

HANDHELD COMPUTER FOR PERSONAL HEALTH ADVICE

by

Cédric Hutchings

Engineer,
ECP - Paris, France, 1999

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2002

© 2002 Massachusetts Institute of Technology

All Rights Reserved

Signature of Author:

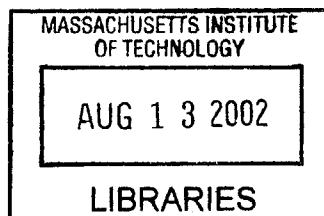
.....
Department of Aeronautics and Astronautics
February 29, 2000

Certified by:

.....
Thomas B. Sheridan
Professor of Engineering and Applied Psychology
Thesis Supervisor

Accepted by:

.....
Wallace E. Vander Velde
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Student



AERO

Handheld Computer For Personal Health Advice

By

Cédric Hutchings

Submitted to the Department of Aeronautics and Astronautics
On February 1, 2002 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Aeronautics and Astronautics

ABSTRACT

The overall objectives of the project were to conceive, design, implement and evaluate a portable personal health advisor. The system would integrate data from various sources and handheld computers on a real-time basis to provide “advice” (intelligent and knowledge) anytime and anywhere so that a consumer can take practical and immediate actions to maximize their vitality to enjoy life.

The conception phase included the identification of technologies and approach to deliver personal and real-time advice at points of decision. Using this knowledge, I developed a prototype in the context of a person shopping in a pharmacy or market to ensure that foods, medicines or other products do not pose hazards relative to that person’s known medical condition. In order to implement a real-life and thus estimate the usability and the user’s perception of the system, we chose to focus on Type II diabetics in this first phase.

A field test was run at a Research Laboratory of the major project sponsor, Procter & Gamble. The experiment confirmed the great appeal of decision-aiding system for people who have to manage strict or complex diet. It also underlined its design challenges, because even an ubiquitous device that provides “perfect advice” would promote a long-term behavior change only if human factor issues related to trust, privacy and user perception are carefully analyzed. It finally revealed the need for a trusted infrastructure, and some of the difficulties to overcome in order to implement the system in a large scale.

Thesis Super: Thomas Sheridan

Title: Professor of Aeronautics and Astronautics

Acknowledgments

The author would like to acknowledge several people who helped in the development of this work. First, Professor Thomas Sheridan from MIT and Dr. Joseph F. Coughlin, Director of the MIT Age Lab provided both support and guidance. Hu Hsin-Yuan, from Procter & Gamble, brought enthusiasm from the main sponsor all along the project, and organized the panel discussion and the field test run at P&G research laboratory in Cincinnati. Victor Mills, from Spencer Hall, helped to prepare, run and analyze the results from the experiment.

Table of contents

1	Motivation.....	6
1.1	Overall objective	6
1.2	Technology: Requirements and Availability	7
1.2.1	Portability.....	7
1.2.2	Wireless communication with the Internet	8
1.2.3	Ease of use	8
1.2.4	Enforce privacy of the user	9
1.3	State of the art of portable computers and wireless technologies	9
1.4	Overall architecture of the system	10
1.5	The set of databases	11
1.5.1	Product Database.....	11
1.5.2	User Database	13
1.5.3	Issues database	13
2	Methodology	14
2.1	Focus: the Diabetics Diet Case	14
2.1.1	Background	14
2.1.2	General Guidelines for control of diabetes	15
2.1.3	Meal planning principles.....	16
2.2	Prototype Design.....	17
2.2.1	Overall view and concept.....	17
2.2.2	Components: functions and technology.....	19
2.2.3	The rules system.....	32
2.2.4	System View	34
2.3	Experiment.....	35
2.3.1	The P&G facilities	35
2.3.2	Field Test.....	36
3	Analysis and Results	39
3.1	User Feedback.....	39
3.1.1	Behavior	39
3.1.2	Hardware and user interface	41

3.1.3	Features	43
3.2	Improvements.....	47
3.2.1	Interface Improvement Suggestions.....	47
3.2.2	Feature Improvement suggestions	49
3.2.3	How to Enforce Trust and Change the Behavior	51
3.3	Perspectives beyond the diabetics case.....	53
3.3.1	Real-time physiological monitoring device	53
3.3.2	Expert system and rules development.....	54
3.3.3	Large-scale deployment	54
3.3.4	Conclusions	56
Appendix A:References		58
Appendix B: Smart Advisor : Tasks Analysis		63
Appendix C: Database structure for the diabetes breakfast application		68
Appendix C: Server Codes.....		71
Appendix D: Client Codes		89

1 Motivation

1.1 Overall objective

Decision-aiding systems have been extensively studied and implemented for advanced applications, including aerospace and nuclear environments for decades. Sophisticated risk assessment tools are widely used in military fields, where an operator alone cannot integrate the complex data to take a decision in the required timeframe.

The research effort explores the potential of hazard monitors, typically used in military aircraft, to very different situations, designated as “personal safety”. A Hazard Monitor (HM) is a rules-based system that alerts human operators to discrepancies between actual and expected system states in a human-centered and context-sensitive manner before the adverse consequences of the discrepancies become unavoidable. Simply defined, personal safety is enhanced when an individual has sufficient warning before taking an action which exposes him or her to a risk.

The overall objectives of the project are to conceive, design implement and evaluate a portable personal health advisor. The conception phase includes the identification of technologies and approach to deliver personal and real-time advice at points of decision.

More precisely, the first goal will be identify technologies and approaches that will enable the development of a highly portable “**Personal Advisor**” that integrates data from various sources and personal-devices on a real-time basis to provide “advice” (intelligence and knowledge) to the consumer anytime and anywhere so that these consumers can take practical and immediate actions to maximize their vitality to enjoy life. Potential types of information will include: (i) personal health monitoring devices (diet & nutrition, physical symptoms, exercise), (ii) an external data-base that “bench-marks” personal data against a population norm, (iii) product information and recommendations (iv) location of shops, availability of products, and prices.

The research effort is in the context of a person shopping in a pharmacy or market to ensure that foods, medicines or other products do not pose hazards relative to that person's known medical condition. Shoppers in a supermarket or drug store often have difficulty in ascertaining whether any particular product contains ingredients which pose some hazard — because of reactions to medications they are taking, or because of dietary or allergy restrictions. When examining a food or drug or cosmetics package:

1. They have to remember or be aware of all their special diets, the substances they must be careful to avoid, and the cross-reactions with their medications or allergies.
2. Then they have to check whether any considered product is incompatible with their own health profile by browsing through the often unreadable, incomprehensible, and incomplete content information printed on the package. The latter is especially difficult for older adults, those with poor eyesight, or those without technical knowledge.

Thus the challenge is to:

1. Design and evaluate an architecture that increase independence of elderly people,
2. Help people avoid hazardous products (cross-reactions, allergies) and promote personal diet,
3. Uncover examples of medical errors cause by bad packaging information
4. Provide health related advice at the point of decisions,
5. Provide a useable and portable consumer decision-aiding system,

and in the end test the usability of a prototype device or system and evaluate the potential change of behavior and other human factors issues of a cross section of individuals using the device.

1.2 Technology: Requirements and Availability

1.2.1 Portability

As the system is to be accessible at points of decision, its most important feature is the portability of the access platform. The user should have access to advice anywhere and at anytime. As the advice is personalized and customized, the user should carry his or her personal

information or have a means to retrieve it easily and safely. Portable handheld computers offer some good opportunities for our research. The personal digital assistant (PDA) is defined as a digital device which can be carried in a pocket and is usually operated by holding it in one's hand and entering data by means of a pressure stylus or a tiny key set. It has been commercialized for a decade or more. First it was a pocket calculator, then a means of storing and retrieving names, addresses and telephone numbers, and then a memory pad for taking notes. More recently powerful software, like spreadsheets or word processing applications have expanded the functional capability. The most promising upgrades have been the addition of extension slots, which enable interfacing with a large number of companion devices. The future promises that it will get smaller still, be integrated with clothing or wristwatch, and take on many functions¹.

1.2.2 Wireless communication with the Internet

In addition to be portable, the system clearly needs wireless capabilities to connect to the Internet through wireless telecommunication carriers. The mission statement mentioned personal advice on site anywhere, anytime. Without entering yet into the architecture design, the wireless network feature appears as a prerequisite by the nature of the data involved. Indeed, in order to provide personalized advice on a product one needs comprehensive information on it. It is not realistic to intend to store an up-to-date comprehensive US food product database on the user's portable computer. Thus some interaction with a remote database server is essential, and data wireless networks offer the most flexible solution. Specific details on the selected technology are given in the prototype design section.

1.2.3 Ease of use

The purpose of the research is to enhance the shopping experience of the user, by making it faster, easier and safer. In line with the Age Lab focus, we think that an aging population might particularly benefit from the system. It would enable a senior to gain back some independence when dealing with his medication or special diet. As a consequence, the user interface requirements are deliberately high, because this segment of population is not today the most

¹ MIThril project at the MIT Media Lab <http://www.media.mit.edu/projects/wearables/index.html>

tech-savvy and the most eager to be equipped with an additional electronic gadget. It implies to minimize the tasks required to interact with the device.

As mentioned above, extension slots and connectors allow adding some valuable input device in order to simplify the product selection. The barcode technology seems a very convenient choice for this task. It offers a simple alternative to the manual input of product names. Thus it drastically reduces the complexity of the task and makes it quicker. Depending on the technology, the user has either to point toward or to scan the traditional barcode displayed on most consumer packages. The other main advantage of relying on the barcode system is that is an industry-wide standard and the database of products along with their barcode could be available. More details on the technology are given further, with specific device and solutions to connect it to the handheld computer.

1.2.4 Enforce privacy of the user

Because some private information can be exchanged, the user might not be willing to share his diet or medical condition with people who shop nearby. This implies some stringent requirements of the interface to ensure the user's privacy. Loud speakers and voice-recognition technologies will not be considered for a first prototype because of the nature of the information conveyed in a public and environment like supermarkets.

In addition to providing privacy at the user interface level, the system should enforce confidentiality along the whole chain of information to protect the user's personal data from unauthorized access. Encryption technologies are today widely available for wireless networks, and with proper use can provide the system with the three fundamental features of authenticity, integrity and confidentiality of the conveyed data. They will be discussed in the context of the selected device (Palm VII).

1.3 State of the art of portable computers and wireless technologies

The quick overview of the required features allows us to narrow the scope in a search for adequate technology. Thus the device of choice for the user platform is a wireless-enabled portable device with extension capability in order to plug a barcode scanner. The different pieces should be integrated well enough to allow a simple and quick manipulation by a senior user in a supermarket. The following lists the potential candidate with their relevant features.

which determines the boundaries of the embedded software, (2) Wireless communication capabilities, in order to achieve the “anytime, anywhere” goal, (3) User Interface, and (4) Hardware expandability (add biosensors and extension device). Two families of PDA, one operate by Palm OS and the other one by Windows CE competed at that time. The latter certainly offered advanced User Interface, with color touch screen and sound. However we selected the Palm VII as the only PDA that had a built-in wireless modem. This allowed us to reduce the number of additional accessories to be externally plugged.

1.4 Overall architecture of the system

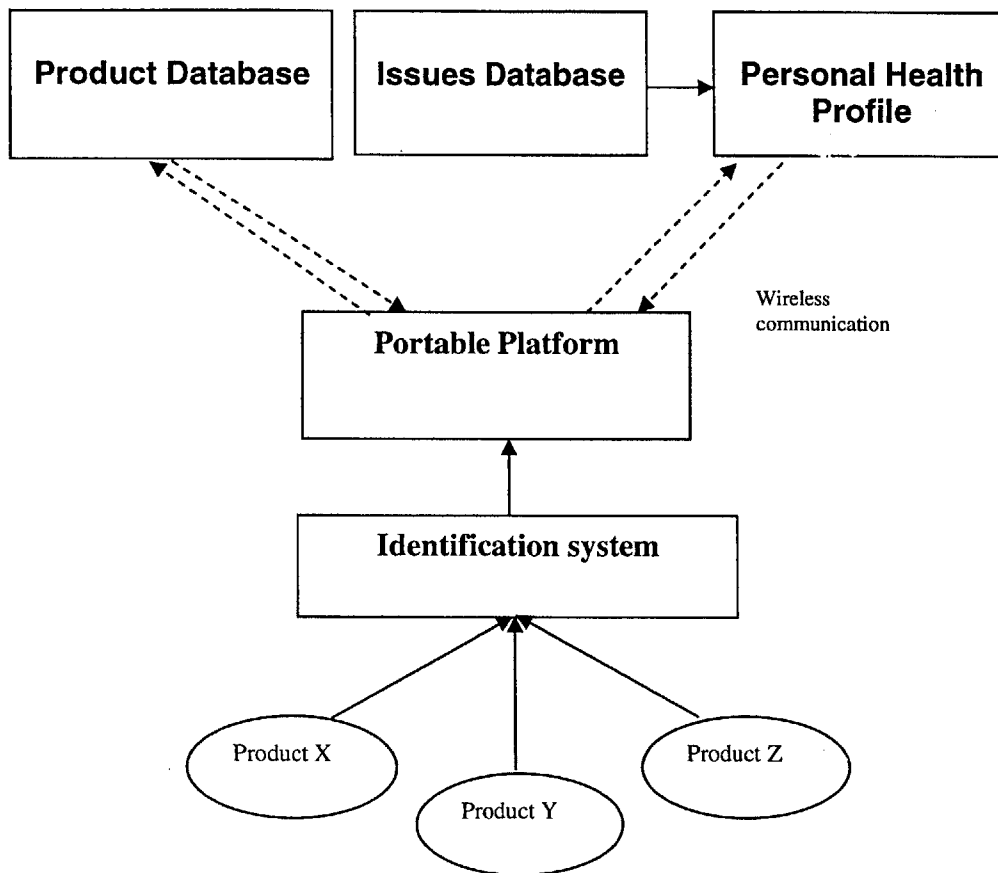


Figure 1: System Architecture

The schema in figure 1 illustrates the initial vision we had for the system, at the conceptual level. The handheld device plays a central role, as a gateway between different sources of information and stakeholders. The various parts are described in the next part.

1.5 The set of databases

1.5.1 Product Database

1.5.1.1 The UPC codes

This database first has to provide a convenient way to identify a target product. A solution currently used by the retail industry is the UPC Barcode label printed on most packaged items. Unfortunately, there is no comprehensive UPC database in the US. Each manufacturer is assigned a range of numbers, and manages their database independently. Keeping this data up to date is a huge business for the manufacturers, who are reluctant to release it publicly. An interesting initiative can be found on <http://www.upcdatabase.com/> . It is a huge UPC code database that is freely available. As it can be updated online by anyone, it is not very reliable, but gives a good example about how a distributed system could be implemented: the management of the central nationwide database could be shared by a large number of actors, insofar as identification and verification processes can be provided.

The UPC database is the very first step toward identification, because it only provides a link between a barcode, and a name of a product (manufacturer...). It is useful because of the need to simplify the identification process for the customer: an elderly person could easily “scan” a package, and forget about the pain of reading complex and sometimes misleading labels. Let’s note that there are or will be alternative ways to achieve this task, like the ID Tag project run at MIT that enables one to uniquely identify an item, not only in the generic and conventional way, but with a greater data exchange using an embedded radio chip in each package.

Once a product is identified in one of the ways cited above, one can retrieve content information on the product. This is a separate process, the first one being a typical logistic problem, and the latter involving more complex data. What do we mean by content information? To achieve the functionalities described below, one needs at least nutrient and ingredient data. I think it is relevant to split the information for several reasons: the sources (and ease of access), the way they are structured, (simple and structured, raw list, see package), their usefulness, and potential applications are quite different.

1.5.1.2 The Nutrient Database

This database contains the “low level” information on a food product. Each nutrient (for example, Sodium, Cholesterol, etc) is quantified, per given amount of the item, usually the serving size. The Nutrient Data Laboratory from the US Department of Agriculture provides such a database (USDA Nutrient Database for Standard Reference). The database describes the content of over 6000 generic food products for 80 nutrients. I found out that this is a primary source of information used by numerous commercial service and products (see Appendix A). Besides that it is freely available, and it has valuable features from our project perspective, and some limitations as well.

One of our goals is to do crosschecking reference, or pattern recognition between the personal health profile and the product information. For that purpose having a database with homogenous nutrient profiles for the food product is very valuable. The limitation is that mostly generic foods are described, rather than brand name products. One still has to fill the gap between the two worlds. For my research, I chose to manually update the database, with the UPC code when a brand name product is described. Although this is acceptable at this stage, it raises some questions for a large-scale infrastructure: how and who should manage a comprehensive nutrient database for brand name products?

1.5.1.3 The ingredient database

An ingredient database provides the system with alternative features as well as specific conflict difficulties to be understood. To watch allergies, or cross reactions with medication for example, one needs to check an item against the presence of specific foods, like peanuts or grapefruit. But the ingredient list of foods product are raw data as compared to well structured and homogenous nutrient profiles. Thus this information is trickier to manage, and the queries to such a system are more complex. In addition, whereas the nutrient profile can be obtained from any food (although costly, for example \$2000 for one sample), one can easily understand that the recipes are proprietary information that the manufacturers are very reluctant to release.

For a first implementation, this can be acceptable because we could restrict ourselves to the potential hazardous ingredients, that the manufacturers are required by law to declare (known

allergens for example). But for a safer and more comprehensive system, the question on the sharing this information will have to be addressed eventually.

1.5.2 User Database

The user database contains all the information that could interfere with a food purchasing decision. That is basically:

Diets (sugar, fat, cholesterol, sodium...)

Allergies (type, level of seriousness...)

Medical prescription with drug-food and Drug-drug cross-reactions.

Depending of its level of complexity, this information should be updated by the patient herself, or her doctor, or a health advisor. That is the main reason for hosting the profile database on a remote server, rather than on the handheld device itself: it enables different actors to collaborate on a single and integrated platform. But it also raises obvious privacy issues: the profile contains personal data that should only accessible to authorized persons. One solution would be to implement different levels of access: only the patient has a full access to his profile, whereas the other actors can interact with a specific part of the database.

1.5.3 Issues database

The issues database will contain the rules that allow generating knowledge from the crosschecking between the personal database and the set of product database. Hence it will be a key element of the system, because the user database described above will be “built” from this one. For example, it contains extensive quantitative descriptions on diets/restrictions associated with specific diseases, and known food-drug or drug-drug interactions. This database exists in pieces for common conditions, but can in principle be compiled from the literature for any domain.

2 Methodology

As mentioned earlier the approach adopted is the development and evaluation of a prototype. This first phase enables us to evaluate the concept and assess the relevant technologies. In order to implement a real-life case and thus estimate the usability and the user perception of the system, it is necessary to narrow the initial scope down to a specific application. The long term goal is to build a platform that handles a large number of health issues like allergies and special diets, but we choose to focus on the Type II diabetic in the first phase of the research. The motivations for this choice are explained in the next part.

Despite the advantages of this bottom-up approach (from a particular application to the general case), it has some drawbacks. The principal risk is to design a system architecture that is only relevant for the particular case, and that is not expandable. To my opinion this risk is minimized if the tradeoff between short and long-term objectives is clearly defined. Therefore it is appropriate to underline them again at this point: In the case study stage (the Diabetics Case) the aim is to implement a simple but realistic application and get feedback from users. Although we should constantly keep in mind during this phase that is not the end goal of the research, we will be free to take some distance from the scalability of the system, and focus on the human factors evaluation side.

2.1 Focus: the Diabetics Diet Case

2.1.1 Background

Some recent reports indicate that in four of the ten leading causes of death in the United States (cardio-vascular disease, certain types of cancer, stroke and diabetes) diet and lack of physical activities are significant contributing factors. It has been well documented that a healthful diet reduces the risk or the consequences of chronic diseases such as cardiovascular diseases and certain forms of cancer. According to the American Diabetes Association, there are about 15.7 million people who have been diagnosed with diabetes (including Type I or inherited and type II or acquired) -- 5.9% of the population of the U.S. That number increases to 25% for seniors over 85. Type 2 diabetes accounts for 90 to 95 percent of diabetes. Type 2 diabetes is nearing epidemic proportions, due to an increased number of older Americans, and a greater prevalence of obesity and sedentary lifestyles.

In addition to being a major public health issue, diabetes is an interesting case to consider in the context of our research for several reasons. First, diabetics are heavy food package label readers, because they have to comply strictly with personal nutrition diets. But the generic data displayed is rarely enough to take informed purchasing decisions. Thus they can benefit from an electronic personal advisor, which would more comprehensively convey the data relevant to their condition. Furthermore, part of the research is to evaluate the potential long term behavior changes that the system could foster. Type II diabetics, whose disease typically starts at when they are middle-aged, have usually developed lifelong diet and behavior patterns that they should then get rid of. For these people, it is particularly hard to alter their nutrition habits, even though they may be perfectly aware that they should do it.

By focusing on Type II diabetes, we are able to address some of the central questions of the research project: Does the system help the user to sustain a healthy behavior? How “strict” should be the system with its user, in order to maximize the compliance with the advice? What are the fundamental requirements in order to enforce “trust” in the system?

Besides the fact that diabetes is appropriate for the human factors aspect for the research, it gives a valuable base case in order to design the system architecture, because it calls for the general features mentioned in the technical requirements statement. A diabetic diet generally involves several health professionals, such as a medical doctor and a nutritionist, heterogenous sets of data on the products, and both the physical activity and blood sugar level of the user. In order to coordinate these resources, one needs to build a platform that will be used later on when expanding toward new applications.

2.1.2 General Guidelines for control of diabetes

This part gives the background on Type II diabetes control necessary to develop the system.

Diabetes is a disease in which the body does not produce or properly use insulin. Insulin is a hormone that is needed to convert sugar, starches and other food into energy needed for daily life. The Type II version is metabolic disorder resulting from the body's inability to make enough of, or properly use, of insulin. It is the most common form of the disease.

Diet is very important in diabetes. Below is a guideline with some general principles on which we will build the personal profile and recommendations.

Patients with Type I diabetes should have a diet that has approximately 35 calories per kg of body weight per day (or 16 calories per pound of body weight per day). Patients with Type 2 diabetes generally are put on a 1500-1800 calorie diet per day to promote weight loss and then the maintenance of ideal body weight. However, this may vary depending on the person's age, sex, activity level, current weight and body style as shown below. More obese individuals may need more calories initially until their weight is less. This is because it takes more calories to maintain a larger body and a 1600 calorie diet for them may promote weight loss that is too fast to be healthy. Men have more muscle mass in general and therefore may require more calories. Muscle burns more calories per hour than fat. (This is also one reason to regularly exercise and build up muscle) Also, people whose activity level is low will have less daily caloric needs.

Generally, carbohydrates should make up about 50 percent of the daily calories (with the accepted range 40-60 percent). In general, lower carbohydrate intake is associated with lower sugar levels in the blood. However the benefits of this can be cancelled out by the problems associated with a higher fat diet taken in to compensate for the lower amount of carbohydrates. This problem can be improved by substituting monounsaturated and polyunsaturated fats for saturated fats.

Most people with diabetes find that it is quite helpful to meet periodically with a dietician or nutritionist for a consultation about what is the best diet for them and how many daily calories they need. It is quite important for diabetics to understand the principles of carbohydrate counting and how to help control blood sugar levels through proper diet. The system we developed assists the diabetic in this task, in various situations of his daily life.

2.1.3 Meal planning principles

- Control carbohydrates

Carbohydrates are a general term that includes both starches and sugars. The total amount of carbohydrate is more important than the source of carbohydrate. Carbohydrates can be either refined and processed such as in soft drinks and candy, occur naturally such as in milk and fruit, and can be found in pasta, bread, rice, dried beans and peas, and starchy vegetables such as potatoes, corn or green peas.

In other words, carbohydrate in the form of cookies doesn't affect blood sugar any differently than carbohydrate from rice. However, carbohydrate foods in the form of grains, fruits or milk have vitamins, minerals and fiber, while sugary foods such as cookies have little nutritional value and are often high in fat.

Therefore, the majority of carbohydrate a diabetic consumes should come from the grains, fruit or milk groups. It's okay to eat high-sugar foods, but they must be substituted for other carbohydrates, rather than just added to the meal plan. One serving of grains, fruit or milk provides about 15 grams of carbohydrate and is often referred to as 1 carbohydrate choice. For most people, 45-75 grams of carbohydrate per meal, and no more than 15-30 grams of carbohydrate per snack is a reasonable goal. However, the meal plan should be individualized to meet specific needs, preferably by a registered dietitian.

- Eat less fat

Diabetics have a higher risk of heart disease, and therefore they are encouraged to eat a low fat diet. They should choose foods that are lower in fat, use lower fat cooking methods and limit the amount of extra fat added to foods.

- Maintain a reasonable body weight

A moderate weight loss of 10-20 lbs is generally beneficial for blood glucose control. In other words a small weight loss can provide large returns in terms of blood glucose.

To lose weight one needs to watch portion sizes - using measuring cups and spoons to measure foods and beverages for a week or so to be sure one is eating reasonably sized portions. Additionally, regular physical activity - 30 minutes per day - will help with weight loss as well as improve blood glucose control

2.2 Prototype Design

2.2.1 Overall view and concept

The prototype is developed to assess the system architecture proposed in part 1, applied for the diabetics' case. This part describes the technologies actually used for the different parts of the system.

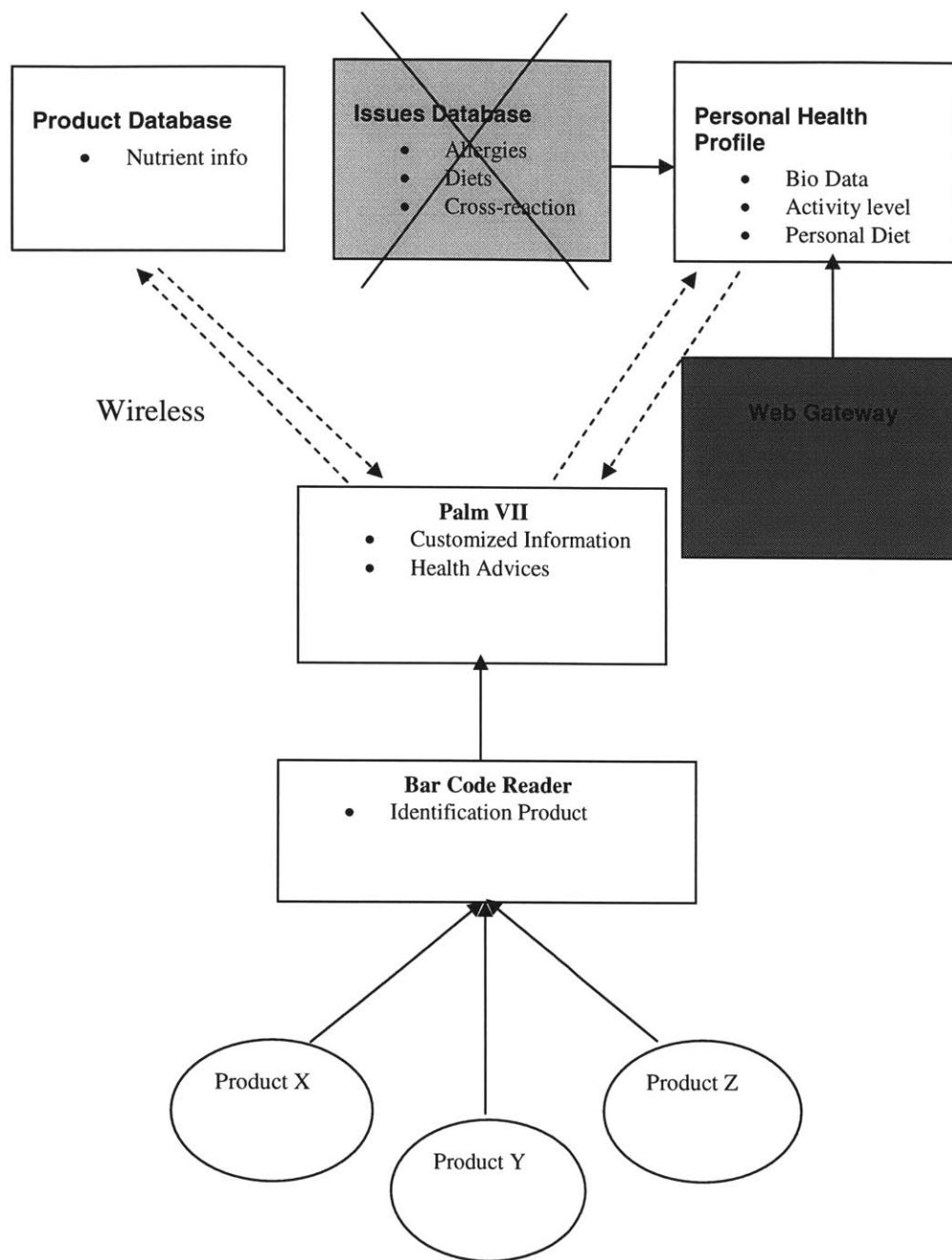


Figure 2: system architecture applied to the diabetes case

The scheme in figure 2 shows how the general architecture described in the introduction is adapted for the Diabetics prototype. First the Issues database is not implemented yet, because we only focus a single condition, Diabetes. Secondly, a web gateway (see below) has been added as an additional means (besides the handheld device) to interact with the profile database.

2.2.2 Components: functions and technology

- **Product Database for the diabetics application**

For the purpose of the field evaluation, we decided to enter manually 70 products in a custom database, inspired from the structure proposed by the US Food and Drug Administration (USFDA), by browsing through the aisles of a local grocery store. The database contains the information required for the diabetes application: UPC barcode, brand name, serving size and amount of seven nutrients involved in the diabetics diet. The structure is given in detail in the Appendix C.

- **Personal Health profile database**

The personal information required in order to provide the user with customized advice and recommendation is stored on this database. It contains basic data on the user (age, sex, weight, height, and level of activity) and nutrient daily intakes recommendations. This information is either entered manually or generated by an algorithm described in the following.

The technology currently adopted for the databases is a MySQL server. MySQL is a fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. It is free software and is licensed with the GNU GENERAL PUBLIC LICENSE. This server is accessible from different platforms. A web gateway allows the user, or any authorized person to update his personal information from any web browser.

- **The online profile gateway**

PERSONAL PROFILE

On this page, you will input some information, and thus update your personal diet guidelines. The following ratio are combined with your personal information by the system to recommend a Carbohydrates intake for your breakfast

Personal Info

	Current Value	New Value
Gender	Female	Select ▾
Wheight	150	<input type="text"/>
Activity	Level 3	Select ▾

Basic ratios

	Current Value	New Value
Percentage of carbohydrate in the total calone intake (recom: 50%)	50 %	<input type="text"/>
Percentage of your daily needs taken during the breakfast Meal (recom: 25%)	25 %	<input type="text"/>

Done Internet

Figure 3: web access to the profile

The online profile (see figure 3) gateway provides the user or any authorized person with simple access to his personal profile database. The information entered through the interface gives to the system the basic knowledge to personalize the diabetics diet to the user's specific needs. He inputs his Gender, Weight and Activity level. The activity level is measured on a numeric scale from 1 to 5, with the following convention:

- **5:** I get no regular exercise, and am not able to perform most daily activities.
- **4:** I perform usual daily activities, but get no real regular exercise.
- **3:** I participate in aerobic activity for 30 minutes 3-4 times per week.
- **2:** I participate in aerobic activity for 30-60 minutes, 5-7 times per week
- **1:** I participate in aerobic activity for 60-90 minutes, 6-7 times a week.

In addition to personal activity information, the user is asked to provide two common ratios involved in a diabetic diet. The percentage of carbohydrate in the total daily calories intake is generally recommended around 50%, but the user can customized the value in accordance with his doctor. The other ratio represents the relative share of the breakfast meal in the daily diet. This value has been added for the particular needs of the field test described below.

Based on these on the five mentioned values manually input by the user, the system dynamically generates personal diet guidelines and stores them in the health profile database. The “rules” used to personalized the diet are based on common diabetic diet recommendations found in numerous references and have been validated by a medical professional. The simple algorithm is explained in the following:

1. First determine your desired weight in pounds:

a. If you are overweight, a 10% loss from your current weight is ideal.

$$\text{desired weight} = \text{_____}\#$$

b. Or base your ideal weight on this formula:

$$\text{Women: } 100\# + (5 \times \text{_____}) = \text{_____}\#$$

inches over 5 ft. goal wt

$$\text{Men: } 106\# + (6 \times \text{_____}) = \text{_____}\#$$

inches over 5 ft. goal wt

Use this weight if you have an average frame.

- for a light frame, subtract 10% = _____#
- for a heavy frame, add 10% = _____#

2. Choose an activity level. A calorie factor is assigned to each level

	men	women
(5) Very sedentary	13	11.5
(4) Sedentary	14	12.5
(3) Moderately active	15	13.5
(2) Active	16	14.5
(1) Super active	17	15.5

3. The total daily calorie need is determined from the 2 previous values:

Daily Calorie need = desired Weight * Calorie Factor

The last parameter to be input by the user is the carbohydrate factor, or the ratio of carbohydrate in the total calorie intake. By default it is 50%, which is the commonly accepted value for average diabetic diets.

4. Input your carbohydrate factor:

Calorie Intake * Carbohydrate Factor = Carbohydrate intake

$$\begin{array}{ccccccc} \underline{\hspace{2cm}} & \# & \times & \underline{\hspace{2cm}} & = & \underline{\hspace{2cm}} \\ \text{Cal/day} & & & \text{carb. Factor} & & \text{carb/day} \end{array}$$

Knowing that each carbohydrate calorie weighed 4 grams, we obtain the total amount of carbohydrate needed per day:

5. Daily carbohydrates need

Carbohydrate calorie intake / 4 = Carbohydrate intake in grams

$$\begin{array}{ccc} \underline{\hspace{2cm}} & / 4 = & \underline{\hspace{2cm}} \\ \text{Carb/day} & & \text{grams CHO/day} \end{array}$$

6. Relative size of each meal

The total daily carbohydrate intake is split into the different meals of the day.

Breakfast grams

AM snack grams

Lunch _____ grams
 PM snack _____ grams
 Dinner _____ grams
 Eve. Snack _____ grams

Example:

60 Carbohydrates at breakfast
 15 Carbohydrates at morning snack
 60 Carbohydrates at lunch
 45 Carbohydrates at afternoon snack
 60 Carbohydrates at dinner
 45 Carbohydrates at bedtime snack

From the daily caloric intake, one can also estimate the desired fat intake, given that 1 gram of fat provides 9 calories.

Fat intake should be 30% or less of caloric intake.

7. Daily fat intake

$$\frac{\text{_____}}{\text{cal/day}} \times 30\% / 9 = \text{_____}$$

grams total Fat/day

Moreover, less than 10% of the daily caloric intake should result from saturated fat.

8. Daily Saturated Fat intake

$$\frac{\text{_____}}{\text{cal/day}} \times 10\% / 9 = \text{_____}$$

grams Sat Fat/day

Finally diabetics need to monitor their intake of cholesterol and sodium.

Dietary cholesterol should be limited to 300 mg or less daily.

2,400 mg or less per day of sodium is recommended.

The simple algorithm described above represents the basis for the diet customization offered for diabetics. The algorithm itself is embedded in the web page source⁶, so that the diet recommendations are generated dynamically. Once the user submits his personal data, the web page displays the corresponding recommendation, and allows him to manually adjust the values, as shown in Figure 4.

The screenshot shows a web browser window with a page titled "Personal Recommendations". It contains two tables and a "Submit" button.

Recommended daily Calones Intake	2025 cal	
Daily Carbohydrates intake	253 g	
Daily Fat intake	67 g	
Daily saturated Fat intake	22 g	
Daily Cholesterol intake	300 mg	
Daily Sodium intake	2400 mg	
Daily fiber intake	30 g	

Nutrient	Current Limit	New Value
Saturated Fat	25 g	<input type="text"/>
Cholesterol	180 mg	<input type="text"/>
Sodium	3500 mg	<input type="text"/>
Dietary Fiber	80 g	<input type="text"/>
Calories	2025 cal	<input type="text"/>
Carbohydrate	253 g	<input type="text"/>
Total fat	80 g	<input type="text"/>

Figure 4: recommendation part on the web profile

Note on the scalability of the prototype:

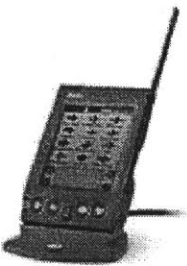
At the beginning of this part, and because we focused specifically on diabetes, we mentioned that we didn't develop the **Issues Database**. As a result of this, the algorithm to build the personal profile was directly embedded in the webpage. In a broader application, the issues database could store this algorithm along with the ones related to other issues. Each entry of this database would consist in a health issue (Diabetes, Peanut Allergy,...) with the list of questions that should be asked to the patient (for Type II diabetics we limited ourselves to simple bio-data)

and the rules or algorithm that computes personal recommendations from the answer to those questions. More details about how we can expand the system are given in Part 2.4.2.

The client platform

The client platform is composed of the Palm VII with a pen-like barcode scanner attached to it. In this part we briefly describe the technologies used and developed, then explain the the functions of the embedded software.

The handheld device

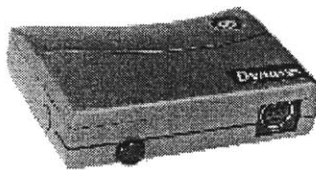


**Figure 5: 3Com
Palm VII**

can be found on the model's white paper web page⁷.

The handheld device is a standard portable digital assistant (PDA) from Palm, Inc. The model used, a Palm VII has a wireless internet access. In exchange of a monthly fee, a subscriber has access from his PDA to the web through the palm.net service. Besides that, the Palm VII has a serial port which allows one to virtually connect any device that support the standard serial communication protocol. Technical details on the palm VII

The Dynasys Copilot Interface

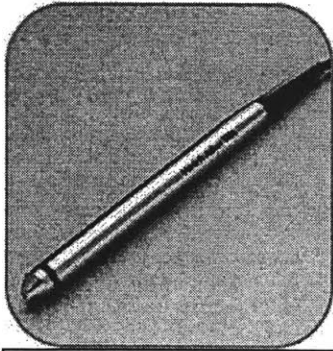


**Figure 6 : Dynasys
Copilot**

Palm port.

The equipment was chosen as an interface between the Palm VII and a bar code scanner described below. It contains additional battery and powers the scanner that is connected to the serial port of the PDA. The copilot is only used as a convenient physical hardware component that allows plugging standard serial components on the

The Barcode scanner



**Figure 7: Scanteamæ
6189**

As seen above, the solution chosen so far to identify the products is the standard bar code (UPC) displayed on every packaged food product in the US. The main advantage of it is that it is already adopted by the manufacturers, and databases with the UPC codes could be available for a large scale deployment.

The selected acquisition device is a pen-like laser scanner. This type was chosen because it allows the device to have a double usage. In addition to its primary function, the pen-scanner can serve as the pointing device for the Palm VII touch screen.

The SCANTEAMÆ 6180 is Welch Allyn's bar code wand reader/decoder that simplifies scanning and decoding by combining both functions into its standard wand housing. The device output is a decoded bar code data in Serial ASCII format and is sent to the Palm serial port through the Dynasys copilot interface.

Software

The software that runs on the Palm Pilot is written in WABA. WABA is a programming platform for small devices. WABA defines a language, a virtual machine, a class file format and a set of foundation classes. The syntax of the WABA programming language is a strict subset of the syntax of the Java language. However, WABA is not a derivative of Java and has no connection to Sun Microsystems, the owner of the Java brand and related trademarks.

A strong advantage of the WABA language is that the software written is portable to most handheld computers that run the WABA Virtual Machine, without any modification. Moreover, The WABA language, virtual machine and class file format were designed to be optimal for small devices. Certain classes allow a direct access to the Palm native functions, which is trickier under JAVA.

At a conceptual level the software acts as a gateway between the user, the food product, and the set of databases stored remotely. It basically enables the device to acquire bar codes and send them to the remote web server through the Palm wireless protocol, along with a specific request, and then displayed the results of this request.

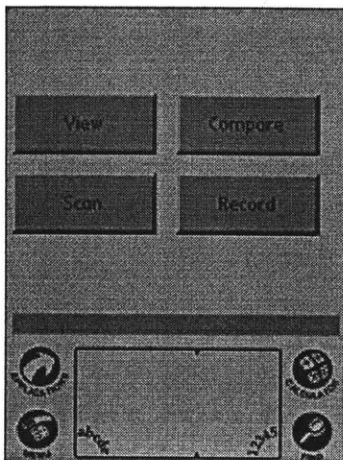


Figure 8 : Welcome menu screen

The central feature is the convenient user interface. The user picks up the functions he would like to use by pushing buttons on the graphical interface with the scanner wand. The information is then displayed in a way that takes account of the limitation on the Palm VII interface, a small monochrome touch screen. As he turn on the handheld computer, the user sees the main menu of the software, composed of four modules to be used in specific situations, each of these corresponding to certain points of decision. The personal advisor helps the user to purchase adequate food when browsing through the aisles of a grocery store, and to plan meals in compliance with his/her personal diet.

Figure 12 gives an overview of the function performed by the four modules (View, Compare, Scan and Record). They were developed in the context of the scenario described in the next part.

- **Scan**

In this mode the user scans a product barcode and obtains personalized advice or information on it. The advice given by the system is displayed of two steps: The user receives first a simple and verbal note on the product, which can be sufficient to make the purchase decision (like “this product is too rich for you in cholesterol”). Along with this advice, the user is offered a better alternative, which is a product from the same food category that better complies with his diet. The rules used to generate this advice are discussed in the next part. The user can obtain a complete quantitative analysis of the product, where her/his personal diet is used as a reference, by pressing a “details” button. Examples of screenshots for the three steps described are given in Figure 9.



Figure 9: Screenshots of the results displayed of the Palm device

Given this information, the user may decide to “buy” it, thus logging the purchase and recording it in his “personal shelves”. The “personal shelves” database contains the food inventory of the user or the complete lists of products available at home.

- **Compare**

The user chooses a specific nutrient (Carbohydrate, Cholesterol...) and then scans an arbitrary number of products. He receives the corresponding list of products ordered in

increasing or decreasing amount of the specified nutrient. This allows quick and focused comparison between candidate products.

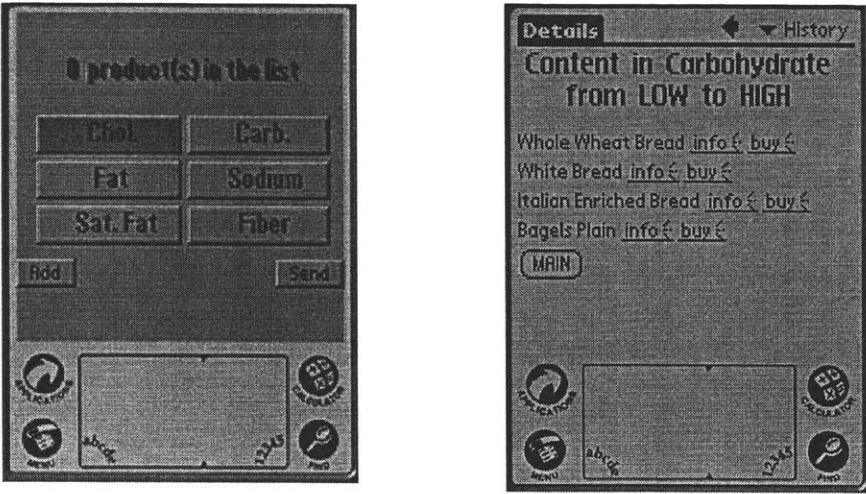


Figure 10: Compare function menu and example of result

- Review

The review mode is composed of two functions. The user can either access her/his inventory (or “personal shelves”) or access the synthesis of her meal plan.

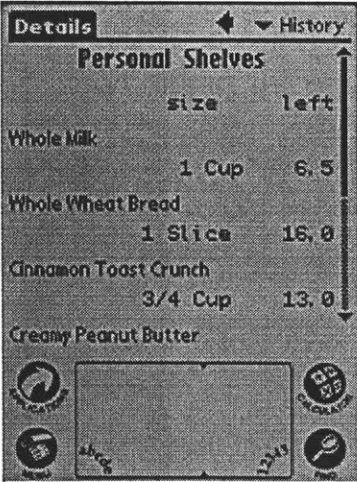


Figure 11: review function result

At the beginning the “personal shelves” are empty, and the user updates this database by pushing “buy” while being in the Scan or Compare modes.

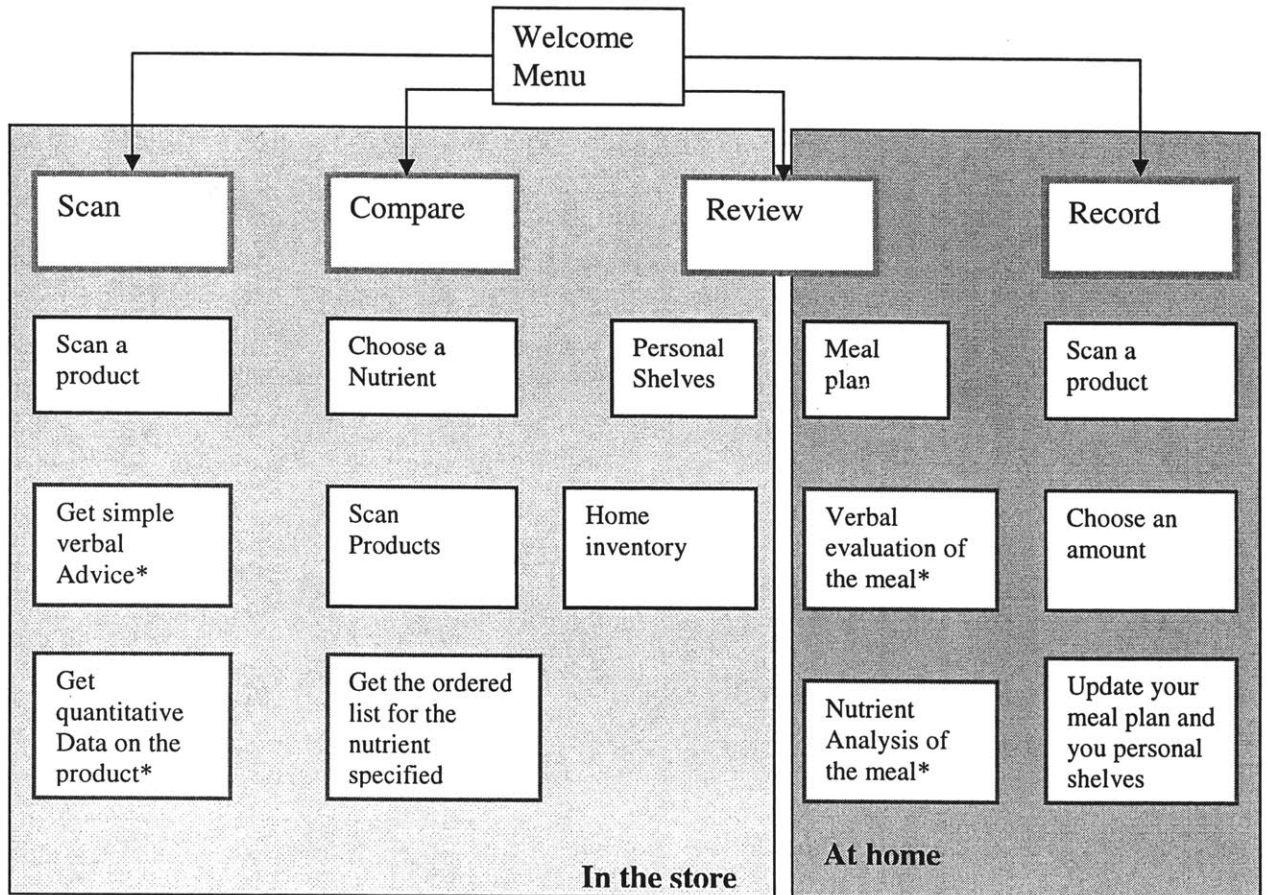
The meal plan is itself broken into 2 two parts: A summary of the analysis of the planned meal is produced, and further details are available on demand. This analysis is obtained by cross-checking two databases: the online profile input at the beginning of the experiment, and the list of products updated with the “record” mode.

- **Record**

This function is used to set up the meal plan in the scenario described below. In this mode, the user scans a product and specifies a number of servings. This action is to be performed each time the user decides to include an item in the meal plan. The information is used to produce a meal plan analysis in the review module.

- Summary of the software features

Application for Diabetics



*The advice and quantitative results are generated by cross-checking the user profile and the products database.

Figure 12: Summary of the Palm software features

Figure 12 gives an overview of the different functions of the embedded software, along with the context of where they can be used. The modules are designed for particular applications (at the store or at home), with the exception of the “review” part that can be useful at both areas. It also shows the two-step advice delivery approach: the idea is to give a simple and “easy to

follow” verbal advice whenever possible, with quantitative details available on demand to the user to enforce trust in the system. A task analysis of the scenario developed for the experiment is given in Appendix B.

2.2.3 The rules system

The software features have been described so far from the user’s perspective. It was mentioned that the user received different levels of advice, based on the information stored in his personal profile and on the products nutrient data. This part gives more details on how the elements of advice are actually generated, the algorithm and the different outputs.

When the user scans a product, the unique barcode is sent to the server to retrieve the product name and its category, along with its full nutrient information: servings size, and amount per serving of the nutrients relevant to the diabetics diet: calories, carbohydrates, total and saturated fat, sodium and cholesterol. The diet of the user, constituted in the default by recommended daily limits for the corresponding nutrients, and stored in the Personal Health Profile database is also retrieved from the server. The amounts of the nutrients contained in one serving are translated in term of percentage of the user’s personal daily value. Simple verbal advice is assigned to the different ranges of values obtained, as shown in Figure 13.

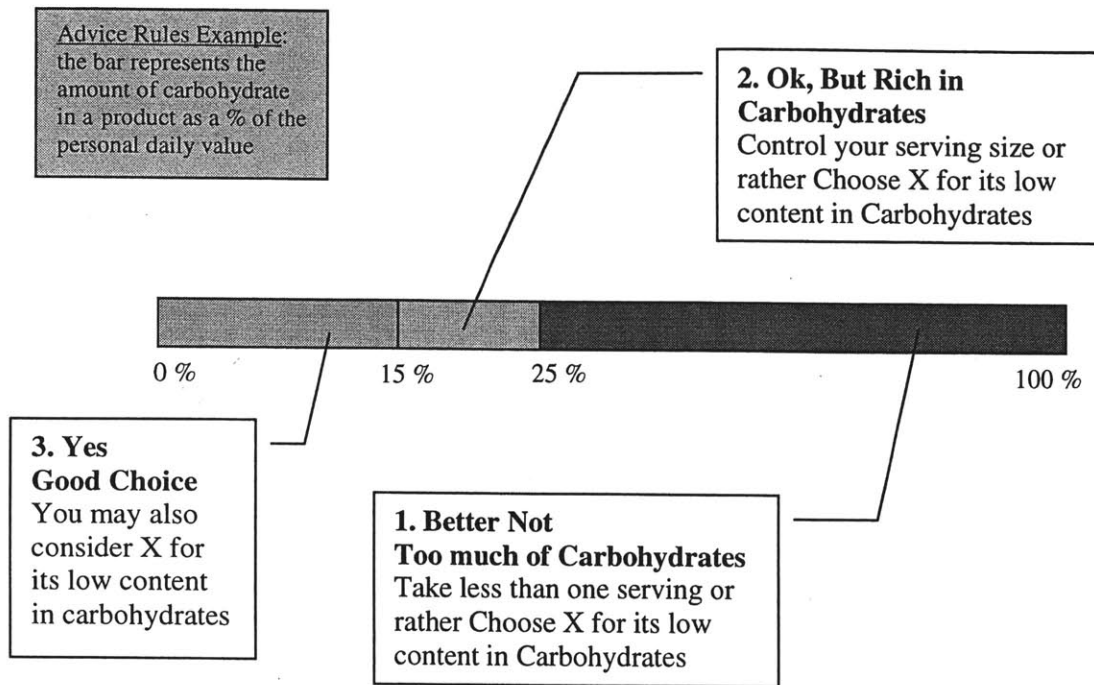


Figure 13: Rules system for the diabetes breakfast application

The thresholds (15%, 25%) were set in the context of the scenario described in Part 2.3.2. It is reasonable to consider that any nutrient intake in a single meal shouldn't account for more than 25% of the personal daily value. As a consequence, the purchase of a product that contains more than this value in one serving size for any nutrient is not recommended by the system (see box 3 in Figure 13). Moreover, we set a "buffer region" between 15% and 25% of the daily value. It is supposed to draw the attention of the user to the fact that a product is rich in a particular nutrient. In that case, the user is warned that he should either carefully control the serving size, or consider the alternative given by the system (see box 2 Figure 13). The alternative is a food product from the same category (cereals, dressing, juice,...) that better complies with the user's diet. The meaning of the advice ("Ok, But") in the "buffer region" is intentionally not perfectly clear. It is up to the user to interpret it as a "YES" or a "NO", and take a decision given other external factors. For example, the user can obtain the numerical values by pressing the "details" button to understand the advice better. One goal of the evaluation described later is to evaluate the variability in the advice's answer, and how one might foster a long term healthier behavior.

The third case occurs when the user scans a product that complies with his diet (see box 1 Figure 13). Along with a positive message, the user is also given a suggestion for a “healthy” product. We evaluated how people react to the “better-product” suggestion, and whether they were willing to follow it, even when their first choice was satisfactory diet-wise.

2.2.4 System View

The platform described so far has been developed to allow an evaluation through a real-life scenario. The handheld device is used at two major points of decision: when viewing the shelves, when the diabetic is wondering whether a product would be adequate for his personal diet, and at home when he is preparing his meal and should adjust the mix of product into a healthy and balanced meal. A system approach of the problem gives a clear view the problem and the role of the Smart Advisor as a flow regulator.

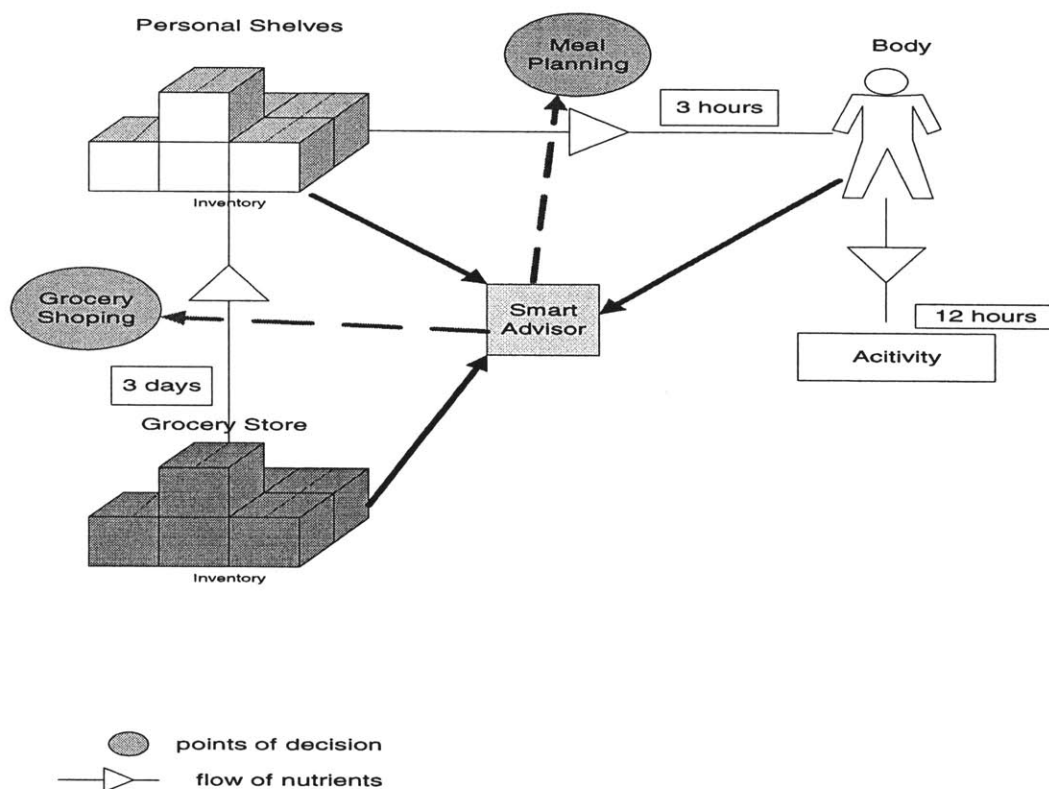


Figure 14: “Flow regulator” representation of the system

Hence, in some aspects the ensemble Grocery Store + Personal Shelves + Human Body can be viewed as an open system where flow of nutrients and information occur.

Each subsystem has its own time constants, constraints and feedback loops. Each subsystem is or should be in a balanced state, meaning that its nutrient content belongs to an acceptable range. To hold the balance of the first two subsystems is mainly a problem of logistics, at the retail store or at the household level. Similarly, a healthy diet leads to a dynamic balance of the human body subsystem.

In this view, the handheld device plays the role of nutrient flow regulator. By integrating data from each of the subsystems (product information, personal inventory and health profile) and making it available at both points of decision, the platform allows the user to control his diet in two steps, when the product enters his personal inventory (Grocery shopping) and when he plans his meal (meal planning).

2.3 Experiment

2.3.1 The P&G facilities

The prototype has been evaluated during a field test run at a Research Laboratory of Procter & Gamble, the major project sponsor. The Future Home Lab facilities offer an ideal environment and set of expertise to assess the system in a real-life simulation. It is composed of two distinct parts that were both used for this experiment.

The “home” part is set up as a fully functional house, where all domestic tasks can be conducted. The rooms are filled with prototypes and commercial high-tech products that are more or less integrated among common objects. The concept is to have these technologies utilized by consumers in real-life situations in order to observe their behavior and get some valuable feedback. To make the panelists more comfortable, the recording devices including video cameras are hidden. These give the opportunity for researchers to observe the scene without the knowledge of the panelists, thus limiting the disturbance.

The other part of the P&G Lab simulates a medium-size grocery store and contains a very conventional set up of cash-registers, shopping carts and product shelves. Speakers reproduce the noise and background music of real stores. The cameras in the apparent video security system are used to analyze every motion of the panelists during their shopping experience.

The panelists thus tend to adopt a natural behavior during the experiment.

2.3.2 Field Test

The objective of the field test was to have a potential user utilize the different modules of the systems, the online profile and the handheld platform, in order to get some insights and make improvements. The test was setup with the assistance of a team at the Future Home Lab. The setup and the professionalism of the moderators who conducted the interviews with the recruited panelists contributed to create a comfortable atmosphere. The staff from Procter & Gamble recruited eight Type II diabetic participants. They were type II diabetics, both male and female, age 25-60, of different economic and education levels, and the primary grocery shopper for their households. Additionally, participants believed that managing and controlling their diet was extremely or very important and extremely or very difficult.

The goals of the experiment were stated as follow:

Assess overall appeal and reactions to MIT Age Lab's wireless personal advisor among diet-restricted, Type II diabetics.

Understand consumer perceptions of the personal advisor's usability and functionality for food purchase decisions, meal planning and diet maintenance.

Assess consumers' ability to learn how to use the device and navigate through the different modes of the program.

A key constraint of this experience was that we wanted it as realistic as possible. Because the evaluation was also time limited (the Facilities being available for a couple of days), it was decided to focus on the Breakfast meal shopping and planning. The sessions were broken into three parts that took place in three areas of the facility: the living-room, the grocery store, and the kitchen. Each panelist was given some explanations on the system and asked to update his online profile; then he was asked to purchase breakfast food in the simulated shop environment; finally he had to plan his meal in the kitchen. The following gives details on these sequences, showing how the prototype was used in each one.

The living-room: review of the system and online profile (25 min)

A preliminary discussion (5 min) with the participant was first conducted to get some basic information about his diabetes and the way he manages his strict diet. The following questions were addressed in an informal way during the conversation that took place in the living-room.

- What kind of health issues do you face? *Probe for diabetes if not mentioned.* When diagnosed?
- Where did you go for information about diabetes and nutritional guidelines? *Probe for MD, Internet, friends etc.*
- How often do you consult your doctor or update your diet? What are the criteria used (Weight, Blood sugar level...)?
- To what extent does your level of activity has impact on your daily diet? How do you define or measure your level of activity?
- How difficult is it for you to plan, purchase and stay on your nutritional plan?
- What methods do you use to stay on track? Daily, weekly, monthly? How do you take into account when you “splurge” and eat foods not necessarily on your plan?

The participant was then given a short presentation on the PDA, scanner and software system of the Smart Advisor. The moderator explained the schedule and the different phases of the test (20 min). Limited information was given on the system to the participant:

“The Smart Advisor gives advice and direction at two major decision points: When the user purchases food, and when he plans his meal. The system provides a platform that combines data on the user’s overall health profile, the user’s level of activity, and the nutritional content of products.”

The user logged on his personal profile webpage (see Part 2.2.2) and updated his personal profile. Given the simplicity of the online profile, the accuracy of the data input by the user was not critical. Indeed, this task was performed to address two principal issues. First it allowed us to evaluate if the user is willing to provide personal health information on a website, and what are his main concerns about it (privacy, usage of the data...). Moreover, it had a pedagogic purpose, as it should convince the user that the advice and recommendations given later in the experiment are specifically tailored to his personal needs. The process followed to fill the profile was rigorously identical to the approach detailed in the description of the online Profile gateway.

Finally the user was given a short training on how to use the Palm VI and to navigate through the different modes of the software, and how to acquire the barcode/UPC with the scanner pen. This phase gave interesting insights about the limits of the prototype usability that are described along with the results of the experiment. The features of the 4 modes of the software are explained, and suggestions about when to use them during the experiment are given. However, the participant was relatively free to choose which function to use.

The grocery store: shopping (30 min)

The participant was then led to the simulated grocery store environment. The task consisted in shopping for breakfast products. An aisle of the store was filled with 65 products organized in 8 categories: bread, cereals, juices, milk, butter, preparation (muffin/pancakes/waffles), topping, yoghurt. The focus on a particular meal was made to achieve a practical trade-off. We wanted to have the choice within a category relatively broad, while keeping the total number of products limited, because the products database was entered manually and the experiment was time-limited. We hoped to elicit realistic behavior where the user had enough alternatives for a particular kind of products, so that he considered what he really liked: the way he resolved conflicts between his taste and any advice based on health issues was hard to assess otherwise.

To keep the experience as natural as possible, little guidance was given to the participant, except that he was invited to share his comments, concerns and criticisms while performing the exercise. These are combined with observations on the participant behavior in the next part.

The kitchen: meal planning (30 min)

The final task was conducted in the kitchen back in the Future Home Lab and consisted in planning a healthy breakfast. The participant picked products and specified the amount that he would take for a regular breakfast by using the record function on the handheld device (see Part 2.2.2). After accessing the analysis of their choice through the review mode, subjects were free to update it and try to correct the issues or not. The primary goal was not to obtain a “healthy” breakfast plan, but rather to observe the participant, and understand why or why not he was willing to comply with the advice. The following questions were addressed to the participant after he performed the tasks.

1. To what extent did you comply with the system’s advice? Why?

2. In what conditions would you rely on the system and change some of the choices you normally make?
3. In what conditions would you ignore the advice of the system, and remain with your standard choice?
4. Who should be in charge for the system or should sponsor it? In other words from whom would you accept health related advice when purchasing food?
5. How do you think the device will impact the amount of time you spend reviewing new or unknown products, by focusing only on the nutrients that concerns you?
6. How useful would be the comparison function to help you choose between different unknown products?
7. How do you think using the system would impact the amount of time you spend in the store? Planning meals?
8. How do you value the personalization of the advice (the fact that it uses some of your personal information to give you accurate recommendations)? How do you perceive the privacy issues in this case?
9. How would you incorporate this system into your daily life?

3 Analysis and Results

The results from the experiment are summarized in 2 stages. A first part reports the direct comments and suggestions from the type II diabetic participants. From these comments and direct observations we will then summarize the learning and the improvements to be made on the prototype.

3.1 User Feedback

3.1.1 Behavior

One need to better understand the behavior of type II diabetics, in order to have them adopt healthier habits. Several questions were asked to the participants about the way they manage their condition.

- Diabetes diagnostic

Most respondents indicated they were shocked when they first found out they had Type II Diabetes. Several respondents mentioned they went into their doctors because they experienced symptoms of dizziness, blurred vision and/or numbing. They mentioned they expected to hear a diagnosis associated with heart disease or stroke, but were somewhat surprised to learn of their diabetes, largely because they were not familiar with the disease, or because, as one respondent said, *“It did not run in the family.”*

- Knowledge of Diabetes

While respondents were aware of the negative side effects of diabetes, they did not appear to understand the specifics of their condition. Even though many of the respondents had been dealing with their Type II Diabetes for 10+ years, they did not appear to be aware of the relationship of insulin to sugar levels and their inability to metabolize carbohydrates. Rather, they knew that they *“had high levels of sugar in their blood”* or as one man referred to it, *“I got sugar.”* They understood that if they did not keep their blood sugar levels regulated through diet and exercise, they could face consequences of kidney failure, amputation, blindness or stroke, but they hadn't a clear vision of the dynamics.

- Diabetics Diet

Respondents indicated that managing and controlling their diabetes is extremely important to them. Nevertheless, on an overall level, respondents did not appear to take extreme measures in their efforts to control their condition. Respondents understand their weight plays a big part in their diabetes, as does the amount and types of food they consume. However, very few of the respondents were happy with their weight and only a few of them reported having a regular exercise program. Also, they mentioned that they do not really stick to a well-defined meal plan because they do not truly understand which food choices are the most appropriate for them. Rather, they monitor their blood sugar levels and take corresponding action if their levels are high. Besides, many respondents indicated that even if they did understand that they were not making a proper food choice, they would still compensate for their 'craving foods' by *“only eating a small portion,”* or *“eating better”* at the preceding or following meals.

- Sources of information and advice

Respondents mentioned that their doctor is their primary source of information regarding their diabetes, but they often read magazine articles for supplemental information. The majority of respondents noted that they visit their doctors every three months, and several mentioned they bring in their daily blood sugar measurements for the past three months for the doctor to review. Some respondents indicated that they had a good working relationship with their doctor regarding their diabetes, while others felt their doctors “*did not care*” and did not take the time to fully explain their condition and ways to manage it.

3.1.2 Hardware and user interface

Inputting data on the website to create their personal profile was logistically very easy for respondents; however, they may not have been able to accurately complete the profile without their doctor’s assistance. Most respondents were familiar with computers and could easily maneuver through the Personal Profile screen. The personal information was easily understood. However, the ratios were somewhat confusing, especially the ratio of Total Carbohydrates in a daily diet. The default ratio recommendations would initially be used by consumers, then perhaps modified based on conversation with their doctors. Additionally, after the Personal Profile information is added, the information should default to the Personal Recommendations on both of the following grids, not just the red Recommendations grid.

The handheld device was appealing to these respondents. While respondents no were PDA users, many had heard of the devices and were very interested in using one. They were comfortable with the touch screen and activating the wireless connection. Although some respondents mentioned they would envision the device getting smaller and easier to handle, they were largely referring to the bulky scan wand connection.

Respondents recognized that scanning a particular item was a good way to ensure they were accurately inputting the correct item, however, they had difficulty scanning items that had plastic outer wrap or had surfaces that were not flat and smooth. They were somewhat forgiving of the inconsistency of the tool, because they felt they “*would get better with practice.*” However, if the efficiency of the operation did not improve significantly over time, respondents could envision themselves being frustrated with the device, and potentially not scanning certain items, or even discontinuing use of the system.

The respondents criticized the lack of navigability of the handheld software. The mistakes could not easily be corrected, and they often had to rescan a product while switching from one function to another one. This major design flaw is partly due to the fact that no information is stored on the palm, even temporarily. Further explanations about this issue and how to address it are given in the next part.

Respondents felt that the meal planning tool was appropriate, as long as they ate foods that were easy to record as single items. They observed that making dishes that required, "*a pinch of this, or a ½ cup of that*" would be difficult and time-consuming to enter into the system.

Overall, the system interface appeared to be relatively simple for respondents to handle. Although respondents made some mistakes or needed reassurance during the actual shopping trip, they felt very comfortable with the device and felt that with some degree of practice they would easily be able to maneuver through the system.

Respondents felt shopping with the device would be relatively easy, and interestingly, many respondents felt the device would actually save them time in the grocery store. Because these respondents experience a certain degree of difficulty deciding what to eat, they often spend an inordinate amount of time in the grocery store reading labels and evaluating alternatives. For example, a few respondents said they spend two or more hours in the store every week, and others said they go several times per week. They felt that by using the device, they would easily be able to determine the appropriate choices for them, and the Smart Advisor would eliminate their struggles, and they would quit wasting time choosing appropriate items.

However, respondents appeared to have difficulty holding the device and shopping at the same time. Upon probing, several respondents mentioned it would be easier if they did not have to occupy both hands with the device, one hand on the PDA and the other on the scanner. One respondent said, "*It would be nice if this were hooked on to the cart so I could have a hand free to grab another product.*"

Several respondents mentioned that they may not take the device with them every single time they went to the store, but they would easily use it on large shopping trips. Respondents mentioned that in their daily routines, they often make quick stops into the store to pick up necessary items, or to "*just pick up something for dinner.*" They mentioned that they might not

use the device on these trips, but that it would be very easy to integrate into their larger shopping trips.

Respondents were unsure of how they would measure and keep track products, and other items that were **not easy to scan**. They know that fruits and vegetables are a very important part of managing their condition; however, it did not seem easy for them to include these items in their meal plan. They suggested having default “buttons” they could use to add an apple, banana, or carrot sticks to their menu

3.1.3 Features

Among Type II diabetics, initial appeal for the Smart Advisor was high. The respondents were very enthusiastic about a single, definitive source of information that could help them manage their diabetes, which they felt they could easily integrate with their daily routine. These consumers are very confused with the information they are provided about the nutritional value of foods, and one respondent summed it up, *“They keep changing their minds about what is good and what’s bad for you. One day something is great, and the next day it’s bad for you.”* They felt that a major benefit of the Smart Advisor was that they could use it as a single source for information, and liked it because it was small enough that they could carry it with them wherever they went.

Respondents were enthusiastic about having the ability to customize the dietary recommendations for themselves, rather than simply using generic guidelines provided on packages. They had significant interest in receiving customized nutritional information, because as one respondent mentioned, *“What’s good for someone else, could be bad for me. I like having information that’s totally about me.”* Respondents recognized that their weight and activity levels could impact their individual nutritional needs. Therefore, they had difficulty interpreting and assimilating the FDA-required “Nutrition Facts” into their eating habits. They felt that by having customized information, they could more easily manage their diabetes, because they could make more informed decisions about what, and what not to eat.

Respondents were comfortable entering personal data and transmitting it over the Internet, as long as they worked with a user ID and password. Even though these respondents were not very computer savvy, they were comfortable entering the required information and making it

available on the Internet. They were confident that the user ID and Password required would sufficiently guard them against any Internet fraud, and as one man added, *“Besides, it’s not like I’m putting my credit card or Social Security number on there.”*

A few respondents noted that they would like to be able to use the device for other people, as well as themselves. These respondents mentioned they had spouses or parents they cared for at home, and would like to be able to use the device not only for themselves, but also for other people they care for. For example, one woman said, *“My husband has high blood pressure and has special needs for his diet. It would be great if I could use this to check for my foods and his at the same time.”*

The “Scan” function was viewed as an appropriate way to obtain product information; however, many respondents wanted improvements in the functionality of the scan tool.

The “Compare” function was very appealing to respondents. They liked having the ability to evaluate separate items on various nutritional attributes. This tool gave them reassurance that they could choose the best of several alternative food choices, based on rankings on a specific attribute. Although it was not intuitive to respondents that they could compare items from different categories (e.g., cookies, brownies, and popsicles), they liked the idea of being able to universally compare items on specific attributes.

Respondents were often surprised at some of the responses they received from the Smart Advisor, and they liked having that information. Often, they found that a food choice they would typically have avoided was not as bad as they thought, or on the other hand, they found that an item they felt was a good choice had better alternatives. They very quickly became comfortable with the Smart Advisor’s advice because they felt it was customized for them, and therefore better for them than any of the prior information they had acted upon. One noted exception was a woman who selected butter, instead of margarine, which was rated a better choice for her using the “Compare” function. She rationalized, *“Even though margarine may be a better choice for me as far a total fat goes, I made the decision long ago that I would not use margarine, because my body knows what butter is and how to digest it. My body does not know what margarine is because there is nothing natural about it.”*

Respondents were visibly pleased when the Smart Advisor told them they made a “Good Choice” on their shopping trip. This positive reinforcement was perceived as a *“nice pat on the*

back” for doing a good job. Since none of the respondents felt they were always making the best choices, they took the praise as a sign of recognition for their accomplishments. This praise would ideally reinforce their decisions, and turn these behaviors into habits of good choices, and therefore, ultimately improve their ability to manage their condition.

For “craving foods” that they knew were not compliant with their diet, many respondents mentioned they would use the “Compare” function to evaluate several items so they could make the best choice , while others mentioned they may not even scan the item. Overall, if these consumers have a craving, or want a certain food, they will find ways to eat it. The Smart Advisor may assist them in making a better choice of something like cookies or sausage, but some respondents mentioned they would still eat cookies and sausage.

Some respondents offered improvements to the “Compare” function in that they would like to have the ability to compare items on two new attributes - “Sugar” and a “Best Overall” choice. Most of these respondents did not completely understand that Carbohydrates are made up of Sugars, Dietary Fiber and Other Carbohydrates. Therefore, they may automatically eliminate an item that may be high in Soluble Fiber because it is high in Carbohydrates, even though it may be a relatively good food choice, compared to other items. They felt that a better attribute on which to judge items might be Sugar content, because ultimately, that is their major concern with their diabetes. Other respondents felt it would be beneficial to have a “Best Overall” attribute, because as one respondent said, *“I don’t know that much about this stuff, but I do know that something that may be low fat could be high in sugar. So if I choose a low fat item, I could totally blow my sugar numbers for the day.”*

Respondents also mentioned that as they became more educated on selecting ‘proper” food choices, they may not scan every item. They noted that after a few trips, they would have used the compare function to evaluate the best choice in a certain category, and they would no longer have to hesitate when making a selection. They would, as one woman said, *“just pick up the item and throw it n the basket because I know it’s a good choice.”*

The “View” function was appealing to respondents in that they could review what they had purchased, and use this information to help them decide what to plan for meals. They liked being able to recognize the items they have in stock and use this to decide which foods might go

together. Additionally, it was interesting to many respondents that the device indicated how many servings of a particular item they had, and that it showed the serving size of each item.

The “Record” function was particularly appealing, in that respondents could easily plan a meal that complied with their customized nutrition plan. They liked the ability to customize the serving size of each individual item, because they often do not eat the indicated serving size on a product’s Nutrition Facts. Also, they appreciated the opportunity to be able to change their meal plan, if the initial plan was determined to be non-compliant with their guidelines.

Respondents felt that the addition of recipes, or meal plans would be a strong improvement to the system. They liked the idea of having “doctor-approved” recipes included with the system so that they could easily shop for and prepare meals that would help them manage their diabetes. In addition to saving time selecting these items, respondents felt they could save time by simply selecting an individual recipe or meal plan for a certain meal occasion, rather than scanning in each individual element of the recipe or meal plan.

Several respondents felt meal planning was an excellent idea to help them learn about appropriate meals to eat; however, they also noted that they might not use this function every time they ate. They mentioned that this function could take a long time to execute, especially for meals with many items in different serving sizes. Therefore, they would use the system to evaluate a meal once or twice, and use the advice to modify the recipe or meal plan as needed to make sure it complies with their guidelines. Then, they would most likely not use the system again for that particular meal plan. Additionally, respondents mentioned that they would not realistically always have time to use the system. They noted that in times when they were in a hurry, they would simply not use the system, and try to recall a meal plan that the system had approved in the past.

Respondents noted that in order to truly and accurately reflect their daily habits, the device should be able to account for common fast-food and restaurant selections. Respondents mentioned that they often eat away from home, and they would like ways to incorporate these occasions into their daily plans. They envisioned the functionality to include several fast food and common restaurant selections that they could get and compare against their dietary guidelines. Even though they realize fast food selections are not likely to be approved, they are realistic with themselves in knowing that they play a role in their lives, and they want to be able

to know how they can better manage these occasions by planning appropriately for other meals on the day when they eat away from home.

Respondents reacted favorably to the idea of the interface being a bit more like a “coach” that gave them a hearty “Good Job!” when they made choices that would help them, and “You can do Better!” when they made choices that would set them back on their quest to manage their diabetes. They mentioned using the system as a tool to help them do a better job in managing their condition, to give them positive reinforcement when they deserved it, and “*got on*” them when they needed to do better. In fact, one respondent mentioned she wanted to see the interface act like “*real person.*” She mentioned likening the interface to “Turbo Tax,” the tax planning software that essentially works like a personal interview, using conversational terms that are easy to follow.

While we only utilized the system for one meal occasion during the research, respondents noted that the system should be simple to use for an entire day’s meals and snacks, so they could easily keep track of their daily nutritional intake. To be truly useful, the system should integrate all the food choices for a given day. For example, the system should “remember” what was consumed for breakfast, lunch and snacks, and when the user plans for dinner. Additionally, this information should be easy to draw upon, rather than requiring the user to recall the past events.

3.2 Improvements

The same way we divided the comments from the users, the summary for improvements to make are treated separately for the **user interface** and the **applications features**. Lastly, we will point out the factors that would be critical to foster a long-term behavior change, and show that these insights are not specific to the diabetes application.

3.2.1 Interface Improvement Suggestions

- **Hardware issues**

The barcode technology was confirmed to be a technology of choice to quickly identify a product. However the scanner wand showed some limitations. With this technology the user has to manually scan a barcode, which was sometimes difficult when it was printed on soft and non-planar packages. Furthermore, the second role of the scanner wand, which was to permit

interactions with the Palm through the touch screen, was not successful either, the wand being described as too bulky for the task.

The use of a built-in scanner, embedded in the front side of the handheld device would solve these issues. Several companies offer these technologies, as part an integrated part of the PDA or a module that can be plugged in. This would also free the second hand, and make the scanning task much more easy.

A simple but important improvement would be to have a “docking station” on the shopping cart, where the PDA could be mounted, one that allows the user to have two hands free to shop and use the device. Alternatives for the hardware set-up have been explored by students from the Rhodes Island School of Design. In their model the platform is permanently attached to the shopping cart, thus allowing having a wider and color touch screen. A user would identify himself by swiping his personal card, which would give permission to the system to access his online profile.

- **Navigability**

Because they are not willing to take drastic measures to manage or control their diabetes, the interface needs to be very simple to use, intuitive, and should be easy to master in a short period of time.

There appeared to be several areas within the interface where the navigability was not optimal and could be improved (by reducing the total number of “clicks” required to perform an action), or made more intuitive (by making the word choices a little more user-friendly). I list below specific corrections that would reduce the number of actions required to perform a single function.

- To have reduce the number of scans, by implementing a single “scan click” shortcut, such as “scan to buy”, for products with which the user is already familiar.
- After scanning and purchasing an item, the user should be able to “Scan Another Item” rather than return to “Main Screen”
- The default serving size in the meal planning module should be set to “1 serving” instead of “0.5 serving”.

- In record mode, the user should have direct access to the meal plan, in order to check what items have been recorded up to that time.
- The user should have the ability to modify or erase each item directly from the Meal Plan view, rather than having to re-scan each alternate item.
- The user should be able to compare products for different nutrients without having to re-scan the items.
- User should have the ability to “Buy” the recommended product without having to scan it.

- **Quality of the user interface**

The users sometimes had difficulties to figure out what they were supposed to do at a given step. It was not often clear when the software expected them to scan a product, push a certain button, or wait for a wireless connection. Improvement of the interface by using color code or sound would make the user better “aware” of the state of the software.

A way to improve the user interface is to modify the existing language and make it more intuitive to consumers. Thus the titles of two of the modules have to be upgraded:

- Change “Record” to “Plan Meals”
- Change “Personal Shelves” to something more familiar to respondents, such as “Pantry,” “Cupboard” or “My Groceries.”

3.2.2 Feature Improvement suggestions

A strong limitation of the prototype is that it relies on the assumption that the user would scan every item he buys and even eats. Alternative ways to register food products in the system should be considered. The user should be able to enter food that is usually not associated with a barcode, like fruit or vegetable, or any non-packaged food. The task can be performed with a set of scroll-down menus, organized by categories of foods.

The meal planning part of the system is based on the daily cumulative nutrient intake of the user. It is not realistic to expect that the user would enter every items of his meal plan, be it with the barcode scanner or manually, only to obtain a judgment on his meal. Nevertheless, some advanced features might add value to this task, like healthy diet recipes. The system could suggest healthy and personalized recipes based on the ingredients that the user has at home. Provided that a “recipes” database is added to the system, this feature might not very difficult to implement.

The comparison feature was also very popular among the participants. However, an integrated comparison would be much more attractive to them. In the first prototype, it was only possible to rank an item based on its content for a single nutrient. It doesn't provide assistance when two criteria are conflicting: what should the user do if the product that contains the least carbohydrate is also the one that contains the most fat? A way to implement an integrated ranking system would be to have the user, or his doctor, set the degree of importance of each nutrient limit. Hence, the user would input a coefficient on a 1-10 scale (1= not so important, 10 = critical) along the daily nutrient limits. This allows one to generate an overall “score” for each product, by adding the relative nutrient intakes multiplied by their corresponding coefficient. In our example, the one with the lowest score would be the recommended product.

The features of the meal plan module were also limited because it didn't provide any suggestion on the change to make when a meal plan was not compliant: it only gives a judgment on the meal plan considered by the user. A valuable improvement would be to add the capacity of the system to generate healthy meal recipes based on the “personal inventory” of the user. This could be done by cross-referencing the personal inventory database with a recipe database. The recipes could be selected by the system using a grading system similar to the one described above.

An important part of the meals are taken out, and the users would certainly not be able to “scan” the ingredients of a dish ordered in a restaurant. A realistic solution would be to store the nutrient content of common restaurants and fast-foods selections.

To take account of the fact that often one person is shopping for an entire family, the system should provide a convenient way to manage multiple personal profiles.

3.2.3 How to Enforce Trust and Change the Behavior

So far we focus our efforts on how to improve the system in order to increase its value for the user, that is, to provide the best features and interface for a consumer who often lacks of information, knowledge, awareness, or structure at various points of decision. However it might be useful to recall here that it is only a means to serve the ultimate goal of the project, which is to foster a long-term behavior change. Building a system that offers accurate personal information and advice, with a user-friendly interface, does not guarantee the success of such a project.

The main challenge is not simply to let someone know what individual decisions would be better for his health, but to actually convince him, over the long run, to take these good decisions. The simulation run with type II diabetics confirmed the fact that even a well informed person would not necessarily comply with good advice. Comments like *“I know it is bad for me, but I can’t help it”* were made by the participants, and more generally they admit that they are ready to compromise with their diet from time to time. These comments and direct observations during the experiment helped to list system features that address the question of behavior change. The learning is certainly valuable in a much broader context than the diabetes application.

- Build trust

Perhaps the most critical point is to build a trust relationship between the user and the system. The system should be accurate and reliable, but it is useless if the user doesn’t believe it. One of the ways to build trust is to have the system give an explanation each bit of the advice. If the user is told by the system to avoid a product, he should also be able to retrieve a justification for this judgment. This way he understands the verdict and can really accept the decision. To have the user participate in the setup of his “online personal profile”, even with the assistance of a doctor when required, can enforce trust by teaching the user about the rule system that generates the advice. In the diabetics case the rules were simple quantitative criteria, numerical thresholds adjusted to the participant’s needs so they were easily understandable. In a more complex case, where several health issues are conflicting and the rules are less obvious, it would still be better to have them understood by the patient. I think that this is a key feature for a personal health decision system to be trusted. For example, when the advice is arguable, the user should at least be given some motivations for it. If not, he will simply not trust it anymore, or even suspect that the suggestions are made for marketing purpose, rather than for his personal health. In addition

to fostering the trust in the system, the explanations can be very valuable because they teach the users about how to manage their own conditions.

- Avoid boredom

It is a common observation that very strict diets are hard to follow. Some participants confirmed that they were bored with their strict and monotonous diet. A strict diet plan that eliminates all 'craving food' will not solve this problem. Users must be able to manage the occasions when they eat foods that lie outside their dietary guidelines, and the device should not severely admonish them for their "indiscretions." Likewise, the participant would certainly not comply with a personal advisor that triggers a red flag too easily and often. Thus it seems very important but challenging to carefully design the messages given to the user. The difficulty is that there is no shared agreement on what is considered to be strict. We mentioned that the participants can react very differently to the same warning. For example, "Ok, but..." was interpreted as a simple warning or a strong no by different participants and thus prompt for different actions. It might be possible to overcome this difficulty by adjusting the type of warnings to the user: This could be done during a trial period when the system adapts its message in order to maximize the rate of compliance. After this period the advice can become gradually more severe, and the user more willing to follow it. The idea is to avoid an initial rejection by the user, who doesn't want to be criticized for all his choices. If so, the advice would simply not be followed and the user would quickly cease to even carry the PDA with him.

- A "friendly" device

The smart advisor should be a "friendly" and easy to use device that users can integrate into their realistic dietary habits. The interface could speak to them almost in a "coaching" and conversational manner. Such a verbal interaction with the user, who is prompted with questions such as "how do you feel today?", "have you exercised?" might achieve this goal. During the experiment, the participants were pleased with positive feedback like "good choice". These kinds of positive messages would encourage the diabetics, and certainly promote their willingness to comply with their diets.

3.3 Perspectives beyond the diabetics case

3.3.1 Real-time physiological monitoring device

Adding biosensors would improve the system in by many ways. It allows us to provide a more accurate and real-time advice, and at the same reduce the number of tasks to be performed by the user, because such a device can upload data seamlessly, without any manual intervention. A biosensor is a device that incorporates a biological sensing element integrated with the signal transducer, to give a sensing system specific to a target compound. Usually a biosensor can be used by non-specialist operators, either continuously on-line or discretely as a throw away device. Transducers are the physical components of the sensor that respond to the products of the biosensing process and output the response in a form that can be amplified, stored, or displayed. We won't discuss details of the transduction technologies, but rather focus on examples of applications relevant to the project. Traditionally, the driving force for the development of biosensors has become mostly from the healthcare industry, where rapid diagnosis and treatment is seriously inhibited by the long time delay between sample taking and the results from a central laboratory. As such one of the most successful commercial biosensors, the ExacTechTM glucose device (now superseded by the PenTM and Companion 2TM) was designed for diabetics. The current blood sugar meter device required having a drop of blood to analyze, thus involving painful manual tasks. Moreover, the tests are usually run not more than 5 times a day by the patient, thus giving only partial data for later diagnostics by the doctors. But new technologies are bringing promising features in the context of the smart advisor. Optical or other high-frequencies electromagnetic sensors will soon allow one to perform non-invasive and continuous measure of blood-sugar level. The combination of these two advanced features is of great interest for the project, because for the first time, this allows real time blood sugar trends to be seen. These new technologies will allow unattended testing three to twelve times an hour. A wearable unit (like a watch²) could then send its results to the smart advisor through a wireless connection.

In addition to blood sugar meters, new wearable bio-sensors enable continuous physiological and lifestyle data collection outside the lab environment. Commercial products that

² www.glucowatch.com

Wireless network technologies like Bluetooth will enable these rich streams of data to be shared across different devices, or uploaded to a remote station in order to be processed.

3.3.2 Expert system and rules development

A major challenge for further developments of the project will be to integrate the flow of data made available by the tools mentioned above, in order to formulate reliable and simple lifestyle advice. An expert system will generate personal guidelines based on the information manually entered by the user (diabetes, hyper-tension, heart condition for example) and the one gathered seamlessly by wearable sensors. The complexity of the system relies in the fact that often people encounter several chronic diseases at the same time. In these cases the guidelines for the user would be a complex integration of the one for each of the disease. In other words, the diet advice for someone suffering from diabetes and anemia is not the sum, or even the intersection of the two diets.

Hence, it seems unavoidable to have a “man in the loop”, a generalist or specialist doctor who will at least partly generate the personal guidelines. The smart advisor idea is not about replacing a health professional and its judgments, but to make them virtually available anytime, anywhere. The portable platform would be used to answer the question: “In this situation, what would my doctor advise me to do?” As demonstrated during the experiment, people need structure as much as relevant information to take the good decision. Moreover, by continuously monitoring the user’s physiological data, the system could record trend and detect pattern, in order to trigger various alarm from warning to the user to emergency’s call. This information can provide precious support during a patient’s periodic medical visit.

3.3.3 Large-scale deployment

The first prototype enabled us to point out some of the challenges needed to be overcome for a large scale deployment of “health smart advisor”. As a platform that provides integrated health services, the infrastructure requires the active involvement from a large number of enablers.

3.3.3.1 Manufacturers

The very first difficulty we faced at the beginning of the project was to obtain product information from the manufacturers. The manufacturers seem not willing to share the database of

their product contents, even the one containing public information as required by the US Food and Drug Administration and displayed on package labels. The system would certainly gain by having more than nutrient data, but information on potential allergen ingredient as well. The nature of the information publicly released to consumers is strictly regulated by the Food and Drug Administration. Having the same kind of mandate, where the companies would be forced to post the products health-related content on a website, is possible, if the administration consider that it would significantly enhance the public health by allowing a broad range of new services.

3.3.3.2 Retail stores

The integration of the system within the retail stores' logistic system is necessary, at least to provide the means of identification of the products. The platform from which one access to the service might belongs to the local store, as shown with the smart cart example. Thus the investment would not be made by the end-user, but by the stores that could provide new services. It is clear that gaining the commitment of the retail industry through partnerships is a key to implement the system. By enabling them to provide new services, the smart advisor would open revenue streams for the industry. The device also enhances the interface with their customers. The system can gather data on their purchase pattern and create opportunities for innovative and personalized marketing strategies. Knowing details on the diet requirements of their customers, the stores could "push" some products with some health-advice arguments.

Although theses concepts are important, because they are incentives for the retail industry to facilitate the smart advisor's implementation, they are sources of failures. We could verify during the project that the user's sensitivity is high to segregate a health service from a marketing tool. The source of the information, as the proper use of her/his personal information will determine how the user would trust and ultimately rely on the system. It seems crucial to have a clear frontier between the two worlds built in the system, as the consequences for the user perception would define the success of it.

3.3.3.3 Health Professionals

The involvement of the health community is obviously necessary to further develop and implement the concept. As mentioned above, the expert system that will generates the rules will

require a human intervention, be it to enter the patient medical profile, or to customize the system to complex and specific needs of the patient.

We hope that the health community would see in the Smart Advisor a tool to better provide their services, and by no way a means to replace or even by-pass them. The smart advisor promotes preventive actions like personalized healthy diets and lifestyle guidelines. It actually facilitates the communication between the patient and her/his doctor by making the latter virtually available 24/24H for specific advice. Moreover, as a data gathering tool and integrator, the system is a platform to that enables knowledge sharing between different health professionals on their common patient.

As with any autonomous system with a human supervisory control (here the doctor), the smart advisor raises complex liability issues. Let's take an example of what can be considered as a "failure" of the device. Despite the use of her "smart advisor", a patient has been exposed to an allergen contained in a common food product, serious enough to require a hospitalization. The chain of responsibility in that case can be hard to establish: The manufacturer might to release the presence of this particular allergen, the product identification might have failed, or the doctor might not have inputted the particular condition in the patient online profile.

3.3.3.4 Ethical issues

We claimed that the smart advisor would extend the relationship between a patient and her doctor. The health professional would have greater access to patient information, by being able to monitor bio-data or behavior pattern in daily-life situations. Although it might enhance the efficiency of the therapeutics and give opportunity to personalize them better, this situation raises ethical issues. An insurance company that sponsors the service could use this information against its customer. It could deny financing a hospitalization, by arguing that the patient, equipped with a smart advisor, was perfectly aware of the risks induced by his behavior. More generally, the system can strengthen the debate on the relationship between the goals of the public health system and the individual behaviors.

3.3.4 Conclusions

The purpose of the present study was to evaluate the possibilities offered by new information technologies in the area of personal safety.

We identified the relevant technologies that would contribute to our project effort. We confirmed the availability of adequate technologies in order to implement a health personal advisor that would integrate data from various sources on a real-time basis to provide “advice” to the consumer. The integration of highly portable computers and wireless communication enabled us to develop a system that is available at any point-of-decision, for example in the context of a person shopping in a pharmacy or market to ensure that foods, medicines or other products do not pose hazards relative to a person’s known medical condition.

A prototype system has been tested and potential change of behavior and other human factors issues of a cross-section of individuals using it have been evaluated. The test confirmed the great appeal of a decision-aiding system for test users who have to manage strict or complex diets. We feared some reluctance from the people to introduce a new and rather technical tool in such common tasks as grocery shopping. But we discovered that consumers were potentially enthusiastic about having an integrated system that would not only provide relevant information, but also personal advice or “judgment”. The field test allowed us to identify some of the challenges to overcome in a real life implementation of the system.

The database architecture is a key factor because it is closely related to the relationships of the enablers of the service, such as the health professionals, the products manufacturers and retailers, or even the insurance companies.

The user interface and the way the message is delivered on the client platform would ultimately determine the user’s acceptance of the system. A careful handle on privacy issues, a clear vision of what drives the system’s decisions and who participate in the process would enable trust and reliance.

At the beginning of the research we focused on the performance and the features offered by the system. However the evaluation of the prototype made us realize the importance of one factor that was underestimated. A theoretical perfect accuracy of the device would not necessarily guarantee compliance with the advice. A deep understanding of a consumer’s decision drivers, the way he or she perceives different kind of risks and achieves trade-off between immediate gratification and possible health hazard decisions, is necessary in order to promote a long-term change of behavior.

Appendix A:References

B. Yang, S. Rhee, and H. Asada. "A Twenty-Four Hour Tele-Nursing System Using a Ring Sensor". International Conference on Robotics and Automation, Leuven, Belgium, May 16-20, 1998.

D. Haytowitz, P. Pehrsson, J. Smith, S. Gebhardt, R. Matthews, and B Anderson. "Key Foods: Setting Priorities for Nutrient Analyses". Journal of Food Composition and Analysis 9, 331-364, 1996.

T. Sheridan. "Telerobotics, Automation, and Human Supervisory Control". MIT Press, 1992.

J. Joneja. "Food Allergies & Intolerance. A Comprehensive Guide". J.A. Hall Publications

Development tools and resources for the client platform

Java. java.sun.com

Palm. www.palm.com

Palm VII description. www.palm.com/products/palmvii

Palm VII white paper. www.palm.com/pr/palmvii/7whitepaper.pdf

Web Clipping documentation www.palm.com/devzone/docs.html

Waba Virtual Machine. www.wabasoft.com

Superwaba SuperWaba Java Virtual Machine. www.superwaba.org

Reference for Waba developer. <http://www.wabaworkbench.com/>

Bar Codes Software, Hardware and Industry news. <http://www.barsnstripes.com/>

Dynasys <http://www.dyna-sys.com/palmpilot/>

Sources for health and nutrition

<http://vm.cfsan.fda.gov/~dms/eafus.html>

EAFUS: A Food Additive Database

This is an informational database maintained by the U.S. Food and Drug Administration (FDA) Center for Food Safety and Applied Nutrition (CFSAN) under an ongoing program known as the Priority-based Assessment of Food Additives (PAFA). It contains administrative, chemical and toxicological information on over 2000 substances directly added to food, including substances regulated by the U.S. Food and Drug Administration (FDA) as direct, "secondary" direct, and color additives, and Generally Recognized As Safe (GRAS) and prior-sanctioned substances. In addition, the database contains only administrative and chemical information on less than 1000 such substances. The more than 3000 total substances together comprise an inventory often referred to as "*Everything*" Added to Food in the United States (EAFUS).

<http://www.nal.usda.gov/fnic/foodcomp/links.htm>

Food composition and nutrition list. Maintained by the US Drug Administration.

<http://www.nal.usda.gov/fnic/foodcomp/Data/SR13/reports/sr13page.htm>

USDA Nutrient Database for Standard Reference,
Release 13

<http://www.allallergy.net>

Comprehensive portal for Asthma, Allergy and Intolerance Information on the Web. The site contains numerous links to publications, public and private organizations and food database.

<http://www.usda.gov>

Official Site of the US Department of Agriculture. The Department publishes an important nutrient database. The *USDA Nutrient Database for Standard Reference (SR)* is the major source of food composition data in the United States. It provides the foundation for most food composition databases in the public and private sectors.

Resources for diabetes

<http://www.diabetes.org>

official site of the American Diabetes Association. The American Diabetes Association is the nation's leading nonprofit health organization providing diabetes research, information and advocacy.

<http://www.niddk.nih.gov/>

National Institute of Diabetes and Digestive Diseases

<http://www.cdc.gov/nccdp/ddt/ddthome.htm>

Center for Disease Control's Diabetes and Public Health Resource

<http://www.aacc.org/>

American Association of Clinical Chemistry

<http://www.diabetes.ca/atoz/general.htm>

Diabetes Facts

<http://www.idi.org/>

International Diabetes Web Sites

<http://www.idf.org/>

International Diabetes Federation

Web commercial services

Note: the commercial applications and services are mentioned here because they are closely related to the field of research of the project. The small description provided is as it appears on the respective websites.

<http://www.foundcount.com>

FoodCount.com is a website designed to help you make smart nutrition decisions by giving you the power of knowledge in tracking vital nutrients contained in the foods and beverages you consume everyday. The first step to a healthy diet is knowing exactly what you are putting in your body.

<http://bowline.net/consumer.htm>

Bowline.net has created the Consumer Access Platform to deliver this vision of consumer-choice advertising. Our goal is to revolutionize the relationship between the advertiser and the

consumer. We have developed a network to give widespread access to the consumer. This network allows the distribution of messages to consumers when and where they want to receive them.

<http://www.epocrates.com>

Developed specifically for the Palm handheld platform, ePocrates Rx™ delivers up-to-date and easy-to-use information on over 99% of the most commonly prescribed medications. Including such information as indication-specific dosing, drug-drug interactions, adverse reactions, mechanism of action, retail pricing and packaging and tablet/capsule descriptions, ePocrates Rx™ was designed by leading physicians and PharmDs from across the country to reduce drug errors and enhance workflow.

<http://www.nutribase.com>

NutriBase is a nutrition and fitness management software package designed to help your clients lose, gain, or maintain weight; to monitor intakes ("Food recalls") for nutrient quality; to help you create and optimize recipes; to help you plan and supervise meal plans; to generate reports and graphs; to track personal and medical information; to track and monitor exercise and activities (including target heart rate training); and to learn about the comparative nutrition offered in the foods you prescribe. NutriBase provides an integrated version of the USFDA Nutrient database for free.

<http://www.barpoint.com>

BarPoint is a revolutionary wireless and Internet product information and shopping service. It uses patent-pending search engine technology to allow businesses and consumers to use the UPC barcode number that appears on more than 100 million retail items to instantly obtain detailed product and price information from the Internet. Cellular phones, personal digital assistants, interactive pagers and other devices can now access the Internet remotely. Shoppers can then make educated decisions when they need to most ... at the point of purchase.

<http://www.beelineshopper.com>

Scan the barcodes of foods and grocery products you regularly use, and we'll create a shopping list filled with everything you need to pick up at the store. Plus, we automatically offer substitution suggestions, based on your dietary needs and the nutrition guidelines provided by the dietitians at the University of Michigan. We have the largest list of food equivalents available anywhere to help you make better choices. If you shop at a participating store, your grocery list will be organized by aisle so you can waltz right through it without a missed step, making a beeline to the best foods and the best values.

www.airclick.com

By simply scanning on a bar code with an AirClic-enabled scanner, or entering a bar code's numbers into a web-based phone, pager or PDA, thousands of possibilities unfold -- AirClic's platform delivers information to the user across a broad array of devices, including PC's and web-enabled phones, PDA's and pagers. The result is information and the ability to transact simply, accurately and quickly, wherever you are.

<http://www.bodymedia.com>

BodyMedia is a body monitoring solutions provider. We've developed a platform that integrates our wireless body monitors with our customizable software applications and other third party devices. Our tools for continuous body monitoring include both wireless, wearable body monitors and powerful, easy-to-use software applications. These revolutionary products are integrated in order to provide a wide range of solutions to your body monitoring needs.

Appendix B: Smart Advisor : Tasks Analysis

Objective: Identify the minimum theoretical steps to be performed by the user for each of the 4 features previously developed, i.e. Scan, Compare, Record and View.

Hypothesis

No particular technological constraints are considered for this exercise. In particular, we will not take account of some features limitations of the Palm VII or any other handheld device. Nonetheless, it appears necessary to define the infrastructure of the system, and give assumptions about where the data involved are stored. It has direct and visible consequences on the user interface, independently of the technology used.

1) Product information

The data on the products is not stored on the handheld device. This is a reasonable assumption, because of the amount and nature of the information. The product database is by nature very large and ever evolving; it could be stored on a distributed basis at the manufacturer or retail store sites, or in a central database. The point is that a wireless communication has to be established by the user to access the product information.

2) Personal Profile

The profile of the user is loaded on his personal device. His diets guideline, allergies, medication and bio-data are carried by the user. The data can be synchronized more or less seamlessly with a remote server, but the assumption made here is that it is not a required task when the user is shopping for example.

3) Items purchased or recently scanned

The content and description of the “personal shelves” is also stored on the handheld device. That means that the data on all the items previously purchased by the user or even scanned are then stored on the personal computer and directly available by the user. In other words, the handheld device can locally store the result of a previous request, like with the cache function of an internet browser.

4) Meal plan

The products along with the amount updated in the user meal plans are also available directly on the handheld device.

The Scenario Approach

A major issue of the software that has been raised during the experiment was its lack of navigability. The users do not perform the different tasks in the linear scenario used for the initial, but rather want to pick up some of the functions, in different orders, based on their needs at the moment. At this point there is a design trade-off to face: How to provide the user with shortcuts and other means to make the navigation flexible, and not to overload the user interface with too many buttons. To later discuss about this trade-off as well as other design issues, I looked at different scenarios and tried to isolate the “shortcuts” requirements.

I identified basic tasks, as displayed below. Each task consists in a simple goal, a required state of the system, an action and a result. These tasks are later combined into different scenarios, which will lead to a discussion on the system design.

Identification	Goal	Required State	Action	Result
Task A	To Specify an Item	System in listening state	To perform a Barcode Scan	The Barcode is stored on the device
Task B	To obtain a simple advice on a product	The product barcode is accessible to the system	Push a button on the screen	The system retrieves the product information from a remote server and displays an advice based on the user profile stored locally
Task C	To obtain Quantitative and detailed information on a product	The product barcode is accessible to the system	Push a button on the screen	The system retrieves the detailed product information from a remote server and displays an ordered list based on the specified criteria
Task D	To compare different products	At least 2 barcodes are accessible by the system	Push a button on the screen and select a criteria	The system retrieves the products information from a remote server and displays an ordered list based on the specified criteria

Task E	To “Buy” a product	The product barcode is accessible to the system	Push a Button on the screen	The product identification is stored in the “personal shelves” section
Task F	To update the meal plan	The product barcode is accessible to the system	Push Record button, select a product and an amount	The Meal plan is updated and the user receives a confirmation
Task G	To receive advice on the meal plan	A meal plan is recorded and accessible	Push View Meal Plan Button	The system retrieves the information on the meal plan and displays the analysis to the user.

Different scenarios that are corresponding of alternative combinations of the 5 tasks are to be considered to optimize the navigation in the system.

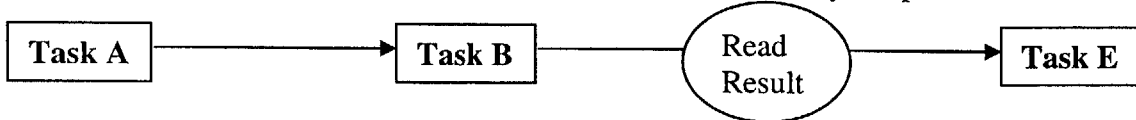
Scenario 1

The user already knows the product that he intends to buy. He wants to simply notify the system of his purchase. The user should then accomplish two tasks: the selection of the item and its registration as “purchased”.



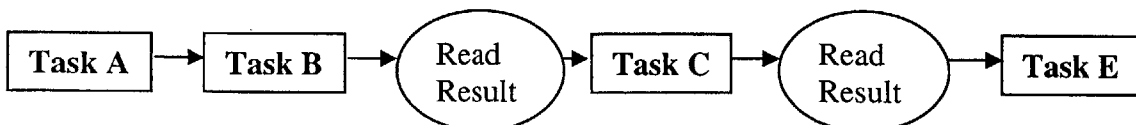
Scenario 2

The user seeks an advice for a particular item. After having scanned it, he pushes the “advice” button and read the instructions. The user then decides to buy the product.



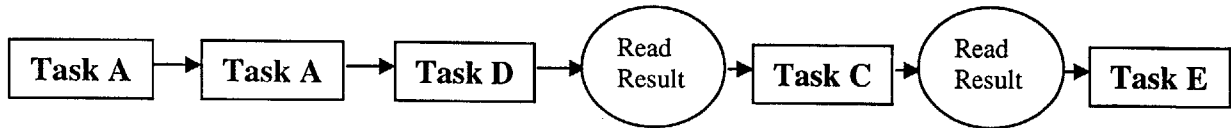
Scenario 3

Same as Scenario 2, except that the user seeks additional and detailed information on the product to support the advice, before he makes a decision.



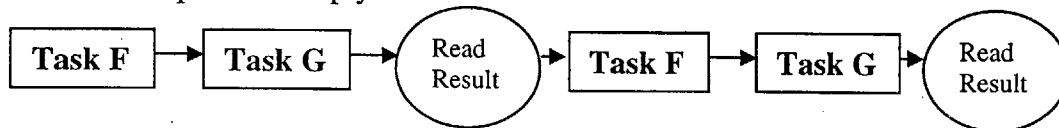
Scenario 4

The user select a list (at least 2) of products that he would like to compare. He then hit the “Compare” button and select a comparison criteria. After obtaining the ranked list, the user looks for more information on the “best” product. He then decides to “buy” the product.



Scenario 5

At home, the user first updates his meal plan by repeating the task F for each product. He then can request an analysis of his meal plan and might want to change one or more entries in the meal plan to comply with the advice.



Discussion

We can point out some general characteristics of the shopping experience from the previous task analysis.

The first one is that the actions are highly **repetitive**: we were able to isolate 5 generic scenarios. The main consequence is that after a learning period, the user would certainly want to minimize the number of tasks to perform. As an example, it appeared during the experiment that the users sometimes want to directly purchase a product, without asking for advice. Indeed, for well known items, the user should have the possibility to only register his purchase. This could ideally be done at the cash-register, without any specific action from the user.

Another major characteristic that came out is the **flexibility** in term of the order of the different tasks. The navigation within the client software should be easy enough to allow the user to correct certain inputs after a first result, or add new items in a comparison list, without having to restart from the beginning. The issue is how to integrate “shortcuts” or “back” button in the user interface and not to overwhelm it at the same time.

Finally we should consider the required **flow of information** as mentioned in the beginning of this paper. Different source of information are necessarily accessed during the user requests. His profile and diet are assumed to be available locally, stored on the device itself (and synchronized periodically with a remote database), whereas it is likely that the product information will be stored remotely, be it on the retail store site or in a remote database. This has an impact on the

user interface by introducing different delays for the different queries. It might also be a little confusing for the end-user, who can have difficulties to understand the constraints, as it appeared during a field test.

Appendix C: Database structure for the diabetes breakfast application

```
# Host: *****
# # Server version: 3.23.27
# PHP Version: 4.0.6
# Database : diabete
# -----
```

- Nutri

// nutrients definition and unit, from the USFDA Nutrient Database

Field	Type	Attributes
Nutr_No	varchar(4)	Code of the nutrient (USFDA Nutrient Database)
Units	varchar(6)	Units
Tagname	varchar(20)	Short name
NutrDesc	varchar(60)	Long Name

- product

// 60 products descriptions, entered manually from package labels

Field	Type	Attributes
Category	varchar(20)	Category of breakfast products
UPC	bigint(20)	Barcode displayed on packages
Brand	varchar(20)	Brand name
label	tinytext	Product name
serv1lab	tinytext	Type of serving (spoon, container, units...)
serv1	decimal(10,0)	Size of 1 serving in grams
servnumb	decimal(10,1)	Number of servings
cal	decimal(10,3)	Calories per serving
carbo	decimal(10,3)	Carbohydrates per serving
sugar	decimal(10,3)	Sugar per serving
fat	decimal(10,3)	Total fat per serving
satfat	decimal(10,3)	Saturated fat per serving
chol	decimal(10,3)	Cholesterol per serving
sodium	decimal(10,3)	Sodium per serving
fiber	decimal(10,3)	Fiber per serving
GI	decimal(10,0)	Glycemic Index

- recom

// Recommended product per category of breakfast food.

Field	Type	Attributes
category	varchar(20)	Category of breakfast products
UPC	bigint(20)	Barcode displayed on packages
brand	varchar(20)	Brand name
label	Tinytext	Product name
comment	Text	Comment associated with the choice

- eat

//Products and amount inputed in the Break Fast meal plan

Field	Type	Attributes
barcode	bigint(20)	Barcode displayed on packages
date	date	Date of the breakfast meal plan
amount	decimal(10,3)	Number of servings

- prof

//personal information and recommendations

//the units for each nutrients are given in the nutri table

Field	Type	Attributes
gender	varchar(10)	Gender
wgt	decimal(10,0)	Weight in pounds
ht	decimal(10,0)	Height in feet
activity	decimal(10,0)	Level of activity (check profile.php for convention)
cal	decimal(10,0)	Recommended daily calories
factor	decimal(11,0)	% of carbohydrates in the calories intake
carbo	decimal(10,0)	Recommended daily carbohydrates
breakf	decimal(10,0)	% of the breakfast in the daily meal plan
fat	decimal(10,3)	Recommended daily fat
satfat	decimal(10,3)	Recommended daily saturated fat
chol	decimal(10,3)	Recommended daily cholesterol
sodium	decimal(10,3)	Recommended daily sodium
fiber	decimal(10,3)	Recommended daily fiber

- Prof1

//former recommendations input. Used by the profile.php script

Field	Type	Attributes
Nutr_No	char(4)	Code of the nutrient (USFDA Nutrient Database)
limits	decimal(10,3)	Former daily recommendation for the nutrient

- Purchase

//list of the products purchased with the date

//the serving is the number of servings left in the personal inventory

Field	Type	Attributes
prod_id	tinyint(4)	Auto-increment
barcode	bigint(13)	Barcode displayed on packages
date	date	Date of purchase
serving	decimal(10,1)	Number of servings left

```
PHP Server Code
#Server version: 3.23.27
# PHP Version: 4.0.6
```

Reviewshelves.php

```
-----
-----
<html>
<head>
    <title>Inventory</title>
</head>
<body>

<?php

// this script retrieve the "personal
inventory" stored in the purchase
// database
// Access: for the palm
// url: reviewshelves.php
// GET variables : none (mono user version)
// Tables used : purchase, product

// Connection to the database

$free_login = 'root' ;
$free_passwd = 'titp4md';

mysql_connect('localhost',$free_login,$free_pa
sswd) or die ("cannot connect to the server");
mysql_select_db('diabete') or die ("cannot
connect to the database");

echo "<H2><CENTER>Personal Shelves</CENTER>" ;
```

```
echo "<BR><BR>" ;
$today = date( "Ymd", time() );
echo "<DIV ALIGN = 'RIGHT'><PRE>size      left
</PRE></DIV>" ;

//retrieve the UPC and the amount left of the
products purchased TODAY

$sql2 = "SELECT barcode, serving FROM purchase
WHERE date = ".$today." GROUP by barcode ORDER
BY barcode " ;

$id2 = mysql_query($sql2) ;
$n = 0 ;
$list= "" ;
while (mysql_fetch_row($id2))
{
    $item[$n] = mysql_result($id2,$n,0) ;
    $amount[$n] = mysql_result($id2,$n,1) ;
    $list = $list." OR UPC = ".$item[$n] ;
    $n++ ;
}

//retrieve the name, serving size label of the
previous products

$req = "SELECT p.label, p.serv1lab FROM
product p WHERE UPC =-1 ".$list." order by
UPC" ;

$id = mysql_query ($req) ;
$row2 = 0 ;

while (mysql_fetch_row($id))

{

    $lab[$row2] = mysql_result($id,$row2,0) ;
    echo "<BR>" ;
```

```

echo $lab[$row2] ;
echo "<DIV ALIGN = 'RIGHT'><PRE>" ;
$ser[$row2]= mysql_result($id,$row2,1) ;
echo $ser[$row2] ;
echo " " ;
echo $amount[$row2] ;
echo "</PRE></DIV>" ;
$row2++ ;

}

// link to the welcome module on the palm
pilot

echo "<a href='palm:OUxL.appl'
BUTTON>MAIN</a>" ;

?>

</body>
</html>

```

Review2.php

```

-----
-----

<html>
<head>
    <title>Breakfast</title>
</head>
<body>
<H2><CENTER>Breakfast Meal Plan</CENTER></H2>

<?php
// this script gives a report on the meal
plan set by the user and
// stored in the eat database.
// Access : from the palm
// URL: review2.php
// GET variables : none (mono user version)
// tables used : eat, prof1, nutri, product

//connection to the database

$login = 'root' ;
$password = 'titp4md';
mysql_connect('localhost',$login,$password) or
die ("cannot connect to the server");
mysql_select_db('diabete') or die ("cannot
connect to the database");

$today = date( "Ymd", time() );

// retrieves the barcode along with the amount
(in servings) of each product entered in
// the meal plan

```

```

$sql2 = "SELECT barcode, amount FROM eat
WHERE date = ".$today." GROUP by barcode ORDER
BY barcode " ;

```

```

$id2 = mysql_query($sql2) ;
$n = 0 ;
$list= " " ;
while (mysql_fetch_row($id2))
{
    $item[$n] = mysql_result($id2,$n,0) ;
    $amount[$n] = mysql_result($id2,$n,1) ;
    $list = $list." OR UPC = ".$item[$n] ;
    $n++ ;
}

```

```

// retrieves the nutrient data of each product
and the limits entered in the personal profile

```

```

$sql = "SELECT p.Nutr_No, p.recom,p.limits,
n.NutrDesc, n.Tagname FROM prof1 p, nutri n
WHERE p.Nutr_No = n.Nutr_No " ;
$id = mysql_query ($sql) ;
$row = 0 ;
$flag = 0 ;

```

```

$req = "SELECT p.label, p.servllab" ;

```

```

while (mysql_fetch_row($id))
{
    $total[$row] = 0 ;
    $nutr[$row] = mysql_result($id,$row,4) ;
    $recom[$row] = mysql_result($id,$row,1) ;
    $lim[$row] = mysql_result($id,$row,2) ;
    $nam[$row] = mysql_result($id,$row,3) ;
    $req = $req.',p.'.$nutr[$row] ;
    $row++ ;
}

```

```

$req = $req.',p.serv1, p.servllab FROM product
p WHERE UPC =-1 '.$list.' order by UPC' ;

```

```

echo "<br>" ;

```

```

$id = mysql_query ($req) ;
$row2 = 0 ;
echo "<H3>" ;

```

```

//display the results on the palm screen

```

```

while (mysql_fetch_row($id))

```

```

{
    $i = 0 ;
    $lab[$row2] = mysql_result($id,$row2,0) ;
    $ser[$row2]= mysql_result($id,$row2,$row+3)
;

```

```

echo round($amount[$row2],1)." *
".$ser[$row2]." of ".$lab[$row2]."<BR>" ;

```

```

While ( $i < $row )
{

```

```

    $quant[$i] = mysql_result($id,$row2,$i +
2) ;

```

```

// a product is forbidden if it contains a
nutrient that is completely forbidden
// (limit=0) in the user's diet.

```

```

if ($lim[$i] == 0) { $flag = 1 ; echo "
not allowed" ;

```

```

    $perc = 101 ;
}

```

```

else

```

```

    $perc =
$quant[$i]*$amount[$row2]/$lim[$i]*100 ;

```

```

// the $flag variable set to 1 indicates that
a nutrient intake, given the amount

```

```
// of the food product taken, exceeds 25% of
the daily need.
```

```
    if ($perc > 25) { $flag = 1 ;
                      $list = $list.$nam[$i]."  
<br>" ;
                      }
}
```

```
$total[$i] = $total[$i] +
$quant[$i]*$amount[$row2] ;
```

```
$i++ ;
}
```

```
$row2++ ;
```

```
}
```

```
echo "</H3>" ;
echo "<BR><BR>";
$m = 0 ;
$list1 = "" ;
$list2 = "" ;
$flag1 = 0 ;
while ($m < $row)
```

```
{
  echo "      " ;
  echo "      " ;
  $perc = (int)($total[$m]/$lim[$m]*100)
```

```
  ;
  if ($perc > 31) {
    $flag1 = 1 ;
    $list1 = $list1."<DIV
align=CENTER><H1>". $nam[$m]. "</DIV></H1>" ;
    $list2 = $list2."<DIV
align=RIGHT><H1>". $nam[$m]. " ". $perc."
%</DIV></H1>" ;
```

```
    else
```

```
{
```

```
    $list2 = $list2."<DIV
align=LEFT><H3>". $nam[$m]. " ". $perc.%
</DIV></H3>" ;
```

```
    $m++ ;
  }
```

```
// $flag1 equals 1 if at least one of the
nutrient intakes exceeds 31% of the daily
needs.
```

```
if ($flag1 == 1)
```

```
{
  echo "<BR>Your breakfast is
very rich in : " ;
  echo $list1 ;
}
```

```
else echo "Your breakfast complies with
your personal diet" ;
```

```
echo "<BR>";
```

```
// link to the application that allows to
update the meal plan
```

```
echo "<a href='palm:JrvX.appl'
BUTTON>Change</a>" ;
echo "<DIV align=RIGHT><a
href=\"#list\">Details</a><BR></DIV>" ;
echo "<a name=\"list\"><H2><CENTER>Nutrient
content</CENTER></H2></a>" ;
echo $list2 ;
echo "<hr>" ;
```

```
?>
```

```
</body>
</html>
```

```
-----
-----
```

profile.php

```
-----  
-----  
<html>  
<head>  
    <title>Profile</title>  
</head>  
<body>  
    <H1><center>PERSONAL PROFILE</center></H1>  
  
    <H2> On this page, you will input some  
information, and thus update your personal  
diet guidelines. The following ratio are  
combined with your personal information by the  
system to  
        recommend a Carbohydrates intake  
for your breakfast</H2>  
  
    <FORM ACTION=profile.php METHOD=GET>  
    <table ALIGN=CENTER WIDTH = 70% cols=3>  
    <tr COLSPAN = 2 ALIGN = CENTER  
><H2>Personal Info</H2></tr>  
  
<?php  
  
// this script has several purposes. It  
displays the personal profile  
// of the user, generates recommendation  
(daily limits) based on its profile  
// and on rules embedded in the script itself  
// allows to update the profile and manually  
change the recommendations  
// Access : from a web client  
// url : profile.php  
// GET variables : none or selfgenerated in  
the page through the POST form  
// tables used : prof, prof1  
  
// connection to the database
```

```
$free_login = 'root' ;  
$free_passwd = 'titp4md';  
  
mysql_connect('localhost',$free_login,$free_pa  
sswd) or die ("probleme connexion");  
mysql_select_db('diabete') or die ("probleme  
dans selection base");  
  
// If the parameters are passed are GET  
variables, they are updated in the prof1 and  
prof profiles  
  
if (isset($wheight))  
{  
    $sql1 = "UPDATE prof SET wgt=$wheight"  
;  
    mysql_query($sql1) ;  
}  
  
if (isset($activity))  
{  
    $sql1 = "UPDATE prof SET  
activity=$activity" ;  
    mysql_query($sql1) ;  
}  
  
if (isset($gender))  
{  
    $sql1 = "UPDATE prof SET  
gender=$gender" ;  
    mysql_query($sql1) ;  
}  
  
if (isset($scarboratio))  
{  
    $sql1 = "UPDATE prof SET  
factor=$scarboratio" ;
```

```

        mysql_query($sql1) ;
    }

if (isset($breakratio))

    {
        $sql1 = "UPDATE prof SET
breakf=$breakratio" ;
        mysql_query($sql1) ;
    }

if (isset($N208))

    {
        $sql1b = "UPDATE prof1 SET limits=$N208
WHERE Nutr_No='N208'" ;
        // $sql1 = "UPDATE prof SET
satfat=$N208" ;
        // mysql_query($sql1) ;
        mysql_query($sql1b) ;
    }

if (isset($N204))

    {
        // $sql1b = "UPDATE prof SET
fat=$N204" ;
        $sql1 = "UPDATE prof1 SET limits=$N204
WHERE Nutr_No='N204'" ;
        mysql_query($sql1) ;
        // mysql_query($sql1b) ;
    }

if (isset($N606))

    {
        $sql1b = "UPDATE prof1 SET limits=$N606
WHERE Nutr_No='N606'" ;
        // $sql1 = "UPDATE prof SET
satfat=$N606" ;

```

```

        // mysql_query($sql1) ;
        mysql_query($sql1b) ;
    }

if (isset($N601))

    {
        $sql1b = "UPDATE prof1 SET limits=$N601
WHERE Nutr_No='N601'" ;
        // $sql1 = "UPDATE prof SET chol=$N601"
;
        // mysql_query($sql1) ;
        mysql_query($sql1b) ;
    }

if (isset($N205))

    {
        $sql1b = "UPDATE prof1 SET limits=$N205
WHERE Nutr_No='N205'" ;
        // $sql1 = "UPDATE prof SET chol=$N205"
;
        // mysql_query($sql1) ;
        mysql_query($sql1b) ;
    }

if (isset($N307))

    {
        $sql1b = "UPDATE prof1 SET limits=$N307
WHERE Nutr_No='N307'" ;
        // $sql1 = "UPDATE prof SET
sodium=$N307" ;
        // mysql_query($sql1) ;
        mysql_query($sql1b) ;
    }

if (isset($N291))

```

```

{
    $sql1b = "UPDATE prof1 SET limits=$N291
WHERE Nutr_No='N291' " ;
    // $sql1 = "UPDATE prof SET fiber=$N291"
;
    // mysql_query($sql1) ;
    mysql_query($sql1b) ;
}
if (isset($N269))

{
    $sql1b = "UPDATE prof1 SET limits=$N269
WHERE Nutr_No='N269' " ;
    // $sql1 = "UPDATE prof SET fiber=$N291"
;
    // mysql_query($sql1) ;
    mysql_query($sql1b) ;
}

// display the personal info as entered the
last time

$sql = "SELECT
gender,wgt,activity,cal,factor,breakf,fat,satf
at,chol,sodium,fiber FROM prof " ;
$id = mysql_query ($sql) ;
$row = 0 ;
$outresult = "" ;
$back = 0 ;

$a = "#dddddd" ;
$c= "#ff6666" ;
$b="#cccccc" ;
$wht = mysql_result($id,0,1) ;

//$cal = mysql_result($id,0,3) ;
$ratio2 = mysql_result($id,0,4) ;
$breafratio = mysql_result($id,0,5);

```

```

if (mysql_result($id,0,0)==0) { $gend =
"Male" ; $actnum = mysql_result($id,0,2) ; }
else {$gend = "Female" ;
$actnum= mysql_result($id,0,2) - 1.5 ; } ;
/*
if (mysql_result($id,0,2)=='13') $act = 'Very
sedentary' ;
if (mysql_result($id,0,2)=='14') $act =
'Sedentary' ;
if (mysql_result($id,0,2)=='15') $act =
'Moderately Active' ;
if (mysql_result($id,0,2)=='16') $act =
'Active' ;
if (mysql_result($id,0,2)=='17') $act =
'Super Active' ;
*/

// set the activity level
// the rules used to generate the
recommendations are described in the Thesis

if (mysql_result($id,0,2)=='13') $act = 'Level
5' ;
if (mysql_result($id,0,2)=='14') $act =
'Level 4' ;
if (mysql_result($id,0,2)=='15') $act =
'Level 3' ;
if (mysql_result($id,0,2)=='16') $act =
'Level 2' ;
if (mysql_result($id,0,2)=='17') $act =
'Level 1' ;

$cal = (int) ($wht*$actnum) ;
$carbo = (int) ($wht*$actnum*$ratio2/4/100) ;
// $sqlcar = "UPDATE prof SET carbo=$carbo" ;
// mysql_query($sqlcar) ;

$sqlcar = "UPDATE prof1 SET limits=$carbo
WHERE Nutr_No='N205' " ;
if (isset($N205)) {} else {
mysql_query($sqlcar) ; } ;

```

```

$ssqlb = "UPDATE profl SET limits=$scal WHERE
Nutr_No='N208'" ;
if (isset($N208)) {} else {
mysql_query($ssqlb) ;} ;

```

```

echo '<tr bgcolor = ' . $c. ' ALIGN =
CENTER><td WIDTH = 30%></td><td WIDTH =
30%>Current Value</td><td>New Value</td></tr>'
;

```

```

echo '<tr bgcolor = ' . $a. ' ALIGN =
CENTER><td WIDTH = 30%>Gender</td><td WIDTH =
30%>' . $gend. '</td>' ;
echo '<td WIDTH = 30%> <select name="gender">
<option value="">Select</option>
<option value="0">Male</option>
<option value="1">Female</option>

```

```

</td></tr>' ;
echo '<tr bgcolor = ' . $b. ' ALIGN = CENTER><td
WIDTH = 30%>Wheight</td><td WIDTH =
30%>' . mysql_result($id,0,1) . '</td>' ;
echo '<td><input type ="TEXT" name="wheight"
size = 10></td></tr>' ;
echo '<tr bgcolor = ' . $a. ' ALIGN =
CENTER><td>Activity</td><td WIDTH =
30%>' . $act. '</td>' ;

```

```

echo '<td WIDTH = 30%> <select
name="activity">
<option value="">Select</option>
<option value="13">Level 5</option>
<option value="14">Level 4</option>
<option value="15">Level 3</option>
<option value="16">Level 2 </option>
<option value="17">Level 1</option>
</td></tr>' ;

```

```

echo ' <tr ALIGN = CENTER>
<td WIDTH = 30%></td>
<td align=CENTER width="30%"> </td>
<td ALIGN= RIGHT>
<input type="submit" name="ACTION"
value="Submit">
</td></tr>' ;

```

```

echo '</table>' ;
echo '<br>' ;
echo '<h1> </h1>' ;

```

```

echo " <table WIDTH = 70% cols=3 ALIGN =
CENTER>" ;
echo " <tr COLSPAN = 2 ALIGN = CENTER
><h2>Basic ratios</h2></tr>" ;
echo '<tr bgcolor = ' . $c. ' ALIGN =
CENTER><td WIDTH = 30%></td><td WIDTH =
30%>Current Value</td><td>New Value</td></tr>'
;

```

```

echo " <tr bgcolor = " . $b. " ALIGN =
CENTER><td WIDTH = 30%>Percentage of
carbohydrate in the total calorie intake
(recom.: 50%)</td><td WIDTH =
30%>" . mysql_result($id,0,4) . " % </td>" ;
echo '<td WIDTH = 30%><input type ="text"
name="carboratio" size = 10></td></tr>' ;
echo " <tr bgcolor = " . $a. " ALIGN =CENTER><td
WIDTH = 30%>Percentage of your daily needs
taken during the breakfast Meal (recom.:
25%)</td><td WIDTH =
30%>" . mysql_result($id,0,5) . " % </td>" ;
echo '<td WIDTH = 30%><input type ="text"
name="breakratio" size = 10></td></tr>' ;
echo ' <tr ALIGN = CENTER>

```

```

<td WIDTH = 30%></td>
<td align=CENTER width="30%"> </td>
<td ALIGN= RIGHT>
<input type="submit" name="ACTION"
value="Submit">
</td></tr>' ;
echo "</table>" ;

```

```

echo "<BR>" ;

// display the current daily limits

echo "<table width=70% cols=3 ALIGN = CENTER>" ;
echo " <tr COLSPAN = 2 ALIGN = CENTER
><H2>Personal Recommendations</H2></tr>" ;
echo '<tr bgcolor = ' . $c. ' ALIGN =
CENTER><TD WIDTH = 30%>Recommended daily
Calories Intake</td><TD WIDTH = 30%>' . $cal. '
ca</td><TD WIDTH = 30%></TD></tr>' ;

echo '<tr bgcolor = ' . $c. ' ALIGN =
CENTER><TD WIDTH = 30%>Daily Carbohydrates
intake </td><TD WIDTH = 30%>' . $carbo. '
g</td><TD WIDTH = 30%></TD></tr>' ;

echo '<tr bgcolor = ' . $c. ' ALIGN =
CENTER><TD WIDTH = 30%>Daily Fat
intake </td><TD WIDTH = 30%>' . (int)
($cal*0.3/9). ' g</td><TD WIDTH =
30%></TD></tr>' ;

echo '<tr bgcolor = ' . $c. ' ALIGN =
CENTER><TD WIDTH = 30%>Daily saturated Fat
intake </td><TD WIDTH = 30%>' . (int)
($cal*0.1/9). ' g</td><TD WIDTH =
30%></TD></tr>' ;

echo '<tr bgcolor = ' . $c. ' ALIGN =
CENTER><TD WIDTH = 30%>Daily Cholesterol
intake </td><TD WIDTH = 30%> 300 mg</td><TD
WIDTH = 30%></TD></tr>' ;

echo '<tr bgcolor = ' . $c. ' ALIGN =
CENTER><TD WIDTH = 30%>Daily Sodium
intake </td><TD WIDTH = 30%> 2400 mg</td><TD
WIDTH = 30%></TD></tr>' ;

```

```

echo '<tr bgcolor = ' . $c. ' ALIGN =
CENTER><TD WIDTH = 30%>Daily fiber
intake </td><TD WIDTH = 30%> 30 g</td><TD
WIDTH = 30%></TD></tr>' ;

```

```

echo "</table>" ;
echo "<BR>" ;

```

```

echo "<br><br>" ;

```

```

echo "
        <table width=70% cols=3 ALIGN = CENTER>
        <tr bgcolor=" . $c. "ALIGN = CENTER>
        <TD WIDTH = 30%
        ALIGN=CENTER>Nutrient</td>
        <TD WIDTH = 30%
        ALIGN=CENTER>Current Limit</td>
        <TD WIDTH = 30% ALIGN=CENTER>New
        Value</TD>
        </tr> " ;

```

```

$free_login = 'root' ;
$free_passwd = 'titp4md' ;

```

```

mysql_connect('localhost', $free_login, $free_pa
sswd) or die ("probleme connexion");
mysql_select_db('diabete') or die ("probleme
dans selection base");

```

```

$sql = "SELECT  p.limits, n.NutrDesc,
n.Units, n.Nutr_No FROM prof1 p, nutri n WHERE
p.Nutr_No=n.Nutr_No ";
$id = mysql_query ($sql) ;
$row = 0 ;
$outresult = "";
$back = 0 ;

while (mysql_fetch_row($id)) {
$name = mysql_result($id,$row,1) ;
$units = mysql_result($id,$row,2) ;
$num = mysql_result($id,$row,3) ;
$lim = ( int)
(mysql_result($id,$row,0)*10)/10;
$a = "#dddddd" ;
$b= "#ff6666" ;
$c="#cccccc" ;

if ($lim == 0) { $bg = $b ; }
    else {
        if ($back == 0)
            {
                $bg = $a ;
                $back = 1 ;
            }
        else
            {
                $bg = $c ;
                $back = 0 ;
            }
    }
$outresult = $outresult.'|
|  |

```

```

}
echo  "$outresult" ;
echo ' <tr ALIGN = CENTER>
<TD WIDTH = 30%></td>
<td align=CENTER width="30%"> </TD>
<TD ALIGN= RIGHT>
<input type="submit" name="ACTION"
value="Submit">
</td></tr> ' ;

echo "</table>" ;

?>

<br>
<br>
-----
-----

```

compare.php

```
-----  
-----  
  
<html>  
<head>  
    <title>Compare</title>  
</head>  
<body>  
  
<?php  
  
    // this script displays a list of products  
    ordered for a given nutrient  
    // the UPC codes and the choice of nutrient  
    are passed as GET variables in the URL  
    // URL: compare.php  
    // GET variables : $u1,..$un list of UPC codes  
    of the product to compare  
    //          $nutri   code of the  
    nutrient used to make the comparison  
    // tables used : product,  
    // access from : Palm Pilot  
  
    // access to the Mysql database  
    $free_login = 'root' ;  
    $free_passwd = 'titp4md';  
  
    mysql_connect('localhost',$free_login,$free_pa  
    sswd) or die ("cannot connect the database  
    server");  
    mysql_select_db('diabete') or die ("cannot  
    connect to the database");  
  
    $sql = 'SELECT n.NutrDesc FROM nutri n WHERE  
    Tagname= "'. $nutri. "' ;
```

```
$id = mysql_query ($sql) ;  
$nam= mysql_result($id,0,0) ;  
  
echo "<H1><CENTER>Content in ".$nam." <BR>  
from LOW to HIGH<BR></CENTER></H1>" ;  
  
//this part generates the Mysql query. It  
parses the url call where an unknown  
//number of barcods are passed as u1, u2, ...  
  
$req= 'SELECT label, servilab, '.$nutri.',UPC  
FROM product WHERE UPC = 0' ;  
$i=1 ;  
$a = 'u'.$i ;  
  
WHILE (isset(${$a})) {  
    $req=$req.' OR  
    UPC="'. $a. "' ;  
    $i++ ;  
    $a = 'u'.$i ;  
  
$req = $req.' ORDER BY '.$nutri.' ASC ' ;  
  
$id2= mysql_query ($req) ;  
  
//display the ordered list of the products.  
  
echo "<BR><BR>" ;  
echo "<table width = 100% cols=3>";  
  
$row = 0 ;  
while (mysql_fetch_row($id2)) {  
    echo "<TR><TD width = 50%>" ;  
    echo mysql_result($id2,$row,0) ;  
    echo "</TD>" ;  
    //link to call for quantitative details on  
    each product  
    echo "<TD><a href=  
'totalrequest1.php?bar=".mysql_result($id2,$ro  
w,3).'> info </a></TD>" ;
```

```

//link to register a purchase of a product
    echo "<TD><a href=
'history.php?bar=".mysql_result($id2,$row,3).
'> buy </a></TD>" ;
    echo "</TR>" ;
    $row++ ;
    }
echo "</TABLE>" ;

//button to call the welcome screen on the
palm pilot

echo "<a href='palm:OUxL.appl'
BUTTON>MAIN</a>" ;

?>

</body>
</html>

```

history.php

```

-----
-----

<html>
<head>
    <title>Products Gateway</title>
</head>
<body>

<?php

// this script update the purchase database
and/or the eat database
// if only a barcode is specified in the url,
then the purchase database is updated
// if an amount is specified in addition, then
the eat database is updated, as well
// as the purchase database (the same amount
is substracted to the purchase database
// URL: history.php
// access from : Palm Pilot
// GET variables : $bar Barcode of the product
to update
//
// $serv (optional) amount to
add to the meal plan and substract
//
// from the inventory
// tables accessed : purchase, eat

$free_login = 'root' ;
$free_passwd = 'titp4md';

mysql_connect('localhost',$free_login,$free_pa
sswd) or die ("cannot connect to the server");
mysql_select_db('diabete') or die ("cannot
connect to the database");

$today = date( "Ymd", time() );

```

```

//if serv is specified then the eat and
purchase database are updated

    if (isset($serv))
    {
        $req0 = "SELECT barcode FROM eat WHERE
barcode=$bar" ;
        $id = mysql_query($req0) ;
        $nb = mysql_num_rows($id) ;
        if ($nb > 0) {
            if ($serv == 0)
            {
                $sql = "DELETE FROM eat
WHERE barcode=$bar" ;
                $note = "The product has
been removed from your breakfast plan" ;
            }
            else
            {
                $sql = "UPDATE eat
SET date=$today, amount=$serv WHERE
barcode=$bar" ;
                $note = "Your
breakfast meal plan has been updated" ;
            }
        }
        else
        {
            $sql = "INSERT into
eat(date,barcode,amount) VALUES
($today,$bar,$serv)" ;
            $note = "Your breakfast meal
plan has been updated" ;
        }
        $sql2 = "UPDATE purchase SET serving =
serving - $serv WHERE barcode = $bar " ;
        $id2 = mysql_query ($sql2) ;
    }
else

```

```

{
    $sql0 = "SELECT servnumb FROM product
WHERE UPC=$bar" ;
    $id0 = mysql_query ($sql0) ;
    $qty = mysql_result($id0,0,0) ;
    $sql = "INSERT into
purchase(date,barcode,serving) VALUES
($today,$bar,$qty)" ;
    $note = "Your personal shelves have been
updated" ;
} ;

$id = mysql_query ($sql) ;
echo $note ;

//call to the welcome module in the palm pilot

echo " <br>      <a href='palm:OUxL.appl'
BUTTON>Main</a>      " ;

?>

</table></html>

```

```

-----
-----

```

newrequest.php

```
-----  
-----  
  
<html>  
<head>  
    <title>Advice</title>  
</head>  
<body>  
  
<?php  
  
// the script returns a personal  
recommendation on a product specified by its  
// barcode in the url query  
// GET variables : $bar barcode of the picked  
food product  
// tables accessed : prof1, nutri, product  
// Access from Palm Pilot  
  
// connection to the database  
  
$free_login = 'root' ;  
$free_passwd = 'titp4md';  
  
mysql_connect('localhost',$free_login,$free_pa  
sswd) or die ("cannot connect to the server");  
mysql_select_db('diabete') or die ("cannot  
connect to the database");  
  
// query that retrieves the daily limits of  
the user  
  
$sql = "SELECT p.Nutr_No, p.recom,p.limits,  
n.NutrDesc,n.Tagname FROM prof1 p, nutri n  
WHERE p.Nutr_No = n.Nutr_No ";  
$id = mysql_query ($sql) ;  
$row = 0 ;  
$flag = 0 ;  
$list = "" ;
```

```
$list2 = "" ;  
  
// query that retrieves the information on the  
products  
// this query uses the results of the previous  
one because only the nutrients that were  
mentioned  
// in the user profile are retrieved from the  
product database  
  
$req = "SELECT p.label, p.label" ;  
while (mysql_fetch_row($id))  
  
    {  
  
        $nutr[$row] = mysql_result($id,$row,0) ;  
        $recom[$row] = mysql_result($id,$row,1) ;  
        $lim[$row] = mysql_result($id,$row,2);  
        $nam[$row] = mysql_result($id,$row,3) ;  
        $tag[$row] = mysql_result($id,$row,4);  
        $req = $req.', p.'.$tag[$row] ;  
        $row++ ;  
  
    }  
  
$req = $req.',p.serv1, p.serv1lab, p.category  
FROM product p WHERE UPC ='.$bar ;  
$i = 0 ;  
$id = mysql_query ($req) ;  
$lab = mysql_result($id,0,0) ;  
$ser= mysql_result($id,0,$row+3) ;  
$serNum = mysql_result($id,0,$row + 2) ;  
$categ = mysql_result($id,0,$row + 4) ;  
  
//retrieve the recommended product for the  
category of the scanned product  
  
$req10 = "SELECT brand, label, comment, UPC  
FROM recom WHERE category ='.$categ.'" ;  
$id10 = mysql_query ($req10) ;  
$brandbest = mysql_result($id10,0,0) ;
```

```

$labelbest = mysql_result($id10,0,1) ;
$commentbest = mysql_result($id10,0,2) ;
$UPCbest = mysql_result($id10,0,3) ;

echo "<H1><CENTER>".$ser." of
".$lab."</CENTER></H1>" ;

While ( $i < $row ) {

$quant[$i] = mysql_result($id,0,$i + 2) ;

if ($lim[$i] == 0) { $flag = 1 ; // echo "
not allowed" ;
    $perc = 101 ;}
    else
        $perc =
$quant[$i]/$lim[$i]*100 ;

if ($perc > 25) { $flag = 1 ; // flag=1
if at least one nutrient amount is over
// 25% of
the dayly limit

    $list = $list.$nam[$i].
<br>" ;
    }
    else if ($perc > 15) // flag=2
if at least one nutrient amount is over
// 15% of
the dayly limit

    { $flag = 2 ;
    $list2 = $list2.$nam[$i]. <br>
" ;
    }

$i++ ;

}

echo "<br>"

```

```

echo "<br>" ;

if ($flag == 1 )
{
    echo "<H1><CENTER><BOLD>Better
not</CENTER></BOLD><H1>" ;
    echo "<br><BR>Too Much of :<BR><BR>" ;
    echo $list ;

// link to retrieve quantitative details on
the product

    echo "    <a
href='http://18.172.3.50/diabete/totalrequest1
.php?bar=$bar'BUTTON>Details</a> " ;

// link to register the purchase of the
product

    echo "    <a
href='http://18.172.3.50/diabete/history.php?b
ar=$bar'BUTTON>Buy</a> " ;

// link to the welcome module in the palm
pilot

    echo "<a href='palm:OUxL.appl'
BUTTON>MAIN</a>" ;
    echo "<DIV align = \"RIGHT\"> <a href
=#recom\">Advice </a></DIV> " ;
    echo "<BR><BR><BR><BR>" ;
    echo "<a
name=#recom\"><H1><CENTER>Take</a> less than
1 serving" ;

    } ;

if ($flag == 0 )

{
    echo
"<H1><CENTER><BOLD>YES</CENTER></BOLD><H1>" ;
}

```

```

        echo "<br><BR><BR>Good Choice<BR><BR>"
;
        echo " <a
href='http://18.172.3.50/diabete/totalrequest1
.php?bar=$bar'>BUTTON>Details</a> " ;
        echo " <a
href='http://18.172.3.50/diabete/history.php?b
ar=$bar'>BUTTON>Buy</a> " ;
        echo "<a href='palm:OUxL.appl'
BUTTON>MAIN</a>" ;
        echo "<DIV align = \"RIGHT\"> <a href
=#recom\">Advice </a></DIV>" ;
        echo "<BR><BR><BR><BR>" ;
    }

if ($flag == 2 )
{
    echo "<H1><CENTER><BOLD>OK,
BUT</CENTER></BOLD><H1>" ;
    echo "<br><BR><BR>Rich in :<BR><BR>" ;
    echo $list2;
    echo " <a
href='http://18.172.3.50/diabete/totalrequest1
.php?bar=$bar'>BUTTON>Details</a> " ;
    echo " <a
href='http://18.172.3.50/diabete/history.php?b
ar=$bar'>BUTTON>Buy</a> " ;
    echo "<a href='palm:OUxL.appl'
BUTTON>MAIN</a>" ;
    echo "<DIV align = \"RIGHT\"> <a href
=#recom\">Advice </a></DIV> " ;
    echo "<BR><BR><BR><BR>" ;
    echo "<a
name=\"recom\"><H1><CENTER>Control</a> your
serving size" ;
    }

if
(
$UPCbest==$bar) { echo "<BR> <a
name=\"recom\"><H1><CENTER>You</a> selected a

```

```

recommended product for your personal diet
!<BR> " ;}
        else { if ($flag == 0)
            {
                echo "<a
name=\"recom\"><H1><CENTER>You</a> may also
consider<BR>" ;
            }
            else echo " <BR> or
rather choose<BR>" ;

                echo
$labelbest.", ".$brandbest." ".$commentbest."."
;
            } ;

// link to go back to the scanner module in
the palm pilot

echo " <br><a href='palm:KSjR.appl'
BUTTON>New Scan</a> " ;

?>

</body>
</html>

```


totalrequest1.php

```
-----
-----

<html>
<head>
  <title>Details</title>
</head>
<body>

<?php

//This script gives the ratio of nutrient
contents for a picked product, based on the
users's daily limit
//URL : totalrequest1.php
//tables used : prof1, nutri, product
//GET variables : $bar barcode of the picked
product
//access from Palm Pilot

// access to the database

$free_login = 'root' ;
$free_passwd = 'titp4md';
mysql_connect('localhost',$free_login,$free_pa
sswd) or die ("probleme connexion");
mysql_select_db('diabete') or die ("probleme
dans selection base");

// query for the daily limits of the user

$sql = "SELECT p.Nutr_No, p.recom,p.limits,
n.NutrDesc, n.Tagname FROM prof1 p, nutri n
WHERE p.Nutr_No = n.Nutr_No ";
$id = mysql_query ($sql) ;
$row = 0 ;
$flag = 0 ;
$list = " " ;
```

```
// query for the nutrient content of the
picked product

$req = "SELECT p.label, p.servllab" ;
while (mysql_fetch_row($id))
{

  $nutr[$row] = mysql_result($id,$row,4) ;
  $recom[$row] = mysql_result($id,$row,1) ;
  $lim[$row] = mysql_result($id,$row,2);
  $nam[$row] = mysql_result($id,$row,3) ;
  $req = $req.',p.'. $nutr[$row] ;
  $row++ ;

}

$req = $req.',p.serv1, p.servllab FROM product
p WHERE UPC = '.$bar ;

//echo $req ;

echo "<br>" ;

$i = 0 ;
$id = mysql_query ($req) ;
$lab = mysql_result($id,0,0) ;
echo "<H3><CENTER>".$lab."</H1></CENTER>" ;
$ser = mysql_result($id,0,$row+3) ;
echo "<H3>Ratio of your daily serving
contained in " ;
$serNum = mysql_result($id,0,$row + 2) ;
echo $ser ;
echo "</H3> " ;
//echo $serNum ;
//echo " grammes " ;
echo "<br>" ;

While ( $i < $row ) {

// echo $nutr[$i] ;
echo "<H3><HR>" ;
echo $nam[$i]." " ;
```

```

$quant[$i] = mysql_result($id,0,$i + 2) ;

if ($lim[$i] == 0) { $flag = 1 ; echo " not
allowed " ; $perc = 101 ;}
else
    $perc = $quant[$i]/$lim[$i]*100 ;

if ($perc > 100) { $flag = 1 ;
    $list = $list.$nam[$i]."
";
    }

echo(int)($perc) ;
echo " % </H3>" ;
$i++ ;
    }
$ser= mysql_result($id,0,$row+2) ;
echo "<br>" ;
echo "<hr>" ;
echo "      <a
href='http://18.172.3.50/diabete/history.php?b
ar=$bar'BUTTON>Buy</a> " ;
    echo "<a href='palm:OUxL.appl'
BUTTON>MAIN</a>" ;
?>

</body>
</html>

```

<p>Java Client Code WABA Version 1.0</p>
--

Compare.java

```
-----  
-----  
  
// send a list of UPC codes and a nutrient to  
the server  
// connect to compare.php  
  
import waba.ui.*;  
import extra.ui.*;      // Libraries contained  
in SuperWaba classes  
import waba.fx.*;  
import waba.sys.*;  
import waba.io.*;  
  
public class Compare extends ExtraMainWindow  
{  
  
    Button readButton;  
    Button sendButton, menuButton;  
    Edit edit;  
    int m = 0 ;  
    String string ;  
    Label nom ;  
    PushButtonGroup pbg ;  
    Graphics g ;  
    SerialPort port;  
    String C = "" ;  
    String D = "" ;  
    Font font = new Font("Helvetica", Font.BOLD,  
14);
```

```
public Compare()  
{  
  
    // display the first line  
  
        setDoubleBuffer(true);  
        Label l;  
        l = new Label("product(s) in the list");  
        l.set3d(true);  
        l.setFont(font) ;  
        l.setRect(35,20,PREFERRED,PREFERRED);  
        add(l) ;  
  
    // display the initial number of products scan  
(0)  
  
        Label Num ;  
        D = "" + m ;  
        nom = new Label(D);  
        nom.set3d(true);  
        nom.setFont(font) ;  
        nom.setRect(25,20,PREFERRED,PREFERRED);  
        add(nom) ;  
  
    // display the 6 buttons associated with each  
nutrients  
  
        pbg = new PushButtonGroup(new  
String[]{"Chol.", "Carb.", "Fat", "Sodium", "Sat.  
Fat", "Fiber"}, true, 0, 3, 10, 3, true, PushButtonGro  
up.NORMAL) ;  
        pbg.maxWidth = 70 ;  
        add(pbg);  
        pbg.setFont(font);  
  
        pbg.setRect(10,50,pbg.getPreferredWidth(),PREF  
ERRED) ;  
  
    // display the button that triggers a scan
```

```

readButton = new Button("Add");
add(readButton,LEFT,120);

// display the button used to launch a
wireless connection and send the scanned
barcodes

sendButton = new Button("Send");
add(sendButton,RIGHT,120);

// display the button used to return to the
"welcome" screen

menuButton = new Button("Main");
add(menuButton,CENTER,135);
}

public void onEvent(Event e)
{
String nutrient = "" ;
byte[] buf = new byte[120];
int n = pbg.getSelected() ;

if (e.type == ControlEvent.PRESSED)
{
if (e.target == readButton)
{
int count = 0;

// popup box while the program is waiting for
a scan

MessageBox mb = new
MessageBox("scan waiting...", "Scan a barcode|a
brief sound signals|a
successfull|aquisition",null);

```

```

mb.setUnpopDelay(500);
popupModal(mb);

// open the serial port for 10 seconds and
store up to 13 bytes (the lenght of a standard
UPC code)

port = new SerialPort(0, 9600, 7,
true, 1);
port.setReadTimeout(10000);
count = port.readBytes(buf, 13*m,
13);
port.close() ;
{
StringBuffer stringBuffer =
new StringBuffer();
StringBuffer byteBuf =
new StringBuffer();

if (count == 0) { ; }

else
{
// convert the barcode in a string

for (int i = 2+13*m; i <
13*m+count - 1 ; i++)
{
stringBuf.append((char)buf[i]);
}

count = m + 1 ;
String dataString =
stringBuf.toString();
C = C + "&u" + count + "="
+ dataString ;

m = m + 1 ;
D = "" + m ;
nom.setText(D) ;

```

```

        }
    }

    // build the url query that contains the
    // nutrient chosen and a list of barcodes

    if (e.target == sendButton)
    {
        if (n==0) nutrient
        = "chol" ;
        if (n==1) nutrient
        = "carbo" ;
        if (n==2) nutrient
        = "fat" ;
        if (n==3) nutrient
        = "sodium" ;
        if (n==4) nutrient
        = "satfat" ;
        if (n==5) nutrient
        = "fiber" ;

        String theURL =
        "http://18.172.3.50/diabete/compare.php?nutri="
        + nutrient + "&" + C ;

        // display the result of the web query in the
        // Clipper application

        Vm.exec("Clipper",
        theURL, 54, false);
    }
}
}
}

```

record.java

```

-----
// used to scan an item and specify an amount
// and sent the UPC code and
// the amount to the server, in order to
// update the meal plan
// connect to history.php

import waba.fx.*;
import waba.sys.*;
import waba.util.*;
import waba.ui.*;
import waba.io.*;

public class Record extends MainWindow
{
    String version="Version. ", serv="", D, C;
    String ml,req ;
    Button btnHi;
    Edit ed;
    ComboBox cbl,
    SerialPort port;
    PushButtonGroup pbg;
    TabPanel tp;
    Button btnMem,sendButton, menuButton;
    Label lmem, lval, ltext;

    public Record()
    {
        String []items = {"0.5 serving","1
        serving","1.5 serving", "2 serving","2.5
        servings","3 servings","3.5 servings","4
        servings","4.5 servings","5 servings"};
        Label l;
        add(l = new Label("Breakfast Meal
        Plan"));
        l.set3d(true);
        l.setRect(CENTER,10,PREFERRED,PREFERRED);
    }
}

```

```

        add(lmem = new Label("Press Scan to Pick
up a product"));
        l.set3d(true);
        lmem.setRect(CENTER,25,150,PREFERRED);

        add(lval = new Label("select a
product..."));
        lval.setRect(30,45,170,PREFERRED);
        lval.setVisible(false);

        add(btnHi = new Button("Scan"));
        btnHi.setRect(80,40,70,30);

        add(ltext = new Label("Choose a number of
servings :"));

        ltext.setRect(10,70,PREFERRED,PREFERRED);

        add(cb1 = new ComboBox(items));
        cb1.setRect(10,85,60,30);

        add(sendButton = new Button("Send"));
        sendButton.setRect(80,85,70,30);

        menuButton = new Button("Main");
        menuButton.setRect(80, this.height -30,
70, 30);
        add(menuButton);
    }

```

```

// Called by the system to pass events to the
application.

```

```

public void onEvent(Event event)
{
    byte[] buf = new byte[120];

```

```

        if (event.type ==
ControlEvent.PRESSED && event.target ==
menuButton)
        {
            // link to the welcome menu
            Vm.exec("Welcome", null ,
0, false);
        }

        if (event.type ==
ControlEvent.PRESSED && event.target == btnHi)
        {
            // open the serial port and wait for a scan
            for 5 sec

                int count = 0;
                MessageBox mb = new
                MessageBox("scan waiting...", "Scan a barcode|a
                brief sound signals|a
                successfull|aquisition",null);
                mb.setUnpopDelay(500);
                popupModal(mb);

                port = new SerialPort(0, 9600,
7, true, 1);
                port.setReadTimeout(5000);
                count = port.readBytes(buf, 0,
13);

                port.close();
                StringBuffer stringBuffer = new
                StringBuffer();
                StringBuffer byteBuf = new
                StringBuffer();

                if (count == 0) { ; }

                else
                {
                    // retrieve the chain from the buffer
                    for (int i = 2; i <
count - 1 ; i++)

```

```

        {
            stringBuf.append((char)buf[i]);
        }

        String
dataString = stringBuf.toString();
        C = dataString
;
        D= "Product
acquired ok" ;

lval.setText(D) ;
lval.setVisible(true);
lmem.setVisible(false) ;
btnHi.setVisible(false) ;

    }

    if (event.type ==
ControlEvent.PRESSED && event.target ==
sendButton)
    {
        String m1 =
cb1.getSelectedItem();
        double n1 =
cb1.getSelectedIndex() ;

//translate the choice of number of servings

        if (n1 == 0) serv = "0.5" ;
        if (n1 == 1) serv = "1" ;
        if (n1 == 2) serv = "1.5" ;
        if (n1 == 3) serv = "2" ;
        if (n1 == 4) serv = "2.5" ;
        if (n1 == 5) serv = "3" ;
        if (n1 == 6) serv = "3.5" ;

```

```

        if (n1 == 7) serv = "4" ;
        if (n1 == 8) serv = "4.5" ;
        if (n1 == 9) serv = "5" ;

// generate the url query with a barcode and a
number of serving

        String theURL =
"http://18.172.3.50/diabete/history.php?bar="
+ C + "&serv=" + serv ;
        Vm.exec("Clipper",
theURL, 54, false);
    }
}

```

Review.java

```
-----  
-----  
  
// allow to review either the meal plan or the  
personal inventory  
// stored in the server  
// connect to rewiw2.php or reviewshelves.php
```

```
import waba.ui.*;  
import waba.fx.*;  
import waba.io.*;  
import waba.sys.*;
```

```
public class Review extends MainWindow  
{  
    Button scanButton;  
    Button compButton;  
    Button eatButton;  
    Button viewButton;  
    Button menuButton;  
    Timer timer;  
    Graphics g ;  
    SerialPort port;  
    String C = "Welcome" ;  
    String D = " to the Health Companion" ;  
    String theURL ;  
    public void onPaint(Graphics g)
```

```
    {  
  
        g.setColor(255, 255, 255); //<--white  
        g.fillRect(0, 0, 200, 150);  
        g.setColor(0,0,0) ;  
  
    }
```

```
    public Review()  
    {
```

```
        scanButton = new Button("Inventory");  
        scanButton.setRect(80, this.height -  
120 , 70, 30);  
        add(scanButton);
```

```
        compButton = new Button("Meal Plan");  
        compButton.setRect(80, this.height -80  
    , 70, 30);  
        add(compButton);
```

```
        menuButton = new Button("Main");  
        menuButton.setRect(80, this.height -  
30, 70, 30);  
        add(menuButton);  
    }
```

```
    public void onEvent(Event event)  
    {
```

```
        if (event.type == ControlEvent.PRESSED)  
        {  
            if (event.target == scanButton)  
            {  
                String theURL =  
"http://18.172.3.50/diabete/reviewshelves.php"  
;                Vm.exec("Clipper", theURL,  
54, false);  
            }
```

```
            if (event.target == compButton)  
            {  
                String theURL =  
"http://18.172.3.50/diabete/review2.php" ;  
                Vm.exec("Clipper", theURL,  
54, false);  
            }
```

```
            if (event.target == menuButton)  
            {
```

```

        Vm.exec("Welcome", null , 0,
false);
    }
}

```

Scanner2.java

```

// used to scan a product and obtain an advice
based on the personal information
// send an UPC code ($bar) as a GET variable
to the server
// connect to newrequest.php

```

```

import waba.ui.*;
import waba.fx.*;
import waba.io.*;
import waba.sys.*;

```

```

public class Scanner2 extends MainWindow
{

```

```

    Button closeButton;
    Button readButton;
    Button stopButton;
    Graphics g ;
    SerialPort port;
    String C = "Please" ;
    String D = " Press the Scan button" ;

```

```

public void onPaint(Graphics g)
{
    g.setColor(255, 255, 255); //<--white
    g.fillRect(0, 0, 200, 150);
    g.setColor(0,0,0) ;
    Font font = new Font("Helvetica",
Font.BOLD, 20);
    g.setFont(font);
    g.drawText (C , 40 , 45) ;
    g.drawText (D , 10 , 55) ;
}

```

```

public Scanner2()
{
    repaint() ;
    closeButton = new Button("Main");
    closeButton.setRect(80, this.height -
30 , 70, 30);
    add(closeButton);

    readButton = new Button("Scan");
    readButton.setRect(80, this.height -70
, 70, 30);
    add(readButton);
}

public void onEvent(Event event)
{
    g = createGraphics();
    g.setColor(0, 0, 0);
    Font font = new Font("Helvetica",
Font.BOLD, 14);
    g.setFont(font);
    byte[] buf = new byte[30];

    if (event.type == ControlEvent.PRESSED)
    {
        if (event.target == closeButton)
// link to the welcome menu
        {
            Vm.exec("Welcome", null , 0,
false);
        }

        if (event.target == readButton)
        {
//when the scan button is hit, a five seconds
window is open to wait for a successful scan

```

```

        MessageBox mb = new
MessageBox("scan waiting...", "Scan a barcode|a
brief sound signals|a
successfull|aquisition", null);
        mb.setUnpopDelay(500);
        popupModal(mb);
        int count = 0;
        port = new SerialPort(0,
9600, 7, true, 1);
        port.setReadTimeout(5000);
        count = port.readBytes(buf,
0, 13);

        port.close() ;
        StringBuffer stringBuffer = new
StringBuffer();
        StringBuffer byteBuf = new
StringBuffer();

        if (count == 0) { ; }
        else
        {
            for (int i = 2; i
< count - 1 ; i++)
            {
                stringBuffer.append((char)buf[i]);
            }
            String
dataString = stringBuffer.toString();
            C =
dataString ;

            g.drawText (C , 55 , 65) ;
//the UPC is sent as a GET variable to the URL
through the proprietary Clipper application on
//the palm VII

            String theURL
=
"http://18.172.3.50/diabete/newrequest.php?bar
=" + C;

Vm.exec("Clipper", theURL, 54, false);

```

```
    }  
  }  
}
```


Welcome.java

// root menu of the application. Provides access to the 4 modules

```
import waba.ui.*;  
import waba.fx.*;  
import waba.io.*;  
import waba.sys.*;
```

```
public class Welcome extends MainWindow  
{  
  Button scanButton;  
  Button compButton;  
  Button eatButton;  
  Button viewButton;  
  Button menuButton;  
  Graphics g ;  
  SerialPort port;  
  String C = "Welcome" ;  
  String D = " to the Health Companion" ;
```

```
  public void onPaint(Graphics g)  
  {  
    g.setColor(255, 255, 255); //<--white  
    g.fillRect(0, 0, 200, 150);  
    g.setColor(0,0,0) ;  
  }
```

```
  public Welcome()  
  {  
    scanButton = new Button("Scan");  
    scanButton.setRect(0, this.height -80 , 70, 30);  
    add(scanButton);  
  
    compButton = new Button("Compare");
```

```
compButton.setRect(80, this.height -120 , 70, 30);  
add(compButton);
```

```
viewButton = new Button("View");  
viewButton.setRect(0, this.height -120 , 70, 30);  
add(viewButton);
```

```
eatButton = new Button("Record");  
eatButton.setRect(80, this.height -80, 70, 30);  
add(eatButton);
```

```
}
```

```
public void onEvent(Event event)
```

```
{  
    if (event.type == ControlEvent.PRESSED)
```

```
        {  
            if (event.target == scanButton)
```

```
                {  
                    Vm.exec("Scanner2", null , 0, false);  
                }  
            }  
        }  
        if (event.target == compButton)
```

```
            {  
                Vm.exec("Compare", null , 0, false);  
            }  
            if (event.target == eatButton)
```

```
                {  
                    Vm.exec("Record", null , 0, false);  
                }  
            }  
            if (event.target == viewButton)
```

```
                {  
                    Vm.exec("Review", null , 0, false);  
                }  
            }  
        }  
    }
```

```
}
```
