Design Of Web Services And Mobile Device Applications
For Integrated Health Information System

By

Andrew C. Ferriere

B.E. Structural Engineering
Visvesvaraya Regional Engineering College, 2000

SUBMITTED TO THE DEPARTMENT OF CIVIL AND ENVIRONMENTAL
ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE
DEGREE OF

Master of Engineering
IN CIVIL AND ENVIRONMENTAL ENGINEERING

At the
Massachusetts Institute of Technology
June 2002

© 2002 Massachusetts Institute of Technology
All Rights Reserved

Signature of Author.................................................................

DEPARTMENT OF CIVIL AND ENVIRONMENTAL ENGINEERING
May 13, 2002

Certified By..............................................................................

JOHN R. WILLIAMS
ASSOCIATE PROFESSOR, CIVIL AND ENVIRONMENTAL ENGINEERING
THESIS SUPERVISOR

Accepted By........

ORAL BUYUKOZTURK
CHAIRMAN, DEPARTMENT COMMITTEE OF GRADUATE STUDENTS

BARKER LIBRARIES
Design Of Web Services And Mobile Device Applications For Integrated Health Information System

By

Andrew C Ferriere

Submitted To The Department Of Civil And Environmental Engineering On May 13th 2002 In Partial Fulfillment Of The Requirement For The Degree Of Master Of Engineering In Civil And Environmental Engineering

ABSTRACT

Imagine a single signon web-based infrastructure integrating all the disparate hospital legacy systems and provides a premier web-based multi-disciplinary point of care system, designed for healthcare professionals who need access to patient information from many different locations. This thesis proposes an outline for the project (eHealth) which provides a true information management system that reduces the cost of providing healthcare and increases the top line revenue of healthcare providers through the creation of information management applications. According to hospital and health news, the number one concern of hospital executives is how to use information systems to improve revenue management, reduce operating expenses, and comply with federal regulations.

This thesis attempts to deal with the above issues and provides a solution by creating web services and mobile web applications, which are used to provide a single web based interface for both patients and physicians. These applications can be rendered both on desktops as well as mobile wireless handheld devices such as the pocket PC. This project uses emerging technologies, such as Microsoft’s .NET Platform which will be explained in the subsequent chapters. This thesis explains the System Architecture and Design used for development of the project as well as the different Health Tools and other applications developed to assist physicians and patients for monitoring the patient’s health.

Thesis Supervisor: John R. Williams
Title: Associate Professor, Civil And Environmental Engineering
Acknowledgements

I would like to express my gratitude to the following people for this thesis:

- Dr. John R. Williams for his guidance and assistance on the thesis as well as on the M.Eng. Project throughout the year.

- My parents and my brother for always challenging me and driving me to reach my fullest potential.

- My M.Eng. Project partners Tashan Yen, Sakda Chaiworawitkul, Osamu Uehara, Glorimar Ripoll, Pai-Fang Hsiao for their support and encouragement.

- To the class of 2002 M.Eng. students and all my friends at MIT for making this an enjoyable year.
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAPTER 1: INTRODUCTION</td>
<td>7</td>
</tr>
<tr>
<td>CHAPTER 2: HEALTH CARE INDUSTRY OVERVIEW</td>
<td>9</td>
</tr>
<tr>
<td>2.1 INDUSTRY ANALYSIS</td>
<td>10</td>
</tr>
<tr>
<td>2.1.1 Structure</td>
<td>10</td>
</tr>
<tr>
<td>2.1.2 Stakeholders</td>
<td>11</td>
</tr>
<tr>
<td>2.2 INDUSTRY DEFINITION</td>
<td>11</td>
</tr>
<tr>
<td>2.2.1 Where does the money come from?</td>
<td>12</td>
</tr>
<tr>
<td>2.2.2 Where does the money go?</td>
<td>13</td>
</tr>
<tr>
<td>2.2.3 Growth</td>
<td>13</td>
</tr>
<tr>
<td>CHAPTER 3: THE HEALTHCARE INFORMATION SYSTEMS INDUSTRY</td>
<td>15</td>
</tr>
<tr>
<td>3.1 OVERVIEW</td>
<td>15</td>
</tr>
<tr>
<td>3.2 HOSPITAL INFORMATION SYSTEMS</td>
<td>17</td>
</tr>
<tr>
<td>3.3 ELECTRONIC DATA INTERCHANGE</td>
<td>18</td>
</tr>
<tr>
<td>3.4 THE INTERNET AS A MEANS OF CONNECTIVITY</td>
<td>19</td>
</tr>
<tr>
<td>CHAPTER 4: PROPOSED SOLUTION</td>
<td>20</td>
</tr>
<tr>
<td>4.1 EVIDENCE-BASED TREATMENT</td>
<td>20</td>
</tr>
<tr>
<td>4.2 OUTCOME-FOCUSED TREATMENT</td>
<td>20</td>
</tr>
<tr>
<td>4.3 BEST PRACTICE GUIDELINES</td>
<td>21</td>
</tr>
<tr>
<td>4.4 APPROACH BENEFITS</td>
<td>21</td>
</tr>
<tr>
<td>4.5 THE SOLUTION</td>
<td>22</td>
</tr>
<tr>
<td>CHAPTER 5: SYSTEM ARCHITECTURE AND DESIGN</td>
<td>25</td>
</tr>
<tr>
<td>5.1 BASIC ARCHITECTURE</td>
<td>25</td>
</tr>
<tr>
<td>5.2 SCENARIOS AND USE CASE DIAGRAMS</td>
<td>29</td>
</tr>
<tr>
<td>CHAPTER 6: .NET TECHNOLOGIES</td>
<td>36</td>
</tr>
<tr>
<td>6.1 WEB SERVICES</td>
<td>36</td>
</tr>
<tr>
<td>6.1.1 Web Services Solution Architecture</td>
<td>36</td>
</tr>
<tr>
<td>6.1.2 SOAP (Simple Object Access Protocol)</td>
<td>39</td>
</tr>
<tr>
<td>6.1.3 WSDL (Web Service Description Language)</td>
<td>41</td>
</tr>
<tr>
<td>6.1.4 UDDI (Universal Description, Discovery, and Integration)</td>
<td>43</td>
</tr>
<tr>
<td>6.2 ASP.NET</td>
<td>44</td>
</tr>
<tr>
<td>6.3 ADO.NET</td>
<td>46</td>
</tr>
<tr>
<td>ADO.NET ARCHITECTURE</td>
<td>47</td>
</tr>
<tr>
<td>CHAPTER 7: SECURITY ISSUES IN HEALTHCARE INDUSTRY</td>
<td>48</td>
</tr>
<tr>
<td>7.1 HIPAA (HEALTH INSURANCE PORTABILITY AND ACCOUNTABILITY ACT)</td>
<td>48</td>
</tr>
<tr>
<td>7.2 SECURITY IMPLEMENTATION</td>
<td>48</td>
</tr>
<tr>
<td>7.2.1 Authenticating Users</td>
<td>49</td>
</tr>
<tr>
<td>7.2.2 Catching Time-Outs</td>
<td>52</td>
</tr>
<tr>
<td>7.2.3 Keep Data in the Dark (SSL)</td>
<td>53</td>
</tr>
</tbody>
</table>
CHAPTER 8: APPLICATIONS FOR MOBILE DEVICES ..............................................56
8.1 A NEW MEDICAL REVOLUTION .......................................................................56
8.2 WIRELESS COMPUTING BASICS ..................................................................56
8.3 MOBILE WEB APPLICATION ARCHITECTURE ..............................................60

CHAPTER 9: FURTHER DEVELOPMENTS .............................................................67
9.1 DATA MINING ..................................................................................................67
9.2 INTEGRATION WITH EXISTING LEGACY SYSTEMS ....................................68
9.3 FURTHER APPLICATIONS FOR MOBILE DEVICES .....................................69

CHAPTER 10: REFERENCES ..................................................................................71

APPENDIX A: SCREEN SHOTS FOR DESKTOP APPLICATIONS .........................72
A.1 FOR PATIENT’S ..............................................................................................72
A.2 FOR PHYSICIAN’S .........................................................................................76

APPENDIX B: SCREEN SHOTS FOR MOBILE APPLICATIONS .............................78
List of Figures

Figure 2.1 Health Expenditures as % of GDP ................................................................. 9
Figure 2.2 Structure of Health Care Industry ................................................................. 10
Figure 2.3 Stakeholders in Health Care Industry ........................................................... 11
Figure 2.4 Where does the money come from ............................................................... 12
Figure 2.5 Where does the money go? ....................................................................... 13
Figure 2.6 Projected Health Care Expenditures ............................................................ 14
Figure 2.7 IT expenditures within hospitals ................................................................. 14
Figure 3.1 HIS Market Segmentation .......................................................................... 16
Figure 3.2 Revenue Outlook for the HIS Industry ....................................................... 19
Figure 4.1. Off-Site and On-Site Database Interaction ................................................. 20
Figure 5.1 Basic Architecture ..................................................................................... 25
Figure 5.2. Web Service Overview ............................................................................. 27
Figure 5.3 Front End Application Back End Model ..................................................... 28
Figure 5.4. Use Case Diagram .................................................................................... 30
Figure 5.5 Class Diagram ............................................................................................ 35
Figure 6.1 Handling HTTP Requests ........................................................................ 38
Figure 6.2 Proxy Classes ............................................................................................. 39
Figure 6.3: Layering SOAP over HTTP ....................................................................... 40
Figure 6.4 Components of a WSDL document and how they relate to one another .... 42
Figure 6.5 ASP.NET Architecture ............................................................................. 45
Figure 6.6 ADO.NET Architecture ............................................................................. 47
Figure 7.1 Encryption using SSL ................................................................................ 55
Figure 8.1 Mobile Internet Toolkit ............................................................................. 61
Figure 8.2 Mobile Devices Accessing Mobile Application .......................................... 61
Figure 8.3 Identification of Device Capabilities ............................................................ 62
Figure 8.4 Compilation of Mobile.aspx page ............................................................... 63
Figure 8.5 Appropriate Generation of mark-up language .......................................... 64
Figure 8.6 WAP Browser Accessing Mobile Application ........................................... 64
Chapter 1: Introduction

The sharply growing health care industry is witnessing some significant trends that need to be addressed to retain industry stability. Particularly, controlling escalating costs and quality of patient care are keys to any effective strategy. These issues, together with the recent introduction of information technologies in the health care system and the rising wave of patient consumerism, are pushing the industry to adopt a new approach to health care. For any new approach to be successful, it must prevent the health care industry from reaching the unsustainable state that many experts predict for the near future.

Many traditional and modern disease-management companies have been successful at reducing costs and improving quality of care. However, leaders of the health care consulting practice expressed “no company has yet assembled anything that resembles a full platform of capabilities” (Matheson). The approach we envision will fill the existing gap in current market offerings.

The three elements of the approach in this thesis are: evidence-based treatment, outcome-focused treatment, and utilization of best practices. The vision is to transform hospitals into knowledge centers. By effectively and seamlessly integrating on-site and off-site knowledge, hospitals will witness benefits of cost reduction and higher patient satisfaction.

This thesis focuses on developing a system that includes a (1) web application with user-customized views, (2) a web service to facilitate database interactions, and (3) tools to manage and control access to the databases. (4) Mobile platform for rendering information on Pocket PC.
Chapter 1: Introduction

The solution is to provide a web-based infrastructure integrating all the disparate hospital legacy systems and provides a premier web-based multi-disciplinary point of care system, designed for healthcare professionals who need access to patient information from many different locations. This thesis proposes an outline for the project (eHealth) which provides a true information management system that reduces the cost of providing healthcare and increases the top line revenue of healthcare providers through the creation of information management applications. According to hospital and health news, the number one concern of hospital executives is how to use information systems to improve revenue management, reduce operating expenses, and comply with federal regulations. This project attempts to deal with the above issues and provides a solution by creating web services and mobile web applications which are used to provide a single web based interface for both patients and physicians. These applications can be rendered both on desktops as well as mobile wireless handheld devices such as the pocket PC.

This project uses emerging technologies, such as Microsoft’s .NET Platform which will be explained in the subsequent chapters. The following chapters will also explain the System Architecture and Design used for development of the project as well as the different Health Tools and other applications developed to assist physicians and patients for monitoring the patient’s health.
Chapter 2 : Health Care Industry Overview

The expenditure on medical resources is placing an ever-increasing burden on the society as a whole. National health expenditures are projected to reach $2.8 trillion in 2011, growing at a mean annual rate of 7.3 percent during the forecast period 2001-2011. During this period, health spending will grow 2.5 percent per year faster than nominal gross domestic product (GDP), so that by 2011 it will constitute approximately 17.0 percent of GDP compared to its 2000 level of 13.2 percent. This projection represents a 0.9 percentage point increase in GDP share by 2010 compared with last year's forecast.

Figure 2.1 Health Expenditures as % of GDP
2.1 Industry Analysis

In the US, IT expenditure in hospital was about $5bn in 1997, thus accounting for the largest proportion within the Healthcare Information System (HIS) market. However, this market is still very fragmented, although consolidation within the healthcare industry has resulted in increased demand for open platforms and interfaceable systems. The hospital segment is mainly using systems enhancing administrative and clinical efficacy, but it is also starting to realize the value of electronic patient records.

2.1.1 Structure

For the sake of simplicity, the healthcare model is comprised of patient, provider, payer, government and pharma / biotechnology. Each of these can be broken down into numerous sub structures and there is a great deal of overlap.

![Figure 2.2 Structure of Health Care Industry](image)
2.1.2 Stakeholders

The primary stakeholders are derived from the structure above.

![Stakeholders in Health Care Industry](image)

Figure 2.3 Stakeholders in Health Care Industry

2.2 Industry Definition

Three powerful and conflicting forces dominate the trajectory of the health care system. The first and most fundamental is the continuing pressure to adopt new technologies while moderating the economic burden on taxpayers, employers, and consumers. New technologies derive from a broader accumulation of scientific and engineering knowledge, from advances in physics, pharmacology, and pathology that highlight opportunities for intervention in the mechanisms of disease, trauma recovery, and repair. These advances do not remain under the exclusive purview of scientific or political elites but are communicated widely to the citizenry, generating strong demand for their
immediate diffusion. However, this enthusiastic embrace of new clinical interventions is not accompanied by a commensurate commitment on the part of the public to pay for them. The increasing wealth of society permits ever-growing investments in health care and it is to be assumed that expenditures will pace the overall growth in the economy. However, even the wealthiest of nations cannot continue on a trajectory that would devote 15, then 20, and then 25% of total resources to health care. The limits on social willingness to pay manifest themselves in taxpayer revolt, in labor market tradeoffs between wages and fringe benefits and in the tens of million of citizens who lack even the most basic of insurance coverage.

2.2.1 Where does the money come from?

![Figure 2.4 Where does the money come from](image_url)
2.2.2 Where does the money go?

![Pie chart showing health care expenditure categories.]

Figure 2.5 Where does the money go?

2.2.3 Growth

National health expenditures are projected to reach $2.8 trillion in 2011, growing at a mean annual rate of 7.3 percent during the forecast period 2001-2011. During this period, health spending is expected to grow 2.5 percent per year faster than nominal gross domestic product (GDP), so that by 2011 it will constitute approximately 17.0 percent of GDP compared to its 2000 level of 13.2 percent.
$500Bn goes to hospitals and IT expenditures within those hospitals account for $13Bn, $56Bn by 2004.
Chapter 3: The Healthcare Information Systems Industry

3.1 Overview

The healthcare information system (HIS) industry is currently growing rapidly, with an expected future annual growth rate of about 15% over the next years. The trend within the industry, now tending towards open architectures and client/server systems, is most likely to gain in importance and will shape the product portfolio of HIS vendors. The HIS market can be segmented according to end-users and thus splits into hospitals, physician groups and payers/managed care organizations.

In the US, IT expenditure in hospitals was about $5bn in 1997, thus accounting for the largest proportion within the HIS market. However, this market is still very fragmented, although consolidation within the healthcare industry has resulted in increased demand for open platforms and interfaceable systems. The hospital segment is mainly using systems enhancing administrative and clinical efficacy, but it is also starting to realize the value of electronic patient records (EPRs).

Physicians invested about $2bn in IT systems in 1997, which was below the industry average. Their main needs lie within administrative systems and small scale medical records. With the trend towards managed care-style concepts, these will however require more sophisticated systems. HIS vendors specialized in this market segment are currently very fragmented in their product range.

The payer segment has realized IT as a differentiating factor for competitiveness and invested approximately $5bn in IT systems in 1997. This market segment is less...
fragmented due to the need for systems fitting into a managed care framework, and its growth prospects are likely to be very good in the future.

With the growth of integrated healthcare networks, the need for interfaceable systems and open architectures will rise substantially, including cross departmental applications such as EPRs.

The major players within the HIS market, including HBOC & Company, Cerner Corporation, IDX and Shared Medical Systems, all share the key characteristic of providing integrated or widely applicable systems which are interfaceable and can offer refined functions such as decision support.

Figure 3.1 HIS Market Segmentation
3.2 Hospital information systems

In 1997, the expenditure on information systems throughout hospitals in the US was estimated to be about $5bn. Although the hospital environment currently accounts for the largest proportion of HIS expenditure within the HIS market as defined above, it is rather slow at fully adopting new technological offerings. Although this sector was one of the first to adopt IT for financial and administrative purposes, it has traditionally been very fragmented and focused on local conditions. The trend to consolidate and bring together different departmental functions has, however, resulted in increased demand for broad product offerings, including open platforms and interfaceable systems.

The usefulness and richness of EPRs can be enhanced by the use of some of the following features:

- dictation and transcription systems;
- portable systems that facilitate data entry at the point of care;
- decision support systems, which provide medical advice based on a patient’s previous medical history by relating this information to comprehensive databases.

Laboratory systems

These are closely related to clinical systems in that they also record events connected to patients. However, they are mainly concerned with test results and laboratory work processes.

Pharmaceutical systems

Pharmaceutical systems are associated with drug formularies and a hospital’s own inventory of drugs. They help to track the use of certain drugs, as well as the charges that
can be passed on to the patient or payer organisations. However, this area of application is rather insignificant and its use across different hospitals takes very different forms.

*Radiology systems*

This application serves the need to store records of radiology departments in an automated fashion, and thus includes capacities for film indexing and patient data storage. The capture and transmission of radiological images in a digital format is, however, still one of the more unexploited applications and could offer a good market opportunity in the future.

### 3.3 Electronic data interchange

EDI means the exchange of information over computers without any human interaction, which leads to lower costs and more speedy and accurate processes. It is generally used to submit claims and other transactions between healthcare providers and payers. This form of data exchange can be divided into two categories, namely:

- real-time EDI;
- batch processing.

Whilst real-time EDI provides an instant reply to any submission of data, batch processing is less prompt. It will be used for processing of less urgent claims, which are usually sent collectively to clearing houses and are then sorted later on. By contrast, real-time EDI is used if an immediate answer is required, for example to check eligibility and obtain authorisations.

Within the physician sector, for instance, EDI is still in its primary stages, but the advantages of automation will lead to increased demand for this technology. Its use
lowers administrative costs and improves the quality of data, especially in central data repositories. Another sector which might profit from this sort of data transmission are pharmacies.

3.4 The Internet as a means of connectivity

Within the HIS market, the Internet is beginning to emerge as a possible means of connection between payers and providers. It forms an inexpensive and easily accessible transmission channel of information between disparate sites. Even though it still only takes a small proportion of HIS applications, it will most likely become an important factor in the near future. The HIS industry is currently growing at a rapid pace, with an expected future annual growth rate of about 15% over the next years. This development has been fuelled by cost containment pressures and the need to improve efficiency within the healthcare delivery segment. Figure 3.2 illustrates the expected revenue development for the HIS industry to the turn of the century.

![Figure 3.2 Revenue Outlook for the HIS Industry](image)
Chapter 4: Proposed Solution

The three main elements adopted in this project are: (1) emphasis on evidence-based treatment, (2) outcome-focused treatment, and (3) utilization of best practice guidelines.

4.1 Evidence-based treatment

This element emphasizes ongoing health maintenance and feedback from health professionals. Traditionally, patient’s information is collected only when the patient is on-site, i.e., at the hospital. Likewise, feedback is only received on-site. Evidence-based treatment allows the patient to transmit and receive information to and from the physician regularly (even when the patient is off-site). Figure 4.1 illustrates the proposed flow of information to and from the medical database. Prevention and proper monitoring of patient conditions will reduce costly hospital visits.

![Patient Off Site Database Patient On Site](image)

Figure 4.1. Off-Site and On-Site Database Interaction

4.2 Outcome-focused treatment

Traditionally, physicians work on optimizing a particular individual procedure or event. They deal with medical events as if they were unrelated to other events, because they want to focus on solving the problem at hand. Through this systematic approach, physicians will focus on optimizing the overall outcome of the patient, not just individual sessions.
4.3 Best practice guidelines

To develop best practice guidelines, it is necessary to analyze the utilization of resources in a given situation over a number of episodes. The database needs to be effectively analyzed to find the most effective practice in any given situation. Physician may choose to deviate from the guidelines based on professional judgment. In any case these guidelines will promote a consistent level in care delivery.

4.4 Approach Benefits

By implementing these three elements, a control can be exercised on the number of serious cases that arrive at the hospital. A system that monitors patients off-site will help prevent, diagnose, and treat conditions to avoid reaching the critical level that becomes so costly for health care providers. For the patients, it is clear that they will benefit from constant attention from doctors. Physicians will be able to reduce the high workload associated with severe cases, allowing them to maintain a high constant quality of care. Furthermore, on a macro level, for hospital operators and other industry players, the approach will promote best practices to control quality of care and escalating costs.
4.5 THE SOLUTION

The solution which this thesis focuses on is to provide a fully scalable, extendable, integrated web based solution in an increasingly complex healthcare environment

- Provides patients with:
  Access to personal details, responsive communication with physician.

- Provides physicians with:
  Interface for point-of-contact note taking and remote access to patient data

PATIENT FEATURES

- Web based internet interface
- Online Health Monitoring Tools
- Email Access to primary physician
- Scheduling of future visits
- Access to laboratory studies, diagnostic testing, outside consultations
- New patient registration
- Update baseline information, social history, interim medical history, changes in family history, new medications
- Access to registration, update and questionnaires
- Health Information
- Personalized Content as well as on-line sources
Chapter 4: Proposed Solution

PHYSICIAN FEATURES

- Web based internet interface.
- Graphical user interface aids the clinician in recording encounter data.
- Alerts regarding potentially problematic interactions.
- General Examination of Patient.
- Order Prescriptions.
- Order Medical and Lab Tests.
- View the patient medical graphs.
- Monitor Diabetes and Heart Symptoms.
- View Patients History of Visit.
The effect of the solution on the various stakeholders is summarized in the following table:

<table>
<thead>
<tr>
<th></th>
<th>Benefits</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Patients</strong></td>
<td>• Increased time with doctor</td>
<td>• Privacy/security concerns</td>
</tr>
<tr>
<td></td>
<td>• Better quality of care</td>
<td>• Data input</td>
</tr>
<tr>
<td></td>
<td>• Decreased time in clinic</td>
<td>• Familiarity/Training</td>
</tr>
<tr>
<td></td>
<td>• Education</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Convenience</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Empowerment</td>
<td></td>
</tr>
<tr>
<td><strong>Physicians / Medical Asst.</strong></td>
<td>• Better quality of care</td>
<td>• Training</td>
</tr>
<tr>
<td></td>
<td>• Increased workflow</td>
<td>• Inconvenience</td>
</tr>
<tr>
<td><strong>Administration</strong></td>
<td>• Increased workflow</td>
<td>• Increased help desk function</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Registering patients</td>
</tr>
<tr>
<td><strong>Labs</strong></td>
<td>• Test administration accuracy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Increased workflow</td>
<td></td>
</tr>
<tr>
<td><strong>Pharmacy</strong></td>
<td>• Reduced Rx errors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Increased workflow</td>
<td></td>
</tr>
<tr>
<td><strong>Health Plan</strong></td>
<td>• Billing</td>
<td></td>
</tr>
<tr>
<td><strong>Researchers</strong></td>
<td>• Information</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 5 : System Architecture And Design

5.1 Basic Architecture

The basic architecture for this project is illustrated in Figure 5.1. In this project a localized model was adopted under which each hospital maintains its own database. This project also take into consideration that some hospitals may already employ certain legacy database systems to manage basic data like patient’s profiles.

Figure 5.1 Basic Architecture
The project is designed using a 3-tiered architecture comprising of:

- **The User Interface Tier**
  
  Implemented using ASP.NET

- **The Business Logic Tier**

  Implemented using Web Services

- **The Data Base Tier**

  The transactions between layer 2 and 3 are implemented using ADO.NET and database is MS SQL Server

These technologies will be discussed in the subsequent chapters.

The main application logic is implemented using Web Services which communicate with the data base layer and render information to the end user based on the user’s request. Web services are built using XML, HTTP and SOAP which provide a powerful mechanism for Remote Procedure Call (RPC). Some of the important benefits of using web services are:

- Web services aren't tied to a particular programming language, business application, or online service.

- Gives end users freedom to work on any access device they choose, from a powerful desktop to smart devices like mobile phones and handhelds.

- Uses XML for data exchange to help applications, services, and devices work together.
The web service is accessed by local clients (through intranet) as well as remote clients (through internet). These client applications will be built by creating .NET web applications. Using .NET technologies, the solution will be flexible enough to allow client applications to consume web services from different sources as well as render information on different client devices. Currently the architecture allows for both wired and wireless connections. Wireless connection will be very useful for mobile devices such as PDA’s. The goal is to allow doctors and nurses to have patients’ information with them when they make rounds and input data when they are away from their computers. Another important goal of using wireless technology is for paramedics to start taking down patient’s personal information and some preliminary assessment in
ambulances so that patients with urgent conditions can receive immediate treatment as soon as they arrive at the hospitals, since their data will already be entered.

Figure 5.3 Front End Application Back End Model
5.2 Scenarios and Use Case Diagrams

Using Unified Modeling Language (UML) terminology, a scenario is one thing that can happen. Each use case represents a set of scenarios tied together by a common goal. The use case diagram in Figure 5.4 connects nine different use cases and three actors. Each oval in the diagram represents a distinct use case.

Using UML terminology again, an actor is a role that a user plays with respect to the system. The main actors of this use case diagram are the (1) patient, (2) the doctor/nurse, and (3) the lab users. Following is a description of the role of each actor.

- **Lab User:** These users work in a unit at a Laboratory. Lab users are responsible for receiving lab orders, assigning lab resources, performing the tests, analyzing samples, and inserting test results to the system.

- **Doctors/Nurses:** These users provide care for a patient. They are responsible for issuing lab orders for patients, and examining lab results. They can also insert new patients’ information into the database.

- **Patient:** They can view and print statistical reports of their lab results and diagnosis. They can also update their information in the database.

A class diagram describes the types of objects that exist in the system and the different relationships that exist among them. A class diagram also illustrates the attributes and operations of a class and its constraints. Figure 5.5 shows the class diagram which was used for this particular use case diagram.
Figure 5.4. Use Case Diagram
Name: Lab Test Reports

Description: Doctors/Nurses may print a list of lab tests available in each laboratory for a particular patient. In addition basic information of each test may be viewed or printed. They may also review/print statistical information of tests performed.

Name: Make Lab Order

Description: Doctors/Nurses can make lab orders for the patient. The priority of the order may also be defined.

Postconditions: The lab order is visible in laboratory after the order has been added to the system. The lab order may be queued. Doctors/Nurses and lab users can follow the progression of the lab process.

Exception1: Lab worker may have marked some lab tests temporarily unavailable on his unit. The system then suggest another unit to the Doctors/Nurses user, if possible.

Exception2: Doctors/Nurses user may cancel the lab order before its completion.

Further development: The system or laboratory manager may make schedules for ordered tests, and the patient may be provided with a note of the time and place of the test. Also a interface to resource planning system of the laboratory may be provided.

Component: Lab process
Name: Patient reports

**Description:** Doctors/Nurses can view and print a *list of patients on a ward.* Doctors/Nurses or lab worker can print *personal and lab information of a patient* to be taken care of. Doctors/Nurses user can also print a *list of lab results of patients on the ward for selected date(s).*

Name: Perform lab test

**Description:** Lab worker encounters the patient and performs the test which has been assigned to him. The sample may then be assigned with an analyzer or the lab worker may analyze the sample himself.

**Preconditions:** Lab test has been assigned with a performer and time according to "Receive lab order" use case.

**Postconditions:** The lab test has been performed and the sample in sent for analysis.

**Exception:** It is possible to cancel lab order at this point, for example if the patient does not show up.

**Further development:** The system may print stickers containing the patient and test information for analysis.

**Component:** Lab process
### Name: Receive lab order

**Description:** Lab user takes lab orders from the queue, according to its priority, confirms the availability of needed resources and assigns time and performer of the test.

**Preconditions:** The lab order has been added to the system according to use case ”Make lab order”.

**Postconditions:** The lab order has been assigned with the lab worker to perform the test and time.

**Exception:** If all resources are not available, it is possible to transfer lab order to different laboratory or unit which is able to complete the test.

**Further development:** See ”Make lab order”, further development

**Component:** Lab process

---

### Name: Register lab result

**Description:** Lab user registers the lab result of a patient to the system. After the result has been registered, the department and Doctors/Nurses user responsible of the lab order are notified and can view the lab result.

**Preconditions:** The patient has been tested according to ”Perform lab test” use case, the specimen has been analyzed and results are available.

**Postconditions:** The lab result of the patient has been registered to the system and lab process as been completed.

**Component:** Lab process
Name: Update lab tests

Description: Lab user may insert and modify lab tests and their descriptions. He can also assign lab tests for his unit and mark tests of his unit temporarily unavailable.

Component: Lab test

Name: Update Profile

Description: Doctors/Nurses user can insert new people to the system, update patients’ personal information and assign patients to a ward. Patients are assigned to at most one ward at a time. Patient can also update their personal information.

Preconditions: Patient information can be viewed by an employee who is participating in the care on a unit or a laboratory. Patient information can be changed by participating Doctors/Nurses user or by patient himself.

Further development: Patient transfers, admissions, discharges.

Component: Person
Name: View lab results

Description: Doctors/Nurses user providing care for the patient may view and print lab results of the patient.

Preconditions: The test result has been added to the system according to “Register lab result” use case.

Component: Lab process

Figure 5.5 Class Diagram
Chapter 6 .NET Technologies

6.1 Web Services

XML Web services are the fundamental building blocks in the move to distributed computing on the Internet. Web services provide a standards-based, open communication medium that is the obvious choice for sending and receiving data between different computing platforms and programming languages. Applications are constructed using multiple XML Web services from various sources that work together regardless of where they reside or how they were implemented.

- XML Web Services expose useful functionality to Web users through a standard Web protocol. In most cases, the protocol used is SOAP.
- XML Web services provide a way to describe their interfaces in enough detail to allow a user to build a client application to talk to them. This description is usually provided in an XML document called a Web Services Description Language (WSDL) document.
- XML Web services are registered so that potential users can find them easily. This is done with Universal Discovery Description and Integration (UDDI).

6.1.1 Web Services Solution Architecture

One of the primary advantages of the XML Web services architecture is that it allows programs written in different languages on different platforms to communicate with each other in a standards-based way. SOAP is significantly less complex than earlier
approaches, so the barrier to entry for a standards-compliant SOAP implementation is significantly lower. The other significant advantage that XML Web services have over previous efforts is that they work with standard Web protocols—XML, HTTP and TCP/IP. A significant number of companies already have a Web infrastructure, and people with knowledge and experience in maintaining it, so again, the cost of entry for XML Web services is significantly less than previous technologies.

The only way to deal with the enormous numbers of heterogeneous entities on the Internet is to use the lowest common denominator. In other words, when bytes are transferred from one box to another, the process needs to use some standard that everyone on the Internet supports. The most common Internet transfer protocol is HTTP, which is used today by essentially all Web browsers to request the pages they display. The emerging cross-platform standard for encoding pure information transferred over HTTP is XML. Microsoft put these ideas together and developed the concept of a Web Service—a seamless way for objects on a server to accept incoming requests from clients using HTTP and XML. To create a Web Service, you simply write a Microsoft .NET server object as if it were going to be accessed directly by local clients, mark it with an attribute that says that you want it to be available to Web clients, and let ASP .NET do the rest. It automatically hooks up a prefabricated infrastructure that accepts incoming requests through HTTP and maps them to calls on your object, as shown in Figure 6.1. By rolling them into a Web Service, objects can work with anyone on the Web who speaks HTTP and XML, which should be everybody. The infrastructure for dealing with Web communication and calling remote objects is provided by the operating system.
On the client side, .NET provides proxy classes that have easy, function-based access to the Web Services provided by any server that accepts HTTP requests, as shown in Figure 6.2. A developer tool reads the description of the Web Service and generates a proxy class containing functions in whatever language you're using to develop the client. When the client calls one of these functions, the proxy class generates an HTTP request and sends it to the server. When the response comes back from the server, the proxy class parses the results and returns them from the function. This allows the function-based client to seamlessly interact with any Web server that speaks HTTP and XML.
6.1.2 SOAP (Simple Object Access Protocol)

Soap is the communications protocol for XML Web services. Once a Web service has been built, other clients/applications will want to invoke it. The Simple Object Access Protocol provides the standard RPC mechanism used for invoking Web services. It implies that the underlying Web service representation is an object when in fact it does not have to be. The SOAP specification provides standards for the format of a SOAP message and how SOAP should be used over HTTP. SOAP also builds on XML and XSD to provide standard rules for encoding data as XML.

SOAP is an application-level protocol so it can work directly over a transport protocol such as TCP. However, today’s Internet infrastructure is riddled with proxies and firewalls that typically allow only HTTP traffic. In order for all Internet-connected applications to communicate, SOAP must be able to flow over the current Internet
infrastructure including firewalls and proxies. To achieve this SOAP can be layered over HTTP as shown in Figure 6.3.

![Diagram of Layering SOAP over HTTP](image)

**Figure 6.3: Layering SOAP over HTTP**

SOAP can be used over HTTP to enable application-to-application communications over existing Internet infrastructure with its firewalls and proxies.

Layering SOAP over HTTP means that a SOAP message is sent as part of an HTTP request or response which makes it easy to communicate over any network that permits HTTP traffic. HTTP is also a good choice because, just like Web browsers, it is pervasive on all computing platforms and devices.

To achieve platform independence and maximum interoperability, SOAP uses XML to represent messages exchanged between the client and the Web service. Like HTTP, XML is also pervasive and an XML parser can be found for nearly any computing platform. By leveraging HTTP and XML, SOAP provides application to application communications between applications running on any platform and connected over the existing Internet infrastructure.
6.1.3 WSDL (Web Service Description Language)

Web Service Description Language (WSDL) is an XML-based grammar for describing Web services, their functions, parameters, and return values. Being XML-based, WSDL is both machine and human readable. Some modern development tools can generate a WSDL document describing your Web service as well as consume a WSDL document and generate the necessary code to invoke the Web service.

In WSDL, a service exposes groups of operations (i.e. methods). Each group of operations is called a *portType* which is roughly analogous to an interface in the COM world. To invoke an operation, the client sends an input message and gets back an output message. The input message contains the data going to the service and the output message contains the data coming back from the service. Each item of data in a message is called a message part or simply part. The actual protocol used to invoke an operation and the actual format of the input and output messages are specified in a binding. The service itself is exposed to the world via one or more ports. Each port specifies two things: A network address where it’s located and the binding to use with this port. A service may be exposed via multiple ports each with a different binding.

Figure 6.4 shows the components of a WSDL document and how they relate to each other. The boxes show the containment relations and the arrows show the reference relations.
Figure 6.4 Components of a WSDL document and how they relate to one another.

A service contains one or more ports and each port references a binding. Each binding references a portType, the operations within that portType and the messages that make up each operation. Each portType contains zero or more operations. Each operation has an input and output message. Each message has zero or more parts and each part is of some data type. The part’s type could be an XSD built-in type such as xsd:int or it could be a custom simple or complex type that’s defined using XSD.
6.1.4 UDDI (Universal Description, Discovery, and Integration)

One of the primary potential uses of Web services is for business-to-business integration. For example, company X might expose an invoicing Web service that the company’s suppliers use to send electronic invoices. Similarly, a vendor V might expose a Web service for placing orders electronically. If company X wanted to purchase computer equipment electronically, it would need to search for all vendors who sell computer equipment electronically. To do this, company X needs a yellow pages-type directory of all businesses that expose Web services. This directory is called Universal Description, Discovery, and Integration or UDDI. UDDI is an industry effort started in September of 2000 by Ariba, IBM, Microsoft, and 33 other companies. Today, UDDI has over 200 community members.

Like a typical yellow pages directory, UDDI provides a database of businesses searchable by the type of business. You typically search using business taxonomy such as the North American Industry Classification System (NAICS) or the Standard Industrial Classification (SIC). You could also search by business name or geographical location. By having a pool of well-known service types, UDDI makes it possible to find out how to do electronic business with a company. This is the primary advantage UDDI has compared to other Web-based business directories.
6.2 ASP.NET

ASP.NET allows in writing dynamic, high-performance Web applications. ASP.NET pages have a variety of new extensions. A basic ASP.NET page uses .aspx as the filename extension. A Web Service uses .asmx. These new filename extensions are there for a reason. ASP.NET runs side by side with the existing ASP infrastructure. They don't share session state, application state, or anything else, so they will peacefully coexist on the same server. These new filename extensions are therefore required so that IIS can call the appropriate ISAPI filter to handle processing.

Figure 6.5 shows the new ASP.NET architecture. One of the main things is the distinction between managed and native code. Managed code is the term Microsoft uses for code that sits on top of the new .NET runtime. This code takes full advantage of the new .NET framework including garbage collection, simplified deployment, etc. One interesting thing to note here is that except for some host-specific code, almost all of the new ASP.NET is written using the .NET runtime in a new language called C#.

The second important thing is that ASP.NET supports multiple hosts. It is expected to run on IIS4 on Windows NT 4.0 as well as in IE 5.5 using a new offline feature called My Web. Finally ASP.NET has added the pluggable module and handler architecture. Using this architecture, it is quite easy a developer, to do most of the things that ASP.NET does internally without having to go through the pain of starting from scratch writing one's own ISAPI extension.
ASP.NET includes features such as compiled language support, an object model for the page, Server Controls, Web Services, xcopy deployment, new configuration features, new session state options, new caching, new authentication/authorization options, a new architecture, and improved availability.
6.3 **ADO.NET**

ADO.NET is comprised of classes found in the `System.Data` namespace that encapsulate data access for distributed applications. However, rather than simply mapping the existing ADO object model to .NET to provide a managed interface to OLE DB and SQL Server, ADO.NET changes the way data is stored and marshaled within and between applications. The primary reason ADO.NET redefines this architecture is that most applications developed today can benefit from the scalability and flexibility of being able to distribute data across the Internet in a disconnected fashion.

ADO.NET is built from the ground up for distributed applications used in today's disconnected scenarios. For example, the central class in ADO.NET is the `DataSet`, which can be thought of as an in-memory XML database that stores related tables, relationships, and constraints. The `DataSet` is the primary mechanism used in .NET applications to cache data and pass it between tiers in a distributed application thereby alleviating the need to rely on proprietary schemes or COM marshalling.

Using XML alleviates several of the burdens of classic ADO. For example, by storing the data as XML it can easily pass through firewalls without special configuration. In addition, by storing related tables and representing the relationships between those tables the `DataSet` can store data hierarchically allowing for the easy manipulation of parent/child relationships. Furthermore, this disconnected model combined with connection pooling schemes frees resources on the database server more quickly, allowing applications to scale by not holding on to expensive database connections and locks.
ADO.NET Architecture

The System.Data namespace consists of two primary parts. The first part is the managed providers that contains ways to connect to a data source, issue commands against the data source, and read data directly from the data store or into a DataSet. The managed providers contain classes analogous to the Connection, Command, and Parameter objects in classic ADO as well as adding support for iterating through a result set in a forward-only manner. The second part is the DataSet and its various supporting classes which contain methods to manipulate data in a disconnected fashion. The DataSet is most like a disconnected Recordset, although much more powerful and flexible.

Figure 6.6 ADO.NET Architecture
Chapter 7: Security Issues in HealthCare Industry

7.1 HIPAA (Health Insurance Portability and Accountability Act)

The Health Insurance Portability and Accountability Act of 1996 (HIPAA) are regulations that govern privacy, security and electronic transaction standards for health care information. These regulations will require major changes in how healthcare organizations handle all facets of information management, including reimbursement, coding, security and patient records.

7.2 Security Implementation

This project gives healthcare organizations a Web-enabled information delivery solution for medical and business documents that can assist in meeting HIPAA requirements for privacy, security and administrative simplification.

This project deals with the following guidelines as laid by HIPAA:

Access Control

- Access is limited to authorized users
- Data access is governed by user-based, role-based and context-based rules

Data Authentication

- Healthcare information output is securely stored and protected against corruption in the database
Chapter 7: Security Issues in HealthCare Industry

Entity Authentication

- Automatic log off for idle end-users
- Each user is assigned a unique user identity and authenticated through a password system

Integrity Controls

- Our product uses secure socket layer communications to protect against unauthorized access to transmitted data
- Optional digital certificates authenticate clients and verify the integrity of transmitted healthcare information

Message Authentication

- Encryption processes protect data from intruders trying to access data through external communication points

7.2.1 Authenticating Users

Forms-based authentication is the process of detecting and redirecting unauthenticated users. Usually this means redirecting the user to a login page for identification. The login page should gather user credentials and compare them against a store of registered users. Once the identification has been made, the user's session can be assigned an authentication cookie that allows access to the rest of the site. In the project the users credentials i.e. login and password were matched with those with that stored in the database in order to authenticate them.
In .NET most of this is handled through the FormsAuthentication object. When forms-based authentication is enabled, any attempt to access a page without a valid cookie results in instant and automatic redirection to a specified login page. Enabling forms-based authentication involves modifying the ASP.NET configuration file called Web.config, which is an XML file provided as part of the default Web application. It contains XML tags that define various settings including security settings. Any page protected by forms-based authentication rejects sessions without a valid cookie automatically. The site redirects rejected sessions to the login page specified in the Web.config security settings. This login page can then handle the logic to authenticate such users against the information of the users stored in the database.

In the Web.config: the authentication mode is specified as:

```xml
<authentication mode="Forms">
  <forms name="HealthTools" loginUrl="UnSecure/Login.aspx"></forms>
</authentication>
<authorization>
  <deny users="?"></deny>
</authorization>
```

The Login.aspx: takes care of the logic of authenticating the user against the values stored in the database. The password is also encrypted before it is stored in the database thus adding one more layer of security during login. The hash algorithm used for this is SHA1.
ds=srl.login(txtLogin.Text,txtPwd.Text);
bool patient=false;
dt = ds.Tables[0];
foreach(DataRow dataRow in dt.Rows )
{
    pass=(string)dataRow["Password"];  
    try//if (patient)
    {
        PatientID=(string)dataRow["PatientID"];  
        patient=true;
    }
    catch//else
    {
        PhysicianID=(string)dataRow["PhysicianID"];  
    }
    count++;
}
string hashpass=
FormsAuthentication.HashPasswordForStoringInConfigFile(txtPwd.Text,"sha1");
if(pass==hashpass)
{
    if (patient)
    {
        sess[0]=PatientID;
        sess[1]=null;
        Session.Add("User",sess);
    }
    else
    {
        sess[0]=PhysicianID;
        sess[1]=null;
        Session.Add("User",sess);
    }
    FormsAuthentication.RedirectFromLoginPage(txtLogin.Text,false);
}
else
lblMsg.Text ="Wrong Password";

*During Login a session object is created for the user as:*

Session.Add("User",sess);
Chapter 7: Security Issues in Healthcare Industry

Subsequently in all the pages which require a login this Session object is checked in the Page_Load() to see if it exists. If it does the user is given access to the requested page or else he is redirected to the Login page.

```csharp
string []id=(string[])Session["User"];  
if(id[0]==null)
{
    FormsAuthentication.SignOut();
    Response.Clear();
    Response.Redirect("UnSecure/Login.aspx");
}
```

7.2.2 Catching Time-Outs

This project also deals with the issue of a user's session timing out, a problem because it tends to leave the user dangling half in and half out of your Web site. The solution is to detect timed-out users, reauthenticate them, and escort them back to the page they were visiting. Also by default if the user doesn’t interact with the site for at least 30 minutes the users session is automatically timed out.

When a session times out, the Session object's contents are lost. A timed-out session can be detected when a page attempts to retrieve the current Member from the Session object. .NET has methods which are used to declare and retrieve a typed object from the Session object in one line, as the following code shows.

```csharp
string []id=(string[])Session["User"];  
if(id[0]==null)
{
    FormsAuthentication.SignOut();
    Response.Clear();
    Response.Redirect("UnSecure/Login.aspx");
}
```
The SignOut() procedure strips the user of his or her authentication cookie, the site reloads the current page; it doesn’t redirect the user back to the login.aspx page. It does this because the signed-out member will fail the default authentication process and get redirected to login.aspx automatically. After the user has logged back in, the RedirectFromLoginPage method reloads the page where the authentication failure occurred originally. Checking for a valid Member object during Page_Load provides each page with an additional layer of security. If the page can’t retrieve a Member object, for whatever reason, it simply signs out and redirects the user to the login page. This process provides a simple catch-all that helps ensure each page operates only if the session has been authenticated successfully and has not timed out.

**7.2.3 Keep Data in the Dark (SSL)**

SSL is the technology used to encrypt and decrypt messages sent between the browser and server. By encrypting the data, messages can be prevented from being read while they are transferred across the Internet. SSL encrypts a message from the browser, then sends it to the server. When the message is received by the server, SSL decrypts it and verifies that it came from the correct sender (a process known as authentication).

SSL consists of software installed on both the browser and server. Recent versions of any of the major browsers, have support for SSL built into the browser. But still SSL has to be activated on the browser and installed on the Web server. SSL uses two kinds of certificates: root certificates and server certificates. Root certificates are installed on the browser, and server certificates exist on the Web server. A root certificate tells the browser that you will accept certificates signed by the owner of the root certificate. For
example, if you install a root certificate signed and issued by Blue Sand Software into your browser, you will be able to authenticate and decrypt messages that were sent from Blue Sand Software. This is vital to ensuring a secure transaction. Once installed, the server certificate allows you to create an encrypted session by loading a target page using the https protocol identifier instead of the usual http identifier. https lets you use the SSL protocol to handle the encryption seamlessly.

The encryption/decryption process goes something like this:

1. The user browses to the secure Web server's site.

2. The user's SSL secured session is started and a unique public key is created for the browser (using the certificate authority's root certificate).

3. A message is encrypted and then sent from the browser using the server's public key. The message is scrambled during the transmission so that nobody who intercepts the message can make sense of it.

4. The message is received by the Web server and is decrypted using the server's private key.

The process of SSL encryption relies upon two keys: the server's public key and private key. The private key only exists on the Web server itself and is used by the Web server to encrypt and decrypt secure messages. The public key exists on any client computer that has installed a root certificate for that Web server. Once the public key is installed, the user can send encrypted messages to and decrypt messages received from the Web server.
Figure 7.1 shows this process. Just to be extra safe, the keys are discarded once the transaction's session ends.

SSL doesn't prevent the message from being intercepted. However, it does make the message useless to the rogue interceptor. In other words, someone could capture the message on its way to the secure Web server, but could not decrypt it because they do not have the server's private key.
8.1 A new medical revolution

As the health care industry’s transition from paper to electronic medical records continues, another technological revolution is taking place in hospitals, clinics, and practitioner’s offices. Health care delivery itself is being increasingly mobilized through the use of wireless technologies. Innovation in wireless technology has exploded for all kinds of users in recent years. Consider, for instance the progression from those bulky, gray cellular phones to their Web-surfing progeny that are now just slightly larger than a matchbox and have infrared capabilities. Widespread reliance upon computer networks (present in almost every kind and size of organization), coupled with the colossal growth of the Internet demonstrates the benefits of shared data and resources. Wireless technology simply adds another critical layer to sharing information: mobility.

8.2 Wireless computing basics

Utilizing wireless technologies on any significant scale requires installation of a wireless local area network (WLAN). WLANs are flexible data communication systems that link together various wireless end users within a physician’s office, hospital building or even medical campus. Using electromagnetic waves (radio and infrared), WLANs transmit data through the air at a defined (and safe) frequency, and thereby minimize the need for wired connections. Thus, WLANs combine data connectivity with user mobility. Albeit on a much simpler scale and different frequency, WLANs are akin to cordless phones, with a base station, antenna, and hand-held wireless receiver. WLANs use
transceivers (a transmitter/receiver) that are linked to ‘access points’ and integrated with a facility’s wired LAN. Access points act as antennae, or repeaters, and extend the distance between end users and the transceiver. End units—the actual pieces of equipment—could be mobile workstations, handheld data assistants or medication barcode readers, etc., and access the WLAN via adapters, which are essentially an antenna installed into the end user device. WLANs and their end units must conform to Institute of Electrical and Electronics Engineers (IEEE) 802.11 standards for safe wireless transmission in medical facilities.

**Wireless healthcare applications:**

- **Administration and resource management**
  - Make quick work of healthcare paperwork.
  - connect hospitals, clinics and doctor's offices directly to insurance companies so that claims can be submitted as the care is actually delivered.
  - keep medical supplies moving efficiently from the distributor, to the warehouse and to the health care facility.
  - logging of equipment, and management of inventory on wireless units can make the
  - work of resource management much more efficient.
Chapter 8: Applications for Mobile Devices

- **Mobile workstations**
  - allow health care personnel to collect a wide array of patient information.
  - carry out real-time charting in the patient’s electronic medical record, and place orders for equipment or other therapies.
  - permit a growing number of functions to take place remotely.
  - found in surgery suites and during emergency procedures keep operating room and emergency room staff in constant communication with each other and with lifesaving patient data.

- **Hand-held data assistants**
  - offer the practitioner a multitude of instant references,
  - patient tracking software and a real-time access to the pharmacy.
  - be able to tie in distant satellite clinics,
  - access the hospital’s patient record system, and connect to insurance companies or hospital database systems.
  - more efficient access to information that will lead to quicker and possibly more accurate medical decision-making.

- **Patient monitoring**
  - A wireless network enables hospitals to make any bed a ‘monitored bed.’
  - allows continuous and non-invasive tracking of patient vital signs without disturbing the patient.
  - wireless mini-laptops that give health care practitioners real-time access to patient data.
look retrospectively to examine trends.

Some devices—which resemble pagers—ring if a predetermined vital sign or waveform (ECG) reaches an unacceptable level or form. A tone or vibration will alert the caregiver, displaying the most recent ECG waveform, heart rate, patient’s name, room number and alarm message.

Home patient monitoring

Wireless technologies can also be applied once the patient has left the reaches of the hospital’s WLAN.

Several wireless units now offer telemetry monitors which track a number of vital signs, such as ECG, heart rate, pulse, noninvasive blood pressure and send it back to a central nursing workstation.

At home, patients can be trained to use these devices to remotely monitor and transmit their medical data to a host computer.

Diabetes care is a popular application where patients can regularly monitor their blood glucose levels and send this data to a central station.
Chapter 8: Applications for Mobile Devices

8.3 Mobile Web Application Architecture

Generally because of the following reasons developing applications for wireless devices, such as cell phones, pagers, and personal digital assistants (PDAs) is challenging:

- Different markup languages are necessary, including HTML for PDAs, wireless markup language (WML) for wireless application protocol (WAP) cell phones, and compact HTML (cHTML) for Japanese i-mode phones.
- Devices have different form factors. For example, devices have varying numbers of display lines, horizontal or vertical screen orientation, and color or black and white displays.
- Devices have different network connectivity, ranging from 9.6 KB cellular connections to 11 MB Wireless LANs.
- Devices have different capabilities. Some devices can display images, some can make phone calls, and some can receive notification messages.

The Microsoft Mobile Internet Toolkit addresses these challenges by isolating them from the details of wireless development. Thus, developers can quickly and easily build a single, mobile Web application that delivers appropriate markup for a wide variety of mobile devices. Currently the Toolkit supports over 80 wireless devices.

The Microsoft Mobile Internet Toolkit extends the functionality of ASP.Net to easily target mobile devices using mobile Web Forms technology. Developers building mobile Web applications can use .Net framework services like XML Web services, ADO.Net for data access and the Common Language Runtime.
Chapter 8: Applications for Mobile Devices

Figure 8.1 Mobile Internet Toolkit

The following 2 examples illustrate the process in which 2 different devices (browsers) want to access our mobile Web application

1. Pocket PC
2. WML browser

Figure 8.2 Mobile Devices Accessing Mobile Application
A Pocket PC that wants to access a mobile Web application will make an HTTP request to the Web server. The HTTP request will be processed on the server in three main stages. The first process is identifying the requesting device in this instance a Pocket PC and the capabilities of that device, for example: browser, mark-up language, and image capabilities. The Microsoft Mobile Internet Toolkit extends the .NET Framework 'machine.config' schema with mobile device capabilities and pre-populates the device data. The 'machine.config' file applies to all applications on the server and the 'web.config' file applies to specific application or v-root. The HTTP request from the Pocket PC contains the User Agent string, Header information and URL that is being requested. The User Agent string is matched against entries in the 'machine.config' file. The URL from the HTTP Request is then used to locate the corresponding mobile Web page which will have a .aspx file extension.

![Diagram of Device Identification]

*Figure 8.3 Identification of Device Capabilities*
The first time that an ASPX page is accessed the page will be compiled. The ASPX page will be sent to the parser once the page has been parsed it will be processed by the compiler. The compiled page is then stored in the Assembly Cache. The server then creates a new instance of the compiled page, and uses it to process the request. Once the page has been compiled, the parsing and compiling steps do not need to be repeated for each request- the compiled page class can be reused, resulting in improved performance.

Figure 8.4 Compilation of Mobile.aspx page

After the ASPX page has been compiled, the page and the Mobile Controls used on it are instantiated. The business logic contained on the ASPX page is then executed. The business logic that is executed could be data retrieval, XML Web Services, or server side objects. This same business logic used in the mobile Web application may also be used by desktop Web applications allowing developers to easily re-use existing code when building mobile Web applications. The device adapters associated with the requesting device and controls used on the page then generate the appropriate mark-up language, in
this case HTML for the Pocket PC. The HTML is then encapsulated in an HTTP response and returned to the requesting device, in this case the Pocket PC.

![Diagram showing the process of generating HTML for mobile devices]

**Figure 8.5 Appropriate Generation of mark-up language**

When a WML browser accesses the same mobile Web application as the Pocket PC it goes through the following steps. The WAP browser makes a WAP request to a WAP Gateway. Usually, these gateways are a service provided by wireless carriers. The WAP Gateway translates the WAP request to an HTTP request and passes it to the Web.

![Diagram showing the WAP browser accessing a mobile application]

**Figure 8.6 WAP Browser Accessing Mobile Application**
The HTTP request that originated from the Microsoft Mobile Explorer contains the User Agent string, Header Information and URL that is being requested. The User Agent string is matched against entries in the ‘machine.config’ file.

The URL from the HTTP Request is then used to locate the corresponding mobile Web page which will have a .aspx file extension. Since this URL was previously accessed by the Pocket PC, the compiled version of the page already exists- the server can just create a new instance of it directly. It is important to note that the mobile .aspx file is only compiled once per “URL” and is not re-compiled for every device that accesses the URL. The page is now executed in the same manner as it was for the Pocket PC request we saw earlier. The page and its mobile controls are instantiated and the business logic is executed. However since the requesting device in this example is the Microsoft Mobile Explorer, and not the Pocket PC, WML device adapters are selected. This results in WML being generated for the Microsoft Mobile Explorer.

The WML is returned to the gateway in an HTTP response. The gateway then processes the response, compiles the WML into code the phone can understand, and send a WAP response back to the requesting WML browser.

Thus, the Microsoft Mobile Internet Toolkit extends the .NET Framework to target multiple mobile devices from a single mobile Web application.
Wish List of features for Mobile Applications:

- Pull patient chart
- Order/access prescription
- Order medical test
- Order/access lab test
- Note taking
- Clinical calculation
- Drug formulary & dosage
- Drug-drug interaction
- Billing capture

Features Implemented for this project

- Pull patient chart
- Order/access prescription
- Order medical test
- Order/access lab test
- Note taking
9.1 Data Mining

Data mining, *the extraction of hidden predictive information from large databases*, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

Most companies already collect and refine massive quantities of data. Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources, and can be integrated with new products and systems as they are brought on-line. When implemented on high performance client/server or parallel processing computers, data mining tools can analyze massive databases to deliver answers to questions such as, "Which clients are most likely to respond to my next promotional mailing, and why?"
Scope in Project:

Outcome research and database analysis: Outcomes research and database analysis play a key role in refining the programs and measuring the costs as well as identifying the effectiveness of various treatments for specific populations. Outcome requirements include clinical results, associated costs, quality of life and patient-provider satisfaction. To achieve this we plan to use Data Mining techniques to analyze the existing data in the centralized Data Base.

9.2 Integration with Existing Legacy Systems

Currently the project uses a centralized dummy database. In the real world scenario hospitals have many existing disparate legacy systems. The vision is to use XML to integrate these existing legacy systems. The final Application Architecture should be able to address the following issues:

- XML-based middleware layer
- Full compliance industry standard XML schemas
- Real-time connection to legacy systems via custom-made XML filters
- Integrates into existing internal systems and other external systems

XML as a Common Representation

XML has a number of things going for it as a common format for data. First, it is entirely represented in text. Unlike binary formats, XML can be processed without change regardless of the receiving platform. Different countries and languages employ different character sets, but XML includes provisions for specifying which set was used to
compose the document. Text, moreover, is the primary format used with such protocols as HTTP and SMTP, so XML needs no special handling when used with the most widely employed protocols of the Internet. XML is also experiencing a wave of popularity at the moment. Technologies built around XML provide powerful capabilities useful to web based applications.

9.3 Further Applications for Mobile Devices

According to a recent survey from Harris Interactive Inc., 26% of the country's practicing physicians used handhelds for professional and personal activities in 2001, up from 15% in 1999. Harris Interactive estimates that 50% of the country's physicians will be using the devices by 2005. But that could change markedly if insurers, employers, hospitals and other providers mandate physician usage. Employer groups already are putting pressure on doctors and hospitals to use computers to improve clinical care. These groups have said their health care purchasing decisions will be influenced by whether hospitals implement computerized physician order-entry systems to reduce medication errors.

Because Web Services and the Mobile Internet Toolkit give us the freedom and flexibility to use the existing application logic to be rendered on different client devices, it is relatively easy to extend the current applications on to handheld devices. Some of the applications which could be developed are:
• Monitors Patient Medication with Handhelds

Nurses use the handheld device to access patient profiles, and review medication that needs to be administered.

• Medication management

Handheld scanning devices can be used to scan the bar-coded medication and the patient’s bar-coded ID bracelet to confirm that the right medication is given to the patient at the right time and right dosage.

• Home patient monitoring

Handheld can be used to track a number of vital signs, such as ECG, heart rate, pulse, noninvasive blood pressure and send it back to a central nursing workstation.
Chapter 10 : References

http://www.hcfa.gov

Healthcare Informatics. Reuters Business Insight, p.12

Deregulation and Regulatory Backlash in Health Care CMR, Volume 43, Number 1,
http://www.hcfa.gov/stats/nhe-oact/tables/chart.htm

Datamonitor

“By The Numbers”, Modern Healthcare

Effect of computerized physician order entry & a team intervention on prevention of serious medication errors; JAMA

HIMSS (Healthcare Information and Management Systems Society)

MARS Project Brief, Dr. John K. Vries, UPMC

Journal of Healthcare Management

HIPAA (Health Insurance Portability and Accountability Act)
www.hcfa.gov/hipaa/hipaahm.htm

HL7 (Health Level Seven)
http://www.hl7.org/

www.microdoft.msdn.com

http://www.fawcette.com/vsm/

http://www.asp.net

http://www.w3.org/

Building XML Web Services with VB .NET By Yasser Shohoud
Appendix A: Screen Shots for Desktop Applications

This appendix shows some screen shots of the Desktop application.

A.1 For Patient's

Figure A.1 Patient's Main Page

Figure A.1 shows the main page of the patient after the patient logs on. The patient's session is tracked by storing the PatientID in a session object. This page shows the patient alerts i.e. the health parameters which have exceeded the critical values as set by his primary physician. From this page the patient can select various other options such as Scheduling an appointment, viewing his/her medical history, sending email, etc. The Health Tools link provided different health monitoring tools so that the patient can keep a track of them over a period of time. The values which the patient can track include Blood Pressure, Pulse, Weight, Calories, and Glucose Level.
Figure A.2 shows the page in which the patient’s can schedule future appointments with their primary physician. The patient can also view their upcoming appointments. The list of available time slots and dates can be modified by the physician so that only available slots show up.
Figure A.3 Patient’s Profile Page

Figure A.3 shows the page in which the patient can modify their profile. The fields marked in red can only be modified after contacting the hospital staff.
Figure A.4 Health Tools (Blood Pressure monitoring)

Figure A.4 shows one of the 6 Health Monitoring tools available to the patient. This page provides the UI for recording the various health parameter as also displays the recorded data in a graph.
Figure A.5 Physician’s Main Page

Figure B.2 shows the main page of the doctor after successfully logging in. This page give the doctor a list of alerts for the patients whose certain health parameters have exceeded the critical values. The doctor can select the patient from the dropdown list in order to exam the patient. It also shows the appointments for today.
Figure A.6 Summary of Patients Records

Figure A.6 represents the summary of the patients records. These records are available according to the patients visit date to the hospital. This screen provides the doctor with a quick summary of the patients diagnosis, symptoms, prescriptions and the different tests.
Appendix B: Screen Shots for Mobile Applications

This appendix shows some screen shots of the mobile application.

Figure B.1 Login Screen

Figure B.2 Doctor’s Main Page

Figure B.1 shows the login screen for authenticating the user. The user submits his/her credentials which are then verified against the values stored in the database. If successfully logged in a session is created for the user to track his/her movements through the site.

Figure B.2 shows the main page of the doctor after successfully logging in. This page give the doctor a list of alerts for the patients whose certain health parameters have exceeded the critical values. The doctor can select the patient from the dropdown list in order to exam the patient.
After selecting the patient the doctor is taken to a page shown in Figure B.3 corresponding to the patient. This page lists the different tools available to examine the patient. The identity of the patient being examined is maintained by storing the PatientID in a session object.

Figure B.4 shows the Examination Screen in which the doctor can give a regular examination of the patient and record his diagnosis and other additional comments.
Figure B.5 shows the prescription ordering page in which the doctor can order prescription for the patients. Figure B.6 shows the different Lab Tests and Medical Tests which can be ordered. Since the values for the above dropdown boxes are pulled from a database these values can be easily updated so that they contain the latest alternative available to the doctor. Also since these values are directly stored in the database they improve patient care and financial performance through reduction in adverse drug events (ADEs). Studies have shown that use of a computerized physician order entry system decreased the rate of non-intercepted serious medication errors by more than 50%.
Figure B.7 Various Graphs Available for a patient

Figure B.7 lists the various HealthLogs which a doctor can pull up for a patient. These Health Logs are used to monitor different parameters such as the patients blood pressure, glucose level, weight, etc. These Logs provide an easy visual reference which is helpful to the doctor to see the variation of the tracked parameters over a period of time. Figure B.8 shows the graph of the patient’s weight.