FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
UNIVERSITI OF MALAYA
KUALA LUMPUR

XML FOR BROWSER-BASED ELECTRONIC MEDICAL RECORDS

MAIZIRAH IBRAHIM
WEK 010150

WXES 3182 : Projek Ilmiah Tahap Akhir II
Session 2004/2005

Supervisor :
PN. SURAYA HAMID

Moderator :
DR. TEH YING WAH

This report is proposed to
Faculty of Computer Science and Information Technology
University of Malaya

To fulfill the requirements for
Bachelor Degree of Computer Science

ABSTRACT

A medical record stores medical history such as diagnosis, results and history of a patient. The patient record is an account of a patient's health and disease after he or she has sought medical help. The record contains findings, considerations, test results and treatment information related to the disease process. Electronic medical record systems should be designed so that they can exchange all their stored data according to public standards. Giving patients control over permissions to view their record as well as creation, collation, annotation, modification, dissemination, use, and deletion of the record is the key to ensuring patients' access to their own medical information while protecting their privacy. Many existing electronic medical record systems fragment medical records by adopting incompatible means of acquiring, processing, storing, and communicating data. The advantages of representing medical records in XML format would be a platform-independent plain-text format. Record systems should be able to accept data (historical, radiological, laboratory, etc) from multiple sources including physician's offices, hospital computer systems, laboratories, and patients' personal computers. Consumers are managing bank accounts, investments, and purchases on line, and many turn to the web for gathering information about medical conditions; they will expect this level of control to be extended to online medical portfolios.

ACKNOWLEDGEMENT

I would like to express my gratitude to each and everyone who was involved in this project. I cannot achieve this without the help and support from all of you who has always been there for me. Among the many amazing people, a special thank you to the one who made it is possible to complete this part of the project is my hardworking, well-determined, helpful and generous partner Hafizam Mat Hussin. Millions of thank you to my supervisor, Pn. Suraya Hamid for your attention, pursuit of excellence, your thoughtfulness and tact in dealing with me and of course for all the guidance and knowledge that you've shared with me. Not forgetting Dr. Teh Ying Wah, you should know how honored I feel knowing you became the moderator for this project. I really appreciate your guidance and opinions given during the VIVA session. A fine group of people in Faculty of Computer Science and Information Technology, University of Malaya helped in supplements, contributing ideas and knowledge sharing. Most importantly, I thank my family for their support, patience, understanding and assistance to bring up this project. To each and everyone whom I forget to mention, for everything you have been to me, I will never forget that. Thank you.

TABLE OF CONTENT

LIST OF TABLE

LIST OF FIGURE

CHAPTER 1 INTRODUCTION

1.0 Project Definition

Although Malaysia is marching towards the vision 2020 and the MSC status, most of the hospitals or clinics in Malaysia still using manual system to manage patient's records. Manual system is less efficient and error prone compare to computer-based system. The transformation of manually to computer-based system will certainly ease the job of management as well as reducing errors, potential risks, waiting time as well as appropriate dispensation to computer-based system will certainly ease the job of management as well as reduced errors, potential risks, waiting time as well as appropriate dispensary.

Electronic Medical records is a system that is used to simplify day to day task in running a clinic group in a modern hospital at the same time provide management with the necessary information to make strategic decisions. It serves essentially as a medium for communication between the diverse collaborating functional subsystems in a group of clinic in a hospital.

Electronic medical record summaries are now being deployed, using XML to represent, store and display information that is extracted from existing hospital systems. An XML data warehouse is designed to provide a clinician-centered view of the patient's information, with a web browser as the user interface.

Patient records are created and maintained as XML documents within an XML database. Data feeds from the Hospital's Information System (HIS) make use of the existing HIS interfaces with some customization to meet the data requirements of the XML patient record database. HIS extracts are triggered by events occurring within the system. As updates take place, the XML database can provide full auditing of transactions that have taken place. Access to the browser application is controlled by a security layer that can be integrated with the HIS system. The Web server provides the middle layer between the user interacting with the browser and the XML database.

This system suits well six types of users; doctors, pharmacists, receptionist, account clerk and management staffs. Only authorized users are allowed to create, edit, delete, update or maintain the data under their respective scope. Validation of the user's login is indeed needs in order to access to the system. Functions provided for this system included managing patient records for a clinic group in a hospital, billings, and also report generation hence, it is hoped that this system will carry lot of benefits to all of the users besides reduce paperwork, working and storing space, the cost and redundant data entry and also increase productivity.

1.1 Project Objectives

Before any further planning or development is carried out, it is vital to draw out the objectives of the system, in order to provide a clearer picture of the requirements and also the needs of implementing the system. A number of objectives have been outlined for this system, which includes:

- **To keep the medical records in more secured way and eliminate the duplicated as well as redundant data.**

  All patient records are stored in a database. Only the authorized personnel are permitted to access the data. Medical records of patient can only be accessed by authorized personnel like doctors. So, there's no need to worry whether the medical report will get noticed by other people. Besides, user is alerted if patient ID or IC is already present when creating a new patient record. This help to avoid case medical records misplaced or lost. It will also eliminate the duplicated data in the database and also the common error with the manual system.

- **To archive the paperless administration.**

  Much paper work will be eliminated when all medical records are input to the computer and stored in the database.

- **To provide doctors safer and easier treatment or prescription writing.**

  The doctor can write patient's prescription into computer for future reference. No more paper or stationary used to note down the prescription. Doctors can be easily referred to patient's previous treatment and profiles. Pharmacists can refer to patient's prescription by clicking the mouse and

keyboard to give suitable medicine to patient. It is safer than manual system because system eliminates common error with the manual system like prescription lost when passing from doctor to pharmacist.

- **To increase the efficiency of handling or retrieving medical records or data.**

  For the first time visit there is no record for the patient. The staff will do the registration for him or her. The staff will key in the patient's information and save it into database. The data of the patient may change from time to time; the staff can update the patient's profiles and save it again in the database. Furthermore, the patient's records can be retrieved easily and quickly. Before the consultation starts, doctors can easily achieve the patient's records through the database. This can avoid the doctors thumbing through several pieces of paper before he finds what he needs. So, the system will reduce the expenses and waiting time.

- **To create an appointment module.**

  Patients will be able to schedule their own appointment with the doctors and they will be informed if the doctor changes the appointment.

- **To produce detailed billing statement for patient's account.**

  This system will provide a more systematic way of handling the billing and payment system. A detailed statement will be produced and patient will be aware of what they are actually being accounted for during their stay or service at the hospital.

- **To produce standard medical reports.**

  The certificate produced in standard format and it is detailed, neat, and easy to read. It is available at any time required.

1.2 Scope and Limitations.

1.2.1 Scope of the project

The system is divided into two separate modules as depicted below:

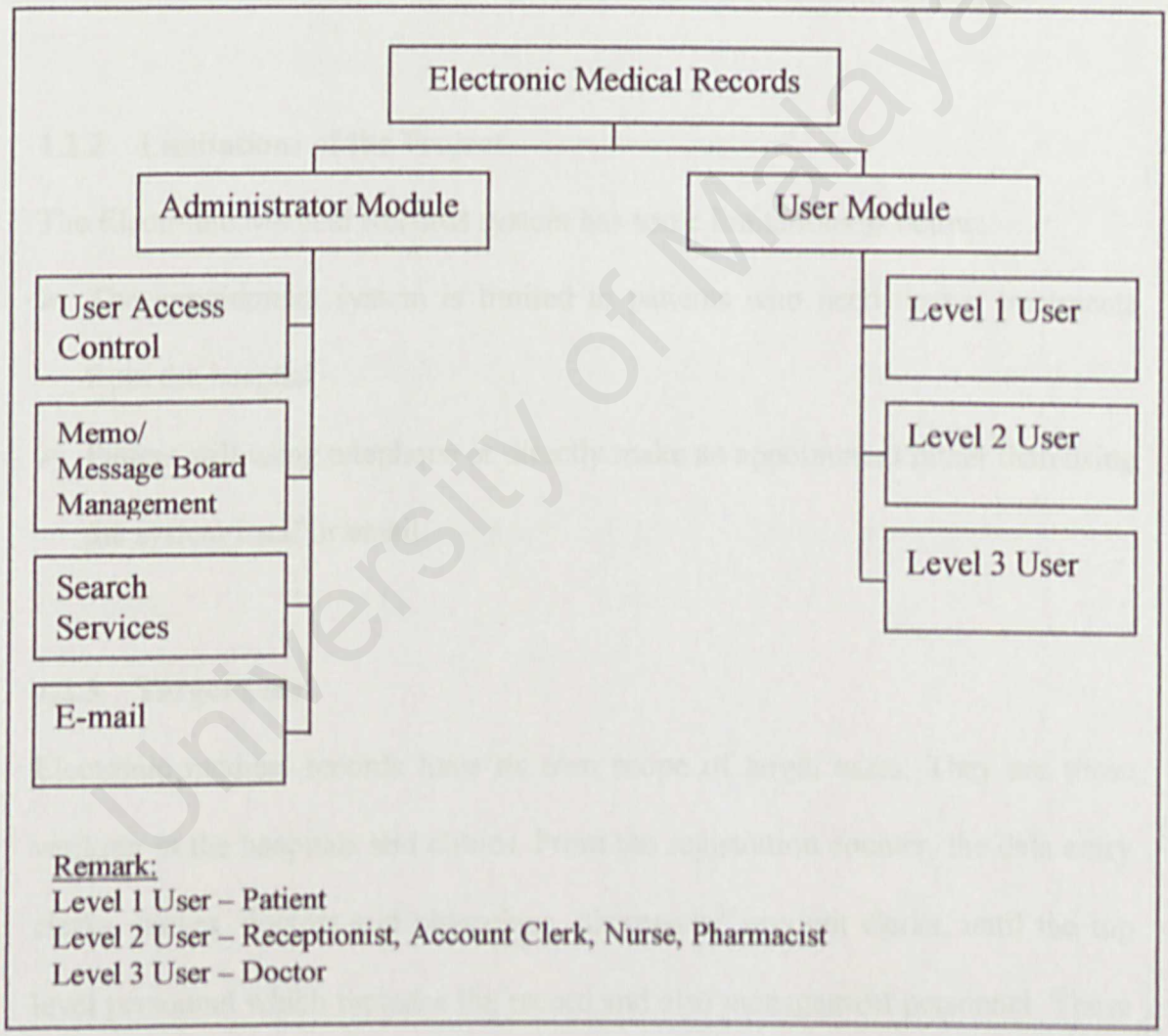

Figure 1.1: The Project Scope

The scope that will cover by the system includes:

- ❖ Login to verify only the authorized person can access data under their respective scope.
- ❖ Develop the computerized system for patient registration.
- ❖ Develop a database system to store all related data.
- ❖ Develop a function that allow patient to make appointment.
- ❖ Generate billing statement for patient.
- ❖ Generate medical reports for reference and analysis purpose.

1.2.2 Limitations of the Project

The Electronic Medical Records system has some limitations as below:

- ➢ The appointment system is limited to patients who need further treatments from the hospital.
- ➢ Patient still using telephone or directly make an appointment rather than using the system itself or email.

1.2.3 Target User

Electronic medical records have its own scope of target users. They are those working in the hospitals and clinics. From the registration counter, the data entry clerks, nurses, doctors and physicians, pharmacist, account clerks, until the top level personnel which includes the record and also management personnel. Those must have permission to login. They are called authorized user and they will have their own username and password that will be requested before they enter the

system. Besides them, the patients themselves will be able to access certain functions in the system such as their medical report, billing statement and can even make appointment with the physicians.

1.3 Expected Outcomes

Once an electronic Medical record is developed, it is expected to achieve the following features:

- User friendly and easy-to-learn system.
- Provide a more efficient and effective way of sharing data.
- Acceptable respond time when retrieved data from a save data to database.
- A good security system to prevent patient's medical records access by unauthorized person.
- A compatible system that support multiple users.
- Provides different view for different user of a same module.
- Data input will be examined and errors will be told by message.

1.4 Project Schedule

A project is a planned undertaking of scheduled activities and its management to reach a goal. Since a project may involve extensive effort, it must be property managed. Since electronic medical records is a final year project, which needs to be completed within a period of time. Planning is done to:

- Define the goals
- Define and allocate resources
- Establish timetable, schedule work
- Track and monitoring project
- Report and document the project

This project is divided into two phases, which will be referred to as Semester I and Semester II. During semester I, research on introduction, literature review and system analysis & design are carried out. In semester II, system coding, testing, evaluation and training will be done. Documentation is done from the beginning of the project until the end of the project.

1.5 Hardware and Software Requirements

The following is a list of facilities that will be needed or essential for the project development:

Hardware Requirements:

- Pentium III 400MHz computer and above
- At least 128 MB RAM
- 20 GB Hard Disk Space
- Standard Input/Output Devices

Software Requirements:

- Extensible Markup Language (XML) 1.0
- Microsoft SQL Server 2000 – Database repository
- Java Scripting Language
- Windows NT Server 4.0- Network Operating System
- Microsoft Windows XP
- Microsoft Internet Explorer 5.0

1.6 Overview of the chapter

Chapter 1: Introduction

This chapter consists of the project definition of electronic medical records, project objectives, scope and limitation project, expected outcomes, project schedule and hardware and software requirements.

Chapter 2: Literature Review

The literature review gives brief explanation on topics researched and studied that are relevant to this project. Anyway, the topics are definition of electronic medical records, comparison between manual and computer-based system, development tools consideration, and etc.

Chapter 3: Methodology

This chapter emphasizes on methodology been employed and information gathering techniques.

Chapter 4: System Analysis

This chapter consists of analysis of the functional requirement and non-functional requirement of the project. It also gives the explanation about the development software and platform chosen to develop this system. System requirement also listed in this chapter.

Chapter 5: System design

This system gives some detail about the design of the proposed system. In this chapter, there will be the functionality design, graphical user interface design and database design of the proposed system. This chapter explains the conceptual and technical design of the system. It covers the structure chart, data flow diagram, database design and user interface design.

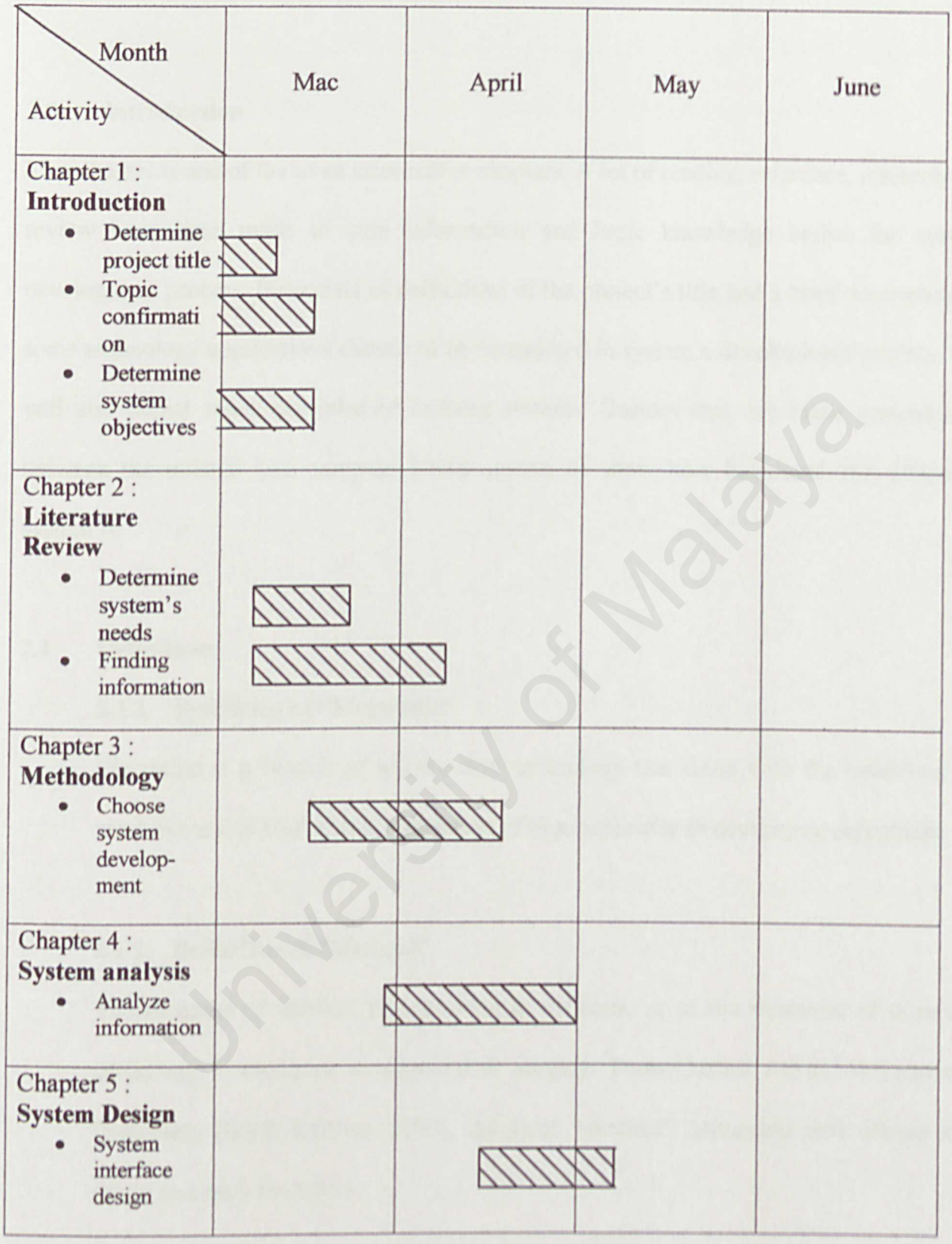

Figure1.2: Project Timeline

CHAPTER 2 LITERATURE REVIEW

2.0 Introduction

This chapter is one of the most informative chapters. A lot of reading, reference, research and review have been made to gain information and basic knowledge before the system development process. It consists of definitions of the project's title and a brief description of some technology applications chosen to be considered in system's development process. We will also depict some examples of existing systems. Besides that, we made comparisons between the manual and computer-based system to show how beneficial this proposed system is.

2.1 Definitions

2.1.1 Definition of "Electronic"

Electronic is a branch of science and technology that deals with the behaviour of electrons and it also means application of this, especially in developing equipment.

2.1.2 Definition of "Medical"

The meaning of medical is pertaining to medicine or to the treatment of diseases, pertaining to medicine as opposed to surgery. From Oxford Advanced Learner's Dictionary (Sixth Edition, 2000), the term "medical" connected with illness and injury and their treatment.

2.1.3 Definition of "Record"

Record is a written account of something that is kept, so that it can be looked at and used in the future. As in computing, record is defined as a collection of related fields within a single entity. It is also a collection of fields arrange in a predefined format.

2.2 Manual-based Medical Records System

In general, there are many hospitals and clinics still using method to manage medical center's information such as patients and medical records. All patient information, medication information and treatment involved huge amount of paper documents. In the traditional way, patient's profiles and medical records are write in the file and stored in the cabinet.

When the patient visits the medical center, or come for consultation, they have to show their registration card or identity card to the receptionist. Then, the staff will search the patient's file in the cabinet based on the registration number or their name if they forget to bring along the registration card. After the patient's record is found, patient will be given one queuing number. The file then will be passed to the physician in the room. Physicians traditionally record treatment and prescription as hand written progress notes in files. Hand written information is neither structured nor codes, and therefore cannot be easily used for automated decision support, research and outcomes analysis. Then, patient's file will be passed to the dispensary department for giving medicine. The overall operating of a group of clinics in a hospital consumes a considerable of time from registration, searching for patient's record, until to pharmacists.

2.3 Computer-based Medical Records System

Due to inefficiency operating using manual system, many medical centers are changing their system to computerized system to assist their daily operation. All information or records are saver in computer. Therefore, user can easily retrieve and manage all related information such as medical records, medication information and others.

When patients come to medical center for treatment, they need to show their identity card to the receptionist. The receptionist will input their identity number or name into the system and easily found their record. The patient's records will be shown in the screen of the computer. This would certainly save a lot of time searching for patient's records compared to the manually operated system. Once the record is displayed, a queuing number will be generated to the patient and put into waiting list. If a new patient comes, registration is carried out. All the data are input through a standard form and a new record is carried out. All the data are input through a standard form and a new record is created for that particular patient. Also, the record of the patient is displayed on the computer screen in the physician's room. All the treatment, diagnostic and prescription of the medicine to the part can key in through the keyboard and save in the database easily. Prescriptions are forwarded to the pharmacist. Pharmacists can base on the information displayed and gives the appropriate medicine to the patient. This will indeed reduce errors, potential risks and patients' waiting time.

The management of the medications and other clinic utilities also benefits from the computerized system. The medications inventory will automatically update after the medicine is giving away to the patient. Pharmacist can know the quantity of the medications

by just clicking a few keyboard buttons rather than to check the remaining one-by-one. The staff only can access the inventory of the clinic utilities in the database. Message will be show when quantity of any clinic utilities below than certain amount. After that, staff in charge will start the procurement for that clinic utility.

2.4 Comparisons between manual and computer-based system

Table 2.1: Comparisons between manual and computer based system

Manual Medical Records System	Computer-based medical Records System
Records are written in papers and stored in cabinet or cupboard	Records are stored in database
Staff needs to search patient's record one-by-one in the cabinet, it may take time	Records can be search easily through database system
Risks lost or misplaced records are high	Records are stored safely and securely in the database
Hand-written may be not structured, it is difficult to read or analyze	All information is keep into standard form and easily to refer
Inventory of medications and other clinic utilities keep in a book. It is difficult to update the quantity of the medications	All inventory of medications and clinic utilities stored into system. Staff easily to notice the quantity of the medications
Analysis of patient's ratio, calculating of total income, reports for medication	Analysis of patient's ratio, calculating of total incomes, reports for medications and etc are easily to be generated

The management way is based on individual. So, if a new staff quits, new staff needs a lot of time to pick up what the previous staff did. Adaptation of new style may caused inconsistency	The management is based on system and not individual, so it is easy for new staff to learn the system and it is more formal

2.5 Review of existing software

2.5.1 Hospital Selayang

http://www.selayanghospital.gov.my

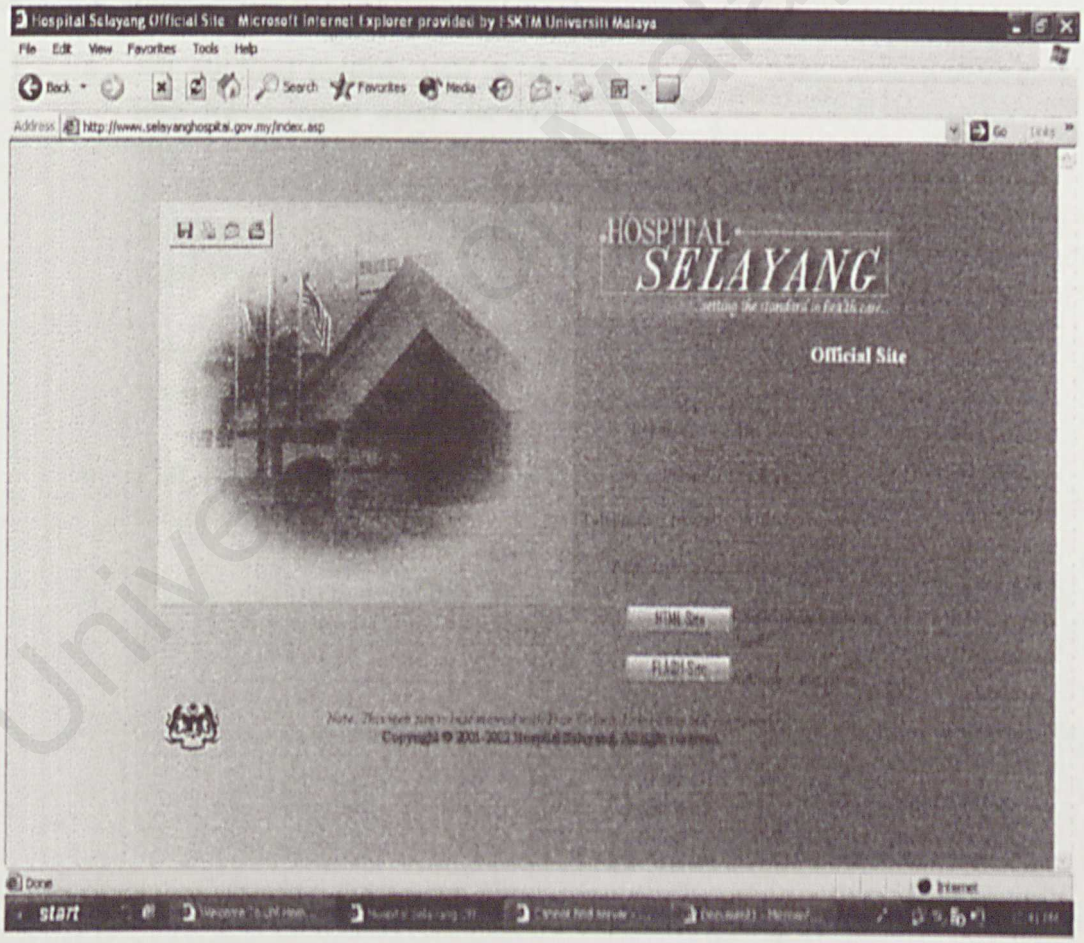

Figure 2.1: Hospital Selayang Main Page

Strength	Weakness
• Patient's medical records guidelines and clinical protocols are instantly available and can be accessed in one integrated workstation at any place and at anytime in the hospital. • Introduced two types of tracking system namely the Folder tracking system and Document Tracking System. These systems monitor the movement of the records in the hospital and provide statistical data • All medical records will be on line for 2 years after the last visit, after which it will be digitally archived off line for duration according to medico-legal requirements. • Provides on-line rescheduling appointment • The design is simple but effective • Well-presented information layout and the graphic images.	• It is more like a website of information. • The real system is a stand-alone system. Therefore, it's not available at this site but only accessible in the hospital. • There's only one on-line service provided. No interaction with public users.

2.5.2 Kajang Medical Center

http://www.kajangmc.com

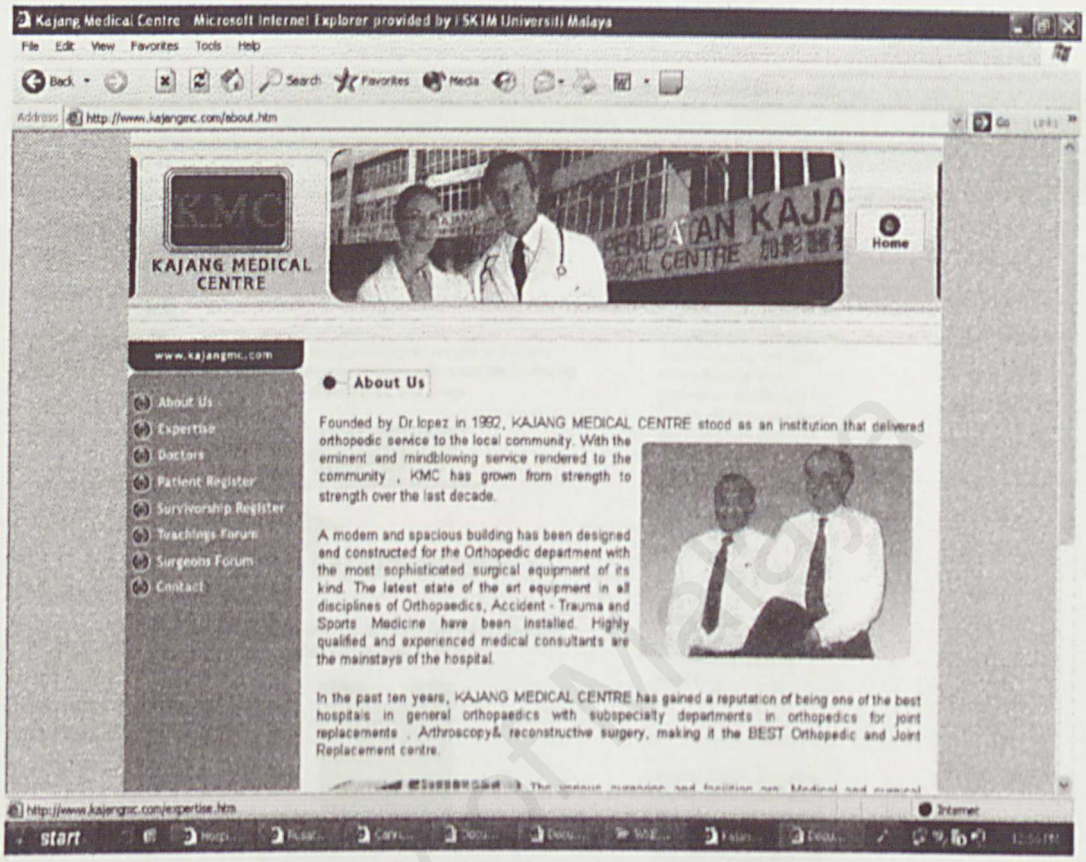

Figure 2.2: Kajang Medical Center Main Page

Strength	Weakness
• Provides patient's record retrieval on-line • It is a small system and very little service provided	• It is limited to one department only • Very little of information provided • Not user-friendly

2.5.3 Patient Centered Access to Secure System Online

http://www.pcasso.com

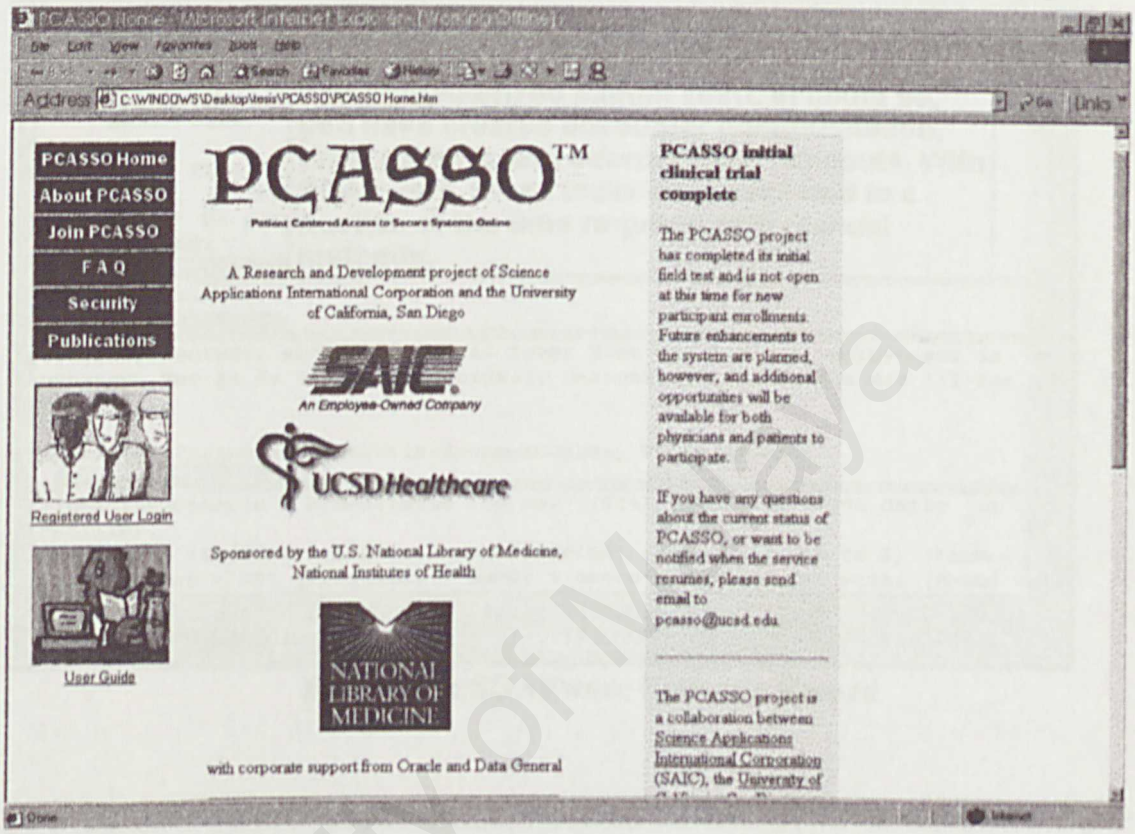

Figure 2.3: PCASSO Main Site

Strength	Weakness
• Good presentation • Attractive interface designs • Make use of colours and graphical coordination	• Not yet established • Not many functions

2.5.4 SOAPware Basic

http://www.docs.com/Products/soapware.htm

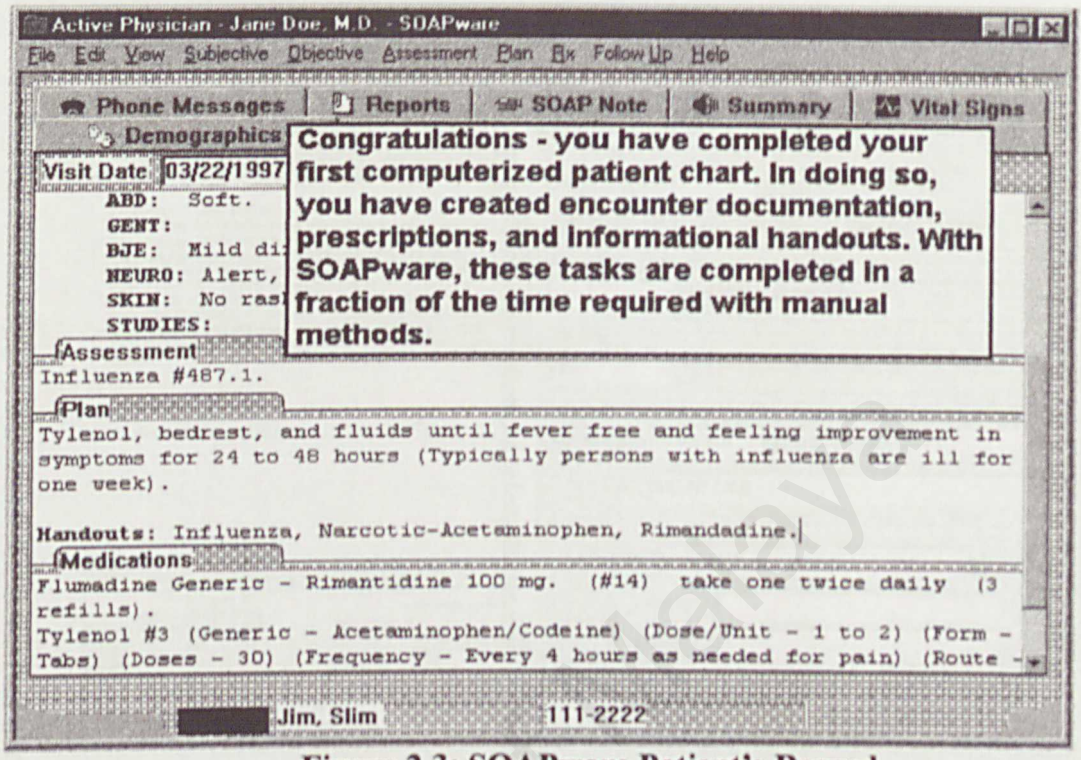

Figure 2.3: SOAPware Patient's Record

Strength	Weakness
• Organized and standard forms • Complete and useful functions • Perfect for warehousing	• Not easy to implement and understand • Quite complicated forms

2.5.5 Misys Electronic Medical Records

http://www.misys.com

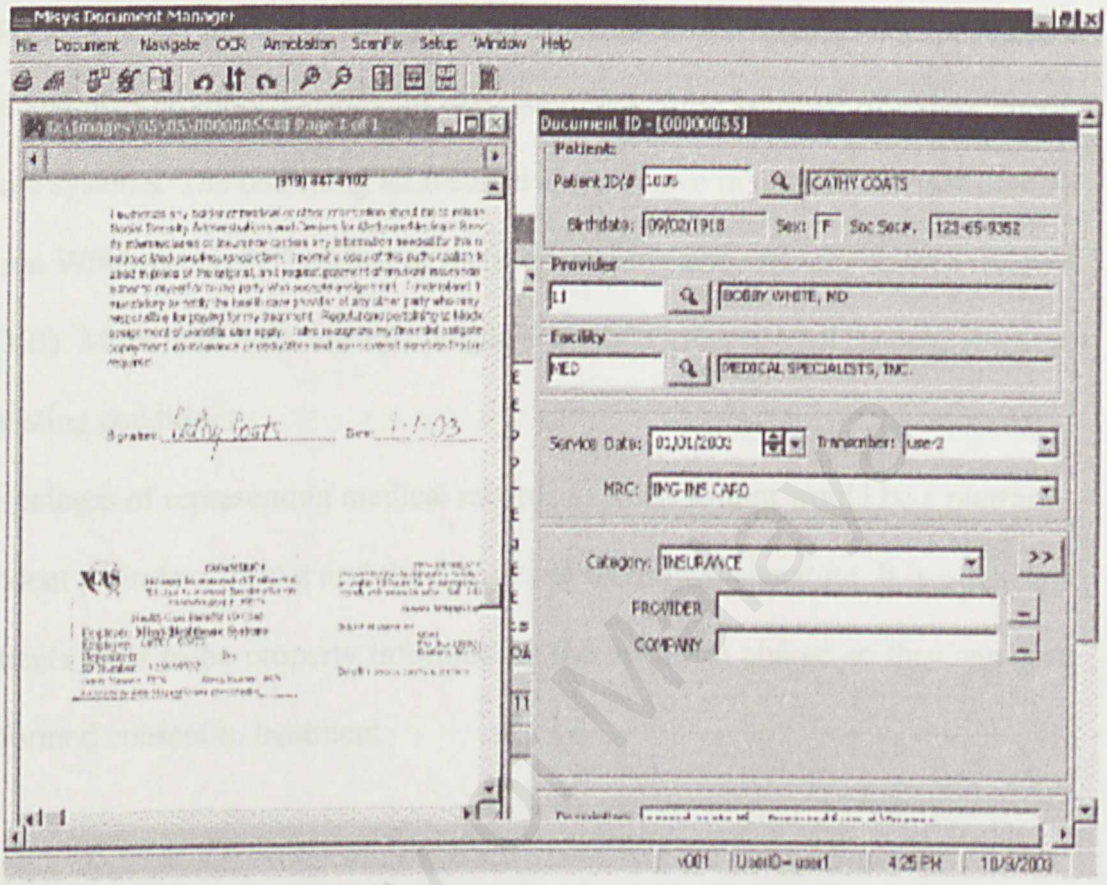

Figure 2.4: Misys Patient's Record

Strength	Weakness
• Very comprehensive • Performs quite a number of modules • Very useful and effective • Simple interfaces	• Not easy to learn • Not available for patient's personal access

2.6 Reviews on Technology Consideration

2.6.1 XML-based Medical Records

XML promises a means of providing flexible, incremental employment of electronic systems and also provides a mean for easy communication between disparate binary electronic systems. The one thing all medical system have in common is that they all (99%) run Windows on PC's. Most of the medical programmers are versed in Visual Basic (VB). More than 80% use this language as their choice. XML should leverage these existing conditions.

The advantages of representing medical records in XML format would be a platform-independent plain-text format and the features of the digital signature. It is often said that patients want to be properly informed by the doctor in charge so they can give their informed consent to treatment.

XML Web Services

XML Web services are the fundamental building block in the move to distributed computing on the Internet. Open standards and the focus on communication and collaboration among people and applications have created an environment where XML Web services are becoming the platform for application integration.

Applications are constructed using multiple XML Web services from various sources that work together regardless of where they reside or how they were implemented.

XML Web services are built on XML, SOAP, WSDL and UDDI specifications. These constitute a set of baseline specifications that provide the foundation for

application integration and aggregation. From these baseline specifications, companies are building real solutions and getting real value from them.

While much work has been done to make XML Web services a reality, more is needed. Today, people are having success with XML Web services, but there are still things that are left as an exercise for the developer e.g. security, operational management, transactions, reliable messaging. The Global XML Web Services Architecture will help take XML web services to the next level by providing a coherent, general purpose model for adding new advanced capabilities to XML Web services which is modular and extensible.

Using XML for flexible data entry in healthcare

The central aspect of the user interface is the transparency of the XML markup for the clinical user. The user is interested in content issues only, not in formatting issues. The XML transparency is achieved with a software component which will be referred to as document manager. The document manager provides the user with an HTML form that corresponds to the XML document (pathology report) and which enables the pathologist to create, specifically select and update XML structured data. For the construction of new reports, the document manager needs to know both, the document structure (nested, repeatable, optional elements) and the form structure (user prompts, different form objects, connections to remote data servers).

The document manager uses the XML template for both, the automatic generation of an HTML form (user interface) and the generation of an XML document (storage format of the data entered by the user). As a consequence, we only need to add new templates or change existing templates and the document manager will automatically provide/adapt the corresponding front & back end. The application engineer can focus on XML templates, i.e. on structural and modeling issues rather than on logical and programming issues. With such an approach changing user requirements can be quickly satisfied and the clinical user no longer has to put up with inflexible data structures.

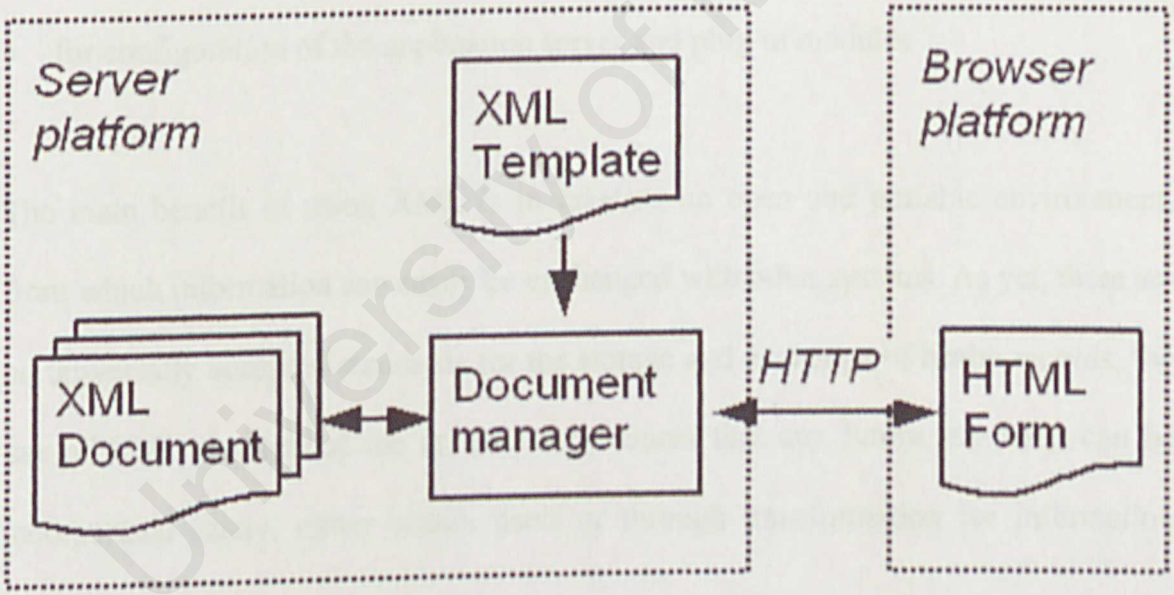

Figure 2.5: Overall Software Architecture

The architecture for the management of XML documents based on a Web platform with a server back end and a browser front end. The template describes and relates both, the front end (form) and the back end (document). The application developer merely provides a template and the document manager will automatically generate a corresponding front & back end.

Benefits of the XML approach

- as the integration data format (with SGML) for legacy systems
- for messaging through TCP/IP sockets in the application server
- for profiling and transformation to pure HTML in the client browsers
- as the base data model (XML DTDs and schema) for the patient record repository
- for configuration of the application server and plug-in modules

The main benefit in using XML is in creating an open and portable environment, from which information can easily be exchanged with other systems. As yet, there are no universally accepted standards for the storage and exchange of health records; the use of XML throughout the architecture ensures that any future standards can be incorporated easily, either within itself or through transformation for information exchange.

XML Coding Example

```
<?xml version="1.0"?>
<project name="HelloXML" default="build" basedir=".">
    <property name="debug" value="true" />
    <target name="clean" description="remove all generated files">
        <delete file="HelloXML.exe" failonerror="false" />
        <delete file="HelloXML.pdb" failonerror="false" />
    </target>
    <target name="build" description="compiles the source code">
        <csc target="exe" output="HelloXML.exe" debug="${debug}">
            <sources>
                <includes name="HelloXML.cs" />
            </sources>
        </csc>
    </target>
</project>
```

2.6.2 Data Warehousing

Data warehousing is a huge repository of historical data used to answer queries and uncover the trends; contains a snapshot of data at a point in time as opposed to operational databases that are updated with every new transaction. Data warehousing has arisen to store huge amount of historical data from such system as retailers' point-of-sale systems.

Briefly, a data warehouse is a database that contains data from many sources including operational sources. It is update periodically, and it comes with a repository of metadata that describes precisely what each type of data means in terms that marketing folks, the sales force, management and others can understand.

5 Steps in data warehousing project

- To define the business uses of the data.
- Create the data model for the warehouse, which means defining the relationships between data elements.
- The 'cleanse-the-data' step which requires moving the data out of the operational systems and transforming it into the desired standardized format.
- To consider the users' point of view by selecting the tools they will use, and then training them on tool use.
- To monitor usage and system's performance.

Characteristics of a Data Warehouse

- ***Subject orientation***. Data are recognized based on how the users refer to them.
- ***Integrated***. All consistencies regarding naming convention and value representations are removed.
- ***Non-volatile***. Data are stored in read-only format (not changed over time).
- ***Time variant***. Data are not current but normally in time-series.
- ***Summarized.*** Operational data are mapped into a decision-usable format.
- ***Large volume.*** Time-series data sets are normally quite large.
- ***Non-normalized.*** Warehoused data can be, and often are redundant.
- ***Metadata.*** Data about data are stored.
- ***Data sources.*** Data come from internal and external un-integrated operational systems.

Table 2.2: Operational data store vs. data warehouse characteristics

Characteristics	Operational Data Store	Data Warehouse
How is it built?	One application or subject area at a time.	Typically multiple subject areas at a time
User requirements	Well-defined prior to logical design	Often vague and conflicting
Area of support	Day-to-day business operations.	Decision support for managerial activities.
Type of access	Relatively small number of records retrieved via a single query	Large data sets scanned to retrieve results from either single or multiple queries
Frequency of access	Tuned for frequent access to small amounts of data	Tuned for infrequent access to large amount of data
Volume of data	Similar to typical daily volume of operational transactions	Much larger than typical daily transaction volume
Retention period	Retained as necessary to meet daily operating requirements	Retention period is indeterminate and must support historical reporting, comparison and analysis.
Currency of data	Up-to-the-minute; real-time	Typically represents a static point in time
Availability of data	High and immediate availability	Immediate availability is less critical

Typical unit of analysis	Small, manageable, transaction level units	Large, unpredictable, variable units
Design focus	High-performance, limited flexibility	High flexibility, high-performance

Maintenance issues for data warehousing systems

- The challenges to learn about business and feeder system changes that will affect the DW/DSS systems
- The needs to figure out if, when, and how to purge data
- Will be motivated to store data in the data warehouse "for data's sake"
- Endless opportunities to tune DW/DSS system databases
- The needs to balance the need for building aggregate structures for processing efficiency with the desire not to build a maintenance nightmare
- Will be pressured to implement a means to interactively correct data in the data warehouse (and perhaps send back corrections to the transaction processing system)
- Uncertainties which tools are most appropriate for a certain task
- The needs to figure out how to test the effect of structure changes on end user written queries and reports
- Maintaining data warehouse architecture may be much harder than establishing the architecture
- You will find it is far more expensive (and complex) to maintain a data warehouse than to build one

2.7 Summary

After conducting research and reviews, we are aware of what the system should looks alike. We have the picture on how the system should be operating and what the system should provide. Besides that, we gained additional knowledge regarding the technologies to be used and the reason of why they should be implemented.

CHAPTER 3 : METHODOLOGY

3.0 Introduction

After analyzing the survey and findings from the literature review in the previous chapter, this chapter will specify the justifications for the chosen methodology for the project, which also include all outlined procedures that should be covered in order for user to understand the project requirements better.

3.1 Fact-finding Techniques

The various resources that were surveyed can be summarized as fact-finding techniques in order to obtain and gather reliable, relevant, adequate and comprehensive information. Several fact-finding techniques are used consecutively to ensure the success in conducting the system development and appropriate planning. A few techniques have been used in this project to gather information. It is also called information gathering or data collection.

- **Surfing the internet**

  Internet surfing today's world is a very efficient way of gathering information. There are many web site available that provide useful and expertise information, which is needed in the system. Some web site had provided very useful samples that can serve as guidelines in determining system requirement.

- **Reading**

  A lot of published literatures have been read in order to gather information of the users' needs, system development needs and technical issues of the proposed system. All these can be categorized into the printed material (especially books and journals)

and non-printed material such as electronic documents and so. Through reading, ideas are managed to get from books, magazines and journal to be implemented in the proposed electronic medical records. Useful information has been found from the computing magazines and newspaper.

- **Library FSKTM**

  Previous seniors' thesis have been read through in order to gain an overall understanding on how a system was developed, what were the functional and non-functional requirements, and other related data. The general structure of each thesis has also been observed to find out the steps taken in carrying out a thesis.

- **Group discussion and Brainstorming session**

  Brainstorming session among group member, course-mates and supervisor, is a productive way in collecting ideas, suggestions and solutions of this project. However advices from supervisor and opinion from partner are also very significant in order to make this project successful.

- **Interview**

  In order to well suit the system with the real world environment, interview is the one important portion to be carried out. By interviewing, actual information can be obtained.

- **Observations**

  The existing search system of the Electronic Medical Records had been reviewed. Through this technique, the method of managing medical record in the existing system has been observed and defined. Besides the existing system is test to find out the functionality and the problem faced by the system itself.

3.2 System Development Model

Software engineering is important in organizing and executing the development of a system. Systematic analysis using *System Development Life Cycle* (SDLC) is a standard methodology to ensure the development process fulfils all the required aspects. The proper sequence of developing a system is called a "life cycle" that is implemented to ensure all individuals involved or interested to be involved in the project has the knowledge on how the system will be developed. This methodology is proven effective and it is still being used at the moment.

The development of a system using SDLC is divided into several software process models such as the Waterfall Model, V Model, Prototyping Model, Transformational Model and other models. This Medical Record System will be developed using the *Waterfall Model*. This model has several advantages over other models that ensure the development process is properly organized, high in quality and meets the standard.

3.2.1 The Waterfall Model

The system development is done based on the 5 major phases to make the system more organized and achieve its objectives. The phases are:

- Early review phase
- System analysis phase
- System design phase
- Coding/Programming phase
- Testing and execution phase

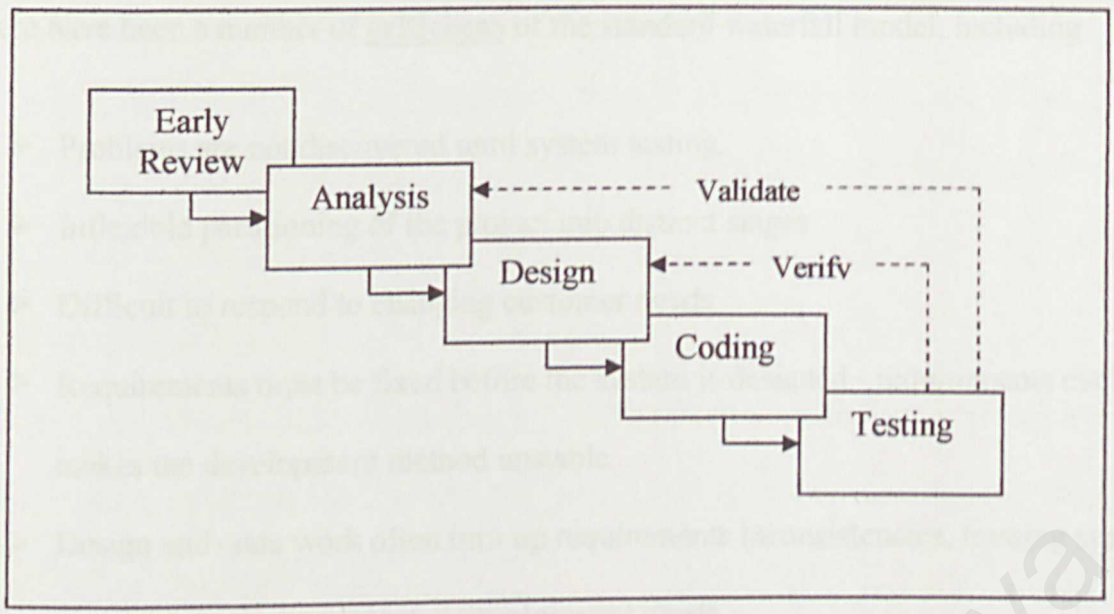

Figure 3.1: The implementation of *Waterfall Model*

The **advantages** of implementing this model are:

- Important stages in the development process can easily be recognized.
- It is easy to separate one stage from another.
- Presents a very high-level view of what of what goes on during development.
- It is easy to estimate period of time needed for an activity or stage.
- Testing is inherent to every phase of the waterfall model.
- It is an enforced disciplined approach.
- It is documentation driven, that is, documentation is produced at every stage.
- Validation ensures that the system has implemented the entire requirements, so that each system function can be traced back to a particular requirement in the specification.
- Verification ensures that all functions work correctly.

There have been a number of **criticisms** of the standard waterfall model, including

- Problems are not discovered until system testing.
- Inflexible partitioning of the project into distinct stages.
- Difficult to respond to changing customer needs.
- Requirements must be fixed before the system is designed - requirements evolution makes the development method unstable.
- Design and code work often turn up requirements inconsistencies, missing system components, and unexpected development needs.
- System performance cannot be tested until the system is almost coded; under capacity may be difficult to correct.

3.2.2 Project Development Phases

As mentioned above, there are 5 stages or phases in the development of this system- Electronic Medical Records. A brief description for each phases stated as below:

Phase I: Early Review

-It is also known as feasibility study. It could also be described as investigation and identification phase.

- Identification and Investigation Phase

- At this point, early review is being done to gather information and description on the system that is going to be developed.
- This phase will give the definition and the objective of the Electronic Medical Record in replacing the existing manual system.

- Early investigation on the manual system has been done to identify the process involved and the problem that arises. User's demand is also being considered in this phase.

Phase II: System Analysis

-The analysis phase is done to understand how the Electronic Medical Record that is going to be developed will be able to solve the problems that exist in the manual system that had been identified from early reviews.

-It includes system review, system requirements and specifications, suggestions for system's content, development tools, analysis and interface medium being used.

Phase III: System Design

-Design is the creative process that requires understanding and natural talent to transform the problem into a solution. It is being acquired by reviewing systems that already exist. The description of a solution is also called design.

-The phase involves explanation about the whole system that is going to be developed and the expected system performance. In this phase all of the system properties such as system architecture, database design, process design, and interface design are being explained. This is done to simplify the interface developing process of the system that is going to be developed.

-The phase is being explained in details in Chapter 5.

Phase IV: Coding/Programming

-Programming is one of the most important aspects in developing a system. It will determine whether the system manage to achieve its objectives. This makes the development environment vital in completing the programming process.

-All analysis and reviews had been done in details to determine the most suitable environment in achieving the project objectives. Besides all that, three main aspects that being emphasis on is the control structures, algorithm and data structure.

Phase V: Testing and Maintenance

-Testing is done to ensure that the system functions as it was suppose to. It is done to detect faults in the system so that all the modules developed are free from errors and the system can response to request effectively.

-Testing is one of the most important elements to ensure whether the system developed will be able to fulfill users' request. High quality system will be able to handle any type of system testing. To achieve this, all specification, design and programs done during the system development stage will be reviewed and re-evaluate.

3.3 Summary

In principle, the result of each phase is one or more documents which approved ('signed off'). The following phase should not start until the previous phase has finished. In practice, these stages overlap and feed information to each other. During **design,** problems with requirements are identified, during **coding** design problems are found and so on. The software process is not a simple linear model but involves in a sequence of iterations of the development activities.

Because of the cost of producing and approving documents, iterations are costly and involve significant rework. Therefore, after a small number of iterations, it is normal to freeze parts of the development, such as the specification, and to continue with the later development stages. Problems are left for later resolution, ignored or are programmed around. This premature freezing of requirements may mean that the system won't do what user wants. It may also lead to badly structured system as design problems are circumvented by implementation tricks.

During the final life-cycle phase **(operation and maintenance)** the software is put into use. Errors and omissions in the original software requirements are discovered. Program and design errors emerge and the need for new functionality is identified. The system must therefore evolve to remain useful. Making these changes **(software maintenance)** may involve repeating some or all previous process stages.

There are pros and cons of using the Waterfall Model and therefore, this model should only be used when the requirements are well understood. However, the Waterfall Model reflects engineering practice and thoroughly consideration has been made beforehand. Consequently, software processes based on this approach are still used for software development, particularly when this is part of a larger system engineering project.

CHAPTER 4: SYSTEM ANALYSIS

4.0 Introduction

System analysis is the process of extracting the needs of a system and what the system must do to satisfy the user's requirements. The goal of this analysis is to first understand the domain of the problem and the system's responsibilities by understanding how the users use or will use the system. In this chapter, we will focus on the requirements analysis. The term 'requirement' is not used throughout the software industry in a consistent way. In some cases, a requirement is seen as a high-level, abstract statement of a service that the system should provide or a constraint on the system. At the other extreme, it is a detailed, mathematically formal definition of a system function.

4.1 System Requirement Analysis

System requirement analysis can be divided into 2 sub-categories:

- Functional requirement
- Non-functional requirement

4.1.1 Functional Requirement

Functional requirement are functions or system abilities. Functional requirement also explain the interaction between the system and its environments. These are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do.

- Password function

Users have to login every time they want to use the system. This function requires user to input login name and password. Accessed will only be given if the password is correct for the login name specified.

- Update record function

The update function enables users to choose weather to eliminate, add or change the records in the database. The function requires input for update process and the updated record will be displayed as a result.

- Patient's information record function

This function will retrieve all patient information. At the end, the function will function will display a message of confirmation that the information has been recorded.

- Patient history check function

This function will display all of the information that had been recorded before from the patient's record.

- Billing function

This function is being used to change fees to patients for treatments and medications received. The information will then be forwarded to charge information form.

- Search function

This function enables user to track down record for a specific patient by their account number or patient identification card number.

4.1.2 Non-Functional Requirements

Non-functional requirements, as the name suggests, are those requirements which are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time, and store occupancy. Alternatively, they may define constraints on the system such as the capabilities of I/O (input/output) devices and the data representations. Non functional requirements are limitations where system must operate to eliminate the limitation. Non-functional requirements for this system are as follows:

- **Reliability**

This system is reliable and will not require a high maintenance cost if it is used according to the correct procedures.

- **Security**

There are security features while accepting input or while retrieving data such as password protection.

- **Effectiveness**

This means that input and output screens have a specific purpose in the system

- **Simplicity**

Screens and instructions are organized properly so that it is much easier for user to understand and use the system.

- **User-friendly Interface**

Interesting interface is a vital aspect needed to encourage user to use the system. It is also a user-friendly based to make it easy for user especially first-timer to use this system.

4.2 Software Requirement

4.2.1 Server Software requirement

- Extensible Markup Language (XML) 1.0
- Microsoft Access 2000
- Active Server Pages 3.0
- Microsoft Windows XP
- Microsoft Internet Explorer 6.0
- Adobe Photoshop 7.0

4.2.2 Client Software requirement

- Microsoft Windows Me / 2000
- Microsoft Internet Explorer 5.0 ++

4.3 Hardware Specification

4.3.1 Server Hardware requirement

- Pentium III 400MHz computer and above
- 20 GB Hard Disk Space
- At least 128 MB RAM
- Other standard computer peripherals
- Standard Input/Output Devices

4.3.2 Client Hardware requirement

- A Pentium 3 with 750 MHz processor
- At least a 64 MB RAM
- Network connection through existing network configuration or modem
- Other standard computer peripherals

4.4 Technology Considerations

4.4.1 Microsoft Windows XP

As the latest operating system, Windows XP is inherently more reliable than Windows NT. In addition to this, any "vulnerabilities" that are found in the operating system is acted upon quickly by Microsoft, ensuring that Windows XP will remain as stable and secure operating system as possible. A number of "behind the scenes" improvements and enhancements also ensure that Windows XP is a significantly better operating system than earlier versions, such as Windows NT. One of the most notable improvements is in the systems resilience

to loss of network - if the PC were to lose it's network connection, Windows XP is able to continue unaffected. This is largely down to the fact that a number of applications, such as Microsoft Office, are installed locally on the hard disk - on Windows NT these applications are accessed over the network.

4.4.2 Internet Explorer 6

Internet Explorer 6 is a set of core technologies in Microsoft® Windows XP system that provides enhanced privacy features and a flexible and reliable browsing experience for users of Windows XP, Windows Millennium Edition (Windows Me), Windows 2000, Windows 98, and Windows NT® 4.0 with Service Pack 6a or later.

Whether you are a home user browsing content or getting e-mail on the Web, an IT administrator deploying and maintaining a rich set of Windows Internet technologies, or a Web developer creating Web content, Internet Explorer 6 gives you the freedom to experience the best of the Internet.

Features and Technologies at a Glance

Internet Explorer 6 includes many new and enhanced features that can simplify the daily tasks that you perform, while helping you to maintain the privacy of your personal information on the Web.Here is a quick look at some of the major features of Internet Explorer 6 that help to provide a flexible and reliable browsing experience on the Web.

Flexibility

With new, innovative browser capabilities and features such as Auto Image Resizing, Image Toolbar, Media Bar, and Print Preview, it's easy to manage, save, and print your pictures and other media from Web pages. Features such as Favorites, Auto Complete, History, and the Search Companion help you quickly find what you need online. And the customizable browsing layout makes it easy to change your layout so you can experience the Web the way you want.

Web Privacy

Internet Explorer 6 helps you manage your security and privacy preferences while on the Internet with tools that help you safeguard your family's browsing experience. Manage cookies to help control the personal information that Web sites collect about you, set different levels of security for different sites on the Web with Security Zones, and use Content Advisor to help block access to objectionable content. These tools support the Platform for Privacy Preferences (P3P), a technology under development by the World Wide Web Consortium (W3C).

Reliability

Internet Explorer 6 helps deliver a more stable and more reliable web browsing experience. New fault collection services help to identify potential problems that need to be fixed in future updates to Windows Internet technologies.

4.4.3 Adobe Photoshop 7.0

Features:

- Image-editing software for photographers, Web and graphic designers
- Visually browse and retrieve images with enhanced file browser
- Simulate painting techniques with painting engine
- Liquify distorting tool and pattern maker plug-in
- Runs native on Microsoft Windows XP

Photoshop 7.0's new file management system comes in the form of a Windows Explorer-like file browser that allows users to easily sort and locate their images within various projects. Users can now organize projects by name, date, resolution, and a number of additional parameters.

The enhanced brush palette allows users to create custom brushes and save them as presets that can be accessed from the Tools options bar. Users can easily vary different aspects of the brush by changing the hue, opacity, or flow of the brushes for pastels, oils, and charcoal. Photoshop 7.0 also introduces a new Healing Brush and Patch Tool. With these tools, users can easily "heal" their images by removing scratches, blemishes, and other imperfections while preserving shading, lighting, and texture attributes.

4.4.4 Internet Information Service 6.0

a) Purpose

Internet Information Services (IIS) turns a computer into a Web server that can provide World Wide Web publishing services, File Transfer Protocol (FTP) services, Simple Mail Transport Protocol (SMTP) services, and Network News Transfer Protocol (NNTP) services. You can use IIS to host and manage Web sites and other Internet content once you obtain an IP address, register your domain on a DNS server, and configure your network appropriately. IIS is a component of the Microsoft® Windows® operating system.

The IIS Software Developer Kit (SDK) is a resource for developers who want to create applications that manage an IIS server, or Web applications that run on an IIS server.

b) Where Applicable

IIS 6.0 is included with Windows Server™ 2003 and is installable via the Add or Remove Programs item in the Control Panel. IIS 6.0 installs in a highly secure state, serving only static HTML content until other features and file types (such as ASP and ISAPI) are enabled.

IIS 5.1 comes with Windows XP Professional and is installable via the Add or Remove Programs item in the Control Panel.

IIS 5.0 comes with Windows 2000 Server and is installed by default. It also comes with Windows 2000 Professional and is installable via the Add or Remove Programs item in the Control Panel.

IIS is also available as a separate product for previous versions of the Windows operating systems, but this SDK covers only versions 5.0, 5.1, and 6.0, with some limited information for IIS 4.0.

c) Developer Audience

Please see this section in "IIS Administration" and "IIS Web Development" for a list of available technologies and their prerequisites.

d) Run-Time Requirements

Please see this section in "IIS Administration" and "IIS Web Development" for a list of run-time requirements.

4.4.5 Microsoft Access 2000

a) What's New for Microsoft Access 2000 Developers

Extensive changes have been made to the Microsoft Access 2000 Visual Basic object model to support new and improved features in the application. Many objects, properties, and methods have been replaced. To provide backward compatibility, most of the replaced components have been hidden rather than removed. This means that they don't show up in the Object Browser by default, but old code that uses the hidden components will still work correctly without modification. When you write new code, however, you should use the new objects, properties, and methods.

b) What's new about the Database window

The Database window in Microsoft Access 2000 provides a variety of options for viewing and manipulating database objects.

- **Use the Database window toolbar** Quickly find commands for creating, opening, or managing database objects.
- **Use the Objects bar** View database objects in the Objects bar its vertical orientation makes it easier to use.
- **Organize database objects into groups** Click the Groups bar to view your groups, which can contain shortcuts to database objects of different types.
- **Use new object shortcuts** In the Database window, quickly create a new database object by using a wizard, or open a new database object in design view.
- **Customize how you select and open objects in the Database window** If you choose, change the default behavior so that you select a database object by resting the pointer over it, and open an object by single-clicking it.
- **Select an object by typing its name** For example, select the Shippers table while viewing the list of table objects by typing **Sh**.

4.4.6 XML

XML is a markup language for documents containing structured information. Structured information contains both content (words, pictures, etc.) and some indication of what role that content plays (for example, content in a section heading has a different meaning from content in footnote, which means something different than content in a figure caption or content in a database table, etc.). Almost all documents have some structure. A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents

XML is an open standard for defining data elements on a web page and business documents. In contrast to HTML markup language, which defines how elements on a web page are displayed, XML defines the structured information those elements contain. XML is also a markup language that was originated from a subset of SGML together with HTML to use over the Internet via web browsers. XML documents can be viewed as containers for information. Within the primary container may be information and more nested containers, which themselves may contain information and more nested containers (Simon, 2000).

An XML document is a database only in the strictest sense of the term. That is, it is a collection of data. As a 'database' format, XML has some advantages [as it is a meta-data language] because of its self- describing and it can describe data in tree or graph structures. It also has some disadvantages as it is verbose and access to the data is slow due to parsing and text conversion (Bourett, 2002)

From the paragraph taken above, a simple deduction can be made that XML and its surrounding technologies constitute a simple database format, or rather like the DBMS format. Some of its advantages are the ability to provide data storage and retrieval in XML document format, data schemas via DTDs and XML schemas, query language via XQuery, XPath, XQL, XML -QL and so on, programming interfaces via DOM and SAX (Bourret, 2002). On the other hand, its advantages are lack of features that are vital in any production environment such as efficient data storage and retrieval, indexes, security, and transactions, data integrity, performance, multi-user accesses and lastly performing complex queries across multiple XML storage. Therefore, it has to be coexisting with RDBMS to complement each other.

XML vs. HTML

In HTML, both the tag semantics and the tag set are fixed. An <h1> is always a first level heading and the tag <ati.product.code> is meaningless. The W3C, in conjunction with browser vendors and the WWW community, is constantly working to extend the definition of HTML to allow new tags to keep pace with changing technology and to bring variations in presentation (stylesheets) to the Web. However, these changes are always rigidly confined by what the browser vendors have implemented and by the fact that backward compatibility is paramount. And for people who want to disseminate information widely, features supported by only the latest releases of Netscape and Internet Explorer are not useful.

XML specifies neither semantics nor a tag set. In fact XML is really a meta-language for describing markup languages. In other words, XML provides a facility to define tags and the structural relationships between them. Since there's no predefined tag set, there can't be any preconceived semantics. All of the semantics of an XML document will either be defined by the applications that process them or by stylesheets

XML vs. SGML

XML is defined as an application profile of SGML. SGML is the Standard Generalized Markup Language defined by ISO 8879. SGML has been the standard, vendor-independent way to maintain repositories of structured documentation for more than a decade, but it is not well suited to serving documents over the web (for a number of technical reasons beyond the scope of this article). Defining XML as an application profile of SGML means that any fully conformant SGML system will be able to read XML documents. However, using and understanding XML documents does not require a system that is capable of understanding the full generality of SGML. XML is, roughly speaking, a restricted form of SGML.

For technical purists, it's important to note that there may also be subtle differences between documents as understood by XML systems and those same documents as understood by SGML systems. In particular, treatment of white space immediately adjacent to tags may be different.

Why XML?

In order to appreciate XML, it is important to understand why it was created. XML was created so that richly structured documents could be used over the web. The only viable alternatives, HTML and SGML, are not practical for this purpose. HTML, as we've already discussed, comes bound with a set of semantics and does not provide arbitrary structure. SGML provides arbitrary structure, but is too difficult to implement just for a web browser. Full SGML systems solve large, complex problems that justify their expense. Viewing structured documents sent over the web rarely carries such justification.

This is not to say that XML can be expected to completely replace SGML. While XML is being designed to deliver structured content over the web, some of the very features it lacks to make this practical, make SGML a more satisfactory solution for the creation and long-time storage of complex documents. In many organizations, filtering SGML to XML will be the standard procedure for web delivery.

XML Development Goals

1. It shall be straightforward to use XML over the Internet. Users must be able to view XML documents as quickly and easily as HTML documents. In practice, this will only be possible when XML browsers are as robust and widely available as HTML browsers, but the principle remains.
2. XML shall support a wide variety of applications. XML should be beneficial to a wide variety of diverse applications: authoring, browsing, content

analysis, etc. Although the initial focus is on serving structured documents over the web, it is not meant to narrowly define XML.

3. XML shall be compatible with SGML. Most of the people involved in the XML effort come from organizations that have a large, in some cases staggering, amount of material in SGML. XML was designed pragmatically, to be compatible with existing standards while solving the relatively new problem of sending richly structured documents over the web.
4. It shall be easy to write programs that process XML documents. The colloquial way of expressing this goal while the spec was being developed was that it ought to take about two weeks for a competent computer science graduate student to build a program that can process XML documents.
5. The number of optional features in XML is to be kept to an absolute minimum, ideally zero. Optional features inevitably raise compatibility problems when users want to share documents and sometimes lead to confusion and frustration.
6. XML documents should be human-legible and reasonably clear. If you don't have an XML browser and you've received a hunk of XML from somewhere, you ought to be able to look at it in your favorite text editor and actually figure out what the content means.
7. The XML design should be prepared quickly. Standards efforts are notoriously slow. XML was needed immediately and was developed as quickly as possible.

8. The design of XML shall be formal and concise. In many ways a corollary to rule 4, it essentially means that XML must be expressed in EBNF and must be amenable to modern compiler tools and techniques. There are a number of technical reasons why the SGML grammar cannot be expressed in EBNF. Writing a proper SGML parser requires handling a variety of rarely used and difficult to parse language features. XML does not.
9. XML documents shall be easy to create. Although there will eventually be sophisticated editors to create and edit XML content, they won't appear immediately. In the interim, it must be possible to create XML documents in other ways: directly in a text editor, with simple shell and Perl scripts, etc.
10. Terseness in XML markup is of minimal importance. Several SGML language features were designed to minimize the amount of typing required to manually key in SGML documents. These features are not supported in XML. From an abstract point of view, these documents are indistinguishable from their more fully specified forms, but supporting these features adds a considerable burden to the SGML parser (or the person writing it, anyway). In addition, most modern editors offer better facilities to define shortcuts when entering text.

4.4.7 Macromedia Dreamweaver MX

Dreamweaver lets you create sites and applications the way you want to -- by hand-coding, visual design, or a combination of the two, and with your choice of server technologies. Dreamweaver MX 2004 expands your options and makes it even easier to accomplish your web and application development goals. Dreamweaver MX 2004 is the professional choice for building web sites and applications. It provides a powerful combination of visual layout tools, application development features, and code editing support. With robust features for CSS-based design and integration, Dreamweaver enables web designers and developers to easily create and manage any website.

Features:

i) State of the art design tools

ii) Use a world-class design and code editor in one tool.

iii) Build sophisticated, standards-based sites with rich CSS support.

iv) Improve end-user experience.

v) Save time with a built-in graphics editor.

vi) Create better user interfaces.

vii) Powerful and open integration

viii) Use one integrated development environment.

ix) Leverage Secure FTP.

x) Seamlessly integrate external files and code.

xi) Enhance productivity .

xii) Create accessible websites

xiii) Streamlined development

xiv) Manage the basics better .

xv) Code more efficiently

xvi) Find things faster

xvii) Jumpstart design and production

4.5 Summary

In practice, it may be difficult to separate functional and non-functional considerations and to identify requirements which relate to the system as whole. In reality, the distinction between these different types of requirement is not as clear cut as these simple definitions suggest. It is also necessary to determine what kind of software or technology to be applied to fulfill the system's requirement. However, when developed in more detailed, it may lead to other requirement that is clearer and more relevant.

CHAPTER 5 SYSTEM DESIGN

5.0 Introduction

System design is the phase in which the requirements analyzed and produced in the system analysis phase are translated and converted into a representation characteristics of the proposed system. The design process involves developing several models of the system at different level of abstraction. As a design is composed, errors and omission in earlier stages are discovered. Generally, this phase will be focusing on architectural design, functional design, database design and the interface design.

5.1 Architectural Design

Large system can be decomposed into sub-system that provides some related set of services. The initial design process of identifying those sub-systems and establishing a framework for sub-system control and communication is called architectural design. It usually comes before the detailed system specification. (Figure 5.1)

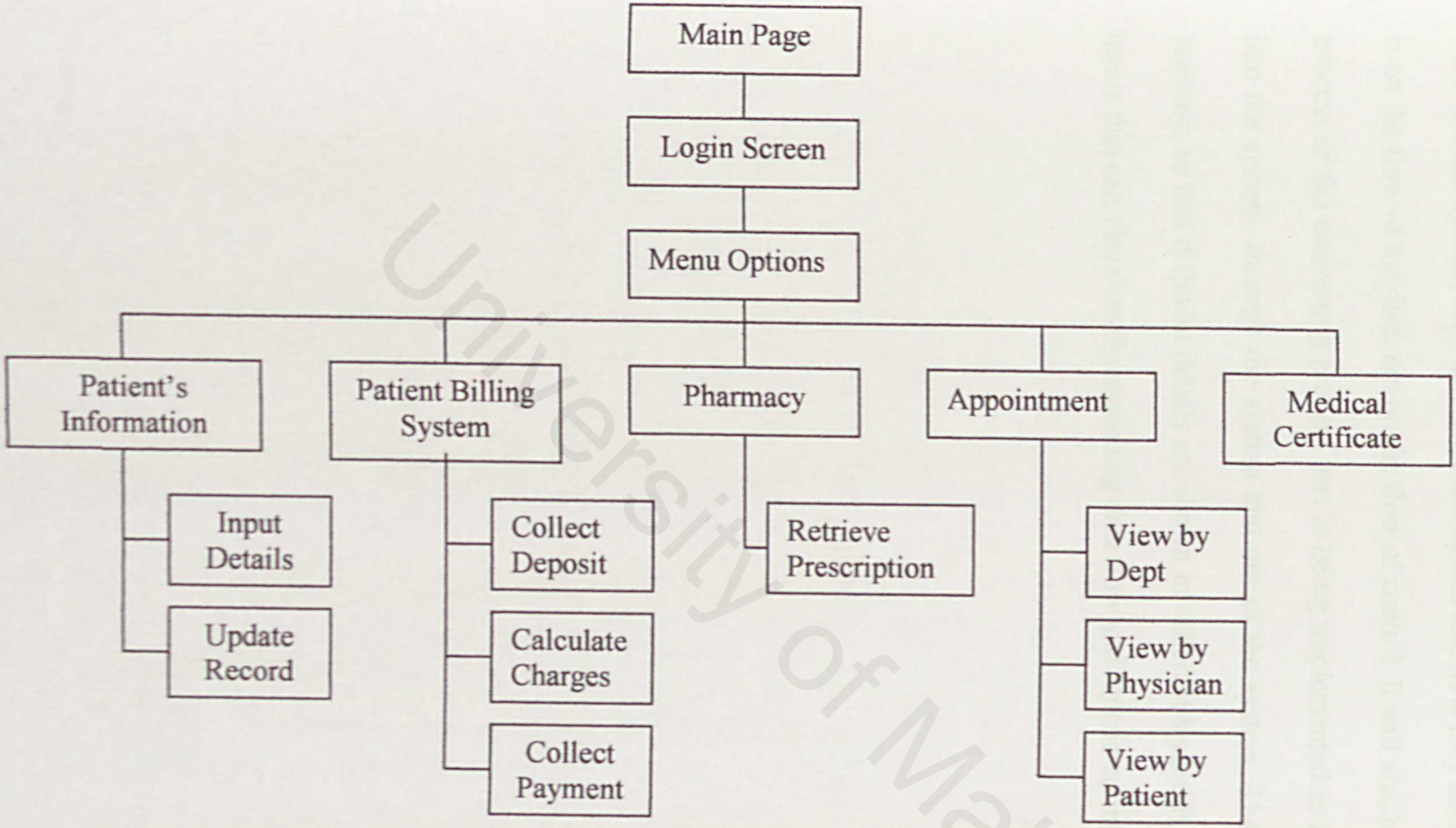

Figure 5.1: Architectural View of Electronic Medical Record

5.2 Process Design

Process design will be visualize using Data Flow Diagram. It is a graphical technique that will display the data flow in the system. As a transformer of data, the diagram shows the data flow into the system, how they are transformed and how they leave the system. The emphasis is on the flow of the data, not on the flow of control. It will also be able to view the changing process or the converting process that is being implemented to the data once the data goes into the system, through the system and out of the system. The hierarchy is expressed by layering, so that different details are shown in different layers. It can be divided into several layers that can visualize the ascending data flow and functional information. (Figure 5.2)

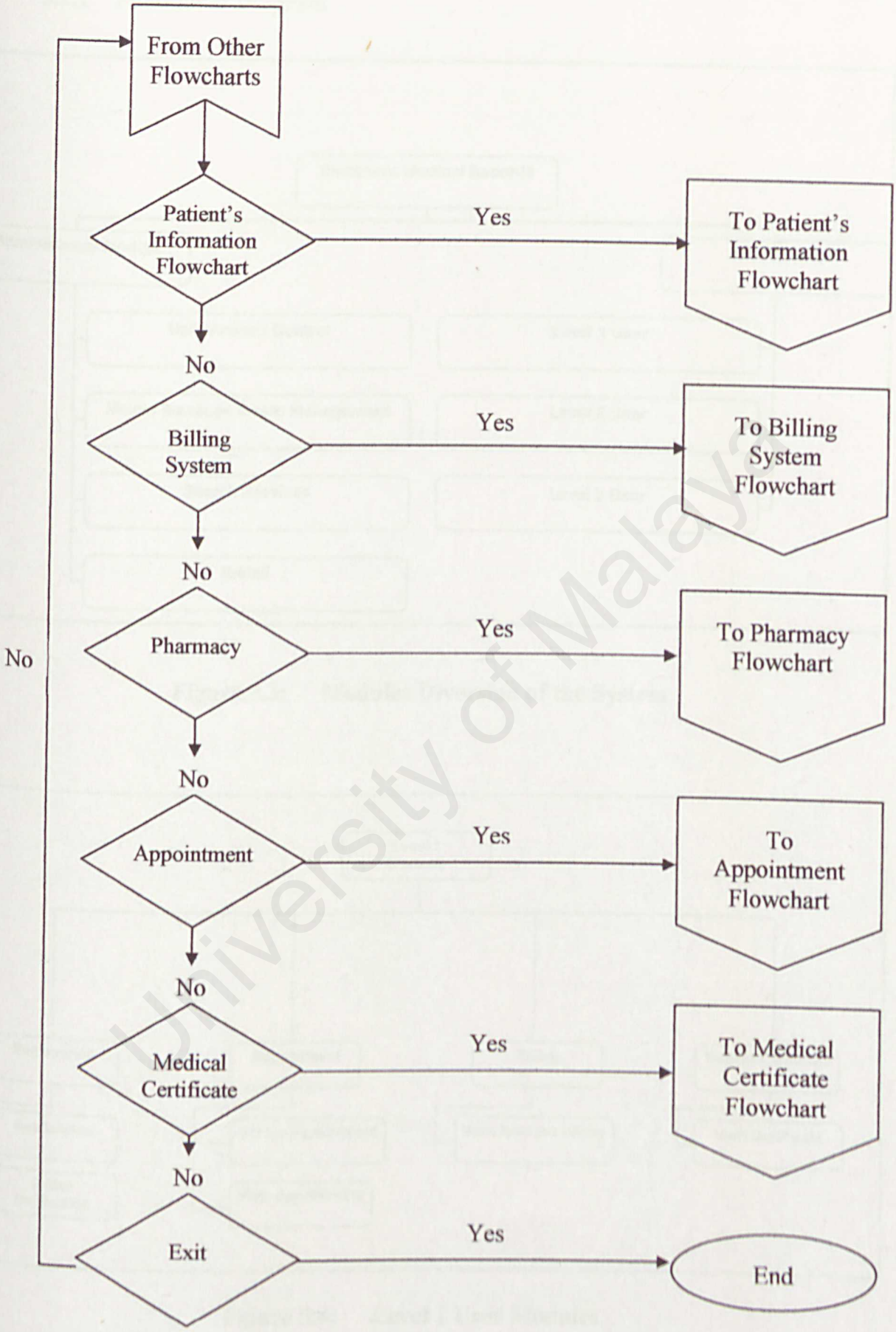

Figure 5.2: Main Screen Flow of Electronic Medical Record

5.2.1 Functional Diagram

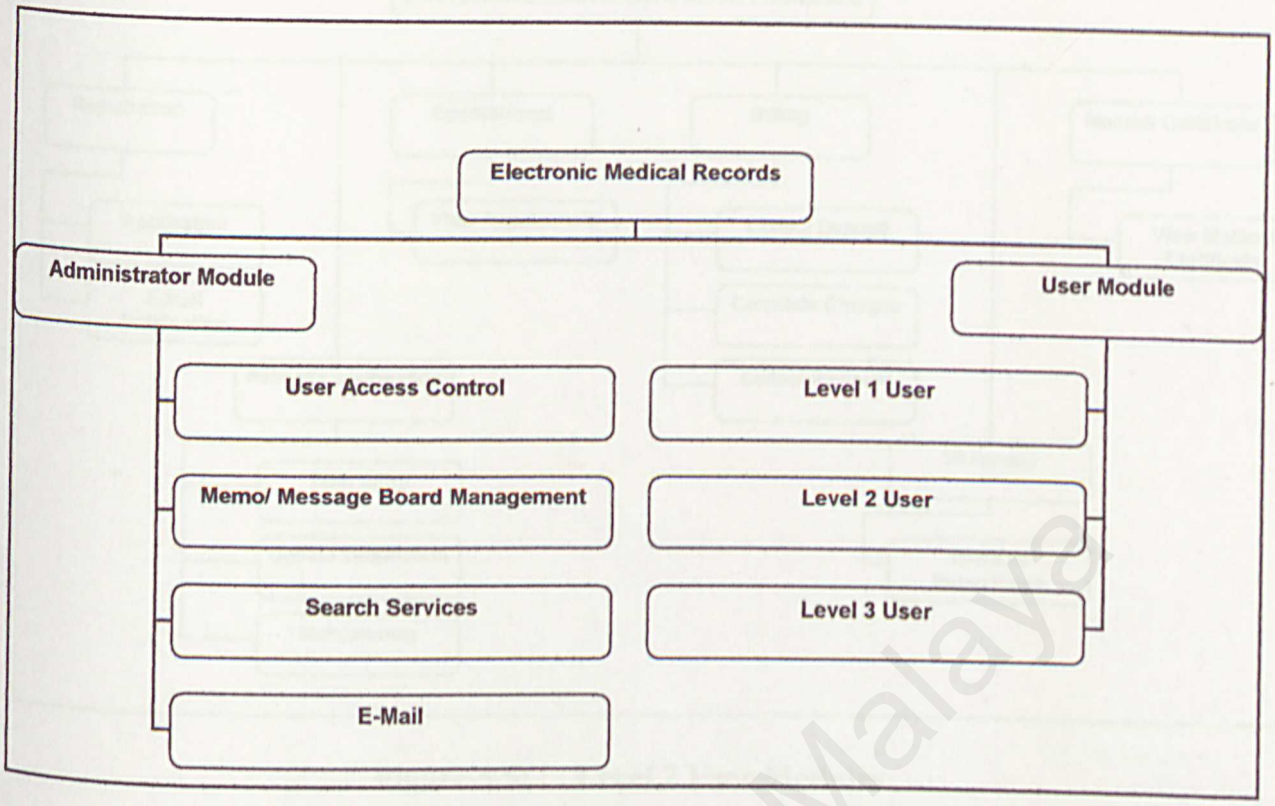

Figure 5.3: Modules Diversion of the System

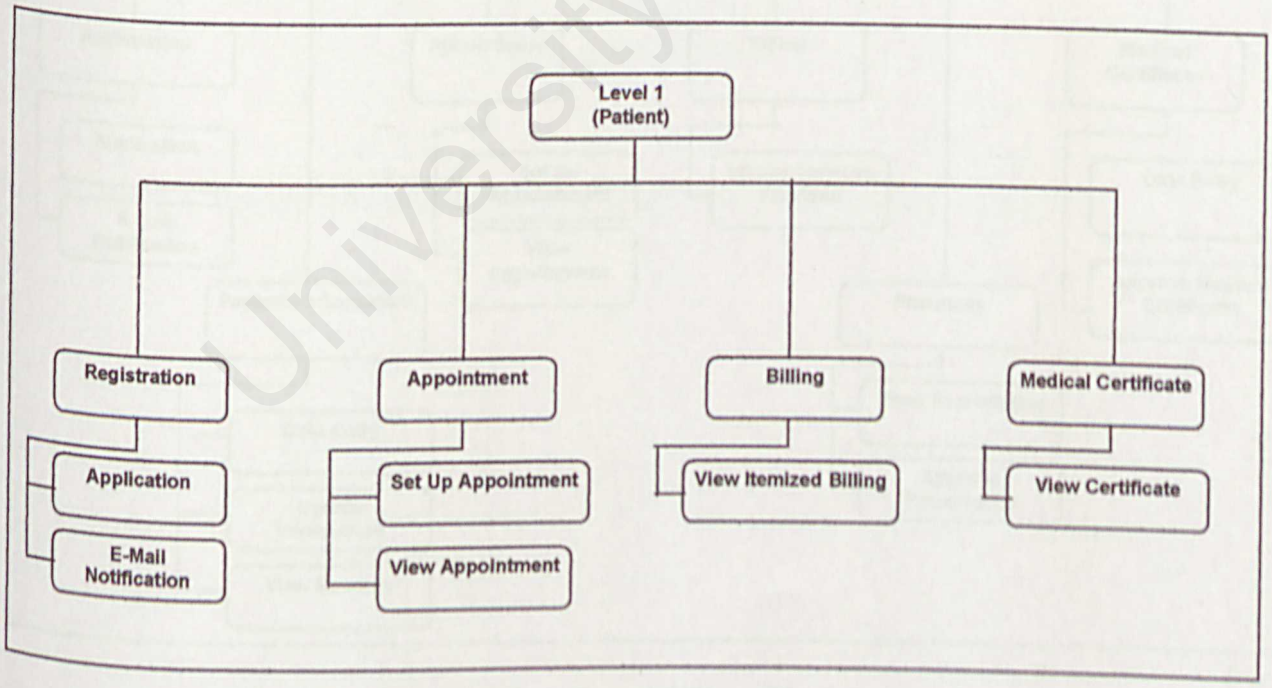

Figure 5.4: Level 1 User Modules

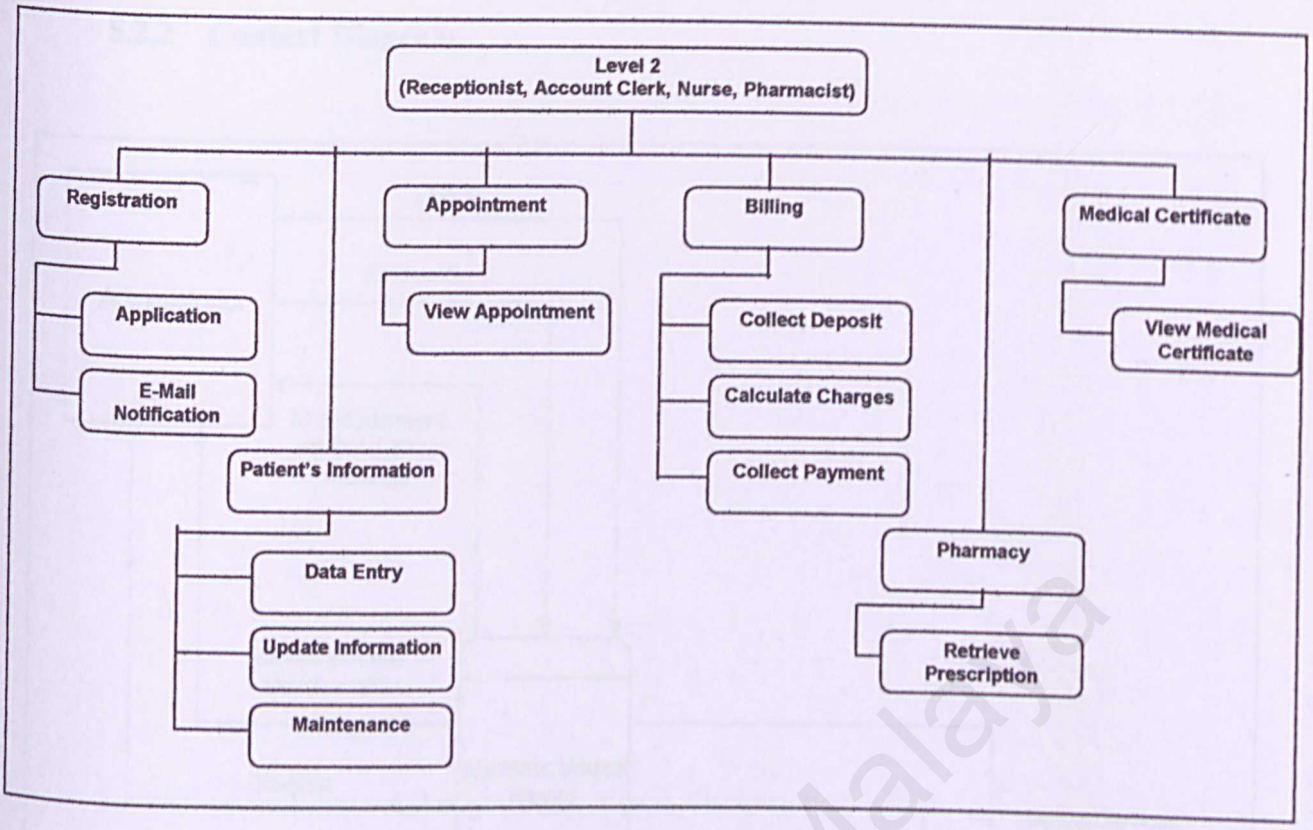

Figure 5.5: Level 2 User Modules

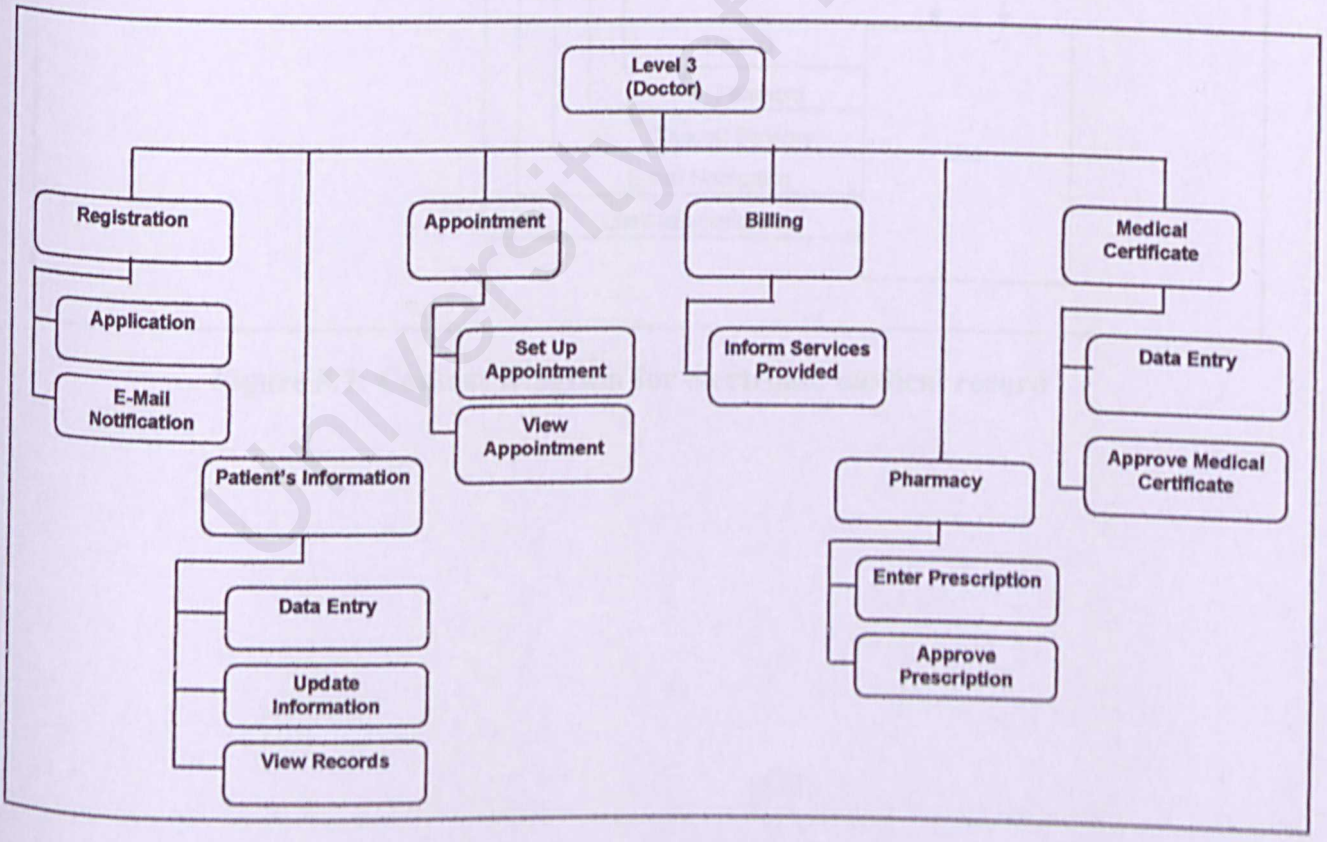

Figure 5.6: Level 3 User Module

5.2.2 Context Diagram

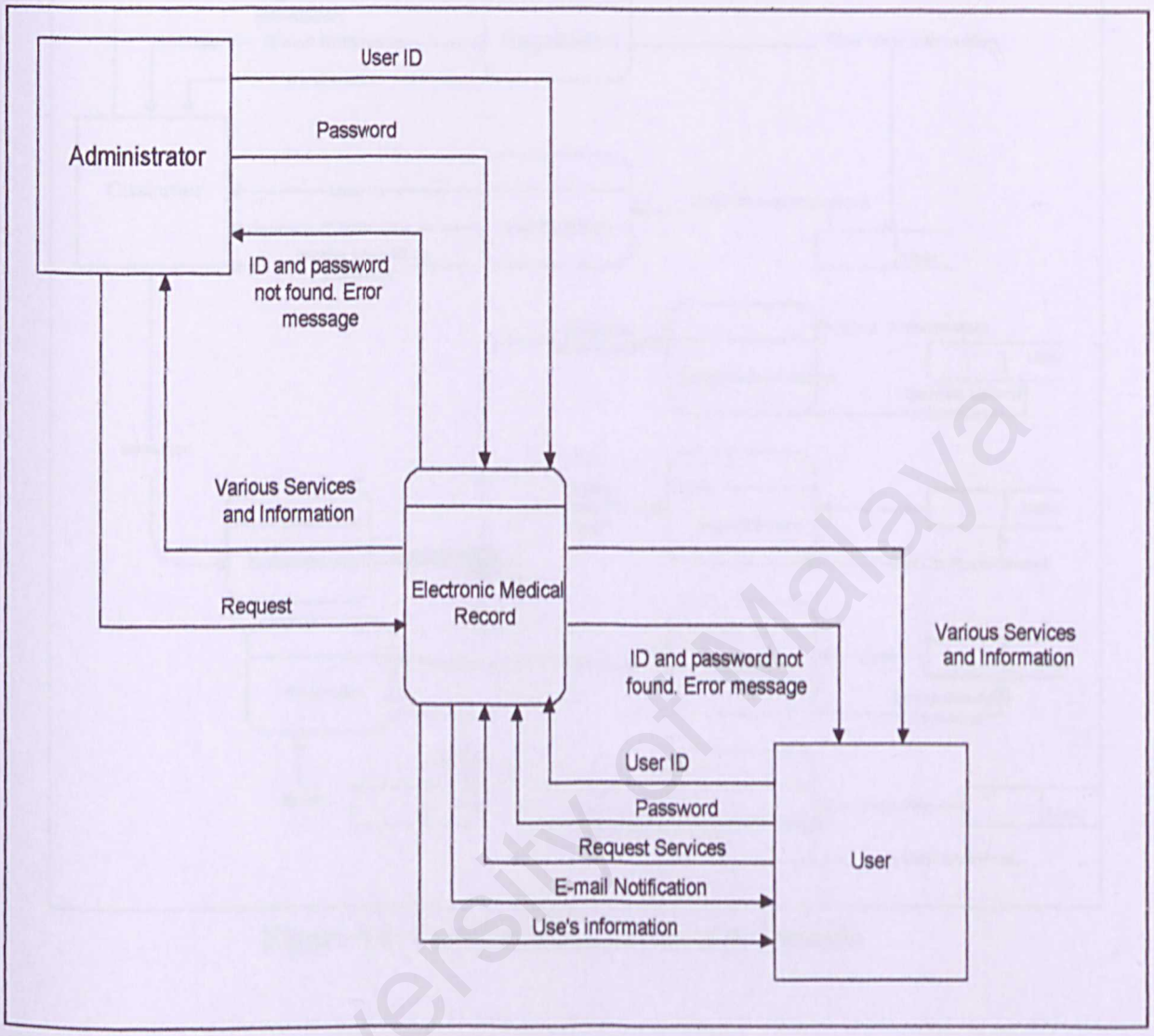

Figure 5.7: Context Diagram for electronic medical record

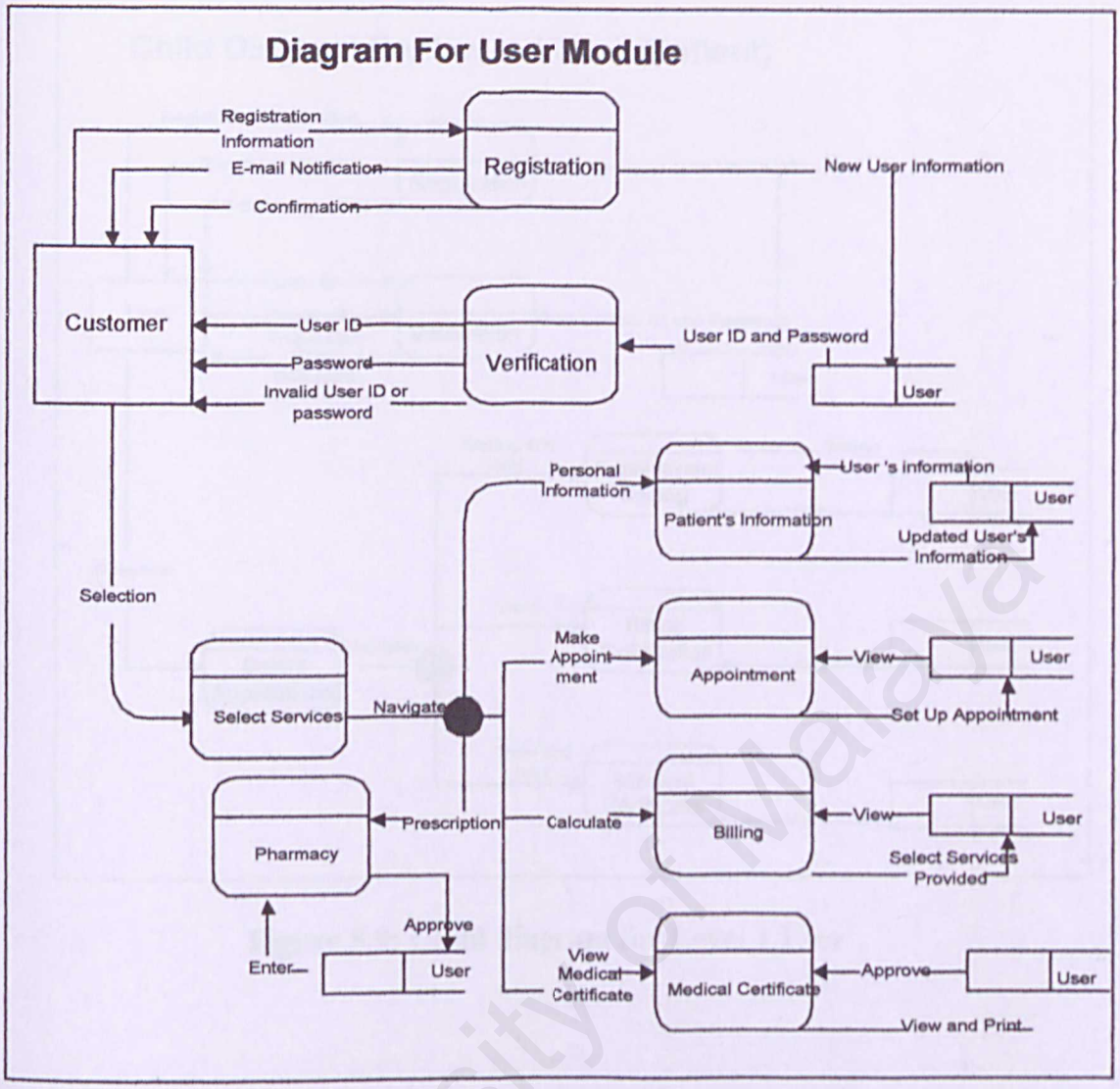

Figure 5.8: Context Diagram for user module

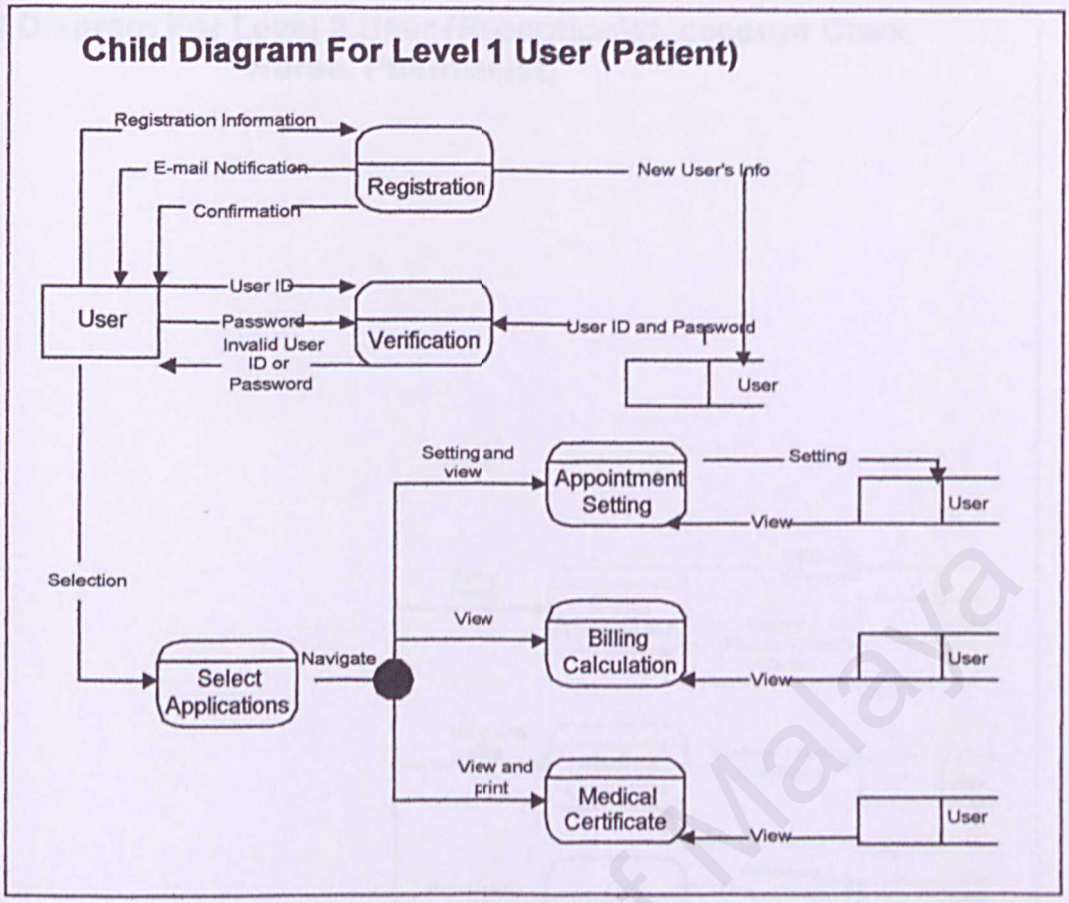

Figure 5.9: Child diagram for Level 1 User

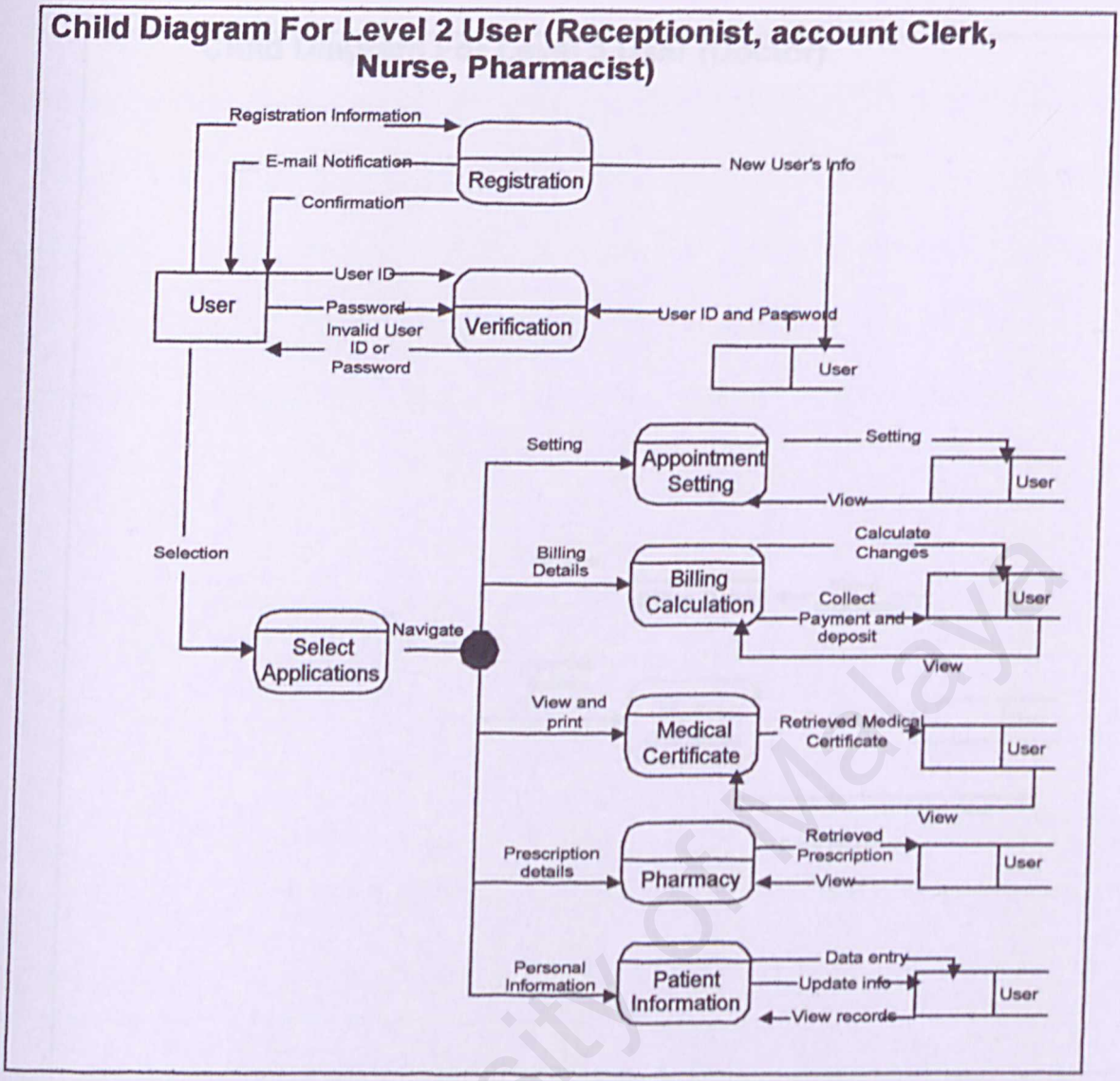

Figure 5.10: Child diagram for Level 2 User

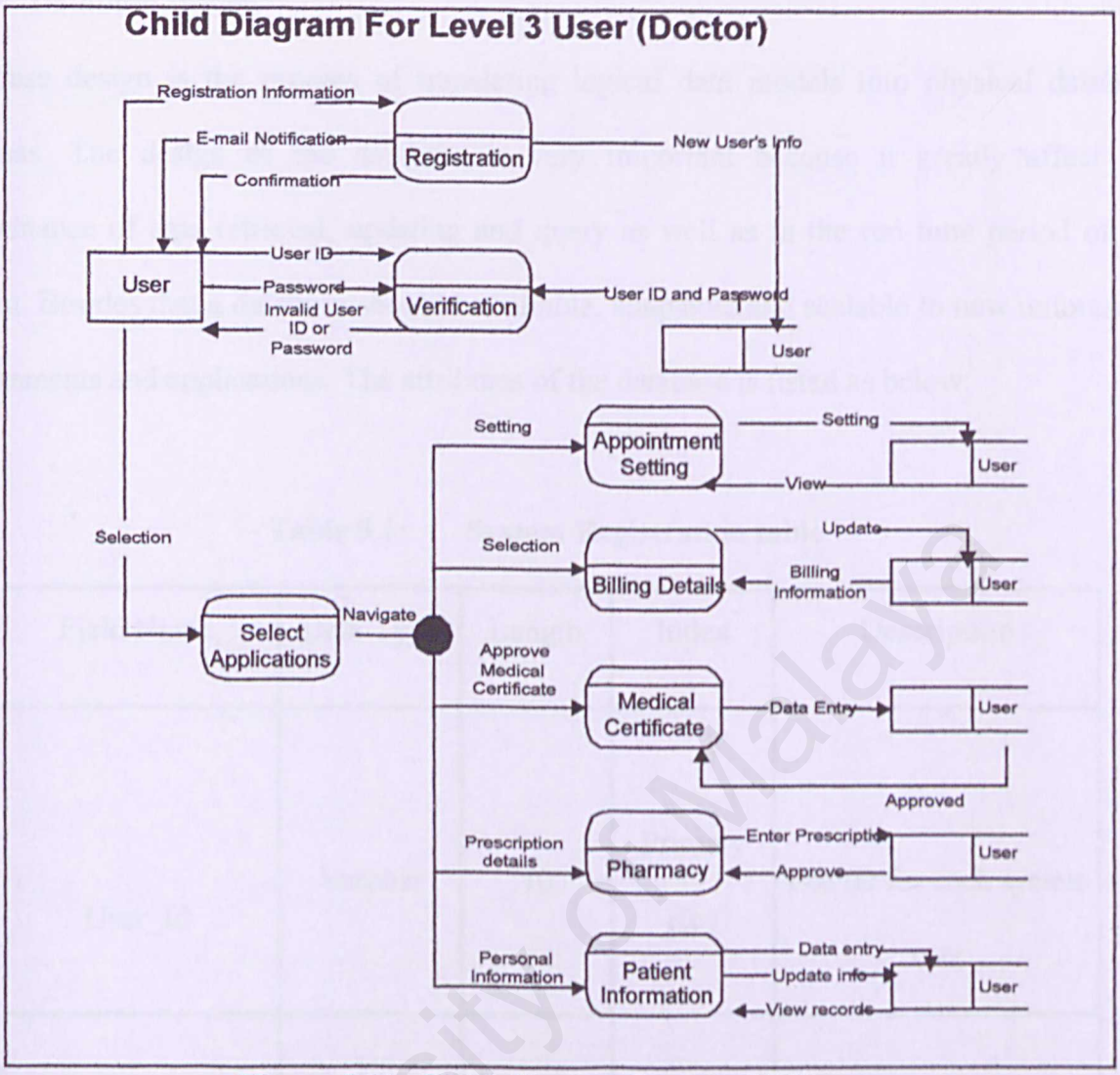

Figure 5.11: Child diagram for Level 3 User

5.3 Database Design

Database design is the process of translating logical data models into physical database schemas. The design of the database is very important because it greatly affect the performance of data retrieval, updating and query as well as in the run time period of the system. Besides that a database should be reliable, adaptable and scalable to new unforeseen requirements and applications. The attributes of the database is listed as below:

Table 5.1: System Registration table

Field Name	Data Type	Length	Index	Description
User_Id	Varchar	10	Primary Key	The ID for each system user
Login_Password	Varchar	20		System user's login password

Table 5.2: The Patient information table

Field Name	Data Type	Length	Index	Description
Patient_Name	Varchar	150		Patient's Name
Patient_IC	Varchar	20		Patient's IC
Patient_Id	Varchar	20	Primary key	Unique ID that represent each patient
DOB	Date/time	10		Patient's Date of Birth
Age	Int	4		Patient's age
Address	Text	500		Patient's address
OfficePhone	Varchar	20		Patient's Office Phone
Homephone	Varchar	20		Patient's homephone
Mobilephone	Varchar	20		Patient's mobilephone
Postal_code	Varchar	20		Postcode
State	Varchar	50		State
Pager	Varchar	50		Pager
Fax	Varchar	50		Fax
E-mail	Varchar	30		Patient's e-mail
Occupation	Varchar	50		Patient's occupation

Table 5.3: Billing Statement table

Field Name	Data Type	Length	Index	Description
PaymentNo	Varchar	20	Primary key	Unique ID that represent each policy
Balance	Varchar	20		Balance of payment
Patient_Name	Varchar	150		Patient's name
Payer	Varchar	150		Payer's name
Department	Varchar	50		Department's name
Bank_Code	Varchar	25		Bank Code number
Change	Varchar	20		Change of payment
RefNo	Varchar	25		Patient's reference Number
Date	Date/Time	10		Date of payment
Remark	Varchar	500		Patient's remark

Table 5.4: Appointment form table

Field Name	Data Type	Length	Index	Description
Patient_Id	Varchar	10	Primary key	Unique ID that represent each patient
Patient_Name	Varchar	150		Patient's name
Date_appointment	Date/time	10		Date of appointment
Time_appointment	Date/time	10		Time of appointment
Doctor_Id	Varchar	10		Doctor's ID
Doctor_Name	Varchar	150		Doctor's name
Remarks	Varchar	500		Remarks of doctor or patient

5.4 User Interface Design

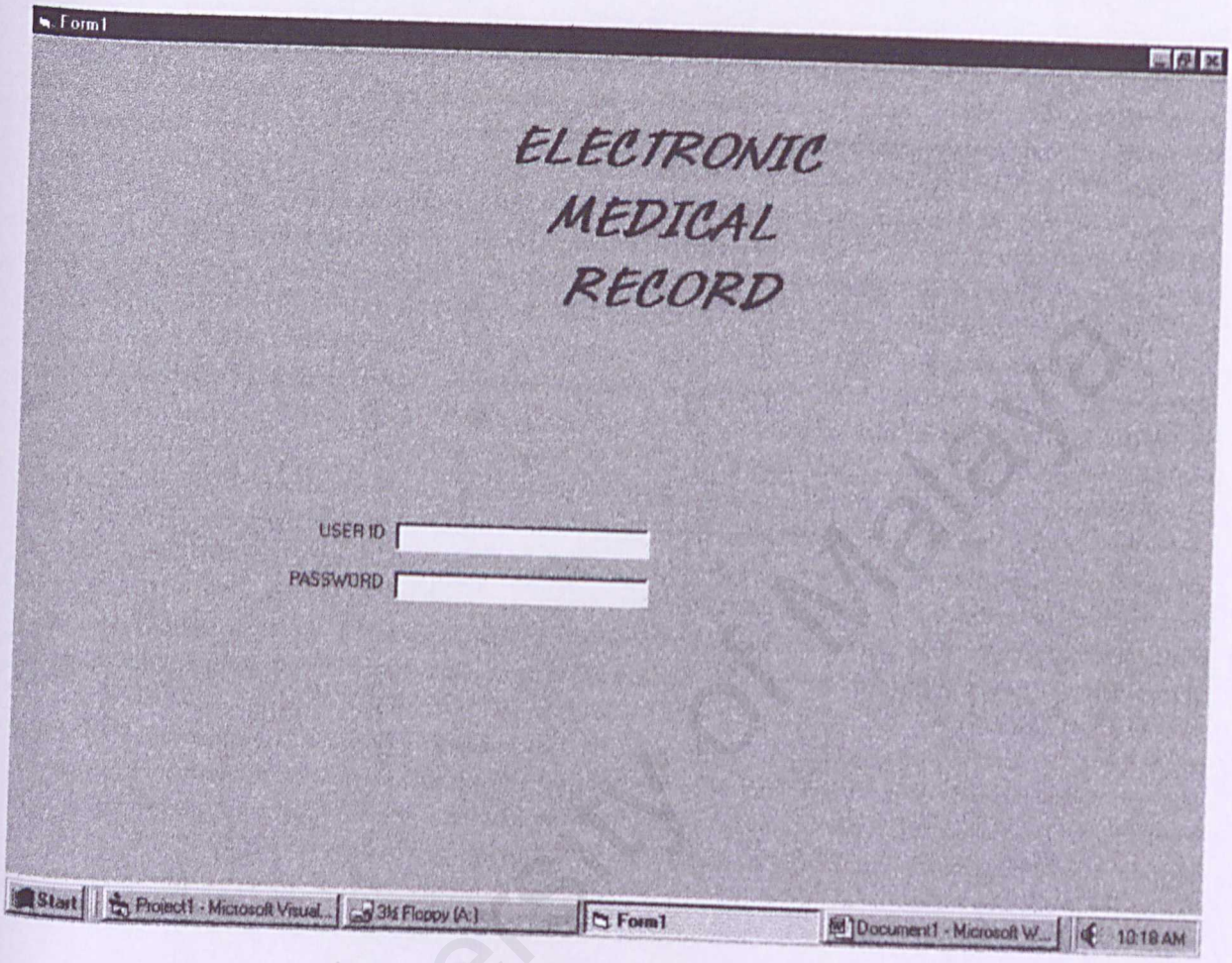

Figure 5.3: Login Page

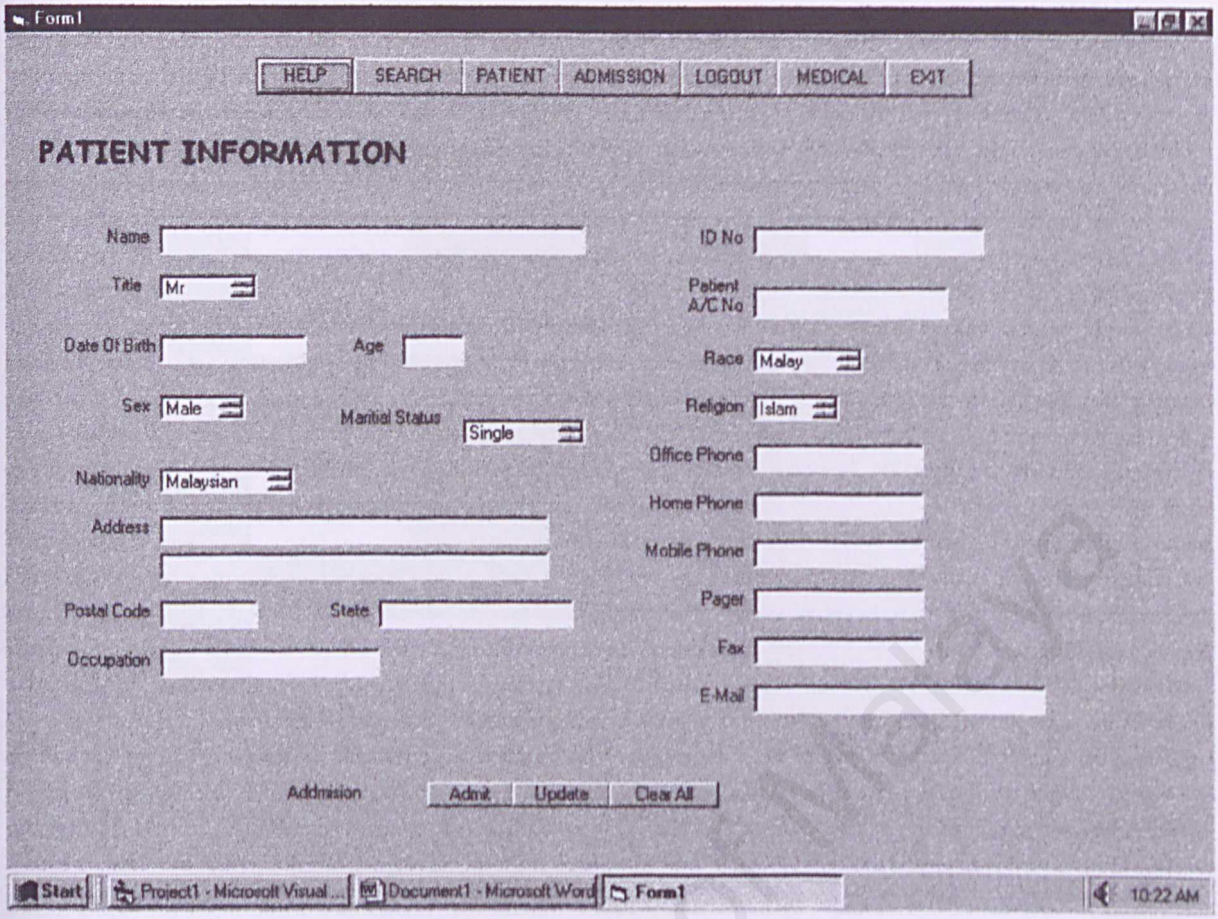

Figure 5.4: Patient's Information Form

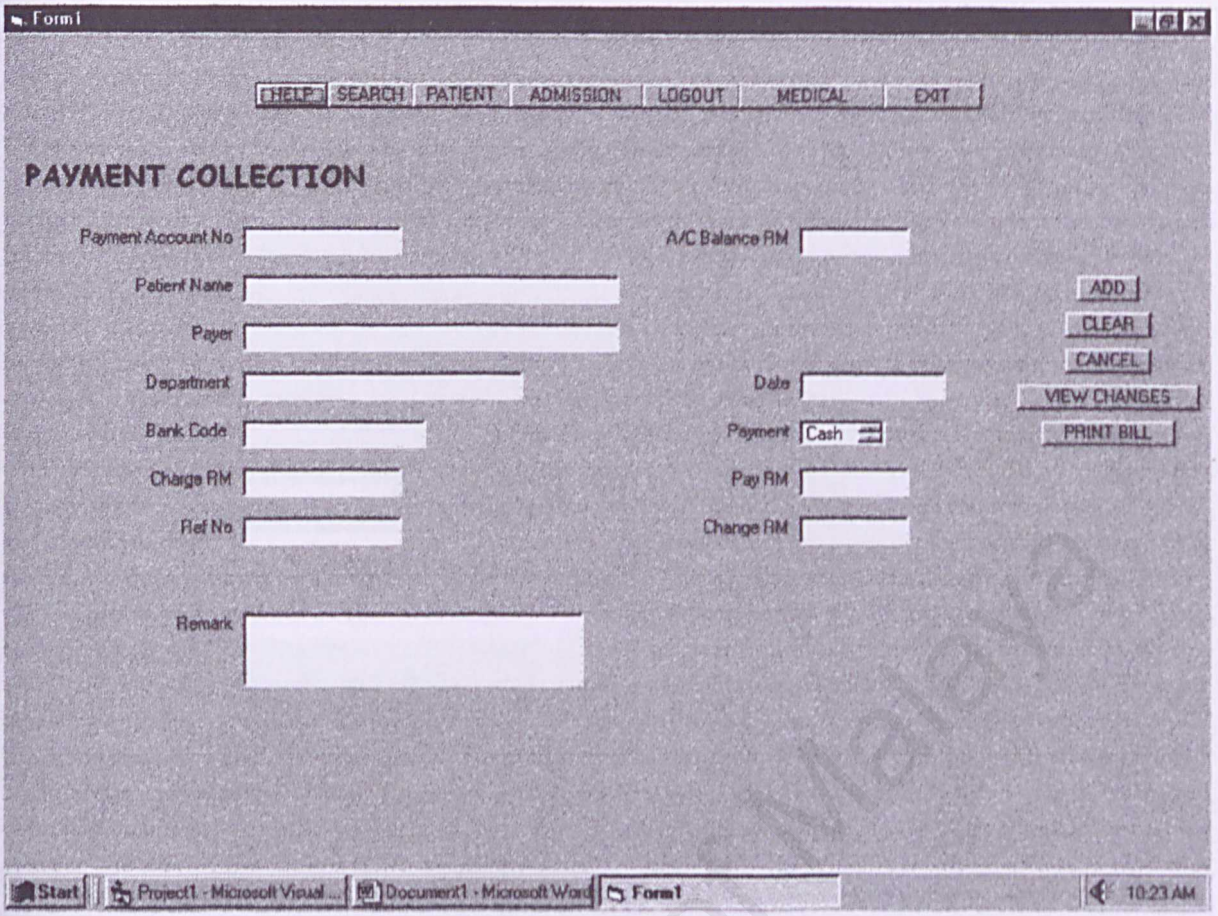

Figure 5.5: Billing Statement

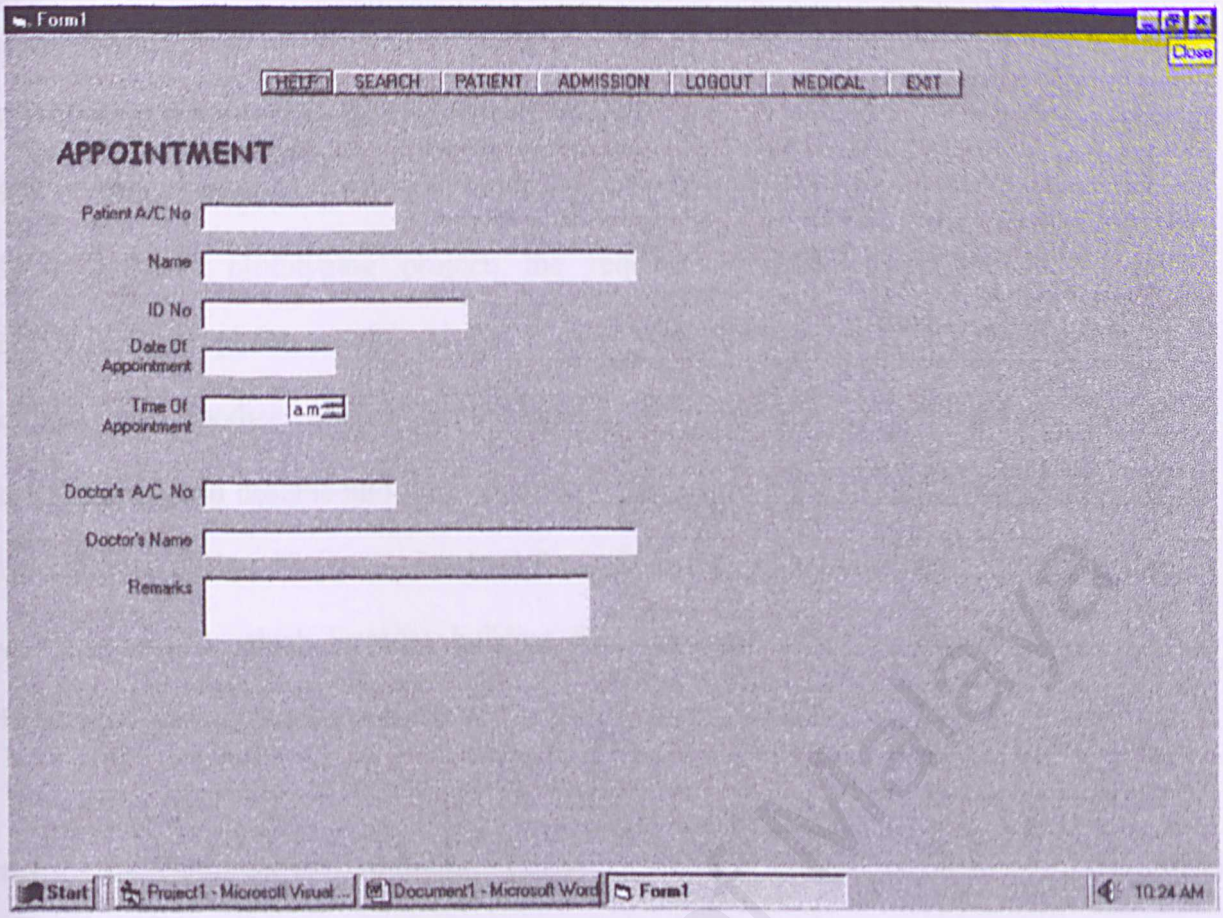

Figure 5.6: Appointment Form

5.5 Summary

The purpose of the design phase is to transform requirements statement from the requirement analysis phase into design specifications for construction. In other words, the design phase addresses how technology will be used in the system. The design phase is concerned with technology-based vies of the system's DATA, PROCESSES, and INTERFACES.

CHAPTER 6 SYSTEM IMPLEMENTATION

6.0 Introduction

In a software prototyping project; the requirement analysis, system design and implementation phase usually do not have a clear boundaries. Each phase tends to overlap one another. System implementation is a process that converts the system requirements and designs into program codes. This phase involves some modifications on the previous design. Electronic Medical Records also developed modularly using the top-down approach, which includes building the high-level software modules that refined further into functions and procedures.

6.1 Development Environment

It is certainly gave an impact on the system's development but helped in determining the success of a project. After implementing the system, the requirement of hardware and software were truly defined and finalized. Some changes have been made beyond thoroughly considerations.

6.1.1 Actual Hardware Requirements

- Pentium III 400MHz computer and above
- 20 GB Hard Disk Space
- At least 128 MB RAM
- Other standard computer peripherals
- Standard Input/Output Devices

6.1.2 Actual Software Requirements

- Extensible Markup Language (XML) 1.0
- Microsoft Access 2000 – Database repository
- Microsoft Windows XP Professional- Operating System
- Microsoft Internet Explorer 5.0 –Browser
- Internet Information Service 6.0 - Web Server
- Macromedia Dreamweaver MX 2004 – Web Design Editor
- Active Server Pages 3.0 – Programming Coding
- Adobe Photoshop 7.0 – Photo and Image Editor

6.2 Active Server Pages Coding (ASP)

An ASP file normally contains HTML tags, just like an HTML file. However, an ASP file can also contain **server scripts**, surrounded by the delimiters **<%** and **%>**. Server scripts are **executed on the server,** and can contain any expressions, statements, procedures, or operators valid for the scripting language you prefer to use. ASP is the main programming coding used in the development of Electronic Medical Records.

6.2.1 Object Usage in ASP

a) Response Object

The ASP Response Object is used to the user from the server. Its collections, properties, and methods are described as below:

Collections

Collection	Description
Cookies	Sets a cookie value. If the cookie does not exist, it will be created, and take the value that is specified

Properties

Property	Description
Buffer	Specifies whether to buffer the page output or not
CacheControl	Sets whether a proxy server can cache the output generated by ASP or not
Charset	Appends the name of a character-set to the content-type header in the Response object
ContentType	Sets the HTTP content type for the Response object
Expires	Sets how long (in minutes) a page will be cached on a browser before it expires
ExpiresAbsolute	Sets a date and time when a page cached on a browser will expire
IsClientConnected	Indicates if the client has disconnected from the server
Pics	Appends a value to the PICS label response header
Status	Specifies the value of the status line returned by the server

Methods

Method	Description
AddHeader	Adds a new HTTP header and a value to the HTTP response
AppendToLog	Adds a string to the end of the server log entry
BinaryWrite	Writes data directly to the output without any character conversion
Clear	Clears any buffered HTML output
End	Stops processing a script, and returns the current result
Flush	Sends buffered HTML output immediately
Redirect	Redirects the user to a different URL
Write	Writes a specified string to the output

b) Session Object

The Session object is used to store information about, or change settings for a user session. Variables stored in the Session object hold information about one single user, and are available to all pages in one application.

Examples

Set and return the LCID

This example demonstrates the "LCID" property. This property sets or returns an integer that specifies a location or region. Contents like date, time, and currency will be displayed according to that location or region.

Return the SessionID

This example demonstrates the "SessionID" property. This property returns a unique id for each user. The id is generated by the server.

A session's timeout

This example demonstrates the "Timeout" property. This example sets and returns the timeout (in minutes) for the session.

Collections

Collection	Description
Contents	Contains all the items appended to the session through a script command
StaticObjects	Contains all the objects appended to the session with the HTML <object> tag

Properties

Property	Description
CodePage	Specifies the character set that will be used when displaying dynamic content
LCID	Sets or returns an integer that specifies a location or region. Contents like date, time, and currency will be displayed according to that location or region
SessionID	Returns a unique id for each user. The unique id is generated by the server
Timeout	Sets or returns the timeout period (in minutes) for the Session object in this application

Methods

Method	Description
Abandon	Destroys a user session
Contents.Remove	Deletes an item from the Contents collection
Contents.RemoveAll()	Deletes all items from the Contents collection

Events

Event	Description
Session_OnEnd	Occurs when a session ends
Session_OnStart	Occurs when a session starts

c) Server Object

The ASP Server object is used to access properties and methods on the server. The ASP Server object is used to access properties and methods on the server. Its properties and methods are described below:

Properties

Property	Description
ScriptTimeout	Sets or returns the maximum number of seconds a script can run before it is terminated

Methods

Method	Description
CreateObject	Creates an instance of an object
Execute	Executes an ASP file from inside another ASP file
GetLastError()	Returns an ASPError object that describes the error condition that occurred
HTMLEncode	Applies HTML encoding to a specified string
MapPath	Maps a specified path to a physical path
Transfer	Sends (transfers) all the information created in one ASP file to a second ASP file
URLEncode	Applies URL encoding rules to a specified string

The CreateObject Method

The CreateObject method creates an instance of an object.

Note: Objects created with this method have page scope. They are destroyed when the server are finished processing the current ASP page. To create an object with session or application scope, you can either use the <object> tag in the Global.asa file, or store the object in a session or application variable.

Syntax

```
Server.CreateObject(progID)
```

Part	Description
progID	Required. The type of object to create

The MapPath Method

The MapPath method maps a specified path to a physical path.

Note: This method cannot be used in Session.OnEnd and Application.OnEnd.

Syntax

```
Server.MapPath(path)
```

Parameter	Description
path	Required. A relative or virtual path to map to a physical path. If this parameter starts with / or \, it returns a path as if this parameter is a full virtual path. If this parameter doesn't start with / or \, it returns a path relative to the directory of the .asp file being processed

d) Application Object

An application on the Web may be a group of ASP files. The ASP files work together to perform some purpose. The Application object in ASP is used to tie these files together. The Application object is used to store and access variables from any page, just like the Session object. The difference is that ALL users share one Application object, while with Sessions there is one Session object for EACH user. The Application object should hold information that will be used by many pages in the application (like database connection information). This means that you can access the information from any page. It also means that you can change the information in one place and the changes will automatically be reflected on all pages. The Application object's collections, methods, and events are described below:

Collections

Collection	Description
Contents	Contains all the items appended to the application through a script command
StaticObjects	Contains all the objects appended to the application with the HTML <object> tag

Methods

Method	Description
Contents.Remove	Deletes an item from the Contents collection
Contents.RemoveAll()	Deletes all items from the Contents collection
Lock	Prevents other users from modifying the variables in the Application object
Unlock	Enables other users to modify the variables in the Application object (after it has been locked using the Lock method)

Events

Event	Description
Application_OnEnd	Occurs when all user sessions are over, and the application ends
Application_OnStart	Occurs before the first new session is created (when the Application object is first referenced)

6.2.2 Main Application Using ASP

a) Clear The Buffer

```
<%
Response.Expires = -1000 'Makes the browser not cache this page
Response.Buffer = True 'Buffers the content so our Response.Redirect will
work

If Session("doctorLoggedIn") = "true" Then
   Response.Redirect("doctormain.asp")
End If
%>
```

b) To check login validity

```
Sub CheckLogin
username = trim(Request.Form("username"))
userpwd = trim(Request.Form("userpwd"))
```

c) Auto-generate ID number

```
set con=Server.CreateObject("ADODB.Connection")
     sConnString = "DRIVER={Microsoft Access Driver (*.mdb)};" & _
          "DBQ=" & Server.MapPath("dbHospital.mdb") & ";"

        con.Open(sConnString)

              set maxPatRs=Server.CreateObject("ADODB.Recordset")
              maxPatRs.open "Select max(idsufix) from
tblPatients",con,1,3

              if isnull(maxpatrs(0)) then
                varSufix=1
              else
                varSufix=maxpatrs(0)+1
              end if

              varPrefix="P" & right(year(date),2)

              varNewPatId= varPrefix & varSufix
```

d) To prompt for not filling all fields

```
if trim(LCase(Request.Form("username")))="" or
trim(LCase(Request.Form("userpwd")))=""  then
   response.write "<font color=red>Please fill all fields</font>"
    ShowLogin
   exit sub
end if
```

e) Checking password table for password validity

```
set con=Server.CreateObject("ADODB.Connection")
     sConnString = "DRIVER={Microsoft Access Driver (*.mdb)};" & _
          "DBQ=" & Server.MapPath("dbHospital.mdb") & ";"

        con.Open(sConnString)

        set loginrs=Server.CreateObject("ADODB.Recordset")
        loginrs.open "Select * from tblDoc where Lcase(docid)='" &
LCase(username) & "'",con,1,3
```

```
if loginrs.recordcount>0 then
  If LCase(Request.Form("username")) = lcase(loginrs("docid")) And
Request.Form("userpwd") = loginrs("docpassword") Then
      Session("doctorLoggedIn") = "true"
      Session("docId")=loginrs("docid")
      Response.Redirect "doctormain.asp"
      loginrs.close
      con.close
  Else
      Response.Write("<font color=red>wrong password Login
Failed.<br><br></font>")
      ShowLogin
  end if
else
      Response.Write("<font color=red>no Dcotor found Login
Failed.<br><br><font>")
      ShowLogin
End If

End Sub
```

f) Change Password

```
response.write "<br><center><b>"

if trim(request.form("oldpass"))="" or trim(request.form("newpass"))=""
or trim(request.form("confpass"))="" then
        response.write "<font color=red>Please fill all
fields</font><br><br><A href='javascript:history.back()'
onclick='history.back();return false'>Go Back</A>"
elseif trim(request.form("newpass"))<> trim(request.form("confpass"))
then
        response.write "<font color=red>Password and Re-type Password
not match</font><br><br><A href='javascript:history.back()'
onclick='history.back();return false'>Go Back</A>"

else
        set con=Server.CreateObject("ADODB.Connection")
    sConnString = "DRIVER={Microsoft Access Driver (*.mdb)};" & _
          "DBQ=" & Server.MapPath("dbHospital.mdb") & ";"

        con.Open(sConnString)

              set checkRs=Server.CreateObject("ADODB.Recordset")
```

```
        checkRs.open "Select * from tblDoc where docid='" &
Session("doId") & "'",con,1,3

        if checkrs.recordcount<0 then
                response.write "<font color=red>Patient not found,
Database error, contact admin</font>"
        elseif checkRs("docpassword")<> trim(request.form("oldpass"))
then
                response.write "<font color=red>Your Old Password is
wrong</font><br><br><A href='javascript:history.back()'
onclick='history.back();return false'>Go Back</A>"
        else
                checkRs("docpassword")=trim(request.form("newpass"))
                checkRs.update
                response.write "<font color=green>Your Password
changed successfully</font>"
        end if
end if

response.write "<br></center></b>"
```

g) To Access Properties and Methods On the Server

```
<%
set con=Server.CreateObject("ADODB.Connection")
     sConnString = "DRIVER={Microsoft Access Driver (*.mdb)};" & _
          "DBQ=" & Server.MapPath("dbHospital.mdb") & ";"
        con.Open(sConnString)

set DocRs=Server.CreateObject("ADODB.Recordset")
        DocRs.open "Select * from tblDoc",con,1,3

        if DocRs.recordcount<0 then
                response.write "<font color=red><b>Sorry, currently, No
Doctor is available</b>"
        else
                response.write "<select name=cmbDoc>"
                do while not docrs.eof
                        response.write "<option value=" & docRs("docid")
& ">" & docRs("docid") & "</option>"
                        docrs.movenext
                loop
        end if
%>
```

h) Appointment Scheduling

```
response.write "<br><center>"
set con=Server.CreateObject("ADODB.Connection")
    sConnString = "DRIVER={Microsoft Access Driver (*.mdb)};" & _
        "DBQ=" & Server.MapPath("dbHospital.mdb") & ";"

    con.Open(sConnString)

    set apptRs=Server.CreateObject("ADODB.Recordset")
    apptRs.open "Select * from tblAppt where docid='" &
Session("docid") & "'",con,3,3

    if apptrs.recordcount <1 then
        response.write "<font color=red><b>No appointments
found</b></font>"
    else
        response.write "<table border=1 cellspacing=0
bordercolor=1><tr><Th>Appointment Id <th>Doctor
Id<th>Date<th>Time<th>Reason<th>Status<th>Action"

        do while not apptrs.eof

          response.write "<tr><td>" & apptrs("apptid")
          response.write "<td>" & apptrs("patid")
          response.write "<td>" & apptrs("adate") & "/" &
apptrs("amonth") & "/" & apptrs("ayear")
          response.write "<td>" & apptrs("atime")
          response.write "<td>" & apptrs("reason")

          if apptrs("apptstatus")=0 then
              response.write "<td>Appointment confirmed"
          elseif apptrs("apptstatus")=1 then
              response.write "<td>Appointment completed"
          elseif apptrs("apptstatus")=2 then
              response.write "<td>Appointment canceled by
              Doctor"
          end if
```

```
        if apptrs("apptstatus")=0 then
                response.write "<td><a
href=docaptcancel.asp?aptid=" & apptrs("apptid") & ">Cancel</a> / <a
href=docaptcomplete.asp?aptid=" & apptrs("apptid") & ">Complete</a>'
                else
                response.write "<td>View"
                end if

                apptrs.movenext
                loop
                response.write "</table>"
        end if
```

i) To Delete An Appointment

```
response.write "<center><Br><BR>"
set con=Server.CreateObject("ADODB.Connection")
    sConnString = "DRIVER={Microsoft Access Driver (*.mdb)};" & _
        "DBQ=" & Server.MapPath("dbHospital.mdb") & ";"

    con.Open(sConnString)

    con.execute "delete from tblAppt where apptid=" &
request.querystring("aptid")

    response.write "<b>Appointment Deleted</b>"
```

j) To Access The XML Files (for admin use ONLY)

```
<%
 Response.Expires = -1000 'Makes the browser not cache this page
Response.Buffer = True 'Buffers the content so our Response.Redirect will
work


' Name of the access db being queried
accessdb="dbHospital.mdb"

' Connection string to the access db
cn="DRIVER={Microsoft Access Driver (*.mdb)};"
cn=cn & "DBQ=" & server.mappath(accessdb)
```

```
' Create a server recordset object
Set rs = Server.CreateObject("ADODB.Recordset")

' Query the states table from the state_info db
sql = "select * from tblPatients "

' Execute the sql
rs.Open sql, cn

' Move to the first record
rs.MoveFirst

' Name for the ouput document
file_being_created= "patients.xml"

' create a file system object
set fso = createobject("scripting.filesystemobject")

' create the text file - true will overwrite any previous files
' Writes the db output to a .xml file in the same directory
Set act = fso.CreateTextFile(server.mappath(file_being_created), true)

' All non repetitive xml on top goes here
act.WriteLine("<?xml version=""1.0""?>")
act.WriteLine("<Patients>")

'Loop to output all the query results to the xml document
do while not rs.eof

' counter to give each record a sequential listing
counter=counter+1

act.WriteLine("<Patient id="""& rs("patid") &""">")
'act.WriteLine("<Patient_id>" & rs("patid") & "</Patient_id>" )
act.WriteLine("<Patient_name>" & rs("patName") & "</Patient_name>" )
act.WriteLine("<Address>" & rs("address") & "</Address>")
act.WriteLine("<Phone>"& rs("phone") & "</Phone>")
act.WriteLine("<E-mail>"& rs("email") & "</E-mail>")
act.WriteLine("</Patient>")

' move to the next record
rs.movenext
loop
```

```
' All non repetitive xml on bottom goes here
act.WriteLine("</Patients>")

' close the object (xml)
act.close

' Writes a link to the newly created xml document in the browser
response.write "<a href='patients.xml'>Click here to see Patients</a>
(.xml) Database <br>"
response.write "on " & now() & "<br>"
%>
```

6.3 XML Coding

```
<?xml version="1.0"?>

<Patients>

<Patient id="P041">

<Patient_name>Maria</Patient_name>

<Address>69 Jalan 44/21, Kelana Jaya</Address>

<Phone>76564234</Phone>

<E-mail>maria@maria.com</E-mail>

</Patient>

<Patient id="P042">

<Patient_name>Eric</Patient_name>

<Address>37/4A Jalan Universiti, Petaling Jaya</Address>

<Phone>09958898</Phone>

<E-mail>eric@eric.com</E-mail>
```

```
</Patient>
<Patient id="P043">
<Patient_name>kk@kk.com</Patient_name>
<Address>No 99, Precint 10, Putrajaya.</Address>
<Phone>01234567</Phone>
<E-mail>kk</E-mail>
</Patient>
</Patients>
```

6.4 Program Optimization

Program optimization is a process of improving the efficiency of the system. Electronic Medical Records is a GUI-based program; the speed at which information appears on the screen often gives user an impression on how well the system will perform. Therefore, it is advisable to perform the optimization. There are two ways in doing so; the first one is to increase the execution speed of the program and the second one is to decrease the amount of memory the system need to run.

6.4.1 Increase Execution Speed

To bring the system to its best performance, it is advised to increase the execution speed of the system. There were few steps taken to increase the execution speed:

a) Avoiding using various data type

Variant data require additional internal program standards to identify the information being stored.

b) Minimize the amount of program initialization

Do not run too many programs at the same time. By doing so, the user perceive that the program is running faster.

c) Use Image Control

Always use image control whenever displaying bitmap image in the program.

6.4.2 Decrease the program size

Two steps can be performed:

a) Reviewing codes for unusual variables, constants and "dead codes"; remove them from the program coding.

b) Assigning the string variables to a zero length string if it is no longer needed

6.5 Summary

System implementation involves a lot of programming. It is the main component of this phase. Programming itself requires a lot of effort and great deal of creativity. I have started from scratch and put a lot of effort to learn and get used of the tools and also the programming style. The design is just a guide helping the real process to get the every function working. There was no boundary limited to us and we have been given the flexibility to express our creativity. Even though what is written in the planning does not have to be exactly implemented but it is more proper and professional to follow everything we have planned earlier. This will prove the capability in best planning method and the skills and intelligence to turn a planning into real product.

CHAPTER 7 SYSTEM TESTING

7.0 Introduction

Testing is another important phase in the system development cycle. System testing is required to ensure that the entire application, of which the modified program was a part, still works. Before going live, the newly developed application system should be thoroughly tested. It is the process of executing the application programs with the intent of finding errors. This is achieved using carefully planned test strategies and realistic data so that the entire testing process is methodically and rigorously carried out. Each function and module in this Electronic Medical Records has been individually tested before integrated together as a system. Later, after the integration they were tested again as a whole complete system.

7.1 The Objective of System Testing

There are four main objectives of system testing:

i) To ensure all the functions is working properly and check whether behavioral and performances requirements have been fulfilled.

ii) To list out all the errors and faults that occur and predict the errors and fault that are most likely to happen.

iii) To repair and correct all the errors and faults that occur while executing the system.

iv) To extrapolate the impact of the changes on program and application throughout and response time from the before-and –after results using the test data and current performance. ***(Regression testing)***

7.2 Testing Techniques

There are two techniques being applied in this testing phase; the white box testing and the black box testing.

7.2.1 White Box Testing

White box testing is a test case design method that uses the control structure of the procedural design to derive test cases in the system. The following test derived in this system by using the white box testing.

i) Ensures all the independent paths within a module have been tested at least once.

ii) Test all the logical decision on their 'true' and 'false' sides.

iii) Test all loops at their boundaries and within their operational bounds.

iv) Test internal data structures to assure their validity.

7.2.2 Black Box Testing

Black box testing derives sets of input conditions in Electronic Medical Records that will fully test all functional requirements for the program. It is not an alternative to white box technique, but a complimentary approach to uncover a different class of errors. Black box testing attempts to look for errors such as:

i) incorrect or missing functions

ii) interface errors

iii) errors in data structures of external database access

iv) performance errors

v) initialization and termination errors

7.3 Types of testing

In general, the Electronic Medical Records had been through general stages of testing including the unit testing, integration testing, and system testing.

7.3.1 Unit Testing

Unit testing is done to discover error in each module. It is a part of the development that tests each component separately. The unit testing involves:

i) Interface testing – to ensure proper information flows (in and out).

ii) Boundary testing – to ensure that the unit is working properly at the boundary values.

iii) Error testing – to check the error paths and correct the errors.

Throughout the development of the system, the unit testing is conducted after the development of each component. If an error is encountered, the debugging strategies will be taken immediately to identify the problem and correcting the error.

7.3.2 Module Testing

A module is a collection of dependent components that encapsulates related components only. Therefore, it is possible to test a module alone without the presence of other modules.

7.3.3 Integration Testing

Integration testing is the process of verifying the components integration and parallelization. All the buttons were tested in order to ensure all the links are

working and the pages that linked together are synchronized. It is also important to ensure that the interface design and coding design are suitable.

7.3.4 System Testing

This part of testing was performed to detect any possible error and conflict during the system's execution. The system was tested as a whole functioning program that performs all of the activities properly and responding to every action taken by the user. The entire system must be tested thoroughly before publishing it to the end users.

7.4 Test Case Design

Test cases assists in keeping track of what has been tested, the outcome of the test and analysis of the results. If a defect is found while testing, a documented test case makes it easy to recreate the problem so that required thoroughly analysis can be done to fix the problem. As such, test cases must be well documented and the test status must be tracked. Although there are no standard or rules for designing test cases for the system are designed so that it can facilitate its intended purpose.

7.5 Debugging Strategies

There are four actions taken in debugging such as:

i) find and track causes of the errors

ii) define the types of error or faults that occur

iii) restructure the program into a simpler test case

iv) determine the test environment

7.6 Summary

Testing is usually based on an intuitive understanding of how these components should operate. Integration testing, however, must be based on a written system specification. Black box testing does not need access to source code. Test cases are derived from the program specification. Structural testing relies on analyzing a program to determine paths through it and using this analysis to assist with the selection of test cases. Integration testing should focus on testing the interactions between components in a system and component interfaces. It is important to test the parts of the system which are commonly used rather than parts which are only rarely exercised.

CHAPTER 8 SYSTEM EVALUATION

8.0 Introduction

System evaluation is a process of evaluating the capability of the developed system. Evaluation is shown as a path of this final phase of the system development life cycle. The process involves several steps including the evaluation by end users, identifying the system strength, system limitation and future enhancements. In this chapter, it is also highlighted about the knowledge gained and the problems encountered during the development of the system and the solutions taken to overcome these problems. Evaluation from the end users also helps in detecting some errors and limitations of the system.

8.1 Problems Encountered and The Solutions

During the implementation of the system, many problems have been encountered to challenge the ongoing process and activities. These problems along with the solutions approaches are listed as below.

8.1.1 Lack of Knowledge in Hardware and Software Configuration

The development environment of hardware and software need to be configured first before the development phase begins. For example, during the first stage of the development, I had difficulties in determining whether all the tools I had chosen are suitable and compatible with the operating system I'm using. I have to go through few rounds of installing tools and uninstalling them again whenever I found it not suitable or unnecessary. To overcome this problem, I've browsed

through the developer community. Besides that, help files and reference from the internet also being used to solve the problem.

8.1.2 Difficulties In Determining The Scope of The System

This system involves a lot of knowledge in hospital management and medical procedures. Therefore, basic knowledge is needed as a foundation in building an application of this nature. However, due to lack of knowledge on this field, it is very difficult to determine the scope of the system and how the actual system should be working. The hospitals in this country are still using the old and tedious manual system which left us with no reference. As a result, a lot of effort is needed to gather information about the scope without knowing how detail we should go.

8.1.3 Time Consuming Learning Method

One of the tools used to develop the system is XML programming language which we are not very familiar with. We had to start learning from scratch through reading and tutorials, and it took quite some time just to know the basic rules. Same thing goes with ASP but it is much easier. It is better to learn through a live demonstration, so we decided to ask help from friends and experts to get to know how to use the software. However, it's quite disappointing that we could only provide one small part of XML in the system.

8.2 System Strengths

This system has several unique strengths. They are stated as below.

8.2.1 Different Level of User Access

This system has successfully implemented different levels of user access approach where the system interfaces and functionality is performing based on the categories of users that logging in. In other words, there are separated modules for each category of users namely "admin", "doctor", and "patient".

8.2.2 Reliable System With Effective Error Handling

To avoid runtime error, this system is developed with error handling. Error message will be displayed when the system encounter exceptions and it will not terminate suddenly. By using "try and catch" block, every error can be handled efficiently.

8.2.3 Easy Data Retrieval

By using database management system, data are not only easier to retrieve but can be managed in proper way. Patient information can now be stored in a paperless and organized manner. Retrieval and management of patient appointment schedule will be more systematic and consistent.

8.2.4 Error Message

Once the input entered is invalid or in an incorrect format, the error message will appear to let you know what the error is and how to re-enter the input correctly.

For example, in the 'change password' function, if user enter different value of password in the 're-type your password' column from the value entered in 'your new password' column, an error message will appear.

8.2.5 Auto-Detect Doctor's Availability for Appointment Setting

Once any of the appointment session is scheduled, the session will be disabled from the screen. The session will be enabled again if the appointment scheduled is cancelled. In addition, the system will do a few rules checking before enabling a patient from setting his appointment. This is to eliminate the problem of overlapping appointment and ambiguous data.

8.3 System Constraints

Although the best design and great implementation efforts has been used to develop the Electronic Medical Records, but still it has several weaknesses that affected the efficiency of the whole system. After overall evaluation, we realized that time constraint is the major cause.

8.3.1 Data Encryption

There is no data encryption implemented in this system. Therefore, the security of the system is much vulnerable. The data transferred over the internet is much exposed to anyone. There is no advance security features implemented because it's not our main focus in developing the system. However, we realized that the security issue is one of the most important issues in developing a browser-based system.

8.3.2 Project Scope

As I have mentioned earlier, due to time limit we also have to narrow down the scope of the system. There are too little functions can be performed by the system. During the development of the system we started to realize that we should added some more functions here and there to make the system more valuable and practical to use. Somehow, we have to accept that with the lack of knowledge in medical field and hospital management, it was not easy to simply put in everything in the system without knowing the right concept and procedures. Thus can only be achieved if we were given extra time to do some research.

8.3.3 Lack of Information

Patients who use this system do not know much about the doctor who examined them and they do not know which doctor that they should see during appointment sessions. It is also necessary to know the doctor's qualification and in what field they are specialized. To make it easier for the patients to set their appointment, they should be able to view the doctor's availability before they choose the date and session.

8.3.4 Lack of Functional Module

Currently, the functionalities provided focus on the patient's information and appointment scheduling. There is no support on the fast growing need of clinical decision support system. This function is not included in the system due to time limitation in research and absence of domain expert.

8.4 Current Enhancement and Upgrading

During the final stage of system development few changes had been made in order to achieve system objective and improvement. The process is still going on until today to produce a newer updated version of this Electronic Medical Records.

8.4.1 Interface Adjustment

The system interface which was designed during the system design phase is modified and changed in order to provide a good effects and impression. The images, colour scheme, and functional buttons are modified for a better presentation and pleased to view.

8.4.2 Database Table Adjustment

The structure of several database tables has been changed during system's coding and implementation. A few new fields were added to make the tables more effective in data storing and retrieving.

8.5 Future Enhancement and Upgrading

Right after we have tested the fully completed system, we have suddenly realized that there are so many things we can add in the system. A lot more functions, interesting way of interacting with users, a better interface design, and also informative pages should be added in the system. However, these will require more time and effort to achieve. Therefore, we hope that in the future, we could come out with a new version of this system and definitely with an upgraded quality and performance.

8.5.1 Drug Prescription Records

In the current version of the system, there is no function that allows the doctors to write a prescription. The prescription has to be made separately and not recorded electronically. By having such function, the prescription can be written in a standardized format and it's easy to refer the records later. This will make integration with the pharmacy department much easier.

8.5.2 Drug Inventory Control

In the future the system should be able to control the flow of drugs in their stores. It is more like an inventory to calculate how much drugs have been used and how many left. This is to ensure that new orders can be made before the drug is out of stock and to prevent from using expired drugs. It will be more systematic for the pharmacist to trace where they keep the drugs and put them in systematic shelves.

8.5.3 Patients Clinical Records

After the completion of the system, we suddenly realized that we have missed one of the important functions in an Electronic Medical Records that is Patients Clinical Records. In the current system, we only provide the patients personal details not the clinical details. We hope that we will be able to create a new page which provides patients clinical records consist of blood type, weight and height, pulse rate, blood pressure level, etc.

8.5.4 Billing Management

The existing Billing System only provides a total sum of what is charged to the patients instead of an itemized billing. We are looking forward to providing a more informative itemized billing to inform the patients on every single item or service that has been charged to their account. It is necessary also to keep this in record and maybe print a computerized receipt for every bill paid.

8.5.5 Data Warehousing

To make the system more effective and more functional, it is suggested that we store all the data in a centralized warehouse to make it easier to retrieve whenever it is needed. Integration with another department such as laboratory, diagnostic and imaging unit, pharmacy, Operation Theater and also therapy center will make the system a widely-used throughout the whole hospital's operation.

8.5.6 Integrated Medical Report

In the traditional procedure, there are too many papers involved in generating a report and patients and staffs have to go to different departments to collect the report for various tests such as urine test, blood test, and x-ray result. Once the upgraded version of this system is released, they do not have to do those anymore but just collect all the result at one stop with just one click; that is on-line!

8.5.7 On-Line Help File

Currently, the Electronic Medical Records does not include on-line help files that can be used as a guide for users except for the user manual alone. Therefore, in the future, it is essential to have this feature in our system to give a brief demonstration and quick tour to learn about the system.

8.5.8 Bulletin Board

This is the space where users can gain information which is posted and organized by the admin. They will post on informative articles and messages on healthcare, disease, healthy lifestyle and also current events associated with health and lifestyle.

8.6 Knowledge and Experience

Throughout the whole developing process, I have learnt a lot of precious and valuable experience and knowledge that I had never imagined I will get. Some of them are listed here.

8.6.1 Communication and Presentation Skills

Communication and presentation skills are very important in a presentation and demonstration. I started to realize this during my VIVA sessions with the supervisor and moderator. I have to give a clear explanation and description on each and every detail about this system. That was not enough! I have been asked a lot of questions regarding the system and I have to tackle and deal with all the questions and critics that I have received.

8.6.2 Knowledge in Development Tools and Technology

Before this, I have never been so interested to learn more about the new technology and tools to develop a system like this but I have learn that you have to learn by yourself if you want to be satisfied with your final output. As for the result, I am very proud of what I have done and now I am a step forward in system development.

8.6.3 Project Planning Skills

During the development phases of this system, I have noticed the importance of a good planning. Previously, I have only learnt the theories through reading and also in-class lectures. However, now I could see with my bare eyes what does it really means and how to practice all theories that I have learnt. Good planning will minimize the risk of losing track of the whole development concept. It is also to prevent developer from last minute work load.

8.7 Conclusion

As a conclusion, I could say that the main objective of the system development had been fully achieved. It is a stepping stone to bring hospital management system to be more effective and efficient slowly starting from managing the resource of Electronic Medical Records and to be expanded to more departments sooner.

The traditional management procedures are tedious, paper-oriented and also time and space consuming. Therefore, we have to adapt this new method in our way of life as our work culture and get used to the technology environment. The rapid growth of the technology requires us to walk alongside it if we do not want to be left behind. When we start to get used of it, we tend to realize that Electronic Medical Records will bring a lot of advantages and change our perspective to a better one.

A lot of work and effort have been put into the research and development of the system. From planning, literature review, functions research, requirements study to development and testing process; everything is done with full attention to bring out the best output. Last but not least, all I can say is that the whole process to come out with the system was not as easy as it seen but above all of that, I think it is indeed an interesting project and very challenging; an experience which is so valuable.

REFERENCE

1) Barbara C. McNurlin, Ralph H.Sprague, Jr. (2002). *Information Systems Management In Practice 5th Edition*. Prentice Hall

2) Frank Boumphrey. (2000). *Designing XML based Medical Applications for Windows*

3) George M. Marakas. (1999) *Decision Support Systems-In the 21st Century.* Prentice Hall.

4) Ian Sommerville. (2001). *Software Engineering 6th Edition*. Addison Wesley.

5) Beizer, B. (1990). *Software Testing Techniques 2nd Edition.* Van Nostrand Rheinhold.

6) Kit, E. (1995). *Software Testing in The Real World: Improving The Process.* Addison Wesley.

7) Jack Vaughan. (2003). *XML Meets the Data Warehouse*

8) Leonard M. Jessup, Joseph Valacich. (1999). *Information System Foundations.* Que Education and Training.

9) Norman Walsh. (October 03, 1998). *What Is XML?*

10) http://www.w3schools.com

11) http://ww.microsoft.com

12) http://www.infoloom.com/gcaconfs/WEB/paris2000/S32-03.HTM

13) http://www.kayodeok.btinternet.co.uk/favorites/kbofficeaccesshowto.htm

14) http://support.microsoft.com/?kbid=237980

15) http://ww.xml.com

16) http:// www.amazon.com