HEALTH INFORMATION SYSTEM (HOSPITAL BILLING SYSTEM) WXES3182

AGNESIA A. SAPATI
WET 990200

DEPARTMENT OF INFORMATION SCIENCE
FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
SESSION 2002/2003

Mom,

I love you so much...

For,

Lewellyn @ Lawrence E. Galakong, Cassius @ Ben Chia & Charley Chia

Thank You For Helping Me A Lot In This Project,
For providing me coding, shelter, transport, food, entertainment,
And especially your friendship,
God Bless You All...

Especially For All My Special & Wonderful Friends,

Jacinta Richard, Suzie Samuel, Neilson Richard, Jeneffer Singkoi, Hilina Jainis, Claire Pain, Norainee Jamal, Suryati Imran & Rajaratanadavi a/p Nadarajah (a.k.a roomate 2002/2003)

Thank you all for giving me a lot of encouragement to handle this project,
Thank you all for always being there,
Your friendships mean so much to me,
God Bless You All...

Acknowledgement

It is with great pleasure and gratitude that I would like to acknowledge the contribution of several people who helped me and guided me throughout all the planning stages of this project proposal.

First of all, I would like to express my gratitude to my project supervisor, Assoc. Prof. Dr.Diljit Singh for his guidance and valuable advice.

Also I wish to express my appreciation to my project moderator, Mrs. Norizan Mohd Yasin and Mr. Mustaffa Kamal Mohd Nor for their generous advice and valuable guidance.

Special thanks to Noor'ain Mokhtar, my course mate for her support and helps during my project proposal.

Not forgetting my family, who always be in my side, in bad or good condition.

Last but not least, I would like to express my gratitude to all the people who had kindly lending me a helping hand throughout the proposal of this project.

Abstract

The project proposal contained herein encompasses the system analysis and literature review of the billing module. It details the software and hardware involved, including discussions on why certain configurations are favoured over other.

The reader will find that this proposal also contains the methodology used for the system's development. The methodology discussed is based on basic software engineering principles and will be applied throughout the development phases of the system which will consist of system analysis, design, implementation, testing and so on.

The billing module will be used to issue bills to the patients for payment.

Extensive research and background surveys on all topics involved have been made prior to the preparation of this proposal. This has been done to ensure a higher level of accuracy in the designing process, along with a better knowledge base to begin with when actual system development begins. It is hoped that users will find this proposal an informative document regarding the system which is to be developed later.

Lists Of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Overview

Many hospitals these days offer comprehensive online web services to potential customers. The services may range from a great variety of choices, from pricing rates to online health advice.

The Health Information System is a system where the information is targeted mainly towards the intranet. Mostly web-based on an interface basis, HIS facilitates the management of various hospital services by efficiently providing information on the fly to hospital staff in charge of said services.

The project dealt with in this proposal is the billing module. This module are submodule of HIS. The module facilitates the generation of payment bills for patient.

The billing module's main purpose will be to generate a bill based on the treatment and drugs used by the patient. The information about the patient will be retrieves from the respective information databases, calculate and then presented in a form which can be printed out. After payment, the billing module will then update the database to indicate that the patient has paid.

1.2 Objectives

The following are the objectives of this project:

a) To ease and semi-automate the billing process where it can easily retrieve data from the database for patient's bills and payment record.
b) To generate a bill to be given to the patient indicating the list of treatments and medications along with the prices.

c) To practice paperless environment where it can reduce the amount of paper used for recording bills and payment made by the patient before.
d) To reduce the amount of time required for searching and handling patient's bill record.

1.3 Scope

The coverage of this system is all kinds of patient's bill and payments – oriented data administration in a hospital. The implementation of the system make all the payment affairs of the hospital integrated electronically. The development of the system is focused to the potential users that are consisting of the medical professional (doctor) and administration staff and pharmacist. The billing module will be developed according to the following parameters:

a) Login to verify only the authorized person can access the system.
b) Develop a database system to store the data in a secured and accurate way.
c) Develop a billing system which will ease the retrieval of pertinent information from the appropriate patient databases and calculate the amount due.
d) Recording data by the medical professionals, staff and pharmacist.
e) Generate a bill upon payment to be given to the patient.
f) Generate a receipt when the patient have made the payment.
g) Not implement credit card payment or insurance company payment.

1.4 Expected Outcome

The expected outcomes for this project is as followed:

- Acceptable respond time when the data are keyed in or retrieved from the database.

- Simple and user-friendly system as it is easy to navigate.
- The capabilities to upsized the database for the advance functions as well as the patient's record.
- Provide a more effective and efficient way of sharing data and overall system.
- Data input will be examined and errors will be reported.

The system will also consider a few years' prospective. This includes the doctor's prospective, cashier or staff's prospective and pharmacist's prospective.

The doctor's prospective:

- ✓ User is required to log in using their username and password to the system.
- ✓ User is required to log out from the system after using it.
- ✓ User is required to input the prescription and lists of the drugs in their dosages and usages.

The cashier or staff's prospective:

- ✓ User is required to log in using their unique ID and password before they can proceed to create and update the patient's bill record.
- ✓ User is required to log out from the system after using it.
- ✓ User is possible to search and look for a patient's bill record by inputting his/her patient ID.
- ✓ Ease of daily transactions such as printing bills and billings receipt.

The pharmacist's prospective:

- ✓ Pharmacist will only give the prescription and drugs if the patient have receipt that prove that the patient have paid.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

Comprehensive research and careful planning is required for the reliable and consistent development of any system. The slightest oversight could result in fatal system flaws which could prove disastrous in the later stages of system design. To prevent this from happening, extensive research and various background surveys have been carried out on the various topics and tools related to the project. These efforts have also the added benefit of providing a deeper insight and understanding of said related subject matters.

2.2 Importance of Literature Review

Literature review is important to a project as it places the project in the context of others, which might have similar charateristics. It also helps to understand the existing features that are offered by a similar system. Besides, literature review also assists to equip the developers with some knowledge of the strength and limitations of several development tools and technologies in the market. This will help the developers to choose the right tools to build up the system.

2.3 Comparisons Between Manual System and Computer-based System.

The comparisons between the manual operated and computer-based systems are as followed:

NO	MANUAL SYSTEM	COMPUTER-BASED SYSTEM
01.	Records are written in papers and stored in cabinet or cupboard.	Records are stored in database.
02.	Time is needed to search	Records can be search

	the record from the place where the record is stored.	easily through clicking the mouse or typing through keyboard.
03.	Records may be misplaced or lost.	Records are stored safely and securely in the database.
04.	Handwritten records may be difficult to read or analyze.	Records are stored in standard fonts and size and easily to read.
05.	The management is based on person. If the staff quits, now staff needs a lot of time to pick up what the system the previous staff did.	The management is based on system and not any particular person. It is easy for new staff to learn the system.
06.	Paper work is important.	Less paper work is involved.

Table 2.0 : Comparisons Between Manual System and Computer-based System

2.4 Technology Consideration

Before starting to develop any systems, it is most suitable to research on the technology, which will used as the basic tools to develop this system. The following part is the summary on some of the suitable technologies to be chosen as the basic tools for the whole development of the Billing System.

2.4.1 Platforms

There are several suitable platform can be used for the development of the system. A designer must familiar with their platform. Each platform must be able to provide a suitable and comfortable environment for the development process. Several platforms have been analyzed such as Windows NT Server 4.0, Windows 98 2nd Edition and Windows 2000 Server.

2.4.1.1 Windows NT Server 4.0

Windows NT Server 4.0 was designed from the ground up to be the most integrated, comprehensive and easy server operating system and to provide several premises such as scalability, reliability and manageability necessary for mission-critical application. There are several features Windows NT provides, such as:

Graphical User Interface (GUI)

Windows NT is extremely easy to administrate though various GUI administrator tools. Those who are relatively new to Windows NT can easily learn how to use their Windows NT.

Ease of Maintenance

Windows NT offers ease of administration or maintenance where administrators can learn, use and manage one system easily. Using an operating system that is easy to administrate saves times and cuts back on administration cost.

Rich Network Features

Windows NT 4.0 provides several new networking features, the most important of which for networking are the Distributed Common Object

Model (DCOM) and substantial improvement in Domain Name Server (DNS) for TCP/IP networks. Other new networking features of Windows NT 4.0 includes an improved print spooler, plus Point-to-Point Protocol (PPTP) and Telephony API (TAPI) 1.0 to users of Windows NT 4.0's Remote Access Service (RAS).

Performance

Windows NT Server 4.0 is a more cost-effective and higher performance file server that Wovell Intranet Ware 4.11. Ms Windows NT 4.0 is 13% faster than Novell Intranet 4.11.

Multipurpose Operating System

Windows NT Server is a multi purpose operating system. It combines the performance of File and Print Servers and the power of Unix Application Severs with to ease the use of Windows.

2.4.1.2 Microsoft Window 98 2nd Edition

Windows 98 was considered as a cheap alternative to serve as the development platform. It is perfectly capable of administering a small site but unfortunately it is unable to handle high loads due to the unavailability of certain features like load balancing, which is available in Windows NT.

It is able to serve web pages due to the inclusion of Personal Web Server on the Windows 98 CD. Installation of this utility would enable Windows 98 to function as a web server for small networks thus precluding the

need to pay for the higher fee that is required to run a copy of Windows NT Server on a machine.

Windows 98 also has a better File Allocation Table Format called FAT32 [MS2]. More disk space is available as a result of Windows' ability to store information more efficiently via FAT32. Disk wastage is less and more life can be squeezed out of a hard drive so that upgrades can be put off for a longer period. Windows 98 also supports a wide range of hardware and peripherals.

2.4.1.3 Linux

Linux is free; UNIX work-alike designed for Intel Processors on PC architecture machine. Linux is not UNIX, as a UNIX is a copyrighted piece of software that demand licenses fees when any part of it's source code is used. Linux was written from scratch to avoid license fees entirely, although the operation of the Linux operating system is based entirely on UNIX and it shares UNIX's command set. It supports a wide range of software, from a text formatting language to a graphical user interface to the GNU C/C++ compiles to TCP/IP networking. Linux is also compliant with the POSIX.1 standard, so porting applications between Linux and UNIX systems is a trap.

2.4.1.4 Conclusion

A comparison between Microsoft Windows NT, Microsoft Windows 98 2nd Edition and Linux was done such as Table 2.2.1.3 below:

Description	Windows NT Server4.0	Microsoft Windows 98 2nd Edition	Linux
Load Balancing	Available in Enterprise Edition.	Not capable.	-
Load Handling	Not robust enough for high traffic.	Unable to handle high traffic.	Able to handle high traffic on the server.
Stability	Stable.	Not so stable.	Very Stable.
User Interface	Graphical User Interface (GUI).	Graphical User Interface (GUI).	Strong command line plus GUI via Windows Managers.
Cost Effective	Cost effective OS, budget is between RM2000, a fully functional Internet Server is running in matter of days.	Expensive.	Free via download.
Security	Vulnerability is high because distribution of source code is widely available.		Secure due to login facilities. Also able to log users who have entered the system.

Table 2.1: Comparisons between Windows NT Server 4.0, Microsoft Window 98 2nd Edition and Linux.

Based on the table above, Windows NT Server 4.0 is selected as a platform for the server side and Microsoft Window 98 2nd Edition is selected as the platform for client side.

2.4.2 Relational Databases

There are several suitable databases can be used for the database development of the system. It is important to contain storage and retrieval of data. It provides mechanisms for storing and organizing data in a manner that facilitates satisfying sophisticated queries and manipulations of the data. The most popular manipulations database systems in use today are Relational Databases. Two relational databases such as Microsoft SQL Server 7.0 and Microsoft Access 2000 have been analyzed.

2.4.2.1 Microsoft SQL Server 7.0

Microsoft SQL Server 7.0 (MS SQL) is a Sequential Query Language (SQL) Server that is designed to cater for enterprise database. It is a relational database system that is scalable, reliable and high performance. MS SQL is designed for Win NT based systems and can highly integrate with Back Office family of Server (MS1). The ease of integration makes it a suitable database for web-based applications.

MS SQL is a reliable database system with the following features.

- Efficient support for text and image values are not sores as part of the data row but in separate collection of pages of their own. For each of these values, all that is stored in the data row is 16-byte pointer. For each row, these pointer points to the location of data. A row containing multiple text or image columns has ease pointer for each column. Text and image support is important for web publishing.

- The intuitive administration interface MS SQL Server uses the Microsoft Management Console (MMS) to provide easy-to-use interface. This will help to ensure the database consistency and accuracy.

2.4.2.2 Microsoft Access 2000

Microsoft Access 2000 is a relational database management (RDBMS) Software created by Microsoft for small office or home user to use for storing data in relational format. Microsoft Access can be used as a database in a client/server system with data access interface paradigm such as Remote Data Object (ROD) and Data Access Object (DAO). It is powerful and easy to use. It provide intuitive and user-friendly interfaces to design and use database.

2.4.2.3 Conclusion

A comparison between these two relational databases was done such as Table 2-3.

MICROSOFT SQL	MICROSOFT ACCESS
Support very large database system.	Support midrange database system.
Allow multi-user update the data at one time.	Only one user can update the database at one time.
Highly security protection.	Provide basic authority protection.
Support large amount of user.	Not recommended.
Support large amount of transactions.	Not supported.
Specially designed for Client/Server computing.	Provide basic support for client/server architecture, but is most recommended for personal use.
Full programmability support. Integration with Microsoft Office Pack.	Minimum programmability support. Fully Integration with Microsoft Office Pack.
Need skills and practice, specially design to meet the intermediate and advanced developer.	User-friendly application, suitable for beginner developer.
It is quite expensive for the registered license.	Cheaper.

Table 2.2: A comparison between Microsoft SQL Server 7.0 and Microsoft Access 2000.

Microsoft SQL Server 7.0 was chosen as the database server to be implemented in the proposed application because of its performance, capabilities and

scalability. Microsoft SQL Server is a powerful data warehousing. It can support users large database system.

In term of peformance issues, SQL Server 7.0 is trusted to be fastest and most reliable database server on the market. Besides, it also integrates fully with Windows NT.

2.4.3 Software Development Tools

Programming language is the most important part of a software development project. The analysis of the language is mainly focus on the application's development languages such as Visual Basic 6.0 and PowerBuilder 7.0.

2.4.3.1 PowerBuilder 7.0

PowerBuilder 7.**0** from Sybase is a powerful development tool, which improves the bridge from client/server to the web. It offers the highest productivity available for enterprise-class development, combined with state of the art mode less and multilane development environment. Thus, integrating distributed Intranet and web-based applications at increasing speed, enhancing also scalability and performance. The minimum configuration recommended for PowerBuilder is 48% with 8MB of RAM. The windows permanent swap file should be set to 12MB [PB1]. And don't have any other application running on the background as the same time.

PowerBuilder 7.0 can develop front-end applications, which access RDBMS (Relational Database Management System) without coding in 3GL (3rd Generation Language) such as C or C++. PowerBuilder 7.0 uses it's own power script that is a basic–like language, that users screen

called pointers to graphically put together applications. Power Script is a 4GL (4th Generation Language) [PB1].

Power Builder's Technical Strengths:

(a) Data Window

The primary strength of PowerBuilder is its preparatory device called Data Window. Some people have called it a product within a product. Others are still discovers new uses for it. It is the primary means by which a PowerBuilder application talks to the database. It has built-in features to format data for display, allow different edit-styles, validate data changes made by a user also the RDBMS it is talking to and scores of other such invaluable features.

(b) Object-Oriented (OO)

PowerBuilder is an Object-Oriented language. Though it is not a pure OO language, it supports inheritance in most of the areas, permits encapsulation and enables polymorphism. Because of these reasons, it is possible to arhitect your applications in such a way as to reuse code within and across applications. If you make use of Object-Oriented features, it also makes it simpler to maintain an application.

(c) Native Drivers

Though ODBC (Open Database Connectivity) is good for accessing multiple databases through a common gateway, it covers only the common minimum features of these databases. PowerBuilder provides native drivers for all the major RDBMS,

such as Oracle, Sybase, Informix, BD2, Microsoft SQL Server…etc, so that you can take advantage of the power of these.

(d) Web-enabled

With PowerBuilder 5.0 and the add-ons, we can build an application, which can access data in an RDBMS through a browser, whether it is on the Corporate Intranet or on the Internet.

2.4.3.2 Visual Basic 6.0

Visual Basic 6.0 is revolutionizing software development with multimedia-intensive, object-oriented, compiles code for conventional and Internet/Intranet-based applications. It is easy to use and fast development period. Visual Basic 6.0 also provide the convenience for creating an interesting user-interface using Graphic User Interfaces (GUI) by providing windows based tools for development such as buttons, text dialog, label and more (VB1).

Visual Basic 6 includes many new features that make it more powerful and easier to use than previous version.

- ✓ *Native code compile*

  Visual Basic 6.0 has capability to compile a program to native code, such as C++. Therefore this will give a faster program. However, Visual Basic runtime library file is still needed to provide a fully functional program.

- ✓ *New Database Features*

  Visual Data Manager is the new feature in Visual Basic 6.0. It eases the maintenance of database structure, as well as to input and edits the actual data. It also helps to create, test and save SQL statements in a program.

- ✓ *Internet Features*

  Visual Basic 6.0 includes ActiveX Controls and Web Browser Control to keep Internet developers or programmers.

- ✓ *Others*

  Other features such as enhancement to code Editor and Development Environment.

2.4.3.3 Conclusion

Visual Basic was chosen as development tool because Visual basic provides the most powerful enterprise package. It provides excellent performance, is the easiest programming language to learn and to use, and provides an unmatched feature set.

2.4.4 Summary of Technologies Consideration

Due to the comparison done for each technology, it is decided that Visual Basic 6.0 are chosen as the development tool using Microsoft SQL Server 7.0 to create and manage database. The Billing System is using the two-tier client/server architecture. Windows platform was chosen for this project. In server side, Windows NT Server 4.0 was chosen, in client side, Windows 98 was chosen.

CHAPTER 3

METHODOLGY

3.1 The Standard Waterfall Life Cycle Model

For the development of this system, its software development life cycle is based on the standard waterfall model. Using this model, development proceeds accordingly to clearly defined phases, a preceding phase must be completed before the next. Starts phase completion is judged by the outcome of the phase matching the requirements defined by the previous phase. The phases are detailed in the figure below:

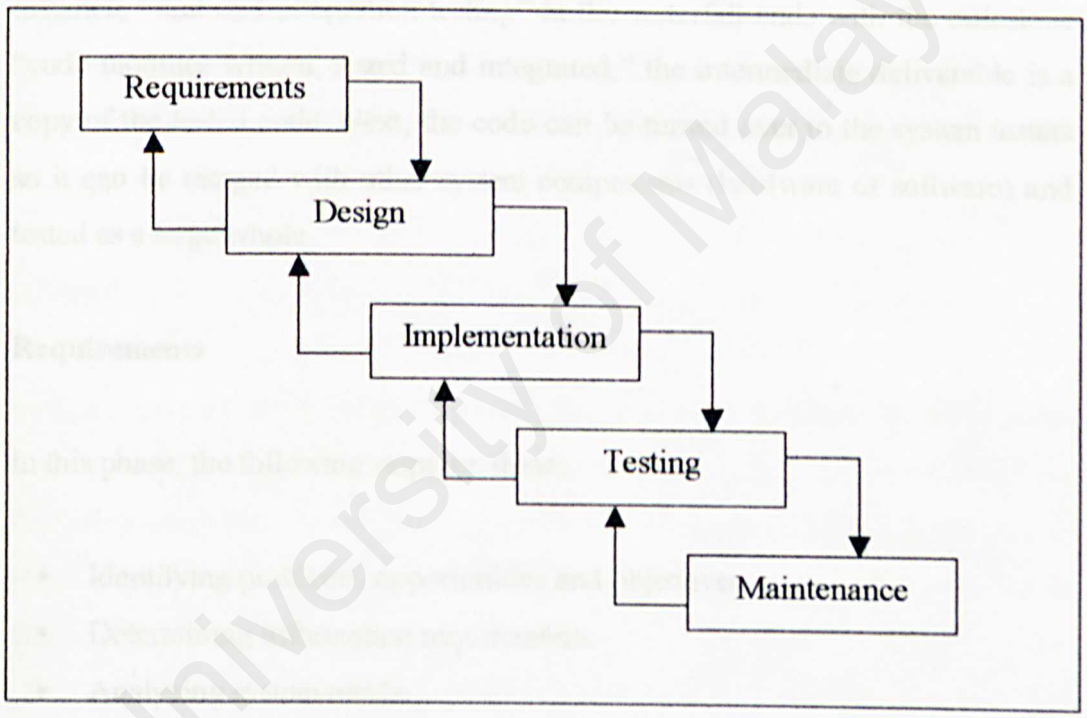


Figure 3.1 : Standard Waterfall Life Cycle Model

As the figure implies, one development stage should be completed before the next begins. Thus, when all of the requirements are elicited from the customer, analyzed for completeness and consistency, and documented in a requirements document, then the development team can go on to system design activities. The waterfall model presents a very high level view of what goes on during

development, and it suggests to developers the sequence of events they should expect to encounter.

The waterfall model has been used to prescribe software development activities in a variety of contexts. For example, it was the basis for software development deliverables in U.S. Department of Defense contracts for many years, defined in Department of Defense Standard 2167-A. Associated with each process activity were milestones and deliverables, so that project managers could use the model to gauge how close the project was to completion at a given point in time. For instance, "unit and integration testing" in the waterfall ends with the milestone "code modules written, tested and integrated;" the intermediate deliverable is a copy of the tested code. Next, the code can be turned over to the system testers so it can be merged with other system components (hardware or software) and tested as a large whole.

3.1.1 Requirements

In this phase, the following steps are done:

- Identifying problems, opportunities and objectives.
- Determining information requirements.
- Analyzing system needs.

The first steps addresses the problems which are faced and also specifies the objectives of the system. Since this overall system is a group project, the group members had to meet and discuss the overall focus of the project before each went off to concentrate on their particular subfocuses. The main consensus among the group members was that the system would be aimed at providing information to the users. This will facilitate whatever task the user is doing

whether the user is a member of the hospital staff attempting to check on the bills record of a certain patient or if the user is just a visitor to the hospital's website.

The second step involves determining information requirements for the system and its potential users. Among the tools used to define the system's information requirements are interviewing, questionnaires and researching the subject matter the information is based on.

The third and final step, analyzing system needs, is where tools such as data flow diagrams are used to chart the input, processes and output of the system's functions in a graphical form. From these diagrams, a data dictionary, which lists all of the data items used in the system along with their specifications, are created.

3.1.2 Design

In this phase, all information collected earlier is used to create accurate data-entry procedure, databases and the user interface. Designing the system based on the information collected earlier will result in a more accurate system with fewer errors. Most of the designing will be done using the diagrams created in the third phase of the requirements phase.

3.1.3 Implementation

This is the phase where the actual coding will be done. The system will be coded during this part of the development cycle. During this period, documentation for the system is also done. This included FAQs, "Read Me" files, online manuals and manuals. This documentation will tell the user how to use the system and what to do should any problems occur.

3.1.4 Testing

Before actual implementation, the system will undergo a testing phase. During this period, extensive testing will be done on the system to root out any problems or bugs, which might have been accidentally left in the system during the coding process prior to this phase in the development cycle.

3.1.5 Maintenance

After the testing has been done, the system will be implemented. This ranges from installing the system on the actual server it is to be run on to training users how to use it. However, after the initial installation and training has been done, maintenance of the system must still be continued, meaning that the system should be modified and kept updated. This phase involves correcting errors, which have gone undetected before, improvement of the system and other forms of support.

3.2 Why Waterfall?

The Waterfall life cycle model is so called a cycle because the process tends to repeat itself. After finishing one phase of the system and having moved on to the next, the discovery of a new problem may force one to return to the previous phase to modify the work done there. Therefore, note that the arrows in Figure 3.1 travel forward and backwards.

Apart from that, any modification done to the system will cause one to repeat the entire cycle again. For example, the system has already been finished and successfully implemented. Later on, maybe the users require that several new features be added to the system. If the change is small, then perhaps we can backtrack a few phases. However, if large changes are to be effected, then the

entire cycle must be repeated again to deliver a reliable and accurate system modification.

A benefit gained from using the waterfall model is its specific design phases. These specific phases of the model provides a firm environment in which to design and implement the system, and also a guideline on how to go about getting things done. Its simplicity makes it easy to explain to customers who are not familiar with software development; it makes explicit which intermediate products are necessary in order to begin the next stage of development. Many other, more complex models are really just embellishments of the waterfall, incorporating feedback loops and extra activities.

CHAPTER 4

SYSTEM ANALYSIS

4.1 Overview

This system will be running on Windows NT 4.0 with SQL Server 7.0 act as the background database. The scripting language, which will be used, will be Visual Basic, while interaction with the SQL Server database will be done with....

4.2 Billing System

4.2.1 Objectives

The main purposes of the billing module are to facilitate the charging of the hospital bills of a patient. This module will gather together all the information requires for the billing procedure, process them and then present the patient with the amount due along with an itemized listing of the items the patient was charged for.

4.2.2 Module Description

This module will retrieve information from a number of tables in the Billing System database. This is because the billing procedure will need to keep track of the treatments and medication which have been given to the patient, the name of the physician who attended to the patient and so on.

The billing process

Below is described how the billing process works:

a) After the doctor checked the patient, he/she will input the list of drugs under the patient's name and specific ID.

b) Upon discharge, the patient approaches the staff/cashier on duty.

c) The cashier inputs the patient's name into the billing form and submits it.

d) The billing module retrieves the medical records of the patient using the patient's name or ID as reference.

e) The prices of the treatments and medications given to the patient are referred to in the appropriate tables and then retrieved.

f) The total sum of the patient's stay / current treatment is calculated and displayed.

g) The billing module generates an itemized bill.

h) Upon payment, the itemized bill is given to the patient.

The decision to print and give the patient an itemized bill of the patient's stay in the hospital stems from the fact that billing processes normally contain many errors. It is typical for a patient to have been scheduled for a certain treatment, have that treatment cancelled and then still billed for the treatment, which the patient didn't receive. With an itemized bill, the patient will be able to check billing to ensure that accidental charges haven't been paid for.

Below is a chart describing the flow of the billing process:

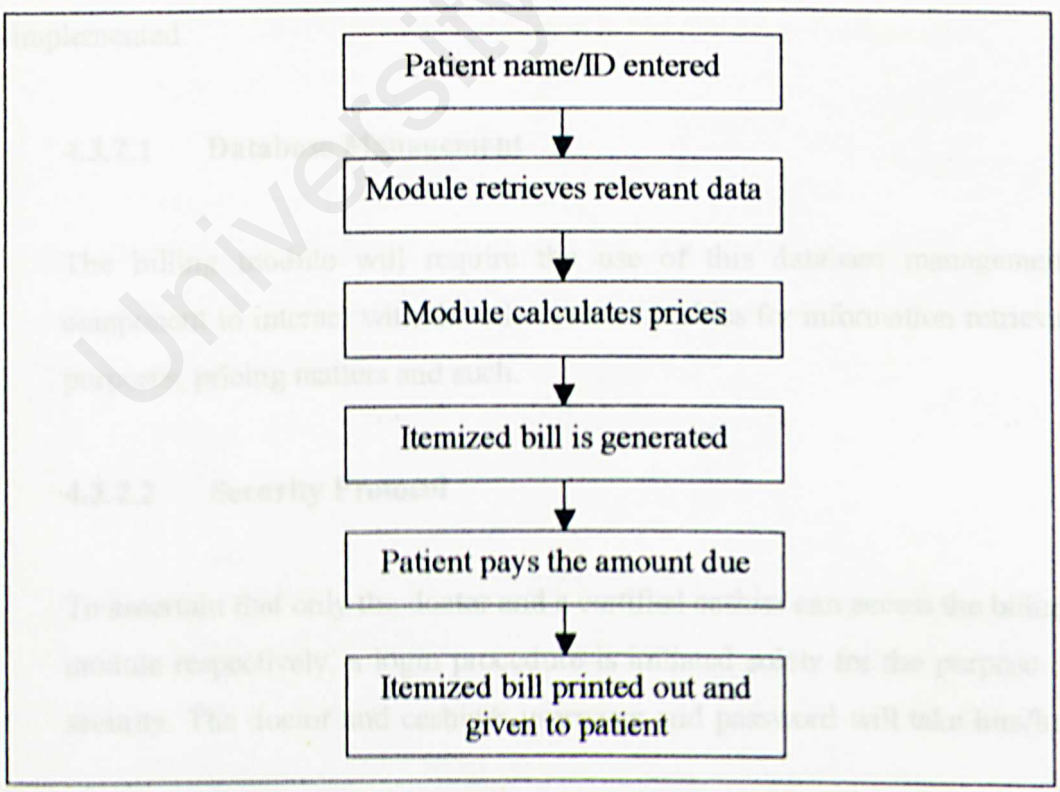


Figure 2.0: Flow of billing process

4.3 Requirements Analysis

4.3.1 Overview

Requirements analysis for this module of the Billing System can be divided into two sections – functional and non-functional requirements. Although some might be of the opinion that the functional requirements are of more paramount importance than the non-functional ones, it should be noted that both requirements are of equal importance to produce a workable and reliable system. The following sections will elaborate more on these two requirements.

4.3.2 Functional Requirements

These functional requirements are the main features, which make up the system. Being so, their development will be critical for the system to be successfully implemented.

4.3.2.1 Database Management

The billing module will require the use of this database management component to interact with the other relevant tables for information retrieval purposes, pricing matters and such.

4.3.2.2 Security Protocol

To ascertain that only the doctor and a certified cashier can access the billing module respectively, a login procedure is initiated solely for the purpose of security. The doctor and cashier's username and password will take him/her

to the billing module. This will prevent other non-authorized users (patients, pharmacists) from reaching those sections.

4.3.3 Non-functional Requirements

4.3.3.1 User-friendliness

The graphical interface designed will be optimized for maximum ease of learning and navigation. As the aim of the system is information, the interface will be modeled to provide optimal information in each page while never confusing the user with too much graphics.

4.3.3.2 Performance

Being a service integral to the running of the hospital, a highly consistent and reliable track record of uptime for the system is required. Downtime and disaster recovery must be simple and swift to prevent loss of efficiency.

4.3.3.3 Communications

Reliable and compatible communication between users and the system is of high priority. Server downtimes should be reduced as much as possible to provide efficient communication to the users.

4.3.3.4 Context Help

To alleviate problems in using the system, context help will be provided along with the system itself. A short help file might be included in the Windows Help Format and pop-up help text will definitely be included for the hospital personnel who will be interacting with the billing system.

4.4 Hardware and Software Requirements

4.4.1 Hardware Minimum Requirements

Development

Three IBM computer fully integrated with:

- Intel Pentium III 996MHz processor
- 64MB SDRAM
- 900MB Hard disk space
- Internet Connection

User Client-Side

Three IBM computers fully integrated with:

- Intel Pentium II 166MHz processor
- 32MB SDRAM
- 20MB Hard disk space

User Server-side

Three IBM computers fully integrated with:

- Intel Pentium II 333MHz processor
- 64MB SDRAM
- 500MB Hard disk space

4.4.2 Software Minimum Requirements

Development

- Windows NT Server 4.0
- Visual Basic 6.0
- Microsoft SQL Server 7.0

User Client-side

- Windows 98 2nd Edition

User Server-side

- Windows NT Server 4.0

CHAPTER 5

SYSTEM DESIGN

5.1 Overview

This chapter will see the translation of the requirements examined in the previous stage into system characteristics. From the billing module, three components will be detailed from a design prospective. They are the architectural design, the database design and the user interface design.

5.2 Architectural Design - Overview

Their process model is based on the system structuring and module decomposition methodology. The system is structured into a number of principal subsystems in system structuring. A system in itself, a subsystem's operation does not rely on the services provided by other subsystems. Each subsystem is in turn broken down into modules, and have defined interface, which are required for communications with other subsystems.

Module decomposition, however, is based on the assignation of functions to components. Beginning with a high-level description of the functions which are to be implemented, the designed then derives low-level accounts on how each component will be organize and relates to other components from it.

5.3 Architectural Design – Billing

5.3.1 Application Architecture

The application architecture of the billing system is shown below:

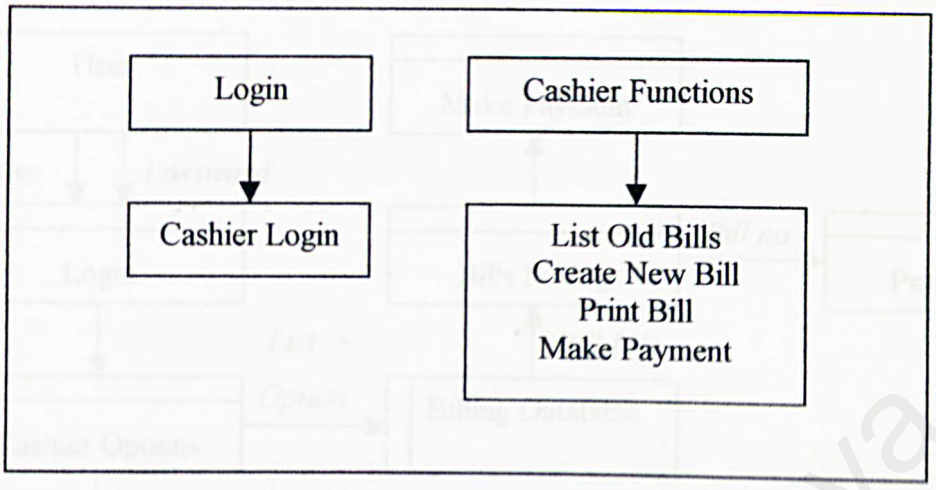


Figure 5.0: Application architecture of the billing system.

5.3.1.1 Login

The login section is where the users are determined if they are cashiers who are permitted to use the billing system. In other words, they can enter this system if they are authorized users.

5.3.1.2 Cashier Functions

Upon logging into the system, the cashiers will be presented a textfield to enter in a specific patient ID. Two options are available then – to list out the bills under that patient's ID, or create a new bill for the patient.

5.3.1.3 Data Flow Diagram

Using the Gane and Sarson notation, the data flow diagram for the billing module is represented below:

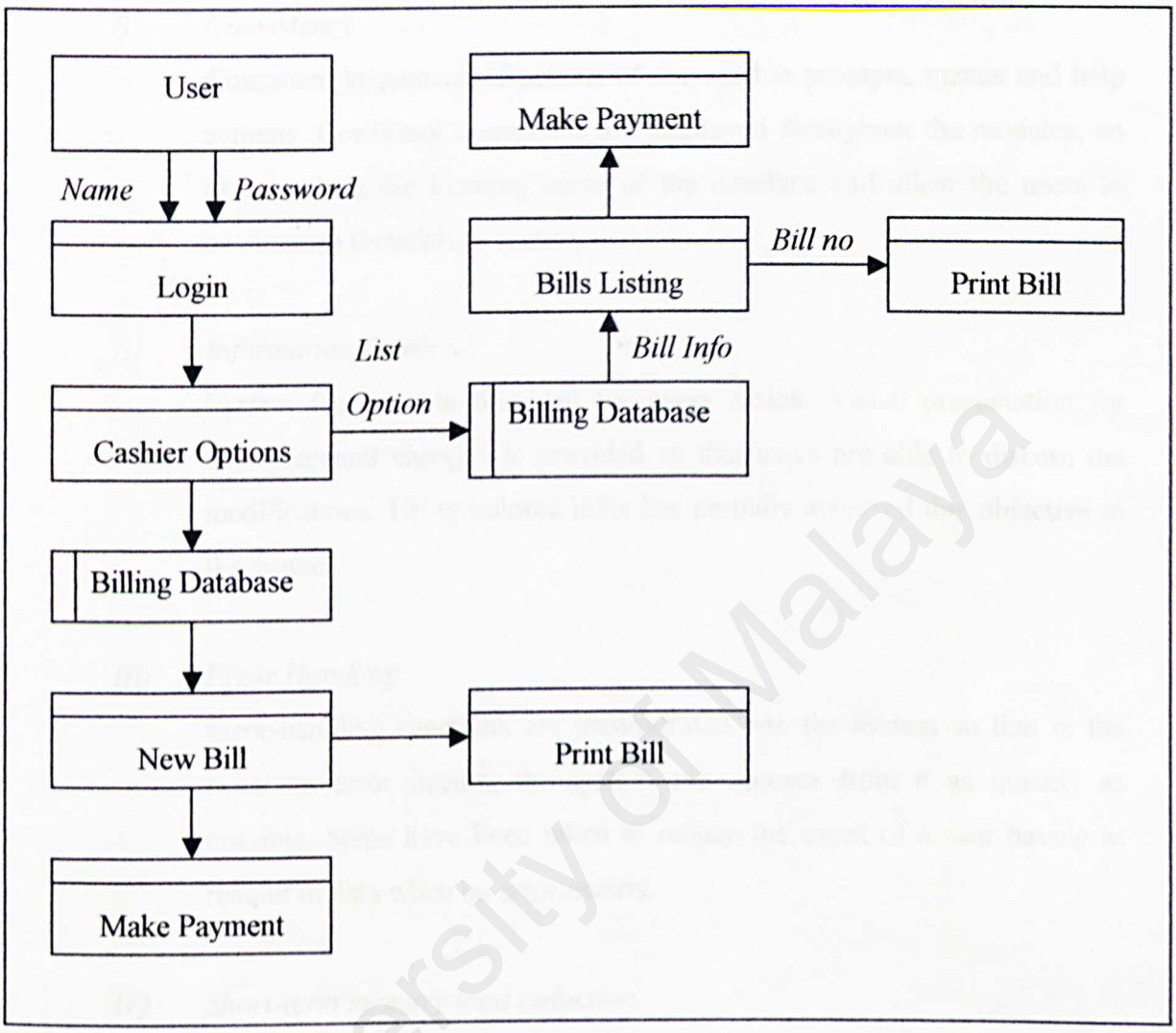


Figure 5.1: Data flow diagram of the billing system

5.4 User Interface Design - Overview

The user-interface design for the billing system based on the GUI (Graphical User Interface) approach. The general principles of HCI (Human-Computer Interface) for designing an interactive system has been considered and applied where applicable. Below are listed some of the guidelines which have been taken into consideration when designing the user interface:

I) Consistency

Consistent sequences of actions of are used in prompts, menus and help screens. Consistent commands are employed throughout the modules, so as to reduce the learning curve of the interface and allow the users to acclimatize themselves sooner.

II) Information Feedback

System feedback is provided for every action. Visual presentation for environmental changes is provided so that users are able to discern the modifications. Using colored links has partially achieved this objective in the system.

III) Error Handling

Error-handling functions are incorporated into the system so that in the event an error occurs; the system can recover from it as quickly as possible. Steps have been taken to reduce the event of a user having to reinput in data when an error occurs.

IV) Short-term memory load reduction

As the human brain processing of short-term memory is extremely limited, displays have been kept simple, window motion frequency has been reduced and everything is done to reduce the amount of memorizing that the user has to do while navigating the site. Online help has been made available for the convenience of the user.

5.4.1 Main User Page

The is the main user page which will greet everybody who enter the Billing System. It provides links to other sub-system within the system. To go to other

sub-system, it requires authorized users to be logged in. There is a user and password login textfield provided.

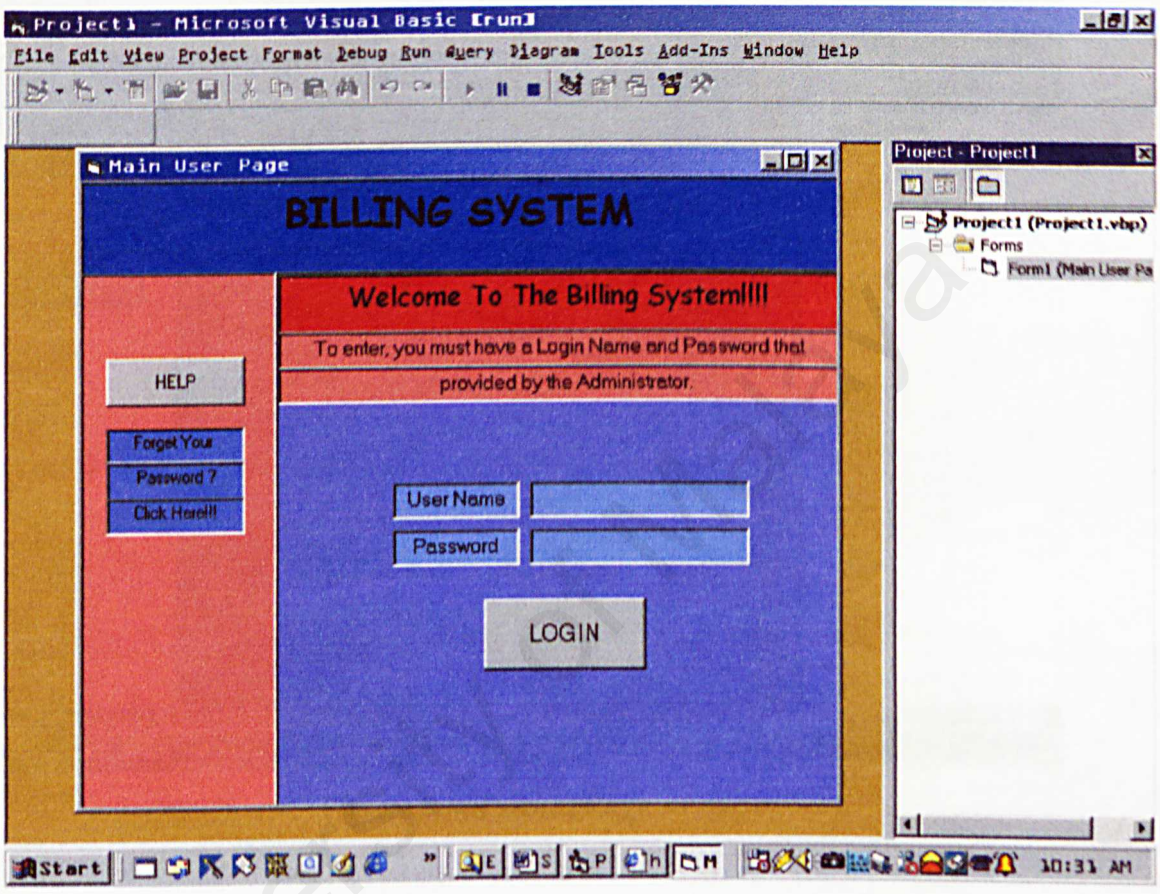


Figure 5.3: Main User Page

5.4.2 Logout Page

This is the page, which the users will see when they log out of the system itself.

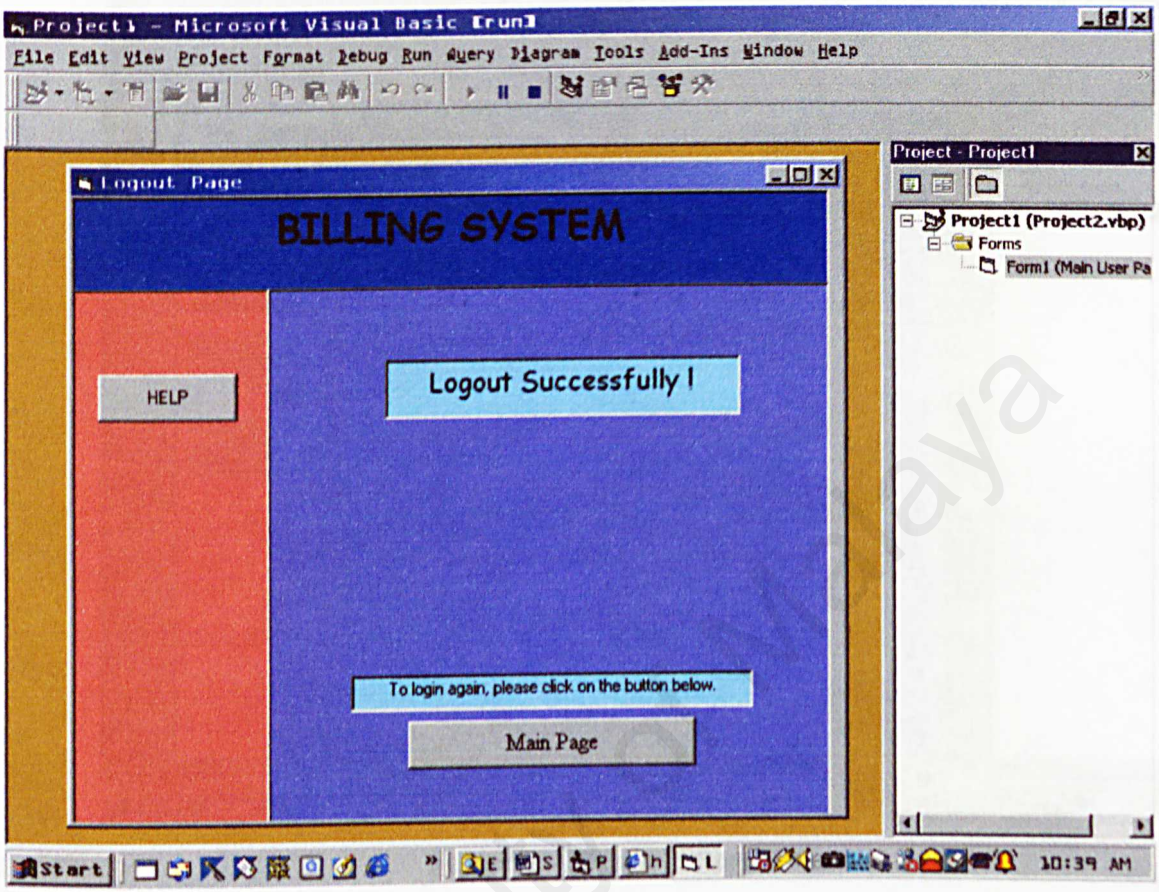


Figure 5.4: Logout Page

5.4.3 Main Billing Page

This is the page, which the cashier will first see upon logging in from the Main User Page. It contains a textfield to enter the patient's ID number and ***List Old Bills*** button to list out all the bills under the patient and a ***Create New Bill*** button to generate a new bill for the patient.

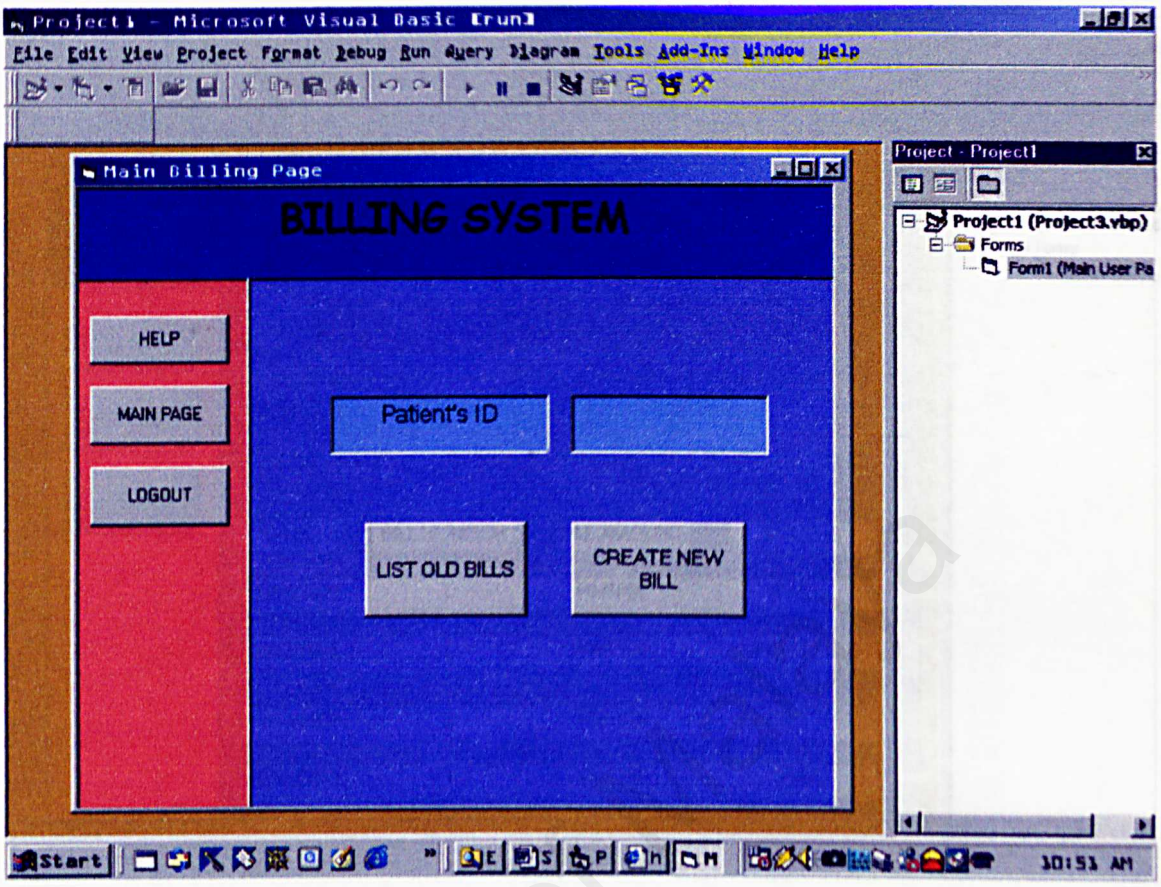


Figure 5.4: Main Billing Page

5.4.4 Old Bills Listing

Upon entering a valid patient ID and clicking on the ***List Old Bills*** button, the bills under that patient will be listed out on this page along with the date of the bill and the bill status, whether the bill has been paid in full or not.

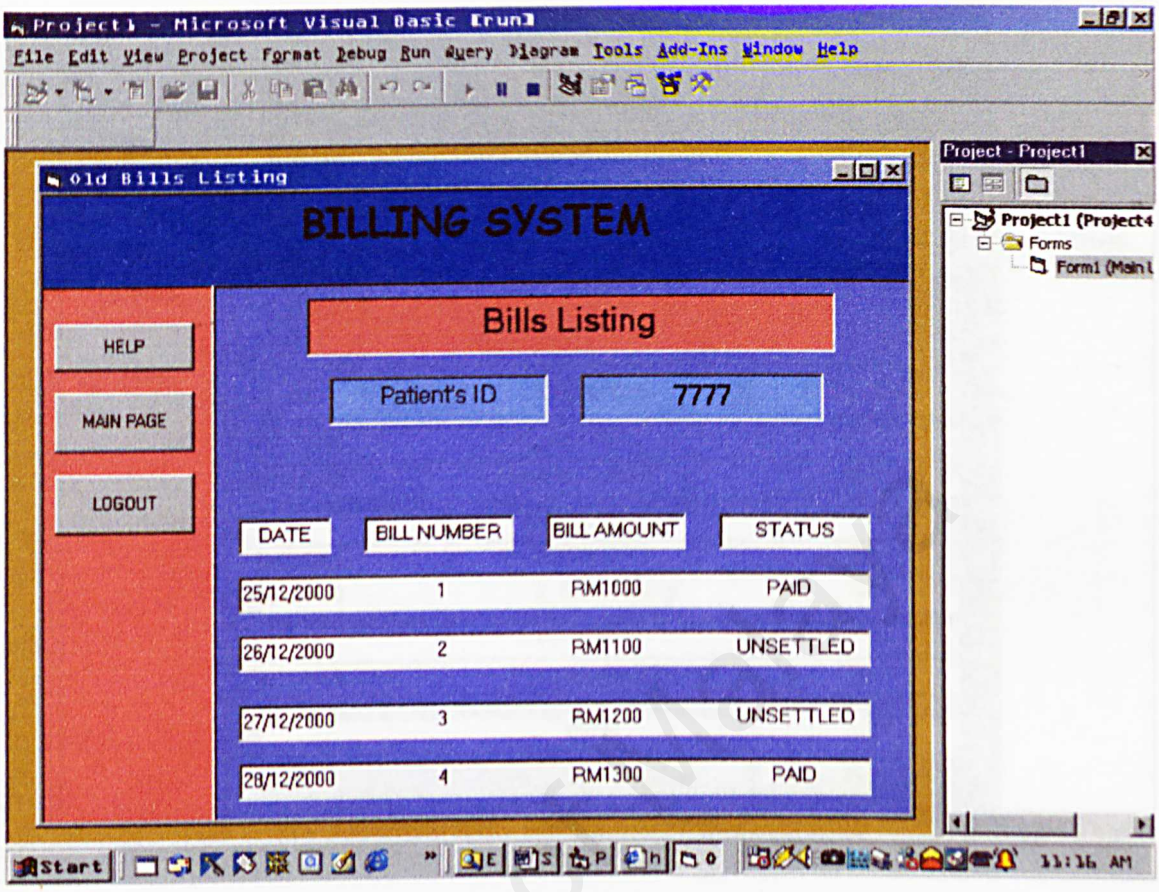


Figure 5.5: Old Bills Listing

5.4.5 Bill View Page

From the main cashier page, clicking on the ***Create New Bill*** button will create a new bill listing out the latest items for the patient from the last admission date to the current date. From here the cashier can either print out the bill (for the patient) or settle the payment of the bill. This page is similar to the one the cashier sees upon clicking on the bill number in the bill listing.

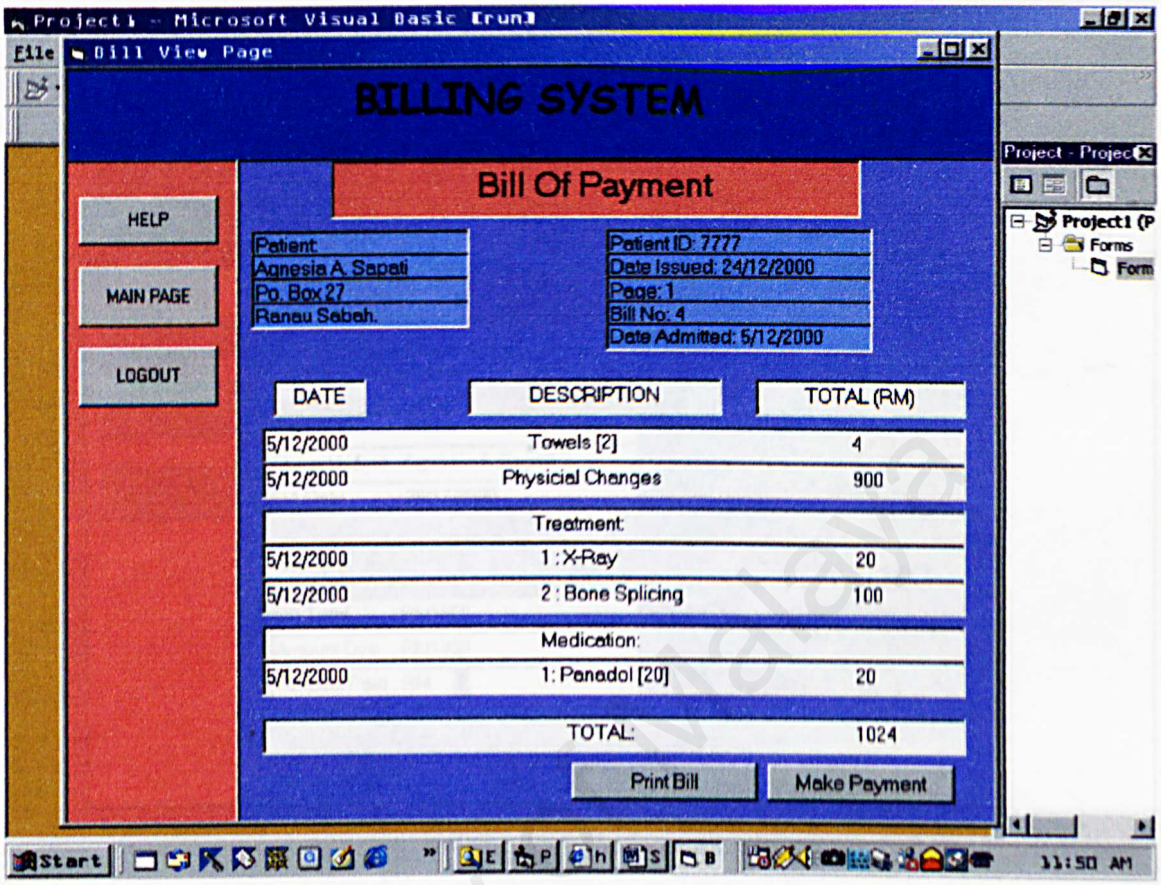


Figure 5.6: Bill View Page

5.4.6 Payment Page

The payment page lists out brief details of the patient, the bill number, the bill total and how much more is due for the bill. Upon receipt of payment, the cashier can enter in the amount received into the textfield provided.

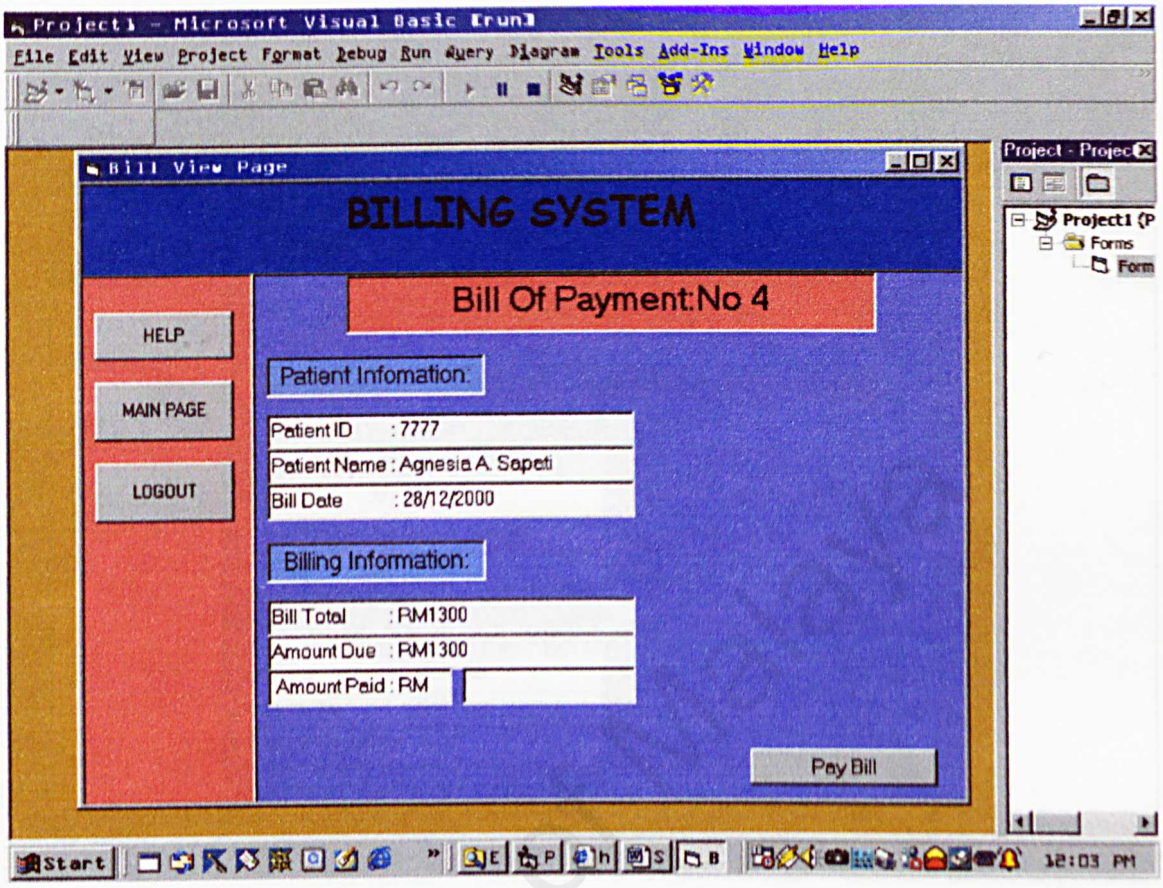


Figure 5.7: Payment Page

5.4.7 Payment Confirmation Page

The payment confirmation page is displays after the cashier keys in the amount which the patient has paid. Note that this allows for a patient to pay the bill in installments if desired so.

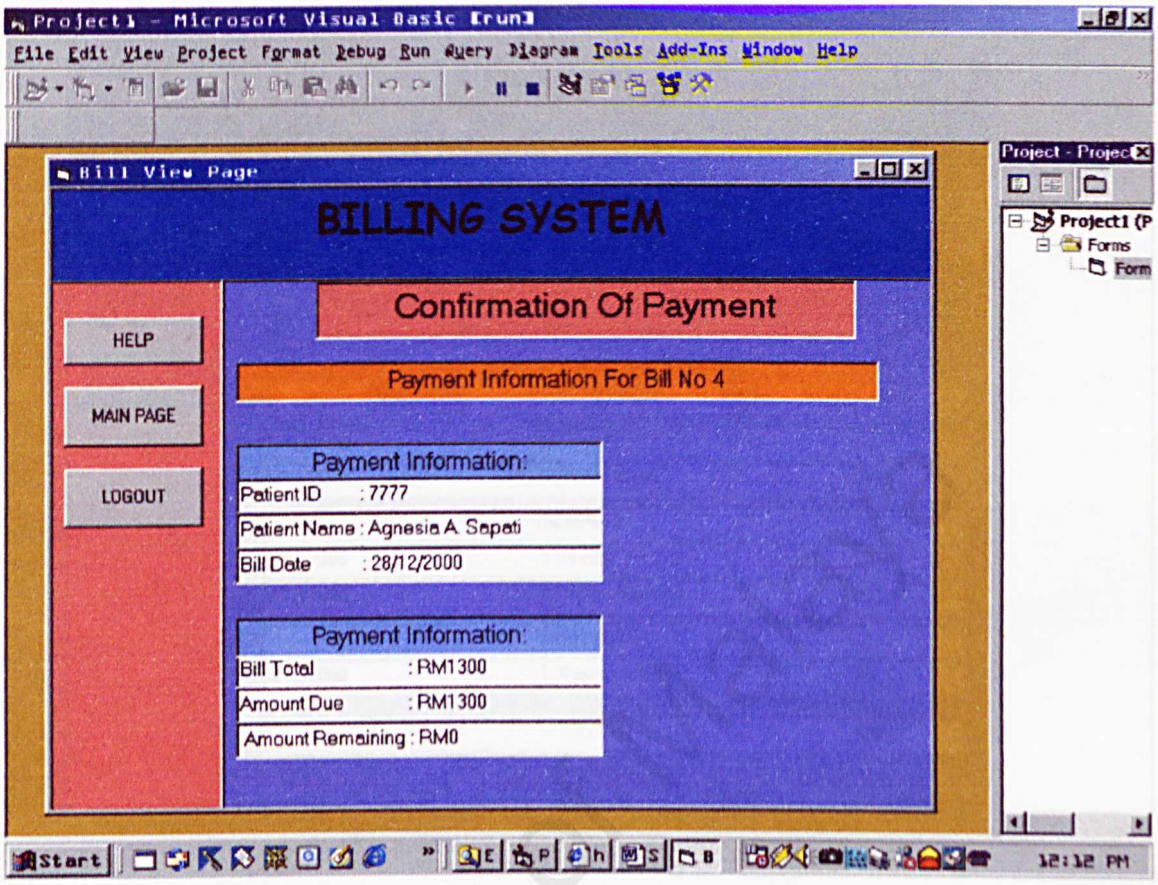


Figure 5.8: Payment Confirmation Page

5.4.8 Printing out the bill

The bill when printed out will be in three different pages – the first page will be of the basic and miscellaneous items the patients uses (room charges,etc) along with the physician's change. The second page will be the treatments the patient received and the third page will contain a list of all the medication given to the patient.

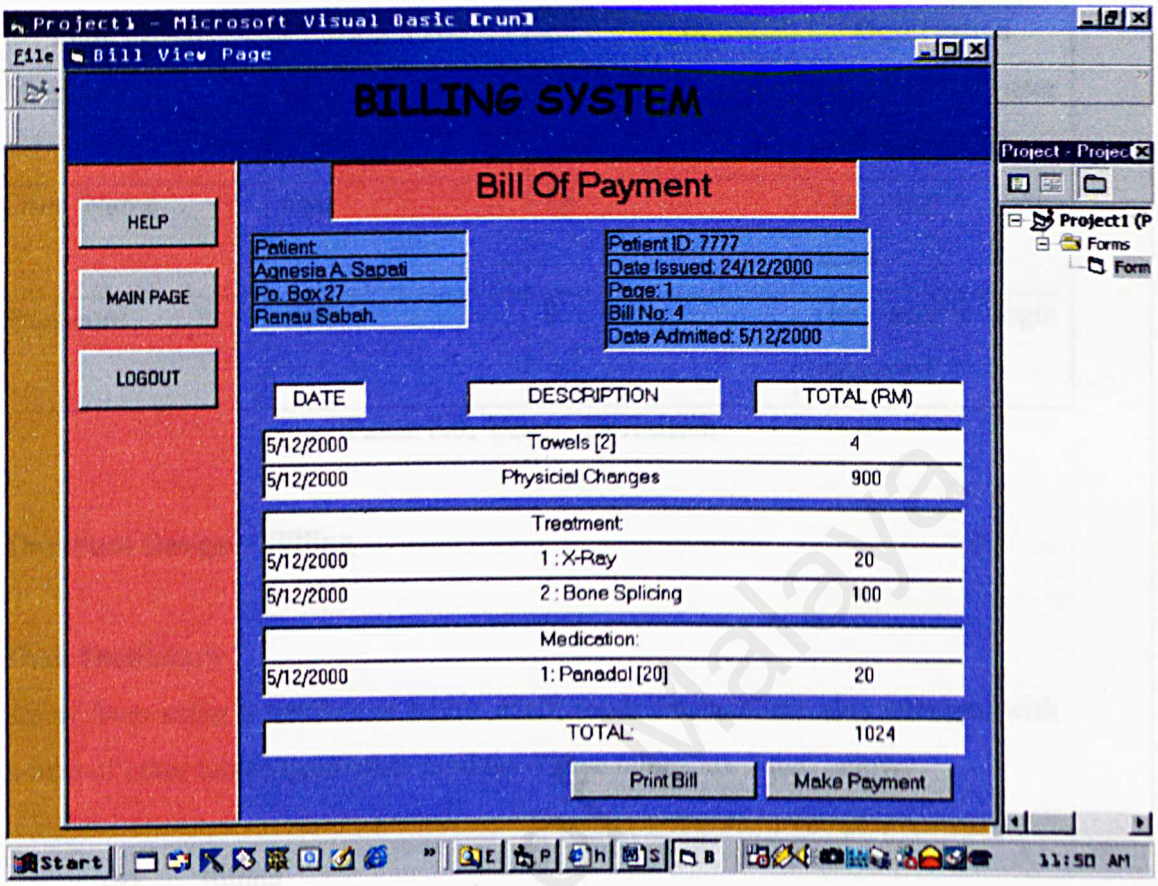


Figure 5.9: Printing out the bill

5.5 Database Design – Overview

5.5.1 Data Dictionary

The data dictionary is a specialized application of the kinds of dictionaries used as references for system analysis to guide them through the design and analysis phases. This system share the usage of the table that shown below:

User – Definition

User to determine if the user is given access in the database.

Field Name	Data Type	Length	Description
User_Id	int	4	The ID of the user in the database.
User_Name	char	15	The user's login name.
Password	char	8	The user's login password.

Table 5.0: User – Definition

5.6 Database Design – Billing

5.6.1 Data Dictionary

Apart from using a number of tables itself, the billing module also interacts with a host of other tables from other module. These tables are listed below:

(a) Billing
(b) Billing_misc_item
(c) Billing_physician
(d) Billing_treatment
(e) Billing_medication
(f) Adt
(g) Patient
(h) Misc_item
(i) Misc_item_code
(j) PhysicianBio
(k) Treatment
(l) Treat_code
(m) Medication
(n) Stock_definition

5.6.2 Billing

The main billing table. This contains the primary billing data, as well as being the table which links all the other billing tables together.

Field Name	Data Type	Length	Description
Bill_no	int	4	The bill number used to identify each bill.
Patient_id	int	4	The ID number of the patient who is being billed.
Bill_date	datetime	8	The date the bill was issued.
Bill_total	decimal	9	The total amount of the bill.
Admission_date	datetime	8	The date of admission for the patient. Together with Bill_date, used to determine the period the bill covers.
Bill_balance	decimal	9	The amount of money the patient still owes the hospital.
Current_bill_no	int	4	The number of the last bill created.

Table 5.1: Billing

5.6.3 Billing_misc_item

This billing table stores information which ties in to all the basic and miscellanous items charged to this patient. To presence the original amount charged to the patient, the old price of items are also stored inside this table.

Field Name	Data Type	Length	Description
Bill_no	int	4	The bill this item is associated with.
Bill_misc_date	datetime	8	The date the item was charged.
Misc_item_id	int	4	The ID number of the item.
Unit_price	decimal	9	The price of the item stored in the table.
Quantity	int	4	The quantity of the items charged.

Table 5.2: Billing_misc_item

5.6.4 Billing_treatment

This table records down the treatments which are charged to the patient.

Field Name	Data Type	Length	Description
Bill_no	int	4	The bill this treatment is charged to.
Bill_treatment_date	datetime	8	The date this

			treatment was given.
Treat_rec_id	int	4	The treatment's ID number.
Unit_price	decimal	9	The price of the treatment.

Table 5.3: Billing_treatment

5.6.5 Billing_medication

The table records down all the medication charged to the patient.

Field Name	Data Type	Length	Description
Bill_no	int	4	The bill this medication is charged to.
Bill_medication_date	datetime	8	The date this medication was prescribed.
Stock_id	varchar	20	The medication's unique ID.
Price_factor	decimal	9	The price per unit for the medication.
Quantity	int	4	The quantity of the medication given.

Table 5.4: Billing_medication

5.6.6 Adt

The ADT table contains information about the patient's admission. The table is used to retrieve the patient's physician and the patient's admission date into the hospital for billing purposes.

Field Name	Date Type	Length	Description
Adt_Id	int	4	The unique ADT identifier.
Patient_ID	int	4	The patient ID.
Date_admisssion	datetime	8	The patient's admission date.
Physician_id	int	4	The physician attending this patient.
Ward	varchar	10	The ward the patient is assigned to.
Remarks	varchar	100	Miscellaneous remarks.

Table 5.5: Adt

5.6.7 Patient

This table contains all the patient data. It is used only for retrieving patient data like names and address.

Field Name	Data Type	Length	Description
Patient_ID	int	4	The patient's unique identifier.

Person_name	varchar	50	Tha name of the patient.
Photo_name	varchar	255	
Passport	char	8	The patient's passport number.
IC_no	char	14	The patient's IC number.
Gender	varchar	6	The patient's gender.
Birthday	smalldatetime	4	The patient's date of birth.
Race	varchar	15	The patient's race.
Religion	char	15	The patient's religion
BloodGroup	varchar	2	The patient's blood group.
Marital_status	smallint	2	The patient's marital status.
Nationality	varchar	20	The patient's nationality.
Home_phone	varchar	10	The patient's home phone number.
Mobile_phone	char	10	The patient's mobile phone number.
Address	varchar	25	The patient's home address.

Address1	varchar	25	The patient's home address.
Postcode	char	5	The patient's postcode.
City	varchar	20	The patient's city.
State	varchar	20	The patient's state.
Country	smallint	2	The patient's country code.
Facsimile	varchar	10	The patient's fac number.
Email	varchar	30	The patient's email address.
Occupation	smallint	2	The patient's occupation code.

Table 5.6: Patient

5.6.8 Misc_item

This table contains the records of the basic and miscellaneous necessities which were given to the patient.

Field Name	Data Type	Length	Description
Misc_item_id	int	4	The item's unique identifier.
Patient_id	int	4	The patient this item is charged to.
Date_administered	datetime	8	The date this item was charged.
Quantity	int	4	The quantity of the items charged.
Remarks	varchar	50	Miscellaneous remarks

Table 5.7: Misc_item

5.6.9 Misc_item_code

This table stores the item's price and description.

Field Name	Data Type	Length	Description
Misc_item_id	int	4	The item's unique ID.
Misc_item_desc	varchar	50	The item's description.
Misc_item_price	decimal	9	The price of the item.

Table 5.8: Misc_item_code

5.6.10 Treatment

The Treatment table contains treatment records of patients.

Field Name	Data Type	Length	Description
Treat_rec_id	int	4	The treatment record's unique ID
Patient_id	int	4	The patient given the treatment.
Treatment_id	varchar	100	The treatment's unique ID.
Date_treated	smalldatetime	4	The date the treatment was given.
Attend_PhyId	int	4	The physician who gave the treatment.
Remarks	varchar	100	Miscellaneous records.

Table 5.9: Treatment

5.6.11 Treat_code

This table stores the treatment's description and price.

Field Name	Data Type	Length	Description
Treatment_Id	int	4	The treatment's unique ID.
Treatment_desc	varchar	50	The treatment's

			description.
Treatment_price	decimal	9	The price of the treatment.

Table 5.10: Treat_code

5.6.12 Medication

The Medication table contains the medication records of patients.

Field Name	Data Type	Length	Description
Patient_id	int	4	The patient given the medication.
Stock_id	varchar	20	The medication's unique ID.
Date_prescribed	datetime	8	The date the medication was prescribed.
Quantity	int	4	The amount of medication prescribed.
Attend_PhyID	int	4	The physician prescribing the medication.

Table 5.11: Medication

5.6.13 Stock_definition

The table which contains all the data on medication. It is used to restore the name of the medication and prices for billing purposes.

Field Name	Data Type	Length	Description
Stock_Id	varchar	20	The medication's unique ID.
Name_generic	varchar	50	The generic name of the medication.
Name_trade	varchar	50	The trade name of the medication.
Reorder_level	decimal	9	The reorder level of the medication.
Quantity_stock	decimal	9	The amount left in stock.
Quantity_per_unit	int	4	The amount per unit of medication.
Quantity__total	int	4	The total amount of medication.
Dosage_per_unit	int	4	The dosage of the medication.
Unit_order	varchar	10	The mesurement used for the medication.
Price	decimal	9	The price per unit medication.

Table 5.12: Stock_definition

CHAPTER 6

SYSTEM IMPLEMENTATION

6.0 Introduction

System implementation is the continuous stage to the previous system design. The implementation will focus on the process of developing the programs. Programs design and implementation are needed for a numbers of a reasons in a system. It will determine the data transfer correctly and to make sure that the system will work accordingly to the users expectation.

Writing program code is one of the big steps in system implementation. In transform the idea from the system design to program code. At the same time, it will implement the idea to the workspace and platform in order to produce what we call the pilot version or prototype of a product.

6.1 Development Tools

During the development of this project, some of the development tools have been chosen to ease up the process of the system development. This has included the tools for creating the database for this billing system and also programming language tools.

As Visual Basic Script has been chosen to be the programming language itself to develop this software, ADO Data Control and Data Access Object has been used to connect The Visual Basic 6.0 with Microsoft Access that acts as the database.

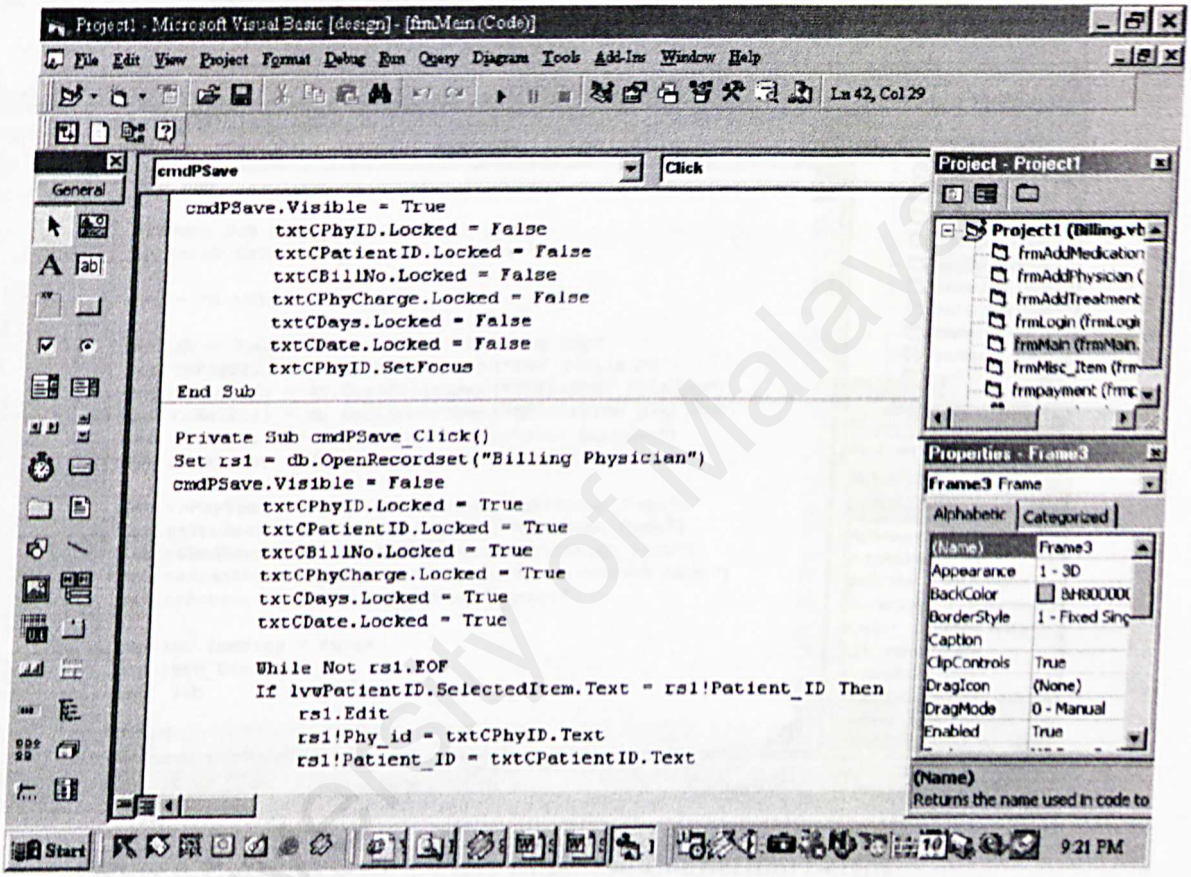


Figure 6.0: Interface of The Visual Basic 6.0

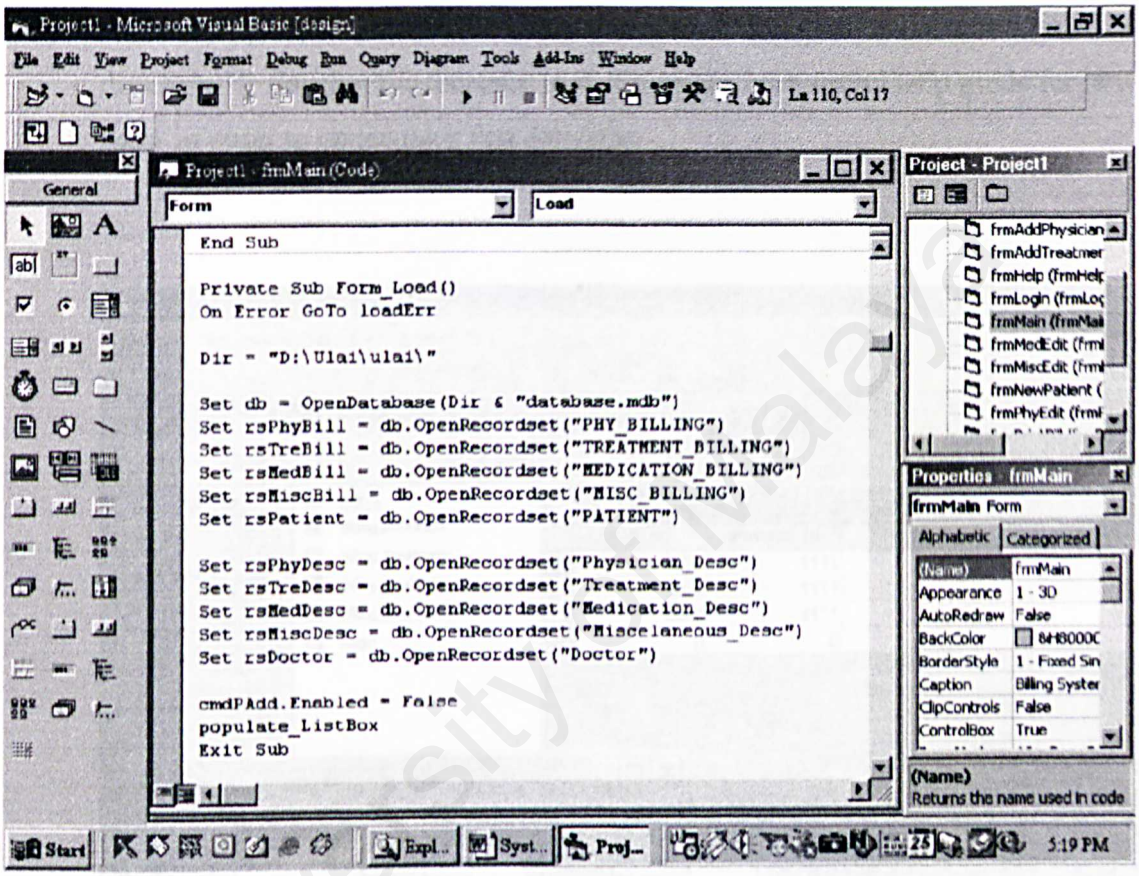


Figure 6.1: Access Data Object Database Connection Script

For the database design of this program, Microsoft Access is the tool chosen to build the database. Microsoft Access provided and easy to use user interface for developer to develop the database in it. It also has been useful help guide for new users in order to create their first database.

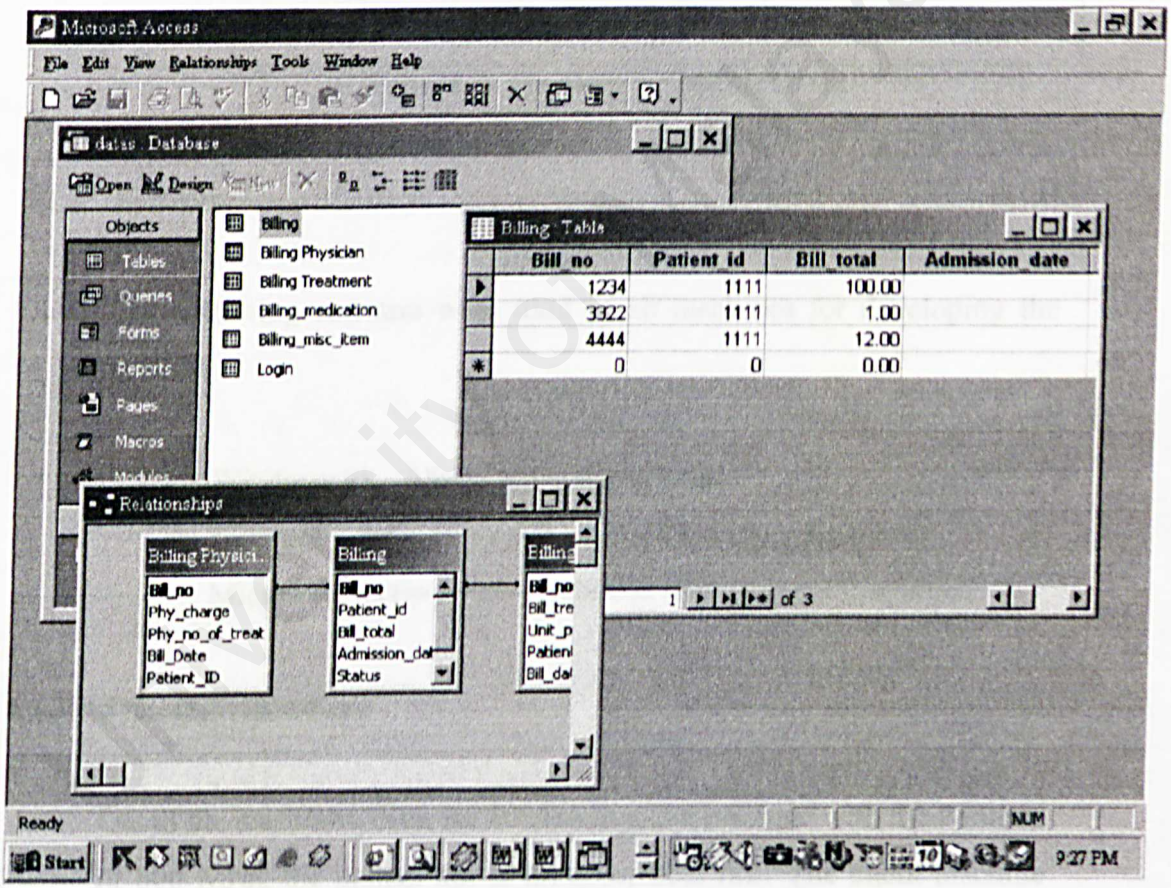


Figure 6.2: Interface of Microsoft Access 2000

6.2 The System

6.2.1 Hardware

The following machines were used for the development of these modules.

Development	Server	Client
Pentium III 800 MHz	Pentium III 1 GB	Pentium III 1 GB
128 MB RAM	256 MB RAM	256 MB RAM

Table 6.0: Hardware Specifications

6.2.2 Software

The following software were used in all machines for developing the modules:

- ✓ Windows 98 – Main Operating System
- ✓ Microsoft Visual Basic 6.0 – Main Developer Software
- ✓ Microsoft Access 2000 – Database Management Properties

6.2.3 System Evaluation

On all the machines, even on the development machine with the Pentium III 800 MHz, the system ran at an acceptable rate. The client machine was able to process and generate results sufficient than the development machine, and mainly because of the higher CPU Power and increased RAM.

For serving the needs of multiple users, it should be noted that server machine would adjust optimally to the request demands, with the additional RAM and CPU Power providing enough support for many users. The development machine, with half the amount of available memory and less then 20% the processors power, will encounter degradation in performance.

6.3 System Coding

6.3.1 Database Connectivity Coding

For database connectivity, that is to connect Visual Basic and Microsoft Access, ADO Data Control and DAO (Data Access Object) has been used. ADO Data Control is one of the special features that Visual Basic 6.0 have for database management purposes. For DAO, it is a coding that you have to write yourself.

Below are shown coding for DAO inside this system that largely used for database connection.

```
'DATABASE CONNECTION

Private Sub Form_Load()
On Error GoTo loadErr

'Path name for the database location.
Dir = "D:\Ulai\ulai\"

Set db = OpenDatabase(Dir & "database.mdb")
Set rsPhyBill = db.OpenRecordset("PHY_BILLING")
Set rsTreBill = db.OpenRecordset("TREATMENT_BILLING")
Set rsMedBill = db.OpenRecordset("MEDICATION_BILLING")
Set rsMiscBill = db.OpenRecordset("MISC_BILLING")
Set rsPatient = db.OpenRecordset("PATIENT")

Set rsPhyDesc = db.OpenRecordset("Physician_Desc")
Set rsTreDesc = db.OpenRecordset("Treatment_Desc")
Set rsMedDesc = db.OpenRecordset("Medication_Desc")
Set rsMiscDesc = db.OpenRecordset("Miscelaneous_Desc")
Set rsDoctor = db.OpenRecordset("Doctor")

cmdPAdd.Enabled = False
populate_ListBox
Exit Sub

loadErr:
   MsgBox Err.Description
End Sub
```

Figure 6.3: Database Connectivity Coding

6.3.2 Physician's Sub Module Coding

Below are shown the coding for Physician's Tab control:

```
'============== PHYSICIAN'S TAB ==============

Public Sub pop_lvwPhyBill()
   Dim col As Integer
   Dim row As Integer
   lvwPhyBill.Enabled = False
   lvwPhyBill.ListItems.Clear
   col = 1
   row = 1
   rsPhyDesc.MoveFirst
   Do Until rsPhyDesc.EOF = True
      If rsPhyDesc!Name <> "99999999" Then
         If rsPhyDesc!Name = strSelected Then
            lvwPhyBill.ListItems.Add , , rsPhyDesc!drname
            lvwPhyBill.ListItems(row).SubItems(col) = rsPhyDesc!Date
            col = col + 1
            lvwPhyBill.ListItems(row).SubItems(col) = Format(rsPhyDesc!Price, "#,##0.00")
            col = col + 1
            lvwPhyBill.ListItems(row).SubItems(col) = Format(rsPhyDesc!ID, "000000")
            col = 1
            row = row + 1
            lvwPhyBill.Enabled = True
         End If
      End If
      rsPhyDesc.MoveNext
   Loop
End Sub
```

Figure 6.4: Physician's Tab control coding.

Agnesia A. Sapati
WET 990200

Below are shown one of the important coding that summed up all the total price of physician's fee for specific patient.

```
'========== TOTAL PAYMENT'S TAB ==========

Public Sub update_PhyPayment()

   'Retrieve bill totals - PHYSICIAN'S

   curPhy = 0
   rsPhyDesc.MoveFirst
   Do Until rsPhyDesc.EOF = True
      If rsPhyDesc!Name <> "99999999" Then
         If rsPhyDesc!Name = strSelected Then
            curPhy = curPhy + rsPhyDesc!Price
         End If
      End If
      rsPhyDesc.MoveNext
   Loop
   lblTotalPhy.Caption = Format(curPhy, "#,##0.00")

End Sub
```

Figure 6.5: Coding For Retrieving Bill Totals From Physician's Module.

Agnesia A. Sapati
WET 990200

6.3.3 Treatment's Sub Module Coding

Below are shown the coding for Treatment's Tab control:

```
'========== TREATMENT'S TAB ==========
Public Sub pop_lvwTreBill()

    Dim col As Integer
    Dim row As Integer
    lvwTreBill.Enabled = False
    lvwTreBill.ListItems.Clear
    col = 1
    row = 1
    rsTreDesc.MoveFirst
    Do Until rsTreDesc.EOF = True
      If rsTreDesc!Name <> "99999999" Then
        If rsTreDesc!Name = strSelected Then
          lvwTreBill.ListItems.Add , , rsTreDesc!TrName
          lvwTreBill.ListItems(row).SubItems(col) = rsTreDesc!Date
          col = col + 1
          lvwTreBill.ListItems(row).SubItems(col) = Format(rsTreDesc!Price, "#,##0.00")
          col = col + 1
          lvwTreBill.ListItems(row).SubItems(col) = Format(rsTreDesc!ID, "000000")
          col = 1
          row = row + 1
          lvwTreBill.Enabled = True
        End If
      End If
      rsTreDesc.MoveNext
    Loop
End Sub
```

Figure 6.6: Treatment's Tab control coding.

Agnesia A. Sapati
WET 990200

Below are shown one of the important coding that summed up all the total price of treatment taken by a specific patient.

```
Public Sub update_TrePayment()

   'Retrieve bill totals - TREATMENT'S
   curTre = 0
   rsTreDesc.MoveFirst
   Do Until rsTreDesc.EOF = True
      If rsTreDesc!Name <> "99999999" Then
         If rsTreDesc!Name = strSelected Then
            curTre = curTre + rsTreDesc!Price
         End If
      End If
      rsTreDesc.MoveNext
   Loop
      lblTotalTre.Caption = Format(curTre, "#,##0.00")

End Sub
```

Figure 6.7: Coding For Retrieving Bill Totals From Treatment's Module.

6.3.4 Medication's Sub Module Coding

Below are shown the coding for medication's Tab control:

```
'================ MEDICATION'S TAB ================

Public Sub pop_lvwMedBill()

    Dim col As Integer
    Dim row As Integer
    lvwMedBill.Enabled = False
    lvwMedBill.ListItems.Clear
    col = 1
    row = 1
    rsMedDesc.MoveFirst
    Do Until rsMedDesc.EOF = True
       If rsMedDesc!Name <> "99999999" Then
          If rsMedDesc!Name = strSelected Then
             lvwMedBill.ListItems.Add , , rsMedDesc!medName
             lvwMedBill.ListItems(row).SubItems(col) = rsMedDesc!Date
             col = col + 1
             lvwMedBill.ListItems(row).SubItems(col) = Format(rsMedDesc!Price, "#,##0.00")
             col = col + 1
             lvwMedBill.ListItems(row).SubItems(col) = rsMedDesc!Quantity
             col = col + 1
             lvwMedBill.ListItems(row).SubItems(col) = Format(rsMedDesc!ID, "000000")
             col = 1
             row = row + 1
             lvwMedBill.Enabled = True
          End If
       End If
       rsMedDesc.MoveNext
    Loop
End Sub
```

Figure 6.8: Medication's Tab control coding.

Below are shown one of the important coding that summed up all the total price of medication taken by a specific patient.

```
Public Sub update_MedPayment()

   'Retrieve bill totals - MEDICATION'S
   curMed = 0
   rsMedDesc.MoveFirst
   Do Until rsMedDesc.EOF = True
      If rsMedDesc!Name <> "99999999" Then
         If rsMedDesc!Name = strSelected Then
            curMed = curMed + rsMedDesc!Price
         End If
      End If
      rsMedDesc.MoveNext
   Loop
   lblTotalMed.Caption = Format(curMed, "#,##0.00")

End Sub
```

Figure 6.9: Coding For Retrieving Bill Totals From Medication's Module.

6.3.5 Miscellaneous Sub Module Coding

Below are shown the coding for miscellaneous Tab control:

```
'========== MISCELANEOUS'S TAB ==========

Public Sub pop_lvwMiscBill()

   Dim col As Integer
   Dim row As Integer
   lvwMiscBill.Enabled = False
   lvwMiscBill.ListItems.Clear
   col = 1
   row = 1
   rsMiscDesc.MoveFirst
   Do Until rsMiscDesc.EOF = True
      If rsMiscDesc!Name <> "99999999" Then
         If rsMiscDesc!Name = strSelected Then
            lvwMiscBill.ListItems.Add , , rsMiscDesc!MiscName
            lvwMiscBill.ListItems(row).SubItems(col) = rsMiscDesc!Date
            col = col + 1
            lvwMiscBill.ListItems(row).SubItems(col) = Format(rsMiscDesc!Price, "#,##0.00")
            col = col + 1
            lvwMiscBill.ListItems(row).SubItems(col) = rsMiscDesc!Quantity
            col = col + 1
            lvwMiscBill.ListItems(row).SubItems(col) = Format(rsMiscDesc!ID, "000000")
            col = 1
            row = row + 1
            lvwMiscBill.Enabled = True
         End If
      End If
      rsMiscDesc.MoveNext
   Loop
End Sub
```

Figure 7.0: Miscellaneous Tab control coding.

Below are shown one of the important coding that summed up all the total price of miscellaneous taken by a specific patient.

```
Public Sub update_MiscPayment()

   'Retrieve bill totals - MISCELANEOUS'S
   curMisc = 0
   rsMiscDesc.MoveFirst
   Do Until rsMiscDesc.EOF = True
      If rsMiscDesc!Name <> "99999999" Then
         If rsMiscDesc!Name = strSelected Then
            curMisc = curMisc + rsMiscDesc!Price
         End If
      End If
      rsMiscDesc.MoveNext
   Loop
   lblTotalMisc.Caption = Format(curMisc, "#,##0.00")

End Sub
```

Figure 7.1: Coding For Retrieving Bill Totals From Miscellaneous Module.

CHAPTER 7

SYSTEM TESTING

7.0 Introduction

Testing a program in system development is very important. No matter how good the skill a programmer has, we cannot produced a zero defect software while write once. This is because many systems deal with large number of states, with complex formulas, activities and algorithm. In addition, programmers usually use the tools they are familiar with, to implement what the client needs. But most of the time, clients themselves are uncertain about what they really want. This has resulted the program to rewrite again and again to fulfill clients' needs. Therefore, we cannot make sure that they are no mistake was made during the rewrite process.

When a system has failed, it means that the system does not do what the requirements describe earlier. There are many types of faults that could have happened on a system. This type of faulty will resulted the system to fail.

Therefore, system testing is very important to make sure all the function and method had been developed accordingly, and all the results from the output are correct. In this project, testing has been done step by step from a single unit to a much complicated sub module, before going into the whole billing system.

7.1 Testing Strategies

A testing strategies is a general methodology to the testing procedure rather than on approach of creating a particular system or component test. Testing provide a way to uncover logical errors within the system and to test the system's reliability. There are a variety of tests are available depending on the subject of the tests, which can range from components and groups of components to an entire system.

During system development, testing usually involves several stages. Firstly, each system component is tested on its own, isolated from the other entire component in the system. This recognized as unit testing or component testing. This testing stage verifies that the component is functioning correctly by receiving valid input and generating the expected output.

Upon completion of unit testing, integration testing is then carried out; where the interaction between different system components is tested to ensure that the components can be integrated without any problems. This step, also known as module testing, verifies that all the components are able to function together as per the module or system design specifications. It also ensures that the interfaces among the components have been defined and handled properly.

System testing is finally performed to ascertain that the entire system is working accordingly to the user's specifications. In this testing stage where the system will be checked against the user's requirements specifications, the users will be joined by the developers to carry out the system tests. If a change is required, especially in the event where the users' requirements are not meet as per the specifications, modification to the system will then be carried out. The system is ready for deployment as soon as the users are satisfied with the system.

7.2 Bottom-up Testing

In the case of this system, bottom-up testing is performed on the system itself as soon as integration between all the various Health Information System components have been done. This testing strategies is also applied to the testing of the modules themselves separately – the lowest – level sections of the modules are tested first, after which only are they integrated together and testing goes on to a higher level, and goes on recursively until it reaches the highest level.

7.3 Test Case Design

During component testing, two types of testing are done – white box testing and black box testing.

For white box testing, the test are all directly involved with the structure of the code within a module or code segment. The code is analyzed and inside knowledge regarding the structure of the component are used to derive test data. Code coverage for white box testing is defined in terms of the following types: segment coverage, node testing, compound condition coverage, basic path testing, data flow testing, path testing and loop testing.

In the case of black box testing, it is assumed that the logic structure of the code is unknown, and it is at this point that the function of the module is tested. A few tests can be carried out in black box testing – error guessing, equivalence classs partitioning, boundary value analysis, cause-and-effect graphing, domain testing and module-interface testing.

7.4 Unit Testing

This form of testing focuses on the smallest unit of the software design, which is the module. A module is a collection of components, which are independent from each other. After each component units have been tested, the interactions between these components are tested. Unit testing is usually done concurrently with the prototyping phase of the project's development timespan. For example, the Add New Doctor function on the Physician module is one of the submodules with many subfunctions, like insertion of new doctor data into the database and error handling, and these functions are tested ethaustively to ensure that the module is bug free.

Examples of units which are independently tested include:

a) Opening and closing database connections.
b) Inserting new records into the database.
c) Modifying existing records in the database.
d) Validating the user during Login to check access authorization.
e) Execution of SQL statements.
f) Returning query results from the search functions.

7.5 Module Testing

The objective of this form of testing is to take modules, which have already undergone unit, testing and build program structures, which encapsulate all the related modules. This testing ensures that the module-calling sequence within the system will be systematic.

In module testing, two or more units where either unit, which uses output data from or provides input data for another unit are tested. These units have related characteristics in which to perform a common goals or function, such as the editing a treatment which comprises retrieval of data from various tables, updating the treatment information and finally displaying the treatment itself.

7.6 Test Cases

To check on whether the output of this billing system is correct or not, some test cases has been develop to make sure that the input is properly converted to the desired output. The input of the test cases has been chosen so that the output demonstrates something about the behavior of the code.

In most of the cases, error message has been outputted to the user to warn about the error that occurred during the running of the program. When doing testing with the test case, all the possibility should be tested to determine the program is really free from error.

In this project, the test cases have been divided to two parts according to the test objectives. First objective is to demonstrate that all statements execute properly. For example, code will be executed to see whether proper screen will be displayed to the user when user presses on a button or selecting an item from the menu. Second objective is to show that every method performed by the code is done correctly. For example, when user input the price of medication per unit, it will total up together with the physician, treatment and miscellaneous item in the total payments section.

A wide variety of test cases were done on the modules and the system. Only a few test cases will be shown here:

7.6.1 Test Cases On Login & Logout Sub Module

The *Login & Logout* sub module is the first module that users will have touch with. An error in this module will prevent users from going inside the billing system at all. Therefore, testing on this module is very important to make sure that this program will run smoothly.

For this module, the test cases have been focused on some of the element listed below:

1) Number of user and password's characters.
2) Incorrect password confirmation.

7.6.2 Test Cases on Add New Physician Sub Module

The *Add New Physician* sub module will allow user to add new patient to the list of patient's name.

Testing on this sub-module will focus on the below cases:

1) Adding a new doctor's name together with their contact number and specialties.
2) Adding an existing doctor under the chosen patient.

7.7 Integration Testing

When module testing has achieved certain degree of success and meets the objectives, the sub modules are combined into a working system. The integration is planned and coordinated so that when a failure occurs, the cause of the failure can be identified much more easier.

7.7.1 Bottom-up Integration

In this project, the integration testing has been done by using the bottom-up testing. When this method is used, each component or unit at the lowest level of this billing system hierarchy is tested individually first. Then the next components to be tested are those who call the previously tested ones. This approach is followed repeatedly until all components are included in the testing.

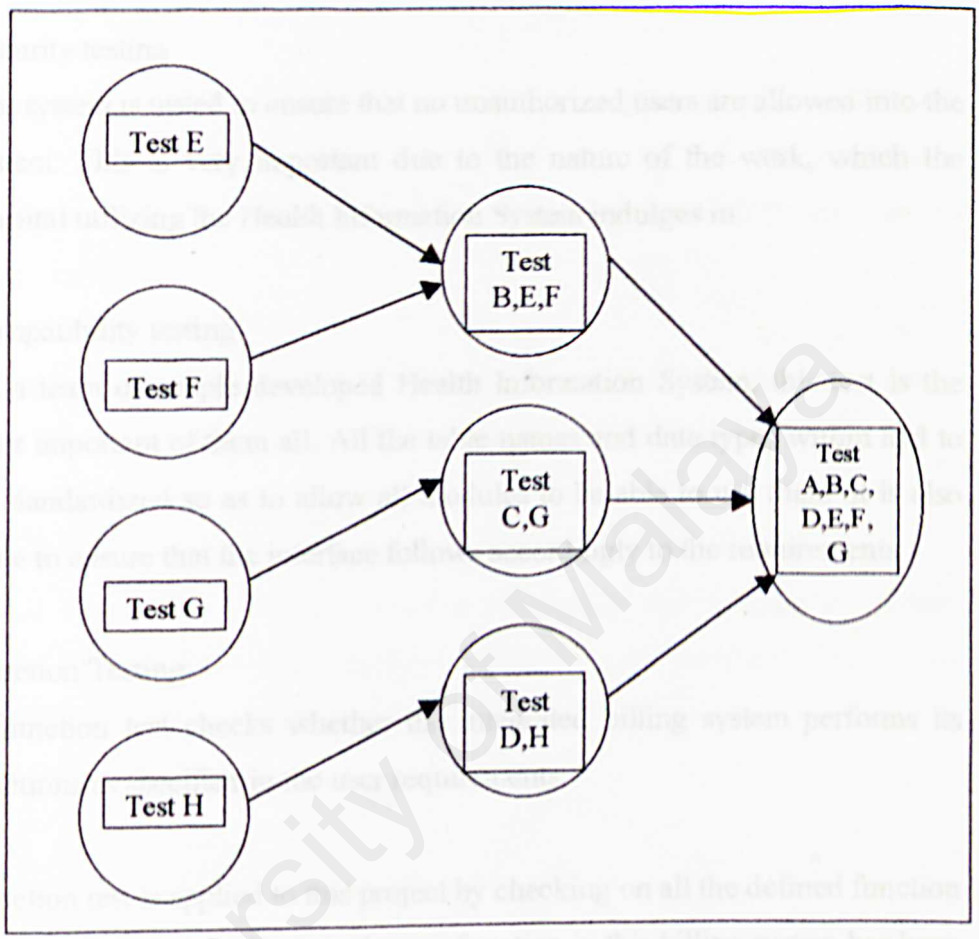


Figure 7.0: Bottom Up Integration Model

7.8 System Testing

The final test procedure is known as system testing. Once all the modules have been tested, they are then tested together to ensure that all the elements are functioning and interfacing with each other as expected. The results of thesis tests will show whether the entire system specifications and objectives have been reached.

Some of the system tests performed are listed below:

a) Security testing

The system is tested to ensure that no unauthorized users are allowed into the system. This is very important due to the nature of the work, which the hospital utilizing the Health Information System indulges in.

b) Compatibility testing

As a team of people developed Health Information System, this test is the most important of them all. All the table names and data types within had to be standardized so as to allow all modules to be able to use them. It is also done to ensure that the interface follows accordingly to the requirements.

c) Function Testing

A function test checks whether the integrated billing system performs its functions as specified in the user requirements.

Function test is applied to this project by checking on all the defined function in this billing system. Each and every function in this billing system has been tested to obtain the result for faulty correction purposes.

d) Performance Testing

Performance testing is run to compare the integrated components with the non-functional requirements. These requirements including security, accuracy, speed and reliability, constrain the way in which the system functions are performed.

If the billing system has passed the test, it will now considered a verified system in which this is the designers' interpretation of the requirements specification. Next this billing system will be compare with the user

requirements. If the program has meet the requirements, this program is now a validated program.

e) Acceptance Testing

Acceptance testing is created to assure to the clients that the system was specially built for them. The acceptance test is run in the real environment but often it will be tested on a system with test facility different from the target location.

For this project, the acceptance test has been run in the project lab to test on how the system will perform and also to test on different system.

f) Installation Testing

Installation testing is to allow users to exercise the program's function, environment in order to document and collect users review and comment on this program. The review and comment will be use to make necessary changes to suit the users needs.

For this project, the installation test has been held in the project lab by some of the tester. The feedback from the tester has been received to further enhance the program.

CHAPTER 8

SYSTEM EVALUATION

8.0 Overview

System evaluation is implemented by more than simply comparing the information obtained with the information, which is expected. It is also related to the user environments, attitudes, information priorities and several other matters which must be given consideration before the actually efficacy can be concluded.

Evaluation is a process, which occurs during all the phases of a system's development. It is a continuous process, drawing from a wide variety of sources of information.

The role of the evaluation phase is to determine:

a) The extent to which the expected outcome has been realized.
b) The value of the outcome with all factors taken into consideration.

8.1 Introduction

System evaluation is very important stage during whole the system development process. This stage has to be done before the full system can be delivered to the users. Evaluation that related to user environment will emphasize on implementation problems and user involvement.

Evaluation was related to user environment, attitudes, information priorities and several other concerns that are to be considered carefully before effectiveness can be concluded.

Throughout the system development life cycle, the entire member involved in this project will work together to evaluate the system in order to give feedback for the eventual improvement.

There are several approaches for evaluating system and it can turned out to be a fruitful technique for measuring the success of a developed system. These approached are listed below:

- Possession utility – who should receive the output
- Form utility – what kind of output to distribute to the decision maker (can be either the user itself or the administrator)
- Place utility – where the information is distributed
- Time utility – when the information is delivered
- Actualization utility – how this information is introduced and used by the decision maker.
- Goal utility – answer the question on whether the output has value on helping user to achieve the objectives.

8.2 Present Enhancements

This is the important part for this chapter because there's big changing from the system proposal and the real system that has been build. One main reason why there is a big changing is that, a very limited module has been proposed.

Below are described some changing that has been done and few new enhancements that has been made to make this system more adaptable to the outside real environment especially for hospital environment.

8.2.1 Change in Database Software

This system's database is made from Microsoft Access 2000 and not Microsoft SQL Server 7.0 as been proposed in the early chapter.

Reason: Amount of the datas that need to enter is not so many and Microsoft Access 2000 already can handle for this kind of amount.

8.2.2 Additional Module

A few new modules have been added to this system because of too limited module exists. Some of the important modules that has been add are:

1) *The Doctor's Module:*
   - ✓ Add and edit doctor's profile for specific patient.
   - ✓ Display all the doctor's profile for each patient.
   - ✓ One patient can have more than one doctor in each bill.
   - ✓ Summed up total fee that charge by the doctor.
   - ✓ The total fee will be added to the bill as Physician's total amount.

2) *The Treatment's Module*
   - ✓ Add and edit treatment's profile for specific patient.
   - ✓ Display all the treatment's profile for each patient.
   - ✓ One patient can have more than one treatment in each bill.
   - ✓ Summed up total price for all the treatment that taken by the patient.
   - ✓ The total price will be added to the bill as Treatment's total amount.
   - ✓ Example treatment: X-Ray, Blood Check, etc.

3) *The Medication's Module*
   - ✓ Add and edit medication's profile for specific patient.
   - ✓ Display all the medication's profile for each patient.
   - ✓ One patient can have more than one medication in each bill.

- ✓ Summed up total price for all the treatment that taken by the patient.
- ✓ The total price will be added to the bill as Medication's total amount.
- ✓ Example medication: Paracetamol, Piriton, etc.

4) The Miscellaneous Module:

- ✓ Add and edit miscellaneous item's profile for specific patient.
- ✓ Display the entire miscellaneous item's profile for each patient.
- ✓ One patient can have more than one miscellaneous item in each bill.
- ✓ Summed up total price for all the miscellaneous item that taken by the patient.
- ✓ The total price will be added to the bill as miscellaneous total amount.
- ✓ Example miscellaneous item: Towel, Bandages, etc.

8.2.3 Change in System Interface

The interface of the system also has been change greatly. Some of the reasons are listed below:

- ❖ To adapt to the few additional module that has been add.
- ❖ Old interface are not suitable for stand-alone system.

Below are shown some of the new main interface that has been created:

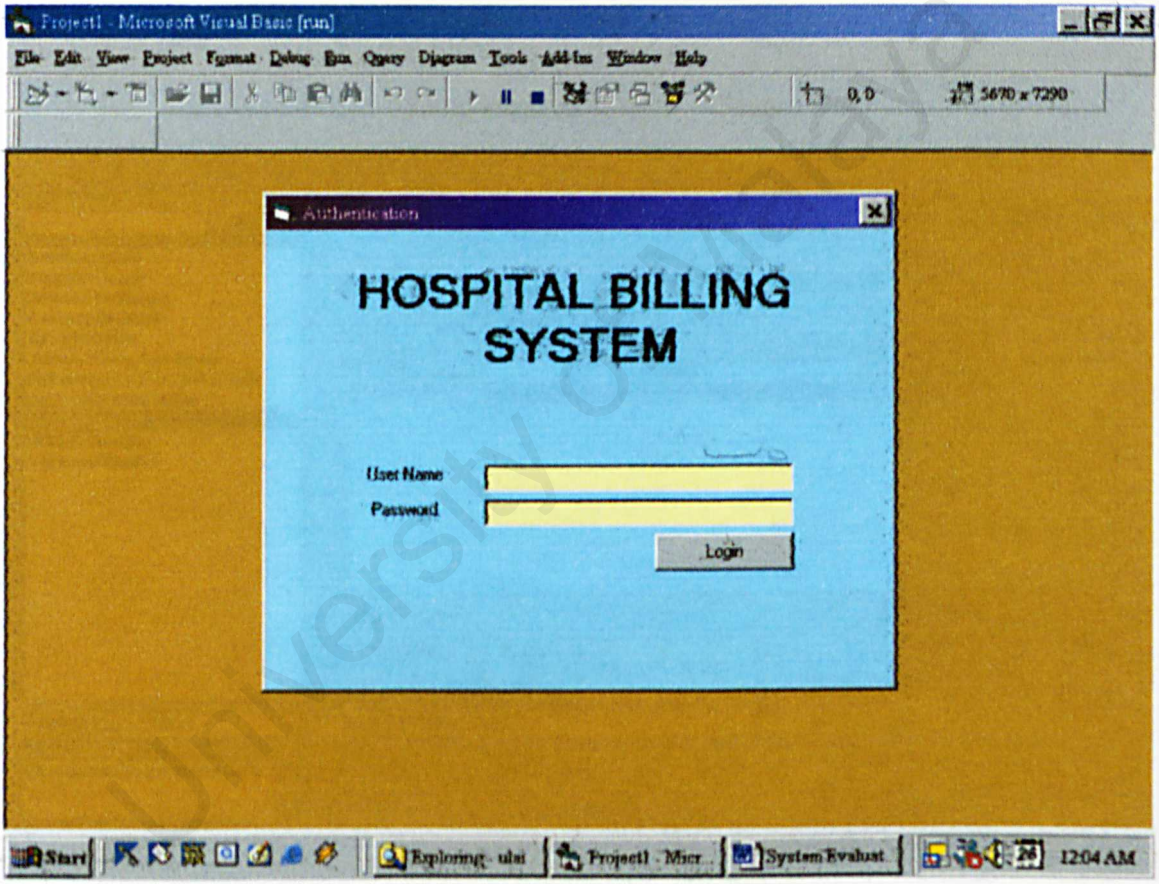


Figure 8.0: New Login Page

Agnesia A. Sapati
WET 990200

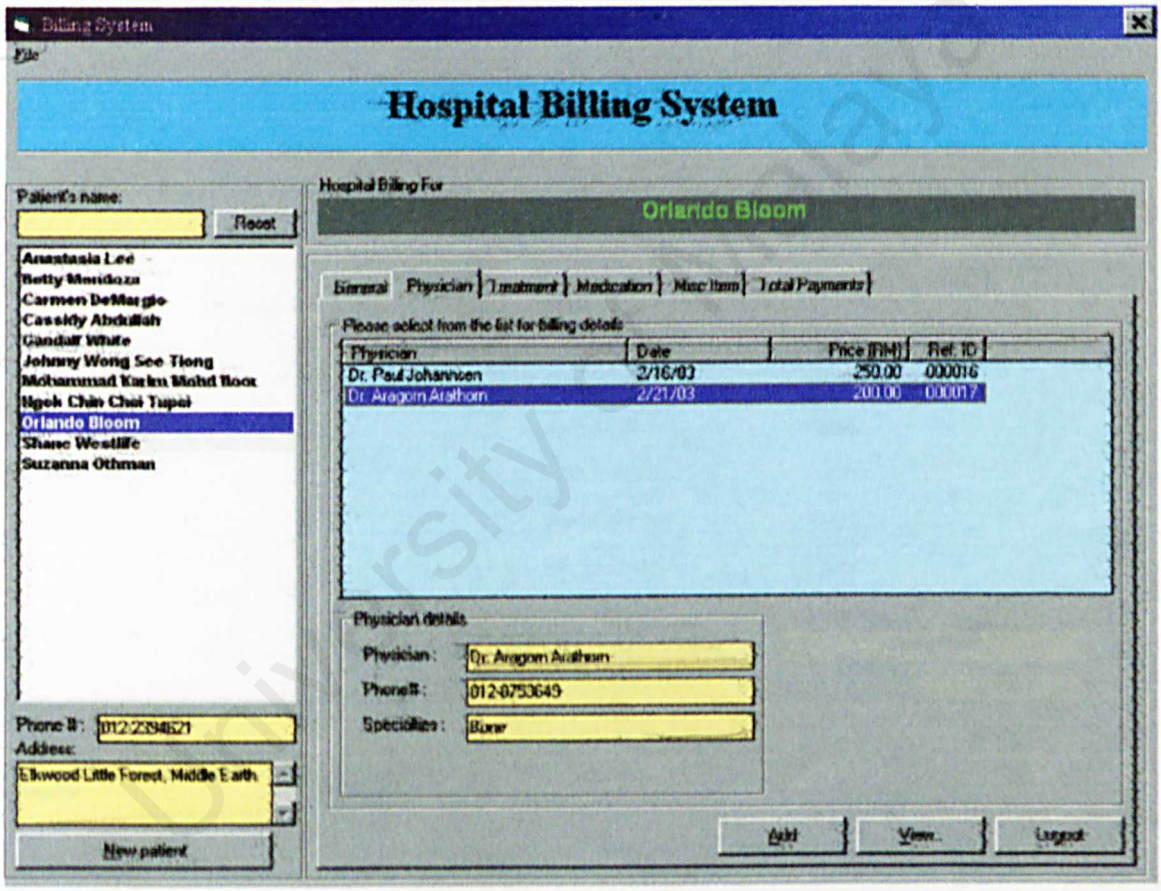


Figure 8.1: New Main Page

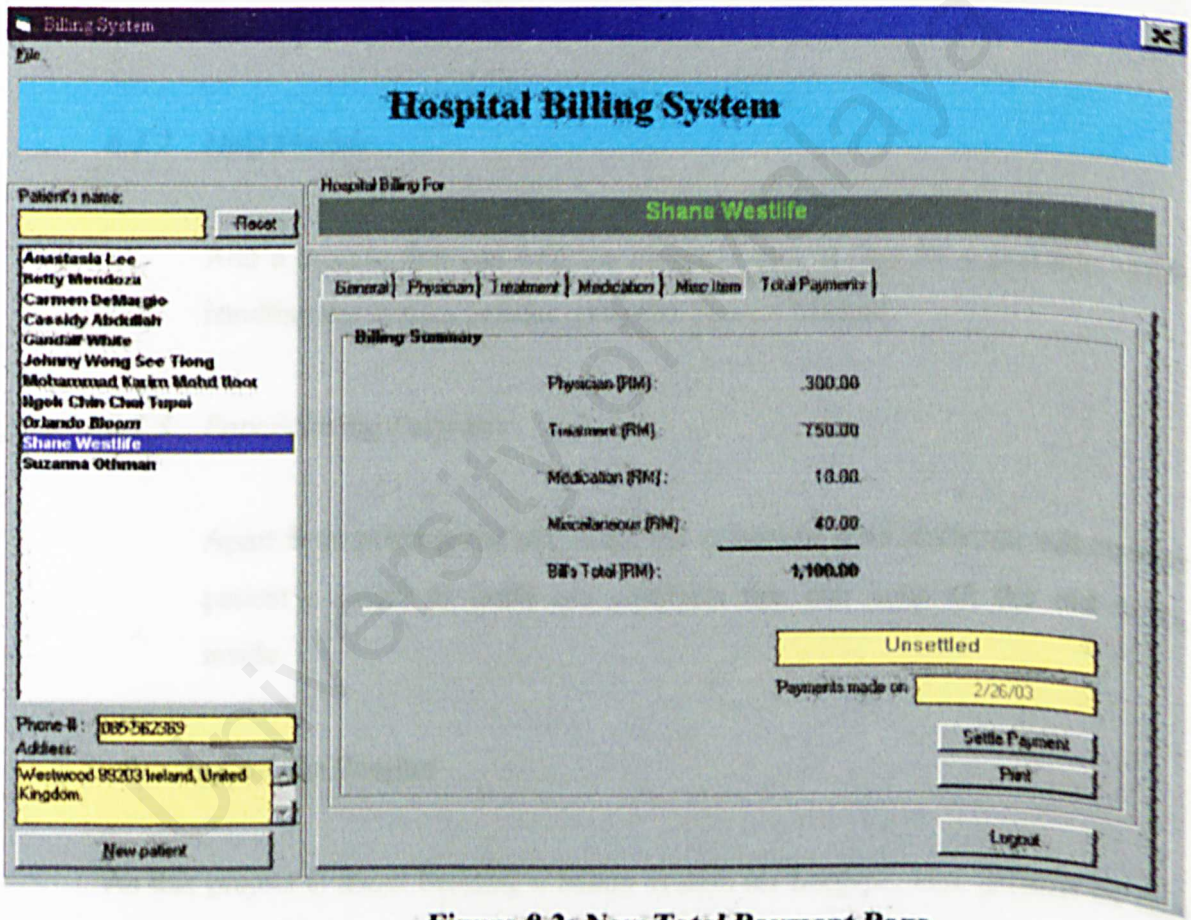


Figure 8.2: New Total Payment Page

8.3 Future Enhancements

8.3.1 Profit and Lose Management Module

Add a module that can sum up all the profit and lose that gained by the hospital for each day.

8.3.2 Help Module

Add a module that can help the system's user if they have problem with handling the system besides using the System Manual.

8.3.3 Paper-Saving Purposes

Apart from printing out bill, make the system to send electronic bill to the patient's e-mail or build one database that can keep all the old bills inside.

8.4 Strength of the Project

As this project is about building a billing system for hospital, here is some of the strength that this system has managed to reveal.

8.4.1 Username and Password Validation

This system has been designed with a username and password authentication for the user. This means that the security has been assured.

8.4.2 Summed-Up Total Price

This system can sum up the entire price from each department automatically without user's manual help.

8.4.3 Printing Out Bill

This system can print out bill for the patient to keep as a prove that they have pay their bills.

Agnesia A. Sapati
WET 990200

Conclusion

This project has succeeded in achieving the objective set out for it. The system has successfully implemented all the old and new module together to become a complete system.

Programming skills are not the only requirement for developing the system even it is the most important basic skills that needed to build one system. System design is also a very important and crucial part, one where which time has been spent on to ensure that the system, which is developed, is as error free as possible.

Throughout the period of this project, many things were learnt in the process of developing this system apart from learning up Visual Basic and Access. One of the things is time management. Time management is very important in developing one system.

Finally, this Billing System has managed to build a strong understanding on system development and this will be a useful tips and experience for future work. In other words, it prepared a path to all undergraduate students to take one step into the world of jobs and career.

References

- ✓ Jeffrey L. Whitten, Lonnie D. Bentley, & Kevin C. Dittman. (2000). **System Analysis and Design Methods.** (5th ed). New York: McGraw-Hill Irwin.
- ✓ Joe Salemi. (1995). **Guide to Client / Server Databases.** (2nd ed). Ziff-Davis Press.
- ✓ Pfleeger, Shari Lawrence. (2001). **Software Engineering**. (2nd ed). Prentice-Hall, Inc.
- ✓ Hartfield, Bill. (1996). **Developing PowerBuilder 5 Applications**. (4th ed). Sams Publishing.
- ✓ Dietel & Dietel, Nieto, T.R. (1999). **Visual Basic 6 – How To Program**. Prentice-Hall, Inc.
- ✓ E. Burrows, William, Langford, Joseph D. (2000). **Programming Business Applications with Microsoft Visual Basic 6.0.** Von Hoffmann Press, Inc.
- ✓ MacDonald, Michael, Cagle, Kurt. (1999). **Visual Basic 6 – Client / Server Programming.** The Coriolis Group, Inc.
- ✓ L. Oxford, Kristie. (2001). **Microsoft Access 2000 with Visual Basic for Applications.** Course Technology.
- ✓ Duffy, Tim. (2000). **Microsoft Office 2000 Professional**. (Brief Edition). Prentice-Hall, Inc.

- ✓ Access 2000 Tutorial / Technology Skill Orientation,
  http://www.fgcu.edu/support/office2000/access/

- ✓ Visual Basic Code / Visual Basic Specific Search Engine
  http://www.vbcode.com/

- ✓ Visual Basic Coding
  http://www.planet-source-code.com/vb/default.asp?lngWId=1

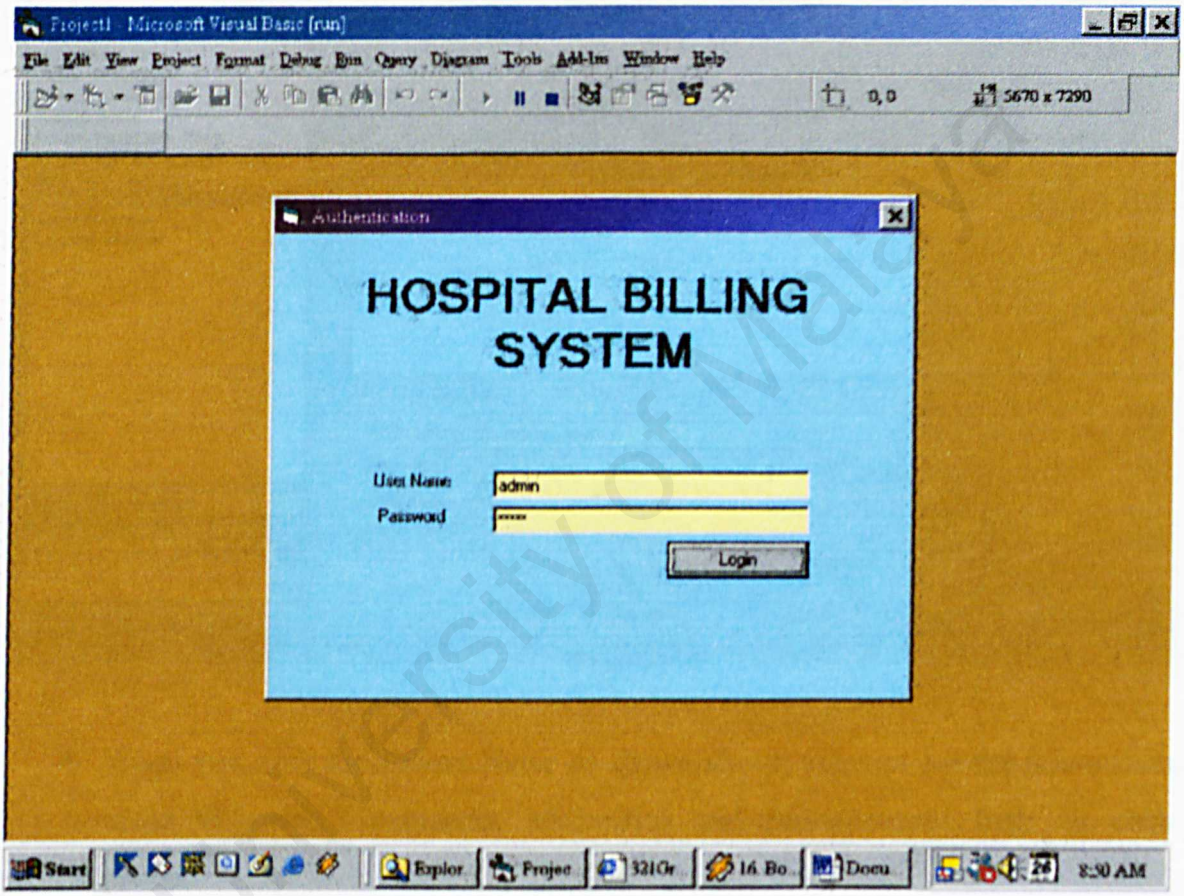
- ✓ Visual Basic Source Coding
  http://www.a1vbcode.com/

APPENDIXES

Appendix 1

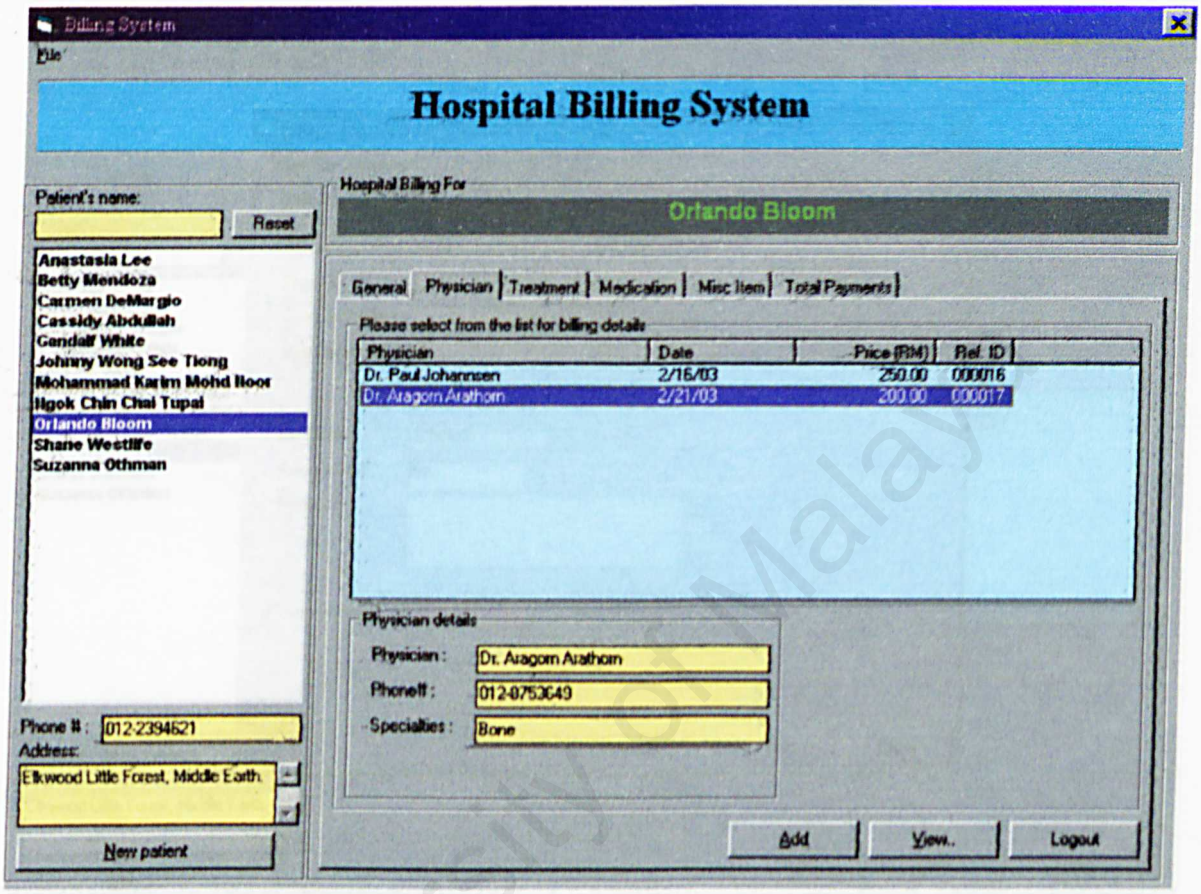
Hospital Billing System's User Manual

1.0 Login Page



Key in your Username and Password that given by the administrator. Click the '**Login**' button.

1.1 Main User Page – Physician Page



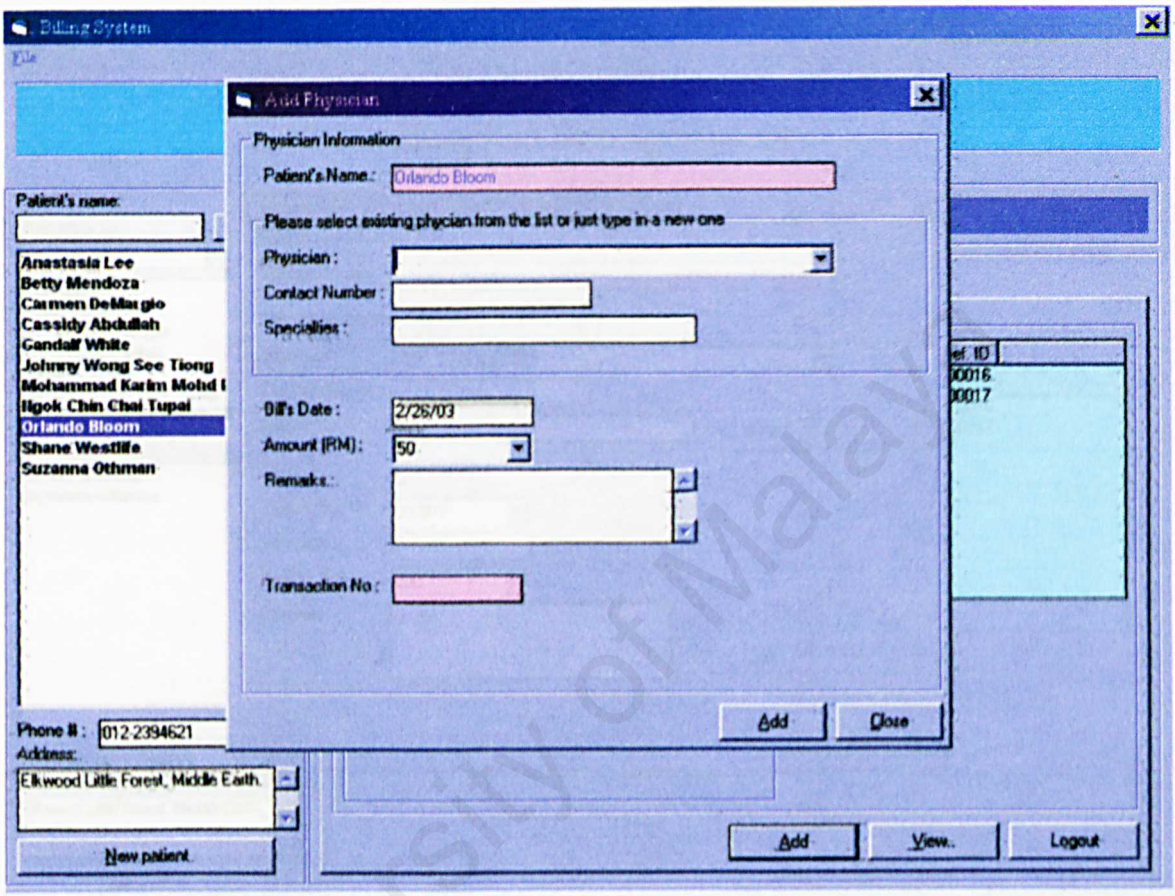
- When you click the Patient Name as shown above, you can see the information about Physician, Treatments, Medication and Miscellaneous Item for each patient.

- Above, in the Physician's Module you can see the entire doctor that has treated the patient.

- When you click the **'New Patient'** button, you can a new patient record to the system.

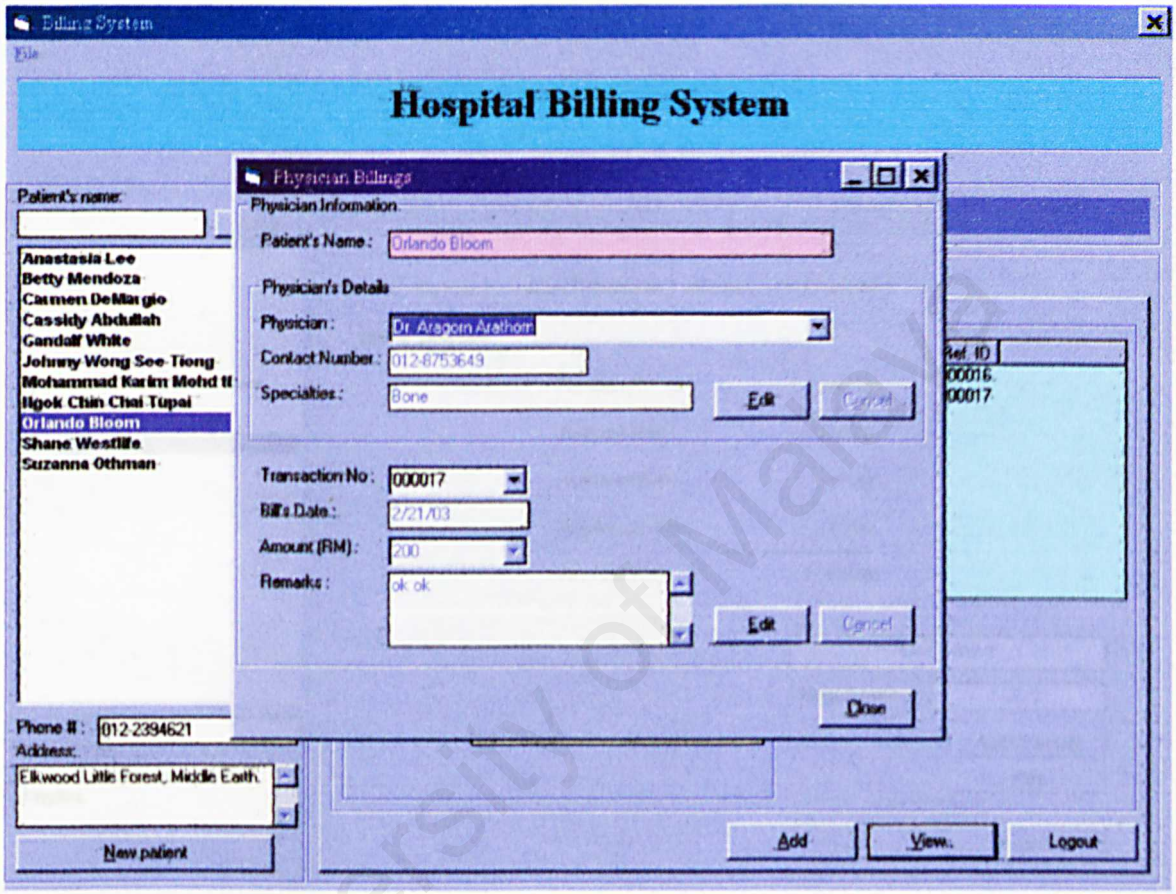
- You can search the patient record by enter the patient name inside the **'Search Box'** that situated above the list of Patient Name.

1.2 Add Physician Page



❖ In the Add Physician Page, you can add a new doctor under that patient or you just select from the list of existing doctor that have been provided.

1.3 View Physician Page



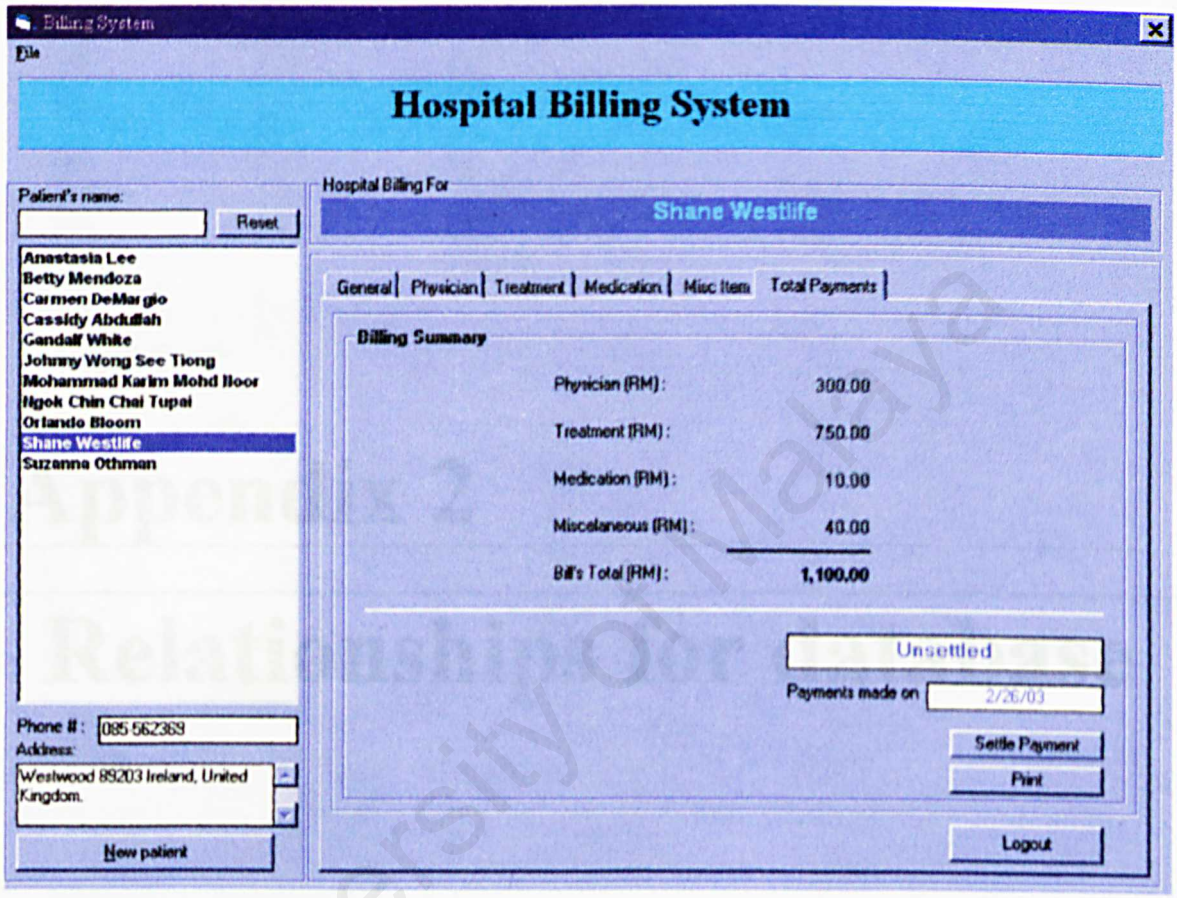
- In this page you can edit the doctor's profile and save it into the system.

****

For Treatment, Medication and Miscellaneous Item, the procedure is just the same with the Physician Page.

****

1.4 Total Payment and Printing Page



- In this page you can see all total up amount for one bill for each patient.
- You click the **'Settle Payment'** button when the patient has pay the bill.
- The date will automatically will follow the date that day.
- Click the **'Print'** to print the bill.

****

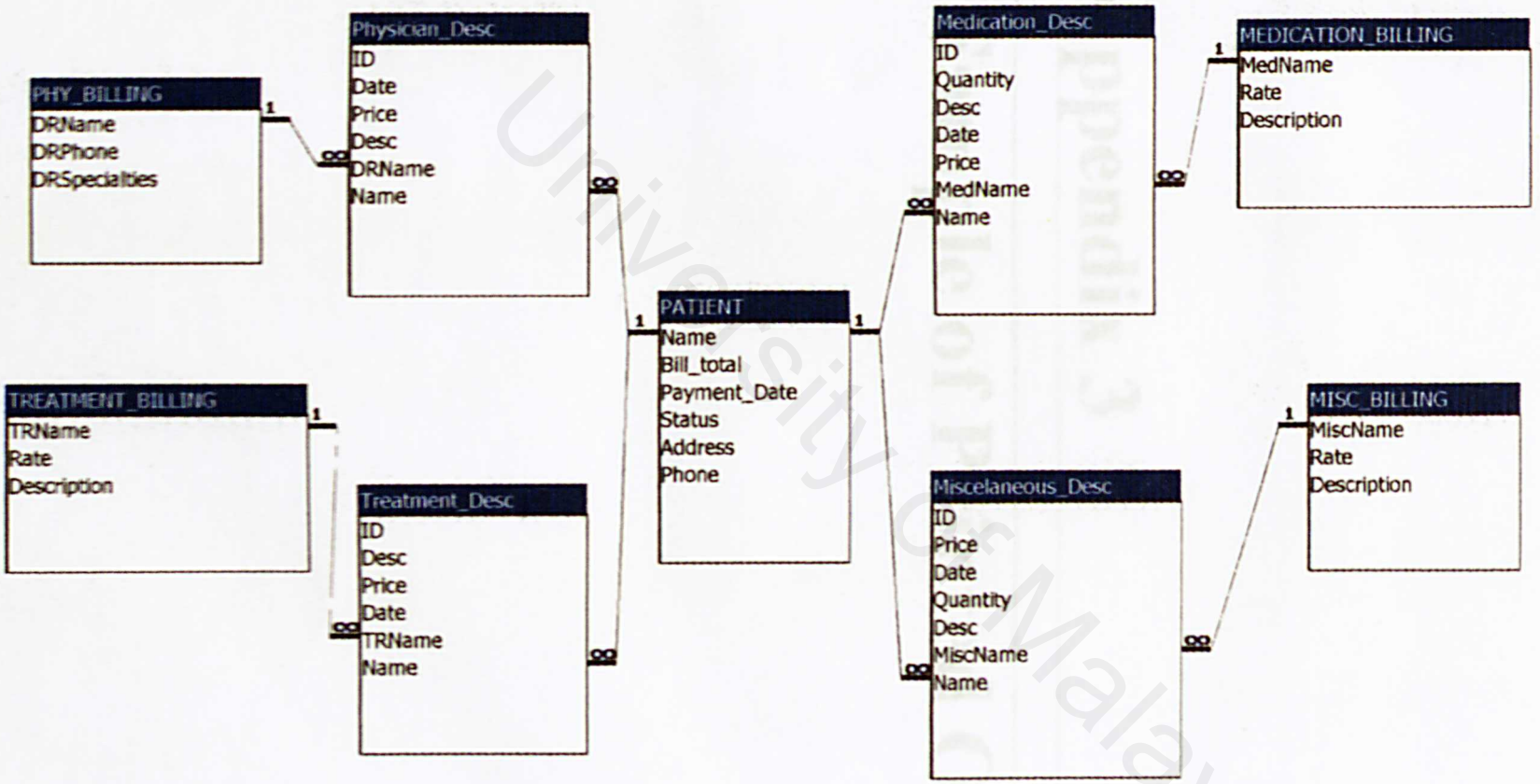
To close the system, just click 'Logout' button.

****

Appendix 2

Relationships for database

Relationships for database
Thursday, February 27, 2003



Appendix 3

Sample of Printed Out Bill

BILLING SUMMARY

Date: 2/27/03

e : Orlando Bloom

ess: Elkwood Little Forest, Middle Earth.

e No: 012-2394621

Physician (RM) :	450.00
Treatment (RM) :	850.00
Medication (RM) :	304.00
Miscelaneous (RM) :	70.00
Bill's Total (RM) :	**1,674.00**

ceived By:

Patient's Signature: