was "the recommended amount," and 100 was "twice as much as needed.") The Desktop–Individual Feedback Group's consumption of fatty foods had a notable decreasing, dropping from an average index of 53.0 over the first week of the experiment to an average index of 47.0 over the last week.

Users in the Desktop–Individual Feedback Group and the Desktop–Competitive Feedback Group saw their caffeine consumption decrease over the course of the study, while the Desktop–No Feedback Group users' level of consumption remained relatively constant. (See Exhibit 30) This data would indicate that greater amounts of feedback led to gradual decreases in caffeine consumption. The Desktop–Individual Feedback Group's caffeine consumption dropped more than half, from 0.39 units to 0.15 units.

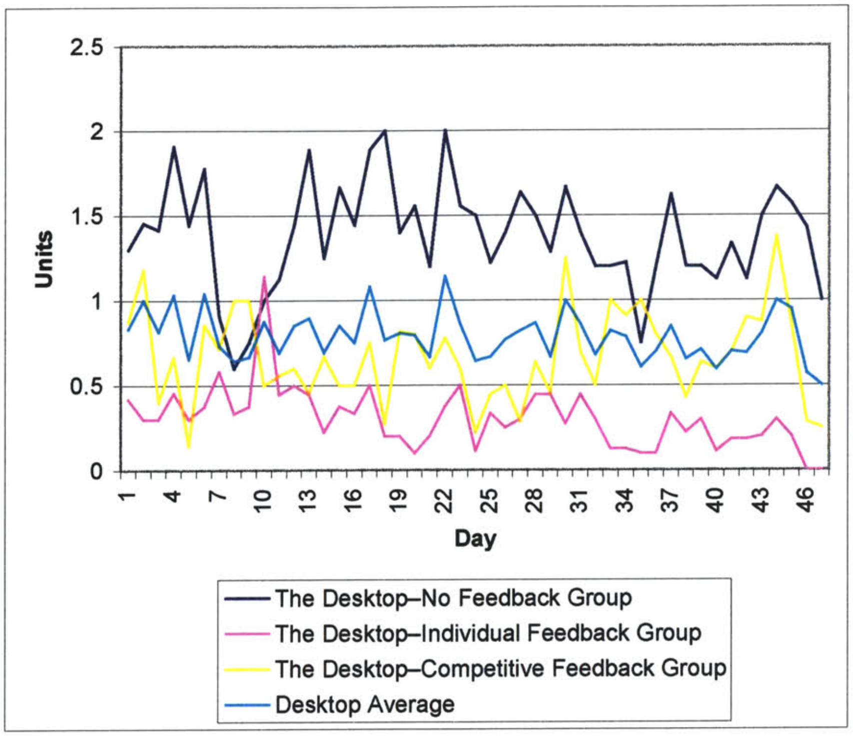


Exhibit 30: Desktop Users' Consumption of Caffeine. Users in the Desktop–Individual Feedback Group and 3 saw their caffeine consumption decrease over the course of the study, while the Desktop–No Feedback Group users' level of consumption remained relatively constant. This data would indicate that greater amounts of feedback led to gradual decreases in caffeine consumption.

All desktop users groups experienced dramatic changes in alcohol consumption during the study. (See Exhibit 30) The Desktop–No Feedback Group users decreased consumption by 22% from the first to last week while the Desktop–Individual Feedback Group users increased consumption of alcohol by 23%. The Desktop–Competitive Feedback Group, meanwhile, decreased consumption by 75%. This data indicates that the more feedback users received, the greater the changes in alcohol consumption.

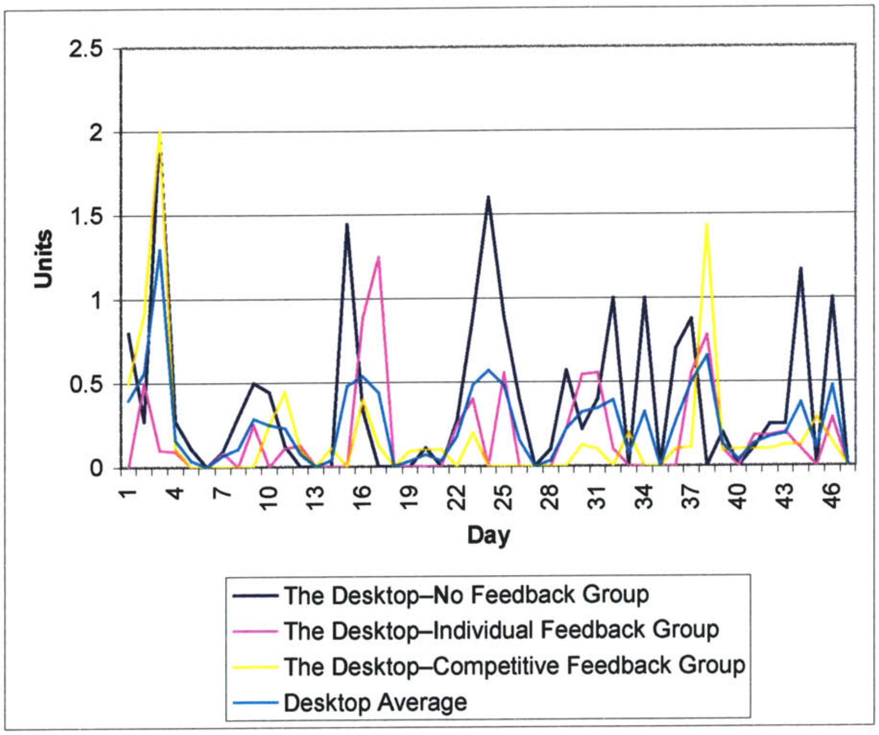


Exhibit 31: Desktop Users' Consumption of Alcohol. The Desktop–No Feedback Group users decreased consumption by 22% from the first to last week while the Desktop–Individual Feedback Group users increased consumption of alcohol by 23%. The Desktop–Competitive Feedback Group, meanwhile, decreased consumption by 75%. This data indicates that the more feedback users received, the greater the changes in alcohol consumption.

The Desktop–Individual Feedback Group and 3 users' exercise amount changed significantly over the course of the experiment, while the Desktop–No Feedback Group users maintained a relatively constant level of exercise time. (See Exhibit 32) In the last week of the study, the Desktop–Individual Feedback Group users exercised 27.6% less than in the first week. The Desktop–Competitive Feedback Group users, on the other hand, exercised over twice as much in the last week of the study compared to the first week.

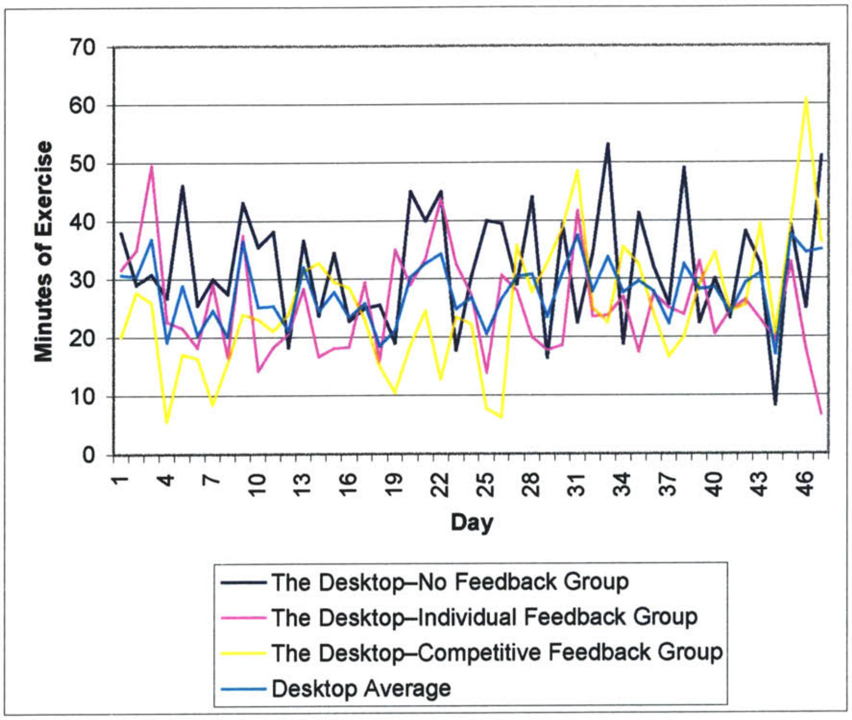


Exhibit 32: Amount of Exercise by Desktop Users. The Desktop–Individual Feedback Group and 3 users' exercise amount changed significantly over the course of the experiment, while the Desktop–No Feedback Group users maintained a relatively constant level of exercise time.

Stress levels increased slightly for both the Desktop–No Feedback Group and the Desktop–Individual Feedback Group, while the Desktop–Competitive Feedback Group users experienced a decline in stress. (See Exhibit 33) Another related metric were the results of a question asking users about their status on homework, determined by whether subjects were behind or ahead of their assigned work. The Desktop–No Feedback Group users improved their homework status by 20%, the Desktop–Individual Feedback Group worsened by 31% and the Desktop–Competitive Feedback Group improved by 21% from the first week to last. (See Exhibit 34) These results correlate slightly with levels of stress. It appears like the further ahead subjects were in completing their homework, the less stress they experienced.

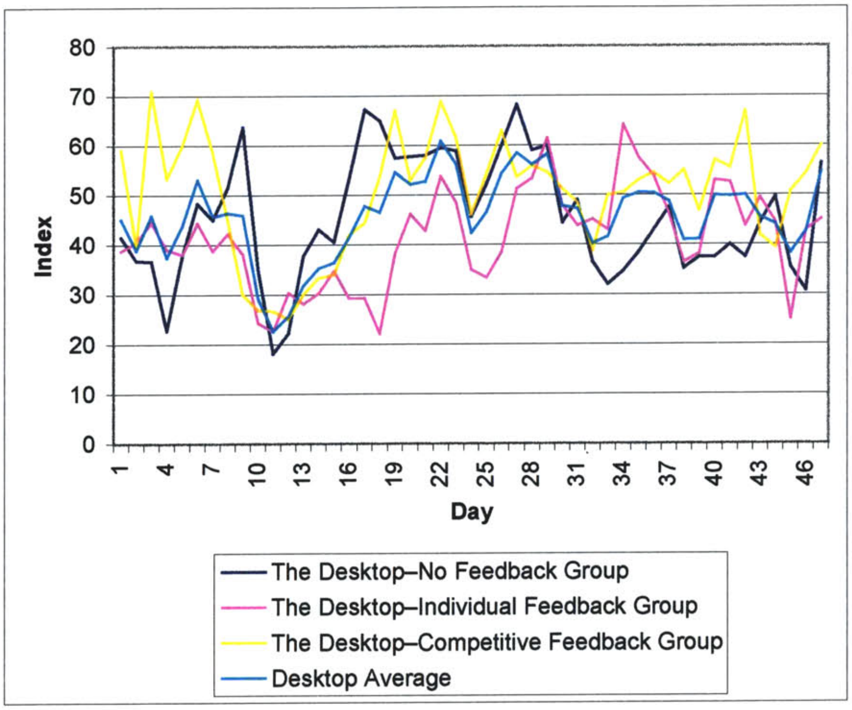


Exhibit 33: Desktop Users' Stress Levels. (Responses were indexed on a scale of 0 to 100, where 0 was "not at all" and 100 was "very much.") Stress levels increased slightly for both the Desktop–No Feedback Group and the Desktop–Individual Feedback Group, while the Desktop–Competitive Feedback Group users experienced a decline in stress.

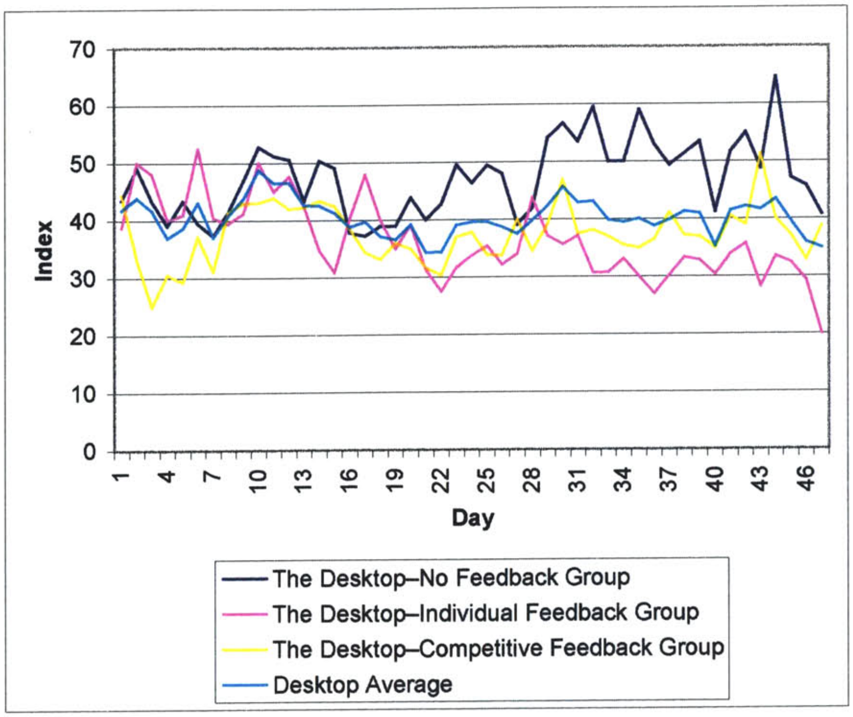


Exhibit 34: Desktop Users' Homework Status. (Responses were indexed on a scale of 0 to 100, where 0 was "very behind," 50 was "on schedule," and 100 was "ahead of schedule.") The Desktop–No Feedback Group users improved their homework status by 20%, the Desktop–Individual Feedback Group worsened by 31% and the Desktop–Competitive Feedback Group improved by 21% from the first week to last. These results correlate slightly with levels of stress. It appears like the further ahead subjects were in completing their homework, the less stress they experienced.

Users in the Desktop–Individual Feedback Group and the Desktop–Competitive Feedback Group displayed a great deal of change, comparing the first to last weeks, in other questions that the Desktop–No Feedback Group exhibited very little change in. These categories included level of happiness, adherence to one's general diet and attention to food intake. For example, Exhibit 35 illustrates the considerable changes in how much attention users paid to their daily food intake. The Desktop–No Feedback Group users decreased 7% in the index while the Desktop–Competitive Feedback Group users dropped 12%. The Desktop–Individual Feedback Group users led the trend by dropping 33%.

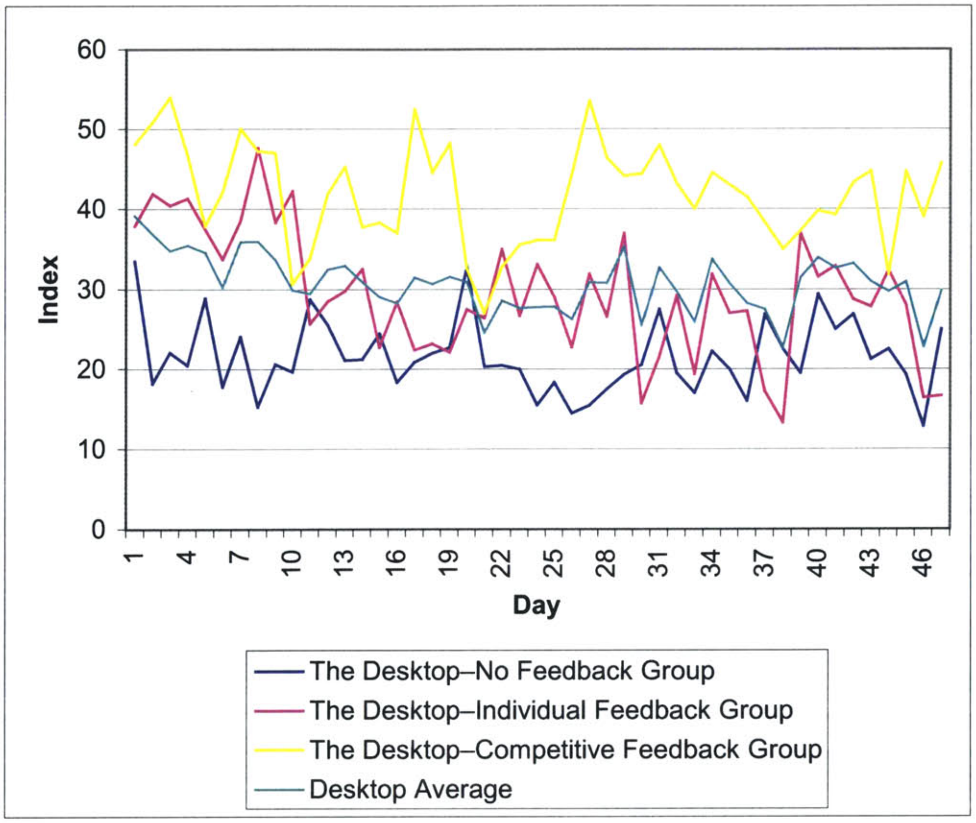


Exhibit 35: Desktop Users' Attention to Daily Food Intake. (Responses were indexed on a scale of 0 to 100, where 0 was "not at all" and 100 was "constantly.") There were considerable changes in how much attention desktop users paid to their daily food intake. The Desktop–No Feedback Group users decreased 7% in the index while the Desktop–Competitive Feedback Group users dropped 12%. The Desktop–Individual Feedback Group users led the trend by dropping 33%.

Desktop users' responses to other questions revealed little trends or patterns. Graphs displaying results to these questions can be found in Appendix T.

9 Wireless Study Results

9.1 Debrief Questionnaire

As was the case in the desktop experiment, there are general trends between the three user wireless groups as well. Exhibit 36 illustrates a rising trend in positive responses to HealthTracker's overall usefulness. 40% of the Wireless–Use Limitations Group users, 67% of the Wireless–No Feedback Group users and 70% of the Wireless–Individual Feedback Group users were enthusiastic about HealthTracker's helpfulness.

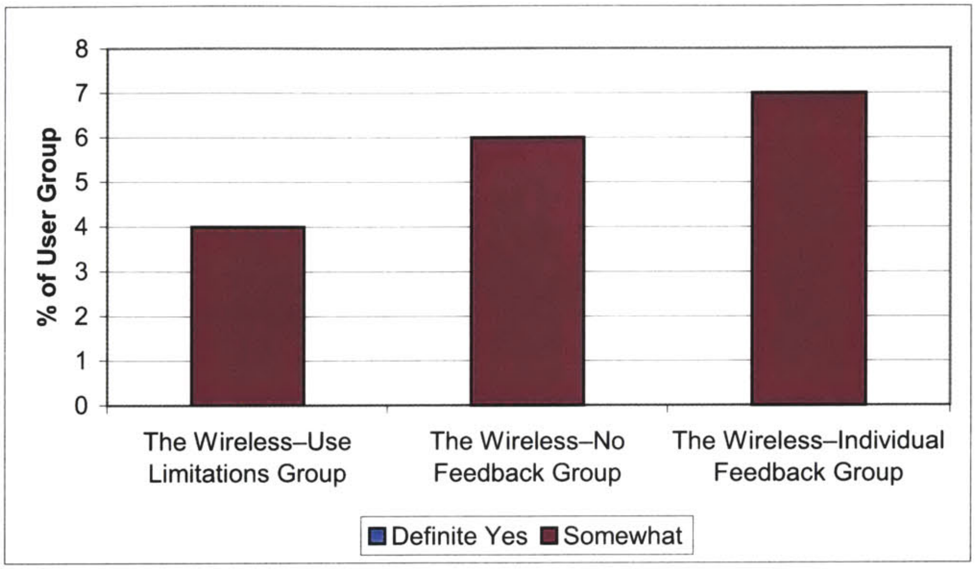


Exhibit 36: Wireless Users' Responses to the Question: "Did you find HealthTracker to be Useful?" This graph illustrates a rising trend in positive responses to HealthTracker's overall usefulness. 40% of the Wireless–Use Limitations Group users, 67% of the Wireless–No Feedback Group users and 70% of the Wireless–Individual Feedback Group users were enthusiastic about HealthTracker's helpfulness

The majority of wireless users echoed the sentiments of the desktop study participants in regards to the general usefulness of HealthTracker. Subjects asserted that the system made them more conscious of their health and well-being as Exhibit 37 and Exhibit 38 illustrate. Individuals were also pleased with how well HealthTracker encouraged them to monitor properties like diet and exercise. One respondent stated: "I realized that I should be more wary of foods I eat and how much more exercise I should be getting. Also, I wasn't aware of how fast my resting pulse was." Another individual commented on how HealthTracker made her realize that she was drinking too much alcohol and should cut back on consumption.

The graph in Exhibit 37 actually illustrates a relatively strong negative trend, indicating that the more subjects were allowed to use their PDA's for other applications and the more feedback they received from the system, the less they felt HealthTracker made them conscious of their diet and physical fitness.

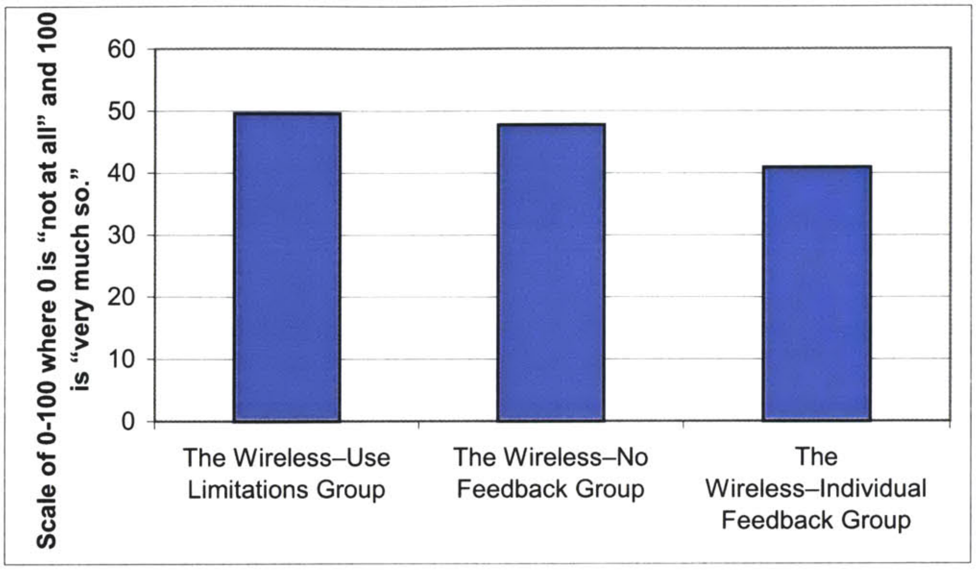


Exhibit 37: How Conscious HealthTracker Made Wireless Users About Their Diet. Wireless subjects generally reported that the system made them more conscious of their health and well-being. This graph illustrates a relatively strong negative trend, indicating that the more subjects were allowed to use their PDA's for other applications and the more feedback they received from the system, the less they felt HealthTracker made them conscious of their diet and physical fitness.

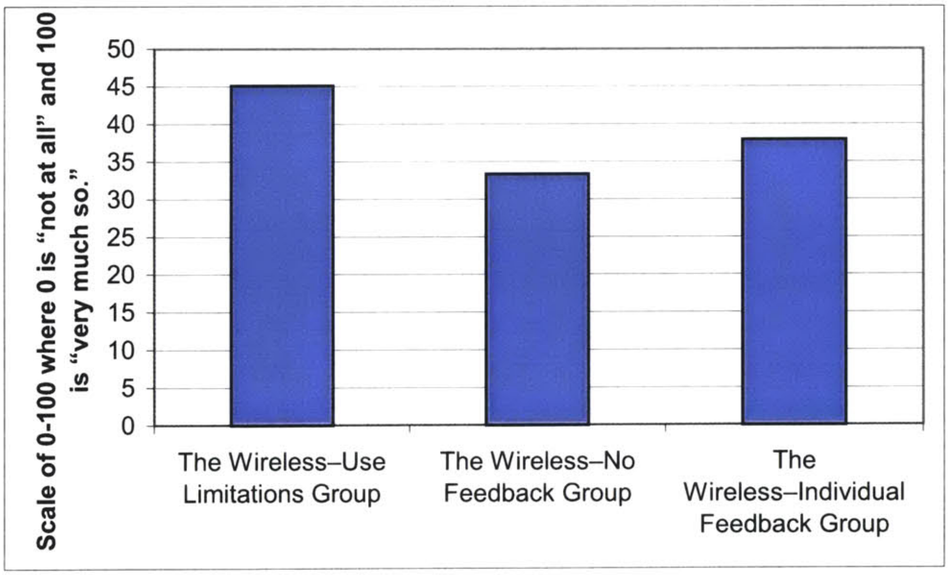


Exhibit 38: How Conscious HealthTracker Made Wireless Users About Their Physical Fitness. Wireless subjects generally asserted that the system made them more conscious of their health and well-being

Wireless users were also relatively indifferent about how conscious HealthTracker made them about their general happiness (See Exhibit 39) and mental health (See Exhibit 40). A moderately strong positive trend in Exhibit 39 indicates that increasing use of the wireless PDA and the addition of feedback both led to an increased consciousness of personal happiness. There is no such trend in Exhibit 40, though it appears that greater use of the Visor did make users more conscious about mental health.

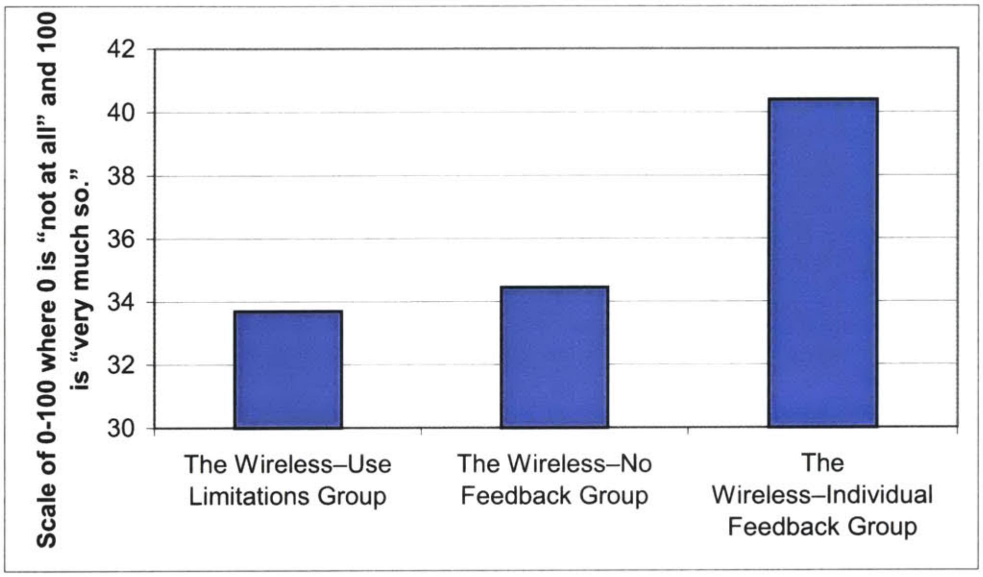


Exhibit 39: How Conscious HealthTracker Made Wireless Users About Their General Happiness. Wireless users were relatively indifferent about how conscious HealthTracker made them about their general happiness. A moderately strong positive trend in this graph indicates that increasing use of the wireless PDA and the addition of feedback both led to an increased consciousness of personal happiness.

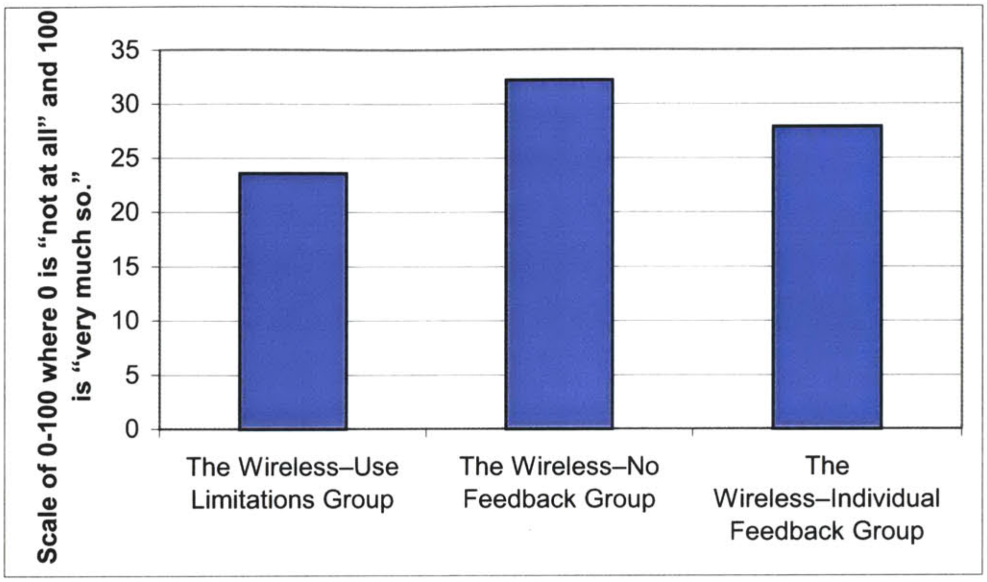


Exhibit 40: How Conscious HealthTracker Made Wireless Users About Their Mental Health. Wireless users were also relatively indifferent about how conscious HealthTracker made them. Though it appears that greater use of the Visor did make users more conscious about mental health.

Complaints about the system's efficacy were again very similar to that of the desktop study. A number of subjects remarked on how HealthTracker didn't reveal any new information, but only forced them to put down "on paper" what they already knew in their heads.

Half of the Wireless–Use Limitations Group and the Wireless–Individual Feedback Group found HealthTracker to be very user-friendly, while all of the Wireless–No Feedback Group liked the system's interface and design. (See Exhibit 41) The majority of complaints stemmed from technical difficulties. Many subjects had trouble maintaining a connection with the website, citing problems with the wireless modem. There were also the same problems with server failures and login problems that desktop users cited.

There were also again comments from users not receiving feedback about how access to their own historical data would have been extremely useful to them. A handful of participants also expressed difficulty using the 0 to 100 scaling for some questions regarding food intake and mental health. For these questions, individuals found it difficult to quantify their responses and would have preferred a different interface. One suggestion was a smaller scale. Another suggestion was to eliminate the numerical scale altogether and use verbal quantifiers like "I ate a lot of meat today → I ate some meat today → I ate just a little meat today → I ate no meat today."

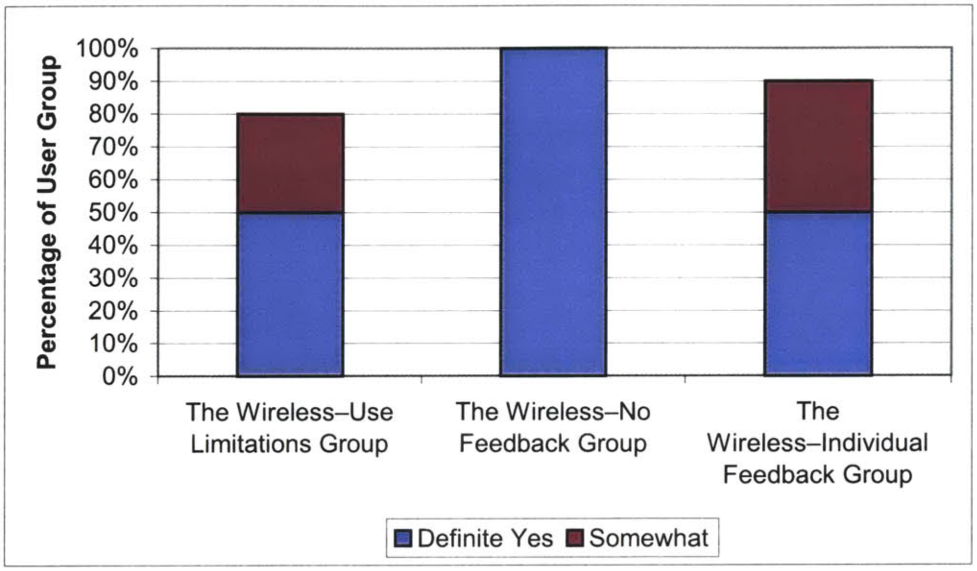


Exhibit 41: Wireless Users' Responses to the Question: "Did you find HealthTracker to be User-Friendly?" Half of the Wireless–Use Limitations Group and the Wireless–Individual Feedback Group found HealthTracker to be very user-friendly, while all of the Wireless–No Feedback Group liked the system's interface and design. The majority of complaints again stemmed from technical difficulties. Many subjects had trouble maintaining a connection with the website, citing problems with the wireless modem. There were also the same problems with server failures and login problems.

In general, most wireless users were not enthusiastic about continuing to use HealthTracker in its current implementation. (See Exhibit 42) Given some small changes to the system's design, however, the majority of the wireless subjects agreed that they would probably continue using the website. One of these changes would be to find a more reliable wireless service provider–one that would have fewer connection problems and power failures. Another change recommended by users in the Wireless–Individual Feedback Group was to make the feedback graphs clearer and more intuitive.

Additional data from the debrief questionnaire can be found in Appendix S.

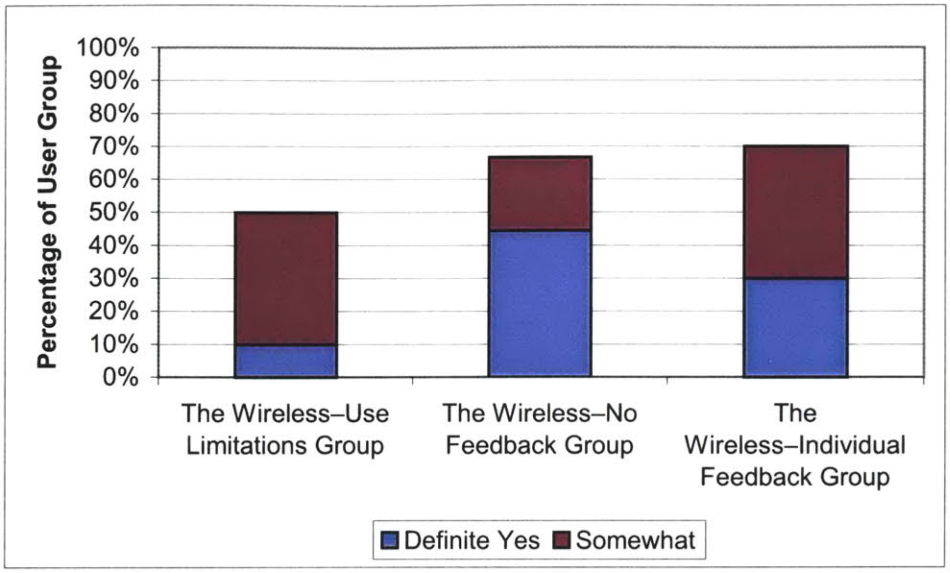


Exhibit 42: Wireless Users' Responses to the Question: "Would you continue using the HealthTracker system?" In general, most wireless users were not enthusiastic about continuing to use HealthTracker in its current implementation. Given some small changes to the system's design, however, the majority of the wireless subjects agreed that they would probably continue using the website.

9.2 Final Questionnaire

Due to technical and timing problems that will be explained in detail in Section 10.5.4, data from the final questionnaire ended up being incomplete and unusable.

9.3 Midterm Questionnaire

As described in Section 7.4, the central purpose of the midterm questionnaire was to evaluate how many questions, as well as which ones, users would respond to. Exhibit 43 illustrates the percentages of each tier of the midterm survey completed by wireless users. As the graph indicates, the three wireless user groups responded to nearly all questions in all four tiers. No group answered fewer than 94%, on average, of any tier of questions. There is no significant distinction in the reply percentages across the four tiers within each group. The standard deviations across the four tiers are 2%, 3% and 2% for the Wireless–Use Limitations Group, the Wireless–No Feedback Group and the Wireless–Individual Feedback Group respectively. Finally, there are also no distinguishable trends between the four groups. The standard deviation of the three groups' averages is only 1%.

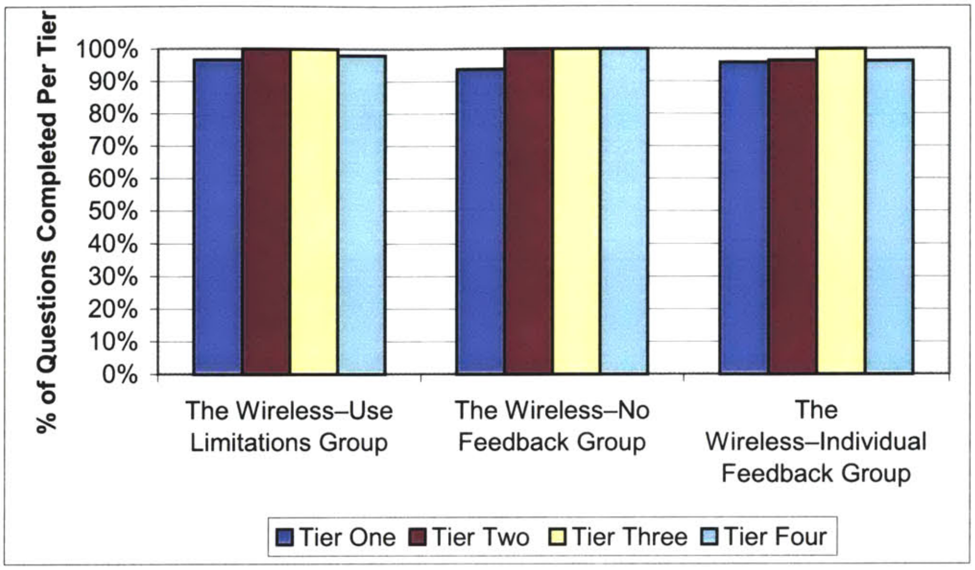


Exhibit 43: Percentage of Midterm Questionnaire Completed by Wireless Users. The three wireless user groups responded to nearly all questions in all four tiers. No group answered fewer than 94%, on average, of any tier of questions.

9.4 Daily Questionnaire

9.4.1 Compliance

Compliance, as defined as the percentage of completed daily questionnaires, was effectively constant between the three wireless groups. (See Exhibit 44) The following group averages reflect the omission of two subjects from the Wireless–No Feedback Group and one subject from the Wireless–Individual Feedback Group who withdrew from the study in its first week. All three groups had about 80% compliance. In fact, the standard deviation between the levels of compliance for the three groups was only 2%.

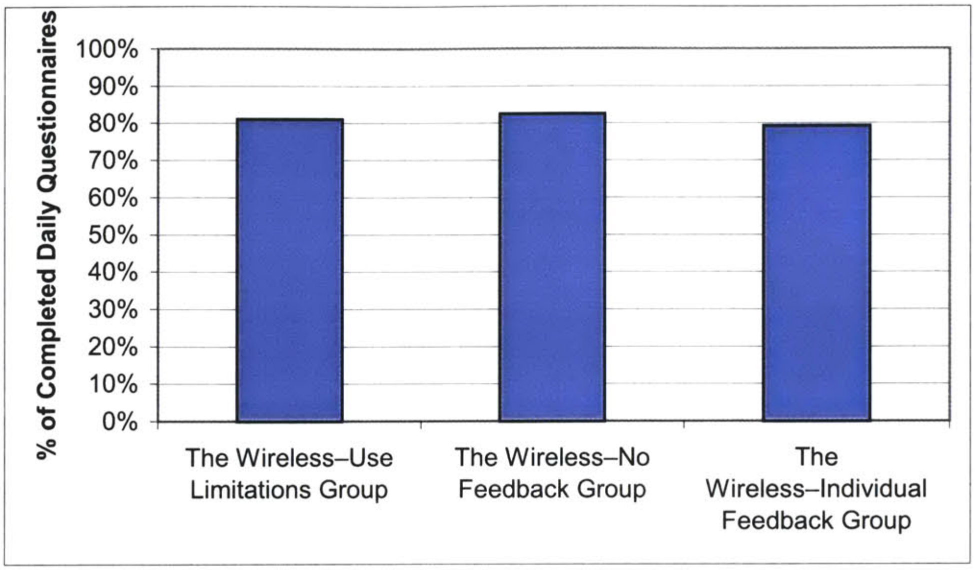


Exhibit 44: Wireless Users Compliance in the Daily Questionnaire. Compliance, as defined as the percentage of completed daily questionnaires, was effectively constant between the three wireless groups.

While compliance was relatively high however, the frequency with which wireless users logged into HealthTracker was rather low. (See Exhibit 45) All three wireless groups, on average, logged into HealthTracker for a little over half of the days of the experiment, or approximately every other day. The Wireless–No Feedback Group users logged in most frequently, on 62% of the total days in the study. The Wireless–Use Limitations Group and the Wireless–Individual Feedback Group users logged in 51% and 57% of the time respectively.

HealthTracker was designed so that users could "make up" questionnaire submissions they had missed from previous days. All users were cautioned to do this only if the questionnaires they were filling out were for relatively recent days. Nonetheless, it appears from this data that, on average, the majority of wireless users filled out two questionnaires every time they logged into HealthTracker.

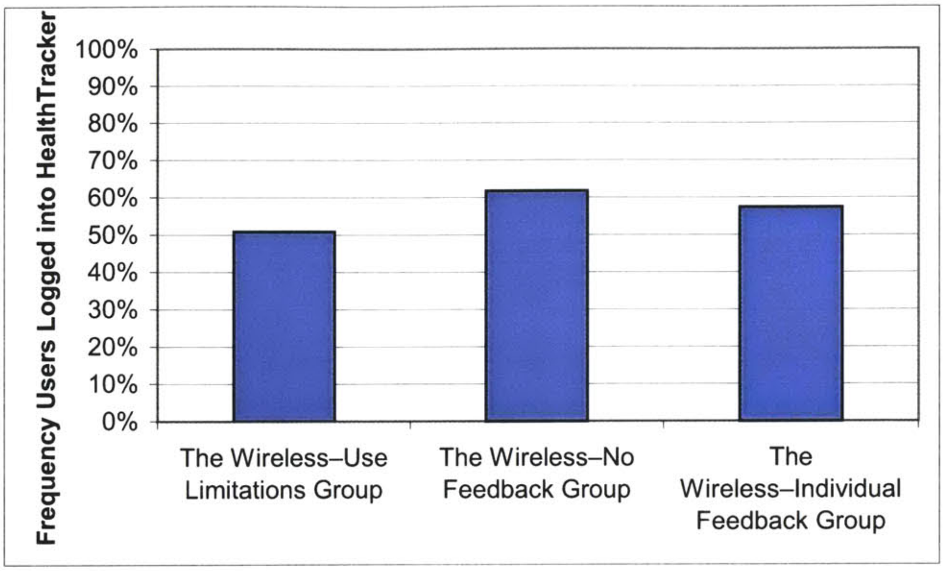


Exhibit 45: How Often Wireless Users Logged Into HealthTracker. The frequency with which wireless users logged into HealthTracker was rather low. All three wireless groups, on average, logged into HealthTracker for a little over half of the days of the experiment, or approximately every other day.

The third compliance metric introduced in Section 8.4.1 was the number of times users changed their responses from day-to-day. Recall that the design of HealthTracker allowed for the approximate counting of skipped questions by identifying two identical responses on consecutive response days. This metric could only be approximate however, since motivations for skipping a question can be varied. Some users might have skipped questions in the interest of time, while others simply had the same answer two days in a row.

Exhibit 46 illustrates the frequency that wireless users changed their responses from day-to-day. The Wireless–Individual Feedback Group users made the most day-to-day changes to their questionnaires, modifying 51% of their responses day-to-day on average. The Wireless–Use Limitations Group and the Wireless–No Feedback Group users modified their responses 45% and 48% of the time on average. As the graph indicates, there is a relatively strong increasing trend as we progress from the Wireless–Use Limitations Group to the Wireless–No Feedback Group to the Wireless–Individual Feedback Group. It appears that the more wireless users interact with the PDA's and the more feedback they receive, the more questions they will answer on the daily survey.

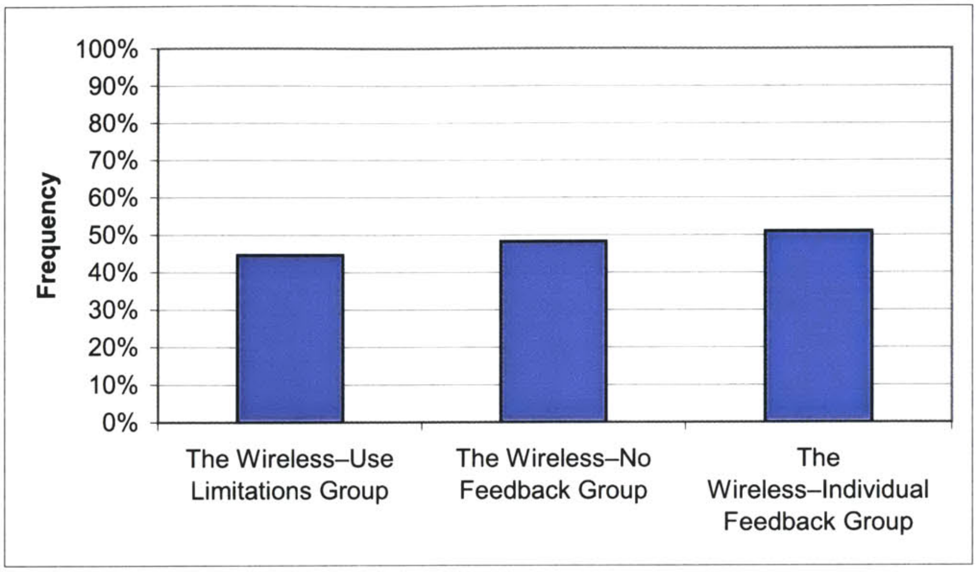


Exhibit 46: Frequency that Wireless Users Changed Responses From Day-to-Day. As this graph indicates, there is a relatively strong increasing trend as we progress from the Wireless–Use Limitations Group to the Wireless–No Feedback Group to the Wireless–Individual Feedback Group. It appears that the more wireless users interact with the PDA's and the more feedback they receive, the more questions they will answer on the daily survey.

9.4.2 Effects of the Daily Questionnaire

The results of the HealthTracker daily questionnaire were mixed for the three wireless groups. The Wireless–No Feedback Group's consumption of meat rose 13% from the first to last week and the same group's intake of fruits and vegetables rose 28%. (See Exhibit 47 and Exhibit 48) Users in the Wireless–Use Limitations Group however, only saw their consumption of the same foods rise 7%. Finally, users in the Wireless–Individual Feedback Group actually saw their intake of meat drop 5% and consumption of fruits and vegetables decrease by 6%.

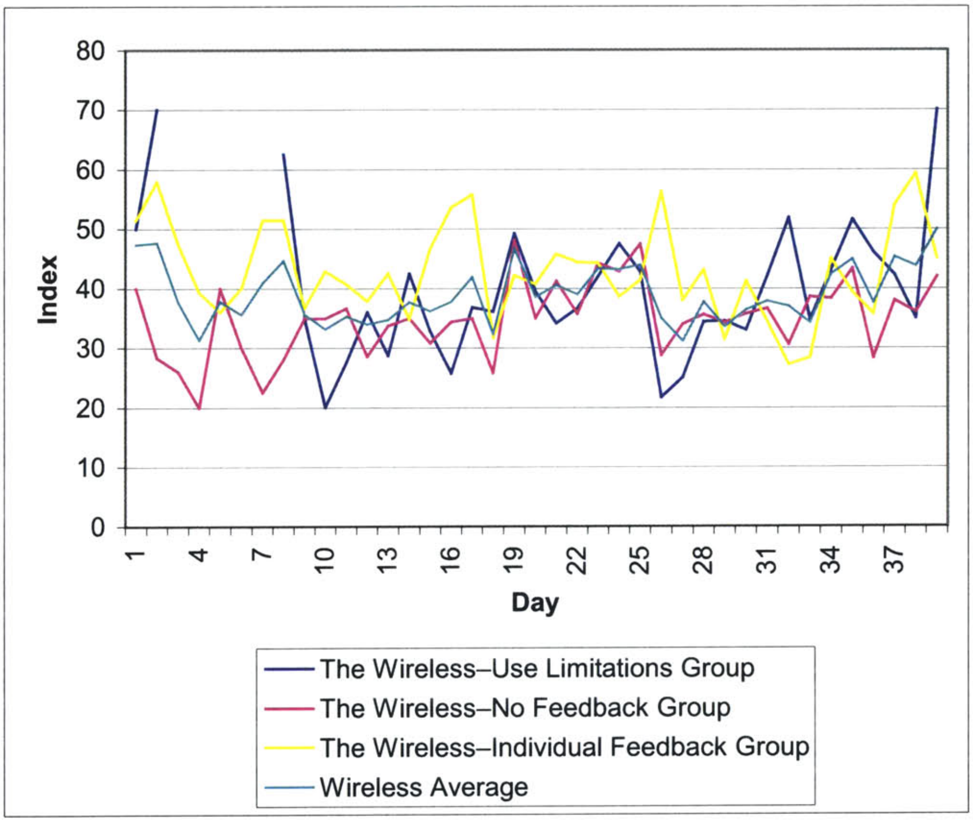


Exhibit 47: Wireless Users' Consumption of Meat. (Responses were indexed on a scale of 0 to 100, where 0 was "none," 50 was "the recommended amount," and 100 was "twice as much as needed.") The Wireless–No Feedback Group's consumption of meat rose 13% from the first to last week. Users in the Wireless–Use Limitations Group however, only saw their consumption of meat rise 7%. Users in the Wireless–Individual Feedback Group actually saw their intake of meat drop 5%.

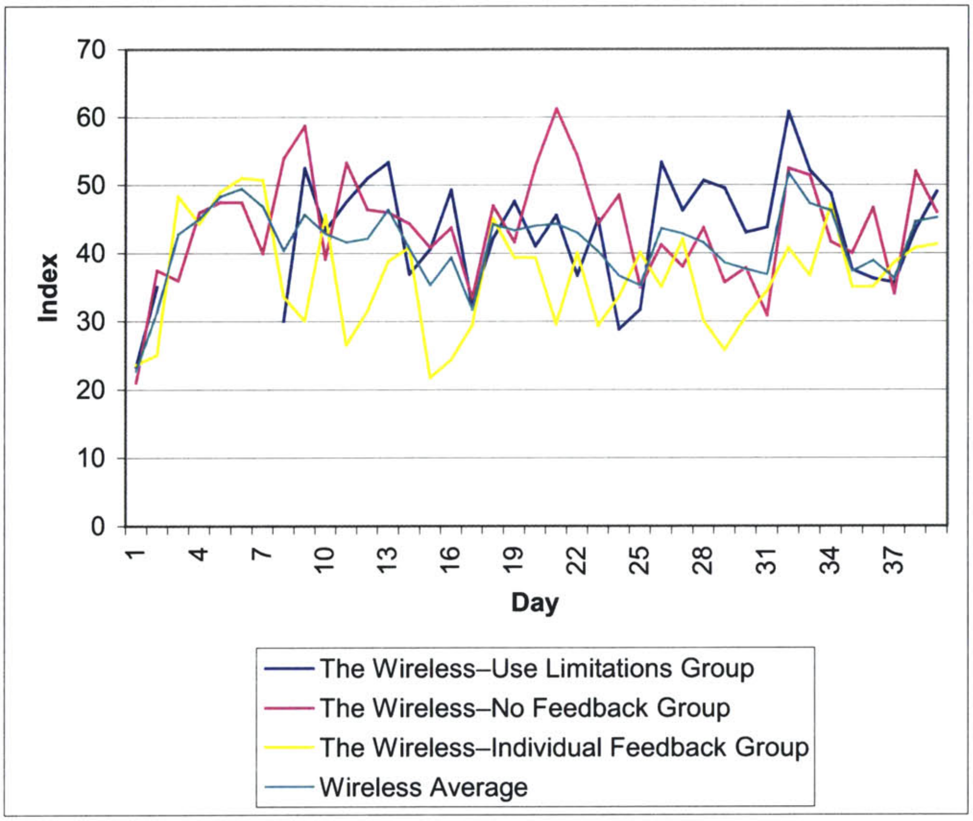


Exhibit 48: Desktop Users' Consumption of Fruits and Vegetables. (Responses were indexed on a scale of 0 to 100, where 0 was "none," 50 was "the recommended amount," and 100 was "twice as much as needed.") The Wireless–No Feedback Group's consumption of fruits and vegetables rose 28%. Users in the Wireless–Use Limitations Group however, only saw their consumption of the same foods rise 7%. Finally, users in the Wireless–Individual Feedback Group actually saw their consumption of fruits and vegetables decrease by 6%.

Users in the Wireless–Use Limitations Group and the Wireless–Individual Feedback Group exhibited dramatic increases in alcohol consumption from the first week of the study to the last. (See Exhibit 49) Users in the Wireless–Use Limitations Group increased their alcohol consumption, on average, 125% from the first to last week. The Wireless–Individual Feedback Group users' consumption increased seven-fold. Finally, the Wireless–No Feedback Group users reported a 50% decrease in consumption.

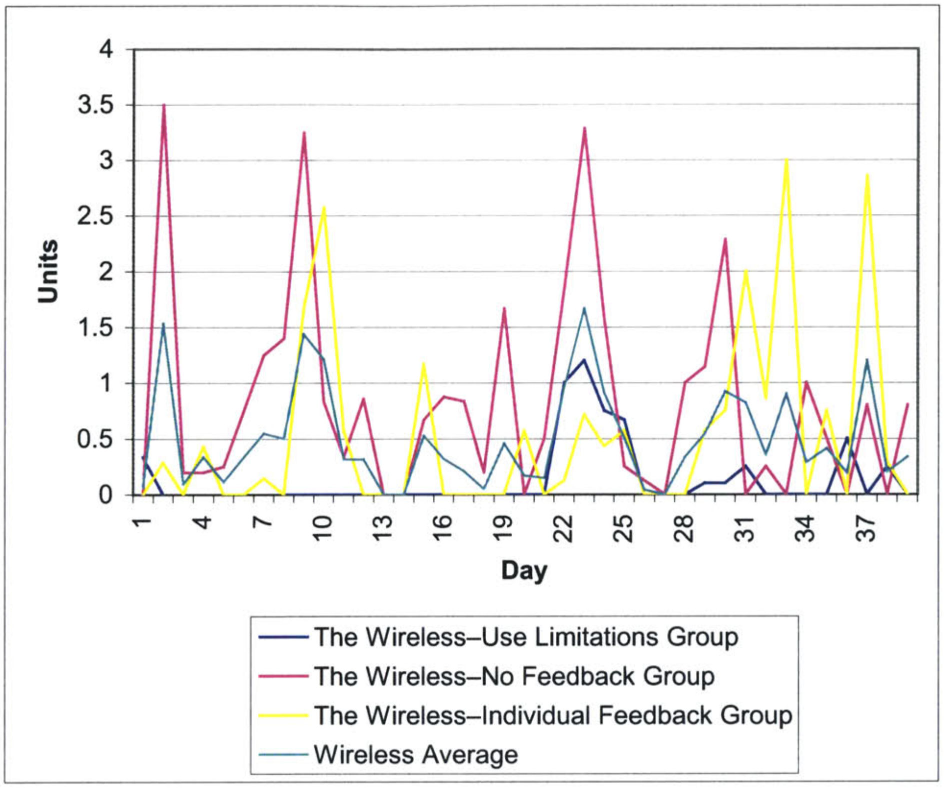


Exhibit 49: Wireless Users' Consumption of Alcohol. Users in the Wireless–Use Limitations Group and the Wireless–Individual Feedback Group exhibited dramatic increases in alcohol consumption from the first week of the study to the last. Users in the Wireless–Use Limitations Group increased their alcohol consumption, on average, 125% from the first to last week. The Wireless–Individual Feedback Group users' consumption increased seven-fold. Finally, the Wireless–No Feedback Group users reported a 50% decrease in consumption.

The Wireless–Use Limitations Group and 6 users' exercise amount both increased over the course of the experiment, while the Wireless–No Feedback Group users maintained a relatively constant level of exercise time. (See Exhibit 32) The Wireless–Use Limitations Group users increased their amount of exercise by 98% while the Wireless–Individual Feedback Group demonstrated a more modest increase of 14%. The Wireless–No Feedback Group subjects' exercise time decreased by just 2%.

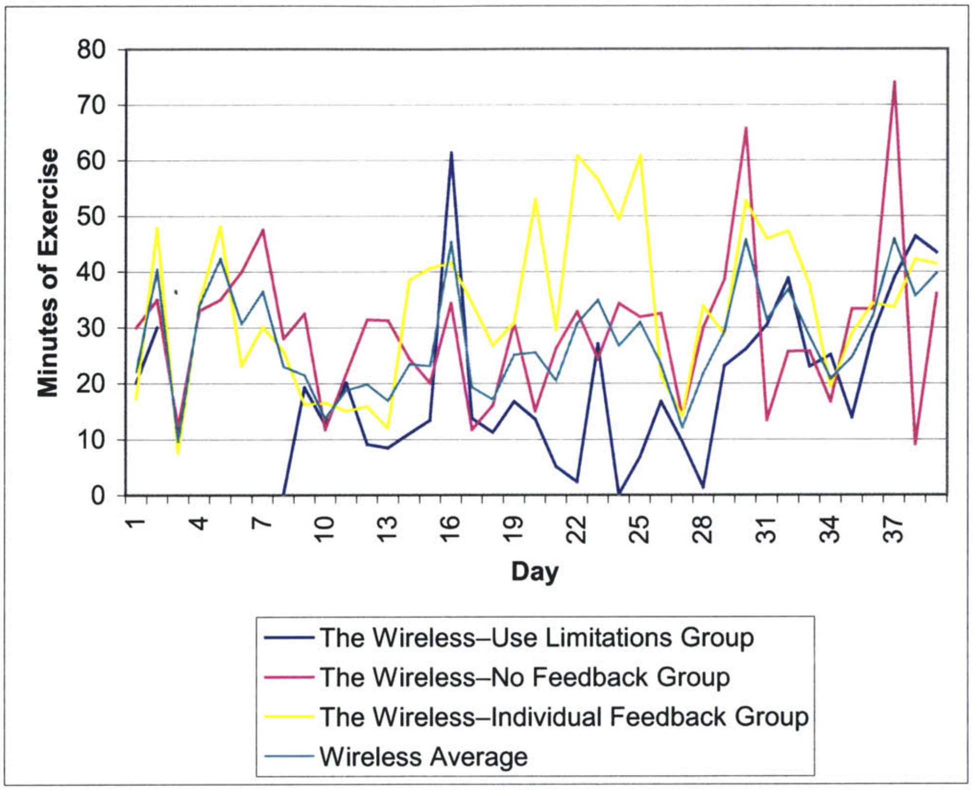


Exhibit 50: Amount of Exercise by Desktop Users. The Wireless–Use Limitations Group and 6 users' exercise amount both increased over the course of the experiment, while the Wireless–No Feedback Group users maintained a relatively constant level of exercise time.

Stress levels amongst the wireless users changed significantly over the course of the experiment. (See Exhibit 33) Subjects in the Wireless–Use Limitations Group reported a 21% decrease in stress levels, while subjects in the Wireless–No Feedback Group and 6 both exhibited relatively sharp increases in stress levels with 46% and 16% increases respectively. It would appear from these results that increased use of the wireless PDA is cause for greater changes in stress levels, but the addition of feedback information dampens the magnitude of these changes.

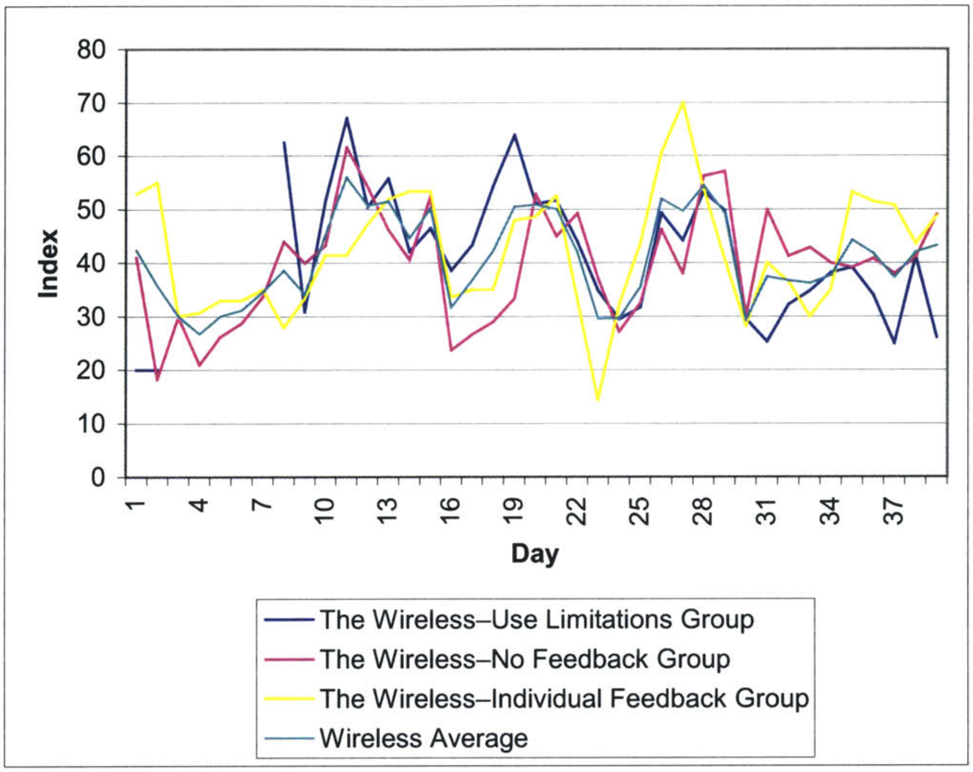


Exhibit 51: Desktop Users' Stress Levels. (Responses were indexed on a scale of 0 to 100, where 0 was "not at all" and 100 was "very much.") Stress levels amongst the wireless users changed significantly over the course of the experiment. Subjects in the Wireless–Use Limitations Group reported a 21% decrease in stress levels, while subjects in the Wireless–No Feedback Group and 6 both exhibited relatively sharp increases in stress levels with 46% and 16% increases respectively. It would appear from these results that increased use of the wireless PDA is cause for greater changes in stress levels, but the addition of feedback information dampens the magnitude of these changes.

There is also a relatively strong correlation between stress levels and homework status, determined by whether subjects were behind or ahead of their assigned work. Subjects in both The Wireless–No Feedback Group and the Wireless–Individual Feedback Group, who experienced increases in stress levels, also found themselves falling behind in their assigned work. The Wireless–No Feedback Group users fell behind in their homework status by 7% and subjects in the Wireless–Individual Feedback Group dropped back by 12%. (See Exhibit 34) On the other hand, the Wireless–Use Limitations Group users, who had displayed a decrease in stress levels, reported a 73% improvement in their homework status from the first to last week.

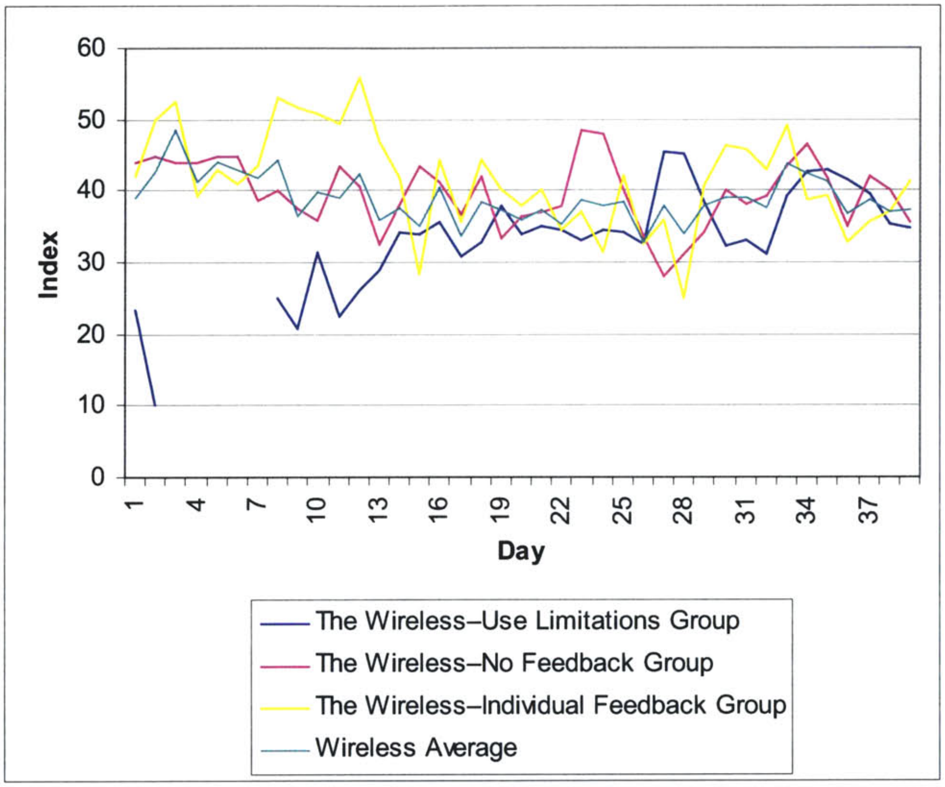


Exhibit 52: Wireless Users' Homework Status. (Responses were indexed on a scale of 0 to 100, where 0 was "very behind," 50 was "on schedule," and 100 was "ahead of schedule.") There is a relatively strong correlation between stress levels and homework status, determined by whether subjects were behind or ahead of their assigned work. Subjects in both The Wireless–No Feedback Group and the Wireless–Individual Feedback Group, who experienced increases in stress levels, also found themselves falling behind in their assigned work.

It also turns out that there is a very strong correlation between stress levels and the intensity of users' academic worries. All three wireless groups revealed an increase in their academic worries, but the Wireless–Use Limitations Group reported the smallest increase with 19%. (See Exhibit 53) The Wireless–No Feedback Group and the Wireless–Individual Feedback Group, who had both reported the largest surges in stress levels, also came in with the largest increases in academic worries: 99% and 41% respectively.

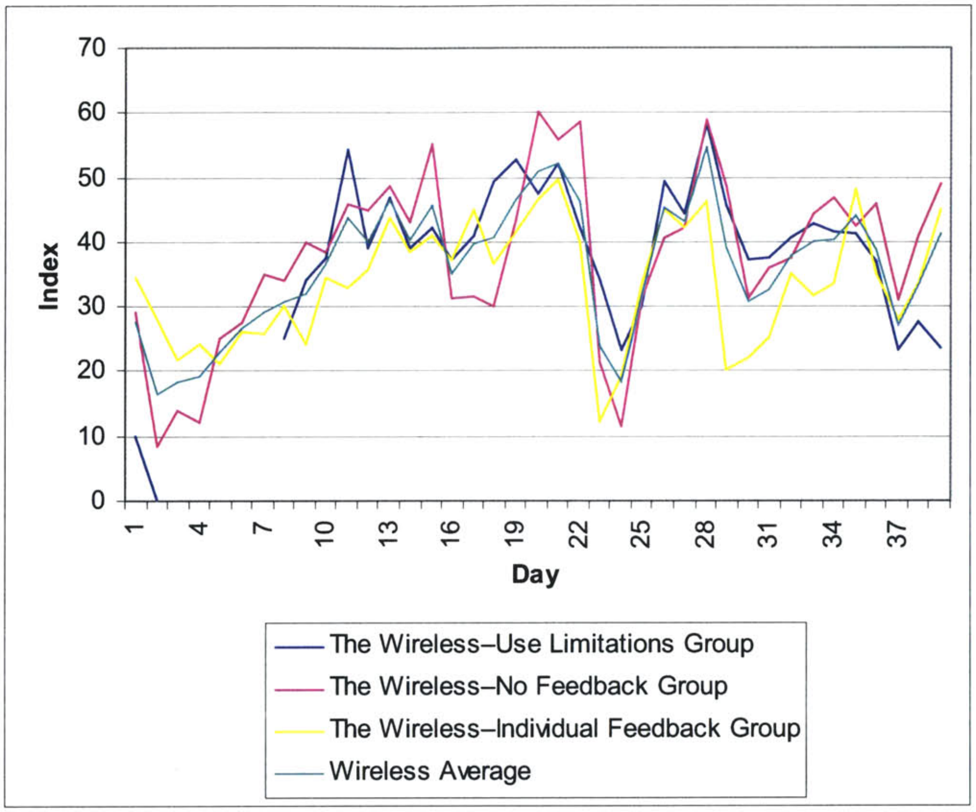


Exhibit 53: Wireless Users' Level of Academic Worries. (Responses were indexed on a scale of 0 to 100, where 0 was "always" and 100 was "never.") There is a very strong correlation between stress levels and the intensity of their academic worries. All three wireless groups revealed an increase in their academic worries, but the Wireless–Use Limitations Group reported the smallest increase with 19%. The Wireless–No Feedback Group and the Wireless–Individual Feedback Group, who had both reported the largest surges in stress levels, also came in with the largest increases in academic worries: 99% and 41% respectively.

A trend was also evident in how closely users in the three wireless groups monitored their food intake. (See Exhibit 54) While subjects in the Wireless–Use Limitations Group gave the same amount of attention to their food intake during the first and last weeks of the experiment, the Wireless–No Feedback Group users increased their attention by 18%. The Wireless–Individual Feedback Group subjects boosted their attention to food intake by 91%. These trends would again indicate that increased use of the wireless PDA led to greater changes in subjects' health and wellness. Furthermore, the addition of individual feedback also seemed to dramatically increase users' attention to daily food intake.

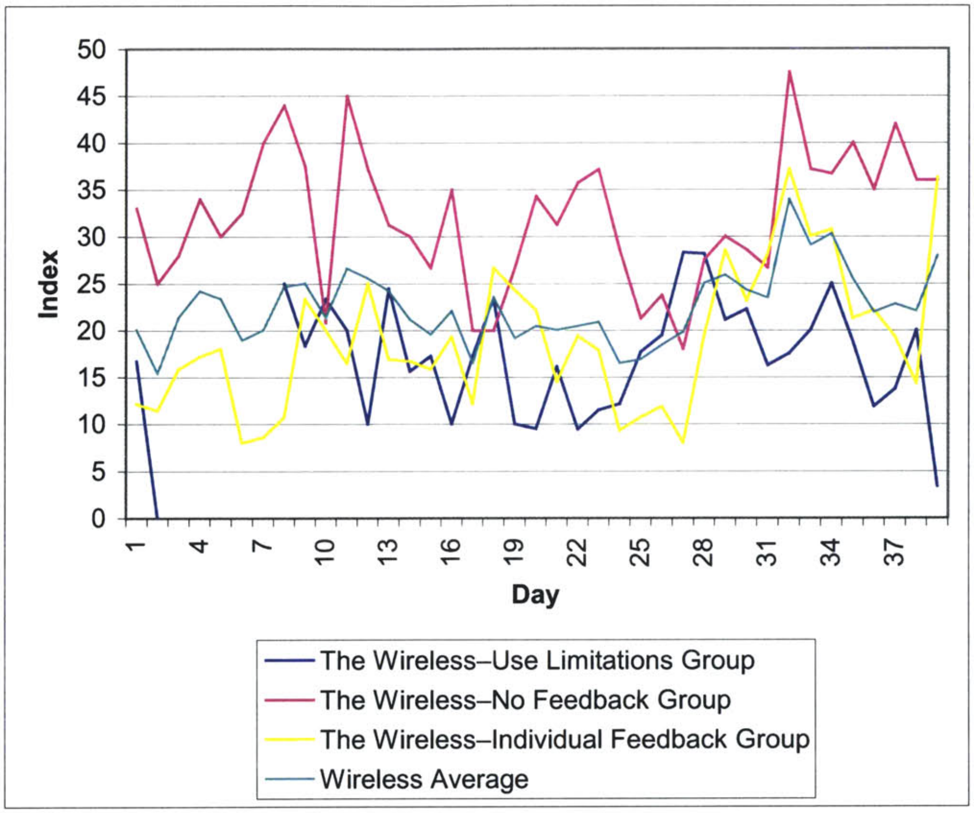


Exhibit 54: Wireless Users' Attention to Daily Food Intake. (Responses were indexed on a scale of 0 to 100, where 0 was "not at all" and 100 was "constantly.") While subjects in the Wireless–Use Limitations Group gave the same amount of attention to their food intake during the first and last weeks of the experiment, the Wireless–No Feedback Group users increased their attention by 18%. The Wireless–Individual Feedback Group subjects boosted their attention to food intake by 91%. These trends would again indicate that increased use of the wireless PDA led to greater changes in subjects' health and wellness. Furthermore, the addition of individual feedback also seemed to dramatically increase users' attention to daily food intake.

Wireless subjects' responses to other questions in the daily survey revealed largely insignificant trends and patterns. Graphs displaying the responses for these questions can be found in Appendix T.

10 Discussion

10.1 Previous Day Feedback

It turned out that users not receiving graphical feedback were still partly affected by the previous day responses displayed as response defaults. Almost all subjects appreciated this particular feature, since it sped up the answering of the daily survey considerably. Many users without formal feedback however, also used these previous day defaults as a form of short-term feedback. These individuals used their answers from the day before as a type of standard and often answered the questionnaire relative to this standard.

This occurrence is a definite indication of the power of feedback information. Even though the previous day defaults only displayed one day's worth of feedback, subjects still valued what little information it offered and used it to make decisions. The lack of additional historical data actually frustrated some feedback-less individuals.

10.2 Individual Feedback

Both desktop and wireless users found HealthTracker to be more useful with individual feedback than without. While 70% of subjects not receiving feedback found HealthTracker to be useful, 78% of desktop and wireless users receiving individual feedback reported HealthTracker to be useful. This data strongly supports the positive benefits of feedback and its power to enhance a medical monitoring system like HealthTracker.

In the debrief questionnaire, subjects offered a number of reasons for why they enjoyed the feedback graphs and why the feedback added to their experience. The most prominent explanation was that individuals liked looking for patterns in their behavior. These trends often made participants realize habits that they had been completely unaware of, such as over-consumption of alcohol or a habitual lack of sleep. Some users were able to extrapolate even more intricate conclusions. For example, one user commented on how he was able to see a distinct relationship between his stress and happiness levels as well as his homework status. The further ahead he was in his schedule, the happier and more stress-free he was. While drawing these types of relationships don't immediately translate into specific health benefits, they do educate users about their own health and well-being.

It was this self-education that seemed to please most users receiving individual feedback. As the background research in Section 3 indicates, there has been an increasing trend in the health industry towards personal health management. In general, more and more people have decided to take charge of their own health and wellness using the increasing availability of online services and alternative medicine. HealthTracker, which could certainly be characterized as one example of these new online services, seems to have elicited a very similar response from its pool of subjects.

In fact, many subjects commented how they enjoyed seeing trends in their data just for the sake of better understanding their own self. Said one participant: "The graphs were the best ways to see patterns and that was the most instructive way for me to realize the kinds of behaviors I may not like to admit to myself." It seems like many people simply need the feeling of being in control of and knowledgeable about one's own condition. Many HealthTracker users mentioned how the system didn't actually change any of their daily habits and their daily questionnaire responses reflected this consistency over time. Yet these same individuals responded very positively to questions about the usefulness of the system and enthusiastically agreed that they would continue using HealthTracker were it available.

The individual feedback graphs were not immediately accepted by most users however. Instead, most subjects explained how it took a few days for them to become accustomed to the feedback information. Although most people also added that there simply wasn't much to see in the first week of the study. Having only three or four points of data on a graph did not provide too much useful information. Some users however, did take some time to simply become familiar with the graph and comfortable with its information. After realizing the potential information that could be gained from the feedback though, these individuals quickly warmed up to the graphs and paid more attention to them.

Based on data collected in the debrief questionnaires, subjects receiving just individual feedback paid 9% more attention to the feedback graphs at the end of the study than at the beginning. And according to many comments, this percentage would have been even higher were it not for the increasing academic pressures and decreasing amount of available time towards the end of the semester. The increasing workload and rising pressures of final exams, papers and projects were definitely reflected in the data. Stress levels, level of academic concerns and general happiness all took a toll. Even more indirect measures of academic pressures like amount of exercise, sleep and eating habits also showed changes as the semester progressed.

In general, there was less positive response about the individual feedback on the wireless PDA's. There appeared to be occasional technical difficulties with the feedback graphs. Sometimes the graphs did not appear to be accurate while other times the graphs would not load at all. These technical issues will be discussed in greater detail in Section 10.5. Another issue with the wireless feedback was that the graphs did not come with labeled axes. Due to limitations imposed by the OmniSky wireless service, including labeled axes would have been too costly and was thus excluded from the design. Section 10.5 will again discuss this topic in more detail. As a result of these problems however, some users in the Wireless–Individual Feedback Group ended up ignoring the feedback graphs. Other members continued to study the graphs and when they were accurate, used them to establish trends. Said one the Wireless–Individual Feedback Group user: "When [the feedback graphs] was accurate, I followed them a lot. Many times it was either not quite up to date or a little odd, but they did still make me think about what they represented. The graphs still served their functions."

A noticeable effect of the individual feedback was that they caused subjects' daily questionnaire responses to fluctuate more from the first week to the last. To determine this, the average response to each question was calculated for the first week and then the last. Then, for every user group, the absolute change between these two values was calculated. So, the absolute change for a 9% increase or decrease in meat consumption would just be 9%. After these calculations were performed, it turned out that the Desktop–No Feedback Group users' daily questionnaire responses changed 11% from the beginning of the study to the end. The Wireless–Use Limitations Group and the Wireless–No Feedback Group users' responses changed 27% and 16% respectively. So, on average, users receiving no feedback changed their responses by 18% over the course of the study. Users in the Desktop–Individual Feedback Group and the Wireless–Individual Feedback Group who were receiving individual feedback, on the other hand, changed their responses by 36.5% on average.

A possible explanation for this significant difference in responses is that the graphical feedback gave subjects more perspective on their health trends. With the ability to see their historical performance over time and compare answers, users were better equipped to answer the questionnaire more accurately and respond to patterns and trends. Users not receiving feedback, on the other hand would have no point of reference for answering the daily questions except for their answers from the previous day. As a result, without any standard of comparison or listing of past answers, individuals without feedback end up not having any perspective or anchoring and are forced to answer questions somewhat more arbitrarily.

For example, a user might represent an increase in fatty foods consumption on one day with a 30 point increase in their response to the fatty foods question. Two weeks later, that same increase in fatty foods consumption might be represented as a 60 point increase. Without feedback to provide framing, users responses become somewhat more arbitrary.

So why would users without feedback exhibit a smaller amount of absolute change in their questionnaire responses? A potential reason is that without a historical graph to provide guidance, users feel safer entering in responses that are relatively close to their previous day's response. As a result, these individuals will have much less variance in their answers.

10.3 Group Feedback

Only users in the Desktop–Competitive Feedback Group received feedback illustrating their performance versus the study average. The purpose of this experiment was to see if additional feedback would further enhance the HealthTracker system and increase compliance.

As it turned out, out of all the desktop groups, subjects in the Desktop–Competitive Feedback Group answered the fewest number of daily questionnaires on average and logged in to the system the fewest number of times. The Desktop–Competitive Feedback Group also had the greatest percentage of skipped questions out of all the desktop groups. Even all the wireless users, except for those in the Wireless–Use Limitations Group, changed answers more frequently than members of the Desktop–Competitive Feedback Group despite the greater complexity of the PDA's user interface.

Despite this seemingly comparatively lackluster interest in the HealthTracker system however, users receiving both individual and group feedback gave the highest approval rating of HealthTracker's usefulness. 91% of the Desktop–Competitive Feedback Group found HealthTracker to be useful. Compared to the 70% and 70% approval ratings of the no-feedback and individual feedback groups, a 91% approval is extremely high.

Comments received in the debrief questionnaire help to explain this apparent discrepancy. The Desktop–Competitive Feedback Group users universally agreed that the group feedback had a negligible effect on their health. In fact, a handful of users ended up simply ignoring the group feedback. These users explained how they were simply not interested in the performance of complete strangers. Most people probably had different health and wellness standards and were coming in with completely different medical conditions. So, there was no point in comparing oneself with an average that was being generated by people with different standards and dissimilar goals. Some of individuals in the Desktop–Competitive Feedback Group also mentioned how they were against the competitive nature of the group feedback. Based on these responses, there would appear to be little reason why the addition of group feedback would in any be reflected in additional changes to users' questionnaire responses or level of compliance.

Why then was the Desktop–Competitive Feedback Group's approval of HealthTracker's usefulness so high then? It turns out that while most the Desktop–Competitive Feedback Group subjects didn't change their habits because of the group feedback, some still looked at it and occasionally searched for patterns and relationships between their performance and the group average. So it would seem that the group feedback still had a novelty value to it. More importantly, the group average was simply more information made available to the users, who were free to accept it or ignore it. As a result, the group average didn't seem to lessen the HealthTracker experience any. Instead, it simply provided additional information that proved to be interesting for a handful of subjects on occasion. This conclusion is in line with this experiment's previous research findings. There continues to be a growing trend towards self health care and a major component of this movement is self-education. The presence of the group average was simply another source of data that could be periodically useful and just its presence made many subjects in the Desktop–Competitive Feedback Group feel more empowered with knowledge.

In fact, many Desktop–Competitive Feedback Group subjects specifically expressed their desire for more feedback information by offering a variety of suggestions that would made the group feedback much more applicable to their needs. For example, one subject described how instead of having feedback on the study average, there should be a range on the graph displaying where a healthy person ought to be. Unlike the group feedback, this range would provide users with the motivation to maintain a healthy and well-balanced lifestyle because it would be created by experienced physicians and would thus represent professional advice.

Another suggestion for improving the group feedback was to incorporate HealthTracker into a medical support group. Said one subject: "For the study average, the only way it would be useful is if you were in a support group. Otherwise, it doesn't matter how others are doing, and it may provide incentive to lie so that you don't look bad compared to the group." The rationale behind this suggestion is that without a better understanding of the other people in the study, users had no idea where the group feedback was coming from and if there was any relevance to the information. In a support group setting however, two things are different. First, individuals will know one another and have a much better understanding of their peers' medical background. This information will allow people to make a much more educated decision about the pertinence of the group feedback. Second, support groups are often created around common medical maladies. For example, a support group could be formed for individuals suffering from heart disease. Because of this overlapping interest and common medical condition, group feedback would take on much more significance.

Finally, it is important to note that two users in the Desktop–Competitive Feedback Group greatly enjoyed the group feedback, explaining that they were naturally competitive people and thus liked to see their performance versus any kind of a benchmark.

So it would appear that the concept of group feedback appeals to many people. In general, the more feedback information that is given to an individual, the more empowered that person feels. Though the feedback may not have any direct effect on a person's lifestyle, the simple presence of that information makes people feel more educated about their health and thus more in control. With better framing of group feedback or by using it in an appropriate context like a support group, the usefulness of group feedback can increase dramatically.

10.4 Human Computer Interaction

10.4.1 Trust Relationship with the HealthTracker Software

In the debrief meetings, the majority of users commented on how the HealthTracker system alone was very effective in just making them more aware of their lifestyle habits. Regardless of what platform subjects used or if any feedback was received, almost all users appreciated how the simple process of having to answer a daily questionnaire about their health and well-being made them consciously think about their lifestyle habits. For many people, especially those not receiving feedback, this constant reminder was powerful enough to make them change their daily habits. In this sense, subjects ended up developing a very strong relationship with the HealthTracker software, trusting it enough for it to have affected their daily habits.

While much of the data presented in Sections 8 and 9 indicate that only a handful of subjects exhibited significant improvement in their health and wellness metrics, there are two important considerations to keep in mind. First, the length of the study ranged between 4 to 6 weeks, depending on the user group. This time period might have been too short for any major health changes to register. Many subjects did comment on how they felt themselves interacting with their graphical feedback much more towards the end of the study and might have begun consciously changing daily habits because of the HealthTracker system had the study kept going.

A second consideration is the timing of the HealthTracker experiment. Because it was conducted towards the end of the academic semester, the pressures of final papers and projects were bearing down upon the majority of users and this certainly affected the survey data as well users' overall experience. More generally, by testing HealthTracker on a group of college students, we are limiting our data to a small sample of a much larger market. It would appear that college students are not the most ideal users of a health monitoring system. These individuals are typically young and healthy and though they might be concerned about their health and physical looks, may not have yet become entirely pro-active about wellness and lifestyle management. In fact, less than 15% of the experiment's subjects mentioned an interest in their personal health as a motivation for participating in the study.

Nevertheless, the positive response generated by the HealthTracker system even by this study group indicates that a web-based health monitoring program may receive an even more enthusiastic response from a more appropriate market, such as the elderly or individuals with chronic diseases.

The purpose of the midterm and final questionnaires was to get a sense for how much users trusted the HealthTracker system and how much personal information they were willing to share with the program. The midterm questionnaire asked increasingly personal questions about subjects' family medical history. The questions progressed from inquiries about migraines and allergies to those about AIDS and mental illness. The users were also told at the very beginning of the questionnaire that they were free to skip some or all of the questions in this completely optional survey.

The results of the midterm questionnaire, presented in Sections 8.3 and 9.3, were then collected and the number of skipped questions was then calculated for each individual. As the data indicates, almost all six user groups responded to nearly 100% of the questions. Furthermore, there was no identifiable pattern between the four tiers of the midterm questionnaire. Remember that each progressive tier asked increasingly personal questions. All four tiers were answered in effectively the same level of completeness for all six study groups.

From this data then, it appears that even by halfway through the experiment, subjects already trusted the HealthTracker program enough to volunteer personal medical information. This level of trust is another indication of the tight relationship between the experiment's subjects and the computer software.

The final questionnaire was intended to examine a similar relationship between the users and the software. However, due to technical problems that will be explained in Section 10.5, the data ended up incomplete and thus unusable. Nevertheless, the combined results of the midterm questionnaire and users' comments in the debrief meetings all indicate that a strong relationship had formed between HealthTracker and its users. This conclusion is further supported by past observations that users receiving feedback increased their attention to their graphical feedback as the study progressed.

10.4.2 Building a Relationship with a Wireless Device

Wireless users found their PDA's to be both fun and convenient. Users in the Wireless–No Feedback Group and the Wireless–Individual Feedback Group, who had no limitations on device usage, reported having used the device for a variety of applications ranging from maintaining task lists and phone numbers to games to email and web surfing. Many users also took advantage of the included docking cradle and downloaded a wide variety of additional applications off the Internet. This practice was strongly encouraged in the experiment's orientation because it was the experiment's intention to foster as close a relationship as possible between wireless users and their devices.

Based on the debrief interviews, it would appear that this tight relationship was successfully achieved. The Wireless–No Feedback Group users reported using their devices for 34 minutes per day, on average. 78% of them also carried their devices around with them pretty much every day. The Wireless–Individual Feedback Group subjects used their devices for 52 minutes per day, on average, and 60% carried their devices around with them on a regular basis. Many of those who didn't carry their PDA's around mentioned that they wanted to but were simply afraid of breaking or losing the devices.

To evaluate the effects of this tight relationship between a user and his/her wireless device, we can compare the results of users in the Wireless–Use Limitations Group, who had strict limitations placed upon their use of the PDA, and subjects in the Wireless–No Feedback Group, who had no limitations. Based on data collected in the debrief interviews, users in the Wireless–No Feedback Group ended up being over five times more attached to their wireless devices than users in the Wireless–Use Limitations Group. And, apparently because of this close relationship, the Wireless–No Feedback Group users logged into the HealthTracker system 22% more often than the Wireless–Use Limitations Group subjects. This higher level of compliance indicates a greater amount of desire to interact with the health monitoring program. The Wireless–No Feedback Group users also commented on how their ability to carry the device with them during the day and their increasing dependence on the PDA and its software increased their acceptance and level of interaction with the HealthTracker system.

Being able to answer the questionnaire anytime, anywhere made the survey much more convenient and much less of a chore. Some users who traveled out of town during the study were also able to log in to the system from a variety of major U.S. cities. For these individuals, the convenience and portability of the wireless system was extremely impressive and contributed even further to their appreciation of and interaction with the program.

So, it appears that increased interactions with a wireless device will naturally increase one's attachment to that device. More importantly, as people become increasingly attached to their PDA's, they become increasingly more trusting and interactive with any software running on that device. In the case of the HealthTracker program, the effect of this closer relationship with a wireless PDA was that subjects found the system more enjoyable and were more willing to log in much more frequently.

10.4.3 Comparison of Desktop and Wireless Platforms

In regards to compliance, both the desktop and wireless users were very similar. Desktop users answered 78% of the daily questionnaires on average, while wireless users answered 80.1%. Desktop users logged into the server for 52% of the days in the study, while wireless subjects logged in 57%. For both statistics, wireless users ranked just slightly higher than desktop users.

An explanation for this pattern can be explained by the comments received by wireless users in the debrief questionnaire. In general, most wireless users praised the PDA's portability and the inherent convenience in being able to connect to the HealthTracker service wherever and whenever. 37% of users in the Wireless–No Feedback Group and the Wireless–Individual Feedback Group reported that HealthTracker would have been less useful to them had they accessed the program with a desktop computer. Another 26% of users in the same group added that they too would have found HealthTracker to be less useful on the desktop had there been fewer technical difficulties with the wireless modem and connection to the server.

In general, the portability of the wireless PDA's and the additional ability to personalize the handheld devices seemed to enhance the HealthTracker system. In just a few days, many wireless users had closely integrated the PDA's into their everyday lives, replacing their paper schedules, address books and task lists with the handheld device. This type of dependency only helped to improve the users' relationship with the health monitoring system since integrating the handheld into their lives also meant bringing HealthTracker in more closely as well.

Yet despite these major advantages of the wireless platform over the desktop PC, wireless users responded just slightly less favorably than desktop users regarding how conscious HealthTracker made them about their health and well-being. Wireless users also reported no significant improvement in their health habits over the desktop users. Why then, would an apparently superior platform not produce superior results? The reason most likely lies in the implementation of the HealthTracker system. As has been mentioned before, wireless users encountered a variety of technical problems with the modems, their wireless connection and the HealthTracker system itself. These problems will be covered in Section 10.5.

In addition to these issues however, were other complaints about the general usability of the wireless device. For example, some subjects commented on how the PDA's small display sometimes made it hard to see the questionnaire. Others mentioned how the periodic need to change the device's batteries and to regularly recharge the wireless modem were somewhat of a nuisance. Finally, a handful of users also described how using Graffiti™, Palm's text entry language, severely slowed down the answering of the questionnaire. All of these different complaints about the hardware's usability can take away from the overall HealthTracker experience.

When users access a health monitoring program with a wireless device, they will identify that program with the PDA. So, any complains or problems with the device itself will translate into a less pleasurable experience with the monitoring system itself. With HealthTracker, wireless users weren't able to extract as much usefulness out of the system because of these technical and usability issues. In fact, while 81% of desktop users found the HealthTracker system to be useful, only 76% of wireless users found the same to be true. And while 31% of desktop subjects claimed they would definitely continue using HealthTracker were it available, 28% of wireless users said the same. The differences in these percentages are not enormous, but they do reflect the slight loss of usefulness that resulted from problems wireless users encountered with their hardware.

And so, it is apparent that the concept of connecting to a health monitoring system using a wireless device is extremely positive, even more so than accessing the same program using a desktop PC. These advantages were clearly illustrated in the HealthTracker study by the wireless users' higher level of compliance and positive comments. What prevented these same users from taking full advantage of the wireless platform however, were a series of technical and usability problems. Some of these problems will be addressed in greater detail in Section 10.5 while recommendations for avoiding these same problems and further improving HealthTracker will be provided in Section 11.

10.5 Technical Problems

10.5.1 Server Problems

During the course of the study, especially towards the beginning of the experiment, the server would crash, causing the HealthTracker website to go off-line. There were a number of reasons for this recurring problem. First, the server was a shared resource, so occasionally other projects running simultaneously on the same machine would become unstable and crash the entire server. A second reason was a series of compatibility problems when the Jakarta Tomcat servlet engine was upgraded. When this upgrade occurred, a number of HealthTracker's source files had to be properly reconfigured and then recompiled. These two problems were resolved within the first two weeks of the study, after which point the server remained relatively stable.

One other recurring problem with the server was an apparent incompatibility between some source code in the HealthTracker system and older versions of the Netscape web browser. It turned out that older versions of Netscape were often unable to run the HealthTracker servlet and would return system errors. In addition, some versions of Netscape also didn't allow users to scroll their screens horizontally. As a result, feedback graphs would get cut off by the browser window, preventing users from viewing their most recent feedback information.

10.5.2 Problems with Wireless Service

Problems with the OmniSky modems and its wireless service persisted throughout the course of the HealthTracker study. The modems themselves would often stop working, requiring users to look for technical support. There are a number of possible reasons for these sudden malfunctions. Draining the modem's power or accidentally removing the modem from the Visor's SpringBoard™ expansion slot could both cause the modem to malfunction. Although neither of these two situations should happen. More often than not, the modems would malfunction for no apparent reason. Whenever the modem malfunctioned, the Visor would no longer recognize the attachment, thus requiring a complete new installation of the OmniSky software and a reconfiguration of the modem's IP address and device settings. Specifically, users would receive an "Unrecognized Device ID" error, meaning that the modem had somehow become disassociated from the Visor handheld. However, in most cases the modem had never been removed the SpringBoard™ expansion slot.

Other times, the OmniSky modem would stop working when users changed the batteries on the Visor handheld. Again, this should normally not happen. In these situations, users would receive either an "Unrecognized Device ID" error or a general fault error. In the latter situation, the device would completely freeze, requiring a complete reboot of the Visor handheld.

The wireless service itself was also rather unstable. Depending on user's geographical location, whether he/she were indoors or outdoors and a variety of other factors, the modem's reception varied greatly. Even in Boston, one of the metropolitan areas covered by OmniSky, a few paces to one side or the other outdoors could cause a device to lose its connection. The connection would also quickly disappear as soon as individuals started traveling away from major cities into the suburbs.

As a result of these connection problems, many wireless users would suddenly lose their wireless connection in the middle of filling out their questionnaire or while surfing the Web. Repeated instances of these dropped connections became frustrating for a number of the wireless users. In other situations, the wireless modem simply would not connect to the Web. In these cases, the modem would simply return an "Internal Proxy Error." Only through repeated efforts to connect would the modem eventually allow users to access the Internet and the HealthTracker questionnaire.

10.5.3 Problems with Feedback

Many wireless users complained about problems with the feedback and its accuracy. What was most likely happening was that the PDA was pulling up older versions of the graphical feedback it had stored in its memory cache. Many mobile devices are designed to rely on their cache to speed up the loading time of regularly visited websites. Sometimes the PDA would simply not load any graphs at all. These situations occurred because of technical faults in the wireless modem. The OmniSky modem is designed to transfer data at 19.2 kpbs, and also has a limited display memory for each html page. These specifications allowed for just three feedback graphs per page. Even then, the wireless modem would sometimes have difficulty displaying all three graphs and would then display only one or two. This was the most probable cause for reports of missing feedback graphs.

Finally, a number of wireless users suggested that labeled axes on the feedback graphs would have been extremely useful. The original design of the wireless feedback included labeled axes. However, due to the wireless modem's technical specifications, including labeled axes would have greatly increased the footprint of each graph and would have allowed for only one graph per page. This would translate into a question per page for a total of twenty pages. Such a trade-off would have significantly lengthened the amount of time it took for wireless users to complete the daily questionnaire and would have greatly increased the chances of wireless connection failures and survey submission problems. As a result, it was decided to exclude labeled axes from this implementation of the wireless system. It is recommended, however, that future implementations of a similar system should look for other alternatives so that graphs are properly labelled.

10.5.4 Problems with Final Questionnaire Administration

Due to timing and technical problems, the final questionnaire data unfortunately ended up being incomplete. The final questionnaire was scheduled to go online just before the end of the HealthTracker experiment, so that we could evaluate users' trust relationship with the system. Since the experiment was to end on Monday April 30, 2001, the final questionnaire was integrated into the system on the previous Saturday: April 28, 2001. However, because many subjects did not log into the system on weekends, as they were supposed to, a significant number of users never even read the final questionnaire.

Some subjects also reported errors accessing the final questionnaire when using older versions of the Netscape web browser, a technical problem that was introduced in Section 10.5.1. This issue prevented a handful of additional subjects from accessing the final questionnaire. In all, only 27 subjects were able to successfully complete the questionnaire, even though many more users had tried to. Because of these technical problems and the resulting, incomplete data, we decided to not use results from the final questionnaire.

11 Future Possibilities for HealthTracker

11.1 Who Will Use It

Based on this experiment's findings, students are not the most ideal users of a health monitoring system. The ideal user is an individual with is truly motivated about monitoring his/her lifestyle. Individuals who are still young and who maintain very busy, active schedules would have much less interest in a health monitoring application. For example, many subjects expressed how they didn't feels as if they were benefiting as much as they could from the HealthTracker system simply because of lack of time. Instead, these individuals suggested that HealthTracker would be best suited for people who had enough time to fully appreciate the system and its feedback information. One subject wrote: "I'm not sure [if I would continue using HealthTracker] during the semester, but I would use it during the summer or during any other time I was trying to improve my health."

Elderly people and individuals with chronic illnesses, on the other hand, would be prime examples of people with an immediate motivation to monitor their health and thus ideal users of a system like HealthTracker. These individuals could use the health monitoring system in conjunction with periodic appointments with doctors to help them maintain a lifestyle appropriate to their condition. The simplicity of HealthTracker's design and the ease with which it can be integrated into a person's daily responsibilities make it perfect for people who have a need to monitor their health but also have a full life to lead.

Many mid-aged professionals would also be very interested in health monitoring system for the reasons we explored in Section 3. The growing push towards self health care has many adults interested in practicing preventive medicine and one way to do this is to regularly monitoring one's own health. Again, HealthTracker's simple design and fast questionnaire would allow these people to keep a watchful eye on such things as diet, exercise, sleep, nutrition and mental wellness without investing too much time.

Many subjects in the HealthTracker experiment mentioned how a certain amount of time is needed to study the graphical feedback and to reflect on the daily questions in order to maximize the systems' benefits. Most importantly, this time has to come willingly from the user. Based on the study's results, most individuals who were largely motivated to participate in the study because of the incentives like the $50 compensation the chance to win a free PDA reported a lower than average rating of the system's usefulness. These individuals also exhibited lower than average improvements in their health. A health monitoring system is only there to *help* an individual better understand his/her own health. It thus requires an active effort on the part of the user in order for the application to work. These findings are consistent with the market strategies of online health services. Companies like WebMD and and websites like IntelliHealth.com are looking for a very specific type of consumer, one that is concerned about their health and wellness and willing to be pro-active and interact with their online offerings.

More generally, many subjects discussed how HealthTracker's potential benefits would probably be maximized if it were used in very particular types of situations. For example, as was presented in Section 10.3, one user suggested that HealthTracker would prove to be extremely useful in a medical support group. Other individuals suggested that the system could be best used in conjunction with a set of goals determined either using HealthTracker, or in conjunction with a knowledgeable professional like a doctor, sports trainer or even oneself. With a set of goals, individual would then have more motivation to comply with the daily survey and to pay more attention to its feedback.

Establishing goals would also give more purpose to the health monitoring process. Many subjects commented on how the system certainly made them more conscious of their health and well-being, but the novelty of this awareness could wear off after a while. Within a few weeks, the system could end up being just another daily chore and would lose much of its effectiveness as users stopped enjoying their interactions with the system. The key to any health monitoring program's success is the willingness of the user to embrace the system. For example, one user wrote: "HealthTracker would have been most helpful for me if it were used with individual meetings with another person. For example, when I discussed my [questionnaire] results with my husband, I consciously tried to improve my diet, etc. But if we didn't talk, then I made no attempts at improvements."

The ability to use HealthTracker from a wireless device also opens the doors to some new markets. For example, professionals that are constantly on the road or away from home, such as consultants, salespeople and business executives, could benefit greatly from HealthTracker. Unlike typical web-based health services that are optimized for the desktop computer, the HealthTracker system has also been specially customized for a PDA. As a result, with just a handheld device and a wireless modem plug-in, a person can remain connected to HealthTracker wherever he/she may be. All major handheld devices are already compatible with a variety of wireless technologies and as the technological shift towards mobile computing continues to pick up momentum, more and more device options will become available to these heavy travelers. In the not so distant future, accessing one's health monitoring application could be done using a color touch screen one's smartphone, a convergent device that combines the functionalities of a handheld device and a cellular phone.

11.2 Personalize the System

Many users in the HealthTracker study commented on how questions like "How many cigarettes did you smoke today?" and "How many units of alcohol did you drink today?" could have been omitted from their survey since they didn't drink or didn't smoke. Though not much time was wasted with these questions since they could just be skipped, they certainly weren't necessary. This observation prompted many subjects to suggest that HealthTracker could be improved if its questions were more personalized to individuals' needs.

Some subjects were only really concerned about their eating and dietary habits. The other questions about mental wellness and stress levels were effectively irrelevant and a waste of their time. Other individuals used the HealthTracker system only to monitor their weight and exercise time. People's motivations to use a health monitoring system are going to vary greatly and so the monitoring program ought to address that fact. In future versions of HealthTracker, it may be useful to develop a collection of questionnaires that are targeted for a variety of focus groups. The specificity of these focus groups would be very flexible. For example, three sets of questionnaires could be developed for people interested in their 1) diet, 2) physical health and 3) mental health. With more input from a medical professional, a wider variety of questionnaires could be developed. For example, surveys could be created for diabetics, people suffering from depression, individuals with eating disorders, etc. By custom-tailoring the HealthTracker experience, users will feel that much more engaged with the system and thus more willing to interact and share information with it.

11.3 Make the System More Interactive

To help increase adoption of HealthTracker, it would also be useful to make the system more interactive. The program's effectiveness is largely based upon how much a user interacts with the system and the level of trust between the user and the application. Both of these metrics can potentially be increased with a more interactive program. For example, with the assistance of medical professionals, HealthTracker could be programmed to offer medical advice and informed suggestions based on a user's daily responses. In this manner, not only has the system become more interactive, but it's also further educating the user and giving him/her a greater sense of empowerment–a key element in personal health management.

There are all sorts of possibilities with returning advice. For example, if someone's consumption of fatty foods has been steadily rising, a screen could pop up the next day suggesting to that person to cut back on junk food. More technical suggestions could be possible as well. For example, if a person suffering from depression indicates that his/her mental health has been progressively sinking, then HealthTracker could offer medical advice that could correct this trend or even suggest that the individual visit his/her physician.

Speech recognition software could also be integrated into the HealthTracker program, giving users the option of traditional text entry methods via the keyboard or the more natural option of just speaking to the application. For many people, being able to verbally communicate with HealthTracker would greatly increase their level of comfort with the program and cause them to trust the system more. Individuals who fall into this category would include those who might not be entirely comfortable or familiar with computers, such as the elderly. Many other people simply prefer verbally communicating to somebody or even something, rather than entering responses into a computer or handheld device using a keyboard or stylus. Using voice recognition would also permit individuals with physical handicaps that prevent them from typing or from holding a PDA stylus to utilize the system as well.

HealthTracker could also be programmed to speak back, thus creating a genuine, two-way conversation. Not only would such a design be extremely interactive, but it would also give

HealthTracker personality. One of the trends described in Section 6.1 explained how artificial intelligence may soon replace doctors in many levels of medicine, such as the process of medical diagnoses. Enabling a computer system to effectively hold a dialogue with a person will certainly smooth over the transition from a real doctor giving medical advice to a computer offering the same suggestions. Implementing two-way conversations in HealthTracker would most certainly expand the possibilities of the program as well as potentially increase its effectiveness by setting more people at ease.

Finally, as HealthTracker gains personality and becomes more interactive, its novelty value also increases. When this happen, new markets may become interested in it. For example, young adults and students would probably be much more interested in HealthTracker if it had voice recognition capabilities, could talk back, and offered sage medical advice. With these types of technical developments, the potential user base of HealthTracker and similar programs could greatly increase.

11.4 Create a Network of Users and Medical Professionals

The possibility of integrating HealthTracker into medical focus groups and doctor visits has already been explored. But another closely related possibility for improvement would be to use HealthTracker to network users, hospitals and medical professionals. Doctors could monitor their patients from the hospital while their patients remain in the comfort of their home. If any survey responses become alarming, HealthTracker could be programmed to alert the appropriate medical caretakers who could then determine the next course of action. Communication between the patient and his/her healthcare provider is the one of the most important properties of personal health management and compliance.[27]

The Internet already offers a very similar type of service in which web surfers can go online and ask real doctors for medical advice. The difference between these current websites and the proposed HealthTracker network would be the volumes of health and wellness data generated by the HealthTracker daily questionnaire. With this information, doctors can become immediately familiar with an individual's condition and offer the appropriate advice. Furthermore, there's greater potential for the HealthTracker system to be more personal since the program will be a part of a person's everyday life.

By connecting users with real-life doctors, HealthTracker also becomes even more interactive, and makes many of the examples in Section 11.3 even more compelling. Users could actually have a two-way, verbal conversation with a doctor using HealthTracker as an intermediary. Using either voice conferencing or even instant messaging, HealthTracker could provide its users with instant access to a real medical practitioner should they have any questions or concerns. Doctors would benefit too, since they would have instant access to users' daily questionnaire history and very quickly ramp up to speed on that person's condition.

This network would truly be in the spirit of the HealthTracker program, for it would allow patients to spend less time in the hospital and more time at home. Hospitals today often need to keep patients in the hospital just so that they can regularly monitor particular biometrics and health readings. Depending on the complexity of these readings, much of this could be easily done at home with the assistance of HealthTracker. Not only would such a system save both time and money, it would also make patients feel much more in control of their lives.

11.5 Provide Incentives for Compliance

One of the reasons why compliance in this HealthTracker study was so high was because of the incentives that were promised to those who successfully completed the experiment. And it seems to be the case that college student were not the most ideal users of a health monitoring system. Nevertheless, it would be overly optimistic to expect consistently high levels of compliance from 100% of any demographic. People do get lazy and will most certainly begin ignoring applications like HealthTracker if there isn't enough motivation to push them to answer. This is even true of people who have a medical motivation like a chronic illness.

So what kinds of incentives can be provided for HealthTracker users? While it might not be possible to offer monetary compensation to everybody who uses a medical monitoring system, there certainly are other ways to incentivize compliance. For example, the program could be tied into a rewards program in which users earn points for every survey they complete. This system would be very similar to modern websites like doubleclick.com and winwin.com that reward web surfers for clicking on hyperlinks on their website. The more linked websites individuals visit, the more points they accrue. These points can then be redeemed for merchandise, gift certificates, airline travel, etc.

This type of points and rewards system is a proven technology and has been applied to a variety of industries. Airline's frequent flyer programs and restaurants' frequent diners programs are all examples of this points and rewards system.

Insurance companies could provide yet another type of incentive. Health and life insurance companies both have financial interests in the health and well-being of their customers. Therefore, they could sponsor programs like HealthTracker, since the application's intention is to use preventive medicine to help users avoid illness and hopefully extend their lifetime. Therefore, health and life insurance companies might want to encourage their customers to use HealthTracker. To do this, they could offer to subsidize or even completely pay for the cost of the HealthTracker service depending on a user's level of compliance. The more surveys a user answers, the greater the amount of the subsidy. Alternatively, insurance companies could offer deductions on healthcare premiums or reduced prices for life insurance based on how many HealthTracker daily questionnaires are completed. The more surveys a person answers, the larger their discount.

Of course, one must be extremely careful when associating HealthTracker with any kind of an incentive program. As was observed in the HealthTracker experiment, many individuals who participated in the study for financial reasons ended up getting less out of the experience than most people. Nevertheless, even many of these people did mention that the system made them more conscious of their health. So incentives may still be well worth the effort. Additional research into this possibility would certainly shed more light on the subject and lead to more conclusive judgments.

11.6 The Integration of Future Technology

New, emerging technology like wearable computers could prove to be extremely useful in further improving the overall effectiveness of HealthTracker. For example, incorporating the HealthTracker system into a piece of clothing would truly make the application a part of a person's everyday life. While this function would not be appropriate for everybody, many people could certainly benefit from having a health monitoring system always ready at hand.

The ongoing development of advanced biosensors could further add to the wearable computing experience. For individuals who might need constant health monitoring, it would be a huge achievement if biosensors were integrated into their clothing and then HealthTracker collected the sensors' data and sent it back to a central database. This possibility would further increase the number of patients who would no longer need to be kept in a hospital just to be monitored around the clock.

12 Appendices

Appendix A: The Desktop Orientation Meeting Instructions

Self Health Care Study

Desktop Study – Instructions

Introduction

Thank you for participating in this study on computer-aided self health care. The purpose of this study is to research the potential effects of a web-based health monitoring system on a person's health and well-being. Ultimately, data from this study will help determine whether or not computers can actually assist a person's health or facilitate the monitoring and regulation of a person's health.

The Daily Questionnaire

Everyday, you will be required to answer an online questionnaire. To access the questionnaire, you will need to visit:

http://ilab.media.mit.edu/~healthcare

To login, please use your Athena login name and the password you selected at your orientation. If you are logging in from a secure home machine, you may also want to select the "remember login information" option.

Once in the system, you will need to answer the questionnaire. This questionnaire will ask you different questions about your current health and well-being. Your answers from the previous day will be displayed by default. Please answer these questions to the best of your ability. The most ideal time to answer this questionnaire will be between the hours of 8pm and 11pm, and we encourage you to do so then.

Please answer the questionnaire only once per day, and when submitting the questionnaire, please only click the "submit" button once.

In the event that you forget to answer a questionnaire, you may still do so by logging on and selecting the appropriate "response date" on the questionnaire. Please do this only if you recall your general health and well-being relatively well for that day. You can miss a couple of questionnaire submissions during the study and still qualify for the $50 and chance to win a Handspring Visor.

Conclusion of the Study

This study will conclude in mid- to late- April. At that time, we will schedule a debrief meeting similar to your orientation meeting. At this time, you will also receive your $50 and possibly a Handspring Visor, assuming successful completion of the study.

Final Remarks

During the course of this study, we reserve the right to send additional instructions or modifications to the study, and will do so via your MIT email account.

If you ever have any questions, comments or concerns, please contact Erick Tseng at erick@mit.edu.

Appendix B: The Group 4 Orientation Meeting Instructions

Self Health Care Study
Wireless Devices Study
Instructions – Group 4

Introduction

Thank you for participating in this study on computer-aided self health care. The purpose of this study is to research the potential effects of a web-based health monitoring system on a person's health and well-being. Ultimately, data from this study will help determine whether or not computers can actually assist a person's health or facilitate the monitoring and regulation of a person's health.

The Handspring Visors and Wireless Modem

We will loan you a Handspring Visor and OmniSky wireless modem for the duration of the study. Both the device and the modem must be returned when the study ends. We will also supply batteries for your Visor. A pair of AAA batteries should last you for 2 months. To learn how to use your Visor and the wireless modem, please read the accompanying user's manuals. For the study, you will need to know how to enter numbers into the device. The Visors use an input language called Graffiti, the same language used by Palm Pilots. Figure 1 displays the essentials of Graffiti.

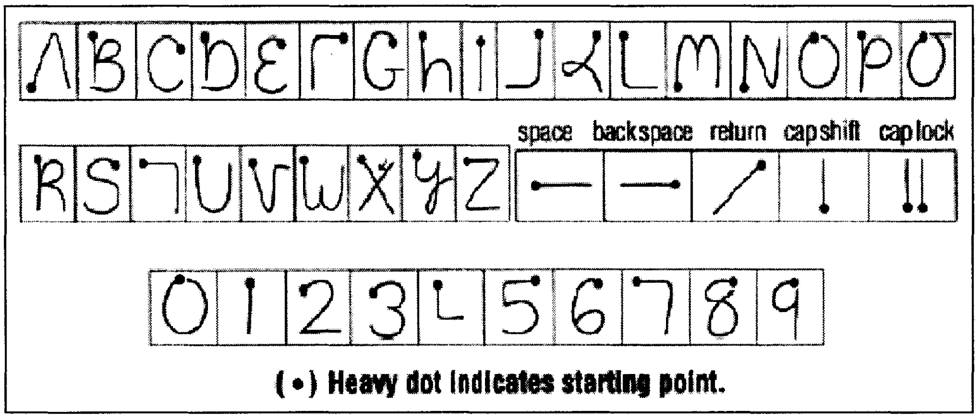


Figure 1: Graffiti Input Language

The wireless modem should work as soon as it is plugged into the SpringBoard expansion slot of your Visor. The modem runs on a rechargeable battery that should be recharged every night with the accompanying AC adapter to ensure proper operation.

To operate the modem, push the single button on the top of the modem. You will be taken to the OmniSky menu as shown in Figure 2. Select the "Web" icon and you will be taken to the OmniSky web browser as shown in Figure 3. Click on the HealthTracker bookmark to access the study website. If you have additional questions about operating your wireless modem, please read the accompanying user's manual or email healthtracker@mit.edu.

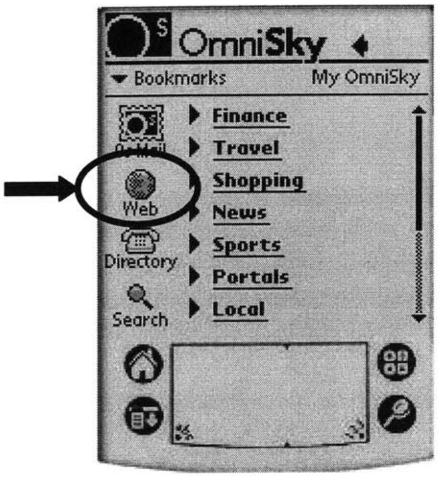


Figure 2: OmniSky Main Menu

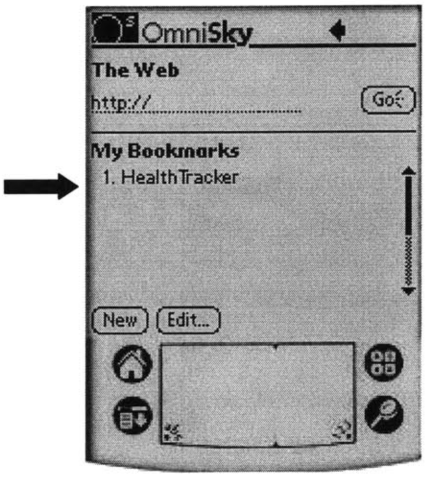


Figure 3: OmniSky Web Browser

For the study, you may *only* use your Handspring Visor and modem to access the study's website, HealthTracker. Using any other applications on the device (including the pre-installed programs) is forbidden. Downloading additional programs onto your device is also forbidden. Finally, you cannot use the device to surf any other websites besides HealthTracker or use any of the other wireless features or services.

In the event that you damage or lose your Visor, the wireless modem, or any of their associated components during the study, you will have to reimburse the study. This reimbursement will be determined on a case-by-case basis.

The Daily Questionnaire

Everyday, you will be required to answer an online questionnaire. You must do so using your Handspring Visor. As described above, open the OmniSky web browser and select the bookmark "HealthTracker" to access the study website. For your reference, the website's URL is http://ilab.media.mit.edu/~healthcare.

Once on the site, you will need to login. To do so, please use your Athena login name and the password you selected at your orientation. To speed up the login process, you may want to select the "remember login information" option that will allow you to save time and avoid entering in your username and password every day.

Once in the system, you will need to answer the questionnaire. This questionnaire will ask you different questions about your current health and well-being. Your answers from the previous day will be displayed by default. Please answer these questions to the best of your ability. The most ideal time to answer this questionnaire will be between the hours of 8pm and 11pm, and we encourage you to do so then.

Please answer the questionnaire only once per day, and when submitting the questionnaire, please only click the "submit" button once.

In the event that you forget to answer a questionnaire, you may still do so by logging on and selecting the appropriate "response date" on the questionnaire. Please do this only if you recall your general health and well-being relatively well for that day. You can miss a couple of questionnaire submissions during the study and still qualify for the $50 and chance to win a Handspring Visor.

Conclusion of the Study

This study will conclude in mid- to late- April. At that time, we will schedule a debrief meeting similar to your orientation meeting. At this time, you will also return the Visor and wireless modem, as well as receive your $50 and possibly a Handspring Visor, assuming successful completion of the study.

Final Remarks

During the course of this study, we reserve the right to send additional instructions or modifications to the study, and will do so via your MIT email account.

If you ever have any questions, comments or concerns, please contact Erick Tseng at healthtracker@mit.edu

Appendix C: The Group 5 Orientation Meeting Instructions

Self Health Care Study

Wireless Devices Study
Instructions – Group 5

Introduction

Thank you for participating in this study on computer-aided self health care. The purpose of this study is to research the potential effects of a web-based health monitoring system on a person's health and well-being. Ultimately, data from this study will help determine whether or not computers can actually assist a person's health or facilitate the monitoring and regulation of a person's health.

The Handspring Visors and Wireless Modem

We will loan you a Handspring Visor and OmniSky wireless modem for the duration of the study. Both the device and the modem must be returned when the study ends. We will also supply batteries for your Visor. A pair of AAA batteries should last you for 2 months. To learn how to use your Visor and the wireless modem, please read the accompanying user's manuals. For the study, you will need to know how to enter numbers into the device. The Visors use an input language called Graffiti, the same language used by Palm Pilots. Figure 1 displays the essentials of Graffiti. We encourage you to become fluent in Graffiti.

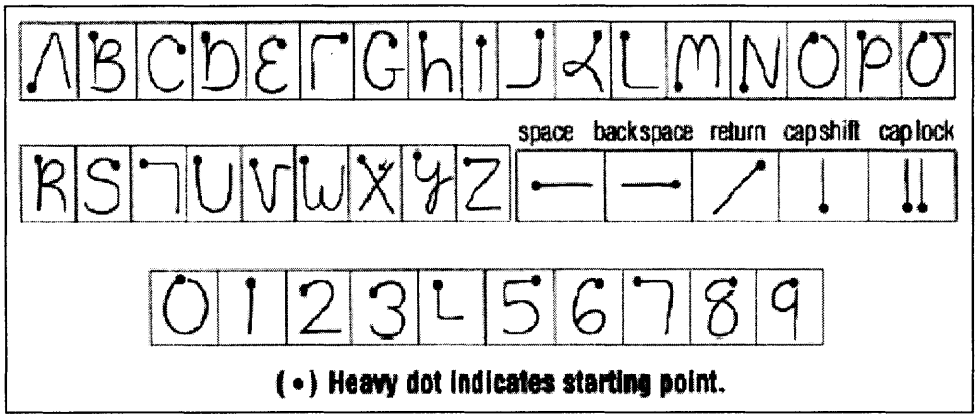


Figure 1: Graffiti Input Language

The wireless modem should work as soon as it is plugged into the SpringBoard expansion slot of your Visor. The modem runs on a rechargeable battery that should be recharged every night with the accompanying AC adapter to ensure proper operation.

To operate the modem, push the single button on the top of the modem. You will be taken to the OmniSky menu as shown in Figure 2. Select the "Web" icon and you will be taken to the

OmniSky web browser as shown in Figure 3. Click on the HealthTracker bookmark to access the study website. If you have additional questions about operating your wireless modem, please read the accompanying user's manual or email healthtracker@mit.edu.

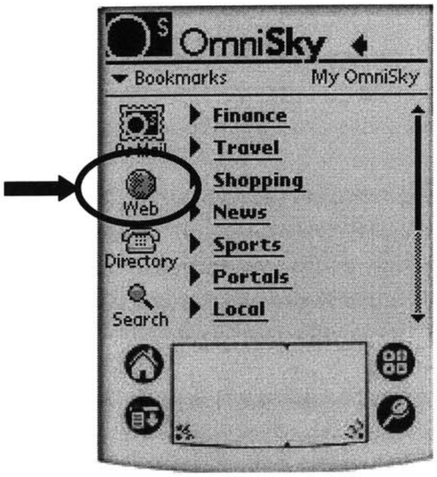


Figure 2: OmniSky Main Menu

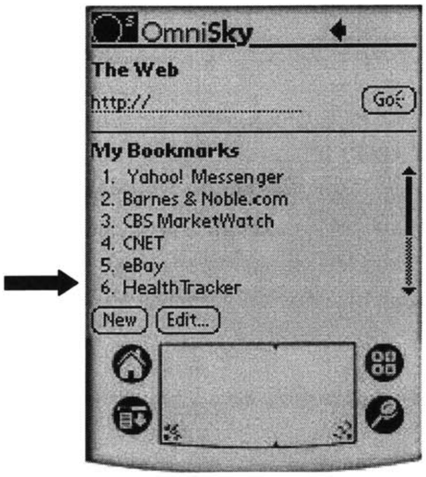


Figure 3: OmniSky Web Browser

Throughout the study, we encourage you to explore your device and use it as much as possible. With your Visor and wireless modem, you can check your email, trade stocks, reserve movie tickets, read the news, keep an address book, take notes, play games, and much more. You can even download free programs onto your device from the web. To do so, you will need to plug your Handspring cradle into your home computer and run the installation CD-ROM. The process is very simple and upon completion, you will also be able to synchronize data between your Visor and your desktop. For more details, consult your user's manual.

Check out http://www.handspring.com/software for free program downloads. Available applications include:

In the event that you damage or lose your Visor, the wireless modem, or any of their associated components during the study, you will have to reimburse the study. This reimbursement will be determined on a case-by-case basis.

The Daily Questionnaire

Everyday, you will be required to answer an online questionnaire. You must do so using your Handspring Visor. As described above, open the OmniSky web browser and select the bookmark "HealthTracker" to access the study website. For your reference, the website's URL is http://ilab.media.mit.edu/~healthcare.

Once on the site, you will need to login. To do so, please use your Athena login name and the password you selected at your orientation. To speed up the login process, you may want to select the "remember login information" option that will allow you to save time and avoid entering in your username and password every day.

Once in the system, you will need to answer the questionnaire. This questionnaire will ask you different questions about your current health and well-being. Your answers from the previous day will be displayed by default. Please answer these questions to the best of your ability. The most ideal time to answer this questionnaire will be between the hours of 8pm and 11pm, and we encourage you to do so then.

Please answer the questionnaire only once per day, and when submitting the questionnaire, please only click the "submit" button once.

In the event that you forget to answer a questionnaire, you may still do so by logging on and selecting the appropriate "response date" on the questionnaire. Please do this only if you recall your general health and well-being relatively well for that day. You can miss a couple of questionnaire submissions during the study and still qualify for the $50 and chance to win a Handspring Visor.

Conclusion of the Study

This study will conclude in mid- to late- April. At that time, we will schedule a debrief meeting similar to your orientation meeting. At this time, you will also return the Visor and wireless modem, as well as receive your $50 and possibly a Handspring Visor, assuming successful completion of the study.

Final Remarks

During the course of this study, we reserve the right to send additional instructions or modifications to the study, and will do so via your MIT email account.

If you ever have any questions, comments or concerns, please contact Erick Tseng at healthtracker@mit.edu

Appendix D: The Group 6 Orientation Meeting Instructions

Self Health Care Study

Wireless Devices Study
Instructions – Group 6

Introduction

Thank you for participating in this study on computer-aided self health care. The purpose of this study is to research the potential effects of a web-based health monitoring system on a person's health and well-being. Ultimately, data from this study will help determine whether or not computers can actually assist a person's health or facilitate the monitoring and regulation of a person's health.

The Handspring Visors and Wireless Modem

We will loan you a Handspring Visor and OmniSky wireless modem for the duration of the study. Both the device and the modem must be returned when the study ends. We will also supply batteries for your Visor. A pair of AAA batteries should last you for 2 months. To learn how to use your Visor and the wireless modem, please read the accompanying user's manuals. For the study, you will need to know how to enter numbers into the device. The Visors use an input language called Graffiti, the same language used by Palm Pilots. Figure 1 displays the essentials of Graffiti. We encourage you to become fluent in Graffiti.

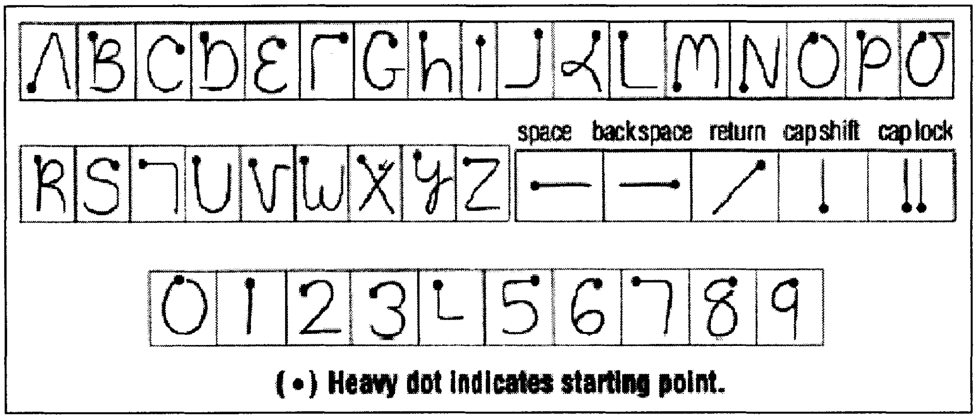


Figure 1: Graffiti Input Language

The wireless modem should work as soon as it is plugged into the SpringBoard expansion slot of your Visor. The modem runs on a rechargeable battery that should be recharged every night with the accompanying AC adapter to ensure proper operation.

To operate the modem, push the single button on the top of the modem. You will be taken to the OmniSky menu as shown in Figure 2. Select the "Web" icon and you will be taken to the

OmniSky web browser as shown in Figure 3. Click on the HealthTracker bookmark to access the study website. If you have additional questions about operating your wireless modem, please read the accompanying user's manual or email healthtracker@mit.edu.

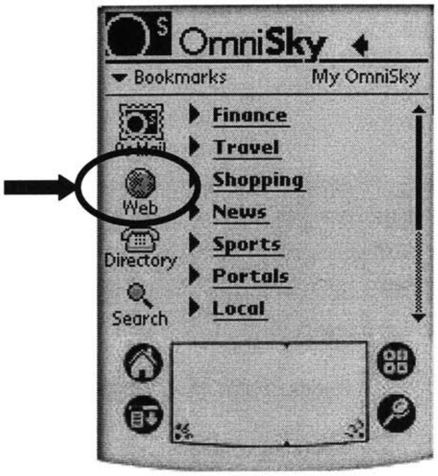


Figure 2: OmniSky Main Menu

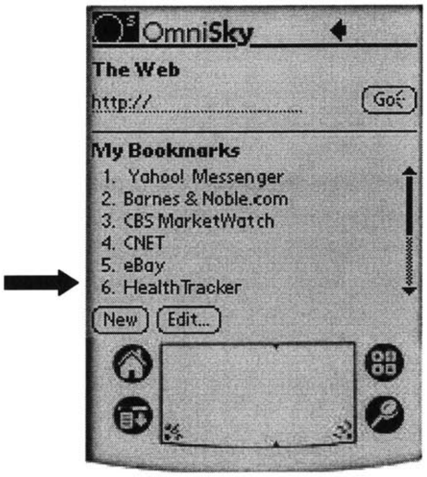


Figure 3: OmniSky Web Browser

Throughout the study, we encourage you to explore your device and use it as much as possible. With your Visor and wireless modem, you can check your email, trade stocks, reserve movie tickets, read the news, keep an address book, take notes, play games, and much more. You can even download free programs onto your device from the web. To do so, you will need to plug your Handspring cradle into your home computer and run the installation CD-ROM. The process is very simple and upon completion, you will also be able to synchronize data between your Visor and your desktop. For more details, consult your user's manual.

Check out http://www.handspring.com/software for free program downloads.

In the event that you damage or lose your Visor, the wireless modem, or any of their associated components during the study, you will have to reimburse the study. This reimbursement will be determined on a case-by-case basis.

The Daily Questionnaire

Everyday, you will be required to answer an online questionnaire. You must do so using your Handspring Visor. As described above, open the OmniSky web browser and select the bookmark "HealthTracker" to access the study website. For your reference, the website's URL is http://ilab.media.mit.edu/~healthcare.

Once on the site, you will need to login. To do so, please use your Athena login name and the password you selected at your orientation. To speed up the login process, you may want to select the "remember login information" option that will allow you to save time and avoid entering in your username and password every day.

Once in the system, you will need to answer the questionnaire. This questionnaire will ask you different questions about your current health and well-being. Your answers from the previous day will be displayed by default. Please answer these questions to the best of your ability. The most ideal time to answer this questionnaire will be between the hours of 8pm and 11pm, and we encourage you to do so then.

Next to every question, you will also see graphs illustrating your responses over time. This feedback can be extremely useful in helping you to control your health and well-being. We encourage you to study these graphs and let them help you make wise health decisions.

Please answer the questionnaire only once per day, and when submitting the questionnaire, please only click the "submit" button once.

In the event that you forget to answer a questionnaire, you may still do so by logging on and selecting the appropriate "response date" on the questionnaire. Please do this only if you recall your general health and well-being relatively well for that day. You can miss a couple of questionnaire submissions during the study and still qualify for the $50 and chance to win a Handspring Visor.

Conclusion of the Study

This study will conclude in mid- to late- April. At that time, we will schedule a debrief meeting similar to your orientation meeting. At this time, you will also return the Visor and wireless modem, as well as receive your $50 and possibly a Handspring Visor, assuming successful completion of the study.

Final Remarks

During the course of this study, we reserve the right to send additional instructions or modifications to the study, and will do so via your MIT email account.

If you ever have any questions, comments or concerns, please contact Erick Tseng at healthtracker@mit.edu

Appendix E: The Orientation Questionnaire

Self Health Care Study

Orientation Questionnaire

First Name______________________________ Last Name______________________________
Athena Username__________________________ Preferred Password______________________
Where will you be over spring break?____

1) On average, how many hours do you sleep per night? (please round to the nearest hour)

2) How many complete meals do you usually eat per day?

3) Please express the relative amount of fatty foods in your diet on a scale from 0 to 100, where 0 is "none," 50 is the "recommended amount," and 100 is "twice as much as needed."

4) Please express the relative amount of fruits and vegetables in your diet on a scale from 0 to 100, where 0 is "none," 50 is the "recommended amount," and 100 is "twice as much as needed."

5) Please express the relative amount of meat in your diet on a scale from 0 to 100, where 0 is "none," 50 is the "recommended amount," and 100 is "twice as much as needed."

6) On average, how many units of caffeinated drinks do you consume daily?

Caffeine Units – Help Box
1 12 oz. can of soda = 1 unit
1 Vivrin pill = 1 unit
1 7 oz. cup of coffee = 2 units
1 7 oz. cup of tea = 1 unit

7) On average, how many units of alcohol do you consume weekly?

Alcohol Units – Help Box
1 12 oz. can of beer = 1 unit
1 glass of wine= 1 unit
1 shot of liquor = 1 unit

8) Express how physically fit you feel on a scale from 0 to 100, where 0 is "completely out of shape" and 100 is "perfectly fit."

9) On average, how many minutes of exercise did you do daily?

10) On average, how many cigarettes do you smoke daily?

11) Express how stressed you are on average on a scale from 0 to 100, where 0 is "not at all" and 100 is "very much."

12) Express how happy you are on average on a scale from 0 to 100, where 0 is "not at all" and 100 is "very much."

13) Express how healthy you are on average on a scale from 0 to 100, where 0 is "not at all" and 100 is "very much."

14) Express how often you get sick on average on a scale from 0 to 100, where 0 is "never" and 100 is "always."

15) How many times have you been sick in the last month?

16) Express how sick you feel now on a scale from 0 to 100, where 0 is "not at all" and 100 is "very much."

17) Express your homework status on average on a scale from 0 to 100, where 0 is "always behind," 50 is "always on schedule," and 100 is "always ahead of schedule."

18) Express how often you worry about academic issues on average on a scale from 0 to 100, where 0 is "always" and 100 is "never."

19) Express how often you worry about non-academic issues on average on a scale from 0 to 100, where 0 is "always" and 100 is "never."

20) Are you worried about your weight?

21) Are you on a diet?

22) Do you consciously monitor your food intake?

23) Do you count calories?

24) Express how interested you are in improving your health on a scale from 0 to 100, where 0 is "not at all interested" and 100 is "extremely interested."

25) Express how interested you are in lowering your weight on a scale from 0 to 100, where 0 is "not at all interested" and 100 is "extremely interested."

26) Express how interested you are in improving your general mental health on a scale from 0 to 100, where 0 is "not at all interested" and 100 is "extremely interested."

27) Express how interested you are in exercising more in general on a scale from 0 to 100, where 0 is "not at all interested" and 100 is "extremely interested."

28) Do you believe a health monitoring application on a wireless device can affect your physical health?

29) Do you believe a health monitoring application on a wireless device can affect your mental health?

Short Answer Response Questions

30) What is your motivation for exercising?

31) What is your motivation for eating healthy?

32) What is your motivation for being healthy?

33) What is your motivation for participating in this study?

The following measurements will be taken using medical devices:

34) What is your current weight?

35) What is your current body fat percentage?

36) What is your current pulse at rest?

37) What is your current blood pressure

Appendix F: The Experiment Participation Consent Form

Self Health Care Study

Desktop Study (Group 3) – Consent Form

I understand that participation in this study is completely voluntary and that I am free to withdraw my consent and to discontinue my participation at any time without prejudice to myself.

Study Procedure
The first task in this study will be to complete a questionnaire that evaluates general physical health as well as attitudes towards health. Next, all participants will receive a personal account to the study's website through which another questionnaire will be answered on a daily basis for approximately six weeks. During the study, participants will also receive feedback illustrating their own responses to the questionnaire versus the study average over time.

In mid- to late-April, the study will conclude with participants' responding to a final questionnaire that will again evaluate general physical health and attitudes towards health. In addition, this final questionnaire will also examine participants' overall response to the study.

Upon successful completion of this study, participants will receive $50 and a chance to win one of several Visor Handspring devices. To successfully complete the study, participants must comply with all the rules of the study as outlined in this document and answer a significant majority of the daily questionnaires.

All information that participants share in this study will remain absolutely confidential and entirely anonymous both during the study and forever afterwards. In addition, all the data that is collected will be reported in such a way that the identity of every participant will be protected.

Study Intention
The purpose of this study is to research the potential effects of a web-based health monitoring system on a person's health and well-being. Ultimately, data from this study will help determine whether or not computers can actually assist a person's health or facilitate the monitoring and regulation of a person's health.

If you have any additional questions or need any points of clarification, please contact Erick Tseng by emailing him at erick@mit.edu.

Consent (A standard statement for all medical-related research at MIT)
In the unlikely event of physical injury resulting from participation in this research, I understand that medical treatment will be available from the M.I.T. Medical Department, including first aid emergency treatment and follow-up care as needed, and that my insurance carrier may be billed for the cost of such treatment. However, no compensation can be provided for medical care apart from the foregoing. I further understand that making such medical treatment available; or providing it, does not imply that such injury is the Investigator's fault. Further information may be obtained by calling the Institute's Insurance and Legal Affairs Office at 253-2822.

I understand that by my participation in this study I am not waiving any of my legal rights. I understand that I may also contact the Chairman of the Committee on the Use of Humans as Experimental Subjects, M.I.T. 253-6787, if I feel I have been treated unfairly as a subject.

By signing below, I consent to my participation in this study and agree to all aspects of this study as outlined in this document.

Name (Print)	Signature	Date

Appendix G: The Daily Questionnaire

Self Health Care Study

Daily Questionnaire

1) On average, how many hours do you sleep per night? (please round to the nearest hour)

2) How many complete meals do you usually eat per day?

3) Please express the relative amount of fatty foods in your diet today on a scale from 0 to 100, where 0 is "none," 50 is the "recommended amount," and 100 is "twice as much as needed."

4) Please express the relative amount of fruits and vegetables in your diet today on a scale from 0 to 100, where 0 is "none," 50 is the "recommended amount," and 100 is "twice as much as needed."

5) Please express the relative amount of meat in your diet today on a scale from 0 to 100, where 0 is "none," 50 is the "recommended amount," and 100 is "twice as much as needed."

6) On average, how many units of caffeinated drinks did you consume today?

7) On average, how many units of alcohol did you consume today?

Caffeine Units – Help Box

1 12 oz. can of soda = 1 unit
1 Vivrin pill = 1 unit
1 7 oz. cup of coffee = 2 units
1 7 oz. cup of tea = 1 unit

8) How many minutes of exercise did you do today?

9) On average, how many cigarettes do you smoke daily?

10) What is your pulse at rest? (measured over a minute)

Alcohol Units – Help Box

1 12 oz. can of beer = 1 unit
1 glass of wine= 1 unit
1 shot of liquor = 1 unit

11) What is your weight? (in pounds)

12) Express how stressed you are on average on a scale from 0 to 100, where 0 is "not at all" and 100 is "very much."

13) Express how happy you are on average on a scale from 0 to 100, where 0 is "not at all" and 100 is "very much."

14) Express how healthy you are on average on a scale from 0 to 100, where 0 is "not at all" and 100 is "very much."

15) Express your homework status on average on a scale from 0 to 100, where 0 is "very behind," 50 is "on schedule," and 100 is "always ahead of schedule."

16) Express how often you worry about academic issues on average on a scale from 0 to 100, where 0 is "always" and 100 is "never."

17) Express how often you worry about non-academic issues on average on a scale from 0 to 100, where 0 is "always" and 100 is "never."

18) Express how well you kept to your general diet today on a scale from 0 to 100, where 0 is "did not follow it at all" and 100 is "follow it perfectly"

19) Express how closely you watched your food intake today on a scale from 0 to 100, where 0 is "not at all" and 100 is "constantly."

20) Express how closely you counted calories today on a scale from 0 to 100, where 0 is "not at all" and 100 is "constantly."

Appendix H: The Midterm Questionnaire

Self Health Care Study

Mid-Point Questionnaire

Thank you for your continued interaction with HealthTracker. At this time, you are invited to answer supplementary questions. Your responses to these questions will enable HealthTracker to better understand your medical history and improve its services.

This questionnaire is given only this one time. After completing this questionnaire, you will then be taken to the daily questionnaire. As always, you are free to answer all, some or none of the questions.

Please indicate with an "x" if any of these individuals in your immediate family have suffered from any of the listed conditions or ailments.

Tier One

	Yourself	Mother	Father	Sibling(s)	None
Broken Bones	□	□	□	□	□
Wear Glasses/Contacts	□	□	□	□	□
Allergies	□	□	□	□	□
Asthma	□	□	□	□	□
Arthritis	□	□	□	□	□
Back Pain	□	□	□	□	□

Tier Two

	Yourself	Mother	Father	Sibling(s)	None
Chronic Fatigue	□	□	□	□	□
Diabetes	□	□	□	□	□
Insomnia	□	□	□	□	□
High Blood Pressure	□	□	□	□	□
Osteoporosis	□	□	□	□	□
Migraines	□	□	□	□	□
Weak joints	□	□	□	□	□

Tier Three

	Yourself	Mother	Father	Sibling(s)	None
Heart Disease	□	□	□	□	□
Glaucoma	□	□	□	□	□
Dyslexia	□	□	□	□	□
Depression	□	□	□	□	□
Multiple Sclerosis	□	□	□	□	□

Tier Four

	Yourself	Mother	Father	Sibling(s)	None
Cancer	□	□	□	□	□
Epilepsy	□	□	□	□	□
Alcoholism	□	□	□	□	□
Alzheimer's Disease	□	□	□	□	□
AIDS	□	□	□	□	□
Hepatitis C	□	□	□	□	□
Parkinson's Disease	□	□	□	□	□
Mental Illness	□	□	□	□	□
Stroke	□	□	□	□	□
Herpes	□	□	□	□	□

Appendix I: The Final Questionnaire

HealthTracker

Final Questionnaire

Thank you for your continued interaction with HealthTracker. At this time, you are invited to answer some supplementary questions. Your responses to these questions will enable HealthTracker to better understand your physical fitness and improve its services.

This questionnaire is given only this one time. After completing this questionnaire, you will then be taken to the daily questionnaire. As always, you are free to answer all, some or none of the questions.

A) What is your current pulse? (measured over a minute)

B) Please stand up straight and then touch your toes. Repeat this five times.

   1. Express how difficult you found this exercise on a scale from 0 to 10, where 0 is "not at all difficult" and 10 is "extremely difficult."
   2. What is your current pulse (measured over a minute)?

C) Please do 10 jumping jacks.

   1. Express how difficult you found this exercise on a scale from 0 to 10, where 0 is "not at all difficult" and 10 is "extremely difficult."
   2. What is your current pulse (measured over a minute)?

D) Please jog in place for a minute.

   1. Express how difficult you found this exercise on a scale from 0 to 10, where 0 is "not at all difficult" and 10 is "extremely difficult."
   2. What is your current pulse (measured over a minute)?

E) Please do 10 sit-ups.

   1. Express how difficult you found this exercise on a scale from 0 to 10, where 0 is "not at all difficult" and 10 is "extremely difficult."
   2. What is your current pulse (measured over a minute)?

Appendix J: The Debrief Questionnaire

Self Health Care Study

Debrief Questionnaire

First Name______________________________ Last Name_____________________________

1) Did HealthTracker make you more conscious of your diet? Please express answer on a scale of 0-100 where 0 "not at all" and 100 "very much so."

2) Did HealthTracker make you more conscious of your physical fitness? Please express answer on a scale of 0-100 where 0 "not at all" and 100 "very much so."

3) Did HealthTracker make you more conscious of your sleeping habits? Please express answer on a scale of 0-100 where 0 "not at all" and 100 "very much so."

4) Did HealthTracker make you more conscious of your consumption of alcohol? Please express answer on a scale of 0-100 where 0 "not at all" and 100 "very much so."

5) Did HealthTracker make you more conscious of how much you exercise? Please express answer on a scale of 0-100 where 0 "not at all" and 100 "very much so."

6) Did HealthTracker make you more conscious of your work habits? Please express answer on a scale of 0-100 where 0 "not at all" and 100 "very much so."

7) Did HealthTracker make you more conscious of your general happiness? Please express answer on a scale of 0-100 where 0 "not at all" and 100 "very much so."

8) Did HealthTracker make you more conscious of your general health and well-being? Please express answer on a scale of 0-100 where 0 "not at all" and 100 "very much so."

9) Did HealthTracker make you more conscious of your mental health? Please express answer on a scale of 0-100 where 0 "not at all" and 100 "very much so."

10) Express HealthTracker's effect on your sleeping habits on a scale from 0 to 100 where 0 is "significantly hurt my sleeping habits," 50 is "no effect" and 100 "significantly improved my sleeping habits."

11) Express HealthTracker's effect on your consumption of fatty foods on a scale from 0 to 100 where 0 is "significantly reduced my intake of these foods," 50 is "no effect" and 100 "significantly increased my intake of these foods."

12) Express HealthTracker's effect on your consumption of fruits and vegetables on a scale from 0 to 100 where 0 is "significantly reduced my intake of these foods," 50 is "no effect" and 100 "significantly increased my intake of these foods."

13) Express HealthTracker's effect on your consumption of meat on a scale from 0 to 100 where 0 is "significantly reduced my intake of meat," 50 is "no effect" and 100 "significantly increased my intake of meat."

14) Express HealthTracker's overall effect on your daily diet on a scale from 0 to 100 where 0 is "significantly hurt my diet," 50 is "no effect" and 100 "significantly improved my diet."

15) Express HealthTracker's overall effect on your consumption of caffeine on a scale from 0 to 100 where 0 is "significantly reduced my intake of caffeine," 50 is "no effect" and 100 "significantly increased my intake of caffeine."

16) Express HealthTracker's overall effect on your consumption of alcohol on a scale from 0 to 100 where 0 is "significantly reduced my intake of alcohol," 50 is "no effect" and 100 "significantly increased my intake of alcohol."

17) Express HealthTracker's overall effect on your physical fitness on a scale from 0 to 100 where 0 is "significantly hurt my fitness," 50 is "no effect" and 100 "significantly improved my fitness."

18) Express HealthTracker's overall effect on your level of exercise on a scale from 0 to 100 where 0 is "significantly reduced how much I exercise," 50 is "no effect" and 100 "significantly increased how much I exercise."

19) Express HealthTracker's overall effect on your smoking habits on a scale from 0 to 100 where 0 is "significantly reduced how much I smoke," 50 is "no effect" and 100 "significantly increased how much I smoke."

20) Express HealthTracker's overall effect on your levels of stress on a scale from 0 to 100 where 0 is "significantly reduced my stress levels," 50 is "no effect" and 100 "significantly increased my stress levels."

21) Express HealthTracker's overall effect on your happiness on a scale from 0 to 100 where 0 is "significantly reduced my happiness," 50 is "no effect" and 100 "significantly increased my happiness."

22) Express HealthTracker's overall effect on your health on a scale from 0 to 100 where 0 is "significantly hurt my health," 50 is "no effect" and 100 "significantly improved my health."

23) Express HealthTracker's overall effect on your homework status on a scale from 0 to 100 where 0 is "significantly hurt my homework status," 50 is "no effect" and 100 "significantly improved my homework status."

24) Express HealthTracker's overall effect on how much you worry about academic issues on a scale from 0 to 100 where 0 is "significantly reduced how much I worry," 50 is "no effect" and 100 "significantly increased how much I worry."

25) Express HealthTracker's overall effect on how much you worry about non-academic issues on a scale from 0 to 100 where 0 is "significantly reduced how much I worry," 50 is "no effect" and 100 "significantly increased how much I worry."

26) Express how interested you are in improving your health on a scale from 0 to 100, where 0 is "not at all interested" and 100 is "extremely interested."

27) Express how interested you are in lowering your weight on a scale from 0 to 100, where 0 is "not at all interested" and 100 is "extremely interested."

28) Express how interested you are in improving your general mental health on a scale from 0 to 100, where 0 is "not at all interested" and 100 is "extremely interested."

29) Express how interested you are in exercising more in general on a scale from 0 to 100, where 0 is "not at all interested" and 100 is "extremely interested."

30) Do you believe a health monitoring application on a wireless device can affect your physical health?

31) Do you believe a health monitoring application on a wireless device can affect your mental health?

32) Are you worried about your weight?

33) Are you on a diet?

34) Do you consciously monitor your food intake?

35) Do you count calories?

Short Answer Response Questions

36) What did you like about the HealthTracker system?

37) What did you dislike about the HealthTracker system?

38) Did you find HealthTracker to be useful? Please explain.

39) Did you find the system to be user-friendly? Why or why not?

40) Would you continue using the HealthTracker system if it were available?

The following measurements will be taken using medical devices

41) What is your current weight?

42) What is your current body fat percentage?

43) What is your current pulse at rest?

44) What is your current blood pressure?

Appendix K: Supplementary Debrief Questionnaire – Group 1

Self Health Care Study

Debrief Questionnaire – Group 1

First Name______________________________ Last Name______________________________

1) Do you think HealthTracker would have been more useful had we given you feedback (graphical, numerical, etc.) of your responses to the daily questionnaire over time? Why or why not?

2) Do you think HealthTracker would have been more useful had we given you feedback (graphical, numerical, etc.) of your responses to the daily questionnaire as well as the study average§ over time? Why or why not? (§ the study average averages the daily responses of all the study's participants for each question.)

Appendix L: Supplementary Debrief Questionnaire – Group 2

Self Health Care Study

Debrief Questionnaire – Group 2

First Name_______________________________ Last Name_______________________________

1) Express on a scale of 0 to 100, how useful you found your graphical feedback, where 0 is "not at all useful" and 100 is "extremely useful."

2) Please explain your response to Question #1.

3) Express on a scale of 0 to 100, how much the graphical feedback affected your responses to the questionnaire, where 0 is "not at all" and 100 "very much so."

4) Please explain your response to Question #3.

5) Express on a scale of 0 to 100, how much attention you gave to the graphical feedback towards the beginning of the study, where 0 is "no attention" and 100 "a lot of attention."

6) Express on a scale of 0 to 100, how much attention you gave to the graphical feedback towards the end of the study, where 0 is "no attention" and 100 "a lot of attention."

7) Please explain your responses to Questions #5 and #6.

8) Express on a scale of 0 to 100, how much the graphical feedback changed your health habits (like eating, exercise, sleep, etc), where 0 is "not at all" and 100 "very much so."

9) Would it have been useful if feedback were also given that displayed how your responses compared to the study average[§] over time? Why or why not? ([§] the study average averages the daily responses of all the study's participants for each question.)

10) Please list additional comments about the graphical feedback on the back of this page. (e.g. suggestions for improvement, What did you like/dislike about the feedback, etc.)

Appendix M: Supplementary Debrief Questionnaire – Group 3

Self Health Care Study

Debrief Questionnaire – Group 3

First Name______________________________ Last Name______________________________

1) Express on a scale of 0 to 100, how useful you found the personal graphical feedback, where 0 is "not at all useful" and 100 is "extremely useful."

2) Express on a scale of 0 to 100, how useful you found the graphical feedback illustrating the study average over time, where 0 is "not at all useful" and 100 is "extremely useful."

3) Please explain your responses to Questions #1 and #2.

4) Express on a scale of 0 to 100, how much the personal[1] graphical feedback affected your responses to the questionnaire, where 0 is "not at all" and 100 "very much so."

5) Express on a scale of 0 to 100, how much the study average feedback affected your responses to the questionnaire, where 0 is "not at all" and 100 "very much so."

6) Please explain your responses to Questions #4 and #5.

7) Express on a scale of 0 to 100, how much attention you gave to the graphical feedback towards the beginning of the study, where 0 is "no attention" and 100 "a lot of attention."

8) Express on a scale of 0 to 100, how much attention you gave to the graphical feedback towards the end of the study, where 0 is "no attention" and 100 "a lot of attention."

9) Please explain your responses to Questions #7 and #8.

10) Express on a scale of 0 to 100, how much the personal feedback changed your health habits (like eating, exercise, sleep, etc), where 0 is "not at all" and 100 "very much so."

11) Express on a scale of 0 to 100, how much the study average feedback changed your health habits (like eating, exercise, sleep, etc), where 0 is "not at all" and 100 "very much so."

12) Please write additional comments about the graphical feedback on the back of this page. (e.g. suggestions for improvement, What did you like/dislike about the feedback, etc.)

Appendix N: Supplementary Debrief Questionnaire – Group 4

Self Health Care Study

Debrief Questionnaire – Group 4

First Name______________________________ Last Name______________________________

1) Express on a scale of 0 to 100, how attached you feel to your Visor, where 0 is "not at all attached" and 100 "very attached."

2) Express on a scale of 0 to 100, how frustrating you found the limitations placed on your use of the Visor, where 0 is "not at all frustrating " and 100 "very frustrating."

3) Please explain your response to Question #1.

4) Do you think HealthTracker would have been more useful/helpful had there been no limitations on your use of the Visor? Why or why not?

5) Do you think HealthTracker would have had a greater effect on your health and well-being had there been no limitations on your use of the Visor? Why or why not?

6) Do you think you would have been more attached to the Visor had there been no limitations on your use of the Visor?

7) Do you think HealthTracker would have been more useful had we given you feedback (graphical, numerical, etc.) of your responses to the daily questionnaire over time? Why or why not?

8) Do you think HealthTracker would have been more useful had we given you feedback (graphical, numerical, etc.) of your responses to the daily questionnaire as well as the study average[§] over time? Why or why not? ([§] the study average averages the daily responses of all the study's participants for each question.)

Appendix O: Supplementary Debrief Questionnaire – Group 5

Self Health Care Study

Debrief Questionnaire – Group 5

First Name_______________________________ Last Name_______________________________

1) On average, how much time did you spend interacting with the Visor every day?

2) Did you regularly carry your Visor around with you? (versus leaving it at home during the day.)

3) Express on a scale of 0 to 100 how attached you feel to your Visor, where 0 is "not at all attached" and 100 "very attached."

4) Please explain your response to Question #3.

5) What did you use your Visor for, besides accessing HealthTracker?

6) Do you think HealthTracker would have had a greater or lesser effect on your health and well-being had you accessed the system via a desktop computer? Please explain.

7) Do you think you would have been more or less attached to the HealthTracker system had you accessed the system via a desktop computer? Please explain.

8) Do you think HealthTracker would have been more useful had we given you feedback (graphical, numerical, etc.) of your responses to the daily questionnaire over time? Why or why not?

9) Do you think HealthTracker would have been more useful had we given you feedback (graphical, numerical, etc.) of your responses to the daily questionnaire as well as the study average[§] over time? Why or why not? ([§] the study average averages the daily responses of all the study's participants for each question.)

Appendix P: Supplementary Debrief Questionnaire – Group 6

Self Health Care Study

Debrief Questionnaire – Group 6

First Name____________________________ Last Name____________________________

1) On average, how much time did you spend interacting with the Visor every day?

2) Did you regularly carry your Visor around with you? (versus leaving it at home during the day.)

3) Express on a scale of 0 to 100 how attached you feel to your Visor, where 0 is "not at all attached" and 100 "very attached."

4) Please explain your response to Question #3.

5) What did you use your Visor for, besides accessing HealthTracker?

6) Do you think HealthTracker would have had a greater or lesser effect on your health and well-being had you accessed the system via a desktop computer? Please explain.

7) Do you think you would have been more or less attached to the HealthTracker system had you accessed the system via a desktop computer? Please explain.

8) Express on a scale of 0 to 100, how useful you found your graphical feedback, where 0 is "not at all useful" and 100 is "extremely useful."

9) Please explain your response to Question #8.

10) Express on a scale of 0 to 100, how much the graphical feedback affected your responses to the questionnaire, where 0 is "not at all" and 100 "very much so."

11) Please explain your response to Question #10.

12) Express on a scale of 0 to 100, how much attention you gave to the graphical feedback towards the beginning of the study, where 0 is "no attention" and 100 "a lot of attention."

13) Express on a scale of 0 to 100, how much attention you gave to the graphical feedback towards the end of the study, where 0 is "no attention" and 100 "a lot of attention."

14) Please explain your responses to Questions #12 and #13.

15) Express on a scale of 0 to 100, how much the graphical feedback changed your health habits (like eating, exercise, sleep, etc), where 0 is "not at all" and 100 "very much so."

16) Would it have been useful if feedback were also given that displayed how your responses compared to the study average[§] over time? Why or why not? ([§] the study average averages the daily responses of all the study's participants for each question.)

17) Additional comments about the graphical feedback (e.g. suggestions for improvement, What did you like/dislike about the feedback, etc.)

Appendix Q: HealthTracker Source Code

Q.1 WirelessForm

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import UpdateTable;
import QueryTable;

public class wirelessForm  extends HttpServlet {

  public void doGet(HttpServletRequest request,
              HttpServletResponse response)
    throws IOException, ServletException
  {
    response.setContentType("text/html");

        String username = request.getParameter("username");
        String password = request.getParameter("password");
        PrintWriter out = response.getWriter();

        String[] fileNameArray = new String[22];
        for(int i = 2; i < 22; i++)
        {
          Integer tempInteger = new Integer(i+1);
          String intString = tempInteger.toString();
          String fileName = "";
          fileName = fileName.concat(username);
          fileName = fileName.concat(intString);
          fileName = fileName.concat(".jpg");
          SmallGrapher temp = new SmallGrapher(feedback[i],fileName);
          fileNameArray[i] = fileName;
        }

        int[] previousDay = QueryTable.PreviousDay(userID);
        int[] prevDay = new int[22];
        for(int i = 2; i < 22; i++)
          {
                prevDay[i] = previousDay[i-2];
          }

        int[] userInfo = QueryTable.LoginCheck(username,password);
        int userID = userInfo[1];
        int userGroup = userInfo[0];
        if(userGroup == 0)
          {
                out.println("<html><head><title>Incorrect Login</title><head>");
                out.println("<body><h1><font color=\"red\">Incorrect Login</font>");
                out.println("</body></html>");
          }
        else
          {
                String[] input = new String[22];
```

```
            input[0] = request.getParameter("month");
            input[1] = request.getParameter("date");
            input[2] = request.getParameter("sleep");
            input[3] = request.getParameter("eat");
            input[4] = request.getParameter("fat");
            input[5] = request.getParameter("fruit");
            input[6] = request.getParameter("meat");
            input[7] = request.getParameter("caffeine");
            input[8] = request.getParameter("alchoholamount");
            input[9] = request.getParameter("exercise");
            input[10] = request.getParameter("smokingamount");
            input[11] = request.getParameter("pulse");
            input[12] = request.getParameter("weight");
            input[13] = request.getParameter("stress");
            input[14] = request.getParameter("happiness");
            input[15] = request.getParameter("health");
            input[16] = request.getParameter("homework");
            input[17] = request.getParameter("academicworry");
            input[18] = request.getParameter("nonacademicworry");
            input[19] = request.getParameter("diet");
            input[20] = request.getParameter("watchintake");
            input[21] = request.getParameter("caloriecount");

            boolean badInput = false;
            int[] data = new int[22];
            for(int i = 0 ; i < 22; i++)
              {
                    Integer tempInteger = new Integer(0);
                    try
                      {
                            tempInteger = new Integer(input[i]);
                            data[i] = tempInteger.intValue();
                      }
                    catch(NumberFormatException e)
                      {
                            badInput = true;
                      }
                    data[i] = tempInteger.intValue();
              }
            if(badInput)
              {
                    out.println("<html><title>Health Tracker</title><head></head>");
                    out.println("<body
background=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?image=bg.jpg\"><h1>Health
Tracker</h1><font color=\"red\">Error: non-integer input</font><br>");
                    out.println("You have entered non-integer input.  Please go back and round all data to
the nearest whole number.");
                    out.println("</body></html>");
              }
            else
              {

            UpdateTable.SendData(userID, data);

  /* create new vector for holding all 22 question vectors from db. */
  Vector vectorVector = new Vector();
```

```
/* create variable for storing size of data. */
int feedbackLength = 0;
/* for each question ... */
for(int i = 0; i < 22; i++)
        {
          /* get the vector that holds the array for question i.*/
          Vector v = QueryTable.VisorFdback(userID,i+1);
          /* add it to the vector of all questions.*/
          vectorVector.add(v);
        }
/* allocate the array of questions and values. */
int[][] feedback = new int[22][5];
/* for each question ... */
for(int i = 0; i < 22; i++)
        {
          /* get the vector for question i. */
          Object tempObject = vectorVector.elementAt(i);
          /* cast it as a vector. */
          Vector tempVector = (Vector)tempObject;
          /* get the array from the vector and cast it as an int[]. */
          int[] fi = (int[])(tempVector.elementAt(0) );
          /* store it as question i. */
          feedback[i] = fi;
        }

        /* get a list of all days that have data entry for this user. */
        int[] dataMonths = feedback[0];
        int[] dataDays = feedback[1];

        /* figure out what days of the study they've been doing it for. */
        /* first make array of study day */
        int[] studyDay = new int[dataMonths.length];
        for(int i = 0; i < studyDay.length; i++)
          {
                studyDay[i] = (dataMonths[i] - 3)*31 + dataDays[i];
          }
        /* get maximum and minimum. */
        int max = 0;
        /*int min =
        for(int i = 0; i < studyDay.length; i++)
          {
                if(studyDay[i] > max)
                  {
                        max = studyDay[i];
                  }
                if(studyDay[i] < min)
                  {
                        min = studyDay[i];
                  }
          }
        boolean[] allStudyDays = new boolean[max + 1];
        for(int i = 0; i < allStudyDays.length; i++)
          {
                allStudyDays[i] = false;
                }*/
        /* for each piece of data ... */
        /*for(int i = 0; i < studyDay.length; i++)
           {
                   /* find out what day it is for and check off that day in the list. */
        /*allStudyDays[studyDay[i]] = true;
           }
           int[] finalData = new int[max-min + 1];*/
        /* for each possible date ... */
        /*for(int i = min; i < allStudyDays.length; i++)
          {*/
                   /* check to see if it's */
        /*             } */

  String[] fileNameArray = new String[22];
  for(int i = 2; i < 22; i++)
        {
           Integer tempInteger = new Integer(i+1);
           String intString = tempInteger.toString();
           String fileName = "";
           fileName = fileName.concat(userName);
           fileName = fileName.concat(intString);
           fileName = fileName.concat(".jpg");
           SmallGrapher temp = new SmallGrapher(feedback[i],fileName);
           fileNameArray[i] = fileName;
        }


/* Graphs of previous entries */
out.println("<html>");
out.println("<head>");
out.println("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">");
out.println("<meta name=\"GENERATOR\" content=\"Mozilla/4.75 [en] (X11; U; IRIX 6.5 IP22)
[Netscape]\">");

out.println("<title>Health Tracker</title>");
out.println("</head>");
out.println("<body bgcolor=\"white\">");
out.println("<table border=0 height=1000>");
out.println("<tr valign=top>");
out.println("<td width=150
background=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?image=bg2.jpg\">");
out.println("</td>");
out.println("<td>");
out.println("<h1>Health Tracker</h1>");
out.println("<h2>Daily Questionnaire</h2>");

out.println("<form method=\"GET\" action=\"wirelessForm2\">");
out.println("<input type=\"hidden\" name=\"username\" value=\"" + userName +"\">");
out.println("<input type=\"hidden\" name=\"password\" value=\"" + passWord +"\">");

out.println("<table border=0 cellpadding=40 cols=2 width=\"100%\">");
out.println("<tr valign=top><td>1. Please enter the date for which you are answering the questions.<p>");
GregorianCalendar myCalendar = new GregorianCalendar();
int month = myCalendar.get(GregorianCalendar.MONTH) + 1;
int date = myCalendar.get(GregorianCalendar.DATE);

out.println("Month <input type=\"text\" name=\"month\" value=\"" + month + "\"><p>");
out.println("Date <input type=\"text\" name=\"date\" value=\"" + date + "\"><br></td><td></td></tr>");


out.println("<tr valign=top><td>2. How many hours of sleep did you have last night? (please round to the
nearest hour)<br>");
out.println("<input type=\"text\" name=\"sleep\" value=\"" + prevDay[2] + "\"></td>");
out.println("<td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[2] + "\"></td>");
out.println("</tr>");
out.println("");

out.println("<tr valign=top><td>3. How many complete meals did you eat today?<br>");
out.println("<input type=\"text\" name=\"eat\" value=\"" + prevDay[3] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[3] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>4. Please express the amount of fatty foods in your diet today ");
out.println("on a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"fat\" type=\"text\" value=\"" + prevDay[4] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[4] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>5. Please express the relative amount of fruits and vegetables in your diet
today ");
out.println("on a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"fruit\" type=\"text\" value=\"" + prevDay[5] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[5] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>6. Please express the relative amount of meat in your diet today on ");
out.println("a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"meat\" type=\"text\" value=\"" + prevDay[6] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[6] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>7. How many <a name=\"caffeine\"></a><a href=\"#caffeine\"
onClick=\"window.open('http://ilab.media.mit.edu:8080/healthcare/servlet/getPage?page=caffeine.html','Ca
ffeine','height=400,width=400');\">units of caffeinated drinks</a> did you consume today?<br>");
out.println("<input type=\"text\" name=\"caffeine\" value=\"" + prevDay[7] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[7] +
"\"><br>");
out.println("</td></tr>");

out.println("<tr valign=top><td>8. How many <a name=\"alcohol\"></a><a href=\"#alcohol\"
onClick=\"window.open('http://ilab.media.mit.edu:8080/healthcare/servlet/getPage?page=alcohol.html','Al
cohol','height=400,width=400');\">units of alcohol</a> did you consume today?<br>");
out.println("<input type=\"text\" name=\"alchoholamount\" value=\"" + prevDay[8] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[8] +
"\"><br>");
out.println("</td></tr>");

out.println("<tr valign=top><td>9. How many minutes of exercise did you do today?<br>");
out.println("<input type=\"text\" name=\"exercise\" value=\"" + prevDay[9] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[9] +
"\"><br>");
out.println("</td></tr>");

out.println("<tr valign=top><td>10. How many cigarettes did you smoke today?<br>");
out.println("<input type=\"text\" name=\"smokingamount\" value=\"" + prevDay[10] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[10] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>11. What is your pulse at rest?  (measured over a minute)<br>");
out.println("<input type=\"text\" name=\"pulse\" value=\"" + prevDay[11] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[11] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>12. What is your weight?  (in pounds)<br>");
out.println("<input type=\"text\" name=\"weight\" value=\"" + prevDay[12] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[12] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>13. Express how stressed you were today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all stressed\" and 100 is \"very stressed.\"<br>");
out.println("<input name=\"stress\" type=\"text\" value=\"" + prevDay[13] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[13] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>14. Express how happy you were today on a scale from 0 to 100, where 0
is \"not at all happy\" and 100 is \"very happy\"<br>");
out.println("<input name=\"happiness\" type=\"text\" value=\"" + prevDay[14] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[14] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>15. Express how healthy you were today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all healthy\" and 100 is \"very healthy.\"<br>");
out.println("<input name=\"health\" type=\"text\" value=\"" + prevDay[15] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[15] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>16. Express your homework status today on ");
out.println("a scale from 0 to 100, where 0 is \"very behind,\" 50 is \"on schedule,\" and 100 is \"ahead of
schedule.\"<br>");
out.println("<input name=\"homework\" type=\"text\" value=\"" + prevDay[16] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[16] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>17. Express how often you worried about academic issues today on
average on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"all the time.\"<br>");
out.println("<input name=\"academicworry\" type=\"text\" value=\"" + prevDay[17] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[17] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>18. Express how often you worried about non-academic issues today on
average on ");
out.println("a scale from 0 to 100, w where 0 is \"not at all\" and 100 is \"all the time.\"<br>");
out.println("<input name=\"nonacademicworry\" type=\"text\" value=\"" + prevDay[18] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[18] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>19. Express how well you kept to your general diet today on ");
out.println("a scale from 0 to 100, where 0 is \"did not follow it at all\" and 100 is \"follow it
perfectly\"<br>");
out.println("<input name=\"diet\" type=\"text\" value=\"" + prevDay[19] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[19] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>20. Express how closely you watched your food intake today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"constantly.\"<br>");
out.println("<input name=\"watchintake\" type=\"text\" value=\"" + prevDay[20] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[20] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>21. Express how closely you counted calories today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"constantly.\"<br>");
out.println("<input name=\"caloriecount\" type=\"text\" value=\"" + prevDay[21] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[21] + "\">");
out.println("</td></tr>");
out.println("</table>");

out.println("<input type=\"SUBMIT\" value=\"Submit\" align=\"middle\">");
out.println("</form></td></tr></table>");
out.println("</body>");
out.println("</html>");
            out.println("<html><title>Health Tracker</title><head></head>");
            out.println("<body
background=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?image=bg.jpg\"><h1>Health
Tracker</h1><h2>Submission Successful</h2>Thank you, your information has been submitted.");
           out.println("</body></html>");
              }
             }
   }
}
```

Q.2 FormServlet2

```
//Presents html form to user.
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import QueryTable;
import Query;
import Grapher;

public class formServlet2 extends HttpServlet {

   public void doGet(HttpServletRequest request,
               HttpServletResponse response)
      throws IOException, ServletException
   {
      response.setContentType("text/html");
```

```
PrintWriter out = response.getWriter();
    out.println("<!doctype html public \"-//w3c//dtd html 4.0 transitional//en\">");
    //get name and pass from login page.
String userName = request.getParameter("username");
String passWord = request.getParameter("password");

    //check login info.
    int userInfo[] = QueryTable.LoginCheck(userName, passWord);
    //int userInfo[] = Query.Check();
    int userGroup = userInfo[0];
    int userID = userInfo[1];
    System.out.println("User Group: " + userGroup);

    //check if user has completed one-time midterm questions.
    int midpointDone = QueryTable.MidtermCheck(userID);

    //check if user has completed one-time final questions.
    int finalDone = QueryTable.FinalCheck(userID);
    if(finalDone == 0)
      {
            userGroup = -3;
      }
    if(userGroup == -3)
      {
            //if user hasn't done final questions, print final form.
    File fileToSend = new File("final1.html");
    FileReader frin = new FileReader(fileToSend);
    int nread;
while( ( nread = frin.read() ) >= 0 )
{
            out.write(nread);
}
frin.close();

    //embed user info into form.
    out.print("<input type=\"hidden\" name=\"userID\" value=\"" + userID + "\"><br>");
    out.print("<input type=\"hidden\" name=\"username\" value=\"" + userName + "\"><br>");
    out.print("<input type=\"hidden\" name=\"password\" value=\"" + passWord + "\"><br>");
    Date d = new Date();
    long msecs = d.getTime();
    Long msecsLong = new Long(msecs);
    String msecsString = msecsLong.toString();
    out.println("<input type=\"hidden\" name=\"st1\" value=\"" + msecsString + "\"><br>");

    File fileToSend2 = new File("final1-2.html");
    FileReader frin2 = new FileReader(fileToSend2);
    int nread2;
while( ( nread2 = frin2.read() ) >= 0 )
{
            out.write(nread2);
}
frin2.close();


            Integer idInt = new Integer(userID);
            String idString = idInt.toString();
```

```
       }
    if(midpointDone == 0)
       {
              //if midpoint form not done...
              if(userGroup < 4)
                 {
                        userGroup = -1;
                 }
              else
                 {
                        userGroup = -2;
                 }

       }
    if(userGroup == -1)
       {
              //print out first page of midterm question for desktop users.
    File fileToSend = new File("midques1.html");
    FileReader frin = new FileReader(fileToSend);
    int nread;
while( ( nread = frin.read() ) >= 0 )
{
              out.write(nread);
}
frin.close();

    out.print("<input type=\"hidden\" name=\"userID\" value=\"" + userID + "\">");
    out.print("<input type=\"hidden\" name=\"username\" value=\"" + userName + "\">");
    out.print("<input type=\"hidden\" name=\"password\" value=\"" + passWord + "\">");

    File fileToSend2 = new File("midques2.html");
    FileReader frin2 = new FileReader(fileToSend2);
    int nread2;
while( ( nread2 = frin2.read() ) >= 0 )
{
              out.write(nread2);
}
frin2.close();

              Integer idInt = new Integer(userID);
              String idString = idInt.toString();

       }
    if(userGroup == -2)
       {
              //print out first midterm form page for wireless users.
    File fileToSend = new File("wmidques1.html");
    FileReader frin = new FileReader(fileToSend);
    int nread;
while( ( nread = frin.read() ) >= 0 )
{
              out.write(nread);
}
frin.close();

    out.print("<input type=\"hidden\" name=\"userID\" value=\"" + userID + "\">");
```

```
            out.print("<input type=\"hidden\" name=\"username\" value=\"" + userName + "\">");
            out.print("<input type=\"hidden\" name=\"password\" value=\"" + passWord + "\">");

            File fileToSend2 = new File("wmidques2.html");
            FileReader frin2 = new FileReader(fileToSend2);
            int nread2;
        while( ( nread2 = frin2.read() ) >= 0 )
        {
                    out.write(nread2);
        }
        frin2.close();

                    Integer idInt = new Integer(userID);
                    String idString = idInt.toString();
              }


            if(userGroup == 0)
              {
                    //if user is not registered, print out incorrect login message.
                    out.println("<html><head><title>Health Tracker</title></head><body
background=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?image=bg.jpg\"
><h1>Health Tracker</h1><font color=\"red\"><h2>Incorrect login</h2>");
             out.println("</font>Please go back and re-enter your Username and Password.</body></html>");
              }

            //load previous day's responses to use as defaults in form.
            //int[] prevDay = userInfo;
            int[] previousDay = QueryTable.PreviousDay(userID);
            int[] prevDay = new int[22];
            for(int i = 2; i < 22; i++)
              {
                    prevDay[i] = previousDay[i-2];
              }
        if( (userGroup == 1) || (userGroup == 4) || (userGroup == 5) )
        {
/* No graphs */
out.println("<html>");
out.println("<head>");
out.println("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">");
out.println("<meta name=\"GENERATOR\" content=\"Mozilla/4.75 [en] (X11; U; IRIX 6.5 IP22)
[Netscape]\">");

out.println("<title>Health Tracker</title>");
out.println("</head>");
out.println("<body bgcolor=\"white\">");
out.println("<table border=0 height=1000>");
out.println("<tr valign=top>");
out.println("<td width=150
background=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?image=bg2.jpg\">");
out.println("</td>");
out.println("<td>");
out.println("<h1>Health Tracker</h1>");
out.println("<h2>Daily Questionnaire</h2>");

if(userGroup == 1)
```

```
{
  out.println("<form method=\"GET\" action=\"reviewServlet\">");
}
else
   {
          out.println("<form method=\"GET\" action=\"wirelessUpdate\">");
   }

out.println("<input type=\"hidden\" name=\"username\" value=\"" + userName +"\">");
out.println("<input type=\"hidden\" name=\"password\" value=\"" + passWord +"\">");
out.println("<table border=0 cellpadding=20>");
out.println("<tr><td valign=top>1.</td><td valign=top>Please enter the date for which you are answering
the questions.<p>");
GregorianCalendar myCalendar = new GregorianCalendar();
int month = myCalendar.get(GregorianCalendar.MONTH) + 1;
int date = myCalendar.get(GregorianCalendar.DATE);

out.println("Month <input type=\"text\" name=\"month\" value=\"" + month + "\">");
out.println("<p>Date <input type=\"text\" name=\"date\" value=\"" + date + "\"><br></td></tr>");

out.println("<tr><td valign=top>2.</td><td valign=top>");
out.println("How many hours of sleep did you have last night? (please round to the nearest hour)<br>");
out.println("<input type=\"text\" name=\"sleep\" value=\"" + prevDay[2] + "\">");
out.println("</td></tr>");
out.println("");
out.println("<tr><td valign=top>3.</td><td valign=top>");
out.println("How many complete meals did you eat today?<br>");
out.println("<input type=\"text\" name=\"eat\" value=\"" + prevDay[3] + "\">");
out.println("</td></tr>");

out.println("<tr><td valign=top>4.</td><td valign=top>");
out.println("Please express the relative amount of fatty foods in your diet today");
out.println("on a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"fat\" type=\"text\" value=\"" + prevDay[4] + "\">");
out.println("</td></tr>");

out.println("<tr><td valign=top>5.</td><td valign=top>");
out.println("Please express the relative amount of fruits and vegetables in your diet today ");
out.println("on a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"fruit\" type=\"text\" value=\"" + prevDay[5] + "\">");
out.println("</td></tr>");

out.println("<tr><td valign=top>6.</td><td valign=top>");
out.println("Please express the relative amount of meat in your diet today on ");
out.println("a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"meat\" type=\"text\" value=\"" + prevDay[6] + "\">");
out.println("</input>");
out.println("</td></tr>");

out.println("<tr><td valign=top>7.</td><td valign=top>");
out.println("How many <a name=\"caffeine\"></a><a href=\"#caffeine\"
onClick=\"window.open('http://ilab.media.mit.edu:8080/healthcare/servlet/getPage?page=caffeine.html','Ca
ffeine','height=400,width=400');\">units of caffeinated drinks</a> did you consume today?<br>");
```

```
out.println("<input type=\"text\" name=\"caffeine\" value=\"" + prevDay[7] + "\">");
out.println("</td></tr>");

out.println("<tr><td valign=top>8.</td><td valign=top>");
out.println("How many <a name=\"alcohol\"></a><a href=\"#alcohol\"
onClick=\"window.open('http://ilab.media.mit.edu:8080/healthcare/servlet/getPage?page=alcohol.html','Al
cohol','height=400,width=400');\">units of alcohol</a> did you consume today?<br>");
out.println("<input type=\"text\" name=\"alchoholamount\" value=\"" + prevDay[8] + "\"></li>");
out.println("</td></tr>");

out.println("<tr><td valign=top>9.</td><td valign=top>");
out.println("How many minutes of exercise did you do today?<br>");
out.println("<input type=\"text\" name=\"exercise\" value=\"" + prevDay[9] + "\">");
out.println("</td></tr>");

out.println("<tr><td valign=top>10.</td><td valign=top>");
out.println("How many cigarettes did you smoke today?<br>");
out.println("<input type=\"text\" name=\"smokingamount\" value=\"" + prevDay[10] + "\"></li>");
out.println("</td></tr>");

out.println("<tr><td valign=top>11.</td><td valign=top>");
out.println("What is your pulse at rest?  (measured over a minute)<br>");
out.println("<input type=\"text\" name=\"pulse\" value=\"" + prevDay[11] + "\">");
out.println("</td></tr>");

out.println("<tr><td valign=top>12.</td><td valign=top>");
out.println("What is your weight?  (in pounds)<br>");
out.println("<input type=\"text\" name=\"weight\" value=\"" + prevDay[12] + "\">");
out.println("</td></tr>");

out.println("<tr><td valign=top>13.</td><td valign=top>");
out.println("Express how stressed you were today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all stressed\" and 100 is \"very stressed.\"<br>");
out.println("<input name=\"stress\" type=\"text\" value=\"" + prevDay[13] + "\">");
out.println("</input>");
out.println("</td></tr>");

out.println("<tr><td valign=top>14.</td><td valign=top>");
out.println("Express how happy you were today on a scale from 0 to 100, where 0 is \"not at all happy\"
and 100 is \"very happy\"<br>");
out.println("<input name=\"happiness\" type=\"text\" value=\"" + prevDay[14] + "\">");
out.println("</input>");
out.println("</td></tr>");

out.println("<tr><td valign=top>15.</td><td valign=top>");
out.println("Express how healthy you were today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all healthy\" and 100 is \"very healthy.\"<br>");
out.println("<input name=\"health\" type=\"text\" value=\"" + prevDay[15] + "\">");
out.println("</input>");
out.println("</td></tr>");

out.println("<tr><td valign=top>16.</td><td valign=top>");
out.println("Express your homework status today on ");
out.println("a scale from 0 to 100, where 0 is \"very behind,\" 50 is \"on schedule,\" and 100 is \"ahead of
schedule.\"<br>");
out.println("<input name=\"homework\" type=\"text\" value=\"" + prevDay[16] + "\">");
out.println("</input>");
out.println("</td></tr>");

out.println("<tr><td valign=top>17.</td><td valign=top>");
out.println("Express how often you worried about academic issues today on average on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"all the time.\"<br>");
out.println("<input name=\"academicworry\" type=\"text\" value=\"" + prevDay[17] + "\">");
out.println("</input>");
out.println("</td></tr>");

out.println("<tr><td valign=top>18.</td><td valign=top>");
out.println("Express how often you worried about non-academic issues today on average on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"all the time.\"<br>");
out.println("<input name=\"nonacademicworry\" type=\"text\" value=\"" + prevDay[18] + "\">");
out.println("</input>");
out.println("</td></tr>");

out.println("<tr><td valign=top>19.</td><td valign=top>");
out.println("Express how well you kept to your general diet today on ");
out.println("a scale from 0 to 100, where 0 is \"did not follow it at all\" and 100 is \"follow it
perfectly\"<br>");
out.println("<input name=\"diet\" type=\"text\" value=\"" + prevDay[19] + "\">");
out.println("</input>");
out.println("</td></tr>");

out.println("<tr><td valign=top>20.</td><td valign=top>");
out.println("Express how closely you watched your food intake today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"constantly.\"<br>");
out.println("<input name=\"watchintake\" type=\"text\" value=\"" + prevDay[20] + "\">");
out.println("</input>");
out.println("</td></tr>");

out.println("<tr><td valign=top>21.</td><td valign=top>");
out.println("Express how closely you counted calories today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"constantly.\"<br>");
out.println("<input name=\"caloriecount\" type=\"text\" value=\"" + prevDay[21] + "\">");
out.println("</input>");
out.println("</td></tr>");
out.println("</table>");

out.println("<input type=\"SUBMIT\" value=\"Submit\" align=\"middle\">");
out.println("</form>");
out.println("</td></tr></table>");
out.println("</body>");
out.println("</html>");
      }

if(userGroup == 2)
{
  /* create new vector for holding all 22 question vectors from db. */
  Vector vectorVector = new Vector();
  /* create variable for storing size of data. */
  int feedbackLength = 0;
  /* for each question ... */
  for(int i = 0; i < 22; i++)
        {
           /* get the vector that holds the array for question i.*/
           Vector v = QueryTable.GrpTwoFdback(userID,i+1);
           /* add it to the vector of all questions.*/
           vectorVector.add(v);
           /*calculate the size of the data. */
           int[] theData = (int[])(v.elementAt(0) );
           feedbackLength = theData.length;
        }
/* allocate the array of questions and values. */
int[][] feedback = new int[22][feedbackLength];
/* for each question ... */
for(int i = 0; i < 22; i++)
        {
           /* get the vector for question i. */
           Object tempObject = vectorVector.elementAt(i);
           /* cast it as a vector. */
           Vector tempVector = (Vector)tempObject;
           /* get the array from the vector and cast it as an int[]. */
           int[] fi = (int[])(tempVector.elementAt(0) );
           /* store it as question i. */
           feedback[i] = fi;
        }

        /* get a list of all days that have data entry for this user. */
        int[] dataMonths = feedback[0];
        int[] dataDays = feedback[1];

        /* figure out what days of the study they've been doing it for. */
        /* first make array of study day */
        int[] studyDay = new int[dataMonths.length];
        for(int i = 0; i < studyDay.length; i++)
            {
                   studyDay[i] = (dataMonths[i] - 3)*31 + dataDays[i];
            }
        /* get maximum and minimum. */
        int max = 0;
        /*int min =
        for(int i = 0; i < studyDay.length; i++)
            {
                   if(studyDay[i] > max)
                       {
                               max = studyDay[i];
                       }
                   if(studyDay[i] < min)
                       {
                               min = studyDay[i];
                       }
            }
        boolean[] allStudyDays = new boolean[max + 1];
        for(int i = 0; i < allStudyDays.length; i++)
            {
                   allStudyDays[i] = false;
                   }*/
        /* for each piece of data ... */
        /*for(int i = 0; i < studyDay.length; i++)
            {
                /* find out what day it is for and check off that day in the list. */
        /*allStudyDays[studyDay[i]] = true;
          }
          int[] finalData = new int[max-min + 1];*/
        /* for each possible date ... */
        /*for(int i = min; i < allStudyDays.length; i++)
          {*/
                /* check to see if it's */
        /*              } */

  String[] fileNameArray = new String[22];
  for(int i = 2; i < 22; i++)
        {
          //create graphs for each question and store as jpegs.
          Integer tempInteger = new Integer(i+1);
          String intString = tempInteger.toString();
          String fileName = "images/";
          fileName = fileName.concat(userName);
          fileName = fileName.concat(intString);
          fileName = fileName.concat(".jpg");
          if(feedback.length == 0)
                {
                  int[] tempArray = new int[1];
                  Grapher temp = new Grapher(tempArray,fileName);
                }
          else
                {
                  Grapher temp = new Grapher(feedback[i],fileName);
                }
          fileNameArray[i] = fileName;
        }

/* Graphs of previous entries */
out.println("<html>");
out.println("<head>");
out.println("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">");
out.println("<meta name=\"GENERATOR\" content=\"Mozilla/4.75 [en] (X11; U; IRIX 6.5 IP22)
[Netscape]\">");

out.println("<title>Health Tracker</title>");
out.println("</head>");
out.println("<body bgcolor=\"white\">");
out.println("<table border=0 height=1000>");
out.println("<tr valign=top>");
out.println("<td width=150
background=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?image=bg2.jpg\">");
out.println("</td>");
out.println("<td>");
out.println("<h1>Health Tracker</h1>");
out.println("<h2>Daily Questionnaire</h2>");

out.println("<form method=\"GET\" action=\"reviewServlet\">");
out.println("<input type=\"hidden\" name=\"username\" value=\"" + userName +"\">");
out.println("<input type=\"hidden\" name=\"password\" value=\"" + passWord +"\">");

out.println("<table border=0 cellpadding=40 cols=2 width=\"100%\">");
out.println("<tr valign=top><td>1. Please enter the date for which you are answering the questions.<p>");
GregorianCalendar myCalendar = new GregorianCalendar();
int month = myCalendar.get(GregorianCalendar.MONTH) + 1;
int date = myCalendar.get(GregorianCalendar.DATE);

out.println("Month <input type=\"text\" name=\"month\" value=\"" + month + "\">");
out.println("<p>Date <input type=\"text\" name=\"date\" value=\"" + date +
"\"><br></td><td></td></tr>");


out.println("<tr valign=top><td>2. How many hours of sleep did you have last night? (please round to the
nearest hour)<br>");
out.println("<input type=\"text\" name=\"sleep\" value=\"" + prevDay[2] + "\"></td>");
out.println("<td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" +
month + "&date=" + date + "&image=" + fileNameArray[2] + "\"></td>");
out.println("</tr>");
out.println("");

out.println("<tr valign=top><td>3. How many complete meals did you eat today?<br>");
out.println("<input type=\"text\" name=\"eat\" value=\"" + prevDay[3] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[3] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>4. Please express the amount of fatty foods in your diet today ");
out.println("on a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"fat\" type=\"text\" value=\"" + prevDay[4] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[4] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>5. Please express the relative amount of fruits and vegetables in your diet
today ");
out.println("on a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"fruit\" type=\"text\" value=\"" + prevDay[5] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[5] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>6. Please express the relative amount of meat in your diet today on ");
out.println("a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"meat\" type=\"text\" value=\"" + prevDay[6] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[6] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>7. How many <a name=\"caffeine\"></a><a href=\"#caffeine\"
onClick=\"window.open('http://ilab.media.mit.edu:8080/healthcare/servlet/getPage?page=caffeine.html','Ca
ffeine','height=400,width=400');\">units of caffeinated drinks</a> did you consume today?<br>");
out.println("<input type=\"text\" name=\"caffeine\" value=\"" + prevDay[7] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[7] + "\"><br>");
out.println("</td></tr>");
out.println("<tr valign=top><td>8. How many <a name=\"alcohol\"></a><a href=\"#alcohol\"
onClick=\"window.open('http://ilab.media.mit.edu:8080/healthcare/servlet/getPage?page=alcohol.html','Al
cohol','height=400,width=400');\">units of alcohol</a> did you consume today?<br>");
out.println("<input type=\"text\" name=\"alchoholamount\" value=\"" + prevDay[8] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[8] + "\"><br>");
out.println("</td></tr>");

out.println("<tr valign=top><td>9. How many minutes of exercise did you do today?<br>");
out.println("<input type=\"text\" name=\"exercise\" value=\"" + prevDay[9] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[9] + "\"><br>");
out.println("</td></tr>");

out.println("<tr valign=top><td>10. How many cigarettes did you smoke today?<br>");
out.println("<input type=\"text\" name=\"smokingamount\" value=\"" + prevDay[10] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[10] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>11. What is your pulse at rest?  (measured over a minute)<br>");
out.println("<input type=\"text\" name=\"pulse\" value=\"" + prevDay[11] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[11] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>12. What is your weight?  (in pounds)<br>");
out.println("<input type=\"text\" name=\"weight\" value=\"" + prevDay[12] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[12] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>13. Express how stressed you were today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all stressed\" and 100 is \"very stressed.\"<br>");
out.println("<input name=\"stress\" type=\"text\" value=\"" + prevDay[13] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[13] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>14. Express how happy you were today on a scale from 0 to 100, where 0
is \"not at all happy\" and 100 is \"very happy\"<br>");
out.println("<input name=\"happiness\" type=\"text\" value=\"" + prevDay[14] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[14] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>15. Express how healthy you were today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all healthy\" and 100 is \"very healthy.\"<br>");
out.println("<input name=\"health\" type=\"text\" value=\"" + prevDay[15] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[15] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>16. Express your homework status today on ");
out.println("a scale from 0 to 100, where 0 is \"very behind,\" 50 is \"on schedule,\" and 100 is \"ahead of
schedule.\"<br>");
out.println("<input name=\"homework\" type=\"text\" value=\"" + prevDay[16] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[16] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>17. Express how often you worried about academic issues today on
average on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"all the time.\"<br>");
out.println("<input name=\"academicworry\" type=\"text\" value=\"" + prevDay[17] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[17] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>18. Express how often you worried about non-academic issues today on
average on ");
out.println("a scale from 0 to 100, w where 0 is \"not at all\" and 100 is \"all the time.\"<br>");
out.println("<input name=\"nonacademicworry\" type=\"text\" value=\"" + prevDay[18] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[18] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>19. Express how well you kept to your general diet today on ");
out.println("a scale from 0 to 100, where 0 is \"did not follow it at all\" and 100 is \"follow it
perfectly\"<br>");
out.println("<input name=\"diet\" type=\"text\" value=\"" + prevDay[19] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[19] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>20. Express how closely you watched your food intake today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"constantly.\"<br>");
out.println("<input name=\"watchintake\" type=\"text\" value=\"" + prevDay[20] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[20] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>21. Express how closely you counted calories today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"constantly.\"<br>");
out.println("<input name=\"caloriecount\" type=\"text\" value=\"" + prevDay[21] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month="
+ month + "&date=" + date + "&image=" + fileNameArray[21] + "\">");
out.println("</td></tr>");
out.println("</table>");

out.println("<input type=\"SUBMIT\" value=\"Submit\" align=\"middle\">");
out.println("</form></td></tr></table>");
out.println("</body>");
out.println("</html>");
}

   if(userGroup == 3)
         {
  /* Graphs of previous entries plus average of all users */
            Vector vectorVector1 = new Vector();
         int feedbackLength1 = 0;
         for(int i = 0; i < 22; i++)
               {
                 Vector v3 = QueryTable.GrpThreeFdback(userID,i+1);
                 int[] theData1 = (int[])(v3.elementAt(1) );
                 vectorVector1.add(theData1);
                 feedbackLength1 = theData1.length;
               }
int[][] feedback1 = new int[22][feedbackLength1];
for(int i = 0; i < 22; i++)
      {
        Object tempObject = vectorVector1.elementAt(i);
        int[] tempArray = (int[])tempObject;
        feedback1[i] = tempArray;
      }

Vector vectorVector2 = new Vector();
int feedbackLength2 = 0;
for(int i = 0; i < 22; i++)
{
  Vector v3 = QueryTable.GrpThreeFdback(userID,i+1);
      int[] theData2 = (int[])(v3.elementAt(0) );
  vectorVector2.add(theData2);
  feedbackLength2 = theData2.length;
      System.out.println("GrpThree average data length : " + theData2.length);
}
int[][] feedback2 = new int[22][feedbackLength2];
for(int i = 0; i < 22; i++)
      {
        Object tempObject = vectorVector2.elementAt(i);
        int[] tempArray = (int[])tempObject;
        feedback2[i] = tempArray;
      }
String[] fileNameArray1 = new String[22];
for(int i = 2; i < 22; i++)
      {
        //create graphs of past user input with overlay of average of all users input.
        Integer tempInteger = new Integer(i+1);
        String intString = tempInteger.toString();
        String fileName = "images/";
        fileName = fileName.concat(userName);
        fileName = fileName.concat(intString);
        fileName = fileName.concat(".jpg");
        if(feedback1[i].length == 0)
              {
                int[] tempArray = new int[1];
                Grapher temp = new Grapher(tempArray,tempArray,fileName);
              }
        else
              {
                int[] exptFeedback = feedback2[i];
                if(feedback1[i].length > feedback2[i].length)
                      {
                        int[] paddedExperiment = new int[feedback1[i].length];
                        for(int k = 0; k < feedback2[i].length; k++)
                              {
                                  paddedExperiment[k] = feedback2[i][k];
                                }
                          for(int k = feedback2[i].length; k < feedback1[i].length; k++)
                                {
                                  paddedExperiment[k] = 0;
                                }
                          exptFeedback = paddedExperiment;
                        }
                  Grapher temp = new Grapher(feedback1[i], exptFeedback, fileName);
                }
          fileNameArray1[i] = fileName;
        }

out.println("<html>");
out.println("<head>");
out.println("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">");
out.println("<meta name=\"GENERATOR\" content=\"Mozilla/4.75 [en] (X11; U; IRIX 6.5 IP22)
[Netscape]\">");

out.println("<title>Health Tracker</title>");
out.println("</head>");
out.println("<body bgcolor=\"white\">");
out.println("<table border=0 height=1000>");
out.println("<tr valign=top>");
out.println("<td width=150
background=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?image=bg2.jpg\">");
out.println("</td>");
out.println("<td>");
out.println("<h1>Health Tracker</h1>");
out.println("<h2>Daily Questionnaire</h2>");

out.println("<form method=\"GET\" action=\"reviewServlet\">");
out.println("<input type=\"hidden\" name=\"username\" value=\"" + userName +"\">");
out.println("<input type=\"hidden\" name=\"password\" value=\"" + passWord +"\">");

out.println("<table cellpadding=40 border=0 cols=2 width=\"100%\">");

out.println("<tr valign=top>");
out.println("<td>1. Please enter the date for which you are answering the questions.<br>");
GregorianCalendar myCalendar = new GregorianCalendar();
int month = myCalendar.get(GregorianCalendar.MONTH) + 1;
int date = myCalendar.get(GregorianCalendar.DATE);
out.println("Month <input type=\"text\" name=\"month\" value=\"" + month + "\">");
out.println("<p>Date <input type=\"text\" name=\"date\" value=\"" + date + "\">");
out.println("</td>");
out.println("<td>");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>2. How many hours of sleep did you have last night? (please round to the nearest hour)");
out.println("<input type=\"text\" name=\"sleep\" value=\"" + prevDay[2] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[2] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>3. How many complete meals did you eat today?<br>");
out.println("<input type=\"text\" name=\"eat\" value=\"" + prevDay[3] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[3] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>4. Please express the amount of fatty foods in your diet today ");
out.println("on a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"fat\" type=\"text\" value=\"" + prevDay[4] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[4] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>5. Please express the relative amount of fruits and vegetables in your diet today ");
out.println("on a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"fruit\" type=\"text\" value=\"" + prevDay[5] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[5] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>6. Please express the relative amount of meat in your diet today on ");
out.println("a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"meat\" type=\"text\" value=\"" + prevDay[6] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[6] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>7. How many <a name=\"caffeine\"></a><a href=\"#caffeine\"
onClick=\"window.open('http://ilab.media.mit.edu:8080/healthcare/servlet/getPage?page=caffeine.html','Ca
ffeine','height=400,width=400');\">units of caffeinated drinks</a> did you consume today?<br>");
out.println("<input type=\"text\" name=\"caffeine\" value=\"" + prevDay[7] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[7] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>8. How many <a name=\"alcohol\"></a><a href=\"#alcohol\"
onClick=\"window.open('http://ilab.media.mit.edu:8080/healthcare/servlet/getPage?page=alcohol.html','Al
cohol','height=400,width=400');\">units of alcohol</a> did you consume today?<br>");
out.println("<input type=\"text\" name=\"alchoholamount\" value=\"" + prevDay[8] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[8] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>9. How many minutes of exercise did you do today?<br>");
out.println("<input type=\"text\" name=\"exercise\" value=\"" + prevDay[9] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[9] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>10. How many cigarettes did you smoke today?<br>");
out.println("<input type=\"text\" name=\"smokingamount\" value=\"" + prevDay[10] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[10] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>11. What is your pulse at rest?  (measured over a minute)<br>");
out.println("<input type=\"text\" name=\"pulse\" value=\"" + prevDay[11] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[11] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>12. What is your weight?  (in pounds)<br>");
out.println("<input type=\"text\" name=\"weight\" value=\"" + prevDay[12] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[12] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>13. Express how stressed you were today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all stressed\" and 100 is \"very stressed.\"<br>");
out.println("<input name=\"stress\" type=\"text\" value=\"" + prevDay[13] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[13] + "\">");
out.println("</td>");
out.println("</tr>");
out.println("<tr valign=top>");
out.println("<td>14. Express how happy you were today on a scale from 0 to 100, where 0 is \"not at all
happy\" and 100 is \"very happy\"<br>");
out.println("<input name=\"happiness\" type=\"text\" value=\"" + prevDay[14] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[14] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>15. Express how healthy you were today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all healthy\" and 100 is \"very healthy.\"<br>");
out.println("<input name=\"health\" type=\"text\" value=\"" + prevDay[15] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[15] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>16. Express your homework status today on ");
out.println("a scale from 0 to 100, where 0 is \"very behind,\" 50 is \"on schedule\" and 100 is \"ahead of
schedule.\"<br>");
out.println("<input name=\"homework\" type=\"text\" value=\"" + prevDay[16] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[16] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>17. Express how often you worried about academic issues today on average on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"all the time.\"<br>");
out.println("<input name=\"academicworry\" type=\"text\" value=\"" + prevDay[17] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[17] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>18. Express how often you worried about non-academic issues today on average on ");
out.println("a scale from 0 to 100, w where 0 is \"not at all\" and 100 is \"all the time.\"<br>");
out.println("<input name=\"nonacademicworry\" type=\"text\" value=\"" + prevDay[18] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[18] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>19. Express how well you kept to your general diet today on ");
out.println("a scale from 0 to 100, where 0 is \"did not follow it at all\" and 100 is \"follow it
perfectly\"<br>");
out.println("<input name=\"diet\" type=\"text\" value=\"" + prevDay[19] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[19] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>20. Express how closely you watched your food intake today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"constantly.\"<br>");
out.println("<input name=\"watchintake\" type=\"text\" value=\"" + prevDay[20] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[20] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("<tr valign=top>");
out.println("<td>21. Express how closely you counted calories today on ");
out.println("a scale from 0 to 100, where 0 is \"not at all\" and 100 is \"constantly.\"<br>");
out.println("<input name=\"caloriecount\" type=\"text\" value=\"" + prevDay[21] + "\">");
out.println("</td><td>");
out.println("<img src=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?month=" + month +
"&date=" + date + "&image=" + fileNameArray1[21] + "\">");
out.println("</td>");
out.println("</tr>");

out.println("</table>");
out.println("<input type=\"SUBMIT\" value=\"Submit\" align=\"middle\">");
out.println("</form></td></tr></table>");
out.println("</body>");
out.println("</html>");
           }

if(userGroup == 6)
{
   /* create new vector for holding all 22 question vectors from db. */
   Vector vectorVector = new Vector();
   /* create variable for storing size of data. */
   int feedbackLength = 0;
   /* for each question ... */
   for(int i = 0; i < 22; i++)
          {
             /* get the vector that holds the array for question i.*/
             Vector v = QueryTable.VisorFdback(userID,i+1);
             /* add it to the vector of all questions.*/
             vectorVector.add(v);
          }
   /* allocate the array of questions and values. */
   int[][] feedback = new int[22][5];
   /* for each question ... */
   for(int i = 0; i < 22; i++)
          {
             /* get the vector for question i. */
             Object tempObject = vectorVector.elementAt(i);
             /* cast it as a vector. */
             Vector tempVector = (Vector)tempObject;
           /* get the array from the vector and cast it as an int[]. */
           int[] fi = (int[])(tempVector.elementAt(0) );
           /* store it as question i. */
           feedback[i] = fi;
        }

        /* get a list of all days that have data entry for this user. */
        int[] dataMonths = feedback[0];
        int[] dataDays = feedback[1];

        /* figure out what days of the study they've been doing it for. */
        /* first make array of study day */
        int[] studyDay = new int[dataMonths.length];
        for(int i = 0; i < studyDay.length; i++)
           {
                studyDay[i] = (dataMonths[i] - 3)*31 + dataDays[i];
           }
        /* get maximum and minimum. */
        int max = 0;
        /*int min =
        for(int i = 0; i < studyDay.length; i++)
           {
                if(studyDay[i] > max)
                   {
                        max = studyDay[i];
                   }
                if(studyDay[i] < min)
                   {
                        min = studyDay[i];
                   }
           }
        boolean[] allStudyDays = new boolean[max + 1];
        for(int i = 0; i < allStudyDays.length; i++)
           {
                allStudyDays[i] = false;
                }*/
        /* for each piece of data ... */
        /*for(int i = 0; i < studyDay.length; i++)
           {
                /* find out what day it is for and check off that day in the list. */
        /*allStudyDays[studyDay[i]] = true;
           }
           int[] finalData = new int[max-min + 1];*/
        /* for each possible date ... */
        /*for(int i = min; i < allStudyDays.length; i++)
          {*/
                /* check to see if it's */
        /*         } */

String[] fileNameArray = new String[22];
for(int i = 2; i < 5; i++)
        {
           Integer tempInteger = new Integer(i+1);
           String intString = tempInteger.toString();
           String fileName = "";
           fileName = fileName.concat(userName);
```

```
            fileName = fileName.concat(intString);
            fileName = fileName.concat(".jpg");
            SmallGrapher temp = new SmallGrapher(feedback[i],fileName);
            fileNameArray[i] = fileName;
        }

/* Graphs of previous entries */
out.println("<html>");
out.println("<head>");
out.println("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">");
out.println("<meta name=\"GENERATOR\" content=\"Mozilla/4.75 [en] (X11; U; IRIX 6.5 IP22)
[Netscape]\">");

out.println("<title>Health Tracker</title>");
out.println("</head>");
out.println("<body bgcolor=\"white\">");
out.println("<table border=0 height=1000>");
out.println("<tr valign=top>");
out.println("<td width=150
background=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?image=bg2.jpg\">");
out.println("</td>");
out.println("<td>");
out.println("<h1>Health Tracker</h1>");
out.println("<h2>Daily Questionnaire</h2>");

out.println("<form method=\"GET\" action=\"wirelessForm2\">");
out.println("<input type=\"hidden\" name=\"username\" value=\"" + userName +"\">");
out.println("<input type=\"hidden\" name=\"password\" value=\"" + passWord +"\">");

out.println("<table border=0 cellpadding=40 cols=2 width=\"100%\">");
out.println("<tr valign=top><td>1. Please enter the date for which you are answering the questions.<p>");
GregorianCalendar myCalendar = new GregorianCalendar();
int month = myCalendar.get(GregorianCalendar.MONTH) + 1;
int date = myCalendar.get(GregorianCalendar.DATE);

out.println("Month <input type=\"text\" name=\"month\" value=\"" + month + "\"><p>");
out.println("<p>Date <input type=\"text\" name=\"date\" value=\"" + date +
"\"><br></td><td></td></tr>");

out.println("<tr valign=top><td>2. How many hours of sleep did you have last night? (please round to the
nearest hour)<br>");
out.println("<input type=\"text\" name=\"sleep\" value=\"" + prevDay[2] + "\"></td>");
out.println("<td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[2] + "\"></td>");
out.println("</tr>");
out.println("");

out.println("<tr valign=top><td>3. How many complete meals did you eat today?<br>");
out.println("<input type=\"text\" name=\"eat\" value=\"" + prevDay[3] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[3] + "\">");
out.println("</td></tr>");

out.println("<tr valign=top><td>4. Please express the amount of fatty foods in your diet today ");
out.println("on a scale from 0 to 100, where 0 is \"none,\" 50 is the \"recommended amount,\" and 100 is
\"twice as much as needed.\"<br>");
out.println("<input name=\"fat\" type=\"text\" value=\"" + prevDay[4] + "\">");
out.println("</td><td><img src=\"http://ilab.media.mit.edu/~healthcare/" +  fileNameArray[4] + "\">");
out.println("</td></tr>");
out.println("</table>");

out.println("<input type=\"SUBMIT\" value=\"Submit\" align=\"middle\">");
out.println("</form></td></tr></table>");
out.println("</body>");
out.println("</html>");
}
  }


  public void doPost(HttpServletRequest request,
            HttpServletResponse response)
    throws IOException, ServletException
  {
    doGet(request, response);
  }

}
```

Q.3 GetPage

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class getPage extends HttpServlet {

  public void doGet(HttpServletRequest request,
            HttpServletResponse response)
    throws IOException, ServletException
  {
    response.setContentType("text/html");
        String fileName = request.getParameter("page");
        PrintWriter out = response.getWriter();
    //PrintWriter out = response.getWriter();

        File fileToSend = new File(fileName);
        FileReader frin = new FileReader(fileToSend);
        int nread;
    while( ( nread = frin.read() ) >= 0 )
    {
              out.write(nread);
    }
    frin.close();
        out.close();
  }
}
```

Q.4 Grapher

```
//Take an array of integers as input and create a line graph.
import java.io.*;
import java.awt.image.*;
import com.sun.image.codec.jpeg.*;
import java.awt.*;

public class Grapher
{
  //Set size image parameters.
        int dayWidth = 17;
        int graphHeight  = 150;
        int bottomBuffer = 40;
        int sideBuffer = 60;
        int topBuffer = 20;

  //Create template image.
        static private BufferedImage   srcImage;
  //Create image to store graph.
        static private BufferedImage   dstImage;
  //input data.
        private int[] data;

  //Function for command line test.
        public static void main( String [] args )
        {
          //Create test array.
                int[] datum = new int[6];
                datum[0] = 20;
                datum[1] = 80;
                datum[2] = 70;
                datum[3] = 40;
                datum[4] = 63;
                datum[5] = 20;

                //Create second test array.
                int[] datum2 = new int[6];
                datum2[0] = 50;
                datum2[1] = 20;
                datum2[2] = 20;
                datum2[3] = 90;
                datum2[4] = 23;
                datum2[5] = 40;

                //Create a Grapher with the test array and filename from command line.
                Grapher g2 = new Grapher(datum,args[0]);
        }

  //Constructor.  Takes data array and file name.
        public Grapher(int[] data, String outputFile)
        {
            srcImage = null;
        dstImage = null;

            try
```

```
                    {
                            //Load template source.
                       srcImage = loadImage( "cz.jpg" );
                    }
                    catch ( IOException e )
                    {
                         System.err.println( "Could not load as JPEG"
                           );
                         System.exit( 1 );
             }

                         //Create image from template.
                    dstImage = createCompatibleDestImage( srcImage, null, data );
                    if ( dstImage == null )
                    {
                         System.err.println( "Equalization failed." );
                         System.exit( 1 );
             }

                         //Command line flag for debugging.
                         System.out.println("before");
                         //Get Graphics object to draw graph onto.
                         Graphics2D g2d = dstImage.createGraphics();
                         //Command line flag for debugging.
                         System.out.println("after");

                         //Draw image border.
                         g2d.fillRect(0,0,data.length*dayWidth + sideBuffer, graphHeight + bottomBuffer +
topBuffer);

                         g2d.setColor(Color.black);

                         //Get highest value from input data.
                         float max = (float)getMax(data);

                         float scale = graphHeight/max;

                         for(int i = 0; i < data.length - 1; i++)
                         {
                            //for each data point, connect it to the next one by drawing a line.
                                    g2d.setColor(Color.red);
                                    g2d.drawLine(     dayWidth*i + sideBuffer,
                                                      graphHeight - (int)(data[i]*scale) + topBuffer,
                                                      dayWidth*(i+1) + sideBuffer,
                                                      graphHeight - (int)(data[i+1]*scale) + topBuffer
                                                      );
                                    g2d.setColor(Color.black);
                         }
                         if(data.length == 1)
                         {
                            //if there is only one point just draw a dot.
                                    g2d.setColor(Color.red);
                                    g2d.fillOval(sideBuffer,
                                                      graphHeight - (int)(data[0]*scale) + topBuffer,
                                                      5,
                                                      5);
```

```
                        g2d.setColor(Color.black);
                }
                //draw vertical axis.
                g2d.drawLine(   sideBuffer,
                                (graphHeight + topBuffer ),
                                (data.length)*dayWidth + sideBuffer,
                                (graphHeight + topBuffer ));

                //draw horizontal axis.
                g2d.drawLine(   sideBuffer,
                                0,
                                sideBuffer,
                                (graphHeight + topBuffer ) );

                for(int i = 0; i < data.length; i++)
                {
                  //draw hash marks and number.
                        g2d.drawLine(dayWidth*i + sideBuffer, graphHeight + topBuffer, dayWidth*i
+ sideBuffer, graphHeight + 5 + topBuffer);
                        Integer day = new Integer(i+1);
                        String dayString = day.toString();
                        g2d.drawString(dayString, dayWidth*i + sideBuffer - 3 , graphHeight + 20 +
topBuffer);
                }

                int graphWidth = data.length * dayWidth;
                int halfGraphWidth = graphWidth/2;
                //label horizontal axis.
                g2d.drawString("Day", halfGraphWidth + sideBuffer - 22, graphHeight + 35 +
topBuffer);
                int yInterval = graphHeight/5;
                if(max < 5)
                {
                        yInterval = (int)(graphHeight/max);
                }
                for(int i = 0; i <= graphHeight; i += yInterval)
                {
                        Integer value = new Integer( (int)((graphHeight - i)/scale) );
                        String valueString = value.toString();
                        g2d.drawLine(sideBuffer - 5, i + topBuffer, sideBuffer, i + topBuffer );
                        g2d.drawString(valueString, sideBuffer - 20, i + 4 + topBuffer);
                }

                try
        {
                  //save the graph to a jpeg file.
                        saveImage(outputFile, dstImage );
        }
        catch ( IOException e )
        {
                System.err.println( "Could not save " +
                  dstImage +
                  " as JPEG"
                 );
                System.exit( 1 );
        }
        }

  //Constructor to include a graph overlay of a second array in green.
        public Grapher(int[] data, int[] average, String outputFile)
        {
            srcImage = null;
        dstImage = null;

                if(average.length == 0)
                  {
                        average = new int[1];
                  }

            try
            {
              srcImage = loadImage( "cz.jpg" );
            }
            catch ( IOException e )
            {
                System.err.println( "Could not load as JPEG"
                  );
                System.exit( 1 );
        }

            dstImage = createCompatibleDestImage( srcImage, null, data );
            if ( dstImage == null )
            {
                System.err.println( "Equalization failed." );
                System.exit( 1 );
        }


                Graphics2D g2d = dstImage.createGraphics();

                g2d.fillRect(0,0,data.length*dayWidth + sideBuffer, graphHeight + bottomBuffer +
topBuffer);

                g2d.setColor(Color.black);

                float maxUser = (float)getMax(data);
                float maxAverage = (float)getMax(average);
                float max = maxUser;
                if(maxAverage > maxUser)
                  {
                        max = maxAverage;
                  }


                float scale = graphHeight/max;

                for(int i = 0; i < data.length - 1; i++)
                {
                        g2d.setColor(Color.red);
                        g2d.drawLine(   dayWidth*i + sideBuffer,
                                        graphHeight - (int)(average[i]*scale) + topBuffer,
                                        dayWidth*(i+1) + sideBuffer,
                    graphHeight - (int)(average[i+1]*scale) + topBuffer
                    );
            g2d.setColor(Color.green);

            g2d.drawLine(   dayWidth*i + sideBuffer,
                    graphHeight - (int)(data[i]*scale) + topBuffer,
                    dayWidth*(i+1) + sideBuffer,
                    graphHeight - (int)(data[i+1]*scale) + topBuffer
                    );
            g2d.setColor(Color.black);
        }
        if(data.length == 1)
        {
            g2d.setColor(Color.red);
            g2d.fillOval(sideBuffer,
                    graphHeight - (int)(data[0]*scale) + topBuffer,
                    5,
                    5);
            g2d.setColor(Color.black);
        }
        /* Write color key. */
        g2d.setColor(Color.red);
        g2d.drawString("Avg.", 0,40);
        g2d.setColor(Color.green);
        g2d.drawString("You", 0, 10);
        g2d.setColor(Color.black);

        g2d.drawLine(   sideBuffer,
                (graphHeight + topBuffer ),
                (data.length)*dayWidth + sideBuffer,
                (graphHeight + topBuffer ));

        g2d.drawLine(   sideBuffer,
                0,
                sideBuffer,
                (graphHeight + topBuffer ) );

        for(int i = 0; i < data.length; i++)
        {
            g2d.drawLine(dayWidth*i + sideBuffer, graphHeight + topBuffer, dayWidth*i
+ sideBuffer, graphHeight + 5 + topBuffer);
            Integer day = new Integer(i+1);
            String dayString = day.toString();
            g2d.drawString(dayString, dayWidth*i + sideBuffer - 3 , graphHeight + 20 +
topBuffer);
        }

        int graphWidth = data.length * dayWidth;
        int halfGraphWidth = graphWidth/2;
        g2d.drawString("Day", halfGraphWidth + sideBuffer - 22, graphHeight + 35 +
topBuffer);
        int yInterval = graphHeight/5;
        if(max < 5)
        {
            yInterval = (int)(graphHeight/max);
        }
```

```
                for(int i = 0; i <= graphHeight; i += yInterval)
                {
                        Integer value = new Integer( (int)((graphHeight - i)/scale) );
                        String valueString = value.toString();
                        g2d.drawLine(sideBuffer - 5, i + topBuffer, sideBuffer, i + topBuffer );
                        g2d.drawString(valueString, sideBuffer - 20, i + 4 + topBuffer);
                }
                try
        {
                        saveImage(outputFile, dstImage );
        }
        catch ( IOException e )
        {
                System.err.println( "Could not save " +
                  dstImage +
                  " as JPEG"
                 );
                System.exit( 1 );
        }
        }

        private static BufferedImage loadImage( String fileName )
    throws IOException
        {
        InputStream in = new FileInputStream( fileName );
        JPEGImageDecoder decoder = JPEGCodec.createJPEGDecoder( in );
        return decoder.decodeAsBufferedImage();
        }

        private static void saveImage( String fileName, BufferedImage image )
        throws IOException
        {
                OutputStream  out = new FileOutputStream(fileName);
                JPEGImageEncoder encoder = JPEGCodec.createJPEGEncoder( out );
                encoder.encode(image);
        }

        public BufferedImage createCompatibleDestImage( BufferedImage src, ColorModel dstCM, int[]
dataArray )
        {
        BufferedImage image;
        if ( dstCM == null )
                dstCM = src.getColorModel();

        int w = (dataArray.length)*dayWidth + sideBuffer;
        int h = graphHeight + bottomBuffer + topBuffer;
        image = new BufferedImage( dstCM,
                    dstCM.createCompatibleWritableRaster( w, h ),
                    dstCM.isAlphaPremultiplied(), null );
        return image;
        }

  //find the maximum element of an array.
        public int getMax(int[] d)
        {
                int current = 0;
        for(int i = 0; i < d.length; i++)
        {
            if(d[i] > current)
            {
                current = d[i];
            }
        }
        return current;
    }
}
```

Q.5 SmallGrapher

```
//take a five element array of integers and create a line graph.
import java.io.*;
import java.awt.image.*;
import com.sun.image.codec.jpeg.*;
import java.awt.*;

public class SmallGrapher
{
  //set image parameters.
    int dayWidth = 17;
    int graphHeight  = 150;
    int bottomBuffer = 40;
    int sideBuffer = 60;
    int topBuffer = 20;

  //template image.
    static private BufferedImage   srcImage;
  //destination image.
    static private BufferedImage   dstImage;
  //input data array.
    private int[] data;

  //command line test function; takes a string, creates an image, and saves the image to the file.
    public static void main( String [] args )
    {
        int[] datum = new int[5];
        datum[0] = 20;
        datum[1] = 80;
        datum[2] = 70;
        datum[3] = 40;
        datum[4] = 63;

        int[] datum2 = new int[6];
        datum2[0] = 50;
        datum2[1] = 20;
        datum2[2] = 20;
        datum2[3] = 90;
        datum2[4] = 23;
        datum2[5] = 40;
        SmallGrapher g2 = new SmallGrapher(datum,args[0]);
    }

  //Constructor.  Takes a data array and saves resulting graph to the filename.
```

```
public SmallGrapher(int[] data, String outputFile)
{
    srcImage = null;
dstImage = null;
       //path to store image.
       String path = "../../../public_html/";
       outputFile = path.concat(outputFile);
       if(data.length < 5)
         {
              //if data array is less than 5, padd with zeros.
              int[] datum = new int[5];
              for(int i = 0; i < (data.length); i++)
                {
                     datum[i] = data[i];
                }
              for(int i = data.length; i < 5; i++)
                {
                     datum[i] = 0;
                }
              data = datum;
         }
    try
    {
         //load template.
      srcImage = loadImage( "t.jpg" );
    }
    catch ( IOException e )
    {
       System.err.println( "Could not load as JPEG"
        );
       System.exit( 1 );
}
       //create image from template.
    dstImage = createCompatibleDestImage( srcImage, null, data );
    if ( dstImage == null )
    {
       System.err.println( "Equalization failed." );
       System.exit( 1 );
}
       //flag for debugging.
       System.out.println("before");
       //get Graphics to draw graph on.
       Graphics2D g2d = dstImage.createGraphics();
       //flag for debugging.
       System.out.println("after");

       //draw graph outline.
       g2d.fillRect(0,0,30,30);

       g2d.setColor(Color.black);

       //get maximum array element.
       float max = (float)getMax(data);

       //scale graph height to size of largest data element.
       float scale = 30/max;
```

```
                for(int i = 0; i < 5 - 1; i++)
                {
                  //draw lines between each successive point.
                        g2d.drawLine(   6*i,
                                        30 - (int)(data[i]*scale),
                                        6*(i+1),
                                        30 - (int)(data[i+1]*scale)
                                        );
                }
                if(data.length == 1)
                {
                  //if only one data point, just draw a dot.
                        g2d.fillOval(0,
                                        30 - (int)(data[0]*scale),
                                        2,
                                        2);
                }
                try
        {
                  //save to file.
                        saveImage(outputFile, dstImage );
        }
        catch ( IOException e )
        {
                System.err.println( "Could not save " +
                  dstImage +
                  " as JPEG"
                 );
                System.exit( 1 );
        }
        }

        private static BufferedImage loadImage( String fileName )
    throws IOException
        {
        InputStream in = new FileInputStream( fileName );
        JPEGImageDecoder decoder = JPEGCodec.createJPEGDecoder( in );
        return decoder.decodeAsBufferedImage();
        }

        private static void saveImage( String fileName, BufferedImage image )
        throws IOException
        {
                OutputStream  out = new FileOutputStream(fileName);
                JPEGImageEncoder encoder = JPEGCodec.createJPEGEncoder( out );
                encoder.encode(image);
        }

        public BufferedImage createCompatibleDestImage( BufferedImage src, ColorModel dstCM, int[]
dataArray )
        {
        BufferedImage image;
        if ( dstCM == null )
                dstCM = src.getColorModel();
        int w = 30;
        int h = 30;
        image = new BufferedImage( dstCM,
                    dstCM.createCompatibleWritableRaster( w, h ),
                    dstCM.isAlphaPremultiplied(), null );
        return image;
        }

  //return maximum element of array.
        public int getMax(int[] d)
        {
                int current = 0;
                for(int i = 0; i < d.length; i++)
                {
                        if(d[i] > current)
                        {
                                current = d[i];
                        }
                }
                return current;
        }
}
```

Q.6 ImageServlet

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class imageServlet extends HttpServlet {

  public void doGet(HttpServletRequest request,
            HttpServletResponse response)
    throws IOException, ServletException
  {
    response.setContentType("image/jpeg");
        String fileName = request.getParameter("image");
        ServletOutputStream out = response.getOutputStream();
    //PrintWriter out = response.getWriter();

        File fileToSend = new File(fileName);
        FileInputStream frin = new FileInputStream(fileToSend);
        int nread;
    while( ( nread = frin.read() ) >= 0 )
    {
                out.write(nread);
    }
    frin.close();
        out.close();
  }
  public void doPost(HttpServletRequest request,
            HttpServletResponse response)
    throws IOException, ServletException
  {
      doGet(request, response);
   }
}
```

Q.7 Update Servlet

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import UpdateTable;
import QueryTable;

public class updateServlet  extends HttpServlet {

   public void doGet(HttpServletRequest request, HttpServletResponse response)
      throws IOException, ServletException
   {
      response.setContentType("text/html");

            String username = request.getParameter("username");
            String password = request.getParameter("password");
            PrintWriter out = response.getWriter();

            int[] userInfo = QueryTable.LoginCheck(username,password);
            int userID = userInfo[1];
            int userGroup = userInfo[0];
            if(userGroup == 0)
               {
                        out.println("<html><head><title>Incorrect Login</title><head>");
                        out.println("<body><h1><font color=\"red\">Incorrect Login</font>");
                        out.println("</body></html>");
               }
            else
               {
                        String[] input = new String[22];
              input[0] = request.getParameter("month");
              input[1] = request.getParameter("date");
              input[2] = request.getParameter("sleep");
              input[3] = request.getParameter("eat");
              input[4] = request.getParameter("fat");
              input[5] = request.getParameter("fruit");
              input[6] = request.getParameter("meat");
              input[7] = request.getParameter("caffeine");
              input[8] = request.getParameter("alchoholamount");
              input[9] = request.getParameter("exercise");
              input[10] = request.getParameter("smokingamount");
              input[11] = request.getParameter("pulse");
              input[12] = request.getParameter("weight");
              input[13] = request.getParameter("stress");
              input[14] = request.getParameter("happiness");
              input[15] = request.getParameter("health");
              input[16] = request.getParameter("homework");
              input[17] = request.getParameter("academicworry");
              input[18] = request.getParameter("nonacademicworry");
```

```
        input[19] = request.getParameter("diet");
        input[20] = request.getParameter("watchintake");
        input[21] = request.getParameter("caloriecount");

        String problem = "";
        int problemInt = 0;
        boolean badInput = false;
        int[] data = new int[22];
        for(int i = 0 ; i < 22; i++)
          {
                Integer tempInteger = new Integer(0);
                try
                  {
                        tempInteger = new Integer(input[i]);
                        data[i] = tempInteger.intValue();
                  }
                catch(NumberFormatException e)
                  {
                        problem = input[i];
                        problemInt = i;
                        badInput = true;
                  }
                data[i] = tempInteger.intValue();
          }
        if(badInput)
          {
                out.println("<html><title>Health Tracker</title><head></head>");
                out.println("<body
background=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?image=bg.jpg\"><h1>Health
Tracker</h1><font color=\"red\">Error: non-integer input</font><br>");
                out.println("You have entered non-integer input.  Please go back and round all data to
the nearest whole number.");
                out.println("<p>" + problem + "<p>" + problemInt );
                out.println("</body></html>");
          }
        else
          {

        UpdateTable.SendData(userID, data);

        out.println("<html><title>Health Tracker</title><head></head>");
        out.println("<body bgcolor=\"white\">");
out.println("<table border=0 height=1000>");
out.println("<tr valign=top>");
out.println("<td width=150
background=\"http://ilab.media.mit.edu:8080/healthcare/servlet/imageServlet?image=bg2.jpg\">");
out.println("</td>");
out.println("<td>");
        out.println("<h1>Health Tracker</h1><h2>Submission Successful</h2>Thank you, your
information has been submitted.</td></tr></table>");
       out.println("</body></html>");
          }
         }
```

```
  }
}
```

Q.8 QueryTable

```
import java.sql.*;
import java.util.*;

public class QueryTable {

  // Validates login and password.  Return values:
  // 0 = invalid, 1 = Group1, 2 = Group2, 3 = Group3
  // 4 = Group4, 5 = Group5, 6 = Group6
  //
  public static int [] LoginCheck(String username, String password) {

       int [] returnLogin = {0, 0} ;

       try {
         Class.forName("postgresql.Driver");
       } catch (Exception e) {
         System.err.println( e.getMessage() );
         System.exit(-1);
       }

       // db name: healthdesk     user login: erick
       //
       String url = "jdbc:postgresql://localhost/healthdesk";
       String user = "erick";
       String pass = "";

       try {

         // Open the connection.
         //
     Connection conn = DriverManager.getConnection(url, user, pass);

     // Create a statement object.
     //
     Statement stmt = conn.createStatement();

     // Make query
     //
     String sql = "SELECT * FROM users " +
          "WHERE login = '" + username + "'" ;

     ResultSet rs = stmt.executeQuery(sql);

     // Loop through the result set
     //
     while ( rs.next() ) {

          String db_pswd = rs.getString("pswd");
          db_pswd = db_pswd.trim();

          int db_user_group = rs.getInt("grp");
```

```
            int db_userid = rs.getInt("userid");

            if ( db_pswd.equals(password) ) {

                    returnLogin[0] = db_user_group;
                    returnLogin[1] = db_userid;
            }

    }

    // Close the statement and the connection.
    //
    stmt.close();
    conn.close();

        } catch (SQLException e) {
           System.err.println( e.getMessage() );
           System.exit(-1);
        }

        return returnLogin;
}

// Returns user's previous day responses
//
public static int [] PreviousDay(int m_userid) {

        int [] rtnPrevDay = new int [20];

        try {
           Class.forName("postgresql.Driver");
        } catch (Exception e) {
           System.err.println( e.getMessage() );
           System.exit(-1);
        }

        // db name: healthdesk      user login: erick
        //
        String url = "jdbc:postgresql://localhost/healthdesk";
        String user = "erick";
        String pass = "";

        try {

           // Open the connection.
           //
           Connection conn =
                  DriverManager.getConnection(url, user, pass);

           // Create a statement object.
           //
           Statement stmt = conn.createStatement();

           // Make query that will report experiment average
           //
           String sql = "SELECT * FROM desktop_data " +
                    "WHERE userid = '" + m_userid + "' " +
                    "ORDER BY quest1 DESC, quest2 DESC, " +
                    "serverday DESC, servertime DESC " +
                    "LIMIT 1" ;

            ResultSet rs = stmt.executeQuery(sql);

            // Loop through the result set
            //
            while ( rs.next() ) {

                    for(int i = 6; i < 26; i++) {

                        int array_index = i - 6;
                        rtnPrevDay[array_index] = rs.getInt(i);
                    }

            }

            // Close the statement and the connection.
            //
            stmt.close();
            conn.close();

        } catch (SQLException e) {
            System.err.println( e.getMessage() );
            System.exit(-1);
        }

        return rtnPrevDay;

}

public static Vector GrpTwoFdback(int userid, int quest_num) {

        Vector returnVector = new Vector() ;

        try {
            Class.forName("postgresql.Driver");
        } catch (Exception e) {
            System.err.println( e.getMessage() );
            System.exit(-1);
        }

        // db name: healthdesk     user login: erick
        //
        String url = "jdbc:postgresql://localhost/healthdesk";
        String user = "erick";
        String pass = "";

        try {

    Connection conn =
              DriverManager.getConnection(url, user, pass);

    Statement stmt = conn.createStatement();
    // Make query that will report personal history
    //
    String sql = "SELECT quest" + quest_num + " FROM desktop_data " +
            "WHERE userid = '" + userid + "' " +
            "ORDER BY quest1, quest2, serverday, servertime" ;

    ResultSet rs = stmt.executeQuery(sql);
        Vector v = new Vector();

    // Loop through the result set
    //
    while ( rs.next() ) {

       int return_data = rs.getInt(1);
       Integer i_return_data = new Integer(return_data);

       v.add(i_return_data);

    }

    int [] indivArray = new int[v.size()];
    for(int i = 0; i < v.size(); i++) {

       Object tempObject = v.elementAt(i);
       Integer tempInteger = (Integer)tempObject;
       indivArray[i] = tempInteger.intValue();
    }

        returnVector.add(indivArray);

    stmt.close();
    conn.close();

       } catch (SQLException e) {
          System.err.println( e.getMessage() );
          System.exit(-1);
       }

       return returnVector;

}

public static Vector GrpThreeFdback(int userid, int quest_num) {

       Vector returnVector = new Vector();

       try {
          Class.forName("postgresql.Driver");
       } catch (Exception e) {
          System.err.println( e.getMessage() );
          System.exit(-1);
       }

       // db name: healthdesk     user login: erick
       //
```

```
  String url = "jdbc:postgresql://localhost/healthdesk";
  String user = "erick";
  String pass = "";

  try {

Connection conn = DriverManager.getConnection(url, user, pass);

Statement stmt = conn.createStatement();

// Make query that will report experiment average
//
String sql_expt = "SELECT AVG(quest" + quest_num + ") " +
        "FROM desktop_data " +
        "GROUP BY quest1, quest2 " +
        "ORDER BY quest1, quest2" ;

ResultSet rs_expt = stmt.executeQuery(sql_expt);
    Vector v_expt = new Vector();

// Loop through the result set
//
while ( rs_expt.next() ) {

  int return_data = rs_expt.getInt(1);
  Integer i_return_data= new Integer(return_data);

  v_expt.add(i_return_data);
}

int[] exptArray = new int[v_expt.size()];
for(int i = 0; i < v_expt.size(); i++) {

  Object tempObject = v_expt.elementAt(i);
  Integer tempInteger = (Integer)tempObject;
  exptArray[i] = tempInteger.intValue();
}

    returnVector.add(exptArray);


// Make query that will report personal history
//
String sql_indiv = "SELECT quest" + quest_num + " " +
        "FROM desktop_data " +
        "WHERE userid = '" + userid + "' " +
        "ORDER BY quest1, quest2, serverday, servertime" ;

ResultSet rs_indiv = stmt.executeQuery(sql_indiv);
    Vector v_indiv = new Vector();

// Loop through the result set
//
    while ( rs_indiv.next() ) {

  int return_data = rs_indiv.getInt(1);
```

```
       Integer i_return_data= new Integer(return_data);

       v_indiv.add(i_return_data);
    }

    int[] indivArray = new int[v_indiv.size()];
    for(int i = 0; i < v_indiv.size(); i++) {

       Object tempObject = v_indiv.elementAt(i);
       Integer tempInteger = (Integer)tempObject;
       indivArray[i] = tempInteger.intValue();
    }

         returnVector.add(indivArray);

         // Close the statement and the connection.
    //
    stmt.close();
    conn.close();

       } catch (SQLException e) {
          System.err.println( e.getMessage() );
          System.exit(-1);
       }

       return returnVector;

}

public static Vector VisorFdback(int userid, int quest_num) {

       Vector returnVector = new Vector() ;

       try {
          Class.forName("postgresql.Driver");
       } catch (Exception e) {
          System.err.println( e.getMessage() );
          System.exit(-1);
       }

       // db name: healthdesk     user login: erick
       //
       String url = "jdbc:postgresql://localhost/healthdesk";
       String user = "erick";
       String pass = "";

       try {

          Connection conn = DriverManager.getConnection(url, user, pass);

          Statement stmt = conn.createStatement();

          // Make query that will report personal history
          //
          String sql = "SELECT quest" + quest_num + " FROM desktop_data " +
                  "WHERE userid = '" + userid + "' " +
              "ORDER BY quest1, quest2, serverday, servertime " +
              "LIMIT 5" ;

          ResultSet rs = stmt.executeQuery(sql);
          Vector v = new Vector();

          // Loop through the result set
          //
          while ( rs.next() ) {

              int return_data = rs.getInt(1);
              Integer i_return_data = new Integer(return_data);

              v.add(i_return_data);
          }

          int [] indivArray = new int[v.size()];
          for(int i = 0; i < v.size(); i++) {

              Object tempObject = v.elementAt(i);
              Integer tempInteger = (Integer)tempObject;
              indivArray[i] = tempInteger.intValue();
          }

          returnVector.add(indivArray);

          stmt.close();
          conn.close();

        } catch (SQLException e) {
          System.err.println( e.getMessage() );
          System.exit(-1);
        }

        return returnVector;

    }

    // Returns user's previous day responses
    //
    public static int MidtermCheck(int m_userid) {

        int MidtermCheck = 0;

        try {
          Class.forName("postgresql.Driver");
        } catch (Exception e) {
          System.err.println( e.getMessage() );
          System.exit(-1);
        }

        // db name: healthdesk     user login: erick
        //
        String url = "jdbc:postgresql://localhost/healthdesk";
        String user = "erick";
        String pass = "";
    try {

      // Open the connection.
      //
      Connection conn =
          DriverManager.getConnection(url, user, pass);

      // Create a statement object.
      //
      Statement stmt = conn.createStatement();

      // Make query
      //
      String sql = "SELECT userid FROM midterm_data " +
          "WHERE userid = '" + m_userid + "'" ;

      ResultSet rs = stmt.executeQuery(sql);

      // Loop through the result set
      //
      while ( rs.next() ) {

          int db_userid = rs.getInt("userid");

          if ( db_userid == m_userid )
            MidtermCheck = 1;

      }

      // Close the statement and the connection.
      //
      stmt.close();
      conn.close();

    } catch (SQLException e) {
      System.err.println( e.getMessage() );
      System.exit(-1);
    }
    return MidtermCheck;
}

// Returns user's previous day responses
//
public static int FinalCheck(int m_userid) {

    int FinalCheck = 0;

    try {
      Class.forName("postgresql.Driver");
    } catch (Exception e) {
      System.err.println( e.getMessage() );
      System.exit(-1);
    }

    // db name: healthdesk     user login: erick
```

```
          //
          String url = "jdbc:postgresql://localhost/healthdesk";
          String user = "erick";
          String pass = "";

          try {

             // Open the connection.
             //
             Connection conn = DriverManager.getConnection(url, user, pass);

             // Create a statement object.
             //
             Statement stmt = conn.createStatement();

             // Make query
             //
             String sql = "SELECT userid FROM final_data " +
                     "WHERE userid = '" + m_userid + "'" ;

             ResultSet rs = stmt.executeQuery(sql);

             // Loop through the result set
             //
             while ( rs.next() ) {

                     int db_userid = rs.getInt("userid");

                     if ( db_userid == m_userid )
                        FinalCheck = 1;
             }

             // Close the statement and the connection.
             //
             stmt.close();
             conn.close();

          } catch (SQLException e) {
             System.err.println( e.getMessage() );
             System.exit(-1);
          }
          return FinalCheck;
   }
}
```

Q.9 UpdateTable

```
import java.sql.*;

public class UpdateTable {

   public static void SendData(int userid, int[] questn) {

      try {
         Class.forName("postgresql.Driver");
      } catch (Exception e) {
```

```
  System.err.println( e.getMessage() );
  System.exit(-1);
}

// db name: healthdesk      user login: erick
String url = "jdbc:postgresql://localhost/healthdesk";
String user = "erick";
String pass = "";

try {

  // Open the connection.
  //
  Connection conn =
    DriverManager.getConnection(url, user, pass);

  // Create a statement object.
  //
  Statement stmt = conn.createStatement();


  // Send questionnaire results to DB via an array
  //  (no responses should be represented as a blank space)
  //
          System.out.println("debug flag 2");

  String sql = "INSERT INTO desktop_data  " +
          "(userid, quest1, quest2, quest3, quest4, quest5, " +
          "quest6, quest7, quest8, quest9, quest10, quest11, " +
          "quest12, quest13, quest14, quest15, quest16, " +
          "quest17, quest18, quest19, quest20, quest21, " +
          "quest22) VALUES ('" + userid + "', '" +
    questn[0] + "', '" + questn[1] + "', '" +
    questn[2] + "', '" + questn[3] + "', '" +
    questn[4] + "', '" + questn[5] + "', '" +
    questn[6] + "', '" + questn[7] + "', '" +
    questn[8]  + "', '" + questn[9] + "', '" +
    questn[10] + "', '" + questn[11] + "', '" +
    questn[12] + "', '" + questn[13] + "', '" +
    questn[14] + "', '" + questn[15] + "', '" +
    questn[16] + "', '" + questn[17] + "', '" +
    questn[18] + "', '" + questn[19] + "', '" +
    questn[20] + "', '" + questn[21] + "')" ;


  stmt.executeUpdate(sql);

  System.out.println(
           "The database has been updated.");

  // Close the statement and the connection.
  //
  stmt.close();
  conn.close();

} catch (SQLException e) {
    System.err.println( e.getMessage() );
    System.exit(-1);
  }
}

public static void SendMidterm(int userid, int[] questn) {

  try {
    Class.forName("postgresql.Driver");
  } catch (Exception e) {
    System.err.println( e.getMessage() );
    System.exit(-1);
  }

  // db name: healthdesk     user login: erick
  String url = "jdbc:postgresql://localhost/healthdesk";
  String user = "erick";
  String pass = "";

  try {

    // Open the connection.
    //
    Connection conn =
      DriverManager.getConnection(url, user, pass);

    // Create a statement object.
    //
    Statement stmt = conn.createStatement();


    // Send midterm results to DB via an array
    //  (no responses should be represented as a blank space)
    //
            System.out.println("debug flag 2");

    String sql = "INSERT INTO midterm_data  " +
            "VALUES ( '" +
      questn[0] + "', '" + questn[1] + "', '" +
      questn[2] + "', '" + questn[3] + "', '" +
      questn[4] + "', '" + questn[5] + "', '" +
      questn[6] + "', '" + questn[7] + "', '" +
      questn[8]  + "', '" + questn[9] + "', '" +
      questn[10] + "', '" + questn[11] + "', '" +
      questn[12] + "', '" + questn[13] + "', '" +
      questn[14] + "', '" + questn[15] + "', '" +
      questn[16] + "', '" + questn[17] + "', '" +
      questn[18] + "', '" + questn[19] + "', '" +

      questn[20] + "', '" + questn[21] + "', '" +
      questn[22] + "', '" + questn[23] + "', '" +
      questn[24] + "', '" + questn[25] + "', '" +
      questn[26] + "', '" + questn[27] + "', '" +
      questn[28] + "', '" + questn[29] + "', '" +
      questn[30] + "', '" + questn[31] + "', '" +
      questn[32] + "', '" + questn[33] + "', '" +
questn[34] + "', '" + questn[35] + "', '" +
questn[36] + "', '" + questn[37] + "', '" +
questn[38] + "', '" + questn[39] + "', '" +

questn[40] + "', '" + questn[41] + "', '" +
questn[42] + "', '" + questn[43] + "', '" +
questn[44] + "', '" + questn[45] + "', '" +
questn[46] + "', '" + questn[47] + "', '" +
questn[48] + "', '" + questn[49] + "', '" +
questn[50] + "', '" + questn[51] + "', '" +
questn[52] + "', '" + questn[53] + "', '" +
questn[54] + "', '" + questn[55] + "', '" +
questn[56] + "', '" + questn[57] + "', '" +
questn[58] + "', '" + questn[59] + "', '" +

questn[60] + "', '" + questn[61] + "', '" +
questn[62] + "', '" + questn[63] + "', '" +
questn[64] + "', '" + questn[65] + "', '" +
questn[66] + "', '" + questn[67] + "', '" +
questn[68] + "', '" + questn[69] + "', '" +
questn[70] + "', '" + questn[71] + "', '" +
questn[72] + "', '" + questn[73] + "', '" +
questn[74] + "', '" + questn[75] + "', '" +
questn[76] + "', '" + questn[77] + "', '" +
questn[78] + "', '" + questn[79] + "', '" +

questn[80] + "', '" + questn[81] + "', '" +
questn[82] + "', '" + questn[83] + "', '" +
questn[84] + "', '" + questn[85] + "', '" +
questn[86] + "', '" + questn[87] + "', '" +
questn[88] + "', '" + questn[89] + "', '" +
questn[90] + "', '" + questn[91] + "', '" +
questn[92] + "', '" + questn[93] + "', '" +
questn[94] + "', '" + questn[95] + "', '" +
questn[96] + "', '" + questn[97] + "', '" +
questn[98] + "', '" + questn[99] + "', '" +

questn[100] + "', '" + questn[101] + "', '" +
questn[102] + "', '" + questn[103] + "', '" +
questn[104] + "', '" + questn[105] + "', '" +
questn[106] + "', '" + questn[107] + "', '" +
questn[108] + "', '" + questn[109] + "', '" +
questn[110] + "', '" + questn[111] + "', '" +
questn[112] + "', '" + questn[113] + "', '" +
questn[114] + "', '" + questn[115] + "', '" +
questn[116] + "', '" + questn[117] + "', '" +
questn[118] + "', '" + questn[119] + "', '" +

questn[120] + "', '" + questn[121] + "', '" +
questn[122] + "', '" + questn[123] + "', '" +
questn[124] + "', '" + questn[125] + "', '" +
questn[126] + "', '" + questn[127] + "', '" +
questn[128] + "', '" + questn[129] + "', '" +
questn[130] + "', '" + questn[131] + "', '" +
questn[132] + "', '" + questn[133] + "', '" +
questn[134] + "', '" + questn[135] + "', '" +
        questn[136] + "', '" + questn[137] + "', '" +
        questn[138] + "', '" + questn[139] + "', '" +
              userid + "')" ;

      stmt.executeUpdate(sql);

      System.out.println(
                  "The database has been updated.");

      // Close the statement and the connection.
      //
      stmt.close();
      conn.close();

    } catch (SQLException e) {
      System.err.println( e.getMessage() );
      System.exit(-1);
    }
  }

  public static void SendFinal(int userid, int[] questn) {

    try {
      Class.forName("postgresql.Driver");
    } catch (Exception e) {
      System.err.println( e.getMessage() );
      System.exit(-1);
    }

    // db name: healthdesk      user login: erick
    String url = "jdbc:postgresql://localhost/healthdesk";
    String user = "erick";
    String pass = "";

    try {

      // Open the connection.
      //
      Connection conn =
        DriverManager.getConnection(url, user, pass);

      // Create a statement object.
      //
      Statement stmt = conn.createStatement();


      // Send final questionnaire results to DB via an array
      //  (no responses should be represented as a blank space)
      //
              System.out.println("debug flag 2");

      String sql = "INSERT INTO final_data  " +
              "VALUES ( '" + userid + "', '" +
        questn[0] + "', '" + questn[1] + "', '" +
        questn[2] + "', '" + questn[3] + "', '" +
        questn[4] + "', '" + questn[5] + "', '" +
            questn[6] + "', '" + questn[7] + "', '" +
            questn[8]  + "', '" + questn[9] + "', '" +
            questn[10] + "', '" + questn[11] + "', '" +
            questn[12] + "', '" + questn[13] + "')" ;

        stmt.executeUpdate(sql);

        System.out.println("The database has been updated.");

        // Close the statement and the connection.
        //
        stmt.close();
        conn.close();

      } catch (SQLException e) {
        System.err.println( e.getMessage() );
        System.exit(-1);
      }
  }
}
```

Q.10 Midpoint Questionnaire

```
//takes input from previous midPoint form page and updates database.
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import QueryTable;
import Query;
import Grapher;

public class midPoint extends HttpServlet {

  public void doGet(HttpServletRequest request,
              HttpServletResponse response)
    throws IOException, ServletException
  {
    response.setContentType("text/html");

    PrintWriter out = response.getWriter();
        out.println("<!doctype html public \"-//w3c//dtd html 4.0 transitional//en\">");
        //get parameters passed from previous page.
        String idString = request.getParameter("userID");
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        Integer idInt = new Integer(idString);
        System.out.println(idString);
        int userID = idInt.intValue();
        int input[] = new int[140];
        for(int i = 1; i <= 28; i++)
          {
                for(int j = 1; j <= 5; j++)
                  {
                        Integer rowInt = new Integer(i);
```

```
                    String row = rowInt.toString();
                    Integer colInt = new Integer(j);
                    String column = colInt.toString();
                    String box = row;
                    box = box.concat("-");
                    box = box.concat(column);
                    String value = request.getParameter(box);
                    if(value == null)
                      {
                      }
                    else
                    {
                      //load input array with form input.
                      input[ (j - 1) + ((i - 1)*5) ] = 1;
                    }
               }
         }
       for(int i = 0; i < 140; i++)
         {
             System.out.println(input[i]);
         }

       //load data into database.
       UpdateTable.SendMidterm(userID, input);
             out.println("<html>");
             out.println("<head>");
             out.println("<title>HealthTracker</title>");
             out.println("</head>");
             out.println("<body>");
             out.println("<h1>HealthTracker</h1>");
             out.println("Thank you, your information has been submitted.  Please continue on to the
daily survey.");
             out.println("<form
action=\"http://ilab.media.mit.edu:8080/healthcare/servlet/formServlet\" method=\"post\">");
             out.println("<input type=\"hidden\" name=\"username\" value=\"" + username + "\">");
             out.println("<input type=\"hidden\" name=\"password\" value=\"" + password + "\">");
             out.println("<input type=\"submit\" value=\"Continue\">");
          out.println("</form>");
             out.println("</body></html>");

  }
}
```

Q.11 FinalUpdate2

```
//Gets results from previous page and submits to database.
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import QueryTable;
import Query;
import Grapher;

public class finalUpdate2 extends HttpServlet {
```

```
public void doPost(HttpServletRequest request,
            HttpServletResponse response)
  throws IOException, ServletException
{
  response.setContentType("text/html");

  PrintWriter out = response.getWriter();
      out.println("<!doctype html public \"-//w3c//dtd html 4.0 transitional//en\">");
      //get parameters passed from previous page.
      String idString = request.getParameter("userID");
      String username = request.getParameter("username");
      String password = request.getParameter("password");
      String q11 = request.getParameter("1-1");
      String q21 = request.getParameter("2-1");
      String q22 = request.getParameter("2-2");
      String q31 = request.getParameter("3-1");
      String q32 = request.getParameter("3-2");
      String q41 = request.getParameter("4-1");
      String q42 = request.getParameter("4-2");
      String q51 = request.getParameter("5-1");
      String q52 = request.getParameter("5-2");
      String st1 = request.getParameter("st1");
      String st2 = request.getParameter("st2");
      String st3 = request.getParameter("st3");
      String st4 = request.getParameter("st4");
      String st5 = request.getParameter("st5");
      Integer idInt = new Integer(idString);
      int userID = idInt.intValue();
      int input[] = new int[14];
      System.out.println("flag");

      //create long integers from string inputs of time each page is accessed.
      Long l1 = new Long(st1);
      Long l2 = new Long(st2);
      Long l3 = new Long(st3);
      Long l4 = new Long(st4);
      Long l5 = new Long(st5);
      long t1 = l1.longValue();
      long t2 = l2.longValue();
      long t3 = l3.longValue();
      long t4 = l4.longValue();
      long t5 = l5.longValue();

      //create integers from long integers by normalizing all values.
      long s1 = t1-t1;
      long s2 = t2-t1;
      long s3 = t3-t1;
      long s4 = t4-t1;
      long s5 = t5-t1;
      int i1 = (int)s1;
      int i2 = (int)s2;
      int i3 = (int)s3;
      int i4 = (int)s4;
      int i5 = (int)s5;
```

```
    Integer int11;
    Integer int21;
    Integer int22;
    Integer int31;
    Integer int32;
    Integer int41;
    Integer int42;
    Integer int51;
    Integer int52;

    //create array of strings from form input.
    String[] qs = new String[9];
    qs[0] = q11;
    qs[1] = q21;
    qs[2] = q22;
    qs[3] = q31;
    qs[4] = q32;
    qs[5] = q41;
    qs[6] = q42;
    qs[7] = q51;
    qs[8] = q52;

    //create integers from string input.
    Integer[] ints = new Integer[9];
    for(int i = 0; i < 9; i++)
      {
            try
              {
                    ints[i] = new Integer(qs[i]);
              }
            catch(NumberFormatException e)
              {
                    ints[i] = new Integer(2222);
              }
      }
    int11 = ints[0];
int21 = ints[1];
    int22 = ints[2];
    int31 = ints[3];
    int32 = ints[4];
    int41 = ints[5];
    int42 = ints[6];
    int51 = ints[7];
    int52 = ints[8];

    int i11 = int11.intValue();
    int i21 = int21.intValue();
    int i22 = int22.intValue();
    int i31 = int31.intValue();
    int i32 = int32.intValue();
    int i41 = int41.intValue();
    int i42 = int42.intValue();
    int i51 = int51.intValue();
    int i52 = int52.intValue();
    //          input = {i11, i21, i31, i32, i41, i42, i51, i52, i1, i2, i3, i4, i5};
    input[0] = i11;
```

```
        input[1] = i21;
        input[2] = i22;
        input[3] = i31;
        input[4] = i32;
        input[5] = i41;
        input[6] = i42;
        input[7] = i51;
        input[8] = i52;
        input[9] = i1;
        input[10] = i2;
        input[11] = i3;
        input[12] = i4;
        input[13] = i5;

        //send data to database.
        UpdateTable.SendFinal(userID, input);

        //print out html page.
                out.println("<html>");
                out.println("<head>");
                out.println("<title>HealthTracker</title>");
                out.println("</head>");
                out.println("<body>");
                out.println("<h1>HealthTracker</h1>");
                out.println("Thank you, your information has been submitted.  Please continue on to the
daily survey.");
                out.println("<form
action=\"http://ilab.media.mit.edu:8080/healthcare/servlet/formServlet\" method=\"post\">");
                out.println("<input type=\"hidden\" name=\"username\" value=\"" + username + "\">");
                out.println("<input type=\"hidden\" name=\"password\" value=\"" + password + "\">");
                out.println("<input type=\"submit\" value=\"Continue\">");
            out.println("</form>");
                out.println("</body></html>");

  }
}
```

Q.12 Final Questionnaire

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import QueryTable;
import Query;
import Grapher;

public class final2 extends HttpServlet {

  public void doGet(HttpServletRequest request,
             HttpServletResponse response)
    throws IOException, ServletException
  {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
        out.println("<!doctype html public \"-//w3c//dtd html 4.0 transitional//en\">");
    String userName = request.getParameter("username");
    String passWord = request.getParameter("password");
    String idString = request.getParameter("userID");
        String q11 = request.getParameter("1-1");
        String st1 = request.getParameter("st1");
        Integer idInt = new Integer(idString);
        int userID = idInt.intValue();
        int input[] = new int[14];

        File fileToSend = new File("final2.html");
        FileReader frin = new FileReader(fileToSend);
        int nread;
    while( ( nread = frin.read() ) >= 0 )
    {
                out.write(nread);
    }
    frin.close();

        out.print("<input type=\"hidden\" name=\"userID\" value=\"" + userID + "\"><br>");
        out.print("<input type=\"hidden\" name=\"username\" value=\"" + userName + "\"><br>");
        out.print("<input type=\"hidden\" name=\"password\" value=\"" + passWord + "\"><br>");
        Date d = new Date();
        long msecs = d.getTime();
        Long msecsLong = new Long(msecs);
        String msecsString = msecsLong.toString();
        out.println("<input type=\"hidden\" name=\"st1\" value=\"" + st1 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"st2\" value=\"" + msecsString + "\"><br>");

        out.println("<input type=\"hidden\" name=\"1-1\" value=\"" + q11 + "\"><br>");

        File fileToSend2 = new File("final2-2.html");
        FileReader frin2 = new FileReader(fileToSend2);
        int nread2;
    while( ( nread2 = frin2.read() ) >= 0 )
    {
                out.write(nread2);
    }
    frin2.close();

  }

  public void doPost(HttpServletRequest request,
            HttpServletResponse response)
    throws IOException, ServletException
  {
        doGet(request, response);
  }
}
```

Q.13 Final Questionnaire – Page Three

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import QueryTable;
import Query;
import Grapher;

public class final3 extends HttpServlet {

  public void doGet(HttpServletRequest request,
              HttpServletResponse response)
    throws IOException, ServletException
  {
    response.setContentType("text/html");

    PrintWriter out = response.getWriter();
        out.println("<!doctype html public \"-//w3c//dtd html 4.0 transitional//en\">");
    String userName = request.getParameter("username");
    String passWord = request.getParameter("password");
    String idString = request.getParameter("userID");
        String q11 = request.getParameter("1-1");
        String q21 = request.getParameter("2-1");
        String q22 = request.getParameter("2-2");
        String st1 = request.getParameter("st1");
        String st2 = request.getParameter("st2");
        Integer idInt = new Integer(idString);
        int userID = idInt.intValue();
        int input[] = new int[14];

        File fileToSend = new File("final3.html");
        FileReader frin = new FileReader(fileToSend);
        int nread;
    while( ( nread = frin.read() ) >= 0 )
    {
                out.write(nread);
    }
    frin.close();

        out.print("<input type=\"hidden\" name=\"userID\" value=\"" + userID + "\"><br>");
        out.print("<input type=\"hidden\" name=\"username\" value=\"" + userName + "\"><br>");
        out.print("<input type=\"hidden\" name=\"password\" value=\"" + passWord + "\"><br>");
        Date d = new Date();
        long msecs = d.getTime();
        Long msecsLong = new Long(msecs);
        String msecsString = msecsLong.toString();
        out.println("<input type=\"hidden\" name=\"st1\" value=\"" + st1 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"st2\" value=\"" + st2 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"st3\" value=\"" + msecsString + "\"><br>");

        out.println("<input type=\"hidden\" name=\"1-1\" value=\"" + q11 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"2-1\" value=\"" + q21 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"2-2\" value=\"" + q22 + "\"><br>");
        File fileToSend2 = new File("final3-2.html");
        FileReader frin2 = new FileReader(fileToSend2);
        int nread2;
     while( ( nread2 = frin2.read() ) >= 0 )
     {
                  out.write(nread2);
     }
     frin2.close();

  }

  public void doPost(HttpServletRequest request,
               HttpServletResponse response)
     throws IOException, ServletException
  {
        doGet(request, response);
  }
}
```

Q.14 Final Questionnaire – Page Four

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import QueryTable;
import Query;
import Grapher;

public class final4 extends HttpServlet {

  public void doGet(HttpServletRequest request,
               HttpServletResponse response)
     throws IOException, ServletException
  {
     response.setContentType("text/html");

     PrintWriter out = response.getWriter();
        out.println("<!doctype html public \"-//w3c//dtd html 4.0 transitional//en\">");
     String userName = request.getParameter("username");
     String passWord = request.getParameter("password");
        String idString = request.getParameter("userID");
        String q11 = request.getParameter("1-1");
        String q21 = request.getParameter("2-1");
        String q22 = request.getParameter("2-2");
        String q31 = request.getParameter("3-1");
        String q32 = request.getParameter("3-2");
        String st1 = request.getParameter("st1");
        String st2 = request.getParameter("st2");
        String st3 = request.getParameter("st3");
        String st4 = request.getParameter("st4");
        String st5 = request.getParameter("st5");
        Integer idInt = new Integer(idString);
        int userID = idInt.intValue();
        int input[] = new int[14];

        File fileToSend = new File("final4.html");
        FileReader frin = new FileReader(fileToSend);
        int nread;
    while( ( nread = frin.read() ) >= 0 )
    {
                out.write(nread);
    }
    frin.close();

        out.print("<input type=\"hidden\" name=\"userID\" value=\"" + userID + "\"><br>");
        out.print("<input type=\"hidden\" name=\"username\" value=\"" + userName + "\"><br>");
        out.print("<input type=\"hidden\" name=\"password\" value=\"" + passWord + "\"><br>");
        Date d = new Date();
        long msecs = d.getTime();
        Long msecsLong = new Long(msecs);
        String msecsString = msecsLong.toString();
        out.println("<input type=\"hidden\" name=\"st1\" value=\"" + st1 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"st2\" value=\"" + st2 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"st3\" value=\"" + st3 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"st4\" value=\"" + msecsString + "\"><br>");

        out.println("<input type=\"hidden\" name=\"1-1\" value=\"" + q11 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"2-1\" value=\"" + q21 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"2-2\" value=\"" + q22 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"3-1\" value=\"" + q31 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"3-2\" value=\"" + q32 + "\"><br>");

        File fileToSend2 = new File("final4-2.html");
        FileReader frin2 = new FileReader(fileToSend2);
        int nread2;
    while( ( nread2 = frin2.read() ) >= 0 )
    {
                out.write(nread2);
    }
    frin2.close();
  }

  public void doPost(HttpServletRequest request,
              HttpServletResponse response)
    throws IOException, ServletException
  {
        doGet(request, response);
  }
}
```

Q.15 Final Questionnaire – Page Five

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import QueryTable;
import Query;
import Grapher;

public class final5 extends HttpServlet {

  public void doGet(HttpServletRequest request,
           HttpServletResponse response)
    throws IOException, ServletException
  {
    response.setContentType("text/html");

    PrintWriter out = response.getWriter();
       out.println("<!doctype html public \"-//w3c//dtd html 4.0 transitional//en\">");
    String userName = request.getParameter("username");
    String passWord = request.getParameter("password");
       String idString = request.getParameter("userID");
       String q11 = request.getParameter("1-1");
       String q21 = request.getParameter("2-1");
       String q22 = request.getParameter("2-2");
       String q31 = request.getParameter("3-1");
       String q32 = request.getParameter("3-2");
       String q41 = request.getParameter("4-1");
       String q42 = request.getParameter("4-2");
       String st1 = request.getParameter("st1");
       String st2 = request.getParameter("st2");
       String st3 = request.getParameter("st3");
       String st4 = request.getParameter("st4");
       Integer idInt = new Integer(idString);
       int userID = idInt.intValue();
       int input[] = new int[14];

       File fileToSend = new File("final5.html");
       FileReader frin = new FileReader(fileToSend);
       int nread;
    while( ( nread = frin.read() ) >= 0 )
    {
              out.write(nread);
    }
    frin.close();

       out.print("<input type=\"hidden\" name=\"userID\" value=\"" + userID + "\"><br>");
       out.print("<input type=\"hidden\" name=\"username\" value=\"" + userName + "\"><br>");
       out.print("<input type=\"hidden\" name=\"password\" value=\"" + passWord + "\"><br>");
       Date d = new Date();
       long msecs = d.getTime();
       Long msecsLong = new Long(msecs);
       String msecsString = msecsLong.toString();
       out.println("<input type=\"hidden\" name=\"st1\" value=\"" + st1 + "\"><br>");
       out.println("<input type=\"hidden\" name=\"st2\" value=\"" + st2 + "\"><br>");
       out.println("<input type=\"hidden\" name=\"st3\" value=\"" + st3 + "\"><br>");
       out.println("<input type=\"hidden\" name=\"st4\" value=\"" + st4 + "\"><br>");
       out.println("<input type=\"hidden\" name=\"st5\" value=\"" + msecsString + "\"><br>");

       out.println("<input type=\"hidden\" name=\"1-1\" value=\"" + q11 + "\"><br>");
       out.println("<input type=\"hidden\" name=\"2-1\" value=\"" + q21 + "\"><br>");
       out.println("<input type=\"hidden\" name=\"2-2\" value=\"" + q22 + "\"><br>");
```

```
        out.println("<input type=\"hidden\" name=\"3-1\" value=\"" + q31 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"3-2\" value=\"" + q32 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"4-1\" value=\"" + q41 + "\"><br>");
        out.println("<input type=\"hidden\" name=\"4-2\" value=\"" + q42 + "\"><br>");

        File fileToSend2 = new File("final5-2.html");
        FileReader frin2 = new FileReader(fileToSend2);
        int nread2;
    while( ( nread2 = frin2.read() ) >= 0 )
    {
                out.write(nread2);
    }
    frin2.close();

  }

  public void doPost(HttpServletRequest request,
            HttpServletResponse response)
    throws IOException, ServletException
  {
        doGet(request, response);
  }
}
```

Appendix R: Results of the Orientation Questionnaire

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
Desktop – No Feedback															
Mean	6.5	2.5	51.5	41.2	45.4	2.2	1.4	54.6	25.5	0.0	49.6	66.9	67.7	21.6	0.7
Standard Deviation	2.1	1.0	19.9	23.0	21.9	3.3	1.9	26.8	26.6	0.0	22.9	24.7	23.4	20.0	0.6
Desktop – Individual Feedback															
Mean	6.1	2.2	53.1	33.5	48.1	0.5	1.1	50.9	29.2	0.0	45.8	65.4	64.6	20.5	1.8
Standard Deviation	2.1	0.8	20.2	15.9	32.3	0.7	1.5	21.0	30.7	0.0	27.3	28.3	25.4	22.2	4.0
Desktop – Individual and Group Feedback															
Mean	6.7	2.4	61.2	38.5	56.5	1.1	1.7	51.9	28.5	0.0	62.6	75.4	64.8	35.8	1.0
Standard Deviation	1.3	0.5	11.9	13.6	14.1	1.3	3.8	23.3	22.3	0.0	22.8	14.8	16.5	25.3	1.0
Wireless – No Feedback with Limitations on Use															
Mean	6.2	2.2	51.5	44.5	38.0	2.0	0.4	48.0	38.5	0.1	58.2	66.0	64.0	34.0	0.9
Standard Deviation	0.9	0.4	12.9	17.9	22.4	1.9	0.5	16.2	30.7	0.3	15.9	22.9	16.8	9.9	0.7
Wireless – No Feedback															
Mean	6.2	2.2	57.5	37.5	48.0	1.6	1.8	47.0	24.0	0.0	54.0	62.5	58.0	18.5	1.0
Standard Deviation	0.6	0.4	16.9	20.4	23.0	0.8	2.2	20.2	35.3	0.0	18.4	21.1	21.5	11.8	0.5
Wireless – Individual Feedback															
Mean	6.7	2.2	66.5	44.0	61.0	1.2	0.8	37.5	18.5	0.0	61.5	62.1	49.5	25.0	0.8
Standard Deviation	1.1	0.8	24.4	24.1	19.0	1.2	1.3	32.3	19.3	0.0	29.9	24.4	31.7	26.7	1.0

Results of the Orientation Questionnaire (continued)

	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29
Desktop – No Feedback														
Mean	5.4	43.5	45.4	50.4	42%	8%	50%	0%	78.1	61.5	56.2	71.2	92%	83%
Standard Deviation	7.8	27.1	25.9	33.9					27.7	32.0	39.4	33.4		
Desktop – Individual Feedback														
Mean	10.4	28.1	37.3	41.9	42%	17%	58%	0%	75.8	35.0	55.0	70.0	50%	42%
Standard Deviation	21.6	15.6	26.2	24.4					28.8	38.6	31.6	33.7		
Desktop – Individual and Group Feedback														
Mean	19.2	40.0	39.2	46.2	77%	23%	69%	8%	80.8	66.9	57.7	69.2	77%	46%
Standard Deviation	27.0	13.7	29.0	26.6					24.2	42.3	32.2	30.7		
Wireless – No Feedback with Limitations on Use														
Mean	20.0	41.8	44.5	47.0	30%	10%	30%	10%	76.0	49.0	68.0	68.0	50%	50%
Standard Deviation	29.5	14.5	29.0	23.2					28.0	43.3	38.5	26.8		
Wireless – No Feedback														
Mean	7.7	38.6	46.5	39.0	40%	10%	30%	0%	74.0	42.0	67.5	72.0	70%	60%
Standard Deviation	18.7	17.6	26.9	24.8					20.1	36.7	22.9	28.0		
Wireless – Individual Feedback														
Mean	22.5	48.0	43.5	38.2	40%	20%	50%	10%	81.5	59.8	67.0	73.5	70%	40%
Standard Deviation	34.1	22.6	35.4	28.7					18.1	36.6	34.3	20.0		

Note: Percentages listed in this table represent the fraction of subjects who responded positively to a particular yes/no question.

Appendix S: Results of the Debrief Questionnaire

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14
Desktop – No Feedback														
Mean	47.0	41.5	29.7	22.0	52.2	32.0	33.3	48.0	18.5	53.0	43.8	60.5	47.5	64.7
Standard Deviation	27.8	18.7	35.4	34.2	15.9	34.7	29.4	25.2	28.4	25.8	15.3	15.9	25.5	17.2
Desktop – Individual Feedback														
Mean	51.8	42.0	28.3	10.0	35.9	21.5	39.5	41.1	29.5	46.1	48.7	54.6	48.6	60.8
Standard Deviation	26.0	23.2	23.3	20.0	24.7	27.1	29.7	28.0	23.0	15.4	14.2	15.4	3.2	8.3
Desktop – Individual and Group Feedback														
Mean	53.6	50.9	44.5	5.5	47.3	30.0	47.7	49.5	35.5	49.1	40.5	57.7	44.5	56.1
Standard Deviation	15.0	12.2	29.2	9.6	11.9	21.4	29.2	19.8	26.6	7.0	9.3	11.0	15.1	17.9
Wireless – No Feedback with Limitations on Use														
Mean	49.6	45.1	30.6	11.0	60.0	27.2	33.7	42.6	23.6	42.5	53.0	58.0	59.0	61.0
Standard Deviation	29.9	30.0	27.6	20.8	25.3	29.4	24.5	24.4	24.8	23.7	19.5	16.2	16.6	16.1
Wireless – No Feedback														
Mean	47.8	33.3	41.1	18.9	38.9	14.4	34.4	39.4	32.2	52.8	43.3	58.9	52.8	59.4
Standard Deviation	21.1	22.9	19.6	28.9	26.2	15.1	20.1	18.8	17.9	4.4	9.0	9.3	6.7	7.3
Wireless – Individual Feedback														
Mean	40.9	37.9	31.0	10.0	41.4	29.4	40.4	39.4	27.9	50.0	45.0	55.5	48.0	52.5
Standard Deviation	28.9	26.4	22.8	26.5	36.6	28.8	27.8	25.8	33.1	0.0	7.1	7.6	6.3	10.3

Results of the Debrief Questionnaire (continued)

	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27
Desktop – No Feedback													
Mean	51.0	49.0	59.7	60.1	50.0	50.0	53.5	61.3	55.0	43.0	41.5	80.0	67.0
Standard Deviation	5.7	3.2	15.6	15.7	0.0	12.5	6.7	17.6	16.5	16.4	17.3	16.8	21.1
Desktop – Individual Feedback													
Mean	51.8	50.6	55.3	59.5	50.0	48.9	49.3	56.7	53.2	51.8	49.5	79.0	46.5
Standard Deviation	9.8	2.1	17.7	7.1	0.0	5.5	3.1	10.1	5.1	6.8	1.5	21.2	38.0
Desktop – Individual and Group Feedback													
Mean	51.5	50.0	60.9	59.5	50.0	49.2	50.5	61.8	52.7	51.8	50.0	84.1	61.4
Standard Deviation	8.8	0.0	8.3	9.3	0.0	8.0	1.5	8.4	9.3	9.8	4.5	14.6	40.3
Wireless – No Feedback with Limitations on Use													
Mean	50.0	50.0	63.0	63.8	50.0	50.0	53.5	57.1	50.0	46.5	52.1	75.7	50.5
Standard Deviation	0.0	0.0	16.2	15.9	0.0	10.3	8.2	10.5	0.0	9.4	6.3	27.9	40.3
Wireless – No Feedback													
Mean	49.4	50.0	55.6	55.6	50.0	50.6	50.0	57.2	52.2	50.6	50.6	71.1	35.6
Standard Deviation	1.7	0.0	5.8	7.3	0.0	6.3	0.0	5.7	4.4	1.7	1.7	11.4	32.5
Wireless – Individual Feedback													
Mean	48.0	48.9	58.0	55.5	50.0	49.0	50.0	54.5	54.0	53.0	52.0	77.3	63.0
Standard Deviation	7.9	7.8	9.5	8.3	0.0	5.7	4.7	6.0	10.7	6.7	4.2	19.7	31.6

Results of the Debrief Questionnaire (continued)

Table 1

	Q28	Q29	Q30	Q31	Q32	Q33	Q34	Q35
Desktop – No Feedback								
Mean	52.0	67.5	100%	60%	40%	20%	50%	20%
Standard Deviation	29.1	28.4						
Desktop – Individual Feedback								
Mean	63.6	73.6	82%	45%	18%	18%	73%	0%
Standard Deviation	26.1	26.3						
Desktop – Individual and Group Feedback								
Mean	63.6	82.7	82%	18%	64%	18%	55%	9%
Standard Deviation	24.1	14.9						
Wireless – No Feedback with Limitations on Use								
Mean	56.5	78.0	90%	60%	30%	10%	30%	20%
Standard Deviation	34.0	25.2						
Wireless – No Feedback								
Mean	56.7	57.8	78%	67%	44%	0%	56%	0%
Standard Deviation	21.2	24.9						
Wireless – Individual Feedback								
Mean	55.3	70.5	100%	70%	20%	20%	50%	10%
Standard Deviation	39.3	23.6						

Note: Percentages listed in this table represent the fraction of subjects who responded positively to a particular yes/no question.

Appendix T: Additional Results From the Daily Questionnaire

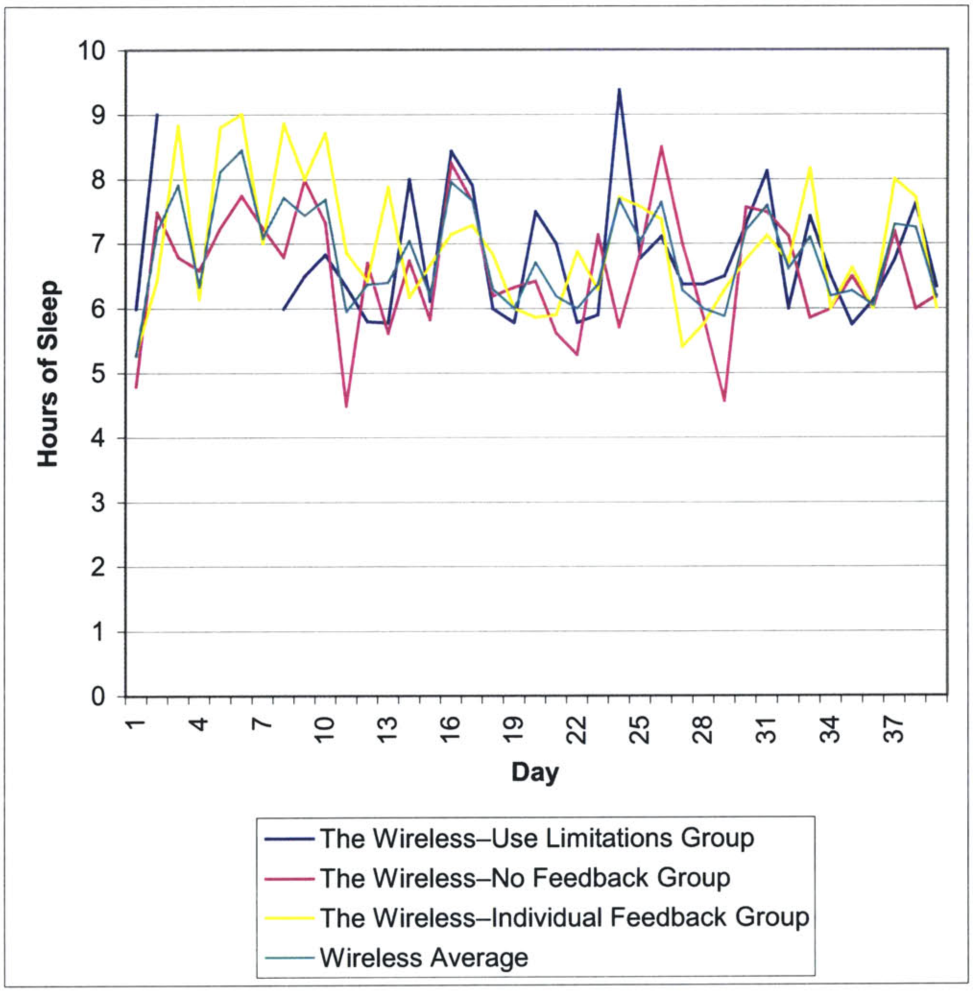


Exhibit 55: Wireless Groups' Sleeping Habits

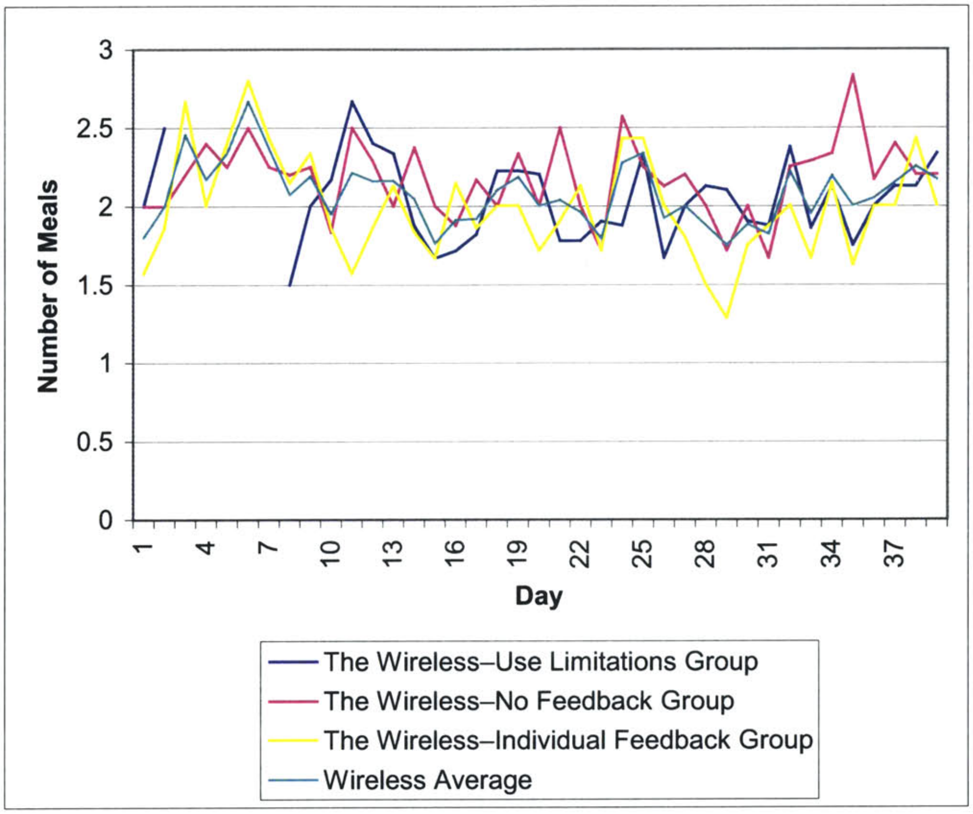


Exhibit 56: Number of Complete Meals Eaten Per Day By Wireless Users

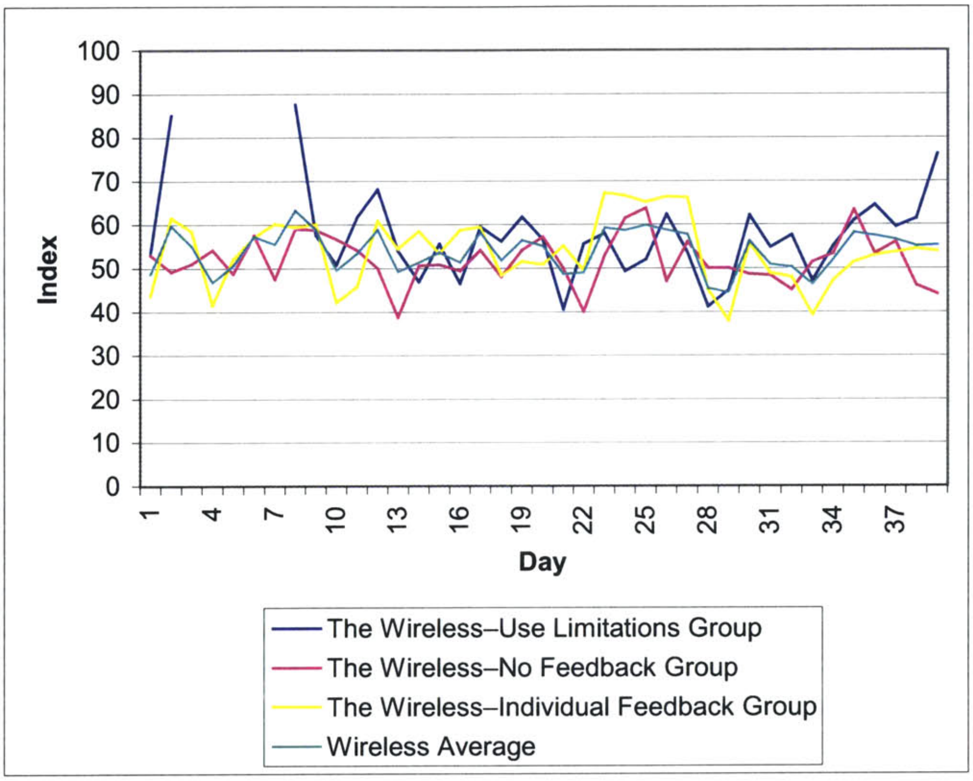


Exhibit 57: Wireless Users' Consumption of Fatty Foods

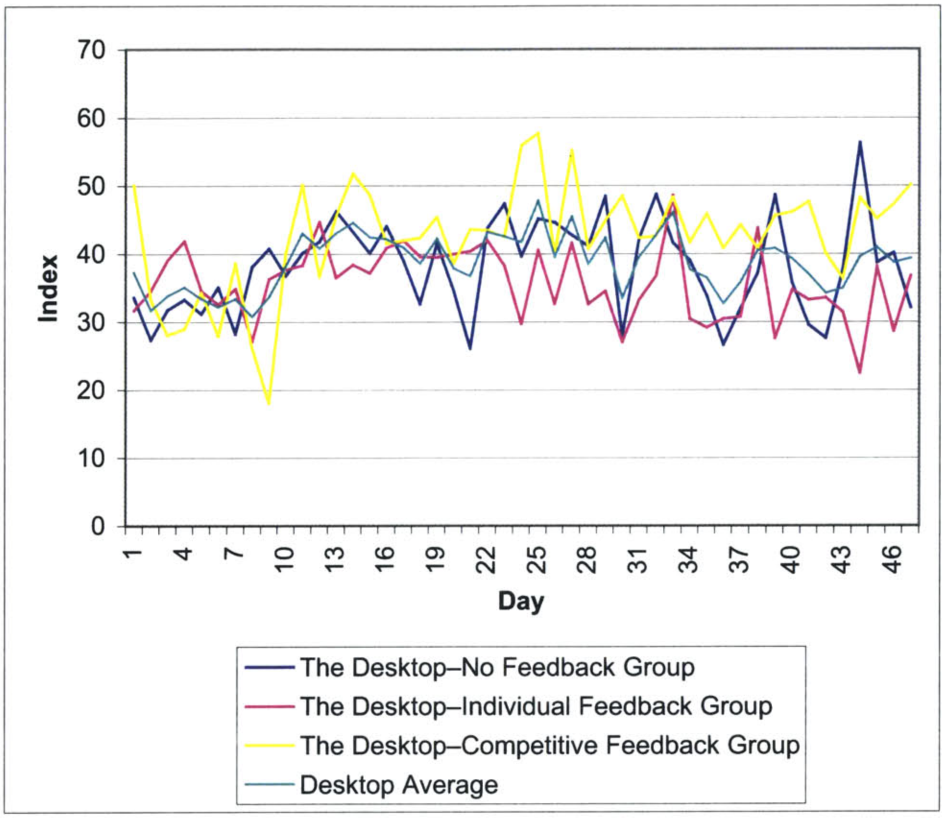


Exhibit 58: Desktop Users' Consumption of Meat

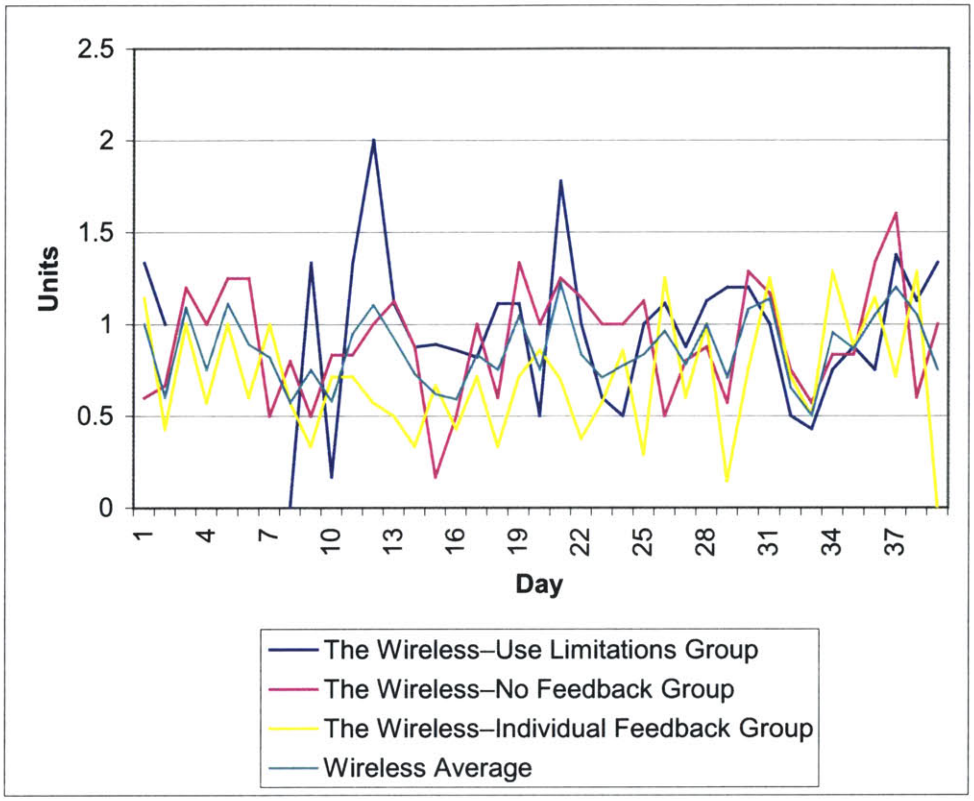


Exhibit 59: Wireless Users' Consumption of Caffeine

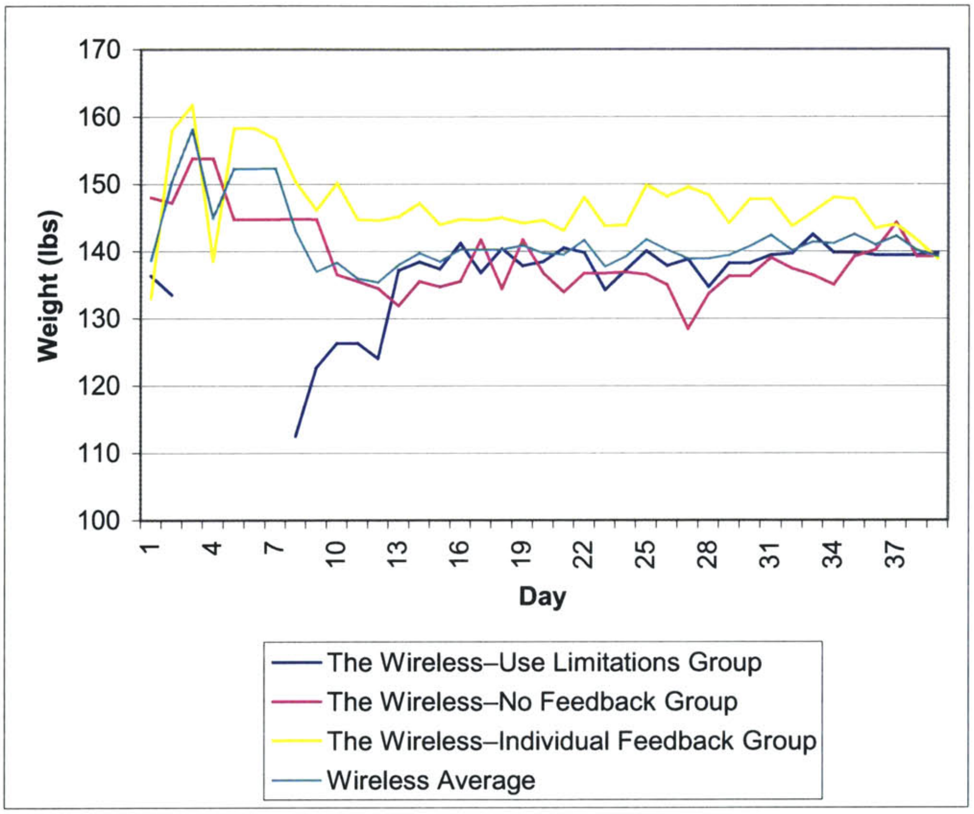


Exhibit 60: Wireless Users' Weight

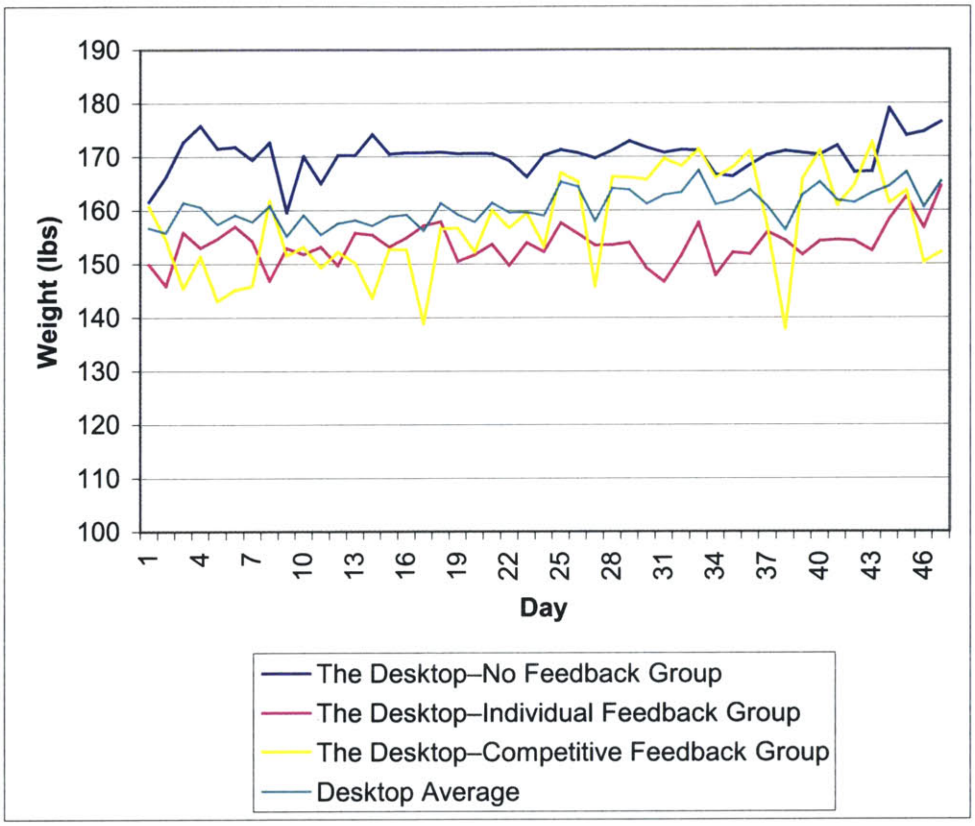


Exhibit 61: Desktop Users' Weight

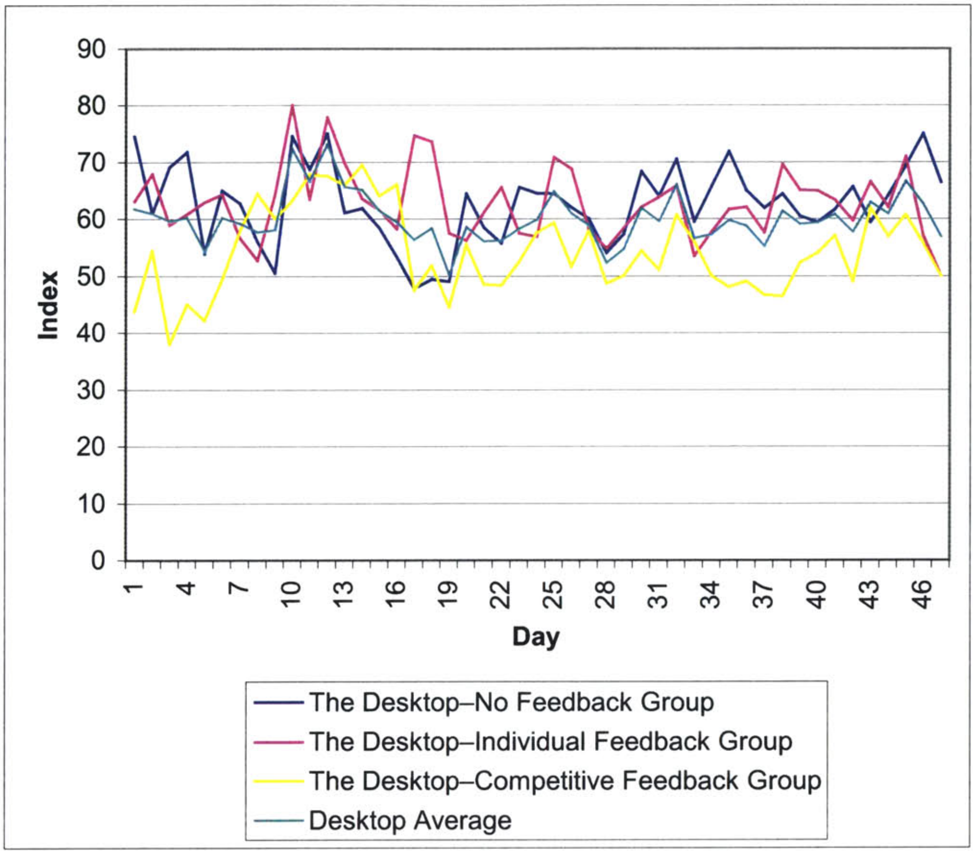


Exhibit 62: Desktop Users Perceived Level of Happiness measured on a scale from 0 to 100, where 0 is "not at all" and 100 is "very much."

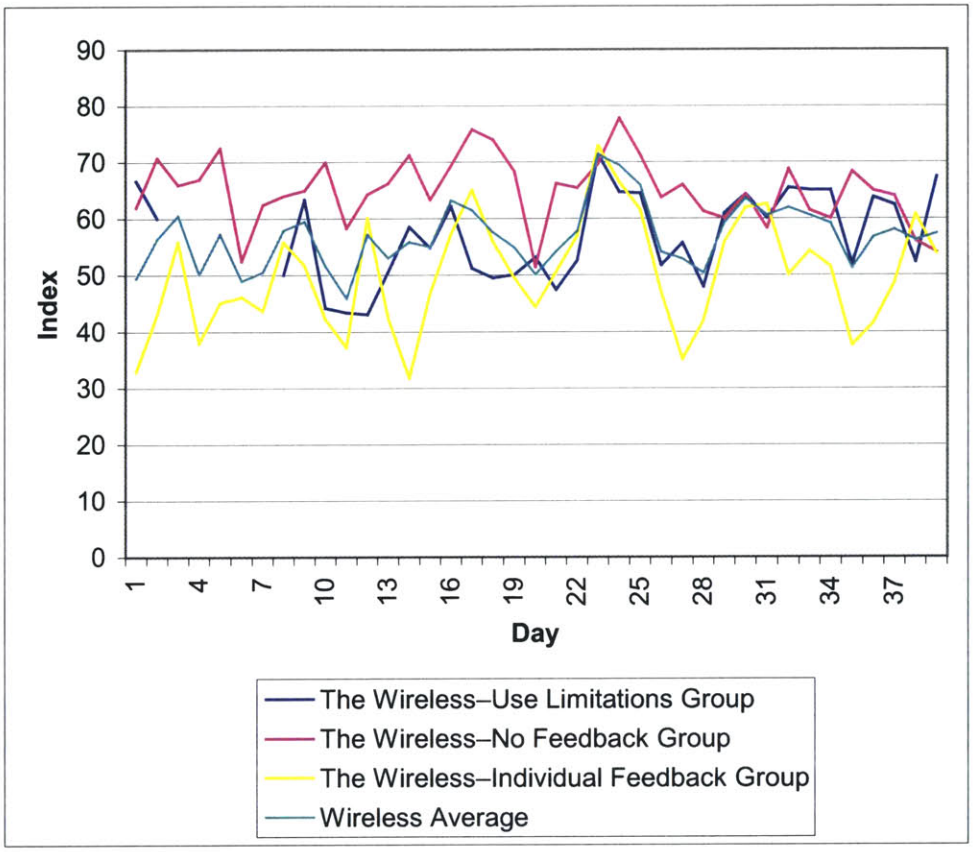


Exhibit 63: Wireless Users Perceived Level of Happiness measured on a scale from 0 to 100, where 0 is "not at all" and 100 is "very much."

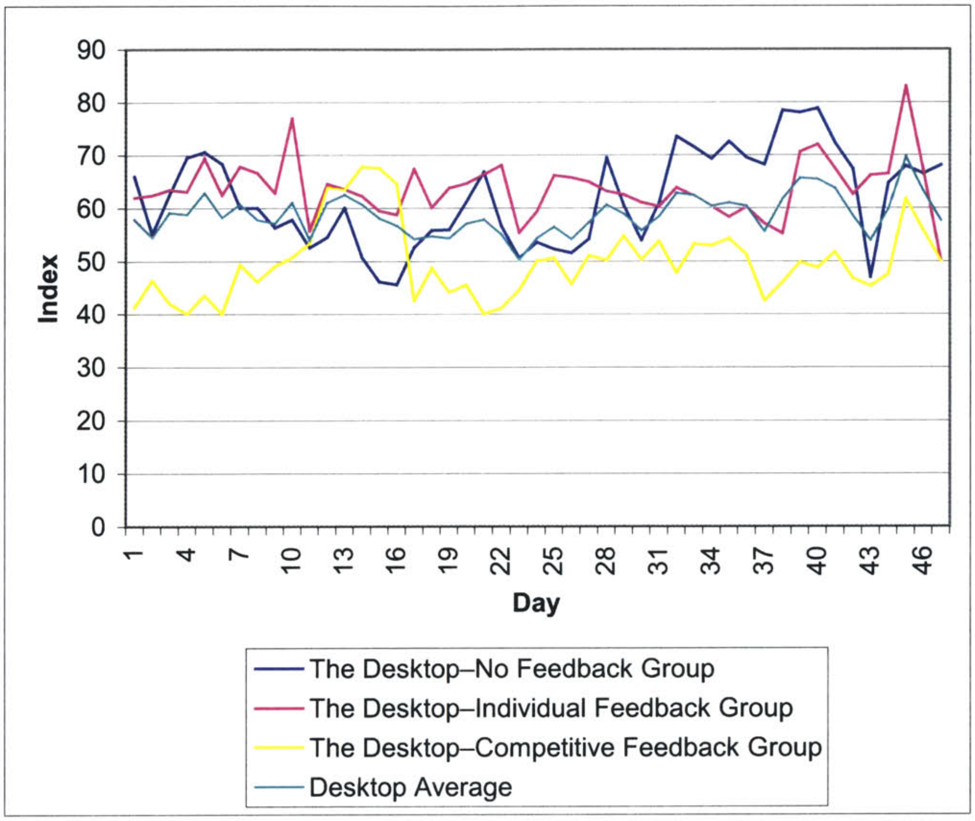


Exhibit 64: Desktop Users' Perceived Level of Healthiness measured on a scale from 0 to 100, where 0 is "not at all" and 100 is "very much."

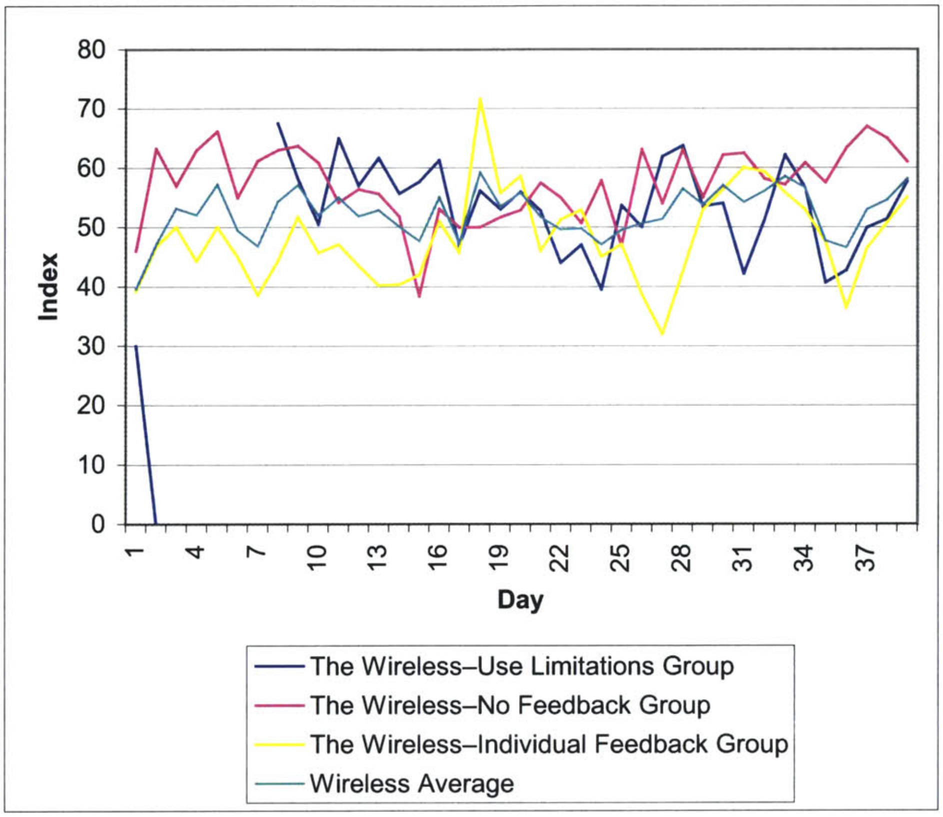


Exhibit 65: Wireless Users' Perceived Level of Health measured on a scale from 0 to 100, where 0 is "not at all" and 100 is "very much."

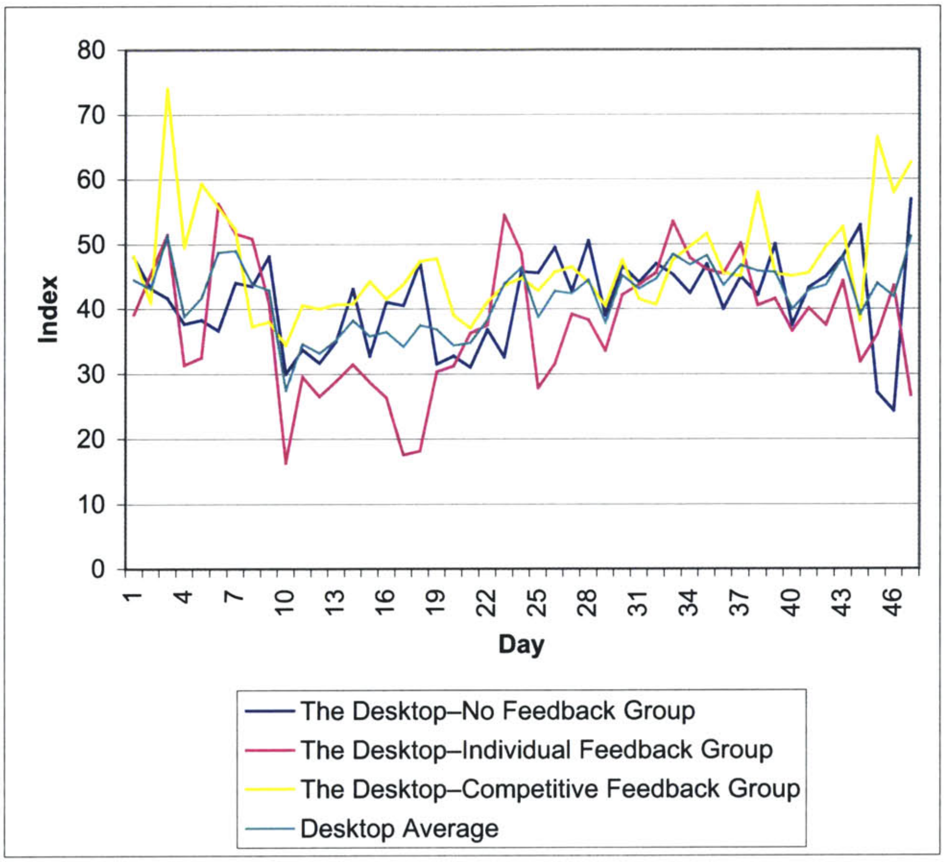


Exhibit 66: Desktop Users' Non-Academic Worries measured on a scale from 0 to 100, where 0 is "always" and 100 is "never."

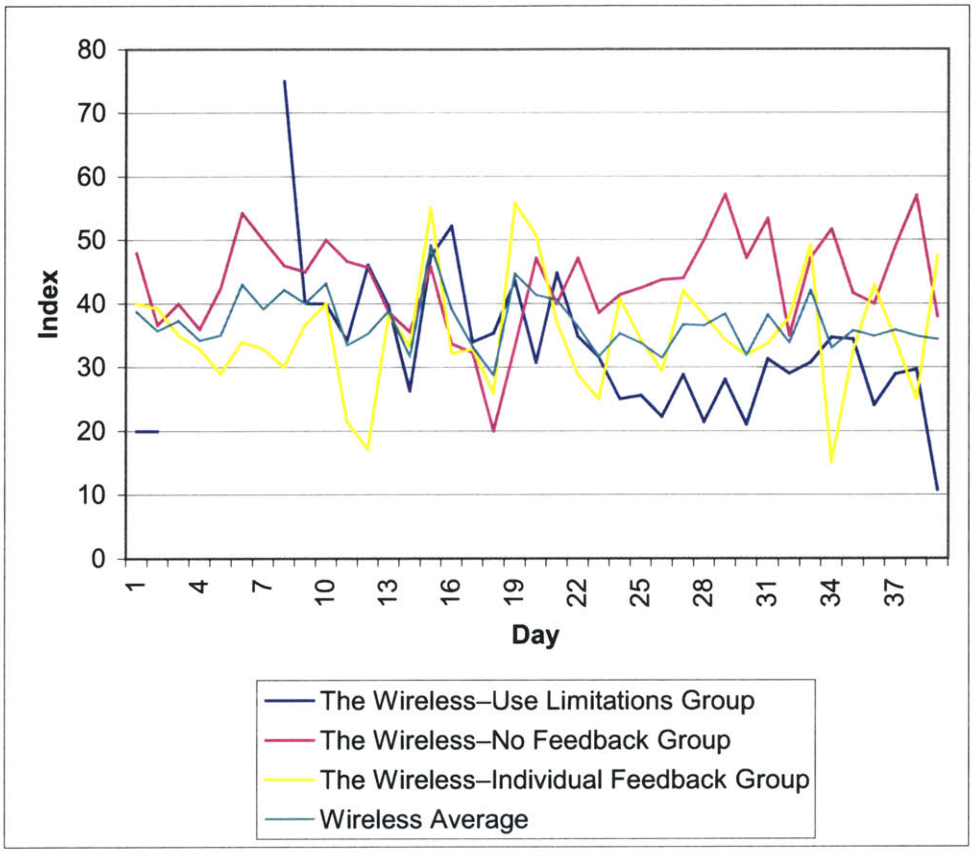


Exhibit 67: Wireless Users' Non-Academic Worries measured on a scale from 0 to 100, where 0 is "always" and 100 is "never."

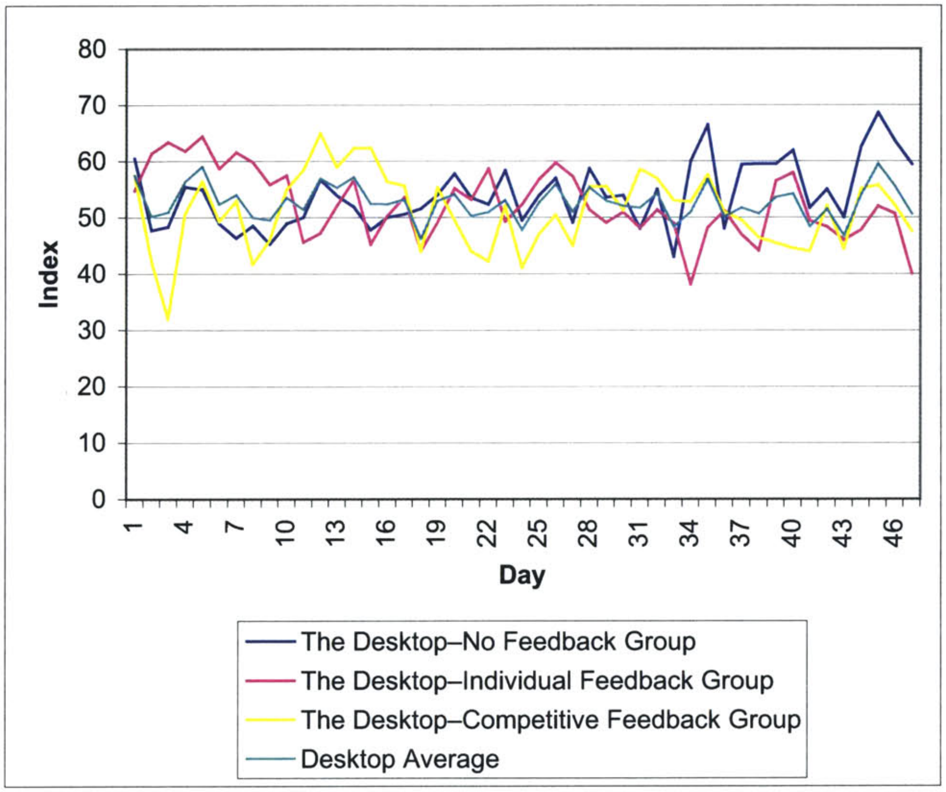


Exhibit 68: Desktop Users' Adherence to General Diet measured on a scale from 0 to 100, where 0 is "did not follow it at all" and 100 is "follow it perfectly."

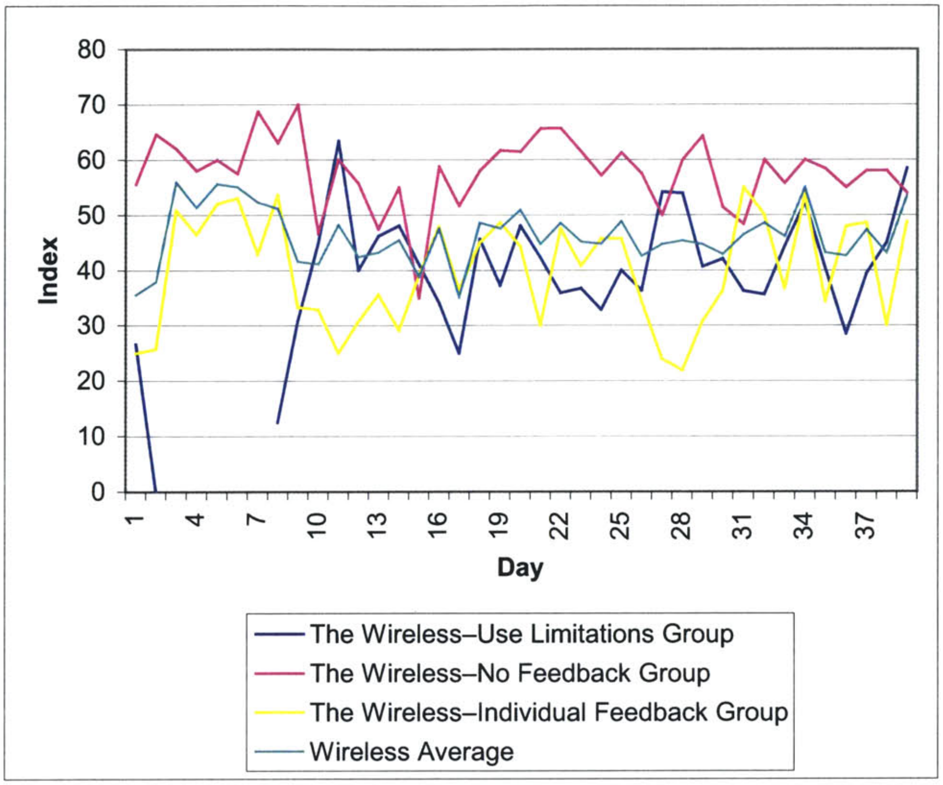


Exhibit 69: Wireless Users' Adherence to General Diet measured on a scale from 0 to 100, where 0 is "did not follow it at all" and 100 is "follow it perfectly."

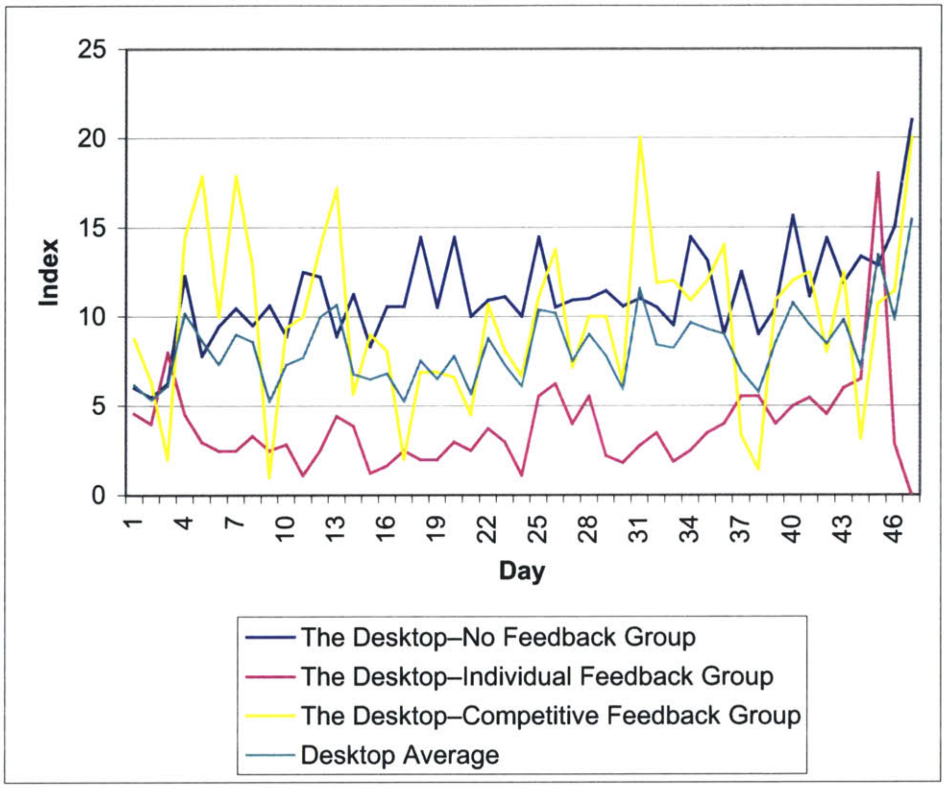


Exhibit 70: Desktop Users' Attention to Intake of Calories measured on a scale from 0 to 100, where 0 is "not at all" and 100 is "constantly."

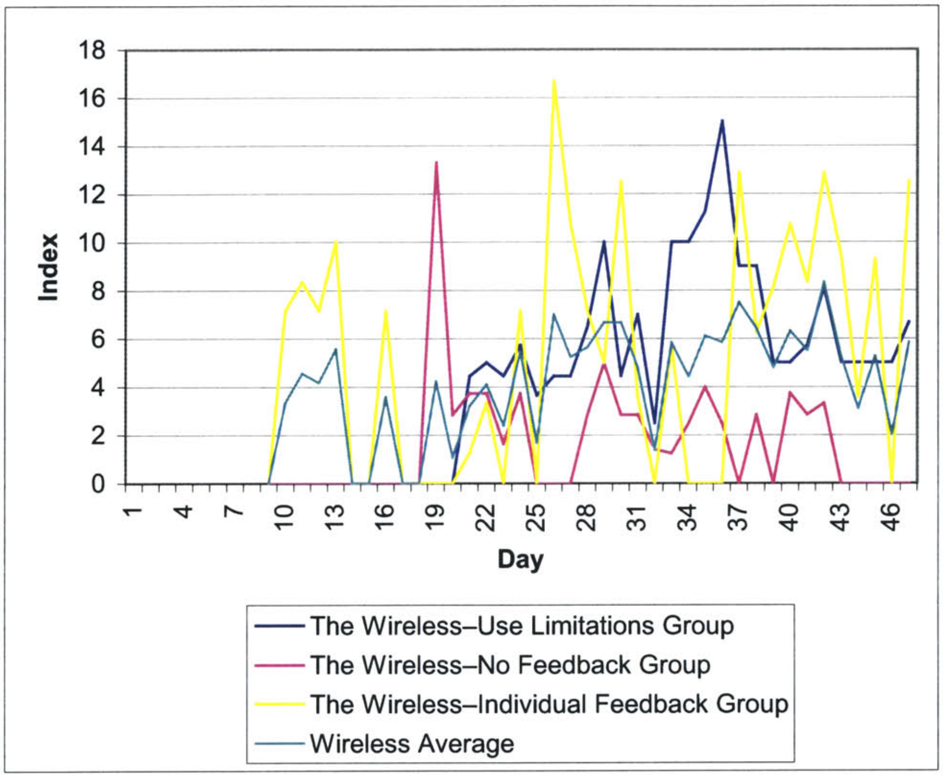


Exhibit 71: Wireless Users' Attention to Intake of Calories measured on a scale from 0 to 100, where 0 is "not at all" and 100 is "constantly."

13 References

[1] Grundfest, Warren S, Chair/Editor. *Health Care Technology Policy II – The Role of Technology in the Cost of Health Care: Providing the Solutions.* 12-12 May 1995. Arlington, VA. *SPIE–The International Society for Optical Engineering*, Volume 24997.

[2] Woody, Todd. "The Trillion-Dollar Opportunity," April 5, 1999. *The Industry Standard*, 42.

[3] Pearce, Paul J. "The One Thing Everybody Wants," Speech for Healthy Alternatives 2000. *Nova Wellness Institute.* January 10, 2000. New Spirit Naturals, Inc. Annual Convention. Monte Carlo Hotel, Las Vegas, NV.

[4] Lathrop, Philip J., Ahlquist, Gary D., Knott, David G. "Health Care's New Electronic Marketplace." *Strategy + Business*, Booz-Allen & Hamilton, Second Quarter, 2000. 35.

[5] Stone, David D. "Why Personal Health Monitoring?" Presentation, June, 2000.

[6] Kilborn, Peter T. "Ambitious Effort to Cut Mistakes in U.S. Hospitals." *New York Times.* December 26, 1999.

[7] WebMD Internet Homepage. http://www.webmd.com/corporate/content/news/1999/052099a_hlth_pr.html, April 2000.

[8] Hartman Group Internet Homepage. http://www.hartman-group.com/wellnessstudy.html, April 2000.

[9] Bette Runck, Ruth Kay (ed.) National Institute of Mental Health – Division of Scientific and Public Information. http://www.psychotherapy.com/bio.html, March 2000

[10] Kamiya, Joe, Barber, T.X., Miller, Neal E., Shapiro, David, Stoyva, Johann. *Biofeedback & Self-Control* (Aldine Publishing Company, Chicago. 1977), 24-25, 39-42.

[11] Abplanalp, H. , Bedell, P. "Visual improvement in an albinotic patient with an alteration of congenital nystagmus," *American Journal of Optometry and Physiological Optics*, 1987. Vol. 64 No. 12, 944-51.

[12] Asato, S., Twiggs, H., Ellison, D. G. "EMG biofeedback training for a mentally retarded individual with cerebral palsy," *Phys Ther*, 1981. Vol. 61 No. 10, 1447-51.

[13] Belar, C. D. and Cohen J. L. "The use of EMG feedback and progressive relaxation in the treatment of a woman with chronic back pain," *Biofeedback Self Regul*, 1979. Vol. 4 No. 4, 345-53.

[14] Asfour, H. L., Khalil, S. S., Waly, T. M., Goldberg, S. M., Rosomoff, M. L., Rosomoff, R. S. "Biofeedback in back muscle strengthening," *Spine*, 1990. Vol. 15 No. 6, 510-3.

[15] Bennink, C. D., Hulst, L. L., Benthem, J. A. "The effects of EMG biofeedback and relaxation training on primary dysmenorrhea," *J Behav Med*, 1982. Vol. 5 No. 3, 329-41.

[16] Allen, E. B. and Blanchard, J. K. "Biofeedback-based stress management training with a population of business managers," *Biofeedback Self Regulation*, 1980. Vol. 5 No. 4, 427-38.

[17] Vaschillo, E., Lehrer, P., Rishe, N., Konstantinov, M. (1984) "Voluntary Control of Slow-Wave Heart Rhythm Structure: Frequency Analysis of the Cardiovascular System." Dissertation Paper.

[18] Lehrer, P., Carr, R. E., Smetankine, A., Vaschillo, E., Peper, E., Porges, S., Edelberg, R., Hamer, R., Hochron, S. "Respiratory Sinus Arrhythmia Versus Neck/Trapezius EMG and Incentive Inspirometry

Biofeedback for Asthma: A Pilot Study." *Applied Psycholphysiology and Biofeedback*, 1997. Vol. 22 No. 2, 96-97, 107.

[19] Tokarev, Vlad. "The Use of Brain Blood Flow Biofeedback in Occupational Medicine." http://www.webideas.com/biofeedback/research/vtokarev.shtml. April 2000.

[20] Allen, G. K., Mills, R. A. "The effects of unilateral plethysmographic feedback of temporal artery activity during migraine head pain," *J Psychosom Res*, 1982. Vol. 26 No. 2, 133-40.

[21] Bertoti, D. B. and Gross A. L. "Evaluation of biofeedback seat insert for improving active sitting posture in children with cerebral palsy: a clinical report," *Phys Ther*, 1988. Vol. 68 No. 7, 1109-13.

[22] "EyeTrace EyeTone system for Auditory Biofeedback" http://www.aso.edu.stockholm.se/~wth0gabc/iota/biofeedback.htm. April 2000.

[23] Lau, M., Debski, M., Wheeler, A., Hawley, M. "ESP: A Complete Solution for Embedded Sensor Packs." MIT Media Laboratory – Personal Information Architecture Research Group. Fall, 1999.

[24] The d'Arbeloff Laboratory for Information Systems and Technology – Wearable Sensors Research Lab Internet Homepage. http://darbelofflab.mit.edu/research/research.html. May 2000.

[25] Health Hero Network Internet Homepage. http://www.healthhero.com/products/index.html. April 2001.

[26] "Health Hero Outcomes: Clinical and Financial Analyses of Programs in Congestive Heart Failure." Health Hero Network Case Study, 2000. [online] http://www.healthhero.com/papers/studies/Meta-Analysis_CHF_Outcomes.pdf.

[27] "Health Hero iCare Platform Overview and Technical Brief." Health Hero Network White Paper, 2000. [online] http://www.healthhero.com/papers/studies/Health_Hero_Whitepaper.pdf.

[28] Roy, S.H., Kupa, E.J. "An Investigation of the Effects of Muscle Adaptation on the EMG Signal." Boston University Electrophysiology Laboratory. http://nmrc.bu.edu/labs/epl/papers/adapt/index.html. March 2000.

[29] Oddsson, L. "Risk Factors of Muscle Injury and Falls Related to Load Handling Technique." Boston University's Injury Analysis & Prevention Laboratory. http://nmrc.bu.edu/labs/iapl/papers/injury/index.html. March 2000.

[30] Abplanalp, H., Bedell, P. "Visual improvement in an albinotic patient with an alteration of congenital nystagmus," *Am J Optom Physiol Opt*, 1987. Vol. 64 No. 12, 944-51.

[31] Dell'Osso, L.F., "Therapies for Congenital Nystagmus." Ocular Motor Neurophysiology Laboratory, Cleveland Veterans Affairs Medical Center.

[32] Tokarev, V. "The Use of Brain Blood Flow Biofeedback in Occupational Medicine." [online] http://www.webideas.com/biofeedback/research/vtokarev.shtml.

[33] Abramowitz, N.W., Bell, S.I. "Biofeedback, self-control and tension headache," *J Psychosom Res*, 1985. Vol. 29 No. 1, 95-9.

[34] Achterberg, D. , Kenner, J., Casey, C. "Behavioral strategies for the reduction of pain and anxiety associated with orthopedic trauma," *Biofeedback Self Regul*, 1989. Vol. 14 No. 2.

[35] Miller, N.E. "Some professional and scientific problems and opportunities for biofeedback," *Biofeedback Self Regul*, 1985. Vol. 10 No. 1.

[36] Miller, N.E. and Dworkin, B.R. "Effects of learning on visceral functions--biofeedback," *N Engl J Med*, 1977. Vol. 296 No. 22, 1274-8.

[37] Manuso, J.S. "The Equitable Life Assurance Society program," *Prev Med*, 1983. Vol. 12 No. 5, 658-62.

[38] Miller, J. "Discussion of the human-potential movement: implications for psychoanalysis," *Am J Psychoanal*, 1977. Vol. 37 No. 2, 111-4.

[39] Halter, P. "Challenges and Opportunities Under Managed Competition." Medical Data International, April 1994.

[40] Harville, Keith. "Closing the System: The Role of Feedback in Controlling Health Care Costs." Information Services Marketing, VHA, May 1995.

[41] Glazkova, T.G. and Petrovski, A.M. "Control Aspects of Medical Diagnosis." Control Aspects of Biomedical Engineering, 1987.

[42] Wennberg, J. "Empowering Patient Decision Making." Medical and Biological Engineering in the Future of Health Care, 1994.

[43] Karlins, Marvin and Andrews, L.M. *Biofeedback – Turning on the Power of Your Mind.* (J.B. Lippincott Company, Philadelphia. 1972)

[44] Inglis, J., Campbell, Dugal, Donald, M.W. "Electromyographic Biofeedback and Neuromuscular Rehabilitation." *Biofeedback & Self-Control*, 1977.

[45] Wyler, A.R., Lockard, J.S., Ward, Arthur, A., Finch, C.A. "Conditioned EEG Desynchronization and Seizure Occurrence in Patients." *Biofeedback & Self-Control*, 1977.

[46] Abowd, G. "Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment." *IBM Systems Journal*, 1999. Vol 38 No. 4.

[47] Abowd, G., Atkeson, C., Essa, I. "Ubiquitous Smart Spaces." College of Computing, Georgia Institute of Technology Dissertation.

[48] Andrade, J. *Medical and Biological Engineering in the Future of Health Care.* (University of Utah Press, Salt Lake City, Utah. 1994)

[49] Beck, B., McGinnis, M. IBM WebSphere Everyplace Suite v1.1 White Paper. September 12, 2000.

[50] Hermans, B. "Intelligent Software Agents on the Internet: An Inventory of Currently Offered Functionality in the Information Society and a Prediction of (Near) Future Developments." *First Monday*, March 3, 1997. Vol. 2 No. 3.

[51] Hodes, T., Katz, R. "Composable Ad hoc Location-based Services for Heterogeneous Mobile Clients." Computer Science Division, University of California, Berkeley. Dissertation, May 5, 1998.

[52] Karlins, M., Andrews, L. *Biofeedback – Turning on the Power of Your Mind.* (J.B. Lippincott Company, Philadelphia. 1972)

[53] Myers, B., Hudson, S., Pausch, R. "Past, Present and Future of User Interface Software Tools." Human Computer Interaction Institute, Carnegie Mellon University. Dissertation, September 15, 1999.

[54] Pentland, A. "It's Alive!" *IBM Systems Journal*, 2000. Vol 39 Nos. 3 & 4.