

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

# The future of health: Integrating medical health information systems and home health monitoring data into predictive Health systems

by: Faiz Rehman  
Student ID: 99049537

## Thesis Supervisors:

- |  |  |
|--|--|
| 1. Dr. Johan Potgieter<br>Senior Lecture in Mechatronics and Robotics<br>Massey University (Albany Campus)<br>E-mail: <a href="mailto:J.Potgieter@massey.ac.nz">J.Potgieter@massey.ac.nz</a> | 2. Dr Olaf Diegel<br>Associate Professor of Product Development<br>Massey University (Albany Campus)<br>E-mail: <a href="mailto:O.Diegel@massey.ac.nz">O.Diegel@massey.ac.nz</a> |
|--|--|

## Papers Published:

1. Electronics New Zealand Conference (ENZCon) - Predictive Health systems through the integration of home captured health data and medical health information systems. F. Rehman, O. Diegel  
[F. Rehman, O. D. (2006). "Predictive Health systems through the integration of home captured health data and medical health information systems." Electronics New Zealand Conference]

## **Abstract**

Health management systems have been around for a decade but none have offered the scalability such as the Internet. Personal patient health management has been the key issue behind deteriorating health. It has been realized that the patient can manage their condition at home while always staying in touch with a medical professional.

Research shows that hypertension, if regularly monitored at home, can be improved. Users who have access to modern technology can utilize this to manage their blood pressure. This thesis discusses the development of an application that inputs user blood pressure data at home and makes intelligent and predictive decisions for the patient. A statistical model is discussed, which can predict future trends in the user's blood pressure measurements and adjust acceptable threshold based on the patient's long term health.

By having the application running on a user's home PC, but with the data stored in a central location, telemedicine is made possible, as the patient's doctor or even the local health board has access to all patient data. If the system picks up any anomalies in the patient's data, a warning can be automatically issued to the patient or doctor so that intelligent decisions can be made to correct the situation.

The overall goal of this research was to develop a windows based / web based software application to capture patient data in a centralized environment for the health professionals to diagnose using up-to-date blood pressure readings.

The concept has been tested on a group of hypertensive users who had access to the software and their results were compared with a group who recorded their data using a conventional method. The collected data was analyzed and it was concluded that use of modern technology does help people manage their blood pressure among people with high blood pressure.

The success of this research will enable patients to manage their blood pressure, predict future readings based on a patient's history and give a medical professional the exact information needed in diagnosing a patient's condition.

## **Acknowledgment**

I would like to take this opportunity to express my deepest gratitude to academic supervisors, Dr. Olaf Diegel and Dr. Johan Potgieter for their continues support and guidance during the course of this thesis. Without their support this work would not have been possible.

I would also like to thank the test subjects, who have taken part in the experiment. Without their effort, their time, the design of the experiment would not have been possible.

I especially would like to thank my family without whose support, encouragement and being there when needed most; this work would not have been possible.

And finally my friends who have continually pestered me about the progress of the thesis, which eventually led to the completion of this work; so my sincere gratitude goes out to my friends.

# Table of Contents

Abstract .....	2
Acknowledgment.....	3
Acronyms - Definitions of terms used.....	7
List of Figures .....	10
List of Tables.....	12
List of Tables.....	12
Chapter 1 – Why?.....	13
Chapter 1 – Why?.....	13
1.0    Introduction.....	13
1.1    Background.....	14
1.3    Motivation.....	16
1.4    Hypothesis .....	18
1.4.1    Problem Description.....	18
Chapter 2 – Literature Review .....	19
2.1    Existing technology .....	19
2.2    Telemedicine.....	20
2.3    Wireless Interface .....	22
2.4    IT Devices in Hospitals.....	24
2.5    Use of mobile phones in the health care industry .....	27
2.6    Personalising patient information over the web.....	28
2.7    ‘Smart Phone’ applications in the real world.....	32
Chapter 3 – Data Centralisation – A concept .....	34
3.1    Why centralise?.....	34
3.2    The approach.....	35
3.3    ‘A’ solution.....	38
Chapter 4 - Methodology .....	45
4.1    The software methodology.....	45
4.2    Why consider different software development methodologies? .....	47
4.3    Different stages in a software development lifecycle .....	48
4.3    Methodology Used.....	50
4.4    RAD.....	50
4.5    Why RAD? .....	51
4.6    The software Architecture.....	53
4.7    The choice of software development platform.....	53
Chapter 5 - A new system .....	57
5.1    What is a new system?.....	57
5.2    So, why do we need a new system?.....	59
5.3    The new system .....	60

5.3.1	Stage 1: The data collection phase.....	60
5.3.2	The Personal Computer based application.....	62
5.3.3	Different aspects of the windows based application.....	64
5.3.4	The web Based Application.....	66
5.3.5	The components of the web based application .....	68
5.3.6	The central repository.....	70
Chapter 6 - Application Design.....		71
6.1	Windows Based Application.....	71
6.2	Reporting feature of the Windows application .....	76
6.3	The web based application.....	77
Chapter 7 - Statistical analysis of the results (Discussion).....		81
7.1	Background.....	81
7.2	Data Gathering Phase.....	82
7.3	Explanation and Scientific Rationale.....	84
7.4	Experiment Design / Analysis.....	85
7.5	Hypothesis Testing .....	90
7.6	Statistical model to predict future readings.....	94
Chapter 8 – Conclusion and Future work.....		96
8.1	Conclusion.....	96
8.2	“Rocky road” – Problems faced during this project.....	97
8.3	Future work / suggestions .....	97
Appendices.....		98
Appendix A – The Heart .....		99
Appendix B – Software Development Lifecycle / RAD .....		102
Appendix C - BPM – RS232 Interface API.....		103
Overview .....		104
Software Settings.....		104
Electrical Specifications .....		104
RS232C .....		104
Voltage Levels.....		104
Commands.....		105
Command Overview.....		105
Command Codes .....		106
Start Communications Task (20H).....		106
Stop Communications Task (21H).....		106
Read Cycle Data from ABPM (22H) .....		106
Write Cycle Data to ABPM (23H) .....		108
Read ID String and Version (24H).....		109
Read Parameters from ABPM (26H).....		109
Write Parameters to ABPM (27H) .....		111

Read Serial Number (3EH) .....	113
Data Specifications .....	114
Data Types .....	114
Data Formats .....	114
Bit Order .....	114
Byte Order: .....	114
Interval Periods .....	114
Language Code .....	115
QTrak Time Periods .....	115
Default Values .....	115
Rules .....	115
Retry Codes .....	116
Initiation Codes .....	117
Error Codes .....	117
Appendix D – Input validation algorithm .....	119
Appendix E – Patient Data Set .....	125
Appendix F – Microlife Blood Pressure Monitor Data Sheet .....	127
Appendix G – Blood pressure data sheet .....	129
References .....	130

## Acronyms - Definitions of terms used

Term	Definition
<b>Select Statement</b>	The SELECT statement is used to select data from a table. The tabular result is stored in a result table (called the result-set).
<b>API</b>	Application Programming Interface
<b>XML</b>	Extensible Mark-up Language
<b>XSLT</b>	Extensible stylesheet language transformation (XSLT) is a language for transforming XML documents into other XML documents. XSLT is designed for use as part of XSL, which is a style sheet language for XML. <a href="http://www.dmreview.com/resources/glossary.cfm">www.dmreview.com/resources/glossary.cfm</a>
<b>RDBMS</b>	Relational Database Management System. Used to store, process and manage data arranged in relational tables. Often used for transaction processing and data warehouses.
<b>RS232</b>	A standard interface between a computer input/output port and a peripheral device.
<b>ASP.NET</b>	NET is Microsoft's new programming technology which is now gaining acceptance and momentum. Soon, it will completely replace standard ASP. Glaserweb.com is already up to speed with .NET technology, and is fully ready for this switch. While .NET development is not as rapid (leading to higher production costs), it is significantly more stable, and runs much faster than older programming technologies, opening up new possibilities for web development.
<b>Microsoft SQL Server 2000/2005</b>	Microsoft SQL Server is a relational database management system produced by Microsoft. It supports a superset of Structured Query Language SQL, the most common database language. It is commonly used by businesses for small to medium sized databases, and in the past 5 years large enterprise databases and competes with other relational database products for this market segment.
<b>MySQL 5.0</b>	MySQL (pronounced "my ess cue el") is an open source relational database management system (RDBMS) that uses Structured Query Language (SQL), the most popular language for adding, accessing, and processing data in a

	database. Because it is open source, anyone can download MySQL and tailor it to their needs in accordance with the general public license. MySQL is noted mainly for its speed, reliability, and flexibility.
<b>PHP</b>	The PHP Hypertext Pre-processor is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications.
<b>Microsoft Visual Studio</b>	Microsoft Visual Studio is an integrated development environment by Microsoft.
<b>.NET</b>	A Microsoft operating system platform that incorporates applications, a suite of tools and services and a change in the infrastructure of the company's Web strategy.
<b>Win32</b>	Windows API is a set of APIs (application programming interfaces) available in the Microsoft Windows operating systems. A software development kit (SDK) is available for Windows, which provides documentation and tools to enable developers to create software using the Windows API and associated Windows technologies.
<b>Web Server</b>	A computer, including software package that provides a specific kind of service to client software running on other computers. More specifically, a server is a computer that manages and shares web based applications accessible anytime from any computer connected to the Internet.
<b>Microsoft IIS 5.0</b>	IIS (Microsoft Internet Information Services or Server) is a set of Internet based services for Windows machines. Originally supplied as part of the Option Pack for Windows NT, they were subsequently integrated with Windows 2000 and Windows Server 2003. The current (Windows 2003) version is IIS 6.0 and includes servers for FTP, SMTP, NNTP and HTTP/HTTPS. Earlier versions also included a Gopher server.
<b>Apache</b>	The most widely available HTTP server on the Internet. It supports the PERL and PHP languages.
<b>JBOSS</b>	JBoss (pronounced Jay Boss) is an Open Source J2EE based application server implemented in pure Java. Because it is Java based, JBoss can be used on any operating system that supports Java. The core developers are now hired by a services company named "JBoss Inc." founded by Marc Fleury,

	the writer of the first JBoss version. The project is backed by a worldwide network of partners. The company profits from a service-based business model.
<b>Microsoft BizTalk 2006</b>	BizTalk is a business process management (BPM) server that enables companies to automate and optimize business processes. This includes powerful, familiar tools to design, develop, deploy, and manage those processes.
<b>Hot swappable</b>	The ability to replace a component (e.g. disk drive, controller, fan, power source) while the system is on line, without having to power down; also referred to as hot-plug removable
<b>Run Time</b>	When a program is running, or executing, it is said to be in runtime. The term is mostly used by software developers to specify when errors in a program occur. A "runtime error" is an error that happens while the program is executing. For example if a program told you that 2 + 2 was 5000 that would be a runtime error. A memory leak where the program sucks up excessive amounts of system memory is also a runtime error.
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol A protocol for communication between computers, used as a standard for transmitting data over networks and as the basis for standard Internet protocols.
<b>HTTP</b>	Hyper Text Transfer Protocol (HTTP), the actual communications protocol that enables Web browsing.
<b>HTTPS</b>	A TCP/IP protocol that is used by World Wide Web servers and Web browsers to transfer and display hypermedia documents securely across the Internet.
<b>UART</b>	A device, usually an integrated circuit chip that performs the parallel-to-serial conversion of digital data to be transmitted and the serial-to-parallel conversion of digital data that has been transmitted.

## List of Figures

Figure 2.1: Describes a very basic telemedicine system [16].....	21
Figure 2.2: Mediator and Ontology based integration of ISCO patient database with the web [21].....	30
Figure 3.1 – A patient’s typical setup .....	35
Figure 3.3 – Single database server for the entire region .....	41
Figure 4.1 - Traditional Life Cycle a.k.a Waterfall model .....	45
Figure 4.2 - RAD optimised from the Traditional software model .....	50
Figure 4.3 – Software Market Share [29] .....	53
Figure 4.4 – Common language runtime (CLR) [31] .....	55
Figure 4.5 – The .NET Framework [30].....	56
Figure 5.1 – Current situation .....	57
Figure 5.2 – Proposed Idea .....	60
Figure 5.3 – Data collection phase.....	61
Figure 5.4 – Basic dataflow model of the windows based application.....	64
Figure 5.5 – Web based Application data flow model.....	66
Figure 5.6 – Web based application architecture.....	68
Figure 5.7 – Web based application open source architecture.....	69
Figure 5.8 – MS SQL Server Architecture [35].....	70
Figure 6.1 - User Interface of the windows based application prototype. ....	71
Figure 6.2 - Patient Profile.....	72
Figure 6.3 – Add new patient.....	73
Figure 6.4 – Update User/Patient Data .....	74
Figure 6.5 – User/Patient Statistics.....	75
Figure 6.6 – Patient report .....	76
Figure 6.7 – Logon screen .....	77
Figure 6.8 – Update User/patient data .....	78
Figure 6.9 – Data View page .....	79
Figure 7.1 – Distribution of the systolic data for Group A and Group B .....	88
Figure 7.2 – Distribution of Diastolic data for Group A and Group B.....	89
Figure 7.3 – Mathematical Hypothesis Statement [40] .....	90
Figure 7.4 – Scatter plot against Systolic vs. Diastolic.....	94

Figure 7.5 – Pearson’s correlation coefficient [39] .....	94
Figure 7.6 – Correlation calculation obtained from Minitab 15 .....	95
Figure A.1 Cross section of the heart.....	99
Figure A.2 Hypertension (High blood pressure).....	100
Figure B.1 – RAD mapped to Software Development Lifecycle .....	102

## List of Tables

Table 2.1: Wireless technology comparison for medical cable replacement [17].....	22
Table 7.1 – A typical user dataset.....	85
Table 7.2 – Descriptive Statistics for Group A.....	86
Table 7.3 – Descriptive Statistics for Group B.....	87
Table 7.4 – MANOVA Test of Between Subjects.....	91
Table 7.5 - Parameter Estimates .....	92

# Chapter 1 – Why?

## 1.0 Introduction

Technological advances in modern medicine have progressed to levels, from which mankind can benefit to vast extents. The so called itinerant age has changed drastically. The era where laptops used to be the highlight has long gone - this is the age of micro devices for instance smart phones, where a person can communicate with another via video calling or just by sending a simple text message.

The traditional way of providing healthcare, where patients queue at the doctor's practice and wait their turn to be diagnosed, will soon diminish, including the need for patients to visit a medical centre for a simple measurement of the patient's blood pressure.

Personal patient health management has always been the key issue behind deteriorating health. Once the patient leaves the doctor's surgery there is no ongoing record of their blood pressure data. A centralised software solution is needed to address this issue of personal blood pressure management. The user/patient is given options (software options) to choose from (windows based application, a web application or a mobile phone application) in order to generate their blood pressure related data.

The prototype applications built during the course of this thesis are to provide user/patients with options for recording their blood pressure data on an ongoing basis and to integrate this with existing medical technology.

## 1.1 Background

Personal health management remains one of the major factors; next to the medical condition itself, for an ever increasing numbers of patients. This is not only affects a person's health but destroys families and certainly costs the tax payers.

The numbers of patients diagnosed with hypertension (high blood pressure) are constantly on the rise; in the year 2000, 16% of the entire world's population was diagnosed with hypertension (obviously, this was based on a sample) [1].

Scientists have predicted that one third of the world's total population with high blood pressure will be in developing countries. [14]. Projections indicate that by 2040 the number of people with chronic conditions will increase by 50 percent (The Robert Wood Johnson Foundation, 1996). The increase in the number of people with hypertension and hypertension related conditions has had a direct cost of \$510 billion in 2000, and is projected to increase to \$1.07 trillion dollars by 2020 (2000, Anderson G).

In the USA alone, there are 65 million people who have high blood pressure, i.e. 1 out of every 3 adults [8], out of which 41.2 million people were aware of their condition and 70% of them were not able to manage their blood pressure. According to a world health report 2002, 75% of all strokes are due to high blood pressure. If we focus on the figures above, the statistics clearly indicate that blood pressure is not properly managed.

Internet is now widely accepted as the means of information flow, information management and communication flow between different organisations. Many organisations trade over the internet, some directly selling products or services, targeting household or commercial consumers and some indirectly by using the infrastructure to trade over the internet protocol; this is a considerably cheaper way to conduct business, for example the cost of placing an order by fax is far greater than emailing an order.

The underlying architecture of the internet can be used to transfer vital patient related information securely via the HTTPS protocol; hence no additional infrastructure cost is involved. The obvious choice to capture patient details in a central data warehouse would be to use the existing underlying infrastructure of the internet.

The focus of this thesis is to implement a system where people can manage their blood pressure in an efficient manner. Throughout the course of this thesis different approaches to better manage blood pressure will be discussed and basic prototypes will be developed and implemented to demonstrate their application in real life.

### **1.3 Motivation**

There are 6 million blood pressure monitors sold every year in the US alone, retailing around \$250 million. As our population grows the consumer market for blood pressure monitors is expected to grow by up to 20% every year. [12]

According to the figures stated above [12], Omron healthcare®, the world leader in manufacturing home blood pressure monitors, have sold 30 million units since 1974. [13]

An increase in the number of blood pressure monitors being sold every year indicates that awareness among people is growing. Yet the number of deaths due to hypertension is on the rise. The problem here is not with people not measuring their blood pressure regularly, but rather with the way their blood pressure is being managed. Currently there are blood pressure monitors readily available, that can be bought at any electronic store or pharmacy, which come with bundled blood pressure management software.

In the case of home blood pressure management, it is the patient's responsibility to enter the data into the software (assuming the person has installed the software to start off with). The software may display the data graphically or even generate reports to give them a better insight on their blood pressure. However, having said that, this scenario is based on the assumption that people actually use the software that accompanies the blood pressure monitoring unit.

It is an individual's tendency to take precautions when needed, just as it is human nature to eat, when hungry. In the same way it is human tendency for people to monitor their blood pressure only when they feel the need and if their blood pressure at that time falls under the normal range, they assume their high blood pressure is under control.

A possible solution is to have a system which can capture the patient's data in a centralised environment, where the user has different options to choose from by which they can update their blood pressure readings, for instance he/she could use their cell phones, or use a web based application, or a windows application or even their e-mail to populate their blood pressure data. This data can then be sent to their physician or accessed by the physician via a

web site, who can send any suggestions to improve or how to better manage their blood pressure.

## **1.4 Hypothesis**

If high blood pressure is inversely related to modern technology, then people with access to technology will be able to manage their blood pressure. If hypertension can be managed using modern technology, then people who have access to modern technology can manage their blood pressure more efficiently.

### **1.4.1 Problem Description**

This research project will be divided into two parts to achieve the common goal of smarter ways of home blood pressure management.

- A.** Developing a system, which is a combination of a web application, Microsoft windows based application to record a patient's blood pressure readings.
- B.** The hypothesis will be validated on the prototypes developed in 'A' using test subjects and the results presented in a case study.

## **Chapter 2 – Literature Review**

### **2.1 Existing technology**

There are currently many health management products on the market today with, most readily available over the counter - some expensive, some not so expensive.

Since, all the aspects of healthcare cannot be discussed, the blood pressure aspect of health management will be discussed here; this thesis will mainly focus on home blood pressure monitoring and its management.

Health management is an emerging field and yet very little research has been done in this area. However, the technological barriers have also adversely affected research in the health management arena.

The focus of this chapter will be on research conducted on the issue of improved access to health management.

## 2.2 Telemedicine

Telemedicine means: “transmission of information followed immediately by medical care”. It is the use of any electrical signal to transmit medical information. The concept of Telemedicine was developed to facilitate efficient medical care, improved services for patients and to provide medical attention to inaccessible or patients who are difficult to access (e.g. patients in rural areas, on ships etc.).

Telemedicine systems consist of two major units; the base unit – doctor’s unit, and a telemedicine unit – mobile unit. User friendly software is installed on both units so that they can both send and receive data from/to each other. The communication between both units is based on TCP/IP protocols.

The main objective of the designed system is to measure the potential at the remote site and store them in a central database. Equipment is required for measuring non-invasive and invasive blood pressure, ECG, Ultrasound, heart rate, temperature, oxygen saturation and still images. After a physician consultation the information is transferred to the telemedicine unit and data is stored in a database at the base unit.

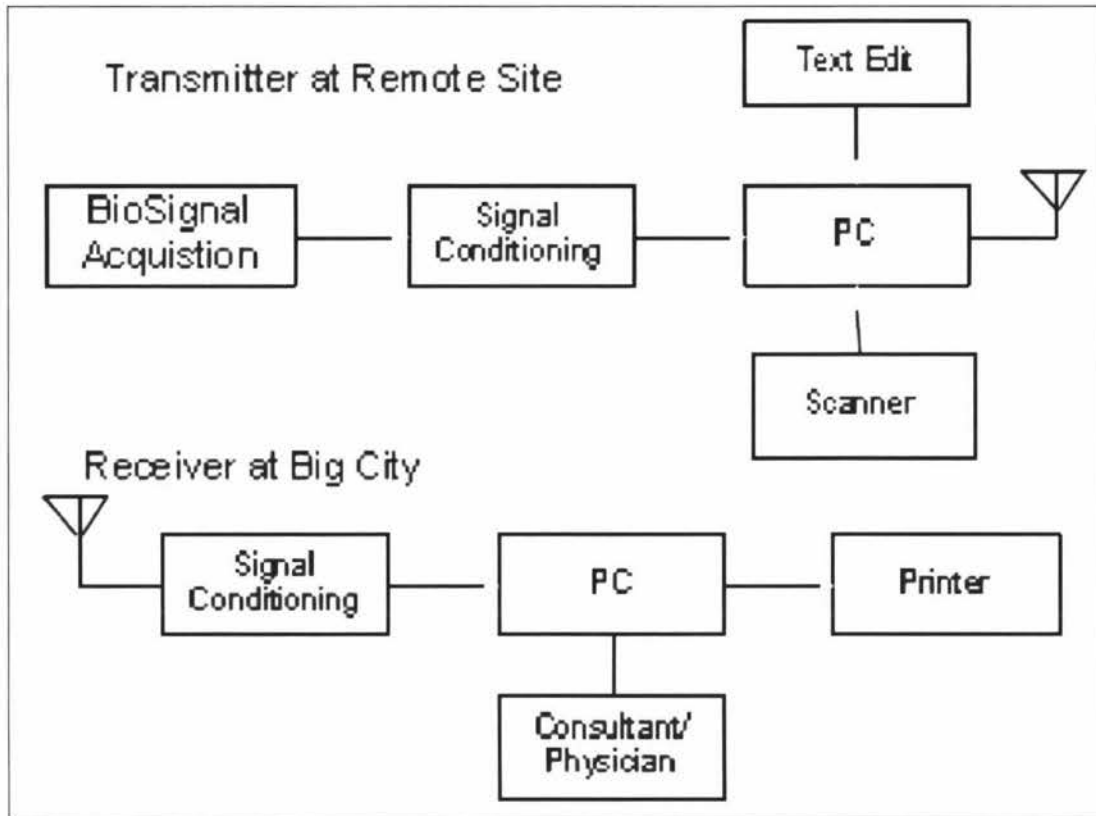
Telemedicine systems are supported by state of the art technology like interactive video, high resolution monitors, high speed computer networks and switching systems and telecommunications super highways including fibre optics, satellite, and cellular telephony.

The telemedicine unit mainly consists of;

- A biosignals acquisition module – responsible for biosignal acquisitions.
- A digital camera - responsible for image capturing
- A processing unit – a personal computer
- A communication module – ISDN, Satellite or POTS modem

A Base Unit (Doctor's unit) consists of;

- A dedicated PC
- A modem – responsible for data interchange.



**Figure 2.1: Describes a very basic telemedicine system [16]**

This system has been in place for a few years now, however is expensive to implement and in most cases is disregarded due to cost associated with the system. Infrastructure cost is definitely high and so is the ongoing cost associated with it.

## 2.3 Wireless Interface

Recent advances in wireless technologies now make it possible to free patients from their equipment, allowing greater freedom and even making health monitoring possible by their health provider while the patient is on the go.

There are different wireless technologies being offered today in the market. Table 1 below lists the pros and cons of the most popular wireless technologies. Bluetooth is one of the most preferred ways of networking among hardware. Going wireless has many other benefits apart from just reducing the number of wires.

- Reduces cabling cost - it is more cost effective to build onboard Bluetooth capability than to have a cable running between two points.
- Provides galvanic isolation - the cable carrying the data also exposes the patient to harmful currents; whereas a wireless solution provides this isolation and also reduces the cost of cabling.

**Table 2.1: Wireless technology comparison for medical cable replacement [17]**

Technology	Pros	Cons
<b>900 MHz</b>	Unlicensed band  Long range	Unnecessary range increases security concern  Different frequencies for U.S. (915 MHz) and EU (868 MHz)
<b>WiFi</b>	Large installed base of equipment Programmable for worldwide band usage	Long-packets are less robust  High current draw
<b>Zigbee</b>	Low-power for intermittent data Short packets are more robust	Protocol stack required on host Network coordinator required
<b>Bluetooth</b>	Large installed base of equipment Optimized for ad-hoc networking	Seven device limit per piconet Instantaneous power similar to Zigbee

A real life example where wireless connectivity could be used is blood pressure monitoring. According to Baisa, the non-invasive measurement of arterial blood pressure is performed by inflating a cuff around an arm or a finger. As pressure is released, the arterial pulsation returns, with the first pressure reading known as systolic.

The pressure in the cuff continues to decrease until pulsations are no longer detected, with the last reading known as diastolic. Typically, conversion of pulses from the physical to electrical domain is accomplished with either a pressure sensor or a piezoelectric strip.

Wireless technology can be used in this situation, especially when the data is time critical. Retrofitting existing equipment can be as simple as attaching a module to the RS232 port, if it exists, or by connecting a module to the UART lines of the embedded software in this way, it is possible to use Dial Up Networking to send results to a doctor.

One of the main benefits of this approach is that the data captured by the apparatus can be directly uploaded to a central location via the users / patient's PC or mobile phone.

This would give immediate visibility over the person's medical condition and healthcare professionals can give improved diagnoses using the data captured.

## 2.4 IT Devices in Hospitals

Technology has played a vital role in evolving modern medicine. Hospitals are one of the largest consumers of electronic products and constantly accommodating the ever-changing technology.

At Carillion Health System 10,000 blood pressure readings are taken by their officials at 100 different sites. Not so long ago about 5,000 readings were entered manually [18].

This was not only a waste of qualified human resources but also time, in a mission critical environment. There were other unnoticed drawbacks to the above practice, namely data integrity was not maintained due to human error while performing the iterative task of keying patient data.

According to Greg Waltson, a senior vice president and Chief Information Officer (CIO) at Carillion, if you use 0.5% error rate, that's 25 times a day, which is close to 10,000 false data in a year. This inaccuracy in the data could potentially lead to other unforeseen consequences, which adversely reflects upon the medical centres and the qualified professionals' credibility [18].

Many such cases of data being mishandled by health professionals have made it to the headlines and faith in the common man has been diminishing regarding the current state of medical care. Health boards have been constantly trying to improve?? this system, in order to decrease these inaccuracies and at the same time free qualified staff to do what they do best.

One such step has been taken by the Carillion Health System, where they have been testing a vital signs monitoring system where patient's blood pressure, blood oxygen and other data is electronically transmitted to the electronic records system. This system has reduced the data entry time of a health professional, giving them freedom to focus on the patient's condition rather than the keying in of health related data. Another benefit is that accuracy is maintained across all data. Obviously, valid data provides true insight on the quality of healthcare.

There is much medical equipment that comes with wireless technology built in. This would reduce the number of ward visits from a medical practitioner just to record the data and the

data can be sent directly from the device to the central repository. However, the equipment is not cheap to manufacture, and although it would be ideal for large expensive equipment to come with built-in technology such as Bluetooth or wireless connectivity, but not for the less expensive heart rate monitors.

There are many advantages in the above approach. For instance the data is present in a central environment and most of all the data is accurate as it is not prone to human error. It gives clinicians time to concentrate on greater numbers of patients, freeing them from the tedious task of iterative data entry. This also has a positive effect on the medical staff; indirectly they will have a sense of satisfaction because they will be doing what they are passionate about, healthcare rather than spending time entering data. When the time comes to retrieve patient data, health professionals can easily do so from the central repository.

Just as there are a lot of advantages to this approach; so there are also a number of disadvantages. The most obvious ones are that the cost of centralising this data acquisition - which is a complex and expensive process. The infrastructure is not cheap and there would be a lot of skilled personnel involved in the setup of this long and exhaustive procedure. Along with a lot of data servers, there would be vast amounts of data storage capacity required and these aren't cheap. Managing diagnostic images would especially be a disadvantage, as these would be large in size and would consume significantly large amounts of disk storage space. According to Agfa Corp, the demand for storage space grows by 40% and is compounded annually. Data storage is a single specific factor that is bringing health institutes to a standstill regarding implementing a centralised health management system.

Another factor which is not currently cost effective in the system is integration of a health institute with another supporting organisation – which at present does not exist. For example data that is captured by the hospital is not passed on or shared with local general practitioners, so he/she has to request that the parent organisation pass on the patient related data, so they can better perform a diagnosis. If this database were centralised, the data would have been readily available and related case notes from the doctors would have been recorded.

Another issue that is not covered by most is, what happens once the patient leaves the centralised environment; that is how is the data recorded or managed by the patient at home, which is his/her comfort area. This is the key issue that few researchers have shed light on.

## **2.5 Use of mobile phones in the health care industry**

According to Dr Jeremy Wheeler, mobile phone technology can improve chronic disease care [19]. He states that telemedicine is not an aspect of technology in healthcare that has excited enthusiasm among doctors or their patients. Using a phone-based system to record patient data has often proved cumbersome and time consuming and has offered little or no advantage over traditional diary based reporting.

The patient has to be dedicated to manually record blood sugar levels, especially when it comes to chronic diseases. They have to take time out to manually enter their readings for example blood sugar levels, peak flow or blood pressure, then log on to a computer and transmit the information to their local doctor or health professional. After a long surgery GPs are reluctant to view patient data, which is understandable.

Dr Wheeler has proposed a new system in which a mobile phone is used in conjunction with a measurement mechanism such as a glucose meter or peak flow meter, simplifying the collection of the patient data. A study where Dr Wheeler was involved with asthma patients found that patients actually enjoyed inputting data into their phone, which led to high rates of compliance. Additionally, the mobile phone based solution gave a greater insight by providing more complete information.

There are many pros and cons with the approach discussed above; firstly the mobile network coverage is not available in a few locations; what happens when a patient does not reside in a mobile network coverage area? Another common issue is that the cost associated with a mobile phone is not cheap. In New Zealand some mobile network providers charge a minimum of 20 cents to send a text message and if the patient has to take a minimum of 3 blood pressure readings and send 3 messages a day, that would cost approximately \$ 220 a year per health related problem. If a patient suffers from high blood pressure, is also a diabetic and has asthma as well, the cost would increase proportionately.

However, if the cost of the mobile usage network drops significantly or if government can subsidise the cost, the success of this concept would be much more likely.

## **2.6 Personalising patient information over the web**

A problem with current medical sources is that they provide the user/patient with enormous amounts of information. The information obtained is either, general, complex or limited with regards to the patient's current medical condition.

Traditionally, patients seek health information from health professionals - although they also pay a lot of attention to medical brochures and leaflets or electronic publications. The reason why patients pay more attention to information from a medical professional is because it is specific to their condition and is intended to provide the patient with an insight into their own medical condition. However, in cases where a few enthusiastic patients would like more in-depth information about their health, they tend to research the material on their own, and the information they might come across is either not relevant to their condition or the terminology used is too sophisticated for a layman. This information gap needs to be bridged and this can be overcome by using the internet as a source of information.

The internet is the largest library in the world, and is constantly growing. However, the information is uncontrolled and the credibility of the information can be questioned if not obtained from a reliable source. According to Al-Busaidi et al., an associate at Cardiff University has conducted a study on personalizing web information for patients - i.e. linking patient medical data with the web via a patient personal knowledge base.

The focus of this study is based on the significance of customising health information for patients, based on their data taken directly from their medical records, in order to assist and control their health. The patient data is taken from the patient database and linked with the web data; this allows the web information to be focussed on the patient's condition.

The issue of semantics (i.e. meaning of the terms, medical in this context) to handle differences and complexities of terms, operates through a programme which finds words similar or related to the words describing patients' medical data. This method is used to remove the complexity in medical terms, so it can be easily understood by patients.

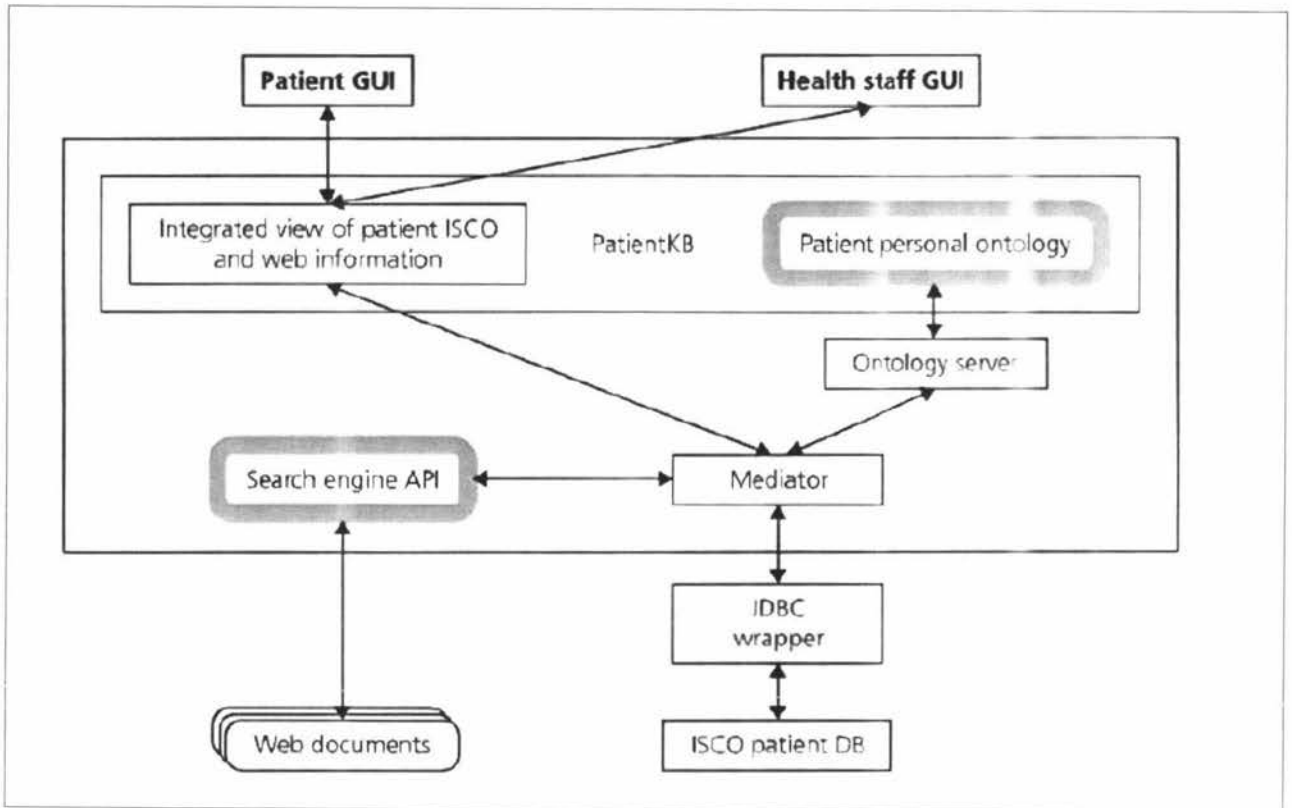
The information from the web is extracted, based on the Information System for Clinical Organisation (ISCO), ISCO and is a clinical information system first developed in 1991 by the Velindre NHA Trust (the south Wales Cancer Center) to meet the information requirements of the trust's oncology service and to collect clinical management information for analysis [20].

The ISCO also accommodates and processes data for patients with diseases other than cancer, for example, ISCO was adopted by Cervical Screening, Wales for use as a Colposcopy system and Movement Disorder e-Network to support the integrated care of movement disorder patients in South East Wales.

The study conducted by Al-Busaidi et al., uses an autonomous version of ISCO as a source for autonomized patient medical records.

The system proposed by the researcher is an integration of the ISCO (which is a relational database) and the Web (which a multitude of semi-structured HTML or XML documents).

The system uses an ontology (thesaurus-like) method, to identify conceptual relationships between terms (for example 'renal' versus 'kidney') in the patient database and web information sources.



**Figure 2.2: Mediator and Ontology based integration of ISCO patient database with the web [21]**

As shown in the diagram above, the architecture is quite self explanatory. The integrated view will be stored in a knowledge base that stores links to patient medical details in the ISCO database and hyperlinks them with relevant live web documents.

The ontology was employed to discover relationships (synonyms) between ISCO and the web and to expand the web search.

The following were the main set of features used in the above architecture

- A set of web documents and anonymous test data from the patient database
- Mediator: this handled and coordinated the interaction between the various components of the system; the patient database, the web, the search mechanism and the patient's personal ontology.
- Patient knowledge base stores the patients personal medical details and relevant web information

There are several advantages to the above mentioned process of personalising web information for patients, as mentioned below:

- Patients have ready access to the basic medical information about their medical condition, diagnosis and treatment, which the patient can refer to whenever they feel the need at their own convenience.
- Upon login each patient is presented with their own, personal information, it is not generic, but rather, is customised for each patient, which makes the patient feel comfortable
- The medical related treatment, diagnostics information and therapy information is sited by a medical professional and is stamped for quality, this gives the patients the confidence that the information they are viewing is credible.
- The information is readily available and the use of the simplified terminology applied makes it easier for the patient to understand.
- The system is web-based, hence can be accessed from anywhere, given there exists web accessibility.
- An indirect affect is the savings made in the health related costs, paid for by the tax payer.

The system would work efficiently if implemented, however the proposed system also has a few drawbacks that might prevent the system being feasible. For example,

- There are medical professionals involved in approval stamping of the information and this would cost resources
- Interpreting and summarising web searches of web pages is an expensive process, as it requires sophisticated computer hardware and complex algorithms, which are not easy to develop.
- As seen in the above Figure 2.2, the architecture would require complex hardware in order to build the proposed system

Having highlighted the advantages and disadvantages of the above system, the system would definitely be of great benefit, if it can be implemented on a global scale or even on a national scale. There would be security issues around the data retrieval and architecture will also have to be simplified in order to reduce the cost, once the implementation is executed on a large scale.

However, the information would help the patient in better managing their condition and evidently, an informed patient responds better to therapy.

## 2.7 'Smart Phone' applications in the real world

A recent article, "The doctor in your pocket", by one of the leading publishers, the Economist, discussed how a nutritionist had helped one of his clients reduce his life threatening cholesterol level to a reasonable level. However, once the businessman, Mr Gary Katz, got back to his normal habits the cholesterol level shot up again, so he turned back to the same nutritionist. This time the nutritionist sent the businessman home with a new toy - a mobile phone.

Each time Mr.Katz got ready to have his meal; he would simply take a photo of his meal and send it to nutritionist, which was much easier than the traditional way of keeping a food diary. With a new service called MyFoodPhone, this was much easier for the busy businessman.

This concept of using a phone to receive health related information is not new. Doctors and nurses have always made house calls and received calls from their patients, which was part of providing a remote medical care service. However, technology has advanced to levels where a mobile phone based camera could come in handy, as in the above situation. HBS consulting, a consultancy based in London, estimates that the global use of telecommunications and information technology to deliver healthcare and related services will grow from 3 billion in 2003 to 7.7 billion in 2006 [22].

According to America's Centres for Disease Control and Prevention close to 100 million Americans suffer from a chronic illness, which accounts for more than 10 trillion US dollars annual spending on healthcare.

A joint venture between Institute of Biomedical Engineering at the Imperial College, London, and Oracle – the world's second largest software developers – has resulted in the setting up of a monitoring system which is being tested this year. A small sensor is attached to a patient's chest which monitors his/her heartbeat and detects irregularities. The resulting ECG (Electro Cardio Gram) data is sent using Bluetooth to a mobile phone, which in turn is transmitted to the doctor [23].

A mobile phone would be a most versatile tool in this scenario as most people are accustomed to this technology and would prefer to use the same rather than carry another piece of hardware around with them. For instance Joseph Kyedar of Partners Telemedicine, also a professor at the Harvard School of Medicine, suggests that insurance companies might give out customised plans for patients who need to go for walks to improve their physical health and the mobile phone could, in turn, be used to transmit that data to the nearby doctor. This information is vital when compared to a patient in a hospital, because it is not monitored as it would be in a hospital environment. Addressing the issue of this uncontrolled data is the key to improving the healthcare system.

There are many obvious advantages of this approach as it gives the patient the freedom and flexibility to do things and have peace of mind knowing that his/her doctor is just a phone call away. As technology evolves, complex hardware will become easily available which then can be used to lay the necessary underlying infrastructure. Once this has been established, technology will be more accessible to the public and concepts such as, "Doctor in your pocket" can be applied. Having said that, technology is getting to a stage where it changes daily, mobile phones such as an iMate cost around \$600 US dollars and someone who is health conscious would consider this as a health safety option.

Cost plays a key role in an application of such principles. Government bodies should focus on making such technology available to everyone - and if this is achieved, then the above principle would be a success.

## **Chapter 3 – Data Centralisation – A concept**

### **3.1 Why centralise?**

During the course of this thesis, for the purpose of data collection, a Microlife (Model No BP 3AC1-1 PC) blood pressure measuring device was provided. It had the basic features of a regular blood pressure measuring device. It had a large LCD display which displayed a user's Systolic, Diastolic and Pulse readings and was also capable of storing 99 readings on its onboard memory (please refer to Appendix F for the product fact sheet). The device also featured a RS232 interface so it could be easily connected to a PC via a USB port. The data could be downloaded from the device onto the PC in a XML file format.

Once the device reached its storage capacity, it had to be reset to store future readings.

The problem here is with the data not being stored resulting in the doctor not having access to the actual patient data on which he/she could diagnose. Hence, if a central storage repository is designed to collect such information, then this gives the doctor access to the current medical related patient data.

For experimental purposes the data was collected from only a few users, hence a database farm was not required. If the prototype discussed here had been implemented in real life, then problems such as storage issues, security around the data and computer infrastructure would have been encountered.

In this section a brief method for data centralisation will be proposed and the various methods that could be adopted to overcome the issues with computer infrastructure, database farms and other issues arising.

### 3.2 The approach

Initially the data would be captured from the blood pressure monitor, copied across to the patient's personal computer via a Universal Serial Bus (USB) interface. The patient's personal computer will host the patient's or household database. In cases where the patient does not have a computer, they can take their apparatus to their local General Practitioner and their data can be uploaded to a central database server.

Where the patient has their own personal computer, the data is loaded onto the local database, which is a single time installation made by the user. Once this has been done, new records are automatically uploaded by the personal computer to a central server (provided it has an internet connection) (as shown in Figure 3.1)

There are 2 stages involved here. Before the data is published to a central server, it is retrieved from the apparatus (in our case it is a blood pressure monitor), the data is validated and the local database is updated. Once this has been done, only the new records are updated in the central database server.

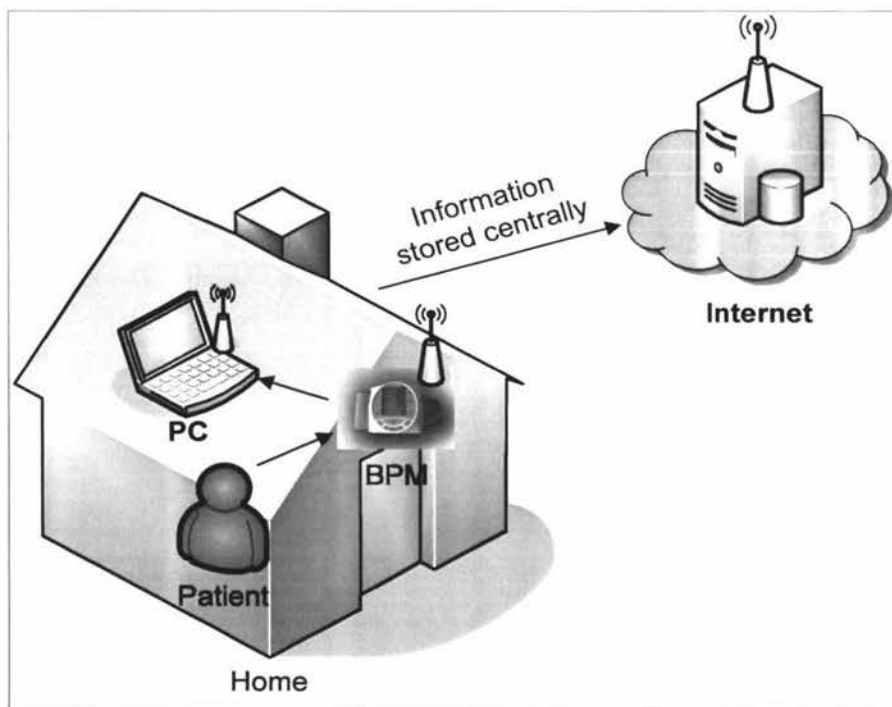


Figure 3.1 – A patient's typical setup

This should be a trouble-free process. Examining this using the statistics mentioned in Chapter 1, there about 65 million people in the USA alone, who suffer from hypertension and, as prescribed by their doctor, on average they would need to monitor their blood pressure four times a day. This is usually advised by a medical professional in extreme cases and cannot be generalised for all patients. Hence considering the above worst case scenario, if all 65 million Americans who suffer from hypertension have been asked to measure their blood pressure four times a day and have been asked to upload their data to a central server, this would mean there would be  $65 \text{ million} * 4 = 260 \text{ million}$  records a day! A SQL server table can hold up to 2 billion records, so get the average number of days a table can operate before reaching its maximum capacity is as follows:

$$\mathbf{2 \text{ billion} / 260 \text{ million} = 7.69 \text{ days}}$$

Hence, a table in the database will reach its maximum capacity in just a little over a week. Regarding the above statistic, the statistics are hardware resource dependent, and feasibility depends on available resources. If we had huge data storage and unlimited hardware resources in terms of RAM, and unlimited bandwidth then this would work - but this is not the case. Everything has a limit and this limit will be reached sooner or later, but we can prepare for scalability and redundancy.

This does not mean that we cannot develop this concept further. There can be multiple (n) tables holding the data, because obviously one cannot expect all the data to reside on a single device/database, so the table size will grow automatically as needed and the cost of the hardware will constantly be reducing as new hardware heads towards the market.

The solution for the above problem would be extensive hardware resources. However, these days technology has advanced to stages where the hardware price is constantly reducing. According to information retrieved from [www.qmb.co.nz](http://www.qmb.co.nz) (a leading computer hardware retailer in New Zealand) a 400GB hard disk costs around \$150 US dollars [24], which is priceless compared to the data it holds. Obviously the data will continuously grow as we have seen above; nevertheless, the hardware resources cost is inversely proportional to time, but that doesn't mean that computers will be free one day - it just means that you will get more computational power for the same price today compared to what you would have got 6 months earlier.

Once we have accepted the fact that there is infrastructure cost associated and we will definitely need more than one disk to capture the data, we will move to looking at how we can solve the problem of running out of hardware resources for data acquisition.

### 3.3 'A' solution

We will need N number of database servers, which will host the data for us. In other words, we will need to construct a database farm, (a database farm is a collection of N servers all running a Relational Database Management system, RDBMS), which can be hosted at the same site or different sites across a geographic area (as seen in the Figure 3.2 below).

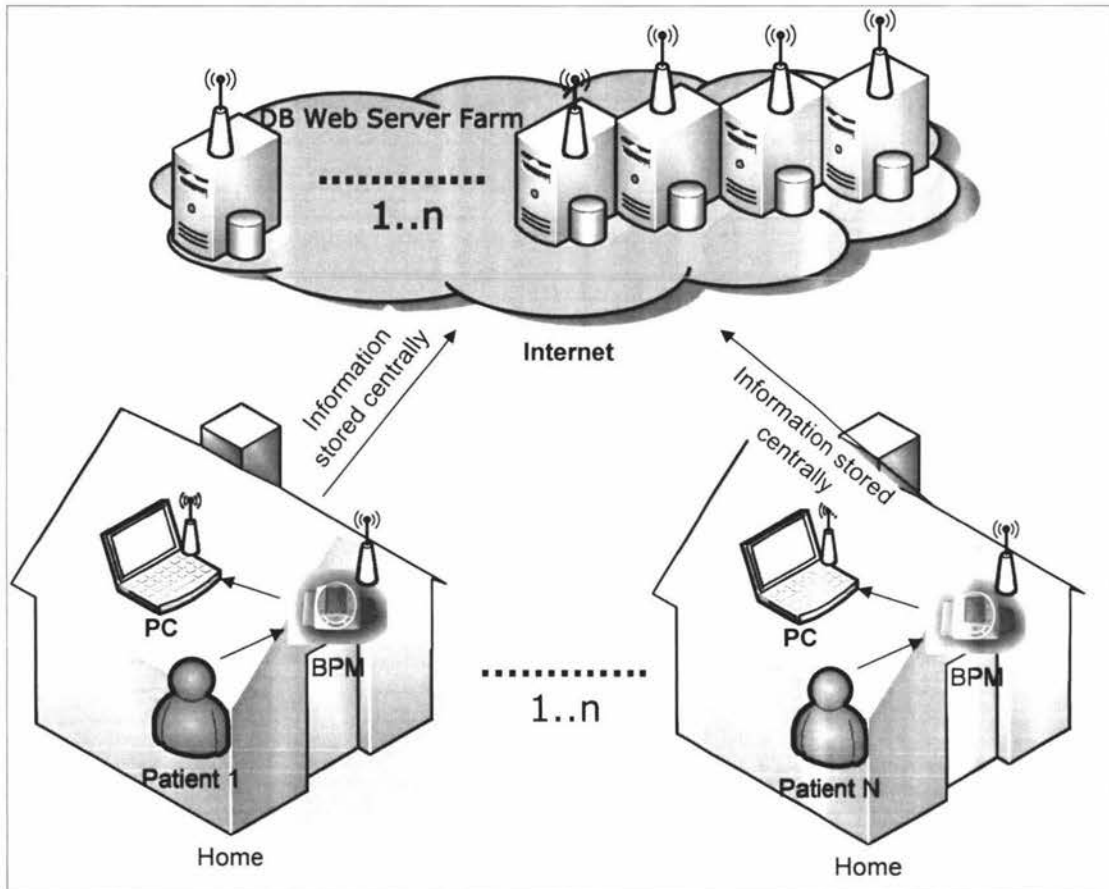


Figure 3.2 – N patients setup

Once the problem of storage is sorted, there are other problems which need to be considered. As we have seen above, the data will grow rapidly; hence a theoretical solution is needed for this. Another related issue would be the retrieval of data from the database.

In order to get a better picture consider an example: we have a Dell server, specifications as follows:

- Intel Xeon 2.6 GHz
- 4 GB RAM
- 400GB hard disk capacity
- Operating system Microsoft windows 2003 standard edition, Service pack 1
- RDBMS – Microsoft SQL Server 2005 Enterprise Edition

To get an idea of the performance of the server in the above example, lets compare it with a normal household computer, For instance a typical University lab computer uses HP Intel Pentium 4 2.6 GHz, with 1 GB RAM, 80 GB Hard disk, the operating system is Windows XP Professional (and the university leases them for about 3 years – at which time they are replaced with a new machine). As it is evident that the server is far better equipped with resources, hence one would expect the server to be faster when processing instructions.

Returning to our issue of examining the retrieval of the data, we have a table on the MS SQL server, which is not extremely large; it contains about 6.7 million rows, and about 35 columns. The table stores string values, no binary values, i.e. only text and no images.

Now, by running a simple select statement [25] as follows  
use SalesWarehouse – specifies which table to use (optional)

Select \* from Sales – specifies what to select

In the select statement you can specify column names or \* to select all.

The above select statement returns a result set, which is a table of 6.7 million rows, this retrieval of the data takes about 18 minutes to complete. According to our previous calculation, the USA alone would generate 260 million records a day and if we have to retrieve all the data using a simple SQL select statement we will be looking at:

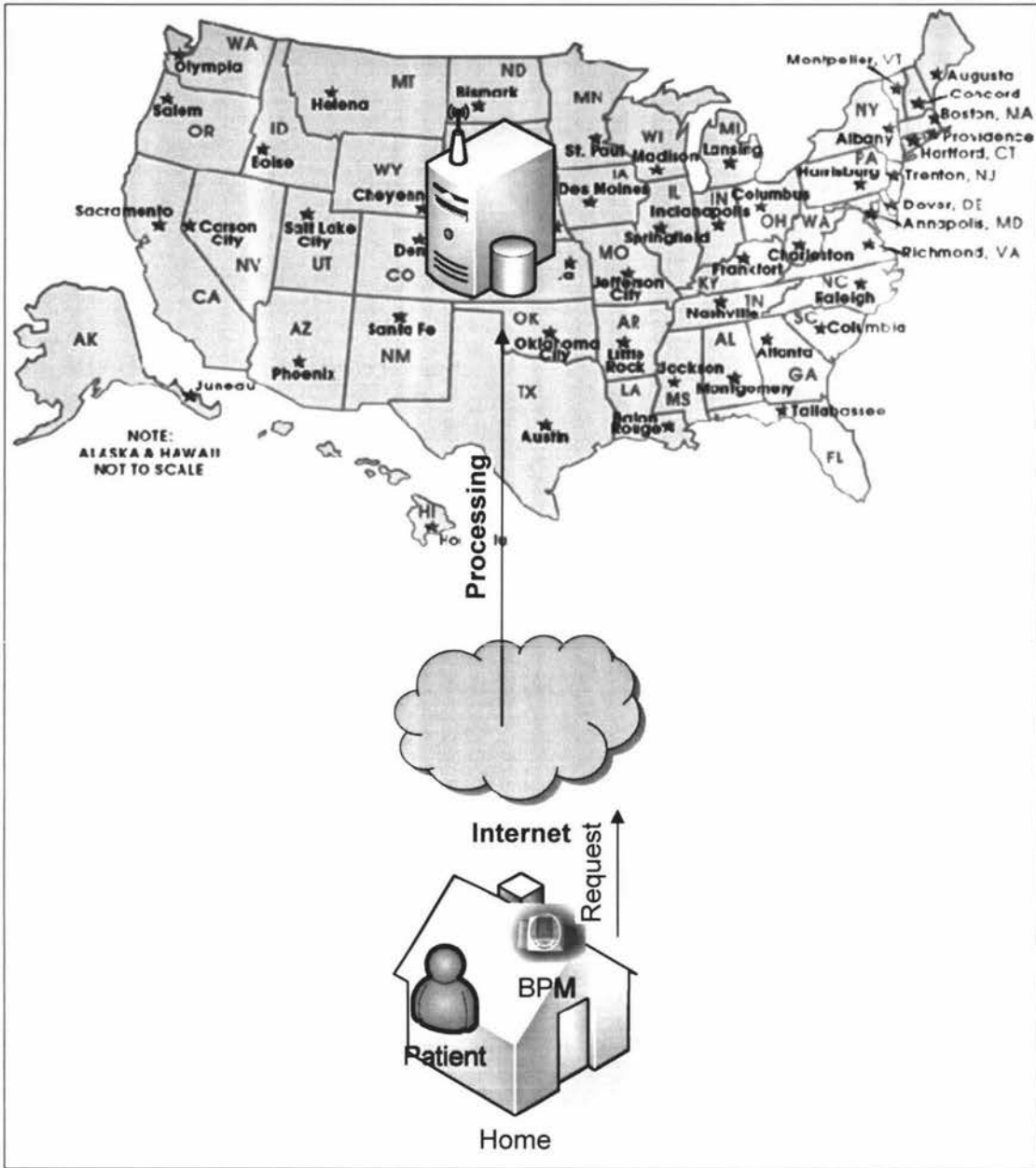
If it takes 18 Minutes to retrieve 6.7 Million records

How long would 260 Million take?

$(260/18) * 6.7 = 11.6$  hours, which doesn't seem very practical! So if the Health Commission would like a report on all the patient records, they would have to give at least 12 hours notice.

Are there any possible solutions for this problem? Happily, there are! We could parallelize the database, and the records coming in can be stored in a database based on the geographical location. For example the United States of America has 50 states, if a database is allocated to each state then that gives 50 individual databases, which increases the number of databases to manage, but also gives flexibility. A patient's data can be written to their region specific database; this will be achieved when the patient's profile is created (an option where the patient specifies their region/state).

An example of the above is as follows: if, there is a single central database server in Nebraska and a multitude of sites accessing data from this single data store, the performance of this single database server would be poor, especially considering the fact that we would be storing data separately for separate regions (states), hence, it would involve querying different databases simultaneously at any given time (see Figure 3.3 below).



**Figure 3.3 – Single database server for the entire region**

A solution involves allocating additional resources, such as introducing more database servers in the architecture, or even initially creating a distributed database server farm. However, consideration of economic factors leads to new discoveries and the funds could be used elsewhere.

Approaching this problem differently to the above - let's define a few scenarios first:

1. We could possibly place a copy of the database at different locations and the request traffic will access the nearest physical database.

a. **Advantages**

- i. Requests are diverted to the nearest database, hence faster response in terms of requests having to travel half way across the world
- ii. Does not clog a central database or internet traffic, as the traffic is constantly diverted to its nearest physically located database

b. **Disadvantages**

- i. A complete copy of the database has to be maintained at each different site
- ii. Synchronisation issues - data could be accessed at a time the synchronisation is taking place, so data access locks will be needed, programming will be required, hence maintenance required.
- iii. Each database at each different location will grow rapidly in size, hence the storage issue will arise
- iv. Retrieval of records will involve searching the entire database; due to the size of the database (as explained above) efficiency will be compromised, unless an exceptional search engine is used to retrieve the patient record set

2. We could create new data stores (as seen in Figure 3.4 below) for different states and the traffic could be diverted, depending on origin of the request.

a. **Advantages**

- i. Each data store would have its own copy of the database and store site specific data
- ii. Decrease in the amount of network traffic, as the requests will be re-routed to the area specific data server
- iii. Increase in transaction processing speed, as traffic is dependent on the origin of request

**b. Disadvantages**

- i. We will need to follow a client server topology - where the server will hold a copy of the information (a master database, with information about all the states); it will also be responsible for backup of all the data and all the regular server tasks.
- ii. Once the data starts to accumulate at site specific servers, data concentration at a particular site/sites might be huge compared to different sites, hence the performance issue.



**Figure 3.4 – Database servers scattered across the region**

Obviously, the second network topology [26] is a better option. However, as the data will be growing continually, we will need extremely powerful search functionality built into the database to retrieve a large dataset. As highlighted above, a Microsoft SQL server will not be capable of handling such large information retrieval tasks.

We could potentially reach a dead end from the above method. In this section we will discuss how genetic algorithms can be used to overcome the search problem.

In a recent article published by NewScientist.com [27], Will Knight discusses how genetic algorithms could develop strategies for web servers when caching data.

Transferring data across the internet repeatedly can be inefficient and costly, so networking companies have developed ways of temporarily storing, or "caching", data at different locations to reduce costs and increase download speeds, but figuring out where to store data and how long to store the data can be a difficult task.

Pablo Funes of Icosystem, a US based Company and Jurgen Branke and Frederik Theil of the University of Karlsruhe in Germany used evolved genetic algorithms for specific types of network. As mentioned by Funes, the algorithms take known variables, such as the number of times a piece of data is requested, the number of points it has to pass through and its overall size, and work out whether it should be stored, and if so, then for how long.

The key to finding an efficient algorithm was "evolving" it from a population of randomly generated ones.

# Chapter 4 - Methodology

## 4.1 The software methodology

During the course of this project, several prototypes will be developed to see if our theory can be justified when applied in practical situations. To design the prototypes, software engineering principles, tools used and best practices employed will be discussed.

As in any engineering discipline, software engineering also has some structured models for software development - namely agile software development, Systems Development Life Cycle (SDLC), Rational Unified Process (RUP) to list a few.

The above mentioned are concepts in developing software are not 'must follow' rules. Each of these concepts have methodologies that fall under different categories. For example under agile software development we have crystal methods, dynamic systems development model (DSDM) and scrum. Under Systems Development Life Cycle we have Rapid Application Development (RAD), waterfall model, Joint Application Development (JAD), etc.

The diagram below shows the development life cycle of traditional systems (also known as a waterfall model), highlighting the stages in which modelling is typically used. However, modelling, and the techniques we will look at, can be, and are used in every type of system development process. The below diagram is itself a model.

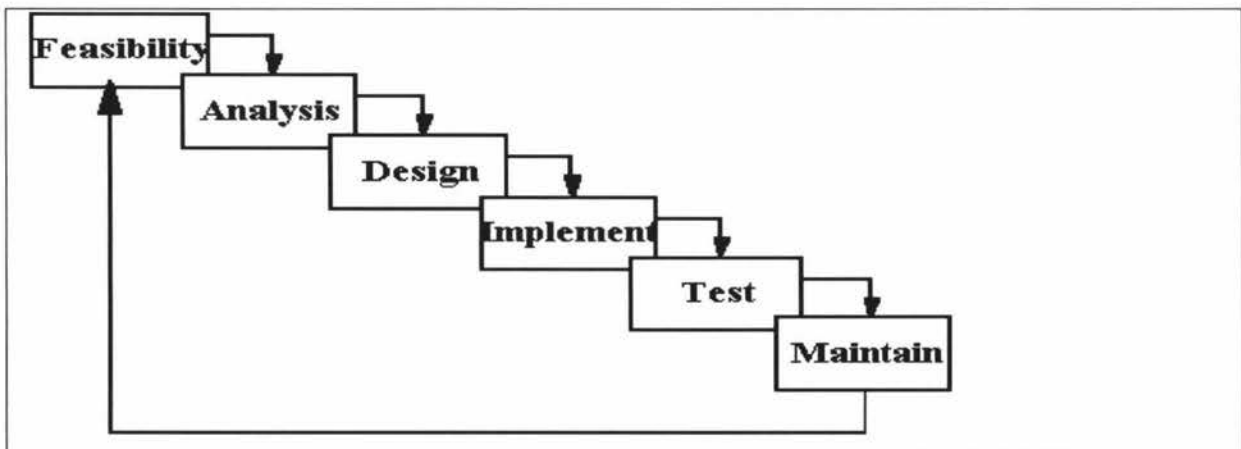


Figure 4.1 - Traditional Life Cycle a.k.a Waterfall model

It is not mandatory to use all the stages of the traditional life cycle. This is a structure you could use to develop software and is versatile. Almost all software models can be customised to suit your project needs, however, it is important to understand that not all projects are the same. They should be treated differently and various methodologies should be taken into consideration when designing software. Obviously, the methodology itself will point out the important factors needed to be taken into consideration when used. When discussing the proposed prototypes, it will become evident that we can skip a stage of the above software development model.

## 4.2 Why consider different software development methodologies?

This is, no doubt, a question that is often asked - isn't software development all about programming? The answer is yes and no!

Just like any other engineering discipline, software engineering has design principles that define concepts which provide guidelines in creating better software. There is programming involved in most of software development projects, however, the requirements need to be determined. Architecture needs to be designed in order to build a solution that meets the customers' (users) requirements.

You wouldn't build a house without first having an architectural design, as there are a lot of factors to consider - for instance, the size of the house. There is also the factor of experience - is the builder experienced? However, having mentioned that a software development project may not always need a methodology, it depends on the size of the project. Planning plays a critical part here and the chosen methodology will provide the guidelines for project management. As the size of the software development project grows, the chosen methodology becomes crucial to the success of the project - as does the successful implementation of the methodology.

Building software is similar to building a house - building good software needs a plan.

Here we will discuss which plan, or software methodology to be precise, was followed during the course of this thesis. This also doesn't mean that our software was successful and other projects that do not follow a methodology will fail. Just as planning or designing a software methodology is important, more critical is the execution of the plan.

### 4.3 Different stages in a software development lifecycle

The **project selection phase** is the **first phase** where you decide on a particular project. In most cases you would have decided on a project before you considered a software development methodology – or, if you work for an organisation, the project you work on will be decided by your managers.

The **second phase** - the **feasibility study** - is carried out to see if the project is viable. In this phase you are asking questions, for example, if the project is possible, are there any alternate solutions? Is the project expected to be completed in a set time frame and budget? Is there sufficient skilled manpower? Can the staff go through training if needed?

In a project you would normally carry out four types of feasibility studies, as listed below along with a cost/benefit analysis.

- Is the schedule feasible – is it possible to build a solution in a given time?
- Is it economically feasible – is the project possible, given the current resource constraints?
- Is it technically feasible – is the current state of technology able to come up with a solution?
- Is it operationally feasible – is the system developed, and will it be used?

The above mentioned procedure is not important in a research project; however it is a good idea to conduct one, as it gives a good indication as to whether it is worth spending the resources and/or how successful a software project can be.

The **third phase** in the development model is **Analysis and requirements gathering**. In this phase you will gather requirements of the business which enables the developer to gain a better insight into the problem and help develop a better solution. This is a very crucial step as it defines the scope of the project. A prototype is also presented by the developer for the users to get a feel of how the application will behave. User interface design is also taken into consideration and both parties (the development team and the customer) come to an agreement on the functionality of the solution.

The second part of the third phase is the system and software design. During this phase the solution is worked on, programmers do the coding and a logical solution is created. During this phase there's only functional testing done - no user testing is performed.

During the next phase, documentation and unit testing is carried out, in which smaller units of the completed software are tested for logical errors, along with test cases, documentation and user documentation.

The next phase is system testing. This is where the users test the system, to see if the system functionality is what they asked for and whether the user interface is easy to use.

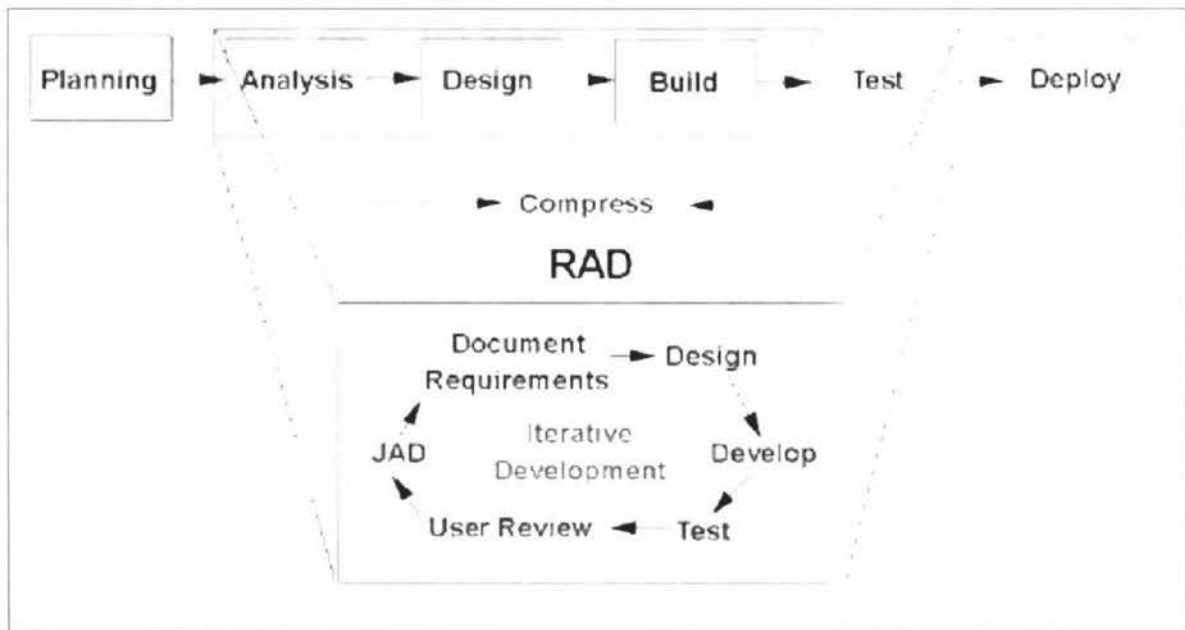
From then on, the last phase is basically maintenance and enhancements of the software.

### 4.3 Methodology Used

The choice of methodology for this project was Rapid Application Development (RAD). There are several reasons why this was the obvious choice but before we begin to look at why, we should start by looking at what RAD is?

### 4.4 RAD

A problem with traditional software development model (see Figure 4.2 below) was that the process took a long time to design a solution, and when the final solution was implemented, the user's requirements had changed by then. So the system that was implemented either lacked functionality or ended up not being used.



**Figure 4.2 - RAD optimised from the Traditional software model**

RAD has six core elements [28]: prototyping, iterative development, time boxing, team members, management approach and development tools. Some might argue that there are four core components not six, but this is because they usually combine some stages into a single stage.

Rapid application development was mainly developed to increase the speed of development and quality. Speed of development is mainly due to the Computer Aided Systems Engineering (CASE) tools used in development - and quality is measured by the degree to which the user's requirements are met. This is done by keeping the user involved during the analysis and design stages.

RAD also has a few disadvantages associated with it. These are, reduced scalability, as an application developed using this method, starts as a prototype and evolves into a finished product. Reduced features occur due to time boxing, features are pushed to later versions in order to finish a release in a short amount of time (Appendix B).

## **4.5 Why RAD?**

During the course of this project, Rapid Application Development was the obvious choice as highlighted below:

- This is a research project, so we did not have a specific goal in mind; we had a purpose to see whether there is a method of managing health in a smarter way.
- Applications built using RAD start with prototypes; therefore we had to build prototypes to test purpose in terms of functionality.
- There were several factors that we did not need to consider, for instance there were no scalability issues, no user involvement from the beginning of the project, so the theory was tested and not the user interface.
- The prototype that was built to test the theory was considered for unit testing and the development of the prototype was stopped once the desired functionality was achieved.

As mentioned above, Rapid Application Development was the obvious choice as the speed of development was much faster than a traditional approach. However as this is a research project, we did not have specific requirements to meet.

Our theory is being evaluated here, so we can minimise the number of stages Rapid Application involves. For instance one of the important components of RAD is team members. This can be eliminated as this project was an individual project, although the supervisor is counted as a member of the team, as he is one who drives the project.

Management is also taken care of by the supervisor to a certain extent, and he is responsible for regular meetings updates on the project and the person conducting the research project is also responsible for management. In his/her case project management, self motivation is the key here. Almost all of the stages are quite clear regarding what needs to be done,

## 4.6 The software Architecture

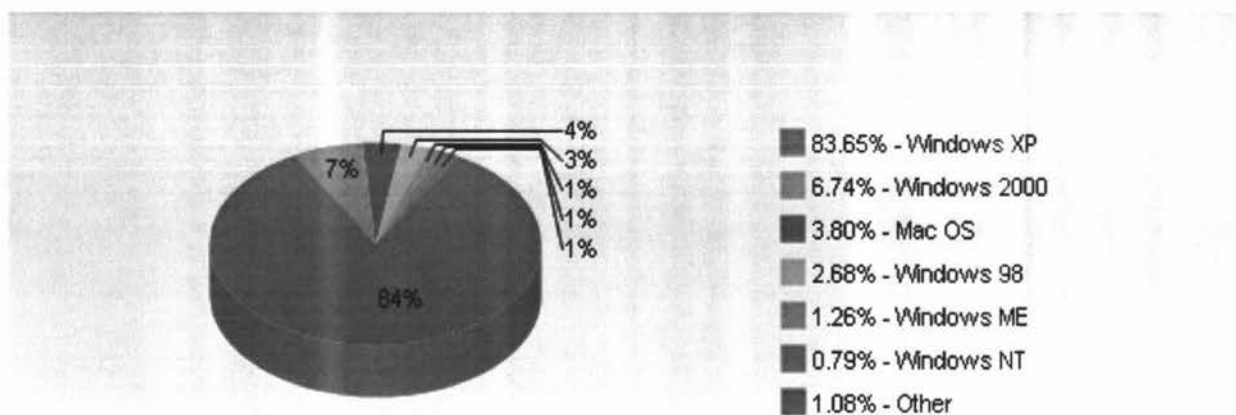
Software architecture highlights the components that go into making the software and how they integrate with each other. One important part of the software architecture is that it details out the high level design documentation that will better facilitate the process of development and apply the reusability principles of software development.

## 4.7 The choice of software development platform

The choice of software development platform was the .NET framework by Microsoft.

There are many reasons why the .NET framework was chosen. To gain insight into the reasons behind the development platform choice, we will focus on the traditional problem with programming language platforms and their run time?

According to [www.marketshare.com](http://www.marketshare.com), Microsoft has captured 90% of the market (refer to Figure 4.3). Most of the business and almost every household PC runs an operating system by Microsoft. It will become clear during the course of this thesis why Microsoft technology was the obvious choice for development.



**Figure 4.3 – Software Market Share [29]**

Microsoft has been investing substantial resources into developing their .NET platform (pronounced dot Net). The .NET framework proposes/offers? a lot of advantages over the

existing technology. Currently the .NET framework is packaged as part of the Microsoft Windows XP (Including Service Pack 2) operating system or a free download from the Microsoft website.

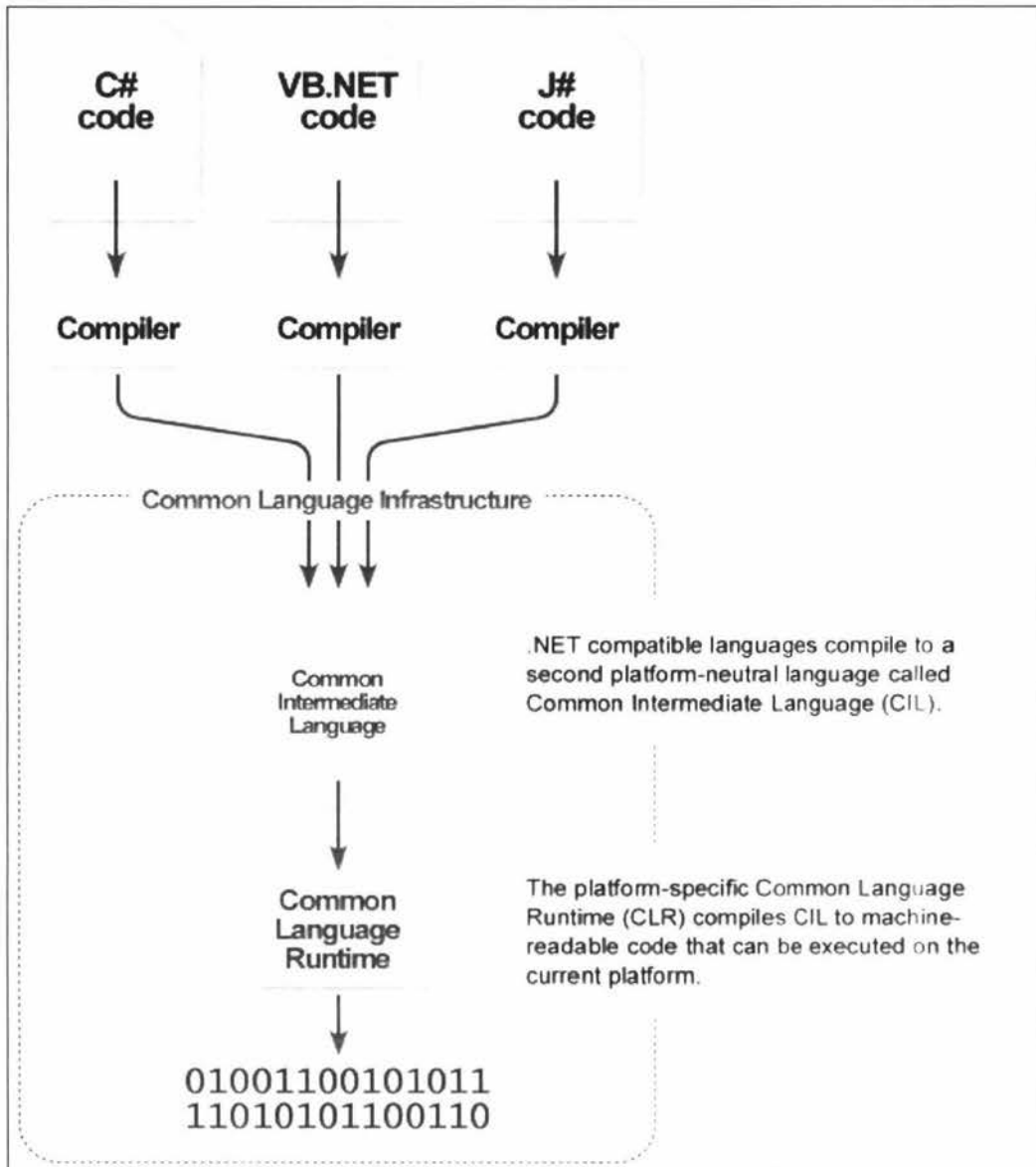
The basic idea behind the .NET framework is that it provides a common platform for developers to implement functionality without worrying about the underlying architecture, the source code can be written in any .NET specific language and the source code is compiled to a common intermediate language regardless of which programming language it was written in.

An example would be - if we have a group of 4 people, all of whom speak a different language, there will be a communication barrier. However, if the language is translated to a common language for instance English, then the problem disappears, as everyone can communicate effectively without worrying about the information/knowledge transfer.

This is, in essence, exactly what the .NET framework does; it removes the barrier of different programming languages by providing a common platform.

There are also other benefits from the framework - one very important feature for developers is memory management. In traditional programming languages such as C++; developers had to explicitly implement memory management functionality but this is all taken care in the .NET framework.

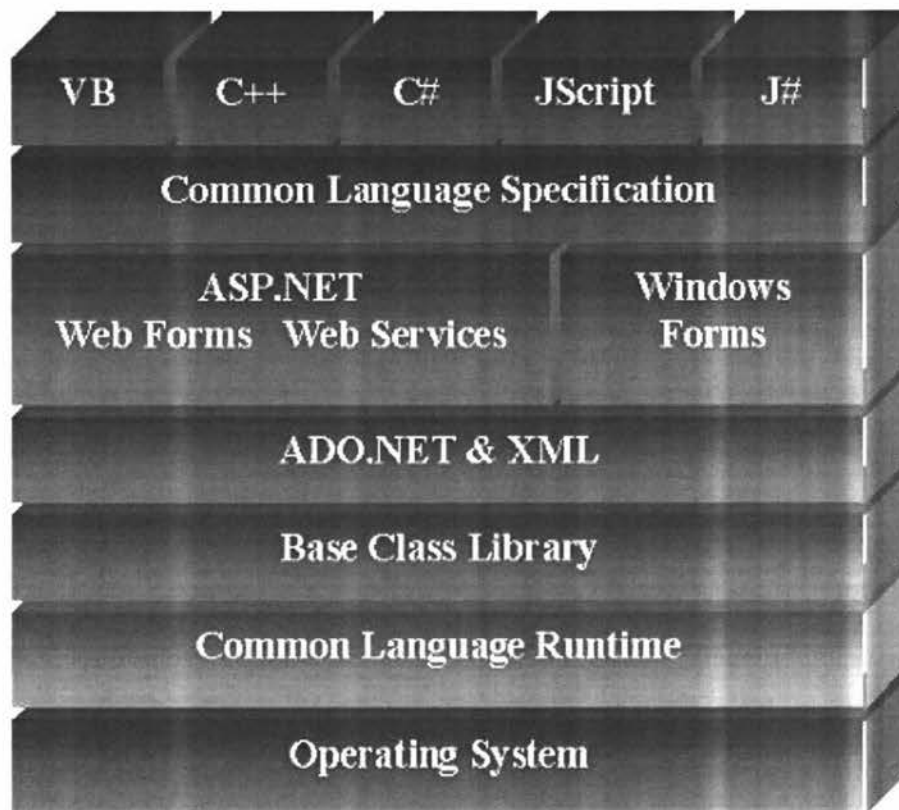
A highlight of the .NET framework is that the code targets the Common Language Runtime (CLR) (as seen in Figure 4.4) and is said to be managed code (meaning controlled by the .NET framework); hence the source code cannot directly execute instructions on the machine processor, making it safe.



**Figure 4.4 – Common language runtime (CLR) [31]**

As seen in Figure 4.5, the .NET framework resides just above the core operating system; hence operating system functionality is also available via the .NET framework.

Version 2.0 of the framework has been released and version 3.0 is under beta testing. Obviously, newer versions, meaning new functionality are available with each new release of the framework.



**Figure 4.5 – The .NET Framework [30]**

As seen from the above diagram and the theory explained above, it is evident why the .NET framework was chosen as the development platform. We can choose any .NET compliant programming development language and build a desktop application, a web based application or even a smart phone or PDA application and that's not all! The developed application is not Microsoft Windows Operating system specific.

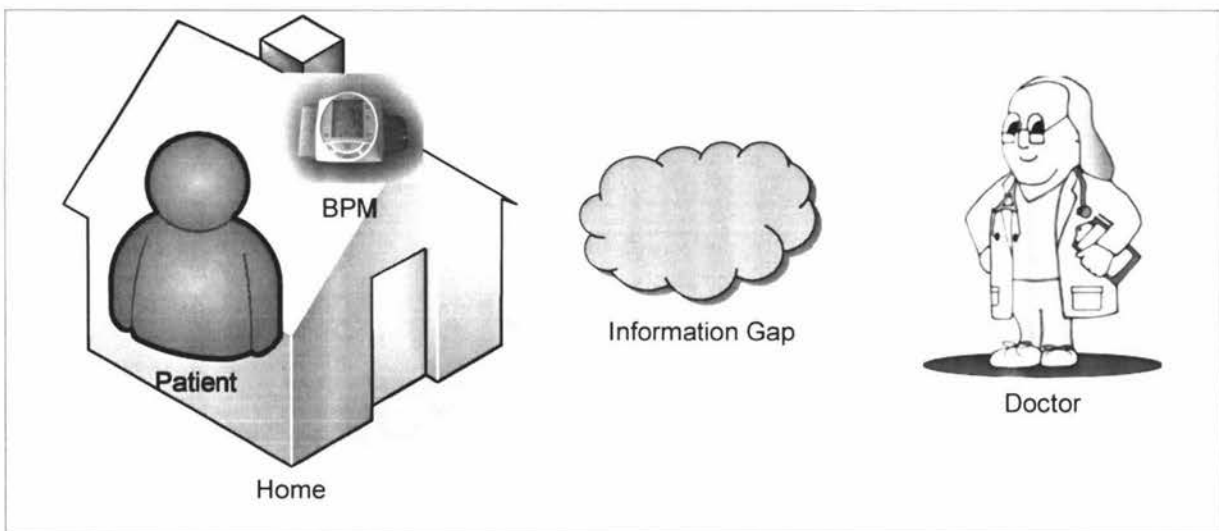
If you are running Linux or any other UNIX based open source operating system, there are open source versions of the .NET framework [32] which will work on any version of Linux, and being 'open source' means you can download it for free and obviously, as explained above, the source code compiles to a common intermediate language, hence the software will run regardless of the underlying operating system.

## Chapter 5 - A new system

### 5.1 What is a new system?

The idea of a new system is to provide users (in this case patients) with a system where they can update their health related data (in this case, blood pressure data, as an example) with the data being stored in a centralised environment which is available to doctors on a request basis.

The aim of the system is to bridge the information gap that currently exists between the patient and the health professional, as seen in Figure 5.1 below.



**Figure 5.1 – Current situation**

Let's consider a situation where a patient suffering from hypertension visits a doctor to get his/her blood pressure measured and the readings at that point are normal. Now this could be due to white coat effect [33, 38].

From the time the patient leaves the doctor's dispensary to the time the patient comes back for a second visit, there is no data available for the doctor, from which he/she could better diagnose the condition of the patient.

This information gap is of significance to us and during the course of this thesis we will focus on highlighting different approaches to see where this gap could be minimised, or in some cases, reduced.

## **5.2 So, why do we need a new system?**

The reason is simple; we are trying to reduce the information gap that currently exists between the medical professionals and their patients.

In terms of a new system; it is not a ground breaking invention, but rather a combination of applications developed to give the patient/user options in keeping their medical data up to date. Where geographical distance is not a factor, the patient can maintain his/her data provided they have access to a computer. Using the tools that are currently available, we will be designing a Microsoft windows based application and a web based application in order to centralize the data. This will give the health professionals more power in terms of patient health data being readily available to help them make more informed decisions.

Smart questions such as, is the patient monitoring their blood pressure? Is the patient's blood pressure under control? Are they using their medication? Are they measuring their blood pressure on a regular basis? Is their diet controlled? Are they involved in any physical activity? These are important question for a medical practitioner to ask and the data from these questions can help medical professionals gain a much more detailed understanding of the lifestyle of a patient.

The focus of this thesis is to answer the above questions, we will do this by developing a series of prototypes to demonstrate, firstly, whether the new system will improve a patient's blood pressure management and secondly, whether it is providing the health professionals with the data they need to make informed decisions.

### 5.3 The new system

This problem will be approached in 3 stages. The first stage will be to design a data collection system, the second stage will be to use the data and perform statistical analyses and the third stage will be to prove whether a statistical model could be used to improve blood pressure management. If a prototype cannot be developed, then we will discuss the theory.

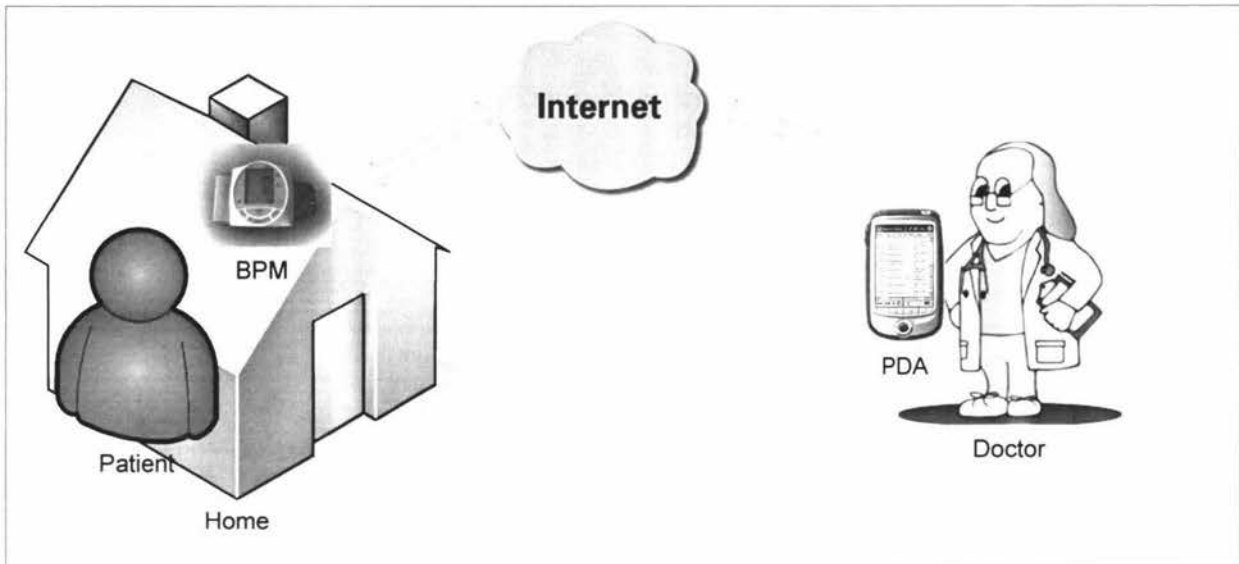
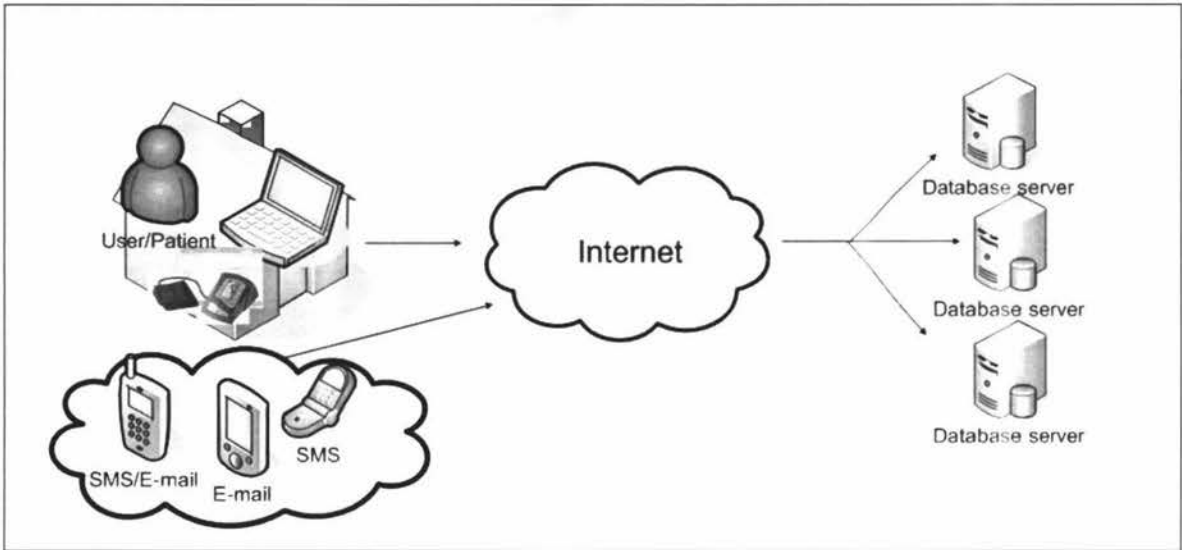


Figure 5.2 – Proposed Idea

#### 5.3.1 Stage 1: The data collection phase

Before looking at the different aspects of data collection, we will discuss how the home blood pressure measuring device captures the data on its onboard memory. The device has a RS232 interface; hence using a RS232 to a USB cable, we can plug the blood pressure device into the PC. Once this connection has been established between the home blood pressure monitoring device and the PC, the vendor-supplied software can be used to extract the readings of the blood pressure device. The hardware is capable of storing 99 readings (systolic, diastolic, pulse, date/time stamp) and the data is extracted in XML format which makes all the XML standard complying languages able to easily access the data. Also, since XML was mainly designed to carry data, it can bypass corporate firewalls.



**Figure 5.3 – Data collection phase**

The above approach would be suitable to use if the Application Programming Interface (API) is not available from the vendor. In most cases, though, there is an Application Programming Interface (API) available to allow developers to enhance the application. However, the datasheet for the hardware can be easily obtained if one opens the box and examines the hardware to find out which IC is being used in the circuit (Note: this should only be carried out with the manufacturer's consent and without violating any copyright laws).

The Application Programming Interface (refer Appendix C) contains a list of functions that can access the hardware. Using these functions, a developer can easily access the data stored in the memory of the device and, as mentioned in the API, the data can be directly accessed from the application - hence the data can be retrieved from the hardware.

However, this retrieval part of the data is considered as out of scope, as it would better fit into electrical or information engineering disciplines.

The blood pressure monitor that was used during the course of this thesis (please refer to Appendix F) comes with bundled software that extracts the data from the hardware and stores it in the form of XML. The Application Programming Interface (API) can be used to access this feature of data extraction. We have made the assumption here, that the data stored on the blood pressure monitor is extracted via a software utility that is supplied with the blood pressure monitor.

In the current situation the software supplied with the blood pressure monitor extracts the data from the hardware and stores on the PC as an XML file; however, this is a manual process and will need to be automated. The data stored within an XML file can easily be migrated to a MS SQL server or a MySQL server. Most of the RDBMS available in the market today come with built in migration wizards, which guide the user in seamless data migration. This, however, will be taken care by the software - not currently implemented, but will be done as part of the future enhancements.

The blood pressure monitor (BPM) used during the course is a microlife® model no. – BP3AC1-1PC (please refer to Appendix F for more information on the blood pressure unit used).

In a normal scenario a user/ patient would measure their blood pressure once or twice a day. The data is stored on the onboard memory of the BPM. The user will have to connect the blood pressure monitor to the computer to transfer the readings from BPM's onboard memory. The Windows based application – prototype, which was developed during this project; extracts the data from the BPM, stores it in the locally installed database; and when the PC connects to the internet, the data is then updated onto the central server. This leaves the user's blood pressure data (health related data) synchronised between the user's PC and the central database server.

### **5.3.2 The Personal Computer based application**

In the windows based test application developed, the data is stored in a relational database. The database that gets installed on the user's PC is a Microsoft SQL server express edition (a free version with a few limitations).

The application can also be set to check for an active internet connection and if there is one, the data is uploaded to a central repository. This is one of the core parts of the application as this approach would centralise the data and the doctors are virtually a click away from viewing patient information.

The central repository (this is in the database server hosted at a data centre or at a managed site not locally on the user/patient's PC) will be either an open source version of MySQL or a

Microsoft version of SQL called Microsoft SQL server, but this means there would be a cost associated with it. Even if the initial choice were to use Microsoft SQL server, in the near future, once the data starts growing in size, the central repository will have to be migrated to MySQL with clustering, as a single server would not be able to hold the data (central repository will be discussed in more detail in a later section)

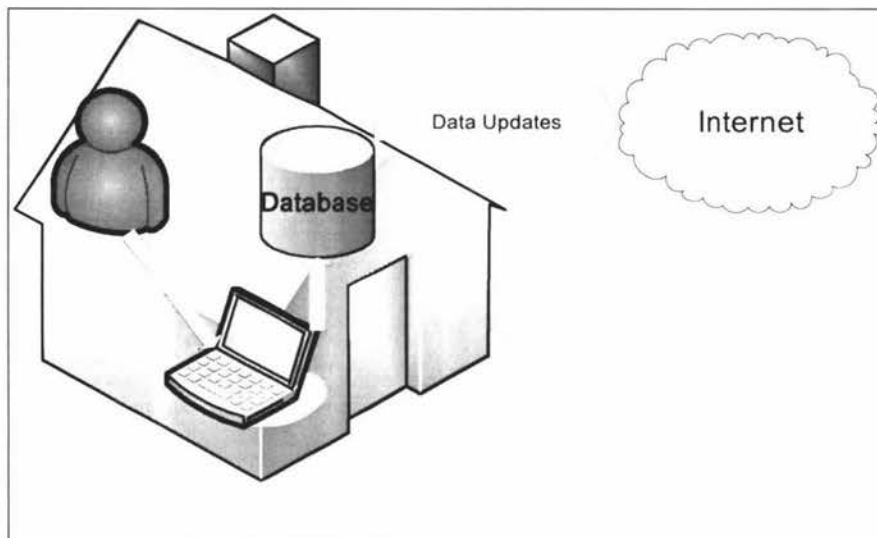
Dr Steve Warren and Richard Craft at Sandia National Laboratories have published an article [34] which talks about developing a framework to centralise the data capture for any kind of health related data which is to be stored in a central location, whether it be from a primary school, general hospital or a prison. All health related information is to be stored in a distributed medical information database. Obviously security is an issue that will need to be closely looked at, but there are many encryption algorithms that make electronic transactions secure. This step towards data centralisation would eliminate data redundancy and the data is made readily available to the medical health professionals.

The Personal Computer (PC) based application also has a few smart features built in. For example the initial phase where the user needs to be set up, there is an option for specifying threshold levels (obviously a realistic number specified by a medical professional will have to be recorded here) and once the threshold level has been set, every data item input from the user is compared with the threshold. If the value entered by the patient is above or below the threshold, then the patient is warned and if he/she specifies that the values are correct, a report is sent to the doctor and the doctor can respond to it.

In general, a patient suffering from Hypertension would have to follow a procedure prescribed by his/her general practitioner. For instance some patients might be asked to measure their blood pressure levels once every day. In extreme cases patients might be asked to measure their blood pressure 3 times a day for the doctor to get a better insight into the patient's condition. Patients can also specify a time for the system to send the data, for example, the data should be transmitted only when there is an internet connection or at scheduled times.

### 5.3.3 Different aspects of the windows based application

The application prototype developed during this research project is a basic data collection and statistical tool and the application also acts as a data service broker between the personal computer and the central database server. The application developed does not have any security built around it, however this is for demonstration purposes and security would be a project on its own. The windows based desktop application uses a Microsoft Access database as its data store; it was the best choice for windows application backend, which stores patient information locally on the patient's/user's machine. The information is then updated to a central database server (as there was no central database server readily available, the scenario was emulated on the development PC)



**Figure 5.4 – Basic dataflow model of the windows based application**

The local database used by the windows application is a MS SQL 2005 Express edition database [34]; this was due to the availability of the product. Microsoft Access could have been used instead due to the simplicity of Microsoft Access (MS Access database is an extremely light weight version of MS SQL Server). However, the MS Access database can be migrated to a MS SQL Express edition for scalability purposes.

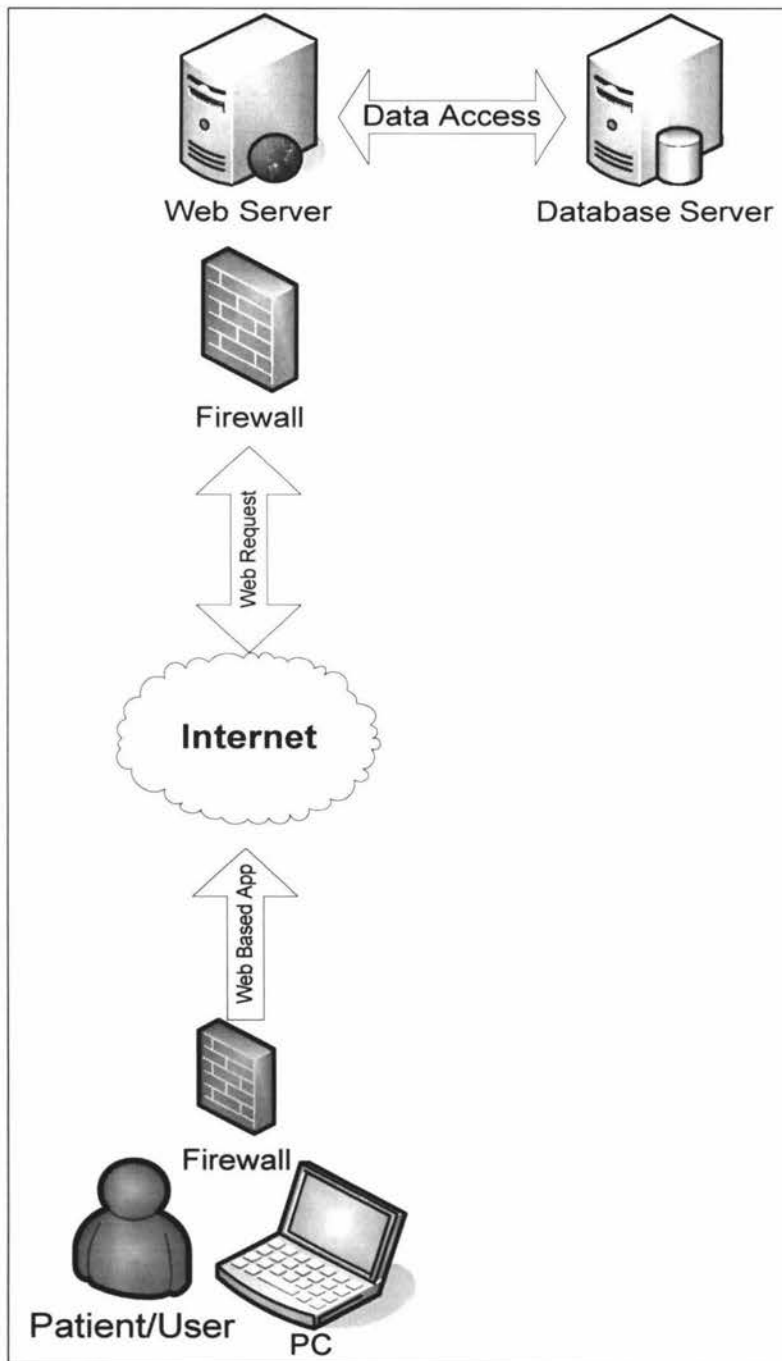
Please refer to Chapter 6 for a detailed overview about the application user interface.

Additional benefits of the system would comprise the following:

- The system will send reminders to the patient and if he/she forgets to enter their readings, the system e-mails them and they can enter the data within their e-mail (so no more opening another application, just to key in some values) and send it back to the system for the data to be updated.
- Patient's blood pressure (in the future medical records) records are always up to date and easily accessible from anywhere over the internet. Hence, if the patient is constantly moving around, he does not have to worry about the medical records not being available
- The new system would continuously monitor the patient data, and if any anomalies lie within the dataset, a report or an alert can be sent out to the medical professional.
- The system can carry out statistical analysis of the patient's results and present the patient with daily, weekly or fortnightly reports
- The system can send dietary and physical activity recommendations directly to the patient (this would be a subscription based service; it does not have to be a pay service but information is only accessed if the user signs up and agrees to the terms and conditions).

### 5.3.4 The web Based Application

The web based application basically takes data centralization process a step closer in achieving our data centralization goal. The idea behind the web based application is that the patient data is stored in a secure, central state. This web based approach would provide the patient/user a way to update their health records and at the same time make the data readily available for the health professionals.



**Figure 5.5 – Web based Application data flow model**

Health professionals can react instantly to the data available by means of e-mail or information can be sent directly to the patient's/user's PC. By having updated patient data readily available, health professionals are able to make informed decisions.

Most of the new generation mobile phones come with WAP (Wireless Application Protocol) enabled web browsers. Using this, the patient does not need to have access to a personal computer as he/she could upload their data via a mobile phone, and this gives the patients freedom to update their health related data on the move.

### 5.3.5 The components of the web based application

The below diagram highlights the different technologies used for running the web based application.

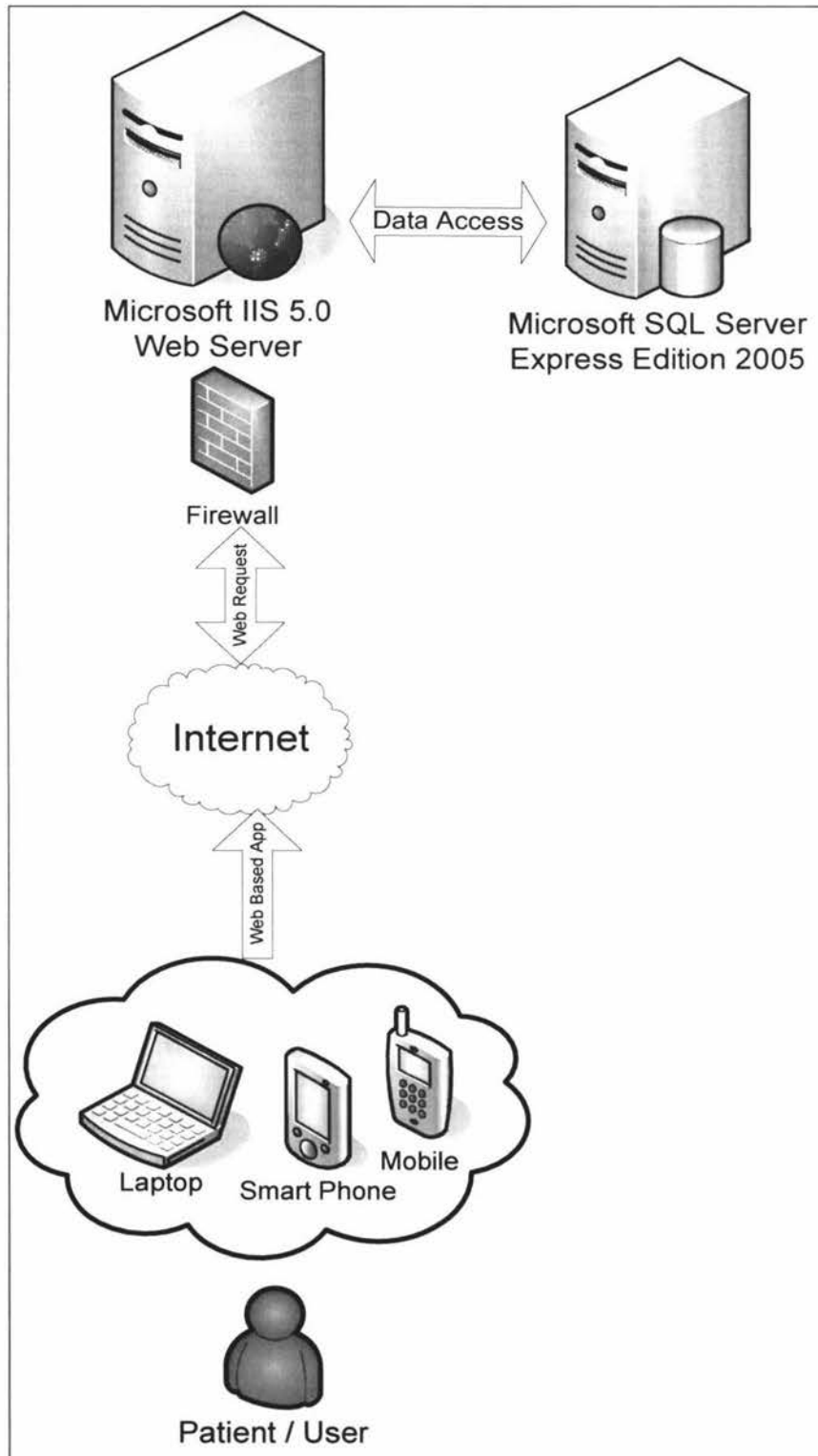
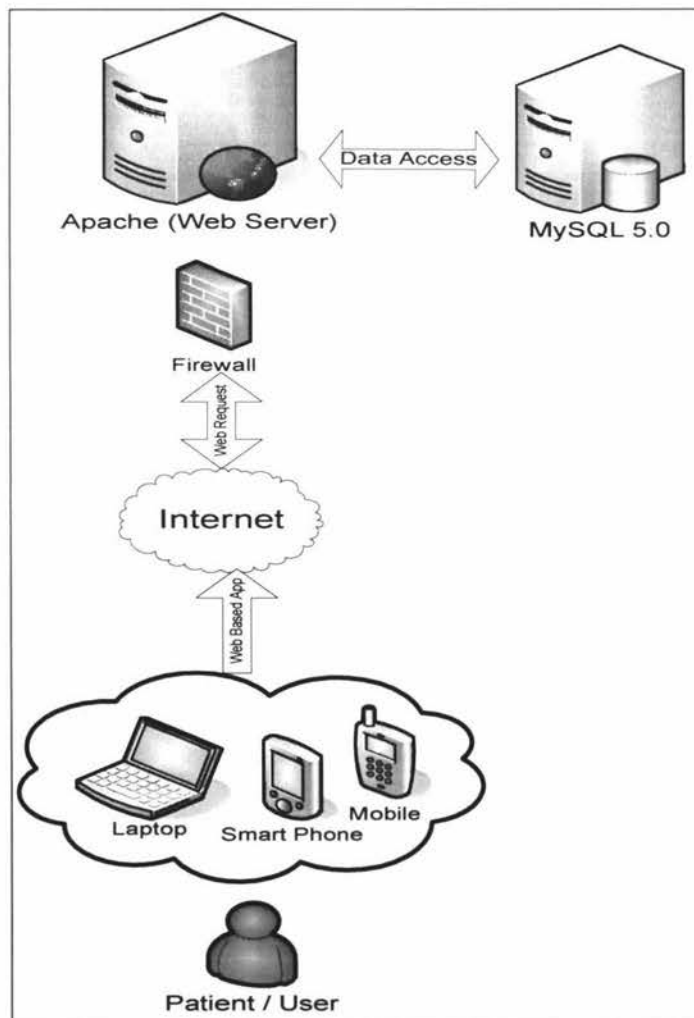


Figure 5.6 – Web based application architecture

The web based application developed during this course was hosted locally on the development machine. The technologies used to develop the web based application are as follows: the web based application developed was in ASP.NET using Microsoft Visual Studio 2003 academic edition, the back end database was a Microsoft SQL 2005 Express edition database and the web server running the ASP.NET application was a Microsoft IIS 5.0.

As mentioned in the previous section (the windows based application) the database can be easily migrated to a Microsoft SQL 2005 server or a free version MySQL 5.0 database server. The web server used is part of Microsoft Windows XP Professional edition, however to save on the licensing cost, an open source web server such as Apache can be used to the above application as long as the .NET framework is installed.



**Figure 5.7 – Web based application open source architecture**

### 5.3.6 The central repository

The central repository of choice here was Microsoft SQL server Express edition. The main reason for using the Microsoft version of SQL during this course work was mainly due to experience using this version of SQL server. You can use any version of SQL here, whether be it Microsoft SQL, MySQL or Oracle. A good option for saving licensing costs would be to use MySQL, which the world's most used open source database.

Our choice was to use the Microsoft version of the database server, Microsoft SQL server express edition; this is a free edition for building light-weight applications, however, a drawback with this edition is that the complete database cannot be larger than 4 GB.

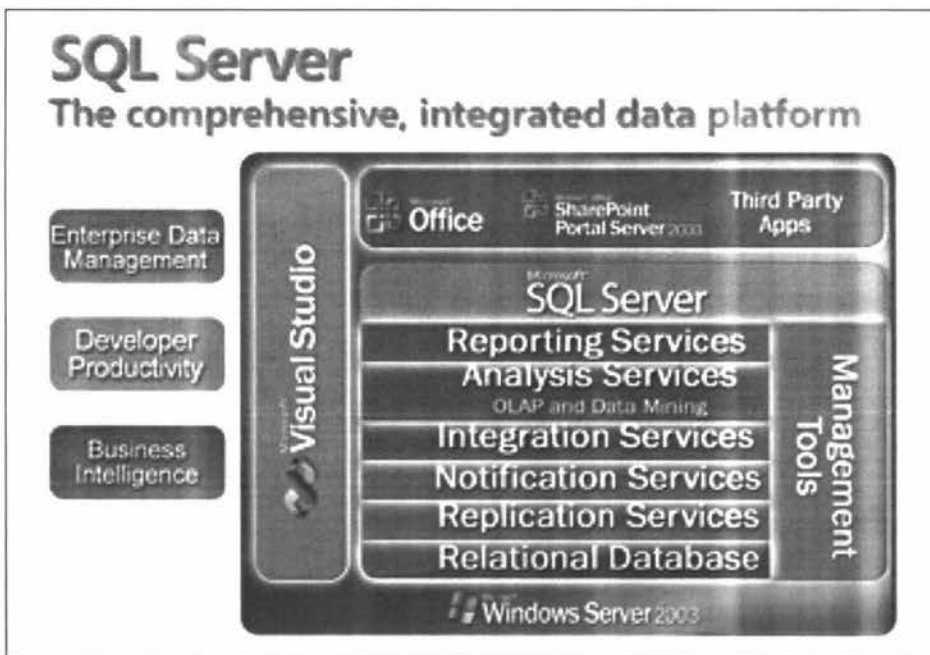


Figure 5.8 – MS SQL Server Architecture [35]

## Chapter 6 - Application Design

This chapter details the prototype application design carried out during the course of this thesis. Here we will discuss the basic functionality built within this application that would help the patient better manage their blood pressure.

### 6.1 Windows Based Application

**Primary User Interface** – As shown in the figure below, the main user interface is simple to navigate.



**Figure 6.1 - User Interface of the windows based application prototype.**

As seen from the Figure 6.1, the user is presented with the easy to navigate menu. The top menu bar lets the user get directly to the user data, reports, profile based suggestion and change the application settings.



**Figure 6.2 - Patient Profile**

The above figure (Figure 6.2) shows the features available when the Patient Profile button is clicked. By clicking the patient profile button, the user can add a new patient/user, edit a patient/user's details, view a patient's/user's details and update patient data.

**Add New Patient**

First Name

Last Name

Date of Birth

Address

Height

Weight

Gender  Male  Female

Smoker  Yes  No

Diabetic  Yes  No

Cholesterol

Family History

Family Doctors Email Address

**Save** **Cancel**

**Figure 6.3 – Add new patient**

The administrator can add a new user as seen in Figure 6.3. This can be done by clicking on the “add new patient” button.

The above win32 form captures the basic mandatory information needed for the patient profile to be created.

**Update Patient Data** [X]

Please select the patient you would like to enter the data for?

SYS (mmHg)

DIA (mmHg)

Pulse (/min)

**Save** **Cancel**

**Figure 6.4 – Update User/Patient Data**

The above figure shows the User Interface for updating user/patient data. This information is updated against the patient ID, stored separately with the database table, so it can be updated with the central database server.

This lets the patient/user update his/her blood pressure data. The interface is designed with blood pressure data as the focus, however not limited to - and can be later extended to cater for other medical conditions such as asthma etc

The above form (Figure 6.4) has a few smart features embedded within it, for instance when the patient/user enters the data in the form and clicks in the save button, the algorithm checks if the data entered is numeric and compares it with the threshold defined during setup. If the values are higher or lower than the threshold and the patient/user confirms that the data entered is correct, the result set (sys, dia and pulse) are e-mailed to the family doctor and the doctor can decide whether urgent attention is needed. For an overview of the algorithm that performs a quick data validation procedure, refer to Appendix D.

The screenshot shows a window titled "Statistics" with a close button in the top right corner. Below the title bar, there is a label "Please select a Name:" followed by a dropdown menu containing the name "Faiz" and a "Go" button. Below this, there are six rows of data, each with a label on the left and a text input field on the right:

<b>Average SYS</b>	121.125
<b>Average DIA</b>	70.5
<b>Average Pulse</b>	84.5
<b>Patient Weight</b>	85
<b>Patient Height</b>	182
<b>Patient Colesterol Level</b>	No

**Figure 6.5 – User/Patient Statistics**

The above figure shows the user interface for user/patient statistics

As seen from Figure 6.5, the average sys, average dia and average pulse are calculated in real time, so the above analysis is always carried out on up-to-date data. The rest of the items are read from the database. This view provides the user with an overview of their blood pressure, This information can also be configured to be sent directly to the doctor via e-mail, as it can be of use to the doctor for a daily or weekly diagnosis of the patient.

## 6.2 Reporting feature of the Windows application

The windows based application has a reporting feature built inside it. This lets the user/patient view their blood pressure statistics. The report has been designed to illustrate the purpose; the reports can be customised in the future to serve the requirement based on the health professional recommendations.

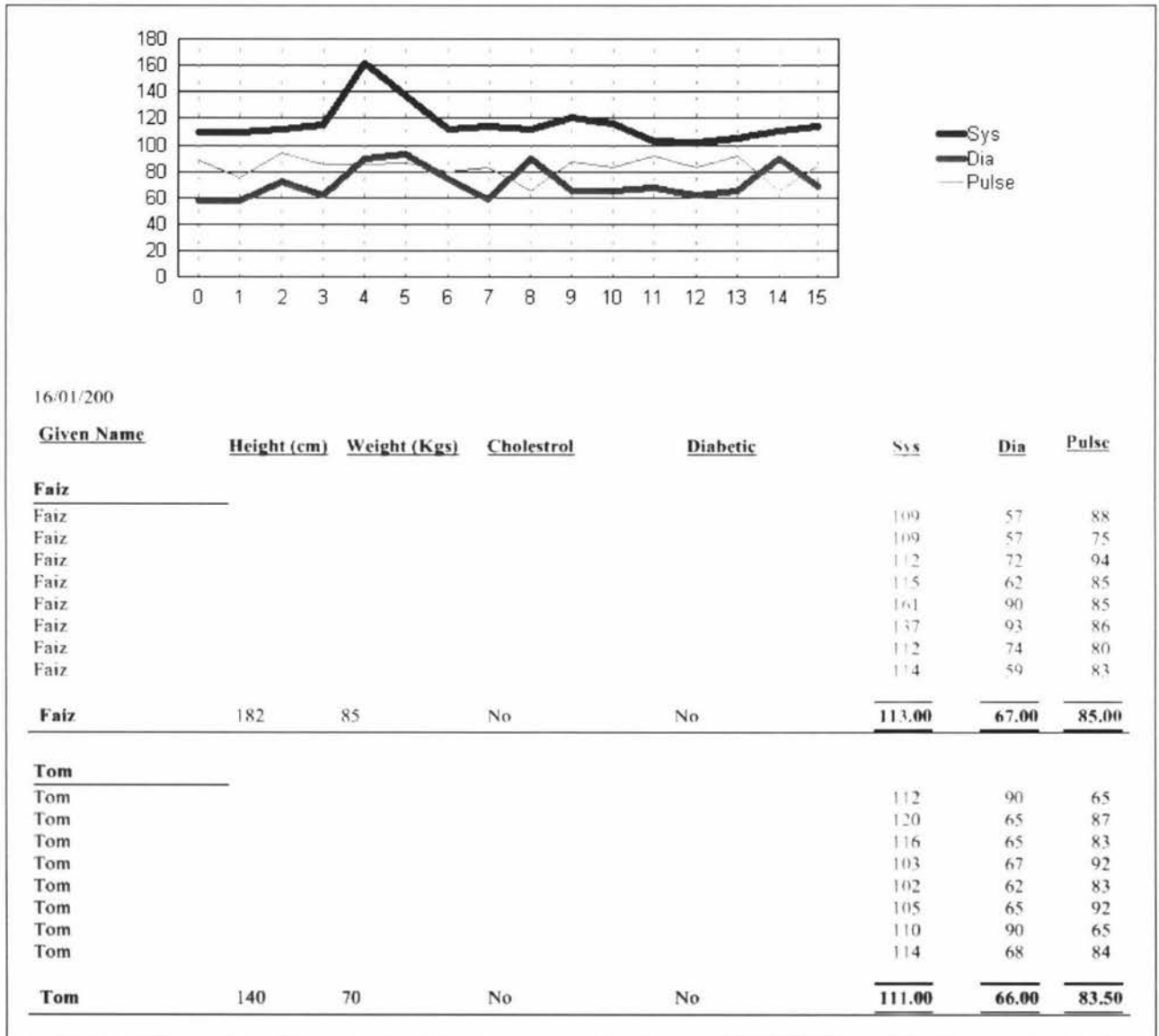


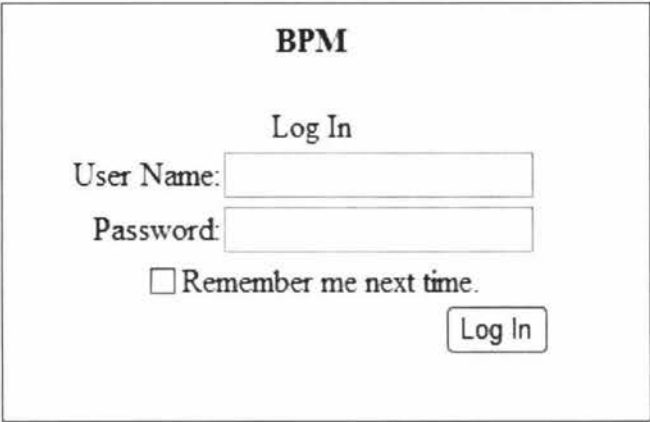
Figure 6.6 – Patient report

Since the technology used in developing the above reporting feature is compatible between windows applications and web applications. The same reporting feature can be used in the web based application prototype.

### 6.3 The web based application

The web based application provides the user/patient with an online portal where he/she can populate their data. The current state of development allows the user to populate their data. The advantage of the web based application (i.e. in production environment) would be its accessibility. The application would be accessible from anywhere in the world, provided the user has access to a PC with an internet connection.

The below figure shows the logon screen, where the user/patient needs to login, prior to any operations being performed. This is the main page of the application.



**BPM**

Log In

User Name:

Password:

Remember me next time.

**Figure 6.7 – Logon screen**

This is the default page of the web based application; a user has to login prior to uploading or viewing their data.

Based on the user logon the user is presented with options for managing their blood pressure and related data. Similar to the windows based application the user can add, edit, view and update patient data. The user interface has been kept similar to the windows based application, so users only need to familiarise themselves with one user interface.

The user/patient can update their data using the **updateData.aspx** web page; this lets them manually update their blood pressure information. This is not the ideal way of updating

information as the patient can tamper with the results but it assumes that only one blood pressure monitoring unit can exist in a particular household.

As discussed in the previous section; the Blood Pressure Monitor (BPM) stores the readings on its on board hardware. This information can be extracted and is extracted as an XML file. The web based application allows the user to upload this XML file.

The system handles the data transformation that occurs from the XML file to the Database. However, a drawback from using this method is that the web based application assumes that only one person is using the BPM as only one file is created for all the readings. This could be a consideration as a future design principle for a newer version of the BPM unit, where multiple profiles can be accommodated on the onboard hardware of the unit.



The screenshot shows a web form titled "Update Data". It contains several input fields: "Date:" with the value "7/01/2007", "Name:" with a dropdown menu showing "Faiz", "Sys:", "Dia:", and "Pulse:", each with an empty text box. Below these is a section for "Upload XML Data file:" with a text box and a "Browse..." button. At the bottom are three buttons: "Submit", "Reset", and "New Record".

**Figure 6.8 – Update User/patient data**

The user/patient can view their blood pressure data by clicking on the viewdata.aspx page (Figure 8). This lets the user patient view their data.

This page (viewdata.aspx) gives the user/patient visibility on their blood pressure data. The most recent data is displayed on the last page, which can easily be configured to display the data on the first page.

Patient ID	Date	Sys	Dia	Pulse
1	03-7-2006	112	74	80
1	05-7-2006	109	57	75
1	07-7-2006	112	72	94
1	10-7-2006	115	62	85
1	12-7-2006	161	90	85
1	14-7-2006	137	93	86
1	17-7-2006	114	59	83
1	19-7-2006	109	57	88
1	21-7-2006	90	78	72
1	24-7-2006	78	78	78

1 2 3 4 5 6

**Figure 6.9 – Data View page**

As seen in the above figure, the view data page provides the user/patient with a simple view of their blood pressure related data. This information is retrieved from the database and presented in real-time. Hence any updates of the patient data are readily available for the user or their doctor to see.

- XML conversation to SQL data
- Upload this information to a central server via web service

The web based application can be enhanced to accommodate more features from which the user can benefit. For instance a diet plan can be customised based on the user data, an exercise plan specific for the user or even a reporting feature added to the existing application; the possibilities are endless. Additional functionality would make the user interact more often with the application and in turn make the user aware of their condition and how they could benefit from a little input.

In the above section we have seen both a Microsoft windows based application and a web based application developed to capture user data and an effort made to centralise this information.

We have looked at the windows based application which offers

- A simple user interface
- The application gives user/patient the ability to save their blood pressure related data in a database for future reference

- A reporting feature built inside the application which allows the user to view their average blood pressure data.
- The application also provides the user/patient information on how to manage their blood pressure level
- The information is stored centrally and is synchronised with a central database server when there is an internet connection.
- The web based application provides the user with the ability to update their blood pressure information and the information is immediately available for the medical practitioners.

In the next section we will look at the results of the experiment performed and see if the software application would indeed help a user/patient to better manage their blood pressure.

## **Chapter 7 - Statistical analysis of the results (Discussion)**

### **7.1 Background**

The user/patient was given a copy of the Microsoft Windows® based application prototype or a web based application prototype. As the web based application was not hosted on the Internet, each personal computer of the user/patient had to be configured specifically to run the application locally on the user's PC simulating a web experience (this was not possible on all the user PC's as some of the users did not run Microsoft Windows XP Professional; a requirement for running **Internet Information Services (IIS)**).

As discussed in Section 7.2, a group of users were asked to populate their blood pressure data using the windows based application or the web based application (where applicable) and a different group was asked to record their weekly data on a printed spread sheet. It was a mandatory task to populate the data; hence it was critical to maintain communication with the test subjects throughout the data collection phase.

In this chapter we will discuss how the experiment was conducted, how the data was gathered and the analysis of the results.

## 7.2 Data Gathering Phase

The users were each given a blood pressure measuring unit to acquire their blood pressure readings. There was access to only one blood pressure unit, which was given by the university for the purpose of this research and 2 out of 6 users had their own blood pressure units.

There were arrangements made for 3 more blood pressure units, so each user could have their own blood pressure unit. However, all 6 devices were calibrated to minimise inaccuracies in the data collected.

The readings obtained were not exactly the same (as expected by data acquisition devices, no two devices will calibrate to be the same) but were similar. On average the readings varied by +/- 5 units. Since normal blood pressure is a range and not a fixed number, a difference of +/- 5 units is considered acceptable. In this case there were no readings that were off the acceptable scale and hence, the experiment began.

For the experiment we had 2 different groups of users, Group 'A' who had access to the software (either Microsoft Windows based application or the web based application) and would only use the software to populate their blood pressure data. Group 'B' was given an individual sheet for each week to record their readings and they would only record their blood pressure data on this weekly sheet. The sheet was collected from them every Sunday during the period of the experiment (the users either posted their sheet or it was personally picked up by the researcher).

Both groups contained 3 users each who had high blood pressure and shared similar characteristics. The factors that could potentially affect the outcome of the experiment were controlled. For instance the age of the users was between 30 and 40; all the users were male and of Indian origin, all the users were non smokers, all the users were above their recommended body weight but not obese and all took light evening walks twice a week. Their condition was mainly due to improper diet which was not in moderate quantities.

Group 'A' consisted of 3 users who were given a copy of the software (Microsoft Windows® based application or web based application) to update their blood pressure readings. This was

installed on their personal PC and they were asked to update their blood pressure data once a week (as the blood pressure monitor stores the readings on its onboard memory). Obviously the users were taken through this process of measuring their blood pressure, storing the results on their blood pressure unit and entering them using the software. Once entered, they were shown how to reset the blood pressure measuring unit, in order to capture their new weekly data..

Group 'B' consisted of 3 users who were not given a copy of the software and were asked to measure their blood pressure and record it on the printed copy of the blood pressure data sheet (refer Appendix G) . The blood pressure data sheet was collected from the user's residence or posted by the user and this data was keyed into the system for data analysis.

The data collected from Group 'A' was migrated from the user's PC to the development PC which emulated a database server. This was a manual process as there was no client/server architecture setup as there would be in reality.

The users were asked to measure their blood pressure 3 times a week. 3 users had their own blood pressure monitoring device. The devices were calibrated to reduce any abnormal readings being captured. This was done by taking 5 readings each and comparing these with a brand new device used during the course of this thesis.

The users were asked to take 3 readings per week (Monday, Wednesday and Friday); Blood pressure was not taken in the weekends as consumption of alcohol and fatty foods is known to be higher then. The data was collected over a period of three months (from the 3<sup>rd</sup> of July 2006 till the 30<sup>th</sup> of October 2006). A total number of 51 readings per user were obtained

### 7.3 Explanation and Scientific Rationale

To reduce bias on data, some underlying assumptions and dependencies for the data and data collection method are mentioned below:

An assumption has been made that an individual datum (observation) collected is independent of every other i.e. a blood pressure reading obtained on a particular day is not influenced in any form by another blood pressure reading obtained on, for example, the previous day. Such an assumption has been made as even a minute degree of dependency among individual observations can distort the  $p$  value in a conventional manner also influencing the alpha (risk) level significantly.

For the purpose of this research report, the alpha or the risk level or the significance level is set at 5% ( $\alpha = 0.05$ ), as conventionally 5% is regarded as sufficiently small to reject the null hypothesis. Also in this research, the use of modern technology is considered as the independent categorical variable (IV) i.e. high blood pressure users using modern technology (Group A) and high blood pressure users not using modern technology (Group B). While Sys and Dia act as two dependent variables (DV) i.e. use of modern technology leads to variance in Sys and Dia. Since this research is attempting to explain the variance in the dependent variables for the two groups of the independent categorical variable, MANOVA has been used to explain this variance and an  $f$ -test to test the hypothesis.

## 7.4 Experiment Design / Analysis

The users from Group 'A' measured their blood pressure readings and populated their data using only the windows based / web based blood pressure software.

A typical user data set looks like the figure below. As discussed in the previous section we capture the date, systolic, diastolic and pulse readings, but these are optional to capture.

Date	Group	SYS	Dia
3/07/2006	A	166	90
5/07/2006	A	134	99
7/07/2006	A	144	103
10/07/2006	A	140	95
12/07/2006	A	140	94
14/07/2006	A	139	104
17/07/2006	A	131	92
19/07/2006	A	180	102
21/07/2006	A	161	90
24/07/2006	A	166	93
26/07/2006	A	147	95

**Table 7.1 – A typical user dataset**

To keep user's personal data confidential, the health specific data was stored in a separate table from the user data. The user data was stored separately, however the table was not encrypted to avoid complexity during the prototype phase.

The emphasis was to capture the blood pressure data using the software prototype, so the users can use the reporting feature and in turn the software can make predictions based on the previous readings.

The above result set (Table 7.1) contains the actual data recorded by a subject from the experiment carried out during this thesis (please refer to Appendix E for the complete data of a user).

The table below shows the statistics of ‘Group A’ users (the statistics have been computed on the same user data as in Appendix E)

<b>Descriptive Statistics: Group A</b>			
<b>SYS</b>		<b>DIA</b>	
Mean	150.0327302	Mean	95.59759049
Standard Error	1.112641675	Standard Error	0.567666393
Median	148.7915769	Median	94.9298009
Mode	#N/A	Mode	#N/A
Standard Deviation	13.76261745	Standard Deviation	7.021645501
Sample Variance	189.4096391	Sample Variance	49.30350555
Kurtosis	-0.803283987	Kurtosis	-0.92041418
Skew ness	0.468479779	Skew ness	0.39185059
Range	49.5727538	Range	24.73100161
Minimum	130.201571	Minimum	85.0216445
Maximum	179.7743248	Maximum	109.7526461
Sum	22955.00771	Sum	14626.43135
Count	153	Count	153
Confidence Level (95.0%)	2.198239364	Confidence Level (95.0%)	1.121535027

**Table 7.2 – Descriptive Statistics for Group A**

The table below shows the statistics of 'Group B' users.

**Descriptive Statistics: Group B**

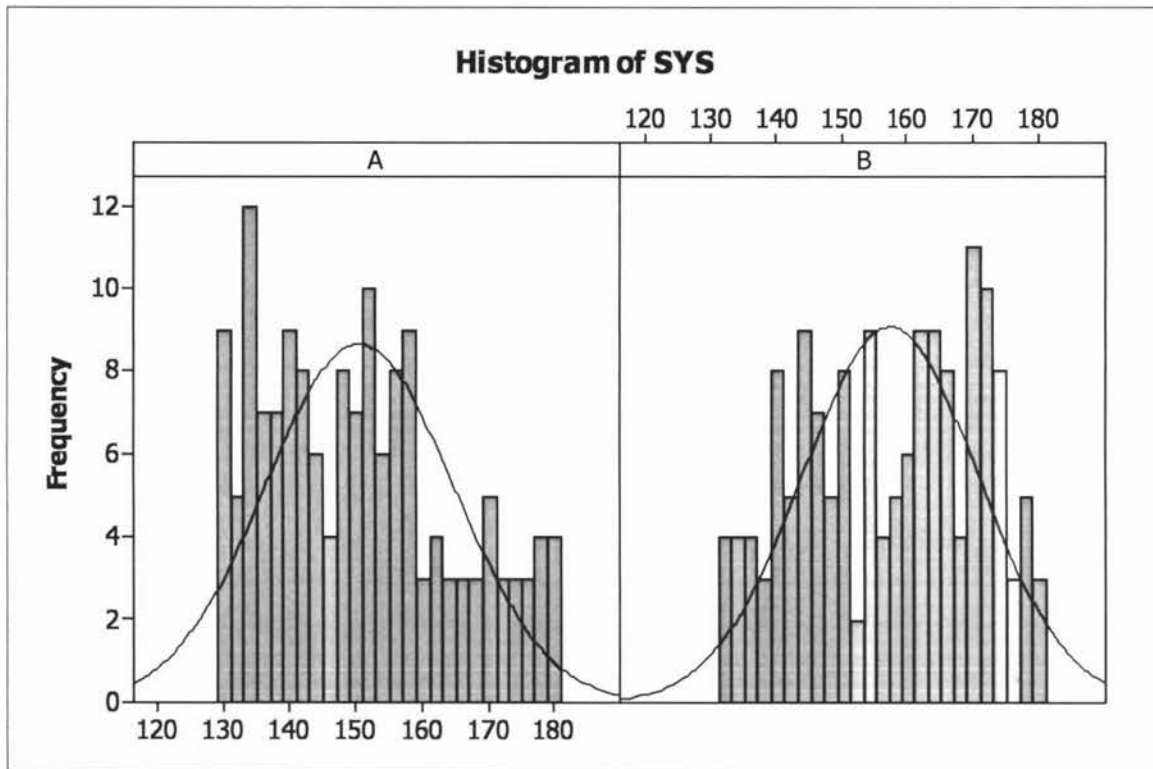
<b>SYS</b>		<b>DIA</b>	
Mean	156.517662	Mean	99.6184689
Standard Error	1.07245316	Standard Error	0.59398836
Median	156.665153	Median	100.833917
Mode	#N/A	Mode	#N/A
Standard Deviation	13.265513	Standard Deviation	7.34723027
Sample Variance	175.973835	Sample Variance	53.9817926
Kurtosis	-1.0980114	Kurtosis	-0.9604879
Skew ness	-0.0390178	Skew ness	-0.4289701
Range	48.5888	Range	24.4351964
Minimum	131.207378	Minimum	85.303219
Maximum	179.796178	Maximum	109.738415
Sum	23947.2023	Sum	15241.6257
Count	153	Count	153
Confidence Level (95.0%)	2.11883916	Confidence Level (95.0%)	1.17353918

**Table 7.3 – Descriptive Statistics for Group B**

In Table 7.2 and 7.3 above, the descriptive statistics have been computed using MS Excel for Group A and B. From the above tables, it can be seen that the means for Sys and Dia from Group A are lower than the means for Sys and Dia from Group B indicating that the average blood pressure readings for users using modern technology are lower than the users not using this technology. Although in terms of variance and standard deviation, there is more variance within Group A than within Group B i.e. the samples are further away from the mean within Group A than within Group B.

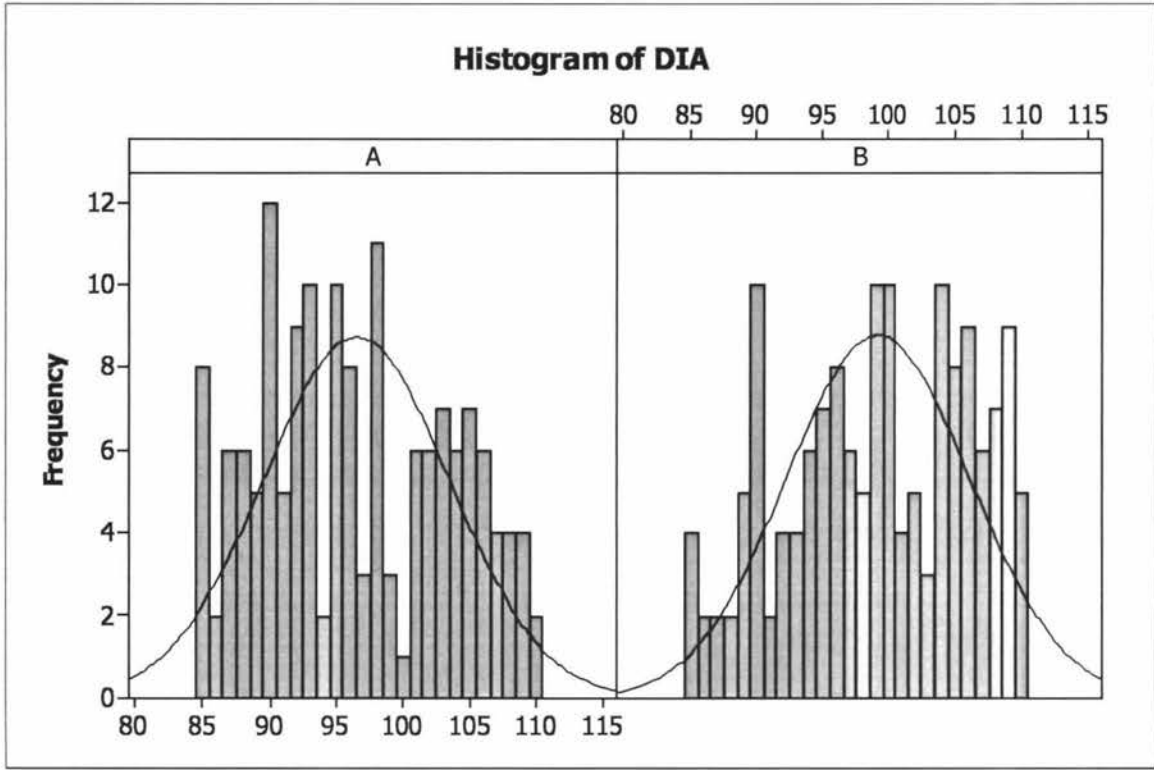
If we look at the histogram (Figure 7.1), we can see that both groups' blood pressure readings fall under the symmetric curve and follow a normal distribution.

This shows that the user data follows a normal distribution with the majority of the blood pressure readings falling under the 130 mmHg and 170 mmHg for Group A and 140 mmHg to 175 mmHg for Group B users. The frequency of users with higher Systolic readings lies with Group B users. As we can see from the histogram below, the majority of the readings of Group B is in the 150 mmHg



**Figure 7.1 – Distribution of the systolic data for Group A and Group B**

The histogram for the diastolic data also shows a normal distribution for both user groups, showing that the data recorded by the user was of the normal form, hence without many irregularities within the readings.



**Figure 7.2 – Distribution of Diastolic data for Group A and Group B**

## 7.5 Hypothesis Testing

Our Hypothesis states that “*high blood pressure is inversely related to modern technology, so that people with access to technology will be able to manage their blood pressure.*”

*If hypertension can be managed using modern technology, then people who have access to modern technology can manage their blood pressure successfully by having better control over it.”*

The normal range for blood pressure is between 130/85 (normal high) and 110/75 (normal low). To avoid confusion - since we cannot test out hypothesis on both values and taking an average would be to misinterpret the results - we will compare our hypothesis with the readings mentioned above.

At present, the following can be fairly stated:

*Null hypothesis  $H_0$ :* Use of modern technology does not allow people to manage blood pressure among who have high blood pressure

*Alternate hypothesis  $H_a$ :* Use of modern technology allows people to manage their blood pressure among who have high blood pressure

**Figure 7.3 – Mathematical Hypothesis Statement [40]**

With the significance or risk level assumed at 5% ( $\alpha = 0.05$ ) and degree of freedom for the between groups ( $df1$ ) = 2 - 1 = 1 and the degree of freedom for the within groups = (153-1) + (153-1) = 304, we derive the critical value of  $F_{crit}$  as 3.87. Thus if  $F_{obs} \geq F_{crit}$ ,  $H_0$  is rejected, otherwise do not reject  $H_0$ .

In order to conclude the findings (as seen in Table 7.2 and Table 7.3) for a population, multivariate MANOVA and f test have been used to explain the variance in the dependent

variables, i.e. Sys and Dia, amongst the two groups A and B created by the independent categorical variable, i.e. Use of Modern Technology. This technique has been used since it examines the groups formed by the categories of the independent variable according to their pattern of dispersion, by measuring the group variances in order to conclude if an independent variable has an effect on the dependent variables.

The p-value is the probability that indicates whether the variation between conditions may have occurred by chance. Minor p-values indicate that the variation is less prone to have occurred by chance and thus more probable to be related to the difference in conditions.

Source	Dependent Variable	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Observed Power(a)
Corrected Model	SYS	5510.141	1	5510.141	30.561	.000	.091	1.000
	Dia	1012.966	1	1012.966	22.210	.000	.068	.997
Intercept	SYS	7159164.150	1	7159164.150	39707.617	.000	.992	1.000
	Dia	2892340.565	1	2892340.565	63416.952	.000	.995	1.000
Group	SYS	5510.141	1	5510.141	30.561	.000	.091	1.000
	Dia	1012.966	1	1012.966	22.210	.000	.068	.997
Error	SYS	54810.287	304	180.297				
	Dia	13864.929	304	45.608				
Total	SYS	7219484.578	306					
	Dia	2907218.460	306					
Corrected Total	SYS	60320.428	305					
	Dia	14877.895	305					

**Table 7.4 – MANOVA Test of Between Subjects**

In the above Table 7.4 the inferential statistics have been computed for MANOVA using SPSS. The F values for the independent variable labelled GROUP (i.e. Use of modern technology) are unlike and with the significance level less than 1% (shown by Sig being .000), it clearly indicates that, on the dependent variables, there is a significant difference between the two groups of users - i.e. the ones using modern technology and the ones not using it.

The above table displaying the Tests of Between-Subjects Effect provides us with essential information such as the Mean Square value, the F values and the significance levels for each dependent variable, while the corrected model under the source column is the variance in the dependent variables that the independent variables accounts for.

The important section of the table is where the source under consideration is the independent variable for the row called GROUP. In this row it can be seen that both the dependant variables are significant ( $p < .05$ ), meaning that the independent variables differ significantly in SYS and DIA.

Dependent Variable	Parameter	B	Std. Error	t	Sig.	95% Confidence Interval		Partial Eta Squared	Noncent. Parameter	Observed Power(a)
						Lower Bound	Upper Bound			
SYS	Intercept	157.201	1.086	144.813	.000	155.065	159.337	.986	144.813	1.000
	[Group=A]	-8.487	1.535	-5.528	.000	-11.508	-5.466	.091	5.528	1.000
	[Group=B]	0(b)	.	.	.	.	.	.	.	.
Dia	Intercept	99.041	.546	181.401	.000	97.967	100.116	.991	181.401	1.000
	[Group=A]	-3.639	.772	-4.713	.000	-5.158	-2.119	.068	4.713	.997
	[Group=B]	0(b)	.	.	.	.	.	.	.	.

**Table 7.5 - Parameter Estimates**

a: Computed using alpha = .05

b: This parameter is set to zero because it is redundant.

The above table is split into rows for each of the dependent variables while within each row are the two groups, one for users of modern technology and the other for non-users. For the SYS and DIA dependent variable, both the groups are significantly different ( $p = 0.000$ ).

For the null hypothesis to be true, the F-ratio should be closer to 1 but as we can see in Table 7.4 above that the F-ratio ( $F_{obs}$ ) for SYS is 30.56 and F-ratio ( $F_{obs}$ ) for DIA is 22.21 which are definitely above 1, substantiating a significant difference between groups A and B.

Since the  $F_{obs}$  for SYS and DIA are 30.56 and 22.21 respectively, both are greater than the  $F_{crit}$  of 3.87, thus successfully completing the F test. Plus  $p = 0.000$  for both the dependant variables, which is less than the assumed alpha or risk level of 5% ( $\alpha = 0.05$ ) clearly

indicating that the probability of this variance occurring by chance is less than 1% ( $p=0.000$ ). So the alternative hypothesis ( $H_1$ ) is accepted, rejecting the null hypothesis ( $H_0$ ) certifying that the high blood pressure patients using modern technology can manage their blood pressure while blood pressure cannot be managed to the same extent by high blood pressure patients not using this technology.

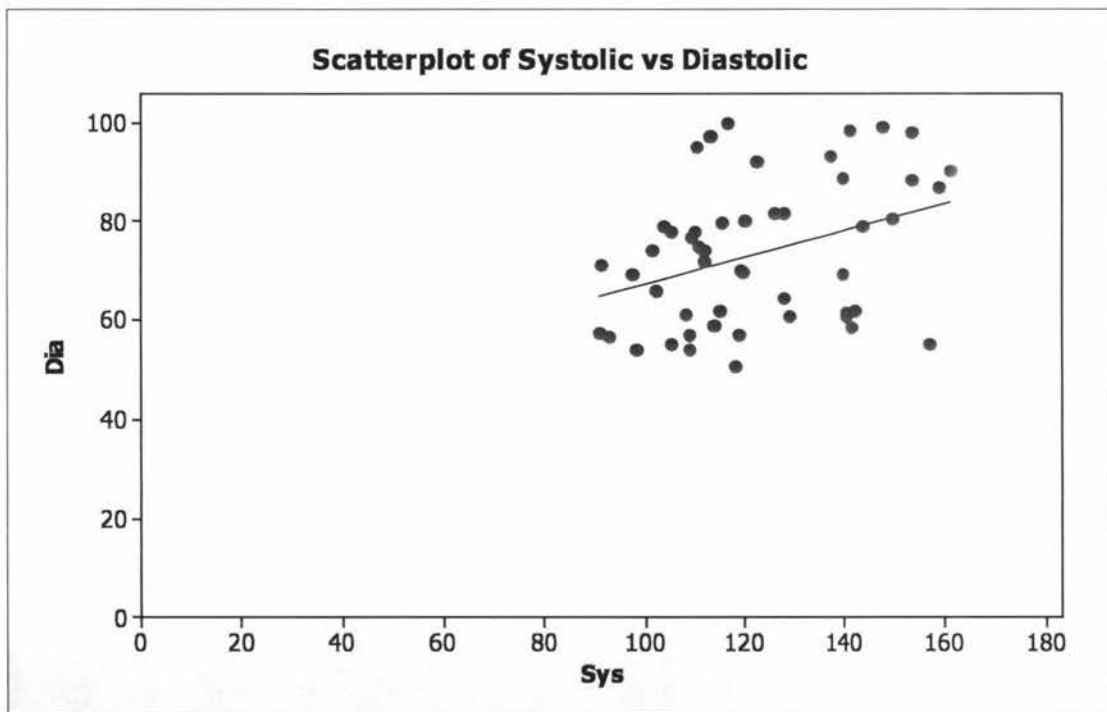
The alternative hypothesis states that “Use of modern technology allows people to manage thier blood pressure among who have high blood pressure” and according to the findings above with the p-value and the f-test, the alternative hypothesis is accepted with 95% confidence level, rejecting the null hypothesis. In conclusion, the author can say with 95% confidence that the use of modern technology can manage blood pressure among people with high blood pressure.

## Statistical model to predict future readings

### Correlation of Systolic and Diastolic readings

Let's look at the relationship between the Systolic and the diastolic readings of a user. Both Systolic and Diastolic are correlated; if one rises it affects the other reading.

If we take a look at the scatter plot below of the Systolic vs. Diastolic, the data plotted does not make much sense but if we pay attention to the regression line; the positive change in gradient is evident enough. Meaning that as the systolic reading increases so does the diastolic reading.



**Figure 7.4 – Scatter plot against Systolic vs. Diastolic**

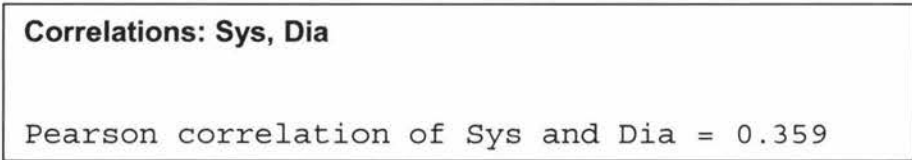
We will examine the case further by calculating the 'Pearson's correlation coefficient  $r$ '; given by:

$$r = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}} = \frac{s_{xy}}{s_x s_y}$$

**Figure 7.5 – Pearson's correlation coefficient [39]**

The Pearson correlation coefficient measures the strength and direction of a linear relationship between the  $X$  and  $Y$  variables.

The above calculation reveals an 'r' value of 0.359. 'r' being a positive value shows that there is a positive relationship between the Diastolic and Systolic readings of this particular user. However, it is not strong enough to conclude that there is a definite relation visible, all we can conclude at this point that there is 40% chance of this being true and that the Diastolic reading will be affected if the Systolic reading is changed.



**Figure 7.6 – Correlation calculation obtained from Minitab 15**

This is a critical observation which is useful. This observation will form the basis of our data prediction model. In theory we can conclude that using the regression line we can predict future readings of a user, provided there is a dataset for the algorithm to work on. As the amount of valid data increases in the system, the algorithm can make accurate predictions.

However, the regression equation would have to be calculated in real time so the updated 'r' value can be used in predicting future values of a user/patient. Hence, in order to do this we would have to implement the algorithm to calculate Pearson's correlation.

## Chapter 8 – Conclusion and Future work

### 8.1 Conclusion

As we can see from Chapter 7, the results of the MANOVA test prove that using modern technology can help manage blood pressure, so the alternate hypothesis  $H_a$  is valid – as use of modern technology does not allow people to manage blood pressure among who have high blood pressure. We are also 95% confident that the use of modern technology i.e. use of a windows based application or a web based application to populate a user/patient's data in a centralised environment gives the user awareness of their condition so he/she can better manage their blood pressure.

Therefore, capturing patient data in a centralised environment, gives the doctors access to up-to-date patient data, giving them more control and more information to better diagnose a user/patient's medical condition

The alternative hypothesis states that “Use of modern technology allows people to manage their blood pressure among who have high blood pressure” and according to the findings above with the p-value and the f-test, the alternative hypothesis is accepted with 95% confidence level, rejecting the null hypothesis.

In conclusion, it can be said with 95% confidence, that the use of modern technology allows people to manage blood pressure among who have high blood pressure.

## **8.2 “Rocky road” – Problems faced during this project**

Like any research project, there were many hurdles encountered during the project process. From the start of initial investigations to the very end where results were analysed, problems were part of this research. The most time consuming process was the data collection phase, as users had to be constantly reminded about measuring their blood pressure and more importantly, recording them.

A constant challenge was to learn new technology (ASP.net, Visual Basic.Net, Visual C#, Microsoft SQL 2005), implement it at the same time and also look for new ways to improve the architectural design to improve performance and reliability.

## **8.3 Future work / suggestions**

- The windows based application and the web based application can be integrated with the medical health system. This way the data transfer between the user’s personal computer and the Health Board medical system can be maintained, hence doctors would have access to updated user data. The data can be encrypted to maintain privacy issues between the client (user) and the server (medical health board system).
- Medical updates can be sent to the patient’s PC, so they can have up to date medical advice from a health professional
- The application can be ported to a PDA or mobile phone and the users can update their medical data by sending a simple text message from their mobile phone.
- The software (windows based application and web based application) can be extended to incorporate other medical conditions such as blood sugar levels.
- Security needs to be implemented to make data transfer secure (using a HTTPS protocol). Also the application (windows based) needs to have authorisation implemented, so only an administrator can add new profiles.

## **Appendices**

## Appendix A – The Heart

In order to better understand how we can manage high blood pressure, let's briefly look at a few concepts that will ease the process later.

### How does the heart work?

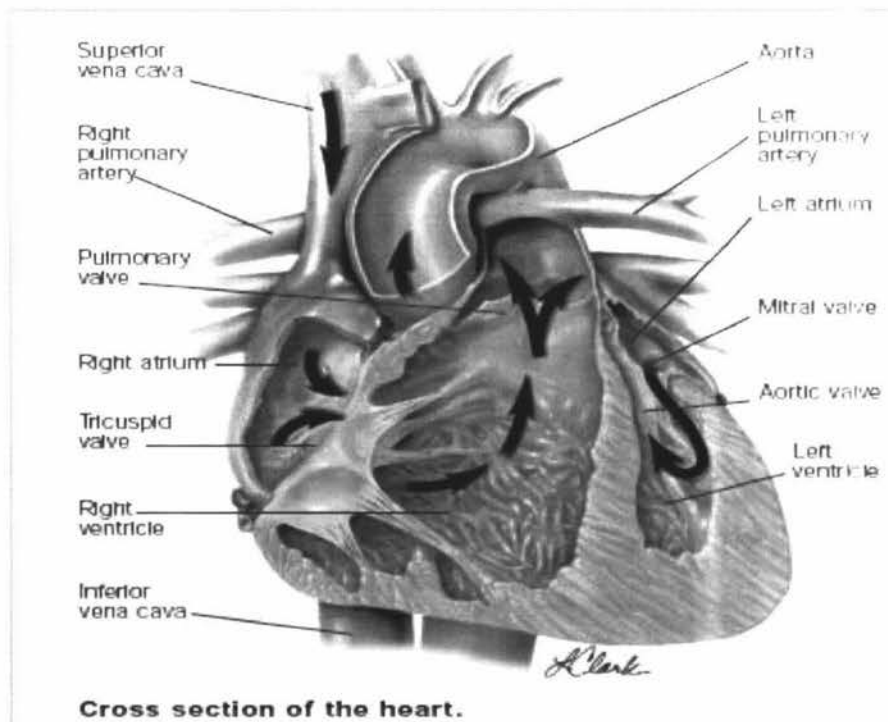
The normal human heart is a strong, muscular pump a little larger than a fist.

On an average day, a heart expands and contracts one hundred thousand times and pumps about 7,600 liters of blood. [3]

### What is a heart beat?

“A heartbeat consists of the contraction of the atria, quickly followed by the contraction of the ventricles. The rate of the heartbeat varies according to different levels of activity: the higher the level of activity or emotional excitement, the faster the heart will beat” [4]

The heart has four chambers through which blood is pumped (figure A.1). The upper chambers are the left/right atria and the lower left/right ventricles. The heart beats in a very highly organized fashion by contracting its four chambers.

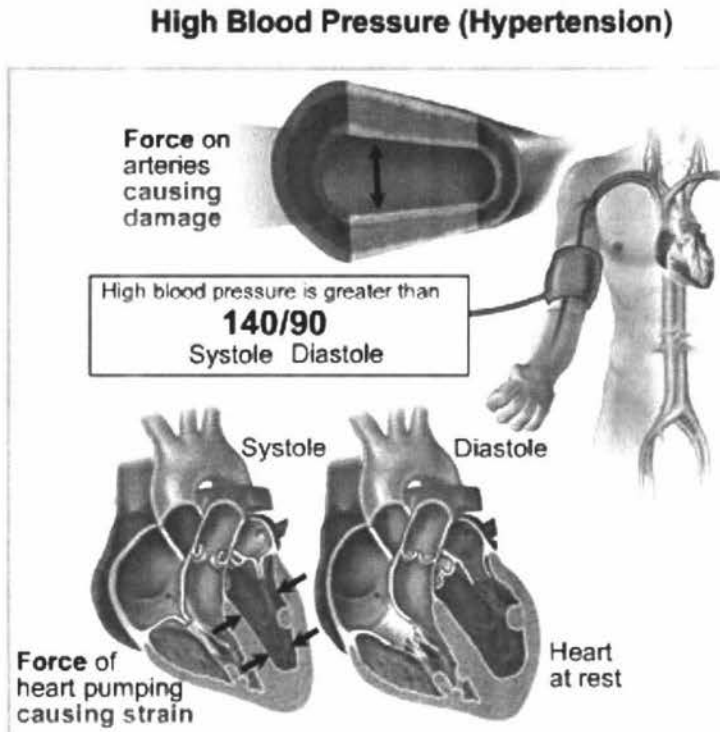


**Figure A.1 Cross section of the heart**

## What is Hypertension?

Another name for high blood pressure (HBP) is hypertension (hi-per-TEN-shun). [6]

Blood pressure is a force on the arteries (figure A.2), measured while the heart beats (systolic pressure) and rests (diastolic pressure). [7]



**Figure A.2 Hypertension (High blood pressure)**

### How is blood pressure measured?

Blood pressure can be measured in two ways either by visiting your local health care centre or at your convenience using an automatic digital machine.

### Risks associated with High blood pressure

There are numerous risks related with high blood pressure namely cardiovascular disease [10], stroke, Coronary Heart Disease (CHD), failure of the kidney, heart failure etc.

## **Why is high blood pressure a problem?**

High blood pressure or Hypertension is known as a silent killer, as there are no symptoms for detection of the condition.

Blood Pressure needs to be checked at regular intervals to detect any variations indicating hypertension.

It is not a disease; it's a condition of the heart where the heart does not pump blood in a normal range (Systolic: 120 or below, Diastolic: 79 or below) [11].

A diagnosis cannot be made after a single or few readings because blood pressure readings vary widely in some people. American heart organisation ® recommends at least 3 readings per day over a period of 3 or more days in order for the diagnosis of Hypertension to be accurate.

## Appendix B – Software Development Lifecycle / RAD

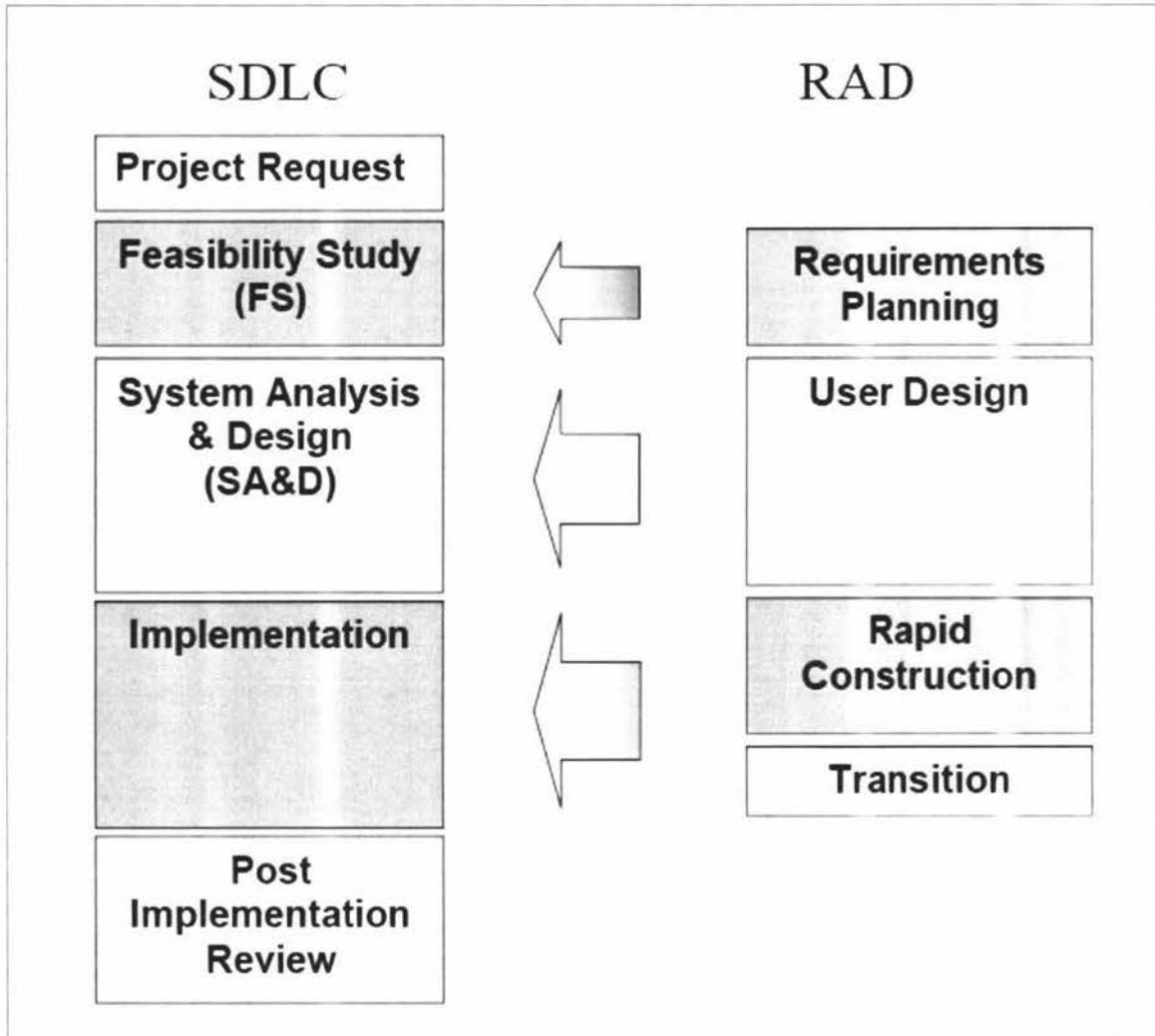


Figure B.1 – RAD mapped to Software Development Lifecycle

If a combined FS/SA and D are to be conducted, the Requirements Planning (RP) and User Design (UD) stages will be conducted consecutively. If a SA and D are to be conducted without a previously conducted FS, the activities in the RP stage will have to be done when necessary, before starting UD stage [41].

# Appendix C - BPM – RS232 Interface API

## Public Serial Interface Specification

### Table of Contents

1. Overview .....	104
1.1 Software Settings.....	104
1.2 Electrical Specifications .....	104
1.2.1 RS232C .....	104
1.2.2 Voltage Levels.....	104
2. Commands.....	105
2.1 Command Overview.....	105
2.2 Command Codes .....	106
2.3 Start Communications Task (20H).....	106
2.4 Stop Communications Task (21H) .....	106
2.5 Read Cycle Data from ABPM (22H).....	106
2.6 Write Cycle Data to ABPM (23H) .....	108
2.7 Read ID String and Version (24H).....	109
2.8 Read Parameters from ABPM (26H).....	109
2.9 Write Parameters to ABPM (27H) .....	111
2.10 Read Serial Number (3EH).....	113
3. Data Specifications.....	114
3.1 Data Types.....	114
3.2 Data Formats .....	114
3.3 Bit Order.....	114
3.4 Byte Order:.....	114
3.5 Interval Periods.....	114
3.6 Language Code.....	115
3.7 QTrak Time Periods .....	115
3.7.1 Default Vallues .....	115
3.7.2 Rules.....	115
3.8 Retry Codes .....	116
3.9 Initiation Codes .....	117
3.10 Error Codes.....	117

## Overview

The purpose of this document is to define the public serial interface protocol for Ambulatory Blood Pressure Monitor (ABPM).

## Software Settings

The communication setup parameters are preset in the ABPM and are not adjustable from outside. The settings are as follows.

Baud	9600
Rate	
Data	8
Bits	
Parity	none
Stop	1
Bits	

## Electrical Specifications

### RS232C

The ANSI RS232C three-wire serial communication protocol is used. The signals used are: transmit, receive and ground.

### Voltage Levels

The voltage of transmit and receive lines are +/- 9 Volts.

## Commands

### Command Overview

This section describes the commands and the command sequence to the ABPM. Every command consists of a four-byte sequence transmitted to the ABPM. The first three bytes are always the same, followed by a unique command code for a specific command.

A typical command sequence would be: 12H, 16H, 18H, XXH. The XXH is the unique command code for a specific command. All command sequences (unless otherwise noted in this spec.) are acknowledged by the ABPM with a 06H acknowledgment code. If an 06H acknowledgment code isn't received by the sender within three seconds then a communications error has occurred. If any other code is received it is in error and must be processed as an error. A typical reason for receiving a bad acknowledgment is an improper command sequence to the ABPM.

The ABPM may negatively acknowledge with either a NAK (15H) or a NAK2 (17H). A NAK response indicates that the command sent to the ABPM was incorrect. A NAK2 response indicates that the command was correct but was not completed successfully.

The communications protocol supports the use of Xon (11H) and Xoff (13H). If the ABPM receives a Xoff it will stop transmitting data within approximately ten bytes. When the ABPM receives a Xon it will resume data transmission.

The ABPM will transmit a Xoff when it wants the sender to stop sending data. When the input buffer only has room for 127 bytes the ABPM will send a Xoff. When the input buffer has room for 193 bytes, the ABPM is again ready to accept data and will send a Xon. A specific command may not support the Xon and Xoff protocol; if it does not, it will be noted in the description of that command.

The command sequence may at times get corrupted. Sending a string of synchronizing bytes (00H) will correct this problem. This allows the sender to flush out the ABPM's input

processor so that a new command sequence may be initiated. The input processor disregards the 00H bytes, allowing an unlimited number of them to be used for synchronizing.

## Command Codes

The following table lists the command codes and a brief description of each command.

<b>Command Table</b>	
Command Hex	Command Description
<b>20</b>	Start Communications Task
<b>21</b>	Stop Communications Task
<b>22</b>	Read Cycle Data from ABPM
<b>23</b>	Write Cycle Data to ABPM
<b>24</b>	Read ID String and Version
<b>26</b>	Read Parameters from ABPM
<b>27</b>	Write Parameters to ABPM
<b>3E</b>	Read Serial Number

### Start Communications Task (20H)

This command initiates a communications session in the ABPM. A communications session needs to be initiated before any other serial commands are sent. The command is acknowledged with an ACK.

### Stop Communications Task (21H)

This command ends the communications session in the ABPM, and should only be sent when all serial communications have been completed. The command is acknowledged with an ACK.

### Read Cycle Data from ABPM (22H)

This command requests the cycle data from the ABPM. A communications session must be in progress before sending this command. The command is acknowledged with an ACK followed by the ASCII cycle data.

The cycle records are sent only if the number of cycles is not zero. Only the number of cycles in memory will be sent. This allows for a variable number of cycles to be sent.

**Note: The oscillometric results should NOT be used for patient diagnosis. The algorithm used to generate these values has NOT been validated.**

**Data Format:**

Number of cycles Word,

Number of office runs Word, (this value corresponds to the number of the next office run)

12 bytes of patient ID, (Null Terminated ASCII String)

(ex. empty record is 11 space characters(hex20) and null (hex00))

Cycle record 1,

...

Cycle record 300,

Checksum Byte

**Cycle Record Format:**

Year Byte, (00 - 99 BCD)

Month Byte, (1 - 12 BCD)

Day Byte, (1 - 31 BCD)

Hour Byte, (0 - 23 BCD)

Minute Byte, (0 - 59 BCD)

Retry/Init Byte, (upper is retry code, lower is initiation code)

Error Code Byte, (error code if any ending the cycle)

Oscillometric Used Byte, (when 1 oscillometric was used, 0 auscultatory used)

Auscultatory results Long, (bits 0 - 9 systolic, 10 - 19 diastolic, 20 - 29 heart rate)

Oscillometric results Long, (bits 0 - 9 systolic, 10 - 19 diastolic, 20 - 29 heart rate)

Each nibble of each byte, word or long is sent as an ASCII character. The highest nibble of a byte, word or long is sent first followed by the lower nibbles. The checksum byte is the sum of the ASCII bytes sent.

## Write Cycle Data to ABPM (23H)

This command sends cycle data to the ABPM. . A communications session must be in progress before sending this command. The command is acknowledged with an ACK, then the ASCII cycle data is sent to the ABPM.

A variable number of cycles can be sent to the ABPM, the number of cycles word defines how many will be sent. The office runs word should be an accurate count of the number of office runs contained in the cycle records sent to the ABPM.

### Data Format:

Number of cycles Word,

Number of office runs Word, (this value corresponds to the number of the next office run)

12 bytes of patient ID, (Null Terminated String)

Cycle record 1,

...

Cycle record 300,

Checksum Byte

### Cycle Record Format:

Year Byte, (00 - 99 BCD)

Month Byte, (1 - 12 BCD)

Day Byte, (1 - 31 BCD)

Hour Byte, (0 - 23 BCD)

Minute Byte, (0 - 59 BCD)

Retry Init Byte, (upper nibble is retry code, lower nibble initiation code)

Error Code Byte, (error code if any ending the cycle)

Oscillometric Used Byte, (when 1 oscillometric was used, 0 auscultatory used)

Auscultatory results Long, (bits 0 - 9 systolic, 10 - 19 diastolic, 20 - 29 heart rate)

Oscillometric results Long, (bits 0 - 9 systolic, 10 - 19 diastolic, 20 - 29 heart rate)

Each nibble of each byte, word or long is sent as an ASCII character. The highest nibble of a byte, word or long is sent first followed by the lower nibbles. The checksum byte is the sum of the ASCII bytes sent.

**Note: If patient ID is less than 11 characters in length, it should be padded with space characters (hex20).**

## **Read ID String and Version (24H)**

This command is used to read the ID String and Version from the ABPM. A communications session must be in progress before sending this command. The command is acknowledged with an ACK, followed by the ASCII ID String and version data.

### **ASCII Data String Format:**

"TYCOS ABPM, XX.XX" character string, (only sends information between quotes)

Carriage Return Byte,

Line Feed Byte,

Checksum Byte

XX.XX is the current system version

The characters in the Data string above and the carriage return and line feed are sent as actual ASCII characters.

Each nibble of the checksum is sent as an ASCII character. The highest nibble of a byte, word or long is sent first followed by the lower nibbles. The checksum byte is the sum of the ASCII bytes sent.

## **Read Parameters from ABPM (26H)**

This command is used to read the current ABPM setup parameters. A communications session must be in progress before sending this command. The command is acknowledged with an ACK, followed by the ASCII data.

### **Data Format:**

month Byte, (BCD)

day Byte, (BCD)

year tens and ones Byte, (BCD)

year thousands and hundreds Byte, (BCD)  
hour Byte, (BCD)  
minute Byte, (BCD)  
second Byte, (BCD)  
tenths of seconds Byte, (BCD)  
ampm status Byte, (0 = no ampm, 1 = ampm)

maximum inflation pressure Word,  
deflation rate Byte,  
number of time periods Byte  
time period one starting hour Byte, (BCD)  
time period one interval period Byte,  
time period two starting hour Byte, (BCD)  
time period two interval period Byte,  
time period three starting hour Byte, (BCD)  
time period three interval period Byte,  
time period four starting hour Byte, (BCD)  
time period four interval period Byte,

maximum systolic pressure Byte,  
minimum systolic pressure Byte,  
maximum change in systolic pressure Byte,  
maximum diastolic pressure Byte,  
minimum diastolic pressure Byte,  
maximum change in diastolic pressure Byte,  
maximum pulse pressure Byte,  
minimum pulse pressure Byte,

start button enabled Byte, (0 - off, 1 - on)  
display on Byte, (0 - off, 1 - on)  
dynamic pressure Byte, (0 - off, 1 - on)  
auto dump Byte, (0 - off, 1 - on)  
use microphone Byte, (0 - no, 1 - yes)  
one day maximum Byte, (0 - off, 1 - on)

language Byte, (0 - 4)

version to the right of decimal Byte,  
version to the left of the decimal Byte

initial valve value Word  
initial sync. pressure Word

Checksum Byte

Each nibble of each byte, word or long is sent as an ASCII character. The highest nibble of a byte, word or long is sent first followed by the lower nibbles. The checksum byte is the sum of the ASCII bytes sent.

## **Write Parameters to ABPM (27H)**

This command sends the setup parameters to the ABPM. A communications session must be in progress before sending this command. The command is acknowledged with an ACK, then the ASCII setup parameter data is sent to the ABPM.

**Note:** When using this command it is necessary to first call the Read Parameters from ABPM command to obtain values for initial valve value” and initial sync. Pressure”. These values should be copied to the corresponding fields in the parameter record before writing new parameters to the unit. Failure to do so can prevent the ABPM from operating correctly.

### **Data Format:**

month Byte, (BCD)

day Byte, (BCD)

year tens and ones Byte, (BCD)

year thousands and hundreds Byte, (BCD)

hour Byte, (BCD)

minute Byte, (BCD)

second Byte, (BCD)

tenths of seconds Byte, (BCD)  
ampm status Byte, (0 = no ampm, 1 = ampm)

maximum inflation pressure Word,  
deflation rate Byte,  
number of time periods Byte  
time period one starting hour Byte, (BCD)  
time period one interval period Byte,  
time period two starting hour Byte, (BCD)  
time period two interval period Byte,  
time period three starting hour Byte, (BCD)  
time period three interval period Byte,  
time period four starting hour Byte, (BCD)  
time period four interval period Byte,

maximum systolic pressure Byte,  
minimum systolic pressure Byte,  
maximum change in systolic pressure Byte,  
maximum diastolic pressure Byte,  
minimum diastolic pressure Byte,  
maximum change in diastolic pressure Byte,  
maximum pulse pressure Byte,  
minimum pulse pressure Byte,

start button enabled Byte, (0 - off, 1 - on)  
display on Byte, (0 - off, 1 - on)  
dynamic pressure Byte, (0 - off, 1 - on)  
auto dump Byte, (0 - off, 1 - on)  
use microphone Byte, (0 - no, 1 - yes)  
one day maximum Byte, (0 - off, 1 - on)  
language Byte, (0 - 4)

initial valve value Word  
initial sync. pressure Word

## Checksum Byte

Each nibble of each byte, word or long is sent as an ASCII character. The highest nibble of a byte, word or long is sent first followed by the lower nibbles. The checksum byte is the sum of the ASCII bytes sent.

Once the correct check sum is received the ABPM returns either an ACK or a NAK to indicate if the data has been successfully received.

## **Read Serial Number (3EH)**

This command reads the serial number from the ABPM. A communications session must be in progress before sending this command. The command is acknowledged with an ACK. Then the eleven bytes of the serial number are sent from the ABPM followed by the checksum.

## Data Specifications

This section details information that is not captured in other areas of this document. In particular the interpretation of codes sent to, and returned from the ABPM.

## Data Types

Byte	unsigned 8-bit
Word	unsigned 16-bit
Long	unsigned 32-bit

## Data Formats

- BCD (Binary Coded Decimal)
- Hexadecimal
- Packed bit fields
- Null terminated ASCII String ("C" string format)

## Bit Order

Bit order is most significant bit first (left most bit), least significant bit last (right most bit).

## Byte Order:

All Words and Longs are in big ending format (most significant Byte first, least significant Byte last).

## Interval Periods

Value	Meaning
0	5 min.
1	10 min.
2	15 min.
3	20 min.
4	30 min.
5	60 min.
6	120 min.

7	Manual
>7	Reserved - Error

## Language Code

Value	Meaning
0	Spanish
1	German
2	French
3	Italian
4	English
>4	Reserved - Error

## QTrak Time Periods

### Default Values

Number of Periods	Start Time	End Time	Interval
1	7 am	7 am	15 min
2	7 am 11 pm	11 pm 7 am	15 min 30 min
3	7 am 11 pm 12 am	11 pm 12 am 7 am	15 min 30 min 60 min
4	7 am 11 pm 12 am 5 am	11 pm 12 am 5 am 7 am	15 min 30 min 60 min 20 min

### Rules

The following four configurations are the only valid options for start time settings (ST).

Number of Periods	Start Time Sequence
1	$ST_1 \Rightarrow ST_1$
2	$ST_1 \Rightarrow ST_2 \Rightarrow ST_1$
3	$ST_1 \Rightarrow ST_2 \Rightarrow ST_3 \Rightarrow ST_1$
4	$ST_1 \Rightarrow ST_2 \Rightarrow ST_3 \Rightarrow ST_4 \Rightarrow ST_1$

- No two-start times can be the same.
- Total duration of all periods must equal 24 hours

## Retry Codes

Value	Meaning
0	No Retry.
1	Invalid
2	Invalid
3	The reading was rejected because it violated the Maximum Allowable Systolic Pressure set in the retry parameters.
4	The reading was rejected because it violated the Minimum Allowable Systolic Pressure set in the retry parameters.
5	The reading was rejected because it violated the Maximum Allowable Change in Systolic Pressure set in the retry parameters.
6	The reading was rejected because it violated the Maximum Allowable Diastolic Pressure set in the retry parameters.
7	The reading was rejected because it violated the Minimum Allowable Diastolic Pressure set in the retry parameters.
8	The reading was rejected because it violated the Maximum Allowable Change in Diastolic Pressure set in the retry parameters.
9	The reading was rejected because it violated the Maximum Allowable Pulse Pressure set in the retry parameters.
10	The reading was rejected because it violated the Minimum Allowable Pulse Pressure set in the retry parameters.
>10	Reserved - Error.

## Initiation Codes

Value	Meaning
0	The cycle was initiated by the unit as a timed cycle.
1	The cycle was initiated by the patient as a Manual cycle.
2	The cycle was initiated by the unit as a Retry cycle.
3	The cycle was initiated by the user as an Office Cycle.
>3	Reserved - Error.

## Error Codes

Value	Meaning
0	No Error.
1	The cycle was aborted by the user by pressing a button on the ABPM.
2	The unit took too long to inflate the cuff. The cuff may have been disconnected or too loose.
3	The ABPM experienced a power failure during the cycle.
4	The ABPM was unable to perform the Autozero function prior to initiating the cycle.
5	The ABPM reached the target pressure too quickly. This usually results when the hose has been pinched off.
6	The unit sensed the cuff pressure exceeding 300mmHg.
7	The unit sensed the pressure in the cuff was above 10mmHg for longer than 180 seconds.
8	The ABPM was unable to determine the Blood Pressure due too much noise or movement by the patient.
9	The unit has maximized its gain settings for the microphone and is still unable to detect Korotkoff sounds.
10	The unit is unable to detect Korotkoff sounds and will adjust the microphone settings for the next reading.
11	The unit has detected Korotkoff sounds that are too loud and will adjust the microphone settings for the next reading.

12	The Blood Pressure determined is outside of the detectable range of the ABPM.
13	The systolic pressure was detected too close to the target pressure.
>13	Reserved - Error.

\*\*\* End of Document \*\*\*

## Appendix D – Input validation algorithm

The code below represents the algorithm that performs the data input validation, checks whether the user has entered numeric and valid data based on the threshold levels.

```
#Region "Update Patient Data Save button clicked"
```

```
Private Sub NewUpdatePatientData_Save_button_clicked() Handles  
NewUpdatePatientData.Save_button_clicked  
  
- = If validate_empty_text = True Then  
  
    Try  
        validate_text()  
  
    Catch e As Exception  
        MessageBox.Show("Please enter valid Data,before saving", "Empty text boxes",  
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)  
    End Try  
  
    ElseIf is_it_textbool = True Then  
        Try  
            is_it_text()  
  
        Catch e As Exception  
            MessageBox.Show("Please enter valid Data,before saving", "Empty text boxes",  
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)  
        End Try  
  
    Else : validate_error_text = True  
        Try  
  
            validate_text_error()  
  
        Catch e As Exception  
            MessageBox.Show("Please enter valid Data,before saving", "Empty text boxes",  
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)  
        End Try
```

End If

Dim newRow As DataRow = DataSet21.Patient\_Data.NewRow

txtSys = NewUpdatePatientData.systxt

txtDia = NewUpdatePatientData.diatxt

txtPulse = NewUpdatePatientData.pulsetxt

txtName = NewUpdatePatientData.nametxt

'Function to return the patient ID value associated with the name selected by the user.

sql\_return()

'These are boolean variables declared earlier to check the status of return values

If validate\_empty\_text And validate\_error\_text And is\_it\_textbool Then

Try

newRow("Sys") = txtSys

newRow("Pulse") = txtPulse

newRow("Dia") = txtDia

newRow("Date") = System.DateTime.Today

' Trying to get the patient id

newRow("PatientID") = 2

DataSet21.Patient\_Data.Rows.Add(newRow)

dbData.Update(DataSet21, "Patient Data")

MessageBox.Show("Add was successful", "Add successful", MessageBoxButtons.OK,

MessageBoxIcon.Information)

mblnSave = True

Catch ex As Exception

MessageBox.Show("Error saving data", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

End Try

' Sets the text boxes to ""

If mblnSave Then

NewUpdatePatientData.txtSys.Text = ""

NewUpdatePatientData.txtDia.Text = ""

NewUpdatePatientData.txtPulse.Text = ""

End If

Else

' MsgBox("Error in validation", MsgBoxStyle.OKOnly)

End If

End Sub

Private Sub sql\_return()

Dim scalarcmd As String

Try

'scalarcmd = "SELECT PatientID FROM PatientInfo WHERE (Given Name = ?)"

'scalarcmd = "Select PatientID FROM PatientInfo WHERE ([Given Name] = ?)"

scalarcmd = "select distinct p.PatientID, i.[Given Name] from [Patient Data] p left outer join PatientInfo  
i on p.PatientID = i.PatientID where p.PatientID =1 order by p.PatientID"

sqlCommand1.CommandType = CommandType.Text

sqlCommand1.CommandText = scalarcmd

'sqlCommand1.Parameters("Given Name").Value = txtName

test = CType(sqlCommand1.ExecuteScalar(), Integer)

Catch e As Exception

bInsql\_return = False

MsgBox(test)

End Try

End Sub

' This procedure validates empty strings

Private Sub validate\_text()

If NewUpdatePatientData.txtSys.Text = "" Then

    MessageBox.Show("Please enter valid Data in the SYS text field,before saving", "Empty text boxes",  
    MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

ElseIf NewUpdatePatientData.txtDia.Text = "" Then

    MessageBox.Show("Please enter valid Data in the DIA text field,before saving", "Empty text boxes",  
    MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

ElseIf NewUpdatePatientData.txtPulse.Text = "" Then

```
    MsgBox.Show("Please enter valid Data in the Pulse text box field,before saving", "Empty text boxes", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
```

```
Else
```

```
    validate_empty_text = True
```

```
End If
```

```
End Sub
```

```
Private Sub is_it_text()
```

```
    If IsNumeric(NewUpdatePatientData.txtSys.Text) = False Then
```

```
        MsgBox.Show("Please enter a numric value in the SYS text field", "Numric value", MessageBoxButtons.OK, MessageBoxIcon.Error)
```

```
    ElseIf IsNumeric(NewUpdatePatientData.txtDia.Text) = False Then
```

```
        MsgBox.Show("Please enter a numric value in the DIA text field", "Numric value", MessageBoxButtons.OK, MessageBoxIcon.Error)
```

```
    ElseIf IsNumeric(NewUpdatePatientData.txtPulse.Text) = False Then
```

```
        MsgBox.Show("Please enter a numric value in the Pulse text field", "Numric value", MessageBoxButtons.OK, MessageBoxIcon.Error)
```

```
Else
```

```
    is_it_textbool = True
```

```
End If
```

```
End Sub
```

'This procedure will validate if the data entered is accurate

'boolean varibale used here to check the status of the procedure

```
Private Sub validate_text_error()
```

```
    If NewUpdatePatientData.txtSys.Text < 100 Or NewUpdatePatientData.txtSys.Text > 140 Then
```

```
        MsgBox.Show("Please check the SYS value", "Check SYS Value", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
```

```
ElseIf NewUpdatePatientData.txtDia.Text < 60 Or NewUpdatePatientData.txtDia.Text > 90 Then
    MessageBox.Show("Please check the DIA value", "Check DIA Value", MessageBoxButtons.OK,
    MessageBoxIcon.Exclamation)
```

```
ElseIf NewUpdatePatientData.txtPulse.Text < 60 Or NewUpdatePatientData.txtPulse.Text > 100 Then
    MessageBox.Show("Please check the Pulse value", "Check Pulse Value", MessageBoxButtons.OK,
    MessageBoxIcon.Exclamation)
```

```
Else
    validate_error_text = True
End If
```

```
End Sub
```

```
Private Sub NewUpdatePatientData_Closing(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles NewUpdatePatientData.Closing
```

```
    If mblnIsDirty Then
```

```
        If MessageBox.Show("Do you want to save changes?", "Patient Data", MessageBoxButtons.YesNo,
    MessageBoxIcon.Question) = DialogResult.Yes Then
```

```
            If DialogResult.Yes Then
```

```
                Try
```

```
                    dbData.Update(DataSet21, "Patient Data")
```

```
                Catch
```

```
                    MessageBox.Show("Error Saving the data:", "Patient Data")
```

```
                End Try
```

```
            ElseIf DialogResult.No Then
```

```
                MsgBox("error in data entered")
```

```
            End If
```

```
        End If
```

```
    End If
```

```
    Try
```

```
        If Me.txtDia <> "" Or Me.txtSys <> "" Or Me.txtPulse <> "" Then
```

```
            MessageBox.Show("Would you like not to save the data entered", "Save?",
    MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation)
```

```
        ElseIf DialogResult = DialogResult.No Then
```

```
            NewUpdatePatientData.Show()
```

```
            Me.Show()
```

```
ElseIf DialogResult = DialogResult.Yes Then  
    NewUpdatePatientData_Save_button_clicked()
```

```
'Else dialogresult = DialogResult.No Then  
    Me.Close()
```

```
End If
```

```
Catch ex As Exception
```

```
    MsgBox("cannot save data")
```

```
End Try
```

```
End Sub
```

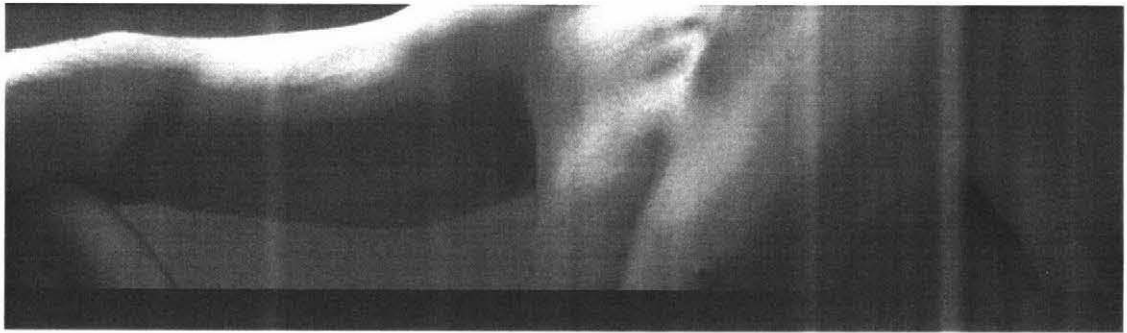
```
#End Region
```

## Appendix E – Patient Data Set

PatientID	Date	Sys	Dia	Pulse
1	03/07/06	112	74	80
1	05/07/06	109	57	75
1	07/07/06	112	72	94
1	10/07/06	115	62	85
1	12/07/06	161	90	85
1	14/07/06	137	93	86
1	17/07/06	114	59	83
1	19/07/06	109	57	88
1	21/07/06	110	78	72
1	24/07/06	105	78	78
1	26/07/06	120	80	65
1	28/07/06	111	75	76
1	31/07/06	119	57	68
1	02/08/06	102	66	67
1	04/08/06	109	54	66
1	07/08/06	150	64	99
1	09/08/06	126	62	79
1	11/08/06	132	82	90
1	14/08/06	108	58	84
1	16/08/06	110	84	90
1	18/08/06	116	79	73
1	21/08/06	135	57	56
1	23/08/06	110	65	67
1	25/08/06	120	88	84
1	28/08/06	152	66	56
1	30/08/06	131	59	95
1	01/09/06	135	85	97
1	04/09/06	138	80	56
1	06/09/06	107	68	66
1	08/09/06	119	83	82
1	11/09/06	113	83	97
1	13/09/06	133	66	57
1	15/09/06	127	76	75
1	18/09/06	137	71	91

1	20/09/06	125	84	72
1	22/09/06	126	85	84
1	25/09/06	132	57	87
1	27/09/06	97	94	77
1	29/09/06	127	60	62
1	02/10/06	114	89	69
1	04/10/06	140	74	94
1	06/10/06	107	87	55
1	09/10/06	103	96	72
1	11/10/06	157	63	94
1	13/10/06	156	89	57
1	16/10/06	96	78	68
1	18/10/06	155	85	80
1	20/10/06	116	51	85
1	23/10/06	136	52	50
1	25/10/06	93	78	84
1	30/10/06	132	62	68

## Appendix F – Microlife Blood Pressure Monitor Data Sheet



### Automatic Blood Pressure Monitor with MAM and PAD Technology

**Monitors** warning signals. Highly **reliable!**

#### Europe / Middle-East / Africa

Microlife AG  
Max Schmidheiny-Strasse 201  
9435 Heerbrugg / Switzerland  
Tel. +41 / 71 727 70 30  
Fax +41 / 71 727 70 39  
Email [admin@microlife.ch](mailto:admin@microlife.ch)  
[www.microlife.com](http://www.microlife.com)

#### Asia

Microlife Corporation  
9F, 431, RuiGang Road, NeiHu  
Taipei, 114, Taiwan, R.O.C.  
Tel. 886 2 8797-1288  
Fax 886 2 8797-1283  
Email [service@microlife.com.tw](mailto:service@microlife.com.tw)  
[www.microlife.com](http://www.microlife.com)

#### North / Central / South America

Microlife USA, Inc.  
424 Skinner Blvd., Suite C  
Dunedin, FL 34698 / USA  
Tel. +1 727 451 0484  
Fax +1 727 451 0482  
Email [custserv@microlifeusa.com](mailto:custserv@microlifeusa.com)  
[www.microlife.com](http://www.microlife.com)

**BHS A/A  
grading**

British Hypertension Society



BP 3AC1-1 PC

**microlife**



**A simple, daily blood pressure check – the intelligent device with MAM and PAD.**

High blood pressure is associated with an increased risk of diseases of the cardio-vascular system. Once identified, blood pressure values can be positively influenced in a variety of ways. The correct method is determined in consultation with the doctor.

This blood pressure monitor allows to check the therapy simply and reliably, with either individual measurements or a triple measurement using MAM technology.

The device also detects the presence of heart arrhythmias during a blood pressure measure-

ment\*. Early recognition of arrhythmias is very important as it helps alert us to possible heart diseases and allows the initiation of treatment. Moreover, this device can be used in connection with a personal computer running the Microlife blood pressure analyzer software. The stored BP reading data can be easily transferred to a PC via the provided USB-cable.

The software is easy to use and gives to patients and doctors an illustrative overview of the blood pressure profile.

Microlife blood-pressure measuring device BP 3AC1-1 PC. **Intelligent device with MAM and PAD – the complete solution.**

- Blood pressure and pulse measurement
- MAM – Microlife Average Mode or standard mode selection
- PAD – Pulse Arrhythmia Detection
- 99 Data Memory
- Jumbo display
- Easy operation
- Fully automatic inflation and deflation
- Oscillometric method
- Blood pressure analyzer software included
- PC connection cable included

MAM  
TECHNOLOGY  
ARRHYTHMIA  
DETECTION  
99 DATA MEMORY  
EASY OPERATION

**Microlife BP 3AC1-1 PC  
Automatic Blood Pressure Monitor**

Technical specifications:

Weight:	508 g (with batteries)
Size:	118 (W) x 176 (L) x 79 (H) mm
Storage temperature:	-20° to +50° C
Humidity:	15 to 90% relative humidity maximum
Operation temperature:	+10° to +40°C
Display:	Liquid Crystal Display (LCD)
Measuring method:	oscillometric
Pressure sensor:	capacitive
Measuring range:	
SYS/DIA:	30 – 280 mmHg
Pulse:	40 to 200 beats / minute
Cuff pressure display range:	0 – 299 mmHg
Measuring resolution:	1 mmHg
Accuracy:	
Pressure:	within ±3 mmHg
Pulse:	within ±5% of reading
Memory:	Storing 99 measurements
Power source:	a) 4 x 1.5V batteries; LR6, size AA b) mains adapter 6V DC 800mA (optional)
Cuffs:	M-size-cuff (22 – 32 cm) or L-size-cuff (32 – 42 cm) – optional
System requirements:	Windows 98 SE/ME/2000/XP, CD-Rom drive, 400 Mhz CPU or higher, 100 MB free hard disk space, 128 MB memory, one free USB port
Reference to standards:	EN 1060-1/-3/-4, NIBP – requirements ANSI/AAMI SP10, NIBP – requirements

\* FORSTNER, K. W. - Pulse Arrhythmia Detection by Oscillometric Blood Pressure Measurement Devices, P. 25, American Society of Hypertension, 18th Annual Scientific Meeting, NYC, New York, 2003



Technical alterations reserved!

**microlife®**

118BP3AC1-1 PC 420M

## Appendix G – Blood pressure data sheet

PatientID	Date	Sys	Dia	Pulse
1	03/07/06	166	96	80
1	05/07/06	157	86	75
1	07/07/06	158	89	94

## References

- [1] Moon, M. A. (3/1/2005). *Global trends in hypertension*. (CLINICAL CAPSULES) (Brief Article). Retrieved May 02, 2006, from [http://www.highbeam.com/library/doc0.asp?fr=1&full=yes&docid=1G1:131780484&refid=ls\\_pub&skeyword=&teaser=&origurl=http://www.highbeam.com/library/doc3.asp?docid=1G1:131780484&refid=ls\\_pub&skeyword=&teaser](http://www.highbeam.com/library/doc0.asp?fr=1&full=yes&docid=1G1:131780484&refid=ls_pub&skeyword=&teaser=&origurl=http://www.highbeam.com/library/doc3.asp?docid=1G1:131780484&refid=ls_pub&skeyword=&teaser)
- [2] High Blood Pressure Pledge. (May 2004, April 2005). Retrieved May 02, 2006, from <http://www.americanheart.org/downloadable/heart/1083166078286pledge%20revised%20042804.pdf>
- [3] *Heart and Stroke Facts*. Retrieved May 03 2006, 2006, from <http://www.americanheart.org/presenter.jhtml?identifier=3000333>
- [4] *Heartbeat*. Retrieved April 17, 2006, from [www.thoratec.com/thoratec-patients/glossary.htm](http://www.thoratec.com/thoratec-patients/glossary.htm)
- [5] *How does the heart work?* Retrieved April 17, 2006, from <http://www.capitalcardiology.com/heartwork.htm>
- [6] *What is high blood pressure?* Retrieved April 17, 2006, from <http://www.americanheart.org/downloadable/heart/110141108543012%20WhatisHBP.pdf>
- [7] *What is high blood pressure?* Retrieved April 17, 2006, from <http://www.americanheart.org/presenter.jhtml?identifier=2112>
- [8] Unknown. *High Blood Pressure Statistics*. Retrieved April 18, 2006, from <http://www.americanheart.org/presenter.jhtml?identifier=4621>
- [9] Stamler, J., Stamler, R., & Neaton, J. D. (1993). *Blood pressure, systolic and diastolic, and cardiovascular risks. US population data* 10.1001/archinte.153.5.598. *Arch Intern Med*, 153(5), 598-615.
- [10] Ramachandran S. Vasan, M. D., Martin G. Larson, Sc.D., Eric P. Leip, M.S., Jane C. Evans, Ph.D., Christopher J. O'Donnell, M.D., M.P.H., William B. Kannel, M.D., M.P.H., and Daniel Levy, M.D. (2001). Impact of High-Normal Blood Pressure on the Risk of Cardiovascular Disease. *The New England Journal of Medicine*, 345:1291-1297(No 18).
- [11] *Blood Pressure Readings. U.S. Pharmacist*. (2000) Retrieved May 02, 2006

- [12] *Blood Pressure Monitors, Information and Resources*. Retrieved May 06, 2006, from <http://www.bloodpressuremonitorsadvice.com/info/index396.html>
- [13] *Omron Automatic Blood Pressure Monitor with Intellisense HEM-711AC Review*. Retrieved May 06, 2006, from <http://products.consumerguide.com/reviews/product.epub?productId=22440>
- [14] *High blood pressure set to afflict one third by 2025*. (2005, January 14, 2005). Retrieved May 06, 2006, from <http://news.scotsman.com/international.cfm?id=49342005>
- [15] Appel, L. J. and W. B. Stason (1993). *Ambulatory Blood Pressure Monitoring and Blood Pressure Self-Measurement in the Diagnosis and Management of Hypertension*. 118: 867-882. <http://www.annals.org/cgi/content/abstract/118/11/867>
- [16] Tayab, D. m., Chowdhry, B., & Mohammad, S. M. General block diagram. *The Potential of Telemedicine System: An Approach Towards a Mobile Doctor*.
- [17] Baisa, N. (2005, April). Wireless technology comparison - Designing wireless interfaces for patient monitoring equipment. *Medical Electronics*.
- [18] "Hospitals Bring Medical Devices into the I.T. Loop: Innovative providers are linking various medical devices to clinical information systems.(information technology)." *Health Data Management*12.2 (Feb 2004): 46. *InfoTrac OneFile*. Thomson Gale. Massey University Library. 29 April 2006. <http://find.galegroup.com/itx/infomark.do?&contentSet=IAC-Documents&type=retrieve&tabID=T002&prodId=ITOF&docId=A112945216&source=gale&srcprod=ITOF&userGroupName=massey&version=1.0>
- [19] Schuerenberg, K. B. (2006, April 26). (B. K. Schuerenberg, Ed.) Retrieved July 20, 2006, from Health data management: <http://www.healthdatamanagement.com/html/current/CurrentIssueStory.cfm?articleId=12881>
- [20] Al-Busaidi, A., Gray, A., & Fiddian, N. (2006). Personalizing web information for patients: linking patient medical data with the web via patient personal knowledge base. *Health Informatics Journal* , 12 (1), 27-39.
- [21] *Mediator and Ontology based integration of ISCO patient database with the web*. UK, SAGE Publications (London, Thousand Oaks, CA and New Delhi).
- [22] (2005). "The doctor in your pocket." *Economist*; 9/17/2005, Vol. 376 Issue 8444, Special section p8-8, 5/6p, 1c Volume 376(8444): 8.

- [23] Risk A, Dzenowagis. Review of Internet health information quality initiatives, Journal of Medical Internet Research 2001.
- [24] Seagate Barracuda 7200.10 500GB 7200RPM SATA II 16MB NCQ 3Gb/s HDD (ST3500630AS), Retrieved March 10<sup>th</sup>, 2007 from <http://qmb.co.nz/p.aspx?106523>
- [25] W3Schools.SQL SELECT Statement. Retrieved July 14<sup>th</sup>, 2006, from [http://www.w3schools.com/sql/sql\\_select.asp](http://www.w3schools.com/sql/sql_select.asp)
- [26] Houlgate, G. (2006, July 2006). Network topology. Retrieved July 15<sup>th</sup>, 2006, from [http://en.wikipedia.org/wiki/Network\\_topology](http://en.wikipedia.org/wiki/Network_topology)
- [27] Knight, W. (2004, July 06). Evolution could speed net downloads. NewScientist.com news service.
- [28] Walter, M (1997). RAPID APPLICATION DEVELOPMENT. Retrieved August 10<sup>th</sup>, 2006, from <http://csweb.cs.bgsu.edu/maner/domains/RAD.htm>
- [29] [www.netapplications.com](http://www.netapplications.com) . "Operating System Market Share for July, 2006 " Retrieved 20 August, 2006 <http://marketshare.hitslink.com/report.aspx?qprid=2>
- [30] .NET Framework architecture – Retrieved 20 August, 2006. <http://www.baboo.com.br/absolutenm/articlefiles/5002-framework.JPG>
- [31] Common language Runtime – Retrieved 21 August, 2006. [http://en.wikipedia.org/wiki/Image:Overview\\_of\\_the\\_Common\\_Language\\_Infrastructure.png](http://en.wikipedia.org/wiki/Image:Overview_of_the_Common_Language_Infrastructure.png)
- [32] Mono Project – Open source .NET Framework – Retrieved 21 August, 2006. <http://www.mono-project.com/Start>
- [33] Steve Warren, Richard L Craft (1999). Designing Smart Health Care Technology into the Home of the Future.
- [34] Maj. Niel A. Johnson, M. D. (2003, 20-09-06). "High Blood Pressure & 'White Coat' Hypertension." Retrieved september 20, 2006, from <http://www.armymedicine.army.mil/hc/healthtips/13/200308highbloodpressure.cfm>
- [35] George, R. (November 2005) SQL Server 2005 Express Edition Technical Articles
- [36] Nelson, TabSoft Software – Retrieved 21 August, 2006. <http://www.tabsoft.com/>
- [37] Brown, Helen. A web-based system to monitor and predict healthcare activity. Health Informatics Journal 2005 11: 63-79
- [38] Genetic algorithm. (2007, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved April 15, 2006, from [http://en.wikipedia.org/w/index.php?title=Genetic\\_algorithm&oldid=120919211](http://en.wikipedia.org/w/index.php?title=Genetic_algorithm&oldid=120919211)

- [39] THOMAS PICKERING, M., DPhil (October 2002) White coat hypertension should it be treated or not? *Cleveland Clinic Journal of Medicine* Volume, 584-585 DOI
- [40] Pearson product-moment correlation coefficient. (2007, April 11). In *Wikipedia, The Free Encyclopedia*. Retrieved, January 15, 2007, from [http://en.wikipedia.org/w/index.php?title=Pearson\\_product-moment\\_correlation\\_coefficient&oldid=121835653](http://en.wikipedia.org/w/index.php?title=Pearson_product-moment_correlation_coefficient&oldid=121835653)
- [41] Statistical hypothesis testing. (2007, April 12). In *Wikipedia, The Free Encyclopedia*. Retrieved, January 15, 2007, from [http://en.wikipedia.org/w/index.php?title=Statistical\\_hypothesis\\_testing&oldid=122245196](http://en.wikipedia.org/w/index.php?title=Statistical_hypothesis_testing&oldid=122245196)