UNIVERSITI MALAYA

HEALTH INFORMATION SYSTEM

Disediakan oleh:

MOHD FAZLEE GHOUSE BIN MOHAMAD KHADER

WEK 990397

SESI 2002/2003

PROJEK ILMIAH
TAHAP AKHIR II

Supervisor:

DR. DILJIT SINGH A/L BALWANT SINGH

Moderator:

PUAN NORIZAN BT. MOHD. YASIN

ABSTRACT

Health Information System or HIS is a database electronic system designed specially to manage information regarding patients and medicine in an outpatient department. This Health Information System also acts as a computerized inventory system to manage medicine in outpatient department that offers health care treatments.

This Health Information System is using the stand-alone approach. It is designed based on a user-friendly concept. It also has an interesting interface design and also a professional look that is easy for users to use the system and to manipulate the data stored in the system.

Hopefully this system that uses a back-end database would be able to replace the manual system for managing patient information. Users will no longer have to use the manual traditional recording method that takes time to find and to update records and might also result in data redundancy in recording patients' information. Besides this, HIS is also equipped with security preferences that will protect the integrity of the system by only allowing authorized users to access the system. HIS is also able to reduce paper requirements in recording data towards developing a paperless environment.

ACKNOWLEDGEMENTS

In the name of ALLAH the Almighty

This Latihan Ilmiah II Project is finally complete in the period of time given as friends and family provide technical and emotional support. It is impossible to list here all those who helped to sustain me during the writing and revising, and I apologize in advance for any omissions.

This success would not have been accomplished if this project were being done individually without help and guidance from others. Many thanks to those who had help me by giving ideas and suggestions during the development of this report.

Dr. Diljit Singh a/l Balwant Singh was a terrific supervisor, suggesting ways to clarify and simplify, giving support and guidance, and helping me to produce a tighter more efficient report. Thank you to Puan Norizan Mohd. Yasin for being the moderator of this project and for giving supports, comments and suggestions for this project.

Finally, to my parent that has given a lot of support and guidance. Fazlin Haney, Khairul Zaki and all my friends that is involved in developing this system. Thank you very much.

LIST OF TABLES

TABLE NAME **PAGE**

Table 1.1: Phase Descriptions 7

Table 4.3.1: Table for PATIENT 35

LIST OF FIGURES

FIGURE NAME **PAGE**

Figure 1.1: Project Development Plan 7

Figure 1.1: Gantt chart for Project Development Plan 8

Figure 2.2: Interface for Procare 2000 13

Figure 2.3 (a): Interface for Pantai Medical Care 15

Figure 2.3 (b): Form Interface for Material Management of Pantai Medical 15

Figure 3.1: Project Development Phase 19

Figure 4.1: Structure Chart for Health Information System 30

Figure 4.2 (a): Context Diagram 31

Figure 4.2 (b): Data flow diagram for patient visit 32

Figure 4.4 (a): Example of User Login Interface 36

Figure 4.7 (a): Example of Patient Information Interface 37

Figure 4.7 (b): Example of payment collection user interface 37

Figure 6.1: Testing Steps 49

Figure 6.2: Unit Testing Scheme 54

CONTENTS

Abstract i

Acknowledgements ii

List of Tables iii

List of Figures iv

Contents v

CHAPTER 1: INTRODUCTION

1.1 Project Background 1

1.2 Project Overview 2

1.3 Project Objectives 4

1.4 Project Scope 4

1.5 Importance of Project 5

1.6 Target Audience 6

1.7 Project Planning 6

1.8 Statement of Expected Outcome 8

CHAPTER 2: LITERATURE REVIEW

Section I: Review and Writing

2.1 Role of Literature Review 9

2.2 System Definition 9

2.3 Findings / Review Methodology 10

2.3.1 Approach to Literature review 10

2.3.2 Writing Method 11

2.4 Manual System Weakness 12

2.5 Advantages of a Computerized System 12

Section II: Existing System Review

2.6 Procare 2000 13

2.7 Pantai Medical Care 15

CHAPTER 3: SYSTEM DEVELOPMENT METHODOLOGY

3.1 Introduction 17

3.2 Methodology 19

3.2.1 Phase I: Early Review 20

3.2.2 Phase II: System Analysis 21

3.2.2.1 System Analysis 22

3.2.2.2 System Requirement Analysis 22

3.2.2.3 Software Requirement Analysis 25

3.2.2.4 Hardware Requirement Analysis 27

3.2.2.5 Interface Design Analysis 27

3.2.2.6 System Ability Analysis 28

3.2.3 Phase III: System Design 28

3.2.4 Phase IV : Programming / Coding 29

3.2.5 Phase V : Testing and Maintenance 29

CHAPTER 4: SYSTEM DESIGN

4.1 Introduction 30

4.2 Structure Design 31

4.3 Process Design 32

4.4 Database Design 34

4.4.1 Database Table 35

4.5 Interface Design 36

CHAPTER 5: SYSTEM IMPLEMENTATION

5.1 Introduction 39

5.1.1 Developing and Testing Network and Database 39

5.1.2 Developing and Testing Program 40

5.1.3 Install and Test the New System 40

5.1.4 Delivering the New System for Operation 41

5.2 Developing Health Information System 41

5.2.1 Contents Preparation 41

5.2.2 Content, Presentation and Programming Integration 42

5.2.3 Coding Phase 43

5.2.3.1 Coding Method 44

5.2.3.2 Coding Approach 45

5.3 Summary 45

CHAPTER 6: SYSTEM TESTING

6.1 Introduction 47

6.2 Types of Testing 48

6.2.1 Unit Testing 48

6.2.2 Module and Integration Testing 48

6.3 Types of Faults 49

6.3.1 Algorithmic Faults 49

6.3.2 Syntax Faults 50

6.3.3 Documentation Faults 50

6.4 System Testing 50

6.4.1 Unit Testing 51

6.4.2 Integration Testing 52

6.4.3 Function Testing 53

6.4.4 Performance Testing 54

6.4.5 Acceptance Testing 54

6.4.6 Installation Testing 55

6.5 System Testing Technique 55

6.6 Creating *Executable and Set-Up* File 56

6.7 Summary 57

CHAPTER 7: MAINTENANCE AND SYSTEM EVALUATION

7.1 Overall System Review 58

7.2 System Maintenance 59

7.2.1 Maintenance Requirements 60

7.2.2 System Maintenance Methodology 60

7.2.3 Disaster Recovery Plan 60

7.3 System Evaluation 61

7.3.1 Introduction 61

7.3.2 Objective Achievement 61

7.3.3 Problems and Solutions 61

7.3.3.1 Problem in System Design 62

7.3.4 User's Feedback 64

7.3.5 System Strength 65

7.3.5.1 More Effective 65

7.3.5.2 Security Preferences 65

7.3.5.3 User Friendly 65

7.3.5.4 More Flexible 66

7.3.5.5 Current Trend 66

7.3.6 System Limitations 66

7.3.6.1 Limited Help System 67

7.3.6.2 Acceptance Test 67

7.3.7 Future Enhancements 67

7.4 Summary 68

OVERALL SUMMARY 70

BIBLIOGRAPHY 69

USER MANUAL

CODING

CHAPTER 1 INTRODUCTION

1.1 PROJECT BACKGROUND

Traditional ways of carrying out health care center daily tasks involved huge amount of paper documents, resulting in a lot of overhead and inefficiency. The need of a more efficient system resulted in the development of Health Information System.

The Health Information System that will be developed is a management tool designed specially to implement efficiency at health centers. The main purpose of the system is to make it easy to manage information at health care centers and at the same time to provide needed information to make strategic decisions.

Health Information System is going to be a Windows application, and is going to be developed using Visual Basic 6.0 that has many features found in object-oriented environments.

For this project, the system that will be developed uses a *back-end* database. This is because user will have to update the records every time a patient comes for check up. This includes recording patient information, diagnostic, drugs given and payment collection. This system is going to be developed to replace manual system for managing outpatient information system. User do not have to use the old ways in recording information and it only takes a few seconds to update and refer to a certain record.

1.2 PROJECT OVERVIEW

There are three main topics contain in the project overview. It is System Layout, Modules and Overall Workflow.

Modules

The Health Management System is divided in 6 main categories

1. Registration
2. Patient
3. Billing
4. Pharmacy
5. Payment Collection
6. Maintenance

Overall Workflow

Registration

- Patient Information
- Allergies
- Guarantor Details
- Financial Information

Consultation

- Pharmacy Order Entry
- Investigations
- Minor Procedures

Order Entry

- Pharmacy Order Entry
- Investigations
- Charges

Laboratory

- Request Slip
- Acknowledge Order

Pharmacy

- Pick List
- Prescription

Diagnostic

- Request Slip
- Acknowledge Order

Billing and Payment

- Billing Personal
- Billing Corporation
- Bill Payment
- Receipt

System Layout

The Health Information System is a *stand-alone* system. This means that any process of eliminating, updating or addition of records must be done on the same computer.

1.3 PROJECT OBJECTIVES

The objectives of this project are:

- Develop a paperless patient information recording.
- Introduce a more systematic and effective way of managing patient information using computer
- Streamlining the treatment flow of a patient in the health care centers, while allowing doctors and other staff to perform to their peak ability, in an optimized and efficient manner.

1.4 PROJECT SCOPE

There are many type of Health Information System that exists. There are systems develop to calculate soil in lands, system to evaluate air and water pollution, system design for hospitals and health information system used in laboratory to check the toxicity of air in the lab. The system that is going to be developed has narrowed to the scope of outpatient that will be used by clinics and other health care departments. The system that is going to be developed will provide computerized management of patient information for health care centers. The project scope is divided into six main categories for outpatient. That is registration, patient, billing, pharmacy, payment collection and maintenance.

a) Health Information System

- This is a standalone system and it is not using an online system. All operation involving updating, adding, checking and deletion of records from database is done by the user.
- Transactions involve registration, diagnostic, information recording, drugs dispensing and payment.

b) Language

This system is using English language so that is much more easier for the records to be transferred to a different health care centers incase needed.

c) User

This system is going to be used by front desk staff of health care centers. The user will be able to delete, add, update or check patient's record whenever a patient comes to the health center for treatment or checkup.

1.5 IMPORTANCE OF PROJECTS

Healthcare is one of the most important aspects of human life. This Health Information System project is developed to help manage patient's information for health care centers while allowing doctors and other staff to perform to their peak ability, in an optimized and efficient manner.

1.6 TARGET AUDIENCE

Health Care Center Owner

This Health Information System is intended for health care center owner that offers health care treatment. The system is intended to provide an easier and more systematic way of handling patient's record. Main processes need to be done using this system is:

- Store patient's information
- Update, delete and add patient's record.

1.7 PROJECT PLANNING

This project begins on the third semester of 2001/2002. Topic confirmation date is 18 March 2002. This project is divided into two phases. The first phase will be executed this semester and the other phase will be executed next semester. The first phase involves System Analysis and System Design. System Analysis began from 15 March 2002 and ends on 30 April 2002 whereas System Design start from 1 April 2002 and ends on 7 June 2002.

The second phase involve coding and testing process. Coding and testing process will strart from 8 Jun 2002 and ends on 30 September 2002. Documentation process is being done all the way. Table 1.1 shows the summary for each phase while Figure 1.1 shows a Gantt Chart for overall project schedule.

Phase	Activity
1. Early Review and System Analysis	- Determine system objectives - Determine system needs - Provide project schedule - Choose system development model
2. System Design	- System interface design
3. Execution / Coding	- Learning Visual Basic 6.0 - Learning Microsoft Access
4. System Testing	- Design test data - Testing modules - Compare test result with the real result
5. System Maintenance	- Improved changes for system

Table 1.1: Phase descriptions

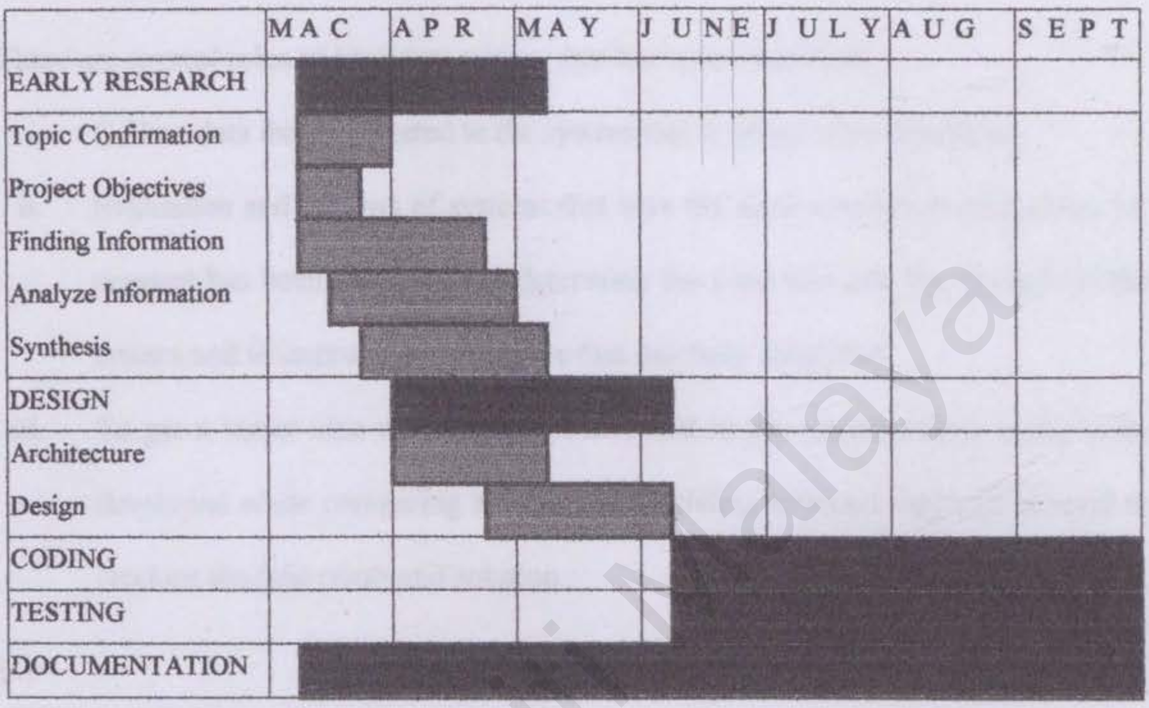


Figure 1.1: Gantt Chart for Project Development Plan

1.8 STATEMENT OF EXPECTED OUTCOME

The system is expected to help user in recording, storing and retrieving information in a more systematic way. Using this system also will reduce space and time needed to store and retrieved increasing patients records manually. Besides that, user will also be able to determine how much profit the health care center made for that day.

CHAPTER 2 LITERATURE REVIEW

SECTION I: REVIEW AND WRITING

2.1 ROLE OF LITERATURE REVIEW

There are several roles of literature review that has been identified:

i. Collect data that are related to the system that is going to be developed.

ii. Evaluation and reviews of systems that uses the same concept or relevant to the concept has been developed to determine the weakness and the strength of the system and to improve the weakness that has been identified.

iii. To get a better idea of the concept involved in the system that is going to be developed while comparing several other existing software that will be used to produce the best result and solution.

2.2 SYSTEM DEFINITION

The word system has several meanings (1); a way to get things done, (2); a group of several parts working together for the same purposes (3); group of organized opinions to perform something.

2.2.1 *Information Definition*

The word information is from several words that means inform, notify or declare.

2.2.2 *Health Definition*

The word health has several means physical condition, strength, well being and vigor

2.3 FINDINGS / REVIEW METHOD

Generally, system development process is not complete if there are no collection and review regarding the system that is going to be developed. Accuracy of information is vital in determining whether the system will accomplish in achieving its objectives. Information can be gathered from a lot of sources and every source provides different type of information depending on the techniques used. Some techniques had been identified to analyze and review the existing system. The techniques are data collection and writing method.

2.3.1 APPROACH TO LITERATURE REVIEW

1. *Interviews*

This method had been chosen to gain information on how the system that exists operates. Respondents are people who are already using a Health Information System or about to use a Health Information System. Informal interviews are also held from time to time to gather more information and opinions on the system that is going to be developed. This helps to identify and understand problems that may arise when the system is developed. Most of the respondents are people who are involved in health care fields such as doctors, nurses, and pharmacist.

2. *Observation*

Observation is being made by visiting health care centers around Klang Valley as a guest. Several weakness of the manual system used by some of the health care centers has been identified. Careful evaluation has been made to compare the systems that exists and how it operates at the health centers.

3. *Documentation / Books and Magazines*

Review and anlysis had been made on ducuments and books related to the system that is going to be developed. Data collection from books and magazines is being made to gain extra information from the reviews.

Most of the documents are provided by the health care centers and most of the books being used as references are located in the main library University of Malaya.

4. *Internet Surfing*

Resources from the internet helps a lot in the success of developing this Health Information System. Most of the information used as reviews are retrieved from the Internet sources. Various information retrieved from the internet is being used to compare the advantages and disadvantages of the existing system being used worldwide.

2.3.2 WRITING METHOD

1. *Document Analysis*

Analysis has been made to all data that has been collected and it has been summarized to a more simple form so that it is much more easier to understand and meets the objective of the system development phase.

2. *Comparative*

Summary of result has been made from comparative studies of several system that has already exists. The system that is going to be developed will use the existing system as a guide to develop a better system.

2.4 WEAKNESS OF MANUAL SYSTEM

Manual system has always been the best method in recording information until information technology era takes place. Although it has been used for centuries there are many weakness of the manual system.

- The problem of losing forms filled by patients often happens.
- It takes time to retrieve a patient's record.
- Patient's medical record will increase by time and more space is needed to store these records.
- Data redundancy might occur causing problems in management level.

2.5 ADVANTAGES OF A COMPUTERIZED SYSTEM

1. *Fast response*

Updating and searching process can be done in just seconds.

2. *High capacity of data storage*

A computer has the ability to store high capacity of information thus eliminate the problem of needing more space for string patient's record.

3. *Friendly user interface*

4. *Updating process made easy*

The use of a computer makes it much more easier for the administration staff to update records and eliminates the problem of data redundancy.

SECTION II: SYSTEM THAT EXISTS

Based on information findings, there is several other health information systems related to health care. The function of this system will be explained as a guide in developing the Health Information System.

2.6 PROCARE 2000

2.2: Interface for Procare 2000

Procare 2000 is a Hospital Information System that uses UNIX operating system. This system enables owner to make sure all transactions are recorded and well maintained. This includes recording information of patients, billing, payment collection and drug dispensing. This system is integrated with the Finance and Inventory Department that enables the system to automatically order drugs and medicine that are almost out of stock.

2.6.1 Advantages of Procare 2000

Front-end application increases efficiency in data exchange

- Multi level password security
- Automatically generate receipt when payment is made
- Automatically add up all charges for the patient including late charges and additional charges.
- The amount need to be paid is deducted from deposit for inpatients
- Supports appointments

2.6.2 Weakness of Procare 2000

User Interface

User interface for systems using UNIX operating system is dull not user friendly. Basically the interface is complex and it is hard for a beginner user to understand how to use the system. This will have a large impact on training periods and transactions processing by user.

2.7 PANTAI MEDICAL CARE

SCS Computer Systems Sdn Bhd develops this software. It only operates on Windows platforms.

Figure 2.3 (a): Interface for Pantai Medical Care

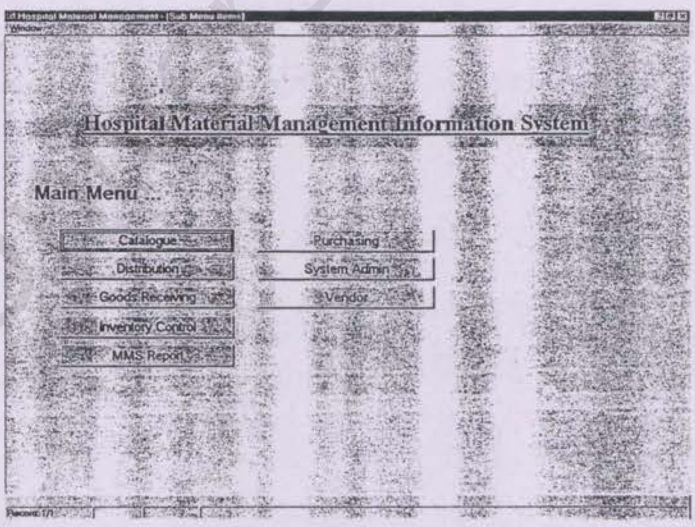


Figure 2.3 (b): Form Interface for Material Management of Pantai Medical

2.7.1 Advantages of Pantai Medical Care

- User friendly interface
- Automatic reminder for due appointments
- Supports networking
- Integrated with a barcode system
- Several options given to print bills in details and summary

2.7.2 Weakness of Pantai Medical Care

Forms created by Pantai Medical Care are too complex making it hard for users to understand the system. Each form requires a lot of detailed information to be filled by the user. Only experienced user will be able to master the system effectively. New user will have to learn how the system works carefully.

CHAPTER 3 SYSTEM DEVELOPMENT METHODOLOGY

3.1 INTRODUCTION

The word "*Methodology*" is a combination of two words that is *methodos* and *logos*. *Methodos* is a Yunani word that means way or route while *logos* means knowledge in careful observation. Method is an important aspect for observing an object carefully as a research material. Accuray in method guidance will result in an accurate research.

After a literature review done as part of a system development process, this section is being done to explain regarding the development methodology of the system. The main purpose of this analysis is to help people understand the system that is going to be developed. The analysis done covers most aspect of strategy and paradigm that will be used in developing the system.

Several stage in project management that need to be completed:

- Determine project start time so that the next development activity can be done effectively.
- Plan the project properly where it explains in details activities and works need to be completed. Execution of every plan must be monitored and updated according to the information received.
- Execute the project according to what has been planned.
- Observed the project flow and make necessary changes if any part of the activity has a problem that cannot be solved or does not meet the original requirements.

- Project closing is where the project management ends and it shows that the project has achieved its objectives and fulfills its original requirements.

Analysis regarding the development tools that will be used is also being done. This analysis consists of reliablility factors and system reusability that is being developed from the aspect of hardware and suitable software.

3.2 METHODOLOGY

Software engineering is important in organizing and executing the development of a system. Systematic analysis using *System Development Life Cycle* is a standard methodology to ensure the development process fulfills 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 System Development Life Cycle is divided into several software process models such as the Waterfall Model, V Model, Prototyping Model, Transformational Model and other models. This Health Information System will be developed using the Waterfall Model with Prototype used as a methodology guide. This model has several advantages over other models to ensure the development process is properly organized, high in quality and meets the standard. The advantages are:

- Important stages in the development process can easily be recognized.
- Easy to separate one stage from the other stage.
- Presents a very high-level view of what goes on during development.
- Easy to estimate period of time needed for an activity or stage.
- Prototyping process is used to examine some aspect of the proposed system and decide whether it is suitable or appropriate for the finished products.

- Validation ensures that the system has implemented all of the requirements, so that each system function can be traced back to a particular requirement in the specification.
- Verification ensures that all function works correctly.

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 – Waterfall Model with Prototype.

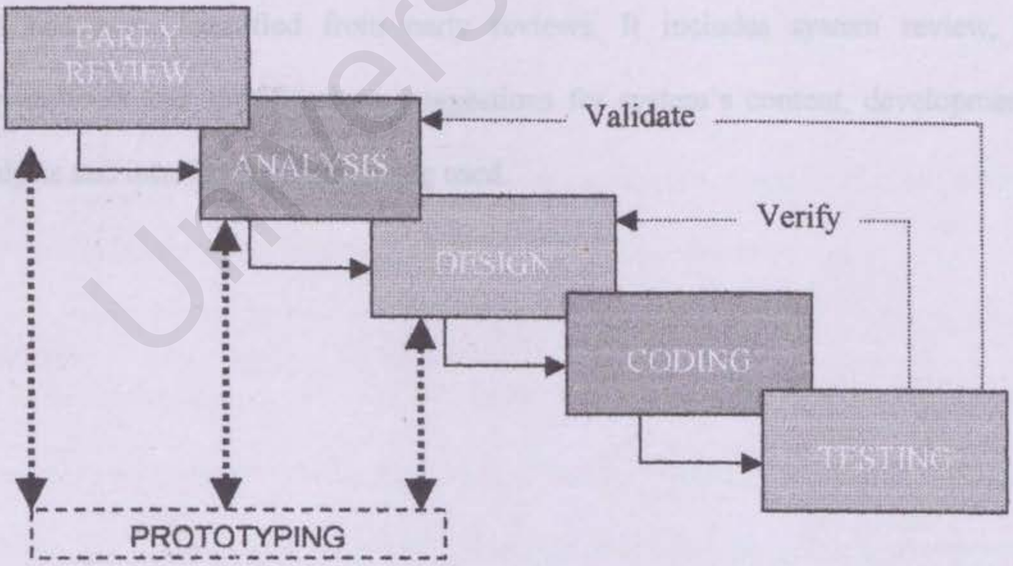


Figure 3.1: Project Development Phase

3.2.1 PHASE I: EARLY REVIEW

This phase is also known as *feasibility study*. It could also be described as Investigation and Identification Phase.

- Investigastion and Identification 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 for the objective of the Health Information System in replacing the manual system. Early investigation on the manual system has been done to identify the process involved in the manual system and the problems that arise. User's demand is also being considered in this phase.

3.2.2 PHASE II: SYSTEM ANALYSIS

The analysis phase is done to understand how the Health Information System 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.

3.2.2.1 System Analysis

- The system that is going to be developed will be a *stand-alone* system where system development is actually done by developing a compact disk that has these properties:
    - ❖ Offline retrieval and it is much faster compare to manual system using keywords.
    - ❖ Maximum data capacity for every compact disc is around 650-750MB.
    - ❖ Low cost
    - ❖ Data displayed at a very interesting interface with multimedia elements.

3.2.2.2 System Requirement Analysis

System Requirement Analysis can be divided into 2 sub-categories:

- Functional Requirements
- Non-Functional Requirements

Functional Requirements

Functional requirements are functions or system abilities. Functional requirements also explain the interaction between the system and its environments.

1. *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.

2. *Update Record Function*

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

3. *Patient Recording Information Function*

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

4. *Patient History Check Function*

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

5. *Billing Function*

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

6. *Search Function*

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

Non-Functional Requirements

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 more easier for user to understand and use the system.

- Interesting Interface

  Interesting interface is a vital aspect needed to encourage user to use the system.

3.2.2.3 System Requirements Analysis

Decision on using the appropriate software for developing the system is important for system implementation. This is to ensure that the software's going to be used to develop this project is suitable with the concept of the project. This includes the main software needed to develop the system and the software's required to support the system and extend the system's capability. The software's that is going to be used in developing this Health Information System are:

- Microsoft Access 97
- Visual Basic 6.0

Microsoft Access 97

After careful review of several systems that can be used to develop the system, Microsoft Access 97 seems suitable to build the database for the Health Information System. Compared to Microsoft Access 2000, Microsoft Access 97 can merge directly with Visual Basic 6.0. Microsoft Access 2000 has several features that cannot be detected by Visual Basic 6.0 and it needs AccessToVB 4.6 to merge with Visual Basic 6.0. Advantages of using Microsoft Access 97 are:

- Easy to learn and execute
- Tables or query can be inserted as a part of a sheet into another sheet.

Visual Basic 6.0

Visual Basic 6.0 is an object oriented programming. This software contains 3 major components. They are *object*, *properties*, and *method.*

- *Object* is a thing or name
- *Properties* explains object attributes such as name, size and color
- *Methods* are functions done by an object such as *move*, *print*, *resize*, *calculate* and *clear*.

Visual Basic also combines a set of software technology called Active X. This technology enables the creation, integration and reuse of software components called control. Among the criteria contains in Visual Basic 6.0 are:

- Interesting user interface
- Properties on forms and sub-forms can be changed easily by using programs.
- The ability to be integrated or automated with other applicaion such as ODBC, Excel, FoxPro, Paradox and others.
- Able to design graph and charts using wizards and also have the ability to process graphs and other data type.
- Visual Basic also could be expanded by adding *Custom Control* and calling procedures in the *Dynamic Link Library* (DLL).
- This software also has the ability to merge with Microsoft Access 2000.

3.2.2.4 Hardware Requirements Analysis

Usually hardware refers to machinery or physical tools that perform basic functions in the process of operation cycle. Minimum specifications required to execute the Health Information System are as follows:

- Intel Pentium MMX Processor
- 32MB of RAM
- 500 MB of Hard Disk space
- 256-color monitor with SVGA display (resolution 800*600)
- Windows 98

3.2.2.5 Interface Design Analysis

- User interface must apply the concept of simplicity to ease interactivity.
- Instructions and display must be clear and easy to understand. This is being done by providing menu labels for icons used.
- Standard and consistency in display
- System must response accordingly to user request to encourage user interaction.

3.2.2.6 System Ability Analysis

- Capacity level adequate to store information about patients and other related field
- System reliability without faults
- Ability to display dynamic contents.

- Ability to analyze user input integrated between forms
- Ability to store every patient's record

3.2.3 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 exists. The description of a solution is also called design.

This 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. Menu design, content presentation, modules and database are parts of the properties that will be explained. This is done to simplify the interface developing process of the system that is going to be developed.

This phase is being explained in details in Chapter 4 (System Design).

3.2.4 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, 3 main aspects that are being emphasis on is control structures, algorithm and data structure. This phase will be executed in the next semester.

3.2.5 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 devloped 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 user's requests. 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 reevaluate.

This phase will be executed after the implementation phase or programming phase.

CHAPTER 4 SYSTEM DESIGN

4.1 INTRODUCTION

Design is the creative process of transforming the problem into a solution. The description of a solution is also called design. It is viewed as a process that represents data structure, program structure, interface properties and procedural information. Design is a method that translates user requirements into a product or finished system. All of the information gathered during the system analysis phase will be converted into smaller modules until a system is being developed.

4.2 STRUCTURE DESIGN

Structure chart will be used to show the workflow of the Health Information System. According to the chart, the Health Information System has been divided into several components depending on its functionality. Components on the higher level will represent functions that are available in the main interface display followed by the rest of the components.

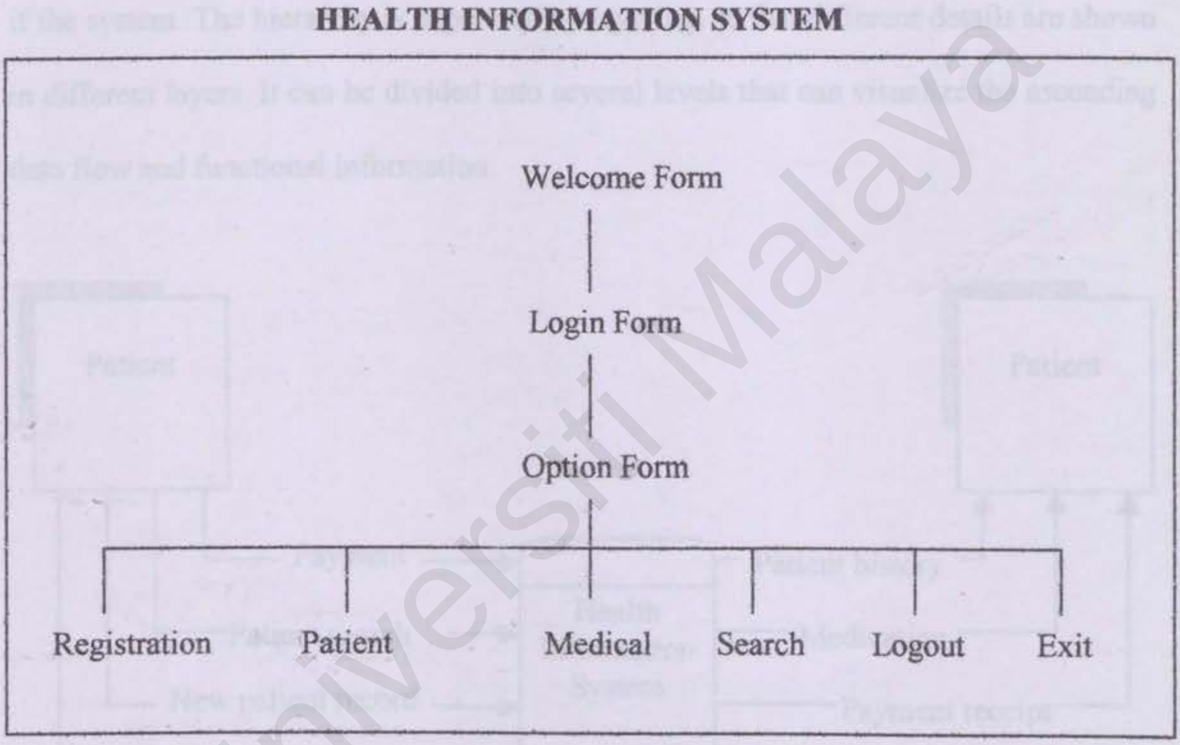


Figure 4.1: Structure chart for Health Information System

4.3 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 if the system. The hierarchy is expressed by layering, so that different details are shown in different layers. It can be divided into several levels that can visualize the ascending data flow and functional information.

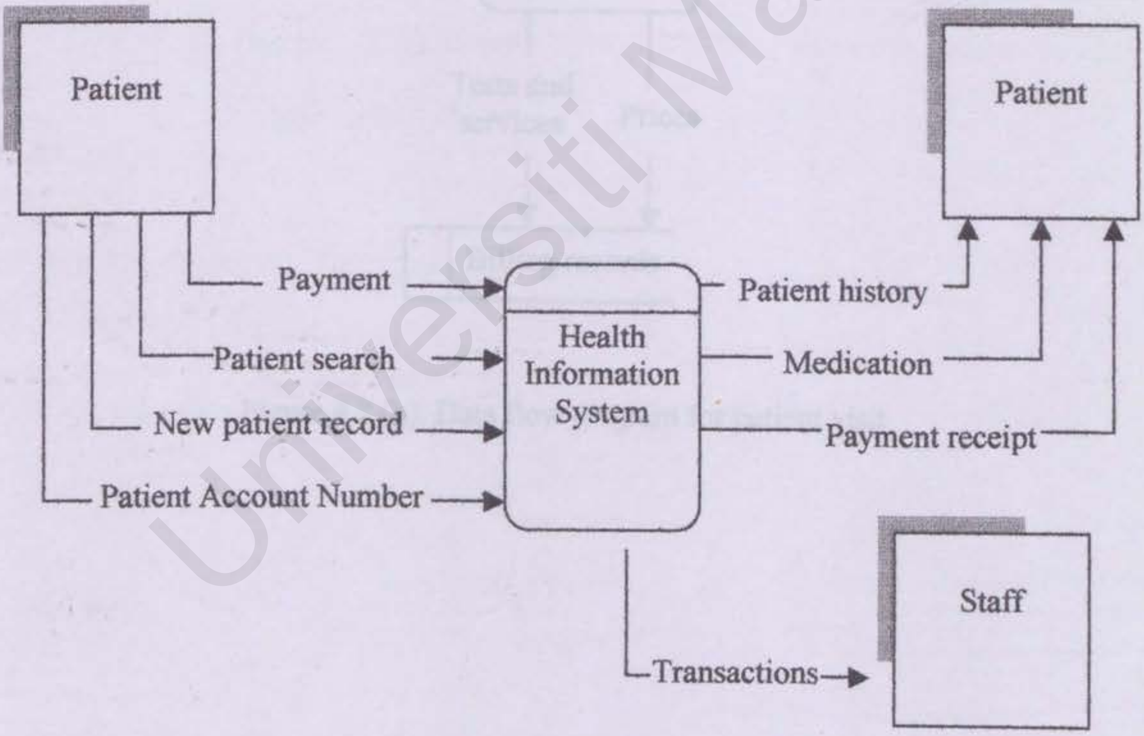


Figure 4.2 (a): Context Diagram

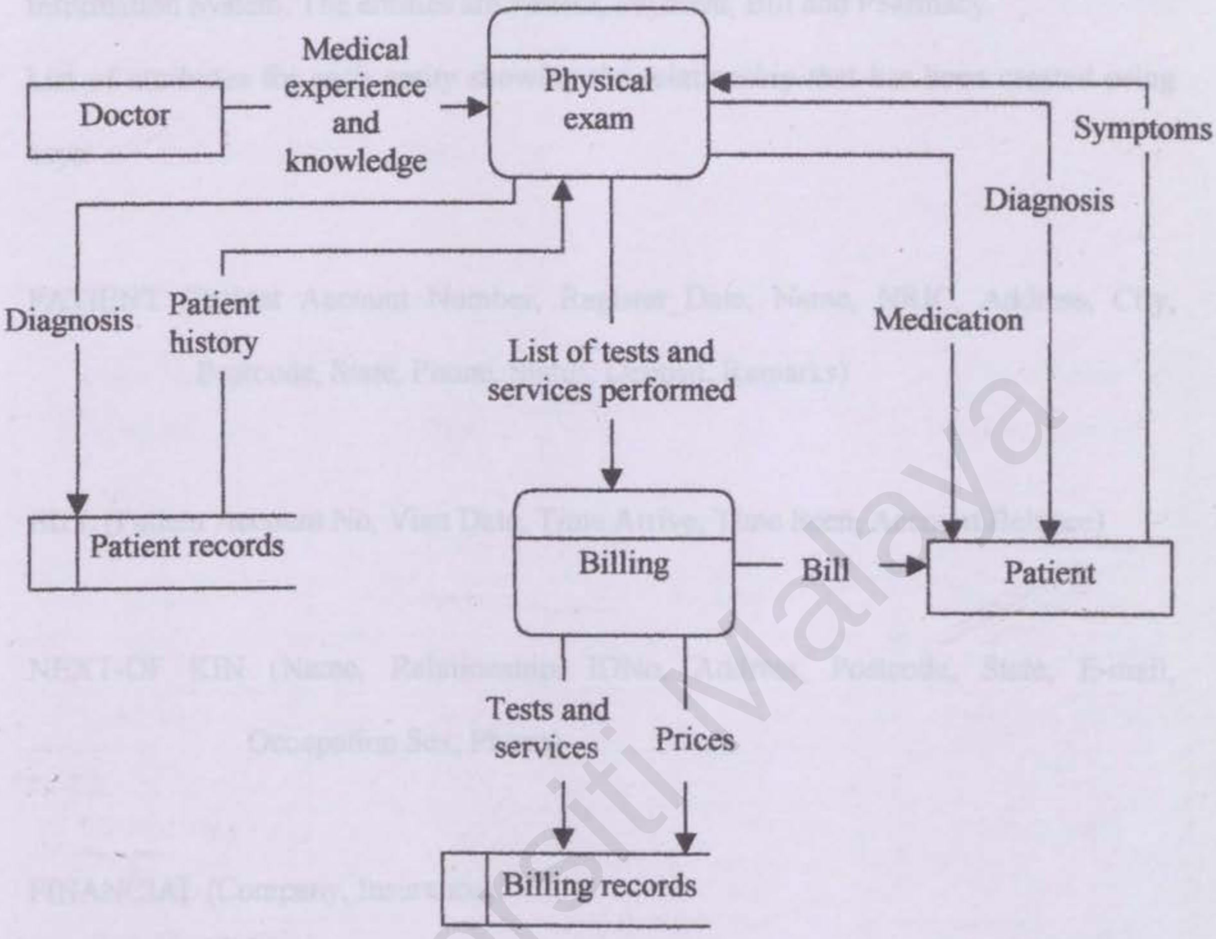


Figure 4.2 (b): Data flow diagram for patient visit

4.4 DATABASE DESIGN

There are a few entities that had been identified in designing the database for the Health Information System. The entities are Patient, Payment, Bill and Pharmacy.

List of attributes for each entity showing the relationship that has been created using keys:

PATIENT (Patient Account Number, Register_Date, Name, NRIC, Address, City, Postcode, State, Phone, Status, Deposit, Remarks)

BILL (Patient Account No, Visit Date, Time Arrive, Time Seen, Account Balance)

NEXT-OF KIN (Name, Relationship, IDNo, Address, Postcode, State, E-mail, Occupation Sex, Phone)

FINANCIAL (Company, Insurance)

PHARMACY (Medication, Allergy, Quantity, Date, Doctor_ID)

4.4.2 Database Table

The database for Health Information System will be developed using Microsoft Access 2000. There are some tables used to streamline the database. All data are stored in tables that will view different entities. The following tables will give an explanation about attribute contained in every entity.

Table for PATIENT

Field Name	**Field Type**	**Size**	**Description**
Patient A/C No.	Text	10	Patient Account Number
Register Date	Date	8	Registration Date
Name	Text	50	Patient Name
NRIC	Text	12	Patient I/C Number
Address	Text	50	Patient Address
City	Text	50	City
Postcode	Text	6	Postcode
State	Text	50	State
Telephone	Text	11	Patient Telephone Number
Status	Registered/ Not Registered	2	Active/Not
Race	Text	10	Patient's Race
Remarks	Text	50	Remarks
Date of Birth	Text	10	Patient's Date of Birth
Nationality	Text	15	Patient's Nationality
Religion	Text	10	Patient's Religion

4.4 INTERFACE DESIGN

User interface design is an important aspect in developng softwares. This is because an interesting user interface design will inspire and motivate a user to test and use the system.

Objectives of building a user interface:

- Effectiveness of using user interface

  This can be accomplish by designing an interface that meets the user requirements and the simplicity if the interface to avoid confusion.

- Interface reliability

  Interface accuracy in performing data capturing without errors.

- User judgement

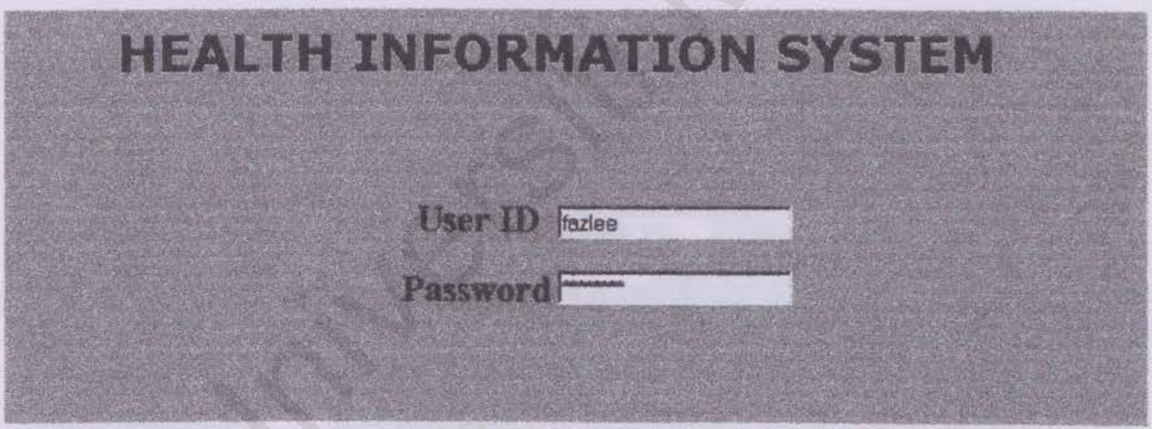
  User interface developed must meets user requirements and respose given by the system must be accurate and meets the user's request.

- Productivity

  This can be measured by developing a good user interface that reduce building cost and response time.

Interface design will visualize the display that will be used as an interaction medium between the user and the system. There are several factors need to be considered when designing the interface:

1. Soft background colour , icons, logos, pictures and appropiate fonts.
2. Consistency betweeen pages to avoid confusion for users.
3. Control measures such as combo box, selection box and check box are used to accelerate data recording process.
4. Suitable type of fonts are used so that the interface would not look too complex and it would be more organized and easy to understand.

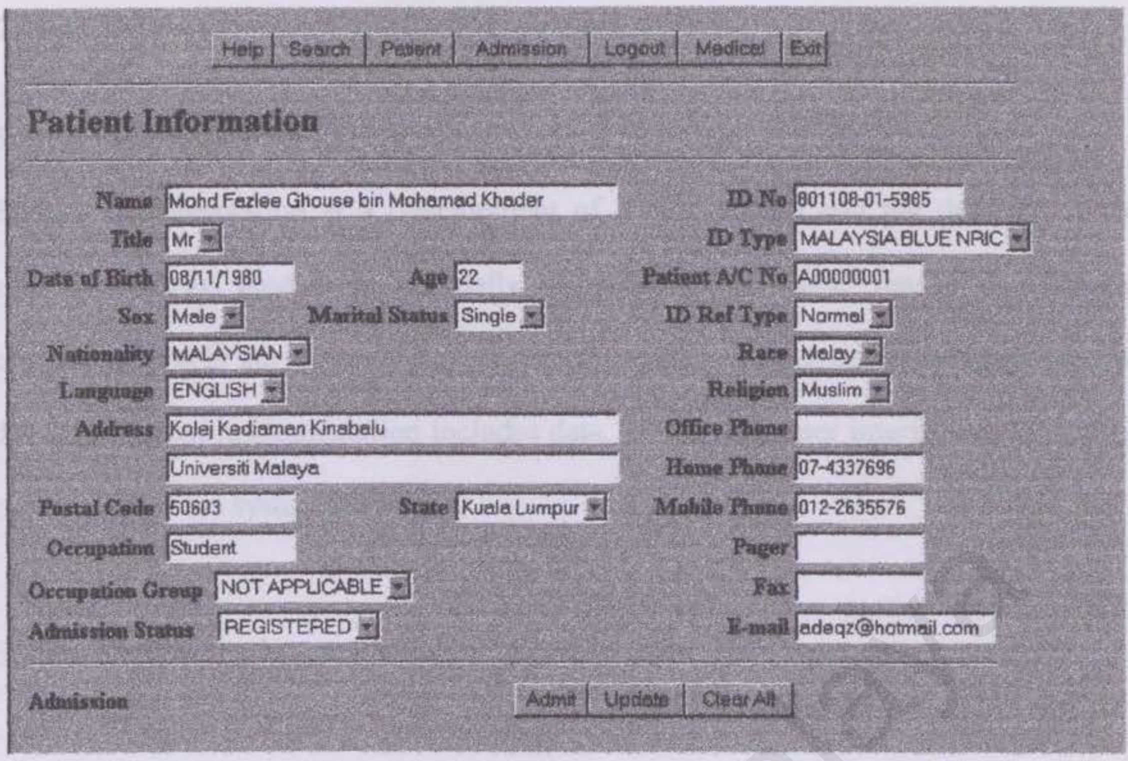


HEALTH INFORMATION SYSTEM

User ID fazlee

Password ********

Figure 4.4 (a) : Example of user login interface

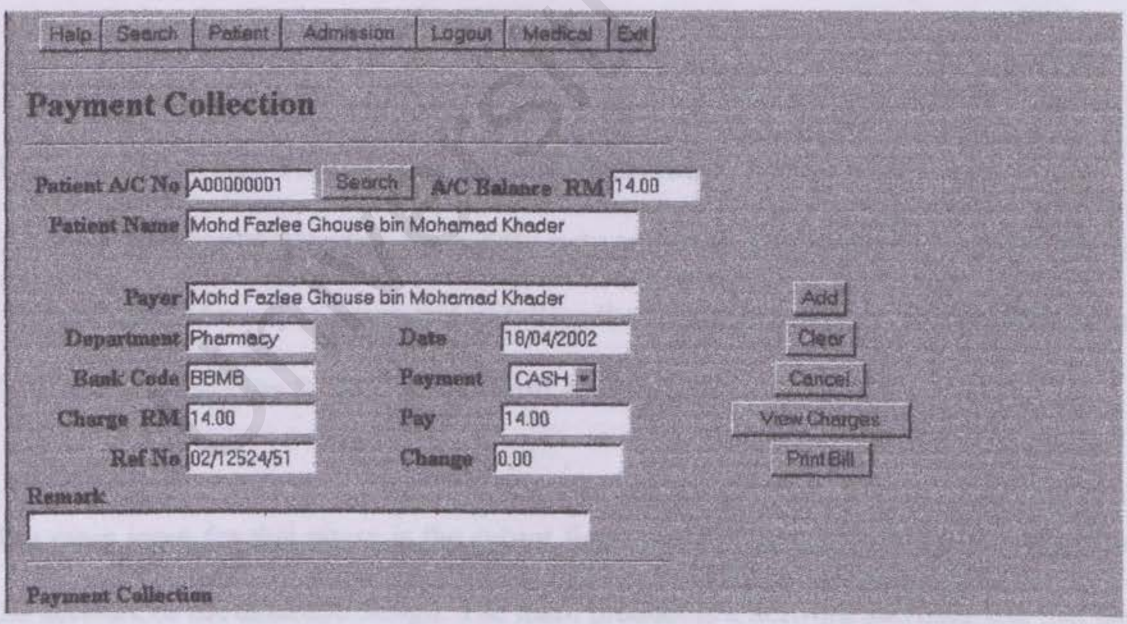


Patient Information

Name: Mohd Fazlee Ghouse bin Mohamad Khader
ID No: 801108-01-5985
Title: Mr
ID Type: MALAYSIA BLUE NRIC
Date of Birth: 08/11/1980
Age: 22
Patient A/C No: A00000001
Sex: Male
Marital Status: Single
ID Ref Type: Normal
Nationality: MALAYSIAN
Race: Malay
Language: ENGLISH
Religion: Muslim
Address: Kolej Kediaman Kinabalu
Universiti Malaya
Office Phone:
Home Phone: 07-4337696
Postal Code: 50603
State: Kuala Lumpur
Mobile Phone: 012-2635576
Occupation: Student
Pager:
Occupation Group: NOT APPLICABLE
Fax:
Admission Status: REGISTERED
E-mail: adeqz@hotmail.com

Admission

Figure 4.7 (a): Example of patient information user interface



Payment Collection

Patient A/C No: A00000001
A/C Balance RM: 14.00
Patient Name: Mohd Fazlee Ghouse bin Mohamad Khader
Payer: Mohd Fazlee Ghouse bin Mohamad Khader
Department: Pharmacy
Date: 18/04/2002
Bank Code: BBMB
Payment: CASH
Charge RM: 14.00
Pay: 14.00
Ref No: 02/12524/51
Change: 0.00
Remark:

Payment Collection

Figure 4.7 (b): Example of payment collection user interface

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

System implementation is a development of a new system and the delivery of the system towards production in daily operation. The main reason for system implementation is the technical design from the system design. Information system that develops system implementation includes data, process and user interface based on the perspective of the system developer.

System implementation can be divided into four main phase that is:-

i) Develop and Test Network and Database

ii) Testing and Developing Program

iii) Installation and New System Testing

iv) Delivering the New System for Operation

5.1.1 Building and Testing Network and Database

If a new application requests for a new network or a new database, it will first be implemented before developing and installing the computer program. The database used for Health Information System is Microsoft Access 2000.

The main input for this phase is the subset from the real technical design that visualize the network or the database design. The main product is a network and database that has not yet been populated. The structure of the database has been implemented but data has not yet been inserted in the structure of the database. Program developer will finally write a program to populate and enchance the database.

5.1.2 Building and Testing the Program

This phase is also known as the development phase. Program developing and testing usually takes a very long time and it is a very tiring phase in developing a system. Program developer has to work from the specification that has been developed and filtered through the prephase and preactivity in the Waterfall Model. If the specification of the system is not clear, not completed, not accurate or ruined, the development phase will be more complicated and takes a longer time.

The main input to this phase is the subset from the technical design statement that contains the specification of the program. If the new or modified network and database will be used, the database that has been implemented but has not yet been populated is the input from the implementation phase that has been done before. The product of this phase is a computer program that has not yet been installed but has been completely tested and debugged to be used for production.

5.1.3 Installing and Testing the New System

The next phase in system implementation is to install and test the new system. The main input to this phase is a subset of the technical design statement that gives the specification on how the program has been developed and tested. Files and database will be piled up in the integrated system. The product of this phase is a system that has been installed and ready to be delivered for production.

5.1.4 Delivering the New System for Operation

The final phase in implementation is to deliver the new system for operation. Normally, a new system represents a change in how the business runs. Smooth changes needs to be done from the old system to the new system along with an extra help to users to overcome normal start-up problem. Training new users and providing a various system manuals helps user in using the new system.

5.2 DEVELOPING HEALTH INFORMATION SYSTEM

The phase that needs a very long period to be completed is the development phase of the HIS itself. It involves interpretation and the implementation of all gathered requirements including system technical design into program codes. The combination of technical and physical design into program codes had been done using Microsoft Visual Basic 6.0.

5.2.1 Content Preparation

The most important aspect in developing the Health Information System is the content of the system that is what is going to be displayed to the user and the operation that can be performed by the user. The amount of the contents in the system depends on the system requirements. It has to generally include all the aspect and function needed by users.

There is a main form that uses *Multiple Document Interface (MDI)* and there are sub forms called *Multiple Document Interface Child* that represents every single module.

The main aspect in the main form is the use of toolbar application and the menu editor as a medium to open all other sub modules. Icons and shortcut key is being use to ease the user of the system.

In order to make the system more interesting, colors, images and appropriate graphical buttons are used. All the buttons can be obtained from the Microsoft Visual Basic 6.0 software.

5.2.2 Content Interpretation, Presentation and Programming

All the system content integration, presentation and programming using Microsoft Visual Basic 6.0 are divided into 2 major components, that is:-

a) Visual Components and

b) Code Components

Visual components or interface refers to screen that will be displayed and used by users. The structure and layout design can be easily determined by the system developer. These components are created using forms and controls. Forms and controls enables the manipulatin of users input, information display, output options and users result. For this, system presentation and programming can be integrated to generate the final product.

Visual components or interface refers to screens that can be viewed and use by users. Coding components refers to program line that has to be typed and invisible to users.

5.2.3 Coding Phase

Coding phase is the phase where all the result from the analysis phase and the design phase is being transform into a real application system. This phase also requires quite a long period of time to be completed because HIS is being develop using high level programming language. The software that has been used to develop HIS is Microsoft Visual Basic 6.0. Appropriate consideration of all the output from phases that has completed must be done by the developer before it is being transformed into an application system.

Activities in the programming level have developed program modules that have been compiled 'cleanly'. All test are done on every program module that has been coded and compiled so that there are no errors in every single program module.

Besides considering the output from the phase before, other limitations factors for developing the system must also be considered. Several limitations factors that has to be considered are:

i. Limited time and energy for this phase

ii. Development cost factor

iii. Application system size factor that has to be considered is the ability of a personal computer to run the software. (Big application size might not run on personal computers).

5.2.3.1 Coding Method

Subsystems in this system are formed based on sequential functions, data requirements and logic equations. Usually every subsystem contains several programs.

In the concept of structured program design, the 'coupled' concept will generate modulated programs while the 'bind' concept will generate structured programs.

a) **Module Programming**

Modulated programming is a programming method that divides a single complex problem into smaller parts so that is much more easier to be programmed. HIS is being programmed this way to overcome the complexity of it and to make it much more easier to be understood.

b) **Structured Programming**

Structured programming is a structured and well-organized programming method. Several coding steps that are used to ensure that the structured programming methods is followed are:

i. Unconditioned forking commands must be eliminate or at least being minimized in every program module.

ii. Commands in every program routine must be based on one logic sequence so that it only contains one input source and one output source from the routine.

iii. Every routine must have a complete code along with comments that are easy to understand. .

5.2.3.2 Coding Approach

The coding approach chosen is quite a complex and complicated process because it needs full commitment and continuous effort. Every module or form was developed using the Waterfall Model With Prototype approach while the project flow control use the bottom-up approach. Documentation contents such as the displayed report is an implementation during the design phase and it is not done during the system runtime.

In this phase the ability of the developer is being tested, where every event that is expected on the system needs manual code development that depends on the knowledge of the developer.

5.3 SUMMARY

From what has been explained in this chapter, it can be summarized that the coding phase was a very complicated phase in the process of developing a system. The coding process generate friendly user interfaces, so that it is much more easier for users to use the system.

CHAPTER 6 SYSTEM TESTING

6.1 INTRODUCTION

In ensuring the quality of software or a system, system testing need to be performed and it is one of the critical elements. This process involves careful examination of all the design specifications and coding process that has been performed along the system development process.

Testing is also performed to ensure that all the modules developed are free from any errors that can cause unreliability to the system from performing as required and to produce result as desired. Usually testing is performed using sample data and logics that are used in coding.

A good test is a test that is able to identify all the errors that are not detected during the analysis phase, design phase and coding phase. The main objectives in system testing are:

i. **Identify errors**

Detailed checking is being performed to every function and behavior of the system to identify errors in the system.

ii. **Removing errors**

Errors are removed from the system by compiling the codes after detecting the cause of errors or by debugging the system.

iii. **Regression test**

To identifies new faults that may have been introduced as current ones are being corrected.

6.2 TYPES OF TESTING

The main intention of the testing process is to evaluate how much fault can be reduced in the program or in the module itself. The correction process on demonstration is against the meaning of testing. Testing is performed on the program to demonstrate existing fault. Since the main objective of testing is discovery of faults, all the faults that might lead to failures during actual system usage will be eliminate to ensure successful testing result. Fault identification is a process to determine fault or the cause of it, while fault correction is a process to make changes to eliminate fault.

6.2.1 Unit Testing

Unit testing includes test on every single program module components separately. Every file in the same module will interact internally or interact will other files in different module.

6.2.2 Module and Integration Test

After knowing every function and unit works perfectly fine and fulfill their objective, every module compenents will be merged to create a system. This merging process will give the real description when system failure occurs. This is what has been performed during the development of this system. There are four main testing method at this level. They are:

- Bottom-up Integration
- Top-down Integration
- Big-Bang Integration
- Sandwich Integration

6.3 TYPES OF FAULT

Every system contains simple fault and complicated fault. When simple fault does not exist system will be tested for segregating many faults by creating options where the system will function as desired. That is why it is important to know what kind of errors need to be identified. Fault or mistake can be divided into three types:

- Algorithmic faults
- Syntax faults
- Documentation faults

6.3.1 Algorithmic faults

Algorithmic faults occur when the algorithmic components or logic components is not producing the desired output for the input given. This happens because of a mistake that occurs during processing steps. This type of mistake is quite easy to be identified just by looking at the program (called desk checking) or by sending an input data to every different data class. The types of mistake are as follows:

- Testing for the wrong condition

- Forget to declare variables or different loop
- Forget to test for certain condition such as an integer divided by zero
- The use of out of range variables such as using local variables at other functions.

6.3.2 Syntax Faults

Syntax faults can be checked during an algorithmic fault occurance. This will cause inaccuracy in writing the code of the programming language. When a program is running, incase there are any syntax errors, it will be identified and the location and type of errors will be stated.

6.3.3 Documentation Faults

Documentation process of a system is not an easy task. It usually contains spelling mistake, inaccurate term and incorrect guide that might lead to instruction's misunderstanding by readers. Documentation is very crucial to ensure that the system that is developed can be modify accordingly in the future for organization needs.

6.4 SYSTEM TESTING

In the testing process of Health information System, there are six main tests that have been conducted. The tests are as follows:

i. Unit Test

ii. Integration Test

iii. Function Test

iv. Performance Test

v. Acceptance Test

vi. Installation Test

Testing sequence is as shown in Figure 6.1. Test performed on HIS is a bottom-up testing technique that is starting the test from the smallest unit until the system is entirely tested including the installation of the system.

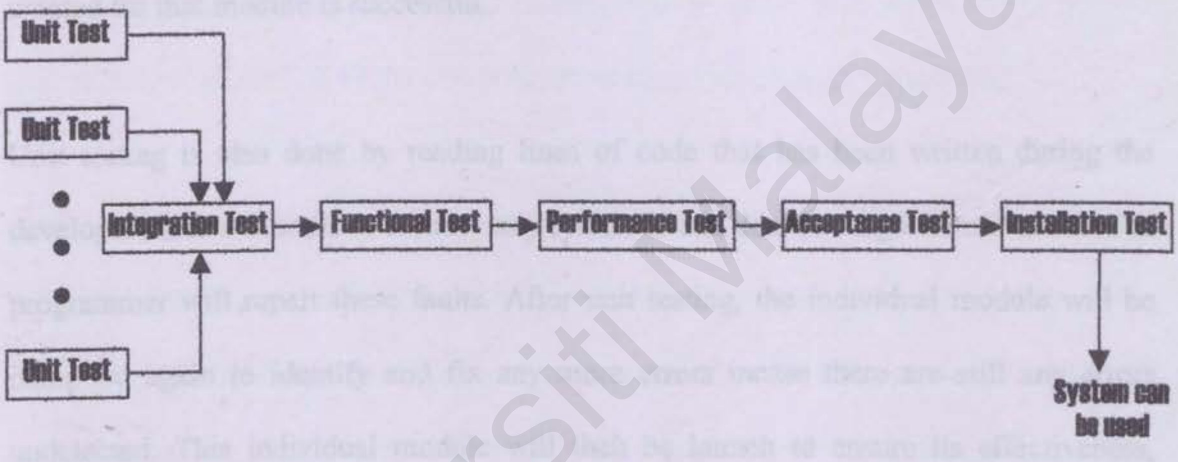


Figure 6.1: Testing Steps

6.4.1 Unit Testing

Unit testing involves testing on smaller unit called module that is focused on logic, accuracy, condition boundaries and error management. In details, it ensures the flow of accurate information that is all the units are able to receive input entered and produce the output as expected.

Besides that, boundary conditions must be executed correctly according to the status that has been set so that one pathway can be transferred to another pathway incase needed. Other than that, it tests every error management pathway to ensure whether the process will continue or it will be transferred to another pathway when an error occurs. For example, after the development of the patient module interface, data will be entered through the interface. The database of the module will then be checked whether all the data entered through the interface are successfully recorded in the database. If all the data entered through the interface are available in the database that means that the unit created for that module is successful.

Unit testing is also done by reading lines of code that has been written during the development of a module to identify any syntax errors, data and algorithmic errors. The programmer will repair these faults. After unit testing, the individual module will be compiled again to identify and fix any more errors incase there are still any errors undetected. This individual module will then be launch to ensure its effectiveness, accuracy and to see whether it functions as desired.

6.4.2 Integration Testing

It is a process that validates the system components so that it can work along together as described in the program design specification and system. It focused on testing one or more module that has been integrated. An example of the integration test performed is an integration test was done on several integrated modules such as the payment module and the registration module that has been integrated. Relevant data are entered in the module according to the specification specified and then the database of these modules will be examined to ensure that the flow of data from one module to the other module is accurate. Integration test is also done to determine whether the integration effect has a negative effect on the performance of each module.

In performing the integration test for this project, bottom-up integration test strategy has been used. In this strategy, every component that exists at the lowest level in the system hierarchy will be tested individually. The next components that will be tested are components that will call the tested components in order to function. This type of approach is used until all of the components involved in the system are tested.

6.4.3 Function Testing

After the information sent between the components is accurate or according to the design specifications, system is tested to ensure it fulfills the functions required for the system. Function testing evaluates the system to determine whether the function described by the requirements specifications can really be presented by the system that has been integrated.

Test performed also involves recovery testing that intends to fail the system and make sure the failure can be recovered either automatically by the system or based on user input.

Besides that, security test is also performed to know whether the system can be trespassed by an unauthorized user. If the system can be trespassed, other security methods must be considered. Security measurements have been taken for this system by creating a login interface to prevent unauthorized access to the system. Users will have to enter a valid username and password at the login interface before they are allowed to enter the system. If the password entered by the user is incorrect, the user will not be given permission to enter the system. A valid and authorized username and password is required to enter the system.

6.4.4 Performance Testing

Testing is performed during system run-time to evaluate the system performance entirely. This includes the aspect of response time, system accuracy and amount of memory used. When the test is completed successfully in the real user environment, it is considered as a valid system.

6.4.5 Acceptance Testing

When function and performance testing are complete, this means that the system already meets all the requirements specified during the initial stages of software development. The next step involves negotiation with the user to ensure that the functions of the system meets all user expectations.

6.4.6 Installation Testing

If acceptance testing has been performed on-site, installation testing may not be needed. However if acceptance-testing conditions were not the same as actual site conditions, additional testing may be needed. The main process in installation testing is that the developer will try to install the developed system in a real environment to ensure that the system is able to function properly as required and tested. If it fails to function as desired by user, the developer will need to modify the system or the system environment so that the system is able to function as desired at optimum level. When the customer is satisfied with the results, testing is complete and the system is formally delivered.

6.5 SYSTEM TESTING TECHNIQUE

Techniques used for testing depends on the testing level that has been set. At unit testing level, white box technique has been used to determine errors as shown in Figure 6.2.

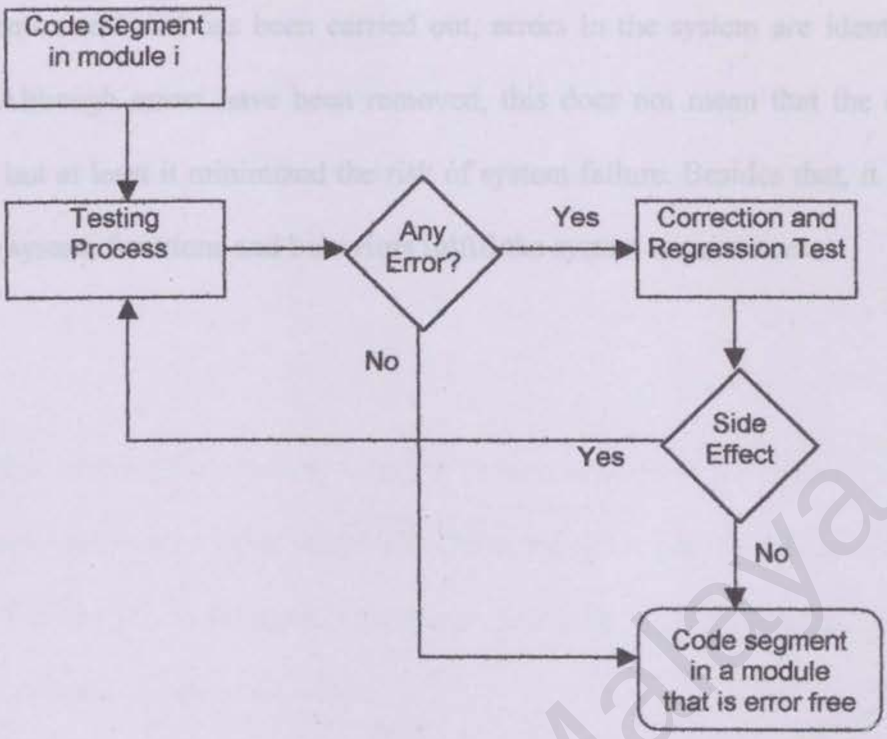

Code Segment in module i
Testing Process
Any Error?
Yes
Correction and Regression Test
No
Side Effect
Yes
No
Code segment in a module that is error free

Figure 6.2: Unit Testing Scheme

6.6 CREATING EXECUTABLE AND SET-UP FILE

After a program has been completely tested, developer will transform the program that has been developed into an executable file or .exe file. Microsoft Visual Basic software can easily helps to create executable files that is easy to be implemented on most personal computers.

Set-up file needs to be generated because it enables the system to run on different computers without the support of Microsoft Visual Basic software.

6.7 SUMMARY

With the entire test that has been carried out, errors in the system are identified and removed. Although errors have been removed, this does not mean that the system is error free, but at least it minimized the risk of system failure. Besides that, it is certain that all the system functions and behaviors fulfill the system requirements.

CHAPTER 7 SYSTEM MAINTENANCE AND EVALUATION

7.1 OVERALL SYSTEM REVIEW

After further analysis, it was found that there are several other functions needed to complete the system. The new functions added to the HIS are:

Queue

This function enables users to know which patient is waiting for treatment and how many patients are waiting to be examined. The name of the patient will be displayed in the queue list and it will be sorted based on a first come first serve basis. This queue function is added in the patient module.

Inventory

This function is integrated with the pharmacy function. It records the quantity of medicine that has been dispensed and the current stock amount in storage. It also enables users to add or remove stock amount in storage. When a prescription is made, the amount of drugs prescribed will be automatically deducted from the storage amount.

Reminder

This reminder function warns users incase there are any medicine in the storage that is low in quantity. It will display the medicine code, medicine name and the remaining quantity.

7.2 SYSTEM MAINTENANCE

System development phase is complete when the system is operational, that is, when users in an actual production environment are using the system. Changes performed to the system after that is considered as maintenance. In this section, the discussion is focused on the system maintenance requirements and how the system can be maintained when the functional requirements tends to change. This is to give guidance and understanding to users that will maintain the system so that it will not affect the system operations entirely during maintenance. Besides that, system recovery method is also included for this system

7.2.1 Maintenance Requirements

Generally, a system has to go through maintenance in a routine basis to make sure the system is operating at optimum level. Maintenance has to be made to this HIS system because of several reasons such as:

i. **Additional Patient and Payment Record**

Since HIS is based on patient data and patient record, additional patient information and patient records will increase the amount of data in the database of the system. The size of storage space available needs to be increased to store more incoming data. It is clear that the increment in patient record will result in the increment of storage space for the database.

ii. **Outdated Data Contents**

Inactive patient data need to be removed and the outdated record regarding the patient need to be erased from database table by the database administrator. Reports and documents for outdated records need to be removed from the database to make sure the database is free from unused data and require less storage space in the system.

7.2.2 System Maintenance Methodology

This system can be maintained through various methods such as:

i. Increment of Storage Size

The increment of the storage size can easily be done because HIS can operates in any personal computer running Windows 95/98/Me as an operating system.

7.2.3 Disaster Recovery Plan

Disaster recovery plan is made to provide support to system's operation incase a disaster occurs. The main contents of the system including the database has to be duplicated in a different storage device such as backup storage media, backup tape, diskette and other media storage device. If a disaster occurs, the system can still be retrieved and it does not have to be redevelop. The contents of the HIS database along with other documentations must be duplicated in the backup device from time to time so that the backup data stored stay updated.

7.3 SYSTEM EVALUATION

7.3.1 Introduction

The best way to develop a system is to involved system evaluation phase in the system development life cycle. This is the phase where a developer can analyze how successful the system that has been developed has reached its objective. Usually the developer will receive responses from users to evaluate a system.

7.3.2 Objective Achievement

The overall performance of the Health Information System reaches the objectives stated for the system. The system is fully functional and fulfills all the desired criteria set by the developer and the user.

7.3.3 Problems and Solutions

In developing this system, there are several problems confronted by the developer. Some of the problems can be solved easily but there are others that were unable to be solved by the developer alone. The developer obtains some help from many parties to solve all the problems that occur. There are several categories of problems confronted by the developer stated below:

7.3.3.1 Problems in System Design

□ Insufficient reference material

At the beginning of the project, the developer had insufficient reference material to develop the system. Reference material at the market is sold at a very high price and the reference material in the library is limited and most of it is outdated.

The developer manage to overcome the problem by taking several alternatives such as borrowing reference books from friends besides having to buy some books that are appropriate in developing the system. The Internet also provides relevant information regarding the system that is developed. The developer also joins discussion forums in the Internet regarding Microsoft Visual Basic to obtain extra knowledge regarding the programming language.

□ Lack of ability

Lack of ability in the Visual Basic programming language and Microsoft Access had cause the developer to have problems at the beginning of the project.

Finally the problem is solved by the developer. It takes quite a long time for the developer to master all the skills needed because the developer has to start from the very basic. The developer also gain help from friends and virtual friends from the Internet.

- Problem in linking software

  In the beginning process of the system development, developer had problems in linking the Microsoft Visual Basic interface with Microsoft Access database.

  Referring books and Internet resources regarding linking Visual Basic interface with Microsoft Access database solve this problem.

- Incomplete database design

  The database design proposed in Chapter 4 is incomplete. More tables are required to store all the data appropriately.

  Developer had to redesign a new database to fulfill the system requirements. More tables are added to ensure that all data can be stored properly according to the appropriate categories.

- Using Microsoft Access 2000 instead of Microsoft Access 97

  In the proposal it is stated that the software that is going to be used for the database is Microsoft Access 97. The developer was unable to get a full version of Microsoft Access 97 for it to work smoothly with Microsoft Visual Basic 6.0.

  The software was replace with a later version of it, which is Microsoft Access 2000. Since there are some incompatibilities, the features that is going to be used in Microsoft Access 97 is not used and had been replaced by using Active Data Object Data Control that is compatible with Microsoft Access 2000.

7.3.4 User's Feedback

Actually the developed Health Information System has not yet reached the end-user evaluation level. This is caused by insufficient time to find an appropriate health care center that offers health care services to perform the acceptance test.

To ensure that the Health Information System developed is effective, several tests have been done on users with the background of computer knowledge. The objective of this test is different. It focused on the aspect of data representation, form arrangements, extra options, toolbars and the use of icons. Among the information given by the system testers are:

- Comments on the choice for user interface
- Comments on the form arrangements
- System disadvantages
- System advantages
- Comments and suggestion on the system

Basically, the tester of the system must have experienced working in a manual health care environment. They will be able to understand the system better that an ordinary user and it is much more easier for them to give comments and suggestions regarding the Health Information System.

The tester is satisfied and comfortable in using the system interface. Form arrangements is simple and easy to understand. The toolbar enables user to move from one form to another form without having to go back to the main menu. As a whole, the tester feels that the flow of this system is satisfying.

7.3.5 System Strength

Compared to the system that already exists, this Health Information System has it own advantages:

7.3.5.1 More Effective

The manual system that has been used in almost all clinics in Malaysia is quite ineffective because of several disadvantages and limitations of the system (disadvantages of a manual system has been discussed in chapter 2). HIS has already overcome most of the disadvantages and limitations of the manual system.

7.3.5.2 Security Preferences

This system only allow registered user who has a valid login name and password to access the system. This is to avoid unauthorized users from accessing the system.

7.3.5.3 User Friendly

This system can be manipulate not only by computer experts but also for beginners that do not have any background knowledge in computer fields. This is because most of the functions are located in buttons and icons and users do not have to explore the system to know how to use the functions. In addition to that, the system uses English language and that is an international language that most people in the world understand. The system also provides interface design that seems professional, interesting and also easy to use. Some of the command button also display an error message incase the operation is unsuccessful. It also provides a reminder message in helping user to operate the system.

7.3.5.4 More Flexible

HIS also provides high flexibility options that is the high ability to update. With this ability, health care centers are able to update easily patient's data along with the medical records and medicine records. This system is not only able to update, it is also very eficient in storing, removing, and adding patient's data and medicine records.

7.3.5.5 Current Trend

In this era of Information Technology, it is an effective and beneficial way to implement computerized system such as HIS at health care centers that provides services for outpatient. This is because systems like HIS are able to process a large amount of workload in just a short period of time compared to a manual system.

7.3.6 System Limitations

Even though the system has many useful strength, it also has several limitations. The limitations that was identified by the developer are:

7.3.6.1 Limited System Help

Since the development period is quite short, the developer was unable to create a complete help mechanism in using HIS. The help system planned to be implemented is just like the help available for Windows operating system and other Microsoft products such as Microsoft Office.

7.3.6.2 Acceptance Testing

Actually the HIS developed has not yet passed the real acceptance test. This is because there is not enough time to find a client that provides health care services to performed the acceptance test for the system in the real environment.

7.3.7 Future Enchancements

The developed system can be upgrade into a more effective system if more funtions can be added into the system. For example, the funtion of the system can be upgraded so that the system can delete all inactive patient record in a certain specified period of time. This will increase database storage capacity and prevent overflow of unimportant data. Only active patients' data will be stored for future reference.

To enchanced the ability of the system in the future, a detailed help menu is suggested to be implement. This menu must be able to help users in using the system without the help from other party especially an outside party such as the developer. This menu must also provide information to train users in using the system.

Besides that, it is also suggested that this HIS has the ability to operate in network mode. In order to support a multiuser environment, a single database can be shared among users. This sharing application can be implemented in three main technique. First, the HIS should be able to operate online. Second, is by locating the database of HIS at a central storage where all objects in the database are accessible by all users. Finally, designing the application based on the client/server model.

7.4 SUMMARY

From this chapter, all the problems and solutions taken by the developer had been discussed in details. Apart from that, all other evaluation had also been considered. Even though there are several limitations, it can be said that HIS has achieve its main objectives.

OVERALL SUMMARY

Basically this Health Information System has been developed to create a more efficient way of managing computerized patient recording process. It is developed to replace the manual system of recording increasing number of patients' record in health care centers. Many reviews and research have been made in completing this system.

Even though there are still several limitations, it can be conclude that HIS managed to achieve its objectives that is to create a paperless environment and introduce a more systematic way to record patients' information using computers. After further analysis, there are several new functions added to the system to improve the system's efficiency. The system also has a very interesting user interface that is user friendly. Besides that, it is certain that all the functions and behaviors of the system fulfill the system's requirements.

Finally, all of the experience gained during the development of this system gives new knowledge regarding ways of developing a system. Hopefully the system will meet user requirements and users will be able to gain benefit from it.

BIBLIOGRAPHY

[1] Kendall, K. & Kendall, J. (1999). **Systems Analysis And Design**, Fourth Edition: Prentice Hall International Inc.

[2] Sellapan, P. (1998). **Information System: A Management Perspective**: Sejana Publishing

[3] McManus, Jeffrof P. (1998). **Database Access With Visual Basic:** SAMS Publishing

[4] Davis Gordon B. & Olson Margrethe H. (1985). **Management Information System: Conceptual Foundation, Structures And Development.** Second Edition: Mc-Graw Hill Book Company

[5] Pressman, Roger S. (2001). **Software Engineering: A Practitioner's Approach,** Fifth Edition: McGraw-Hill

[6] Alan TH Khoo, Khoo Teik Lin (1998). **Building Application In Access 97: A Step By Step Approach Through Business Example**: Federal Publications Sdn. Bhd.

[7] Sheikh Othman Sheikh Salim (1993). **Kamus Dewan Edisi Baru**: Dewan Bahasa dan Pustaka

[8] Clayton Walnum (1998). **Complete Idiot's Guide To Visual Basic 6**: MacMillan Computing Publishing

[9] Sellapan, P. (2000). **Programming In Visual Basic 6**: Sejana Publishing

[10] Callahan Evan (1997). **Step By Step Microsoft Access 97 Visual Basic**: Microsoft Press

[11] Winemiller, E., Roff, J., Heyman, B. & Groom, R. (1999). **Visual Basic 6 Database How-To:** MacMillan Computing Publishing

[12] Mike Amundsen**. Teach Yourself Database Programming with Visual Basic 5 in 21 Days.** Second Edition: SAMS Publishing

[13] Reselman, B., Pruchniak, W., Peasley, R. & Smith, E. **Using Visual Basic 6:** MacMillan Computing Publishing

[14] Visual Basic

www.visualbasic.about.com/cs/database

www.vbweb.co.uk

www.freevbcode.com

www.planet-source-code.com

www.abstractvb.com

[15] C-DAC's Sushrut Hospital Information System (HIS)

http://www.cdacindia.com/html/his/register.asp (15 April 2002)

USER MANUAL

USER MANUAL

CONTENTS

Contents i

Figure List ii

INTRODUCTION 1

1.0 MINIMUM REQUIREMENTS 2

1.1 Hardware Requirements 2

1.2 Software Requirements 2

2.0 INSTALLATION PACKAGE 4

3.0 SYSTEM USAGE 6

3.0.1 Running The System 6

3.0.2 Patient Menu 10

3.0.3 Billing Menu 19

3.0.4 Pharmacy Menu 20

3.0.5 Inventory Menu 21

3.0.6 Reminder Menu 23

3.0.7 Search Menu 25

3.0.8 Exit Menu 26

USER MANUAL

CONTENTS

Contents i

Figure List ii

INTRODUCTION 1

1.0 MINIMUM REUIREMENTS 2

1.1 Hardware Requirements 2

1.2 Software Requirements 2

2.0 INSTALLATION PACKAGE 3

3.0 SYSTEM USAGE 6

3.0.1 Running The System 6

3.0.2 Patient Menu 10

3.0.3 Billing Menu 19

3.0.4 Pharmacy Menu 20

3.0.5 Inventory Menu 21

3.0.6 Reminder Menu 23

3.0.7 Search Menu 24

3.0.8 Exit Menu 25

FIGURE LIST

Rajah 2.1 Setup Screen 3

Rajah 2.2 Installation Screen 4

Rajah 2.3 Program Group Screen 5

Figure 3.0.1: Login Screen 6

Figure 3.0.2 Incorrect Username Message 7

Figure 3.0.3 Incorrect Password Message 7

Figure 3.0.4 Splash Screen 8

Figure 3.0.5 Main Interface Display 8

Figure 3.2.1: Patient Form Display Screen 10

Figure 3.2.2: Add new patient record screen 11

Figure 3.2.3: Message Regarding Empty NRIC Field 12

Figure 3.2.4: Patient Information Screen 14

Figure 3.2.5: Patient Prescription Record Screen 15

Figure 3.2.6: Empty Medicine ID Field 16

Figure 3.2.8: Add Prescription Information 17

Figure 3.2.9: Message Regarding Empty Quantity Field 18

Figure 3.2.10: Delete Confirmation Message 19

Figure 3.2.11: Delete Confirmation Message 19

Figure 3.2.12: Payment Record Screen 21

Figure 3.3.1: Billing Screen 23

Figure 3.3.2: Error Message When Company ID Is Empty 24

Figure 3.3.3: Error Message When There Are No Billing Records 24

Figure 3.3.4: Billing Information Screen 25

Figure 3.3.5: Message If There Are No Unpaid Charges 26

Figure 3.4.1: Pharmacy Form Display 28

Figure 3.5.1: Inventory list screen 29

Figure 3.5.2: Screen to add medicine stock to storage 30

Figure 3.6.1: Reminder Screen 31

Figure 3.7.1: Search Screen 32

Figure 3.8.1: Exit Message Box 33

USER MANUAL

INTRODUCTION

Welcome to Health Information System (HIS). HIS is a system specially developed to manage information regarding patient, medicine, prescription and also payment from patient. This system can be used by health care providers at their outpatient department (OPD) in which this system offers interesting and user friendly interface. This system also works as an inventiry system for managing medication records.

This system has a variety of ability that has been manipulated at optimum level. In the future, this system can be a virtual assistant in which it simplify updating process, easily records new types of medicine available and records the distribution of medications from the storage.

Security preferences is also available, such as password protection to prevent access to unauthorized users. Thus, increase the integrity of the system.

1.0 MINIMUM REQUIREMENTS

Health Information System can be used on any personal computer. The minimal requirements tu run the system is listed as below:

1.1 Hardware Requirements

- Intel Pentium MMX Processor or AMD 200Mhz
- Minimum 32MB RAM
- 4.0GB Hard Disk
- 2 MB of hard disk space to run the system using Compact Disc
- 24X CD-ROM
- Mouse
- Keyboard
- SVGA monitor (resolution 800*600)

1.2 Software Requirements

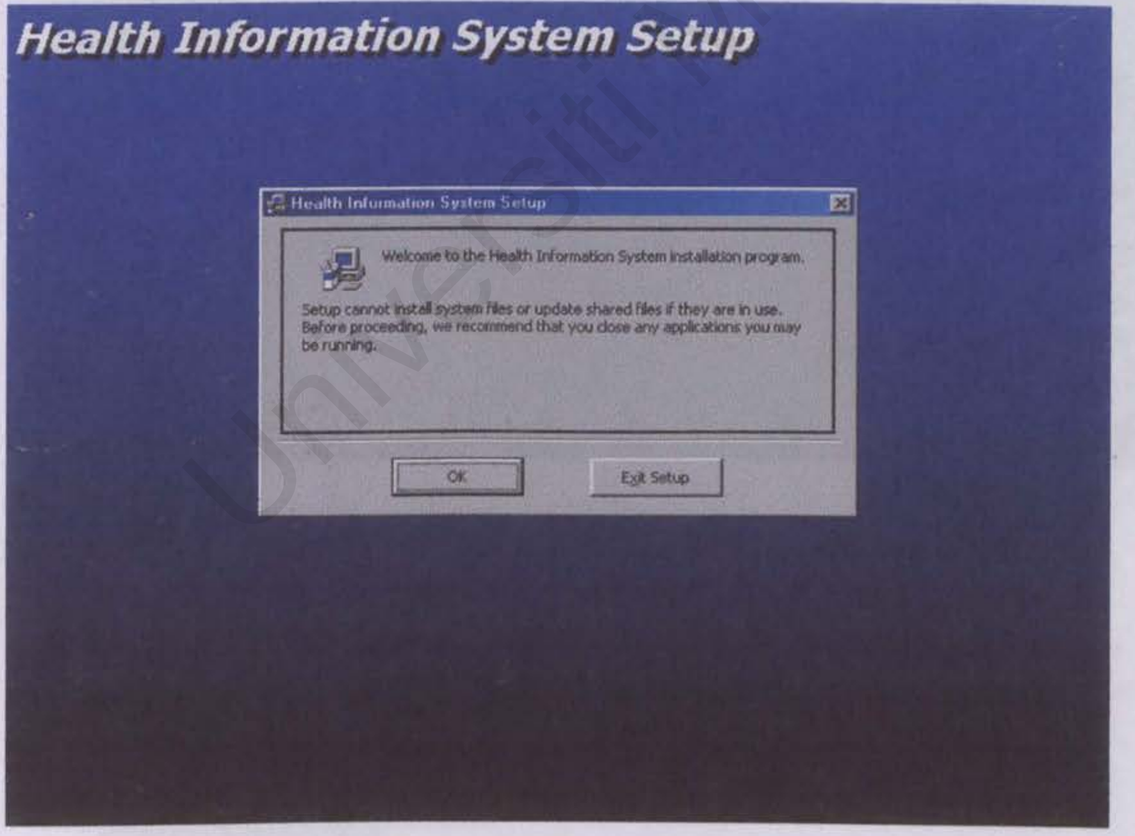
- Windows 95/98/Me
- HIS Application

2.0 INTRODUCTION PACKAGE

Installation on Win32 platform.

1. Before using this package, users must insert the Health Information System compact disc in to the CD-ROM drive.

2. Users must access the Health Information System directory and click on the setup launcher file that is setup.exe. This will activate the installation wizard as shown on the Figure below.



Health Information System Setup

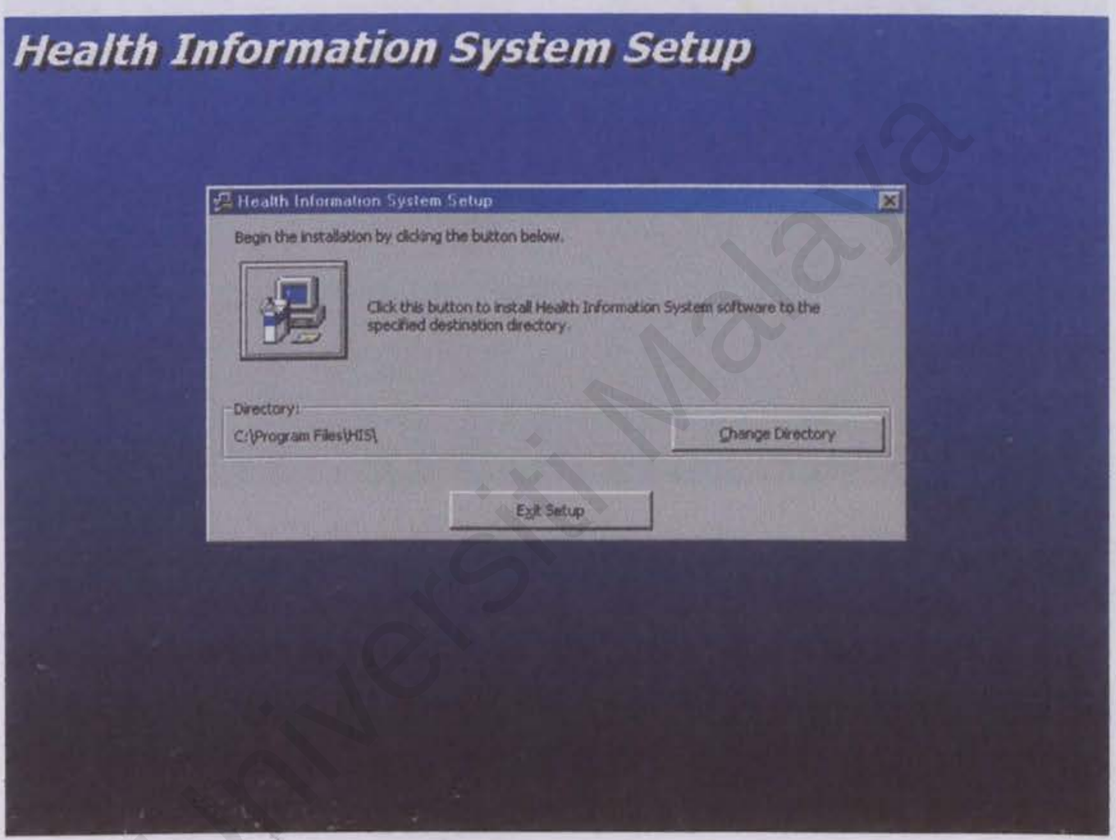
Health Information System Setup

Welcome to the Health Information System installation program.

Setup cannot install system files or update shared files if they are in use. Before proceeding, we recommend that you close any applications you may be running.

Figure 2.1: Setup Screen

3. Click on the "OK" button to proceed. Users will be given a choice to install the application on the suggested directory that is "C:\Program Files\HIS\". Users can change the location or the name of the directory by clicking on the "Change Directory" button.



Health Information System Setup

Health Information System Setup

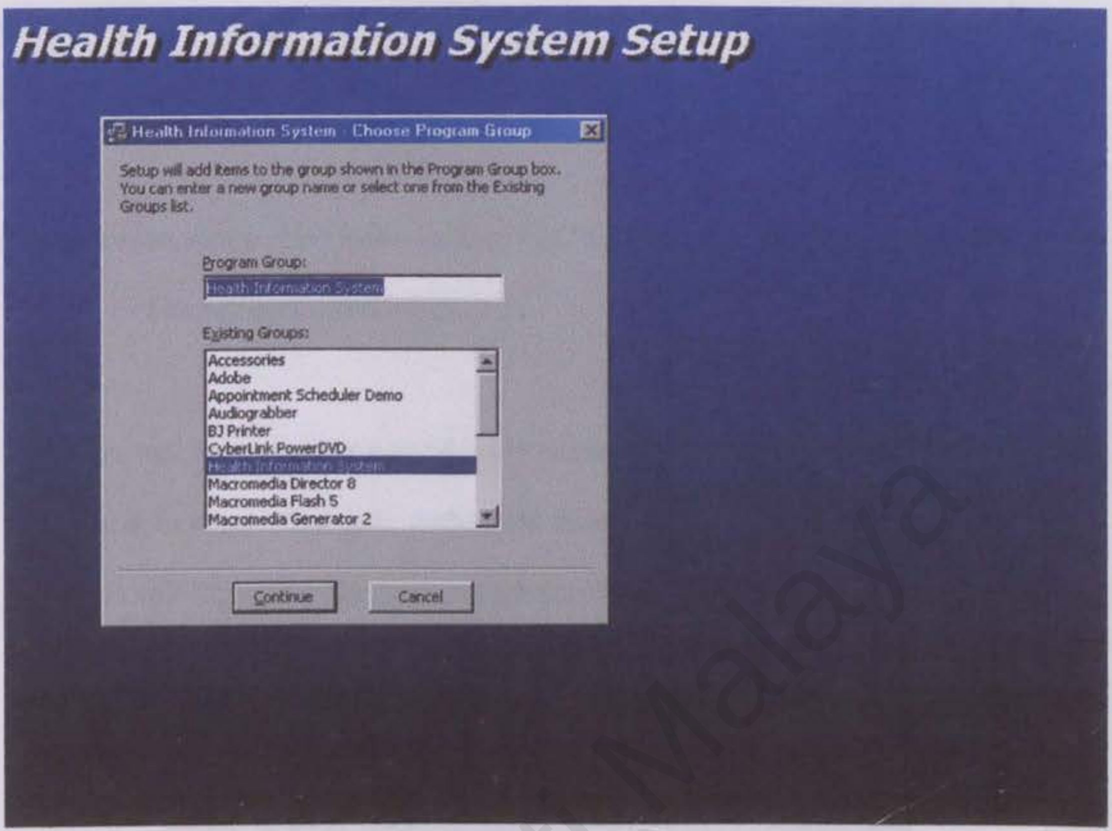
Begin the installation by clicking the button below.

Click this button to install Health Information System software to the specified destination directory.

Directory:
C:\Program Files\HIS\

Figure 2.2: Installation Screen

4. Click on the computer icon to install system in the selected directory. "Program Group" selection screen will be displayed after the computer icon is clicked as shown in Figure below:


Health Information System Setup

Health Information System - Choose Program Group

Setup will add items to the group shown in the Program Group box. You can enter a new group name or select one from the Existing Groups list.

Program Group:
Health Information System

Existing Groups:
Accessories
Adobe
Appointment Scheduler Demo
Audiograbber
BJ Printer
CyberLink PowerDVD
Health Information System
Macromedia Director 8
Macromedia Flash 5
Macromedia Generator 2

Figure 2.3 Program Group Screen

5. Next, click on the "Continue" button to complete the setup process. Finally, a confirmation screen will be displayed if the Health Information System is successfully installed in the computer.

3.0 SYSTEM USAGE

3.0.1 Running the system

1. Users can access this Health Information System package in the path below:

   Start\Programs\HIS\HIS

2. Users will have to enter a valid login name and password to access the system. This is to ensure that only authorized users are allowed to access the system. Users will be required to enter the correct username and password in at the login screen as shown in Figure

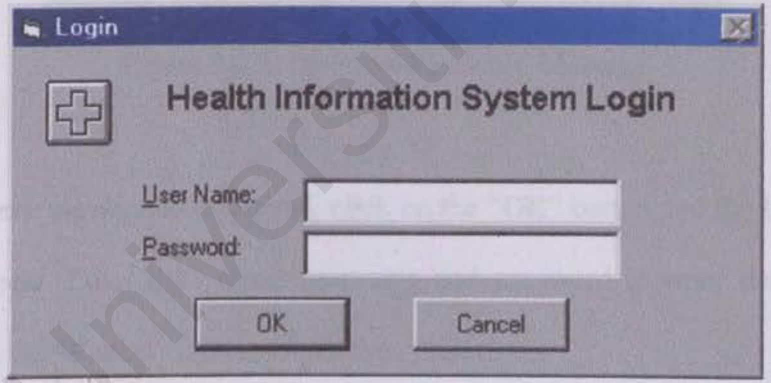
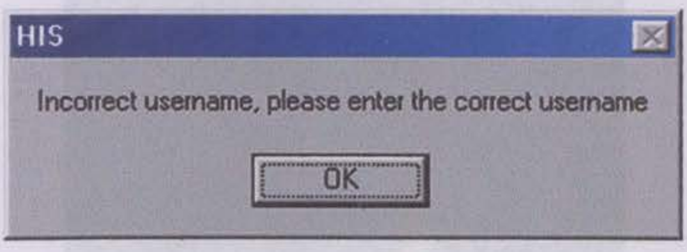


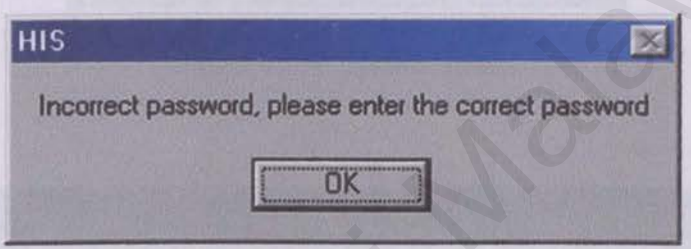
Figure 3.0.1: Login Screen

3. If the user name and password is incorrect, the system will display an error message stating the the username or password entered is incorrect.


HIS

Incorrect username, please enter the correct username

Figure 3.0.2 Incorrect Username Message


HIS

Incorrect password, please enter the correct password

Figure 3.0.3 Incorrect Password Message

4. If the error message is displayed, click on the "OK" button and the login screen will appear. Enter the correct username and password to enter the system ot click on the "Cancel" button to exit the system.

5. After the user has entered a valid username and password, the system will display a splash screen for 3 seconds before displaying the main interface for the system. The screen is as shown the Figure 3.0.4 .

Figure 3.0.4 Splash Screen

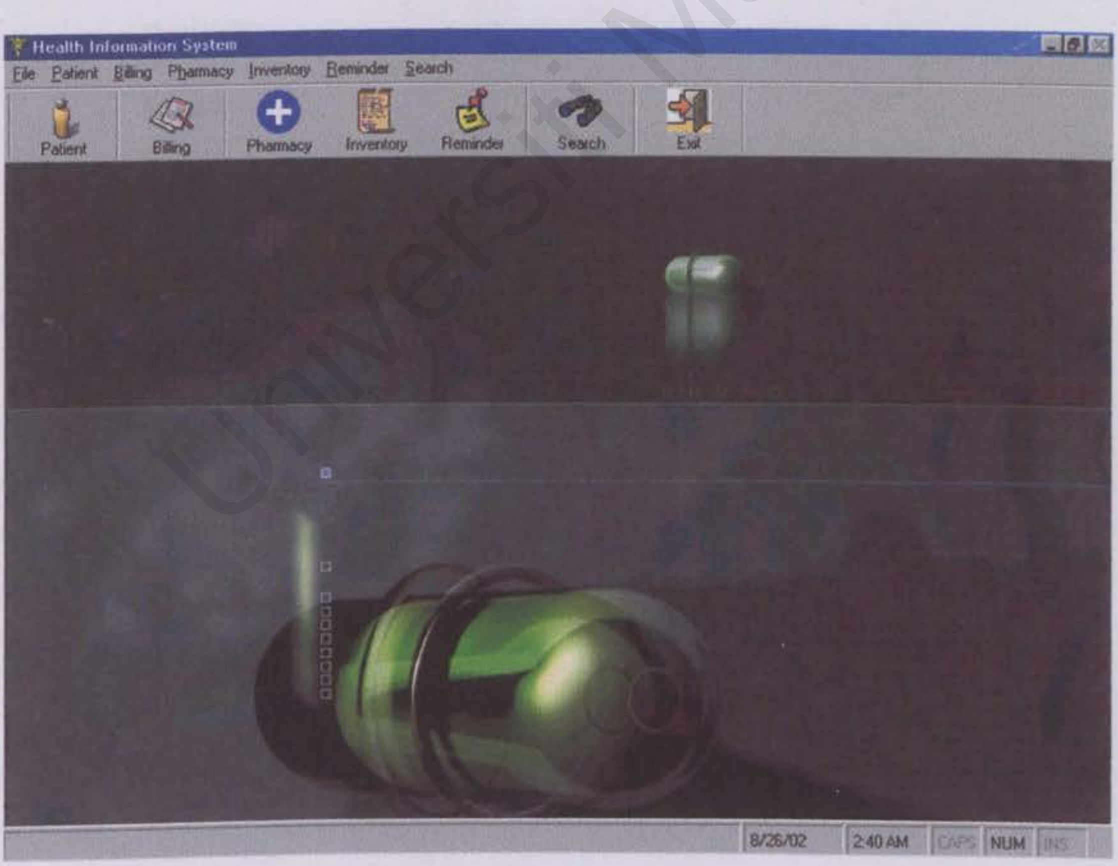


Figure 3.0.5 Main Interface Display

6. From this screen, user can access to the desired form using the menu and buttons located on top of the screen.

7. The next part of this manual will discuss about the contents of each menu. As shown in Figure , there are seven main menus. The menus are:

    a. Patient
    b. Billing
    c. Pharmacy
    d. Inventory
    e. Reminder
    f. Search
    g. Exit

3.0.2 Patient Menu

1. This menu is intended to record any information regarding the patient. The information can be added, erased, edit and viewed from the HIS database.

2. When the menu button is pressed, the system will show the patient form.

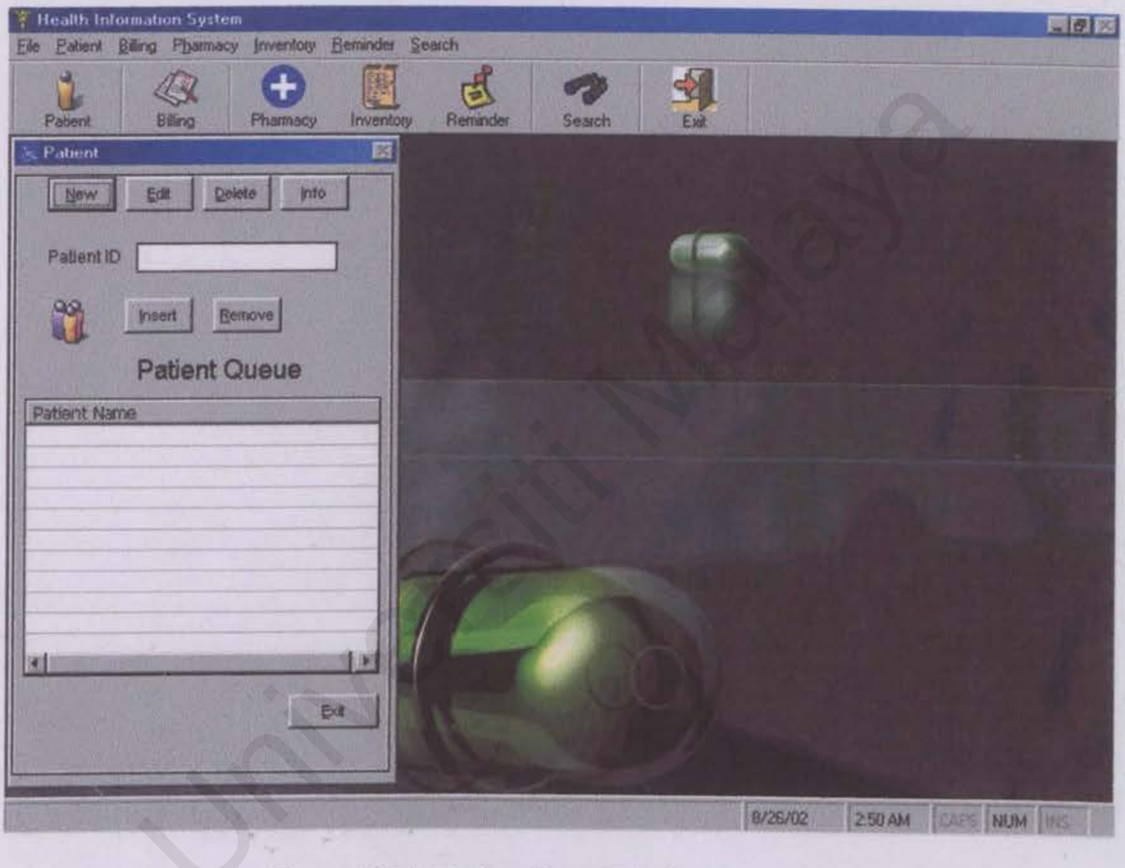
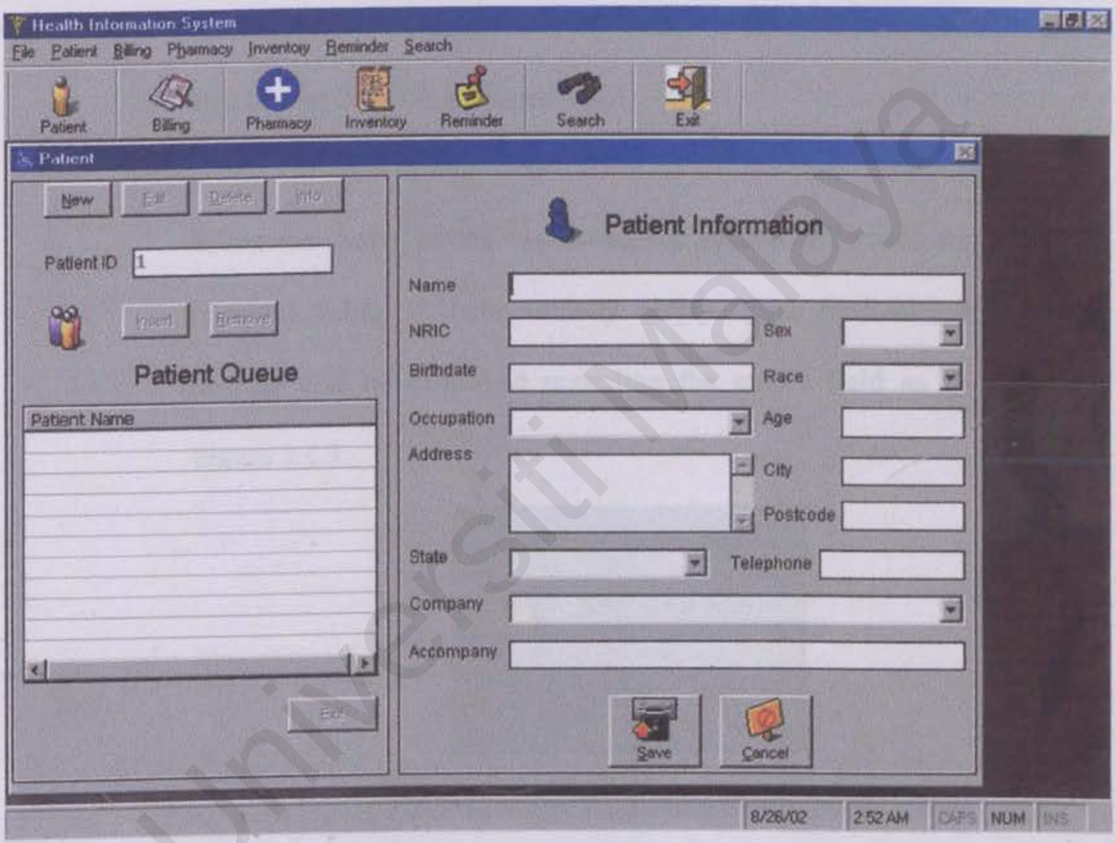


Figure 3.2.1: Patient Form Display Screen

3. There are buttons at the form toolbar that represents specific function. The function of the buttons are as follows:

i. New

Function: Add new record to the databse. When the'New' button is pressed, full view of the Patient form will be displayed as shown in Figure 3.2.2. All the text boxes will be automatically cleared and the cursor will be at the Patient Name text box.



Patient ID 1

8/26/02 2:52 AM

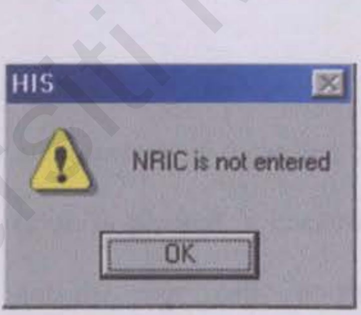
Figure 3.2.2: Add new patient record screen

There are 2 graphical buttons that will be displayed along with it, which is the 'Save' and 'Cancel' button.

The 'Save' button is used to store all the data keyed-in form in the HIS database.

The 'Cancel' button is used when the user decide not to add new information or it also can be used if the user accidently pressed the 'New' button. When the'Cancel' button is pressed, the Patient form will be displayed as in Figure 3.2.1.

When the 'Save' button is pressed, the system will check for all the required fields, if there are any empty space available, an error messge will be displayed regarding the empty field as shown in Figure 3.2.3.



NRIC is not entered

Figure 3.2.3: Message Regarding Empty NRIC Field

The Patient ID is automatically generated by the system to prevent redundancies.

ii. Edit

Function: Edit all information regarding the patient in the patient form. When the 'Edit' button is pressed, the patient form will be at full display along with the selected patient information and the 'Update' and 'Cancel' button as shown in Figure 3.2.2.

The 'Cancel' button is the same button as before shown in Figure 3.2.2.

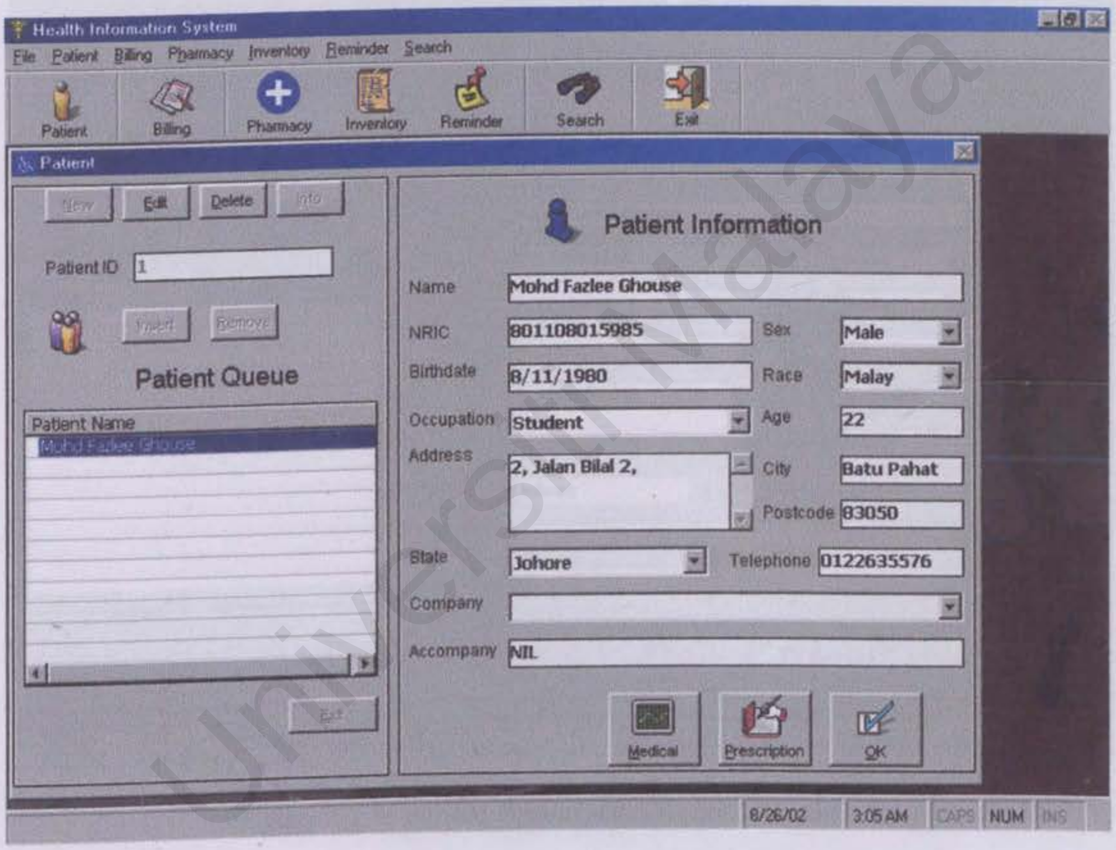
The 'Update' button is used to store all the updated fields in the HIS database.

iii. Delete

Function: To delete all information regarding the selected patient. When the 'Delete' button is pressed, a confirmation message will appear whether or not the user really wants to delete all the information regarding the selected patient. This confirmation message is to ensure that records are not deleted accidently by pressing the 'Delete' button. Record deleted from the HIS database can no longer be retrieved.

iv. Info

Function: To display information regarding seected patient. Figure below shows the screen that will appear when the 'Info' button is pressed. Besides that, patient information can also be retrieved by double clicking on the name that appears on the queue list.



Patient ID: 1

Patient Name: Mohd Fazlee Ghouse

Name: Mohd Fazlee Ghouse

NRIC: 801108015985

Sex: Male

Birthdate: 8/11/1980

Race: Malay

Occupation: Student

Age: 22

Address: 2, Jalan Bilal 2,

City: Batu Pahat

Postcode: 83050

State: Johore

Telephone: 0122635576

Company:

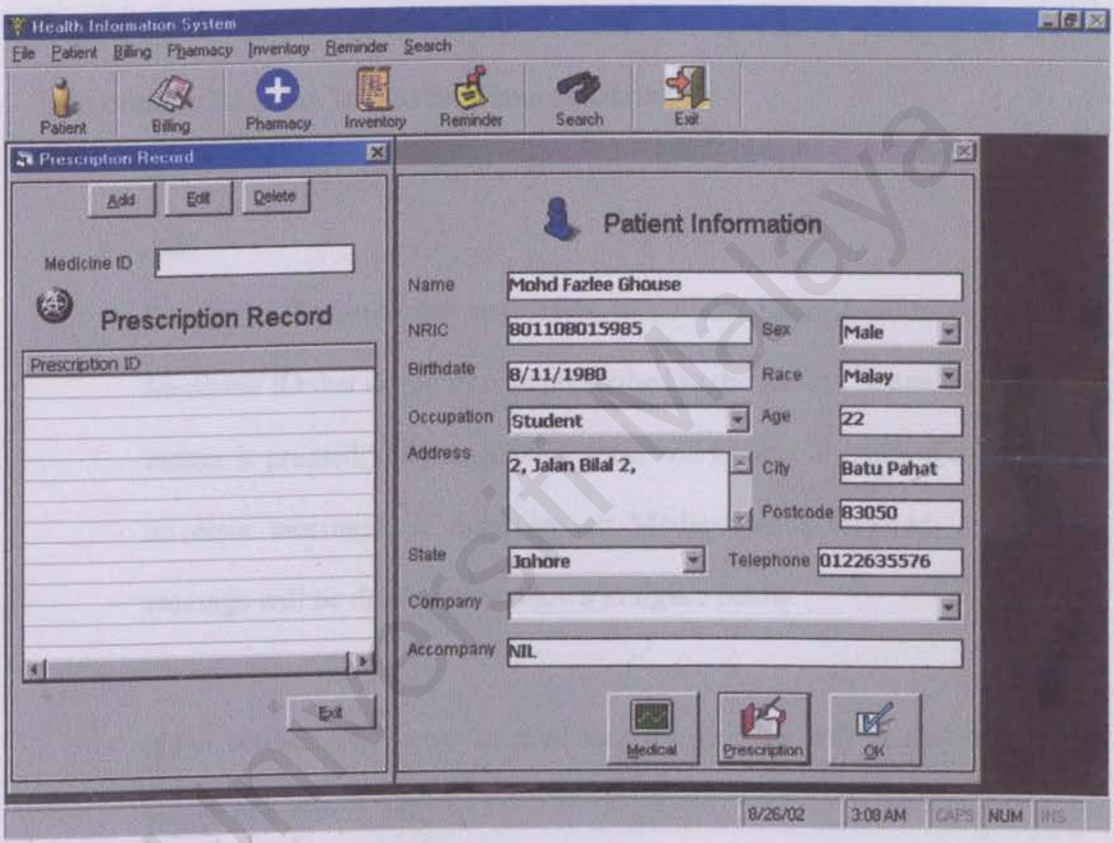
Accompany: NIL

8/26/02 3:05 AM CAPS NUM INS

Figure 3.2.4: Patient Information Screen

There are 3 graphical buttons in the patient information screen that is the 'Prescription' 'Medical' and 'OK' button.

When the 'Prescription Record' button is pressed, the system will display the entire patient's prescription record. Basically this button displays the prescription history of a specific patient. Figure below shows the screen that will be displayed when the 'Prescription' button is pressed.



Name: Mohd Fazlee Ghouse
NRIC: 801108015985
Sex: Male
Birthdate: 8/11/1980
Race: Malay
Occupation: Student
Age: 22
Address: 2, Jalan Bilal 2,
City: Batu Pahat
Postcode: 83050
State: Johore
Telephone: 0122635576
Company:
Accompany: NIL

8/26/02 3:08 AM

Figure 3.2.5: Patient Prescription Record Screen

Prescription

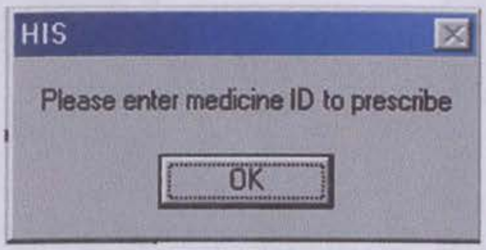
1. This prescription form enables users to prescribe medications to patients. Prescriptions item can be add, edit or delete.

2. There are three buttons available that is 'Add', 'Edit' and 'Delete'. Information regarding the prescribed items can be retrieved by double clicking on the Prescription ID listed. Button functions as follows:

   i. Add

   Function: To prescribe medication to patient. Users have to enter the Medicine ID that is going to be prescribed to the patient. When the 'Add' button is pressed, the prescription form will be in full mode if there are no error messages. If there are no Medication ID entered, an error message will be displayed as shown in figure below.

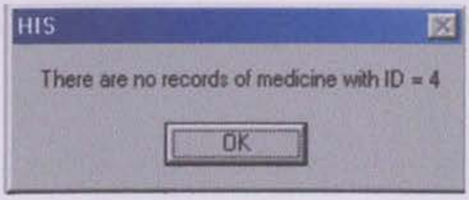
   If the Medicine ID is not entered an error message will be displayed as shown in Figure 3.2.6.



Please enter medicine ID to prescribe

Figure 3.2.6: Empty Medicine ID Field

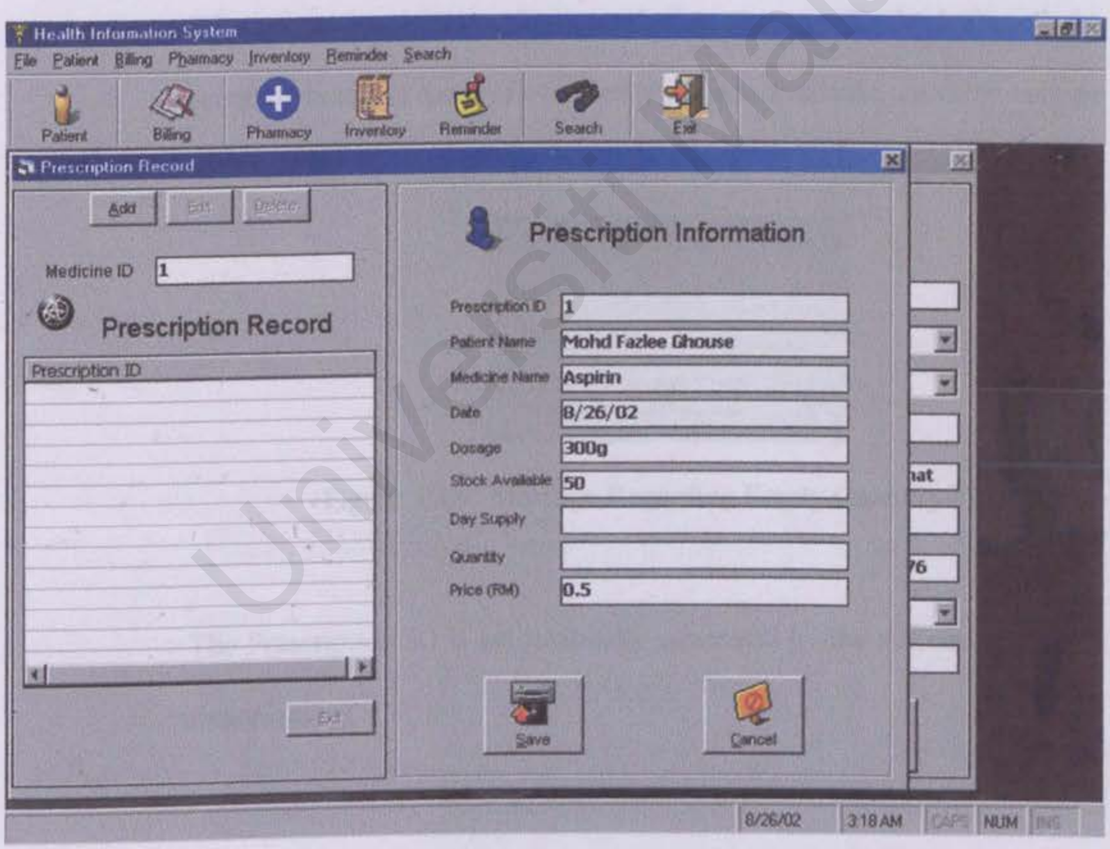
If an invalid Medicine ID is entered an error message will be display as shown in Figure 3.2.7.



There are no records of medicine with ID = 4

Figure 3.2.7: Invalid Medicine ID Message

If the Medicine ID is valid, users will have to enter the Quantity and the Supply Day of the medicine.



Medicine ID	1
Prescription ID	1
Patient Name	Mohd Fazlee Ghouse
Medicine Name	Aspirin
Date	8/26/02
Dosage	300g
Stock Available	50
Day Supply	
Quantity	
Price (RM)	0.5

8/26/02 3:18 AM CAPS NUM INS

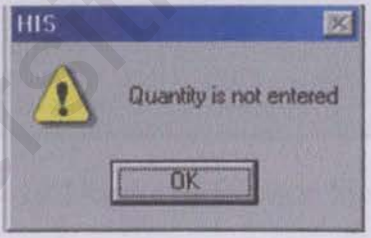
Figure 3.2.8: Add Prescription Information

At this point, there are two graphical buttons available that is the'Save' and the 'Cancel' button.

The 'Save' button is used to store all the data keyed-in form in the HIS database.

The 'Cancel' button is used when the user decided not to add prescibe any medicine to the patient When the'Cancel' button is pressed, the Prescription form will be displayed as in Figure 3.2.1.

When the 'Save' button is pressed, the system will check for all the required fields, if there are any empty space available, an error messge will be displayed regarding the empty field as shown in Figure 3.2.3.



Quantity is not entered

Figure 3.2.9: Message Regarding Empty Quantity Field

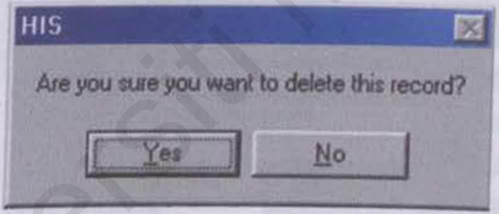
The Prescription ID is automatically generated by the system to prevent redundancies.

ii. Edit

Function: To edit the prescription information of a patient. Only the Supply Day field can be edit by the user. New prescription will be needed to add the quantity of medicine given to patient. The 'Update' and 'Cancel' button has the same function as in the Patient form.

iii. Delete

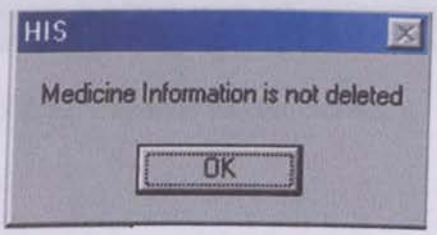
Function: To delete information of a prescription. A message will appear as shown in Figure 3.2.10 if the 'Delete' button is pressed to delete the selected item. This is to confirm that the 'Delete' button is not pressed accidently.



Are you sure you want to delete this record?

Figure 3.2.10: Delete Confirmation Message

If the user chooses not to delete the information, a message will appear stating that the information is not deleted.



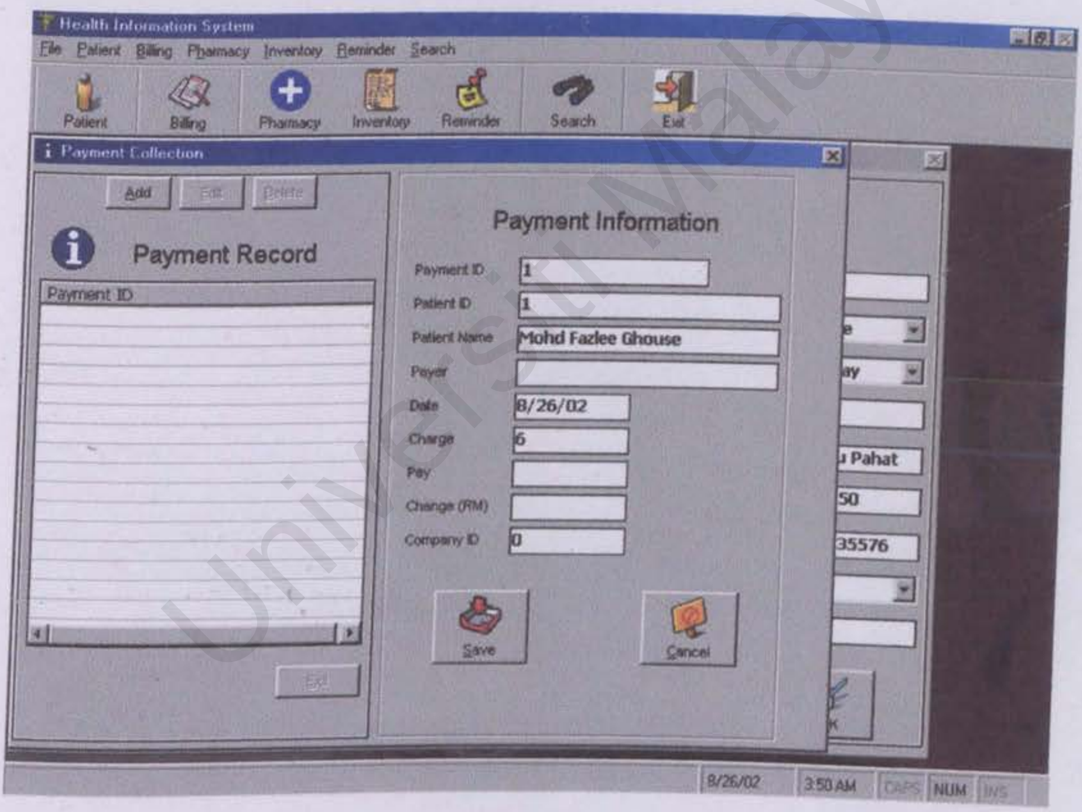
Medicine Information is not deleted

Figure 3.2.11: Delete Confirmation Message

Payment

1. Once a prescription is made and the 'Exit' button is pressed. The 'Payment' button will appear.

2. If a company pays for a patient's expenses, and the company's name is selected during registration, the pay amount will automatically set to zero. If not then the patient will have to pay cash.



Payment ID: 1
Patient ID: 1
Patient Name: Mohd Fazlee Ghouse
Payer:
Date: 8/26/02
Charge: 6
Pay:
Change (RM):
Company ID: 0

8/26/02 3:50 AM

Figure 3.2.12: Payment Record Screen

3. The 'Add', 'Edit' and 'Delete' buttons perform the same function as in the Prescription form.

Medical

1. This medical form stores information regarding the patient health record such as doctor's name, diagnosis and remarks.
2. The buttons available are 'New', 'Edit' and 'Delete'. All the buttons has the same function as in the prescription form only that it applies to the medical records. Double clicking on the Medical ID in the list retrieve information regarding the patient health records.

B) Billing Menu

1. This menu is used to record payments from companies that pay for their workers medical expenses. All the charges from patients with the same company will be total up to be paid by the company.

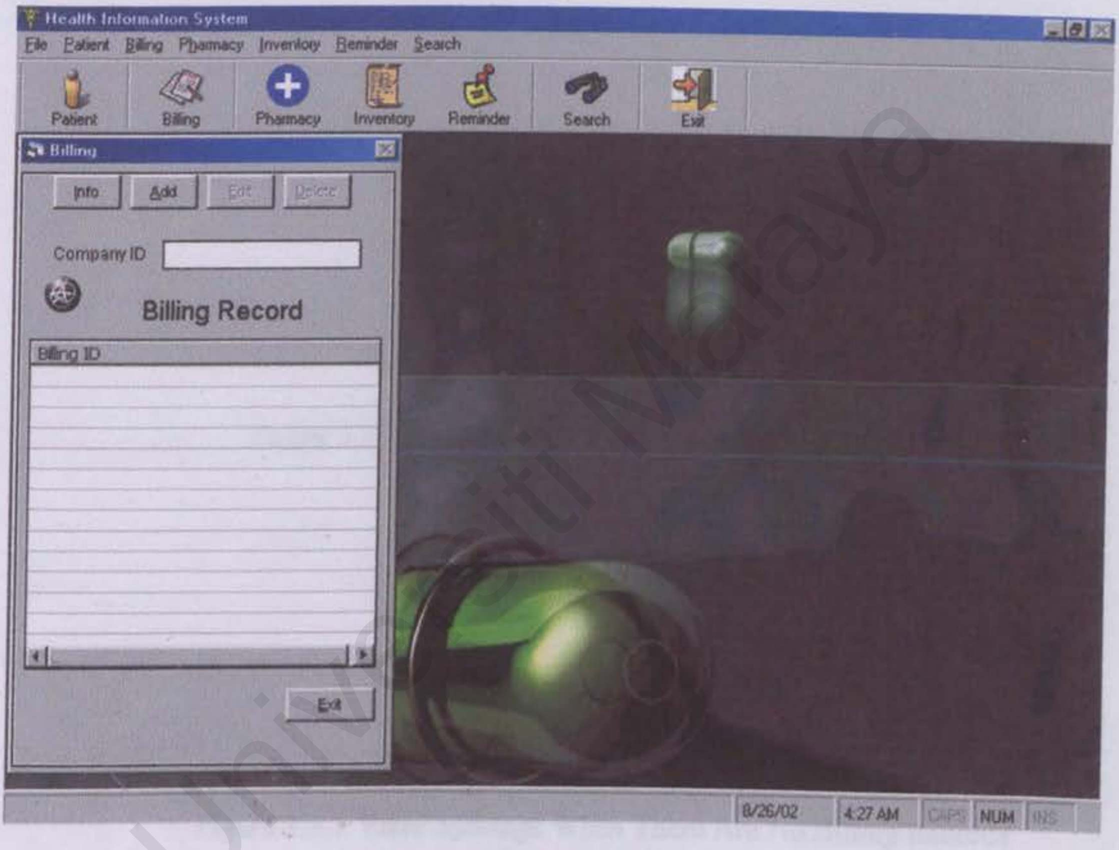
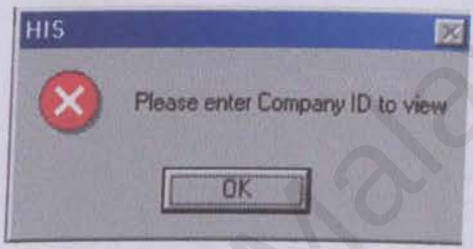


Figure 3.3.1: Billing Screen

2. There are four buttons at the top of the billing screen. Their function are as follows:

   i. Info

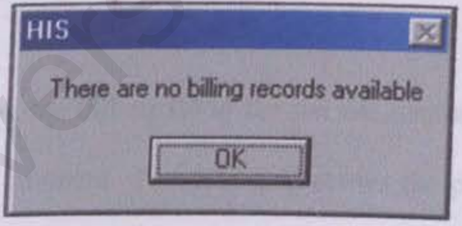
      Function: To view all of the company billing records. If there are no Company ID entered or if there are no history of records of that company an error message will be displayed as shown in figure below:



HIS

Please enter Company ID to view

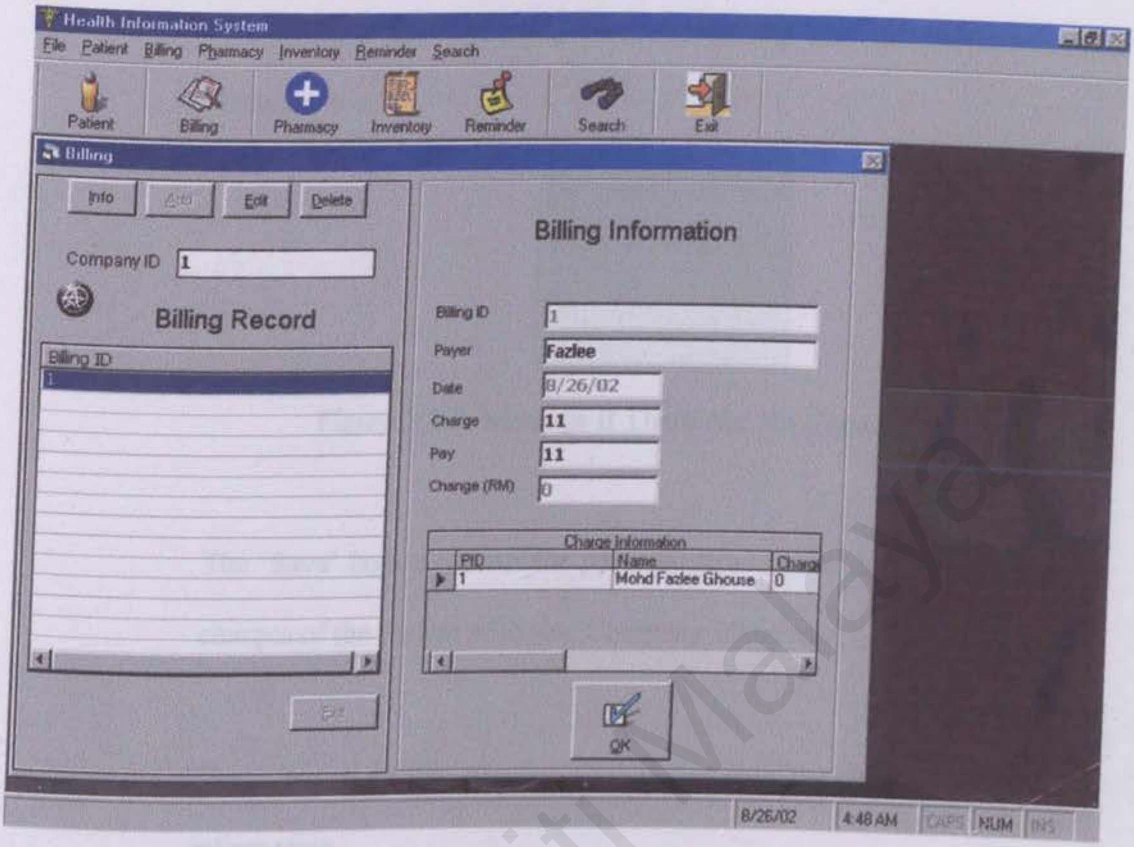
Figure 3.3.2: Error Message When Company ID Is Empty



HIS

There are no billing records available

Figure 3.3.3: Error Message When There Are No Billing Records



Company ID 1

Billing Record

Billing ID
1

Billing Information

Billing ID 1

Payer Fazlee

Date 8/26/02

Charge 11

Pay 11

Change (RM) 0

Charge Information

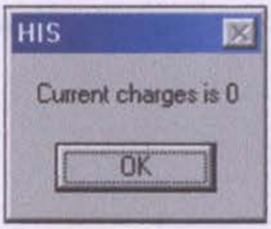
PID	Name	Charge
1	Mohd Fazlee Ghouse	0

Figure 3.3.4: Billing Information Screen

Double clicking on the billing ID in the list wil display the information regarding the bill payment. The data grid shows the patient ID, patient name and the charges of patients with the equiivalent company ID.

ii. Add

Function: To add payment records of a company. If the Company ID entered does not exist an error message will be displayed as shown in Figure 3.3.2. If the current charges applied to the company is zero then the system will display a message stating that the current account balance is zero.



Current charges is 0

Figure 3.3.5: Message If There Are No Unpaid Charges

The 'Save' button records the payment from the company and resets the charges of the patient with that Company ID to zero.

The 'Cancel' button can be used if the user decided not to save the information.

iii. Edit

Function: This 'Edit' button will only be enabled if there are any exixting record that can be modified. Otherwise this button will be disabled. It can only be used to modify the name of the payer. Other fields are locked to protect data integrity.

The 'Update' button can be use to update the records that have been modified.

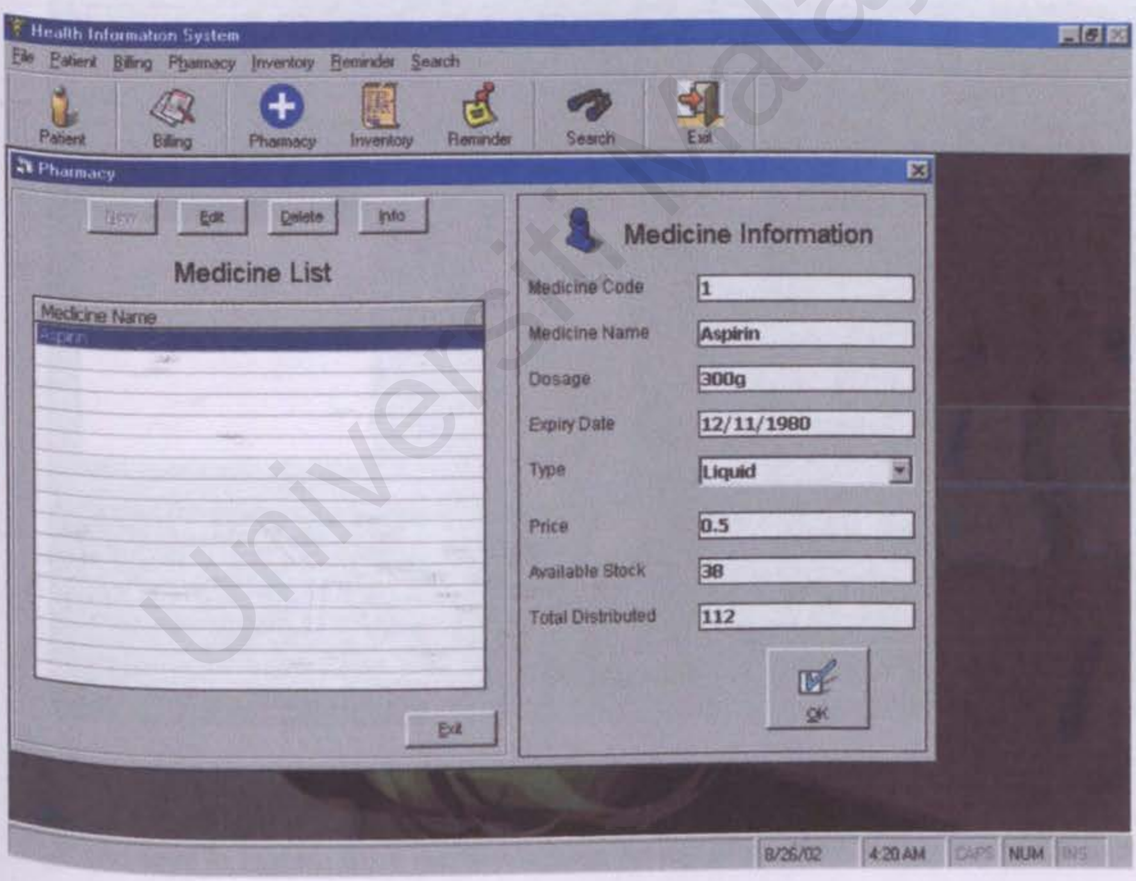
iv. Delete

Function: This 'Delete' button will only be enabled if there are any exixting billing record. It can be used to delete a specific billing records.

C) **Pharmacy Menu**

1. Basically this pharmacy form looks like the medical form and it uses the same concept. All the buttons perform the same function. Only in this form it is about medicine.

2. This menu enables the user to record any information regarding a specific medicine. The information can be stored, retrived, edit and also can be deleted from the HIS database.



Medicine List

Medicine Name
Aspirin

Medicine Information

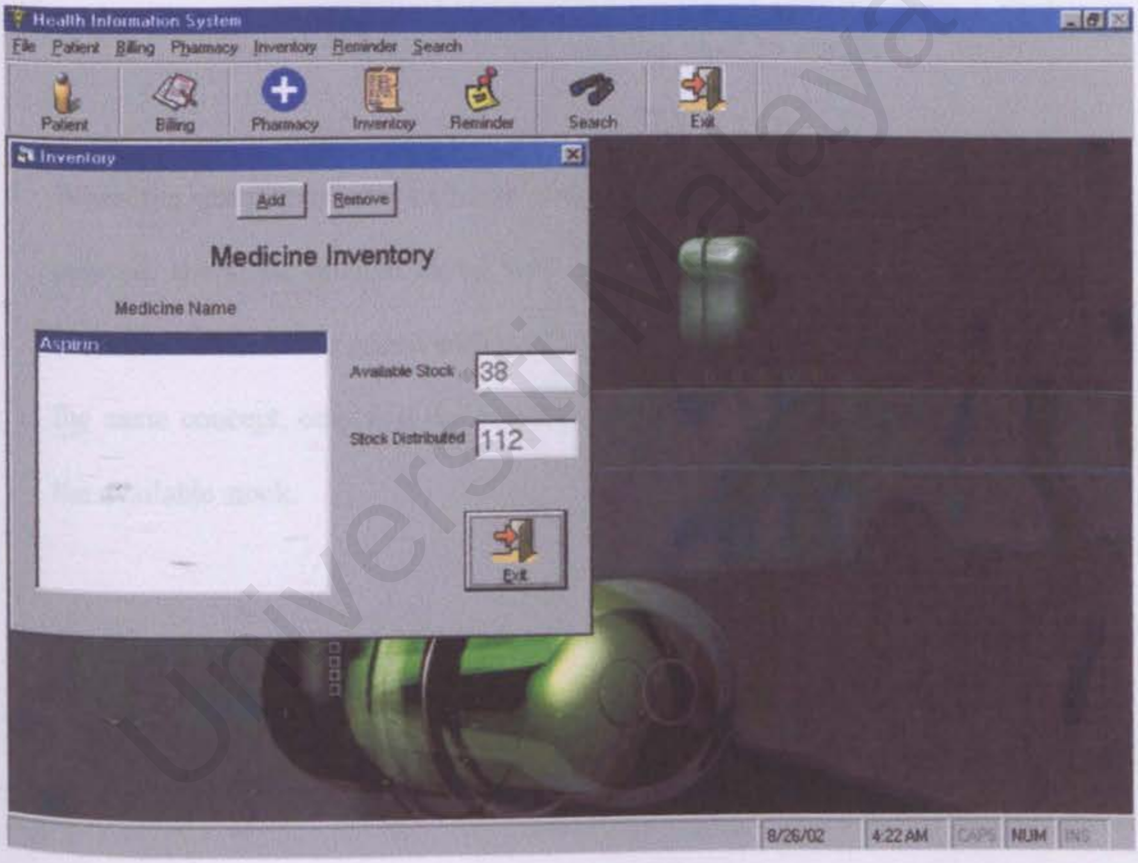
Medicine Code	1
Medicine Name	Aspirin
Dosage	300g
Expiry Date	12/11/1980
Type	Liquid
Price	0.5
Available Stock	38
Total Distributed	112

Figure 3.4.1: Pharmacy Form Display

D) Inventory Menu

1. This menu enables user to view the inventory record of a specific medicine in stock. This function allows user to know and control the amount of medicine stock available and also the amount of medicine stock distributed.

2. Figure shows the inventory form.



Medicine Inventory

Medicine Name: Aspirin

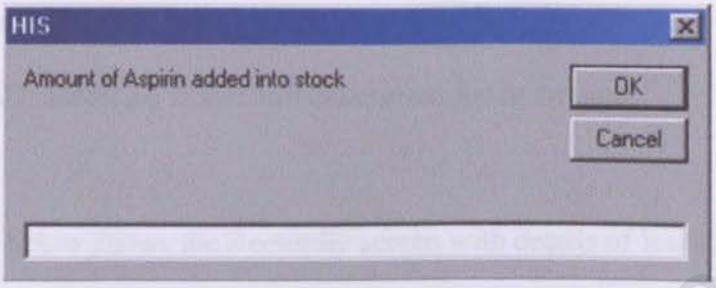
Available Stock: 38

Stock Distributed: 112

Figure 3.5.1: Inventory list screen

3. As seen in Figure, there are two buttons on top of the form that is the 'Add' and the 'Remove' button. The main purpose of these buttons is to add and to remove the specified amount of stock to be added or removed from the stock. Figure, showing the input screen box that appears when the 'Add' button is pressed. The

input box requests the amount of stock to be added into the current stock amount.



Amount of Aspirin added into stock

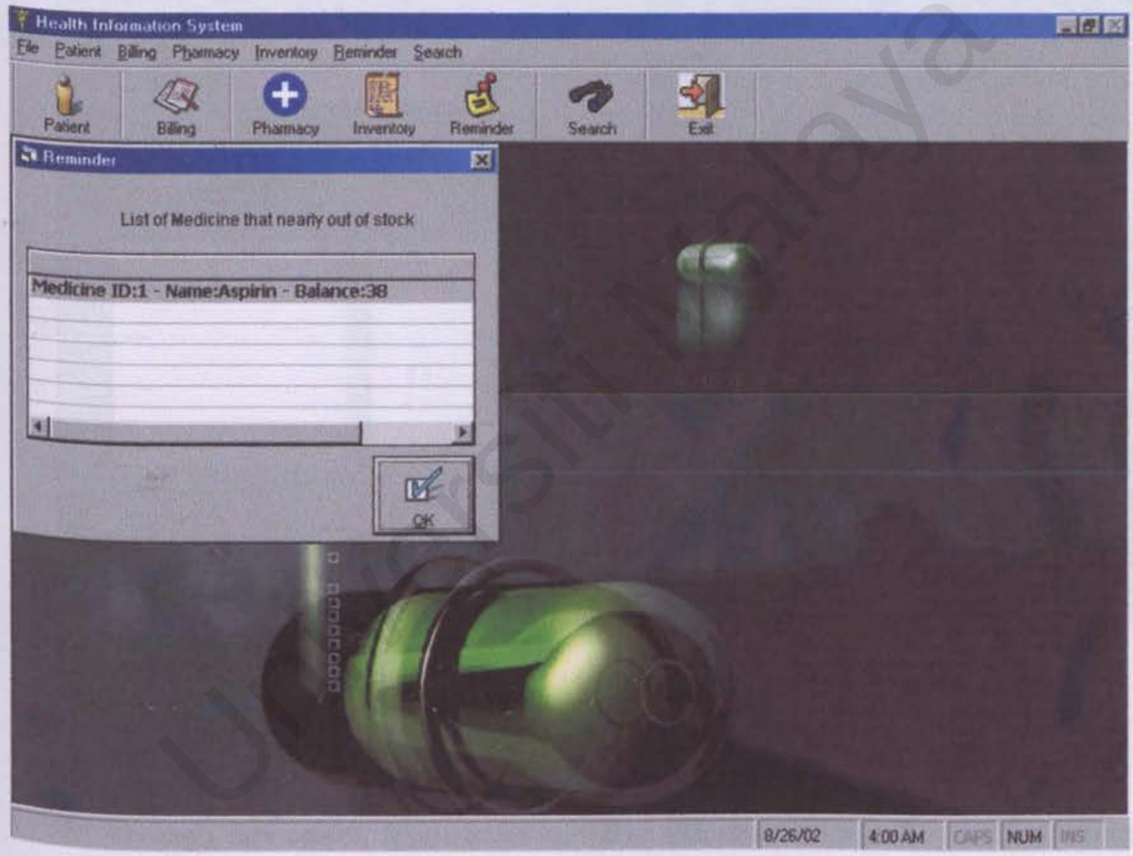
Figure 3.5.2: Screen to add medicine stock to storage

4. When the quantity of the medicine's stock is entered and the 'OK' button is pressed, the stock amount listed will updated automatically. The amount of available stock will be added with new stock amount. The 'Remove' button uses the same concept, only that the remove button reduces the stock amount from the available stock.

E) Reminder Menu

1. This menu enables the system to inform the user if the stock amount of a specific medicine is below the amount of 50. The reminder form will show the Medicine ID, Medicine Name and the current Stock amount.

2. The figure below shows the Reminder screen with details of low stock amount.



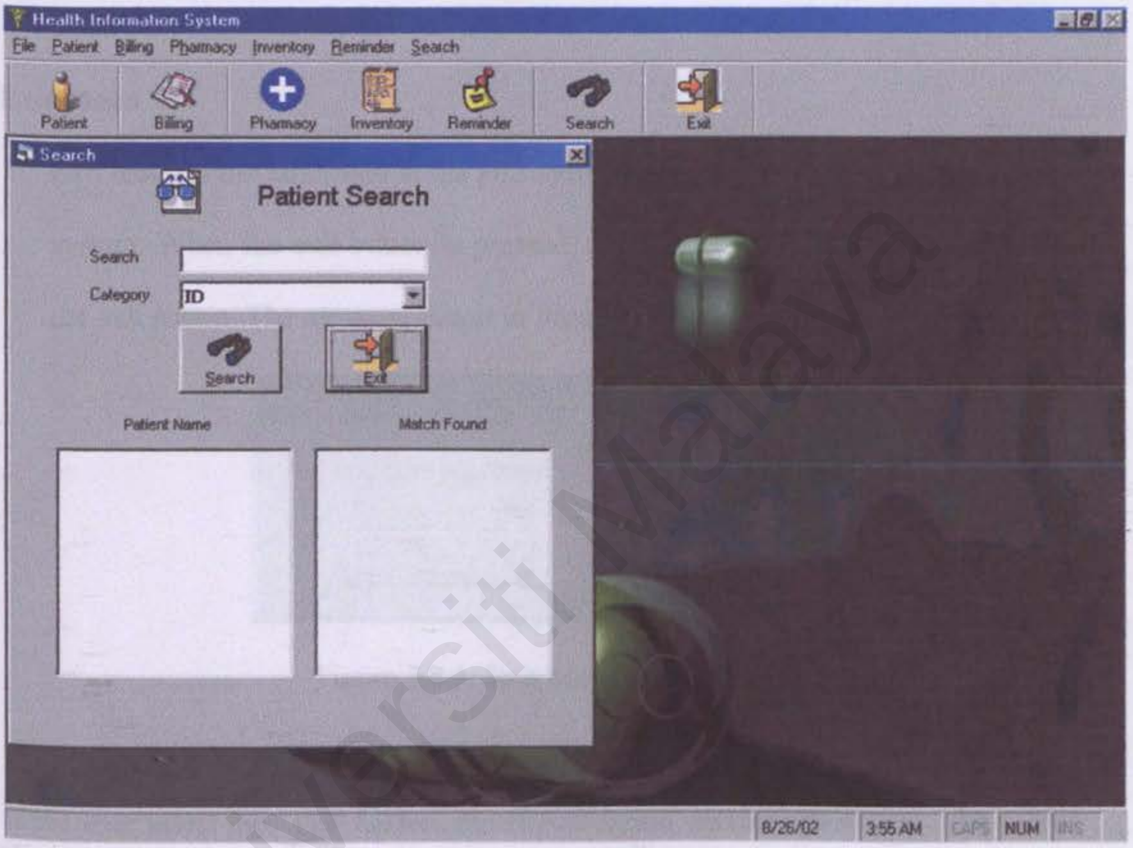
List of Medicine that nearly out of stock

Medicine ID:1 - Name:Aspirin - Balance:38

Figure 3.6.1: Reminder Screen

F) **Search Menu**

1. This menu is used for searching related information regarding patient using the patient NRIC, ID or Company.



Search

Category: ID

Figure 3.7.1: Search Screen

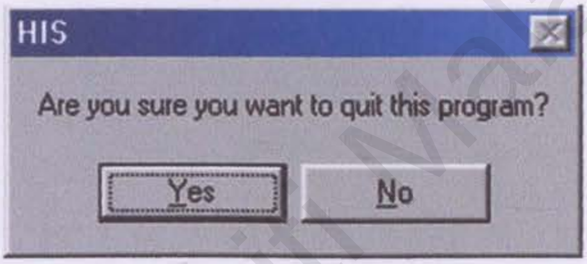
2. Through this search form, users can find information regarding patient using 'exact word' search method.

3. The 'Search" button will search for all records related to the keyword entered by user when the 'Search' button is pressed. All the results will be displayed in the

list box. Additional information regarding the search results can be obtained by double clicking the result at the 'match' column.

G) Exit Menu

1. Exit menu is the last menu in the HIS system. This menu enables user to exit the system. When the exit button is pressed, a message box will appear to confirm the exit action. The message box is in Figure 3.9.1.



Are you sure you want to quit this program?

Figure 3.8.1: Exit Message Box

2. If the user select the 'Yes' button, the HIS program will be terminated. If the 'No' button is pressed the user will be returned to the main screen.

CODING

frmLogin – Prevent unauthorized access to the system

```
Option Explicit
Private Declare Function DeleteMenu Lib "user32" (ByVal hMenu As Long, ByVal
nPosition As Long, ByVal wFlags As Long) As Long
Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal
bRevert As Long) As Long
Private Const MF_BYPOSITION = &H400&
Public LoginSucceeded As Boolean

'Function to remove the windows close button (X)
Private Sub RemoveMenus()
   Dim hMenu As Long
   hMenu = GetSystemMenu(hWnd, False)
   DeleteMenu hMenu, 6, MF_BYPOSITION
End Sub

Private Sub Form_Load()
RemoveMenus
End Sub

'If cancel button is pressed [illegible]
Private Sub cmdCancel_Click()
    LoginSucceeded = False
    Dim out
    out = MsgBox("Are you sure you want to exit this program?", vbYesNo, "HIS")
    If out = vbYes Then
    End
    ElseIf out = vbNo Then
    txtUserName.SetFocus
    Exit Sub
    End If
End Sub

'Check for correct password and username
Private Sub cmdOK_Click()
    If txtPassword = "password" And txtUserName = "HIS" Then
        LoginSucceeded = True
        Unload Me
        frmSplash.Show
    Else
        If txtUserName = "HIS" Then
        MsgBox "Incorrect password, please enter the correct password", , "HIS"
        txtUserName.SetFocus
        Else
        MsgBox "Incorrect username, please enter the correct username", , "HIS"
```

frmLogin – Prevent unauthorized access to the system

```
Option Explicit
Private Declare Function DeleteMenu Lib "user32" (ByVal hMenu As Long, ByVal
nPosition As Long, ByVal wFlags As Long) As Long
Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal
bRevert As Long) As Long
Private Const MF_BYPOSITION = &H400&
Public LoginSucceeded As Boolean

'Function to remove the windows close button ('x')
Private Sub RemoveMenus()
   Dim hMenu As Long
   hMenu = GetSystemMenu(hWnd, False)
   DeleteMenu hMenu, 6, MF_BYPOSITION
End Sub

Private Sub Form_Load()
RemoveMenus
End Sub

'If cancel button is pressed quit HIS
Private Sub cmdCancel_Click()
   LoginSucceeded = False
   Dim out
   out = MsgBox("Are you sure you want to quit this program?", vbYesNo, "HIS")
   If out = vbYes Then
   End
   ElseIf out = vbNo Then
   txtUserName.SetFocus
   Exit Sub
   End If
End Sub

'Check for correct password and username
Private Sub cmdOK_Click()
   If txtPassword = "password" And txtUserName = "HIS" Then
      LoginSucceeded = True
      Unload Me
      frmSplash.Show
   Else
      If txtUserName = "HIS" Then
      MsgBox "Incorrect password, please enter the correct password", , "HIS"
      txtUserName.SetFocus
      Else
      MsgBox "Incorrect username, please enter the correct username", , "HIS"
```

```
        txtPassword.SetFocus
        SendKeys "{Home}+{End}"
        End If
    End If
End Sub
```

frmSplash – Loaded after the Login Screen and before the Main Interface

```
Private Sub frmSplash_Click()
  Unload Me
End Sub

'Set action for time
Private Sub Timer1_Timer()
Unload Me
frmMain.Show
End Sub
```

frmMain – Option for users to choose desired menu

```
Private Declare Function DeleteMenu Lib "user32" (ByVal hMenu As Long, ByVal
nPosition As Long, ByVal wFlags As Long) As Long
Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal
bRevert As Long) As Long

Private Const MF_BYPOSITION = &H400&

'Function to remove the windows close button ('x')
Private Sub RemoveMenus()
  Dim hMenu As Long
  hMenu = GetSystemMenu(hWnd, False)
  DeleteMenu hMenu, 6, MF_BYPOSITION
End Sub

'During Main Form Loading
Private Sub MDIForm_Load()

  Width = Screen.Width - 2
  Height = Screen.Height - 2
  Me.Height = Height - 375
  Move (Screen.Width - Width) / 2, 2

  'to remove the windows close button ('x')
  RemoveMenus

  frmReminder.Show
End Sub

'Toolbar events
Private Sub tbrHIS_ButtonClick(ByVal Button As MSComctlLib.Button)
On Error Resume Next
  Select Case Button.Key
    Case "Patient"
      mnuPatient_Click
    Case "Billing"
      mnuBilling_Click
    Case "Pharmacy"
      mnuPharmacy_Click
    Case "Inventory"
      mnuInventory_Click
    Case "Search"
      mnuSearch_Click
    Case "Exit"
      mnuExit_Click
    Case "Reminder"
      mnuReminder_Click
  End Select
End Sub

'Display frmPatient window
Private Sub mnuPatient_Click()
  Unload frmPatient
  frmPatient.Show
End Sub

'Display frmBilling window
Private Sub mnuBilling_Click()
  Unload frmBilling
  frmBilling.Show
End Sub

'Display frmPharmacy window
Private Sub mnuPharmacy_Click()
  Unload frmPharmacy
  frmPharmacy.Show
End Sub

'Display frmInventory window
Private Sub mnuInventory_Click()
  Unload frmInventory
  frmInventory.Show
End Sub

'Display frmReminder window
Private Sub mnuReminder_Click()
  Unload frmReminder
  frmReminder.Show
End Sub

'Display  frmSearch window
Private Sub mnuSearch_Click()
  Unload frmSearch
  frmSearch.Show
End Sub

'Exit System
Private Sub mnuExit_Click()
  Dim out
  out = MsgBox("Are you sure you want to quit this program?", vbYesNo, "HIS")
  If out = vbYes Then
```

```
   End
   ElseIf out = vbNo Then
   Exit Sub
   End If
End Sub
```

frmPatient – Add, Delete, Edit and View patient information

```
Private Declare Function DeleteMenu Lib "user32" (ByVal hMenu As Long, ByVal
nPosition As Long, ByVal wFlags As Long) As Long
Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal
bRevert As Long) As Long
Private Const MF_BYPOSITION = &H400&

'Function to remove the windows close button ('x')
Private Sub RemoveMenus()
   Dim hMenu As Long
   hMenu = GetSystemMenu(hWnd, False)
   DeleteMenu hMenu, 6, MF_BYPOSITION
End Sub

Private Sub Form_Load()
cmdPayment.Visible = False
Me.Move 0, 0, 4230
frmControls
frmFill
RemoveMenus

'List items in the listview
With Adodc3
   .Refresh
   .RecordSource = "select * from Queue"
   .Refresh
End With

With Adodc3.Recordset
   If .RecordCount = 0 Then Exit Sub
   .MoveFirst
      While Not .EOF
      ListView1.ListItems.Add , , !Name
      .MoveNext
      Wend
End With
End Sub

'Add new patient records
Private Sub cmdNew_Click()
Me.Width = 10590
clearFields
frmEnabled
txtName.SetFocus
```

```
cmdSave.Visible = True
cmdCancel.Visible = True
cmdOK.Visible = False
cmdUpdate.Visible = False
cmdPrescription.Visible = False
cmdMedical.Visible = False
cmdEdit.Enabled = False
cmdInfo.Enabled = False
cmdDelete.Enabled = False
cmdInsert.Enabled = False
cmdRemove.Enabled = False
cmdExit.Enabled = False

txtID.Locked = False
txtID.Enabled = False
txtNRIC.Locked = False
txtName.Locked = False
txtAddress.Locked = False
txtCity.Locked = False
cboState.Locked = False
cboSex.Locked = False
cboRace.Locked = False
cboOccupation.Locked = False
txtPostcode.Locked = False
txtTelephone.Locked = False
txtAccompany.Locked = False
cboCompany.Locked = False

'Generate ID
On Error Resume Next
Adodc1.Refresh
Adodc1.Recordset.MoveLast
X = Adodc1.Recordset.Fields(0)
Adodc1.Recordset.AddNew
txtID.Text = X + 1

ListView1.Enabled = False
End Sub

'Edit patient records based on ID entered
Private Sub cmdEdit_Click()
Dim Name As String
Dim toedit As String

toedit = txtID.Text
If toedit = "" Then
```

```
    MsgBox "Please enter Patient ID to edit", vbCritical, "HIS"
    txtID.SetFocus
    Exit Sub
End If

With Adodc3
.Refresh
.RecordSource = "select * from Patient where ID = '" & txtID.Text & "'"
.Refresh
End With

With Adodc3.Recordset
    If .RecordCount = 0 Then
    MsgBox " There are no records of patient with ID = " & txtID.Text & ""
    Exit Sub
    End If
End With

ListView1.Enabled = False
cmdNew.Enabled = False
cmdInfo.Enabled = False
cmdDelete.Enabled = False
cmdInsert.Enabled = False
cmdRemove.Enabled = False
cmdExit.Enabled = False

Me.Width = 10590
Name = txtID.Text
showRecord Name, "patient"

txtNRIC.Locked = False
txtName.Locked = False
txtAddress.Locked = False
txtCity.Locked = False
cboState.Locked = False
cboSex.Locked = False
cboRace.Locked = False
cboOccupation.Locked = False
txtPostcode.Locked = False
txtTelephone.Locked = False
txtAccompany.Locked = False
cboCompany.Locked = False

cmdUpdate.Visible = True
cmdCancel.Visible = True
cmdSave.Visible = False
```

```
cmdOK.Visible = False
cmdPrescription.Visible = False
cmdMedical.Visible = False

frmEnabled
txtName.SetFocus
End Sub

'Delete patient from Patient, Payment and Queue
Private Sub cmdDelete_Click()
Dim todelete As String
todelete = txtID.Text

If todelete = "" Then
   MsgBox "Please enter Patient ID to delete", vbCritical, "HIS"
   txtID.SetFocus
   Exit Sub
End If

'Check database for Patient ID
With Adodc3
.Refresh
.RecordSource = "select * from Patient where ID = '" & txtID.Text & "'"
.Refresh
End With

With Adodc3.Recordset
   If .RecordCount = 0 Then
   MsgBox " There are no records of patient with ID = " & txtID.Text & ""
   Exit Sub
   End If
End With

cmdNew.Enabled = False
cmdInfo.Enabled = False
cmdEdit.Enabled = False
cmdInsert.Enabled = False
cmdRemove.Enabled = False
cmdExit.Enabled = False

'Delete confirmation message
On Error Resume Next
Dim del
del = MsgBox("Are you sure you want to delete this record?", vbYesNo)
If del = vbYes Then
```

```
  Me.Width = 4230
  cmdOK.Visible = False
  cmdPrescription.Visible = False
  cmdMedical.Visible = False
  cmdNew.Enabled = True
  cmdInfo.Enabled = True
  cmdEdit.Enabled = True
  cmdInsert.Enabled = True
  cmdRemove.Enabled = True
  cmdExit.Enabled = True

'Delete record from Table Patient
  With Adodc1.Recordset
   .MoveFirst
     While Not .EOF
       If todelete = !ID Then
       .Delete
       'Delete record from Table Queue
       cmdRemove_Click
       Exit Sub
       Else
       .MoveNext
       End If
     Wend
  End With

ElseIf del = vbNo Then
  MsgBox "Patient Information is not deleted", , "HIS"
  cmdNew.Enabled = True
  cmdInfo.Enabled = True
  cmdEdit.Enabled = True
  cmdInsert.Enabled = True
  cmdRemove.Enabled = True
  cmdExit.Enabled = True
  Exit Sub
End If
End Sub

'View record based on ID entered
Private Sub cmdInfo_Click()
Dim Name As String
Dim toview As String

toview = txtID.Text

If toview = "" Then
```

```
    MsgBox "Please enter Patient ID to view", vbCritical, "HIS"
    txtID.SetFocus
    Exit Sub
End If

'Check database for patient validity
With Adodc3
.Refresh
.RecordSource = "select * from Patient where ID = '" & txtID.Text & "'"
.Refresh
End With

With Adodc3.Recordset
    If .RecordCount = 0 Then
    MsgBox " There are no records of patient with ID = " & txtID.Text & ""
    Exit Sub
    End If
End With

ListView1.Enabled = False
cmdNew.Enabled = False
cmdDelete.Enabled = False
cmdEdit.Enabled = False
cmdInsert.Enabled = False
cmdRemove.Enabled = False
cmdExit.Enabled = False

Name = txtID.Text
showRecord Name, "patient"

Me.Width = 10590
cmdOK.Visible = True
cmdSave.Visible = False
cmdCancel.Visible = False
cmdUpdate.Visible = False
cmdPrescription.Visible = True
cmdMedical.Visible = True

frmDisabled
End Sub

'Insert existing patient into Queue based on ID
Private Sub cmdInsert_Click()
If txtID = "" Then
    MsgBox "Please enter Patient ID"
Else
```

```
 With Adodc1
  .Refresh
  .RecordSource = "select * from Patient where ID = '" & txtID.Text & "'"
  .Refresh
 End With

 If Adodc3.Recordset.RecordCount = 0 Then
  With Adodc1.Recordset
   If .RecordCount = 0 Then
   MsgBox " There are no records of patient with ID = " & txtID.Text & ""
   Exit Sub
   Else
   ListView1.ListItems.Add , , !Name
   ListView1.Refresh
   End If
  End With
 save

Else
'Check for duplicate Patient ID value
 With Adodc3.Recordset
 .MoveFirst
  Do While Not .EOF
   If .Fields("ID").Value = txtID.Text Then
   MsgBox "Patient already in queue", , "HIS"
   Exit Sub
   Else
   save
   .MoveNext
   End If
  Loop
 End With

 With Adodc1.Recordset
  If .RecordCount = 0 Then
  MsgBox " There are no records of patient with ID = " & txtID.Text & ""
  Exit Sub
  Else
   ListView1.ListItems.Add , , !Name
   ListView1.Refresh
  End If
 End With

End If
End If
Unload Me
frmPatient.Show
End Sub

'Remove patient from Queue
Private Sub cmdRemove_Click()
If Adodc3.Recordset.RecordCount = 0 Then
   MsgBox "Queue is empty"
Else
   On Error Resume Next
   Dim del
   del = MsgBox("Are you sure you want to remove this patient from queue?",
vbYesNo)
   If del = vbYes Then

   cmdOK.Visible = False

   Dim todelete As String
   todelete = ListView1.SelectedItem.Text

   With Adodc3.Recordset
   .MoveFirst
      While Not .EOF
         If todelete = !Name Then
         .Delete
         ListView1.ListItems.Remove ListView1.SelectedItem.Index
         Exit Sub
         Else
         .MoveNext
         End If
      Wend
   End With

   ElseIf del = vbNo Then
      MsgBox "Patient is still in queue", , "HIS"
      Exit Sub
   End If

Me.Width = 4230

End If
End Sub

'Exit Form
```

```
Private Sub cmdExit_Click()
Unload frmPrescription
Unload frmMedical
Unload Me
End Sub

Private Sub cmdOK_Click()
Me.Width = 4230
frmEnabled

ListView1.Enabled = True
cmdNew.Enabled = True
cmdDelete.Enabled = True
cmdEdit.Enabled = True
cmdInsert.Enabled = True
cmdRemove.Enabled = True
cmdExit.Enabled = True
cmdInfo.Enabled = True
cmdOK.Visible = False
End Sub

Private Sub cmdUpdate_Click()
'check for empty fields
If Len(txtID.Text) = 0 Then
   MsgBox "ID is not entered", vbExclamation, "HIS"
   txtID.SetFocus
   Exit Sub
ElseIf Len(txtName.Text) = 0 Then
   MsgBox "Patient name is not entered", vbExclamation, "HIS"
   txtName.SetFocus
   Exit Sub
ElseIf Len(txtNRIC.Text) = 0 Then
   MsgBox "NRIC is not entered", vbExclamation, "HIS"
   txtNRIC.SetFocus
   Exit Sub
ElseIf Len(txtBirthdate.Text) = 0 Then
   MsgBox "Birthdate is not entered", vbExclamation, "HIS"
   txtBirthdate.SetFocus
   Exit Sub
ElseIf Len(txtAddress.Text) = 0 Then
   MsgBox "Address is not entered", vbExclamation, "HIS"
   txtAddress.SetFocus
   Exit Sub
ElseIf Len(txtCity.Text) = 0 Then
   MsgBox "City is not entered", vbExclamation, "HIS"
   txtCity.SetFocus
   Exit Sub
ElseIf Len(txtTelephone.Text) = 0 Then
   MsgBox "Telephone is not entered", vbExclamation, "HIS"
   txtTelephone.SetFocus
   Exit Sub
ElseIf Len(txtAge.Text) = 0 Then
   MsgBox "Age is not entered", vbExclamation, "HIS"
   txtAge.SetFocus
   Exit Sub
ElseIf cboSex.ListIndex = 0 Then
   MsgBox "Sex is not entered", vbExclamation, "HIS"
   cboSex.SetFocus
   Exit Sub
ElseIf cboOccupation.ListIndex = 0 Then
   MsgBox "Occupation is not entered", vbExclamation, "HIS"
   cboOccupation.SetFocus
   Exit Sub
ElseIf cboState.ListIndex = 0 Then
   MsgBox "State is not entered", vbExclamation, "HIS"
   cboState.SetFocus
   Exit Sub
ElseIf cboRace.ListIndex = 0 Then
   MsgBox "Race is not entered", vbExclamation, "HIS"
   cboRace.SetFocus
   Exit Sub
End If

'check if postcode is numeric
If Not IsNumeric(txtPostcode.Text) Then
   MsgBox "Postcode must consists of integers", vbExclamation, "HIS"
   txtPostcode.SelStart = 0
   txtPostcode.SelLength = Len(txtPostcode.Text)
   txtPostcode.SetFocus
   Exit Sub
End If

'check if NRIC is numeric
If Not IsNumeric(txtNRIC.Text) Then
   MsgBox "NRIC must consists of integers", vbExclamation, "HIS"
   txtNRIC.SelStart = 0
   txtNRIC.SelLength = Len(txtNRIC.Text)
   txtNRIC.SetFocus
   Exit Sub
End If

'check if age is numeric
```

```
If Not IsNumeric(txtAge.Text) Then
   MsgBox "Age must consists of integers", vbExclamation, "HIS"
   txtAge.SelStart = 0
   txtAge.SelLength = Len(txtAge.Text)
   txtAge.SetFocus
   Exit Sub
End If

'check if telephone is numeric
If Not IsNumeric(txtTelephone.Text) Then
   MsgBox "Telephone must consists of integers", vbExclamation, "HIS"
   txtTelephone.SelStart = 0
   txtTelephone.SelLength = Len(txtTelephone.Text)
   txtTelephone.SetFocus
   Exit Sub
End If

ListView1.Enabled = True
cmdNew.Enabled = True
cmdDelete.Enabled = True
cmdInfo.Enabled = True
cmdInsert.Enabled = True
cmdRemove.Enabled = True
cmdExit.Enabled = True

With Adodc1.Recordset
   While Not .EOF
      If .RecordCount = 0 Then Exit Sub
      If !Name = ListView1.SelectedItem.Text Then
      !ID = txtID.Text
      !NRIC = txtNRIC.Text
      !Name = txtName.Text
      !Address = txtAddress.Text
      !City = txtCity.Text
      !State = cboState.ListIndex
      !Sex = cboSex.ListIndex
      !Race = cboRace.ListIndex
      !Occupation = cboOccupation.ListIndex
      !Postcode = txtPostcode.Text
      !Telephone = txtTelephone.Text
      !Company = cboCompany.ListIndex
      If txtAccompany = "" Then
      txtAccompany = "NIL"
      End If
      !Accompany = txtAccompany.Text
      .Update
```

```
        .MoveNext
        Else
        .MoveNext
        End If
    Wend
End With

Me.Width = 4230
cmdCancel.Visible = False
cmdUpdate.Visible = False
End Sub

Private Sub cmdCancel_Click()
txtID = ""
txtID.Locked = False
txtID.Enabled = True

frmControls
Me.Width = 4230

cmdCancel.Visible = False
cmdUpdate.Visible = False
cmdPrescription.Visible = True
cmdMedical.Visible = True

cmdNew.Enabled = True
cmdEdit.Enabled = True
cmdInfo.Enabled = True
cmdDelete.Enabled = True
cmdInsert.Enabled = True
cmdRemove.Enabled = True
cmdExit.Enabled = True

Adodc1.Recordset.CancelUpdate
ListView1.Enabled = True
End Sub

Private Sub cmdSave_Click()
'check for empty fields
If Len(txtID.Text) = 0 Then
    MsgBox "ID is not entered", vbExclamation, "HIS"
    txtID.SetFocus
    Exit Sub
ElseIf Len(txtName.Text) = 0 Then
    MsgBox "Patient name is not entered", vbExclamation, "HIS"
    txtName.SetFocus
    Exit Sub
ElseIf Len(txtNRIC.Text) = 0 Then
    MsgBox "NRIC is not entered", vbExclamation, "HIS"
    txtNRIC.SetFocus
    Exit Sub
ElseIf Len(txtBirthdate.Text) = 0 Then
    MsgBox "Birthdate is not entered", vbExclamation, "HIS"
    txtBirthdate.SetFocus
    Exit Sub
ElseIf Len(txtAddress.Text) = 0 Then
    MsgBox "Address is not entered", vbExclamation, "HIS"
    txtAddress.SetFocus
    Exit Sub
ElseIf Len(txtCity.Text) = 0 Then
    MsgBox "City is not entered", vbExclamation, "HIS"
    txtCity.SetFocus
    Exit Sub
ElseIf Len(txtTelephone.Text) = 0 Then
    MsgBox "Telephone is not entered", vbExclamation, "HIS"
    txtTelephone.SetFocus
    Exit Sub
ElseIf Len(txtAge.Text) = 0 Then
    MsgBox "Age is not entered", vbExclamation, "HIS"
    txtAge.SetFocus
    Exit Sub
ElseIf cboSex.ListIndex = 0 Then
    MsgBox "Sex is not entered", vbExclamation, "HIS"
    cboSex.SetFocus
    Exit Sub
ElseIf cboOccupation.ListIndex = 0 Then
    MsgBox "Occupation is not entered", vbExclamation, "HIS"
    cboOccupation.SetFocus
    Exit Sub
ElseIf cboState.ListIndex = 0 Then
    MsgBox "State is not entered", vbExclamation, "HIS"
    cboState.SetFocus
    Exit Sub
ElseIf cboRace.ListIndex = 0 Then
    MsgBox "Race is not entered", vbExclamation, "HIS"
    cboRace.SetFocus
    Exit Sub
End If

'check if postcode is numeric
If Not IsNumeric(txtPostcode.Text) Then
    MsgBox "Postcode must consists of integers", vbExclamation, "HIS"
    txtPostcode.SelStart = 0
    txtPostcode.SelLength = Len(txtPostcode.Text)
    txtPostcode.SetFocus
    Exit Sub
End If

'check if NRIC is numeric
If Not IsNumeric(txtNRIC.Text) Then
    MsgBox "NRIC must consists of integers", vbExclamation, "HIS"
    txtNRIC.SelStart = 0
    txtNRIC.SelLength = Len(txtNRIC.Text)
    txtNRIC.SetFocus
    Exit Sub
End If

'check if age is numeric
If Not IsNumeric(txtAge.Text) Then
    MsgBox "Age must consists of integers", vbExclamation, "HIS"
    txtAge.SelStart = 0
    txtAge.SelLength = Len(txtAge.Text)
    txtAge.SetFocus
    Exit Sub
End If

'check if telephone is numeric
If Not IsNumeric(txtTelephone.Text) Then
    MsgBox "Telephone must consists of integers", vbExclamation, "HIS"
    txtTelephone.SelStart = 0
    txtTelephone.SelLength = Len(txtTelephone.Text)
    txtTelephone.SetFocus
    Exit Sub
End If

'Save all fields
With Adodc1.Recordset
    .Fields("ID").Value = Left(txtID.Text, 20)
    .Fields("Name").Value = Left(txtName.Text, 30)
    .Fields("NRIC").Value = Left(txtNRIC.Text, 30)
    .Fields("Sex").Value = Left(cboSex.ListIndex, 20)
    .Fields("Race").Value = Left(cboRace.ListIndex, 20)
    .Fields("Birthdate").Value = Left(txtBirthdate.Text, 20)
    .Fields("Age").Value = Left(txtAge.Text, 20)
    .Fields("Occupation").Value = Left(cboOccupation.ListIndex, 20)
    .Fields("Address").Value = Left(txtAddress.Text, 20)
    .Fields("City").Value = Left(txtCity.Text, 20)
    .Fields("Postcode").Value = Left(txtPostcode.Text, 20)
```

```
  .Fields("State").Value = Left(cboState.ListIndex, 20)
  .Fields("Telephone").Value = Left(txtTelephone.Text, 20)
  !Company = cboCompany.ListIndex
  If txtAccompany = "" Then
  txtAccompany = "NIL"
  End If
  !Accompany = txtAccompany.Text
  .Update
  frmControls
  ListView1.ListItems.Add 1, , txtName.Text
End With

save

fraPelanggan.Refresh
Me.Width = 4230

ListView1.Enabled = True

cmdEdit.Enabled = True
cmdInfo.Enabled = True
cmdDelete.Enabled = True
cmdInsert.Enabled = True
cmdRemove.Enabled = True
cmdExit.Enabled = True
End Sub

Private Sub cmdPrescription_Click()
ListView1.Enabled = True
Unload frmPrescription
frmPrescription.Show
End Sub

Private Sub cmdMedical_Click()
ListView1.Enabled = True
Unload frmMedical
frmMedical.Show
End Sub

Private Sub cmdPayment_Click()
Unload frmPayment
frmPayment.Show
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
If KeyCode = 27 Then
```

```
  Unload Me
End If
End Sub

'Clear Form
Private Sub clearFields()
txtName.Text = ""
txtNRIC.Text = ""
txtBirthdate.Text = ""
txtAge.Text = ""
txtAddress.Text = ""
txtCity.Text = ""
txtPostcode.Text = ""
txtTelephone = ""
txtAccompany = ""

cboRace.ListIndex = 0
cboState.ListIndex = 0
cboSex.ListIndex = 0
cboOccupation.ListIndex = 0
cboCompany.ListIndex = 0
End Sub

Private Sub ListView1_Click()
Dim Name As String

cmdPrescription.Visible = True
cmdMedical.Visible = True

If ListView1.ListItems.Count = 0 Then Exit Sub
Name = ListView1.SelectedItem.Text
showRecord Name, "patient"

cmdSave.Visible = False
cmdCancel.Visible = False
cmdUpdate.Visible = False
cmdOK.Visible = True

With Adodc4
.Refresh
.RecordSource = "select * from prescription where PID='" & txtID.Text & "'"
.Refresh
End With

If Adodc4.Recordset.RecordCount = 0 Then
cmdPayment.Visible = False
```

```
Else
cmdPayment.Visible = True
End If
End Sub

Private Sub ListView1_DblClick()
ListView1_Click

Me.Width = 10590
cmdOK.Visible = True
cmdNew.Enabled = False
cmdInsert.Enabled = False
cmdRemove.Enabled = False
cmdInfo.Enabled = False
cmdInsert.Enabled = False
cmdRemove.Enabled = False
cmdExit.Enabled = False
End Sub

'Display data in selected ID fields
Public Sub showRecord(Name As String, loadby As String)
If loadby = "patient" Then

  With Adodc1.Recordset
    If .RecordCount = 0 Then Exit Sub

      .MoveFirst
      Do While Not .EOF

      If !Name = Name Or !ID = Name Or !NRIC = Name _
      Or !Address = Name Then
      GoTo endOfSearch
      Else
      .MoveNext
      End If
     Loop
    Exit Sub

endOfSearch:
txtID = !ID
txtNRIC = !NRIC
txtName = !Name
txtAddress = !Address
txtCity = !City
txtPostcode = !Postcode
cboState.ListIndex = !State
```

```
txtTelephone = !Telephone
cboRace.ListIndex = !Race
cboSex.ListIndex = !Sex
cboOccupation.ListIndex = !Occupation
cboCompany.ListIndex = !Company
txtAccompany = !Accompany
  End With
End If
End Sub

'Saving data to Queue table
Private Sub save()
With Adodc1.Recordset
  If .RecordCount = 0 Then Exit Sub
  txtID = !ID
  txtName = !Name
End With

With Adodc3
  .Refresh
  .RecordSource = "select * from Queue"
  .Refresh
End With

On Error Resume Next

With Adodc3.Recordset
  .AddNew
  .Fields("ID").Value = Left(txtID, 50)
  .Fields("Name").Value = Left(txtName, 30)
  .Update
End With
End Sub

'Lock fields
Private Sub frmControls()
txtNRIC.Locked = True
txtName.Locked = True
txtAddress.Locked = True
txtCity.Locked = True
cboState.Locked = True
cboSex.Locked = True
cboRace.Locked = True
cboOccupation.Locked = True
txtPostcode.Locked = True
txtTelephone.Locked = True
```

```
txtAccompany.Locked = True
cboCompany.Locked = True

cmdSave.Visible = False
cmdCancel.Visible = False
cmdUpdate.Visible = False
End Sub

Private Sub frmFill()
'State dropdown list'
cboState.AddItem ""
cboState.AddItem "Johore"
cboState.AddItem "Kelantan"
cboState.AddItem "Kedah"
cboState.AddItem "Kuala Lumpur"
cboState.AddItem "Malacca"
cboState.AddItem "N. Sembilan"
cboState.AddItem "Pahang"
cboState.AddItem "Penang"
cboState.AddItem "Perak"
cboState.AddItem "Perlis"
cboState.AddItem "Selangor"
cboState.AddItem "Sabah"
cboState.AddItem "Sarawak"
cboState.AddItem "Terengganu"
cboState.ListIndex = 0

'Race dropdown list
cboRace.AddItem ""
cboRace.AddItem "Chinese"
cboRace.AddItem "Malay"
cboRace.AddItem "Indian"
cboRace.AddItem "Sikh"
cboRace.ListIndex = 0

'Sex dropdown list
cboSex.AddItem ""
cboSex.AddItem "Female"
cboSex.AddItem "Male"
cboSex.ListIndex = 0

'Occupation dropdown list
cboOccupation.AddItem ""
cboOccupation.AddItem "Government"
cboOccupation.AddItem "NGO"
cboOccupation.AddItem "Student"
cboOccupation.ListIndex = 0

'Company dropdown list
cboCompany.AddItem ""
cboCompany.AddItem "ABC"
cboCompany.ListIndex = 0
End Sub

'Disabled all fields
Private Sub frmDisabled()
txtID.Enabled = False
txtNRIC.Enabled = False
txtName.Enabled = False
txtAddress.Enabled = False
txtCity.Enabled = False
cboState.Enabled = False
cboSex.Enabled = False
cboRace.Enabled = False
cboOccupation.Enabled = False
txtPostcode.Enabled = False
txtTelephone.Enabled = False
txtBirthdate.Enabled = False
txtAge.Enabled = False
txtAccompany.Enabled = False
cboCompany.Enabled = False
End Sub

'Enabled all fields
Private Sub frmEnabled()
txtID.Enabled = True
txtNRIC.Enabled = True
txtName.Enabled = True
txtAddress.Enabled = True
txtCity.Enabled = True
cboState.Enabled = True
cboSex.Enabled = True
cboRace.Enabled = True
cboOccupation.Enabled = True
txtPostcode.Enabled = True
txtTelephone.Enabled = True
txtBirthdate.Enabled = True
txtAge.Enabled = True
txtAccompany.Enabled = True
cboCompany.Enabled = True
End Sub
```

frmPrescription – Add, Edit, Delete and View medicine prescribed to patients

```
Private Declare Function DeleteMenu Lib "user32" (ByVal hMenu As Long, ByVal
nPosition As Long, ByVal wFlags As Long) As Long
Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal
bRevert As Long) As Long
Private Const MF_BYPOSITION = &H400&

'Function to remove the windows close button ('x')
Private Sub RemoveMenus()
   Dim hMenu As Long
   hMenu = GetSystemMenu(hWnd, False)
   DeleteMenu hMenu, 6, MF_BYPOSITION
End Sub

Private Sub Form_Load()
Me.Move 0, 0, 4230
frmControls
RemoveMenus
'Display item in listview
With Adodc3
   .Refresh
   .RecordSource = "select * from Prescription where PID = '" & frmPatient.txtID.Text
& "'"
   .Refresh
End With

With Adodc3.Recordset
   If .RecordCount = 0 Then Exit Sub
   .MoveFirst
      While Not .EOF
      ListView1.ListItems.Add , , !ID
      .MoveNext
      Wend
   End With
End Sub

Private Sub cmdAdd_Click()
If txtMID = "" Then
   MsgBox "Please enter medicine ID to prescribe"
   txtMID.SetFocus
Else
'Check for Medicine ID validity
   With Adodc2
      .Refresh
      .RecordSource = "select * from Medicine where Code = '" & txtMID.Text & "'"
      .Refresh
```

```
    End With

    With Adodc2.Recordset
        If .RecordCount = 0 Then
            MsgBox " There are no records of medicine with ID = " & txtMID.Text & ""
            Exit Sub
        Else
            txtMID = !Code
            txtMName = !Name
            txtDosage = !Dosage
            txtPrice = !Price
        End If
    End With

txtSupply.Text = ""
txtQuantity.Text = ""
txtSupply.SetFocus
txtSupply.Locked = False
txtQuantity.Locked = False

ListView1.Enabled = False

cmdEdit.Enabled = False
cmdDelete.Enabled = False
cmdExit.Enabled = False
cmdSave.Visible = True
cmdCancel.Enabled = True
cmdCancel.Visible = True
cmdSave.Enabled = True
cmdUpdate.Visible = False
cmdOK.Visible = False
cmdClear.Visible = False

txtStock.Visible = True
Stock.Visible = True

txtDate = Date

With Adodc1
  .Refresh
  .RecordSource = "select * from Patient where ID = '" & frmPatient.txtID.Text & "'"
  .Refresh
End With

On Error Resume Next
With Adodc3
```

```
   .Refresh
   .RecordSource = "select * from Prescription "
   .Refresh
End With

With Adodc1.Recordset
  If .RecordCount = 0 Then Exit Sub
  txtPID = !ID
  txtName = !Name
End With

'Generate ID
Dim X As Long
On Error Resume Next
Adodc3.Refresh
Adodc3.Recordset.MoveLast
X = Adodc3.Recordset.Fields(0)
Adodc3.Recordset.AddNew
txtID.Text = X + 1

Me.Width = 9840

End If
End Sub

'Edit presciption record
Private Sub cmdEdit_Click()
Dim toedit As String
toedit = ListView1.SelectedItem.Text

If toedit = "" Then
   MsgBox "Please choose a Medicine to edit", vbCritical, "HIS"
   Exit Sub
End If

cmdUpdate.Enabled = True

Dim Name As String
Name = ListView1.SelectedItem.Text
showRecord Name, "medicine"

txtSupply.Locked = False

cmdClear.Visible = True
cmdUpdate.Visible = True
cmdCancel.Visible = False
```

```
cmdSave.Visible = False

txtSupply.SetFocus
Me.Width = 9840
End Sub

'Delete selected records
Private Sub cmdDelete_Click()
On Error Resume Next
Dim del
del = MsgBox("Are you sure you want to delete this record?", vbYesNo)
If del = vbYes Then
cmdOK.Visible = False

Dim todelete As String
todelete = ListView1.SelectedItem.Text

With Adodc3.Recordset
.MoveFirst
    While Not .EOF
        If todelete = !ID Then
        .Delete
        ListView1.ListItems.Remove ListView1.SelectedItem.Index
        Me.Width = 4230
        Exit Sub
        Else
        .MoveNext
        End If
    Wend
End With

ElseIf del = vbNo Then
    MsgBox "Medicine Information is not deleted", , "HIS"
    Me.Width = 4230
    Exit Sub
End If
End Sub

'Exit form and show cmdPayment if prescription record >1
Private Sub cmdExit_Click()
If Adodc3.Recordset.RecordCount = 0 Then
    frmPatient.cmdPayment.Visible = False
Else
    frmPatient.cmdPayment.Visible = True
End If
```

```
Unload Me
End Sub

'Clear fields
Private Sub cmdClear_Click()
txtMID.Locked = False

ListView1.Enabled = True
cmdAdd.Enabled = True
cmdDelete.Enabled = True
cmdEdit.Enabled = True
cmdExit.Enabled = True

clearFields
Me.Width = 4230
txtMID.SetFocus
frmUnlocked

End Sub

'Cancel action
Private Sub cmdCancel_Click()
Me.Width = 4230
ListView1.Enabled = True
cmdCancel.Visible = False
clearFields

Adodc1.Recordset.CancelUpdate
Adodc2.Recordset.CancelUpdate
Adodc3.Recordset.CancelUpdate
End Sub

Private Sub cmdSave_Click()
'Check for empty fields
If Len(txtSupply.Text) = 0 Then
   MsgBox "Supply is not entered", vbExclamation, "HIS"
   txtSupply.SetFocus
   Exit Sub
ElseIf Len(txtQuantity.Text) = 0 Then
   MsgBox "Quantity is not entered", vbExclamation, "HIS"
   txtQuantity.SetFocus
   Exit Sub
End If
'Remove stock amount and add distributed amount
With Adodc2.Recordset
   txtMName = !Name
   txtPrice = !Price
   txtPrice.Text = txtPrice.Text * txtQuantity.Text
   !Stock = !Stock - txtQuantity.Text
   !Distributed = !Distributed + txtQuantity.Text
   .Update
End With

'Save fields
On Error Resume Next
With Adodc3.Recordset
   .AddNew
   .Fields("ID").Value = Left(txtID.Text, 50)
   .Fields("PID").Value = Left(frmPatient.txtID.Text, 50)
   .Fields("Name").Value = Left(txtName.Text, 30)
   .Fields("MID").Value = Left(txtMID.Text, 200)
   .Fields("MName").Value = Left(txtMName.Text, 200)
   .Fields("Dosage").Value = Left(txtDosage.Text, 20)
   .Fields("Supply").Value = Left(txtSupply.Text, 20)
   .Fields("Quantity").Value = Left(txtQuantity.Text, 20)
   .Fields("Price").Value = Left(txtPrice.Text, 20)
   .Fields("Date").Value = Left(txtDate.Text, 20)
   .Update
   ListView1.ListItems.Add 1, , txtID.Text
End With

cmdOK.Visible = True
cmdCancel.Enabled = False
cmdSave.Enabled = False
ListView1.Enabled = True
fraPrescription.Refresh
frmControls
End Sub

Private Sub cmdUpdate_Click()
'Check for empty fields
If Len(txtSupply.Text) = 0 Then
   MsgBox "Supply is not entered", vbExclamation, "HIS"
   txtSupply.SetFocus
   Exit Sub
ElseIf Len(txtQuantity.Text) = 0 Then
   MsgBox "Quantity is not entered", vbExclamation, "HIS"
   txtQuantity.SetFocus
   Exit Sub
End If

cmdUpdate.Enabled = False
```

```
With Adodc2.Recordset
   txtPrice = !Price
   txtPrice.Text = txtPrice.Text * txtQuantity.Text
End With

'Update fields
On Error Resume Next
With Adodc3.Recordset
   .Fields("Supply").Value = Left(txtSupply.Text, 20)
   .Fields("Quantity").Value = Left(txtQuantity.Text, 20)
   .Fields("Price").Value = Left(txtPrice.Text, 20)
   .Update
End With

ListView1.Enabled = True
fraPrescription.Refresh
frmControls
End Sub

Private Sub ListView1_Click()
frmControls
txtMID.Locked = True

Dim Name As String

If ListView1.ListItems.Count = 0 Then Exit Sub
Name = ListView1.SelectedItem.Text
showRecord Name, "medicine"

cmdOK.Visible = True
cmdSave.Visible = False
cmdCancel.Visible = False
cmdUpdate.Visible = False
cmdClear.Visible = False

txtStock.Visible = False
Stock.Visible = False
End Sub

Private Sub ListView1_DblClick()
frmControls
ListView1_Click
Me.Width = 9900
cmdAdd.Enabled = False
cmdExit.Enabled = False
```

```
End Sub

'Display prescription information
Public Sub showRecord(Name As String, loadby As String)
If loadby = "medicine" Then

  With Adodc3.Recordset
    If .RecordCount = 0 Then Exit Sub

       .MoveFirst
       Do While Not .EOF

       If !ID = Name Or !MName = Name Then
       GoTo endOfSearch
       Else
       .MoveNext
       End If
      Loop
    Exit Sub

endOfSearch:
txtID = !ID
frmPatient.txtID = !PID
txtMID = !Mid
txtMName = !MName
txtName = !Name
txtDosage = !Dosage
txtPrice = !Price
txtQuantity = !Quantity
txtSupply = !Supply
txtDate = !Date

  End With

End If
End Sub

Private Sub cmdOK_Click()
Me.Width = 4230
txtID = ""
ListView1.Enabled = True
cmdAdd.Enabled = True
cmdDelete.Enabled = True
cmdEdit.Enabled = True
cmdExit.Enabled = True
cmdOK.Visible = False
```

```
End Sub

'Lock fields
Private Sub frmControls()
txtID.Locked = True
txtMName.Locked = True
txtName.Locked = True
txtDosage.Locked = True
txtPrice.Locked = True
txtQuantity.Locked = True
txtSupply.Locked = True
txtDate.Locked = True
txtSupply.Locked = True
txtQuantity.Locked = True
txtStock.Locked = True
End Sub

'Unlock fields
Private Sub frmUnlocked()
txtSupply.Locked = False
txtQuantity.Locked = False
End Sub

'Clear form fields
Private Sub clearFields()
txtID.Text = ""
txtMID.Text = ""
txtName.Text = ""
txtMName.Text = ""
txtDosage.Text = ""
txtSupply.Text = ""
txtQuantity.Text = ""
txtStock.Text = ""
txtPrice.Text =""
End Sub
```

frmPayment – Add, Edit, Delete or View payment from patients

```
Dim X As Long
Private Declare Function DeleteMenu Lib "user32" (ByVal hMenu As Long, ByVal
nPosition As Long, ByVal wFlags As Long) As Long
Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal
bRevert As Long) As Long
Private Const MF_BYPOSITION = &H400&

'Function to remove the windows close button ('x')
Private Sub RemoveMenus()
   Dim hMenu As Long
   hMenu = GetSystemMenu(hWnd, False)
   DeleteMenu hMenu, 6, MF_BYPOSITION
End Sub

Private Sub Form_Load()
Me.Move 0, 0, 4230
frmControls
RemoveMenus
'Display items in listview
With Adodc2
   .Refresh
   .RecordSource = "select * from Payment where PID = '" & frmPatient.txtID.Text &
"'"
   .Refresh
End With

With Adodc2.Recordset
   If .RecordCount = 0 Then Exit Sub
   .MoveFirst
      While Not .EOF
      ListView1.ListItems.Add , , !ID
      .MoveNext
      Wend
   End With
End Sub

'Add payment record
Private Sub cmdAdd_Click()
txtPayer.Text = ""
txtPay.Text = ""
txtChange.Text = ""
txtPayer.SetFocus
txtPayer.Locked = False
txtPay.Locked = False
```

```
With Adodc1
  .Refresh
  .RecordSource = "select * from Patient where ID = '" & frmPatient.txtID.Text & "'"
  .Refresh
End With

'Check if patient company value is more than 0
On Error Resume Next
With Adodc1.Recordset
   If .RecordCount = 0 Then Exit Sub
   If !Company > 0 Then
     txtPay = 0
     txtPayer = "Company"
     txtPID = !ID
     txtName = !Name
     txtCompany = !Company
   Else
     txtPID = !ID
     txtName = !Name
     txtCompany = !Company
   End If
End With

ListView1.Enabled = False

cmdEdit.Enabled = False
cmdDelete.Enabled = False
cmdExit.Enabled = False
cmdSave.Visible = True
cmdCancel.Enabled = True
cmdCancel.Visible = True
cmdSave.Enabled = True
cmdUpdate.Visible = False
cmdOK.Visible = False
cmdClear.Visible = False

txtDate = Date

With Adodc2
   .Refresh
   .RecordSource = "select * from Payment"
   .Refresh
End With

With Adodc3
```

```
   .Refresh
   .RecordSource = "select * from Prescription where  PID= '" & frmPatient.txtID.Text
& "'"
   .Refresh
End With

'Sum all prescription price
On Error Resume Next
With Adodc3.Recordset
   If .RecordCount = 0 Then Exit Sub
   .MoveFirst
      Do While Not .EOF
         If !PID = frmPatient.txtID Then
         Dim sum As Long
         txtCharge = !Price
         sum = sum + txtCharge
         .MoveNext
         End If
      Loop
End With

If sum = 0 Then
txtCharge = txtCharge
Else
txtCharge = sum
End If

'Generate ID
Dim X As Long
On Error Resume Next
Adodc2.Refresh
Adodc2.Recordset.MoveLast
X = Adodc2.Recordset.Fields(0)
Adodc2.Recordset.AddNew
txtID.Text = X + 1
Me.Width = 9420
End Sub

'Edit payment record
Private Sub cmdEdit_Click()
Dim toedit As String
toedit = ListView1.SelectedItem.Text

If toedit = "" Then
  MsgBox "Please choose a record to edit", vbCritical, "HIS"
  Exit Sub
```

```
End If
Me.Width = 9420
frmUnlocked
cmdUpdate.Visible = True
cmdCancel.Visible = True
cmdSave.Visible = False
txtPayer.SetFocus
End Sub

'Delete selected payment record
Private Sub cmdDelete_Click()
On Error Resume Next
Dim del
del = MsgBox("Are you sure you want to delete this record?", vbYesNo)
If del = vbYes Then
cmdOK.Visible = False

Dim todelete As String
todelete = ListView1.SelectedItem.Text

With Adodc2.Recordset
.MoveFirst
   While Not .EOF
      If todelete = !ID Then
      .Delete
      ListView1.ListItems.Remove ListView1.SelectedItem.Index
      Exit Sub
      Else
      .MoveNext
      End If
   Wend
End With

ElseIf del = vbNo Then
MsgBox "Payment Information is not deleted", , "HIS"
Exit Sub
End If

Me.Width = 4230
End Sub

'Exit form
Private Sub cmdExit_Click()
Unload Me
End Sub
```

```
'Clear Form
Private Sub cmdClear_Click()
txtID.Locked = False

ListView1.Enabled = True
cmdAdd.Enabled = True
cmdDelete.Enabled = True
cmdEdit.Enabled = True
cmdExit.Enabled = True

clearFields
Me.Width = 4230
txtID.SetFocus
frmUnlocked
End Sub

'Cancel action
Private Sub cmdCancel_Click()
Me.Width = 4230
ListView1.Enabled = True
cmdCancel.Visible = False
clearFields

cmdAdd.Enabled = True
cmdEdit.Enabled = True
cmdDelete.Enabled = True
cmdExit.Enabled = True

Adodc1.Recordset.CancelUpdate
Adodc2.Recordset.CancelUpdate
End Sub

Private Sub cmdOK_Click()
Me.Width = 4230
txtID = ""
ListView1.Enabled = True
cmdAdd.Enabled = True
cmdDelete.Enabled = True
cmdEdit.Enabled = True
cmdExit.Enabled = True
cmdOK.Visible = False
End Sub

Private Sub cmdSave_Click()
```

```
'Check for empty fields
If Len(txtPayer.Text) = 0 Then
   MsgBox "Payer is not entered", vbExclamation, "HIS"
   txtPayer.SetFocus
   Exit Sub
ElseIf Len(txtPay.Text) = 0 Then
   MsgBox "Pay is not entered", vbExclamation, "HIS"
   txtPay.SetFocus
   Exit Sub
End If
'Save data in database
On Error Resume Next
With Adodc2.Recordset
   .AddNew
   .Fields("ID").Value = Left(txtID.Text, 50)
   .Fields("PID").Value = Left(frmPatient.txtID.Text, 50)
   .Fields("Name").Value = Left(txtName.Text, 30)
   .Fields("Payer").Value = Left(txtPayer.Text, 200)
   .Fields("Date").Value = Left(txtDate.Text, 200)
   .Fields("Charge").Value = Left(txtCharge.Text, 20)
   .Fields("Pay").Value = Left(txtPay.Text, 20)
   .Fields("Change").Value = Left(txtPay.Text - txtCharge.Text, 20)
   .Fields("Company").Value = Left(txtCompany.Text, 20)
   .Update
End With

cmdOK.Visible = True
cmdCancel.Enabled = False
cmdSave.Enabled = False
ListView1.Enabled = True
frmControls
ListView1.ListItems.Add 1, , txtID.Text
End Sub

Private Sub cmdUpdate_Click()
'Check for empty fields
If Len(txtPayer.Text) = 0 Then
   MsgBox "Payer is not entered", vbExclamation, "HIS"
   txtPayer.SetFocus
   Exit Sub
ElseIf Len(txtPay.Text) = 0 Then
   MsgBox "Pay is not entered", vbExclamation, "HIS"
   txtPay.SetFocus
   Exit Sub
End If
'Update data in database
```

```
On Error Resume Next
With Adodc2.Recordset
   .Fields("Payer").Value = Left(txtPayer.Text, 20)
   .Fields("Pay").Value = Left(txtPay.Text, 20)
   .Fields("Charge").Value = Left(txtCharge.Text, 20)
   .Fields("Change").Value = Left(txtPay.Text - txtCharge.Text, 20)
   .Update
End With

ListView1.Enabled = True
fraPrescription.Refresh
frmControls
cmdUpdate.Enabled = False
End Sub

Private Sub ListView1_Click()
frmControls
txtID.Locked = True

Dim Name As String

If ListView1.ListItems.Count = 0 Then Exit Sub
Name = ListView1.SelectedItem.Text
showRecord Name, "medicine"

cmdSave.Visible = False
cmdCancel.Visible = False
cmdUpdate.Visible = False
cmdClear.Visible = False
End Sub

Private Sub ListView1_DblClick()
frmControls
ListView1_Click
Me.Width = 9420
cmdAdd.Enabled = False
cmdExit.Enabled = False
cmdOK.Visible = True
End Sub

'Display selected data information
Public Sub showRecord(Name As String, loadby As String)
If loadby = "medicine" Then

  With Adodc2.Recordset
     If .RecordCount = 0 Then Exit Sub
```

```
         .MoveFirst
         Do While Not .EOF

         If !ID = Name Or !Payer = Name Then
         GoTo endOfSearch
         Else
         .MoveNext
         End If
        Loop
      Exit Sub

endOfSearch:
txtID = !ID
txtPID = !PID
txtName = !Name
txtPayer = !Payer
txtDate = !Date
txtCharge = !Charge
txtPay = !Pay
txtChange = !Change
txtCompany = !Company
   End With

End If
End Sub

'Lock fields
Private Sub frmControls()
txtID.Locked = True
txtPID.Locked = True
txtName.Locked = True
txtPayer.Locked = True
txtDate.Locked = True
txtCharge.Locked = True
txtPay.Locked = True
txtChange.Locked = True
End Sub

'Unlock fields
Private Sub frmUnlocked()
txtPayer.Locked = False
txtPay.Locked = False
txtCharge.Locked = False
End Sub
```

```
'Clear fields function
Private Sub clearFields()
txtID.Text = ""
txtPID.Text = ""
txtName.Text = ""
txtPayer.Text = ""
txtDate.Text = ""
txtCharge.Text = ""
txtPay.Text = ""
txtChange.Text = ""
End Sub
```

frmMedical - Add, Delete, Edit and View patient medical information

```
Private Declare Function DeleteMenu Lib "user32" (ByVal hMenu As Long, ByVal
nPosition As Long, ByVal wFlags As Long) As Long
Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal
bRevert As Long) As Long

Private Const MF_BYPOSITION = &H400&

'Function to remove the windows close button ('x')
Private Sub RemoveMenus()
  Dim hMenu As Long
  hMenu = GetSystemMenu(hWnd, False)
  DeleteMenu hMenu, 6, MF_BYPOSITION
End Sub

'Display item in the list view
Private Sub Form_Load()
Me.Move 0, 0, 4230
frmControls
RemoveMenus

With Adodc2
   .Refresh
   .RecordSource = "select * from Medical where PID = '" & frmPatient.txtID.Text &
"'"
   .Refresh
End With

With Adodc2.Recordset
   If .RecordCount = 0 Then Exit Sub
   .MoveFirst
      While Not .EOF
      ListView1.ListItems.Add , , !ID
      .MoveNext
      Wend
End With
End Sub

'Add new medical records
Private Sub cmdAdd_Click()
Me.Width = 9420

txtDoctor.Text = ""
txtRemarks.Text = ""
txtDiagnosis.Text = ""
```

```
txtDoctor.SetFocus
txtDoctor.Locked = False
txtRemarks.Locked = False
txtDiagnosis.Locked = False

ListView1.Enabled = False

cmdEdit.Enabled = False
cmdDelete.Enabled = False
cmdExit.Enabled = False
cmdSave.Visible = True
cmdCancel.Enabled = True
cmdCancel.Visible = True
cmdSave.Enabled = True
cmdUpdate.Visible = False
cmdOK.Visible = False
cmdClear.Visible = False

txtDate = Date
txtTime = Time

With Adodc1
   .Refresh
   .RecordSource = "select * from Patient where ID = '" & frmPatient.txtID.Text & "'"
   .Refresh
End With

With Adodc2
   .Refresh
   .RecordSource = "select * from Medical "
   .Refresh
End With

On Error Resume Next
With Adodc1.Recordset
   If .RecordCount = 0 Then Exit Sub
   txtPID = !ID
   txtName = !Name
End With

Dim X As Long
On Error Resume Next
Adodc2.Refresh
Adodc2.Recordset.MoveLast
X = Adodc2.Recordset.Fields(0)
Adodc2.Recordset.AddNew
```

```
txtID.Text = X + 1
Me.Width = 9420
End Sub

'Edit selected medical records
Private Sub cmdEdit_Click()
Dim toedit As String
toedit = ListView1.SelectedItem.Text

If toedit = "" Then
  MsgBox "Please choose a Record to edit", vbCritical, "HIS"
  Exit Sub
End If

frmUnlocked
cmdClear.Visible = True
cmdUpdate.Visible = True
cmdCancel.Visible = False
cmdSave.Visible = False
txtDoctor.SetFocus
Me.Width = 9420
End Sub

'Delete selected medical records
Private Sub cmdDelete_Click()
On Error Resume Next
Dim del
del = MsgBox("Are you sure you want to delete this record?", vbYesNo)
If del = vbYes Then
cmdOK.Visible = False

Dim todelete As String
todelete = ListView1.SelectedItem.Text

With Adodc2.Recordset
.MoveFirst
  While Not .EOF
    If todelete = !ID Then
    .Delete
    ListView1.ListItems.Remove ListView1.SelectedItem.Index
    Exit Sub
    Else
    .MoveNext
    End If
  Wend
End With
```

```
ElseIf del = vbNo Then
MsgBox "Medicine Information is not deleted", , "HIS"
Exit Sub
End If

Me.Width = 4230
End Sub

Private Sub cmdExit_Click()
Unload Me
End Sub

Private Sub cmdOK_Click()
Me.Width = 4230
txtID = ""
ListView1.Enabled = True
cmdAdd.Enabled = True
cmdDelete.Enabled = True
cmdEdit.Enabled = True
cmdExit.Enabled = True
cmdOK.Visible = False
End Sub

Private Sub cmdClear_Click()
txtID.Locked = False

ListView1.Enabled = True
cmdAdd.Enabled = True
cmdDelete.Enabled = True
cmdEdit.Enabled = True
cmdExit.Enabled = True

clearFields
Me.Width = 4230
txtID.SetFocus
frmUnlocked
End Sub

Private Sub cmdCancel_Click()
Me.Width = 4230
ListView1.Enabled = True
cmdCancel.Visible = False
clearFields

Adodc1.Recordset.CancelUpdate
```

```
Adodc2.Recordset.CancelUpdate
End Sub

'Save record and check for empty space
Private Sub cmdSave_Click()
If Len(txtDiagnosis.Text) = 0 Then
   MsgBox "Diagnosis is not entered", vbExclamation, "HIS"
   txtDiagnosis.SetFocus
   Exit Sub
ElseIf Len(txtRemarks.Text) = 0 Then
   MsgBox "Remarks is not entered", vbExclamation, "HIS"
   txtRemarks.SetFocus
   Exit Sub
ElseIf Len(txtDoctor.Text) = 0 Then
   MsgBox "Doctor is not entered", vbExclamation, "HIS"
   txtDoctor.SetFocus
   Exit Sub
End If

On Error Resume Next

With Adodc2.Recordset
   .AddNew
   .Fields("ID").Value = Left(txtID.Text, 50)
   .Fields("PID").Value = Left(frmPatient.txtID.Text, 50)
   .Fields("Name").Value = Left(txtName.Text, 30)
   .Fields("Doctor").Value = Left(txtDoctor.Text, 200)
   .Fields("Date").Value = Left(txtDate.Text, 200)
   .Fields("Time").Value = Left(txtTime.Text, 20)
   .Fields("Diagnosis").Value = Left(txtDiagnosis.Text, 20)
   .Fields("Remarks").Value = Left(txtRemarks.Text, 20)
   .Update
   ListView1.ListItems.Add 1, , txtID.Text
End With

cmdOK.Visible = True
cmdCancel.Enabled = False
cmdSave.Enabled = False
ListView1.Enabled = True
fraPrescription.Refresh
frmControls
End Sub

'Check for empty space and update records
Private Sub cmdUpdate_Click()
If Len(txtDiagnosis.Text) = 0 Then
```

```
   MsgBox "Diagnosis is not entered", vbExclamation, "HIS"
   txtDiagnosis.SetFocus
   Exit Sub
ElseIf Len(txtRemarks.Text) = 0 Then
   MsgBox "Remarks is not entered", vbExclamation, "HIS"
   txtRemarks.SetFocus
   Exit Sub
ElseIf Len(txtDoctor.Text) = 0 Then
   MsgBox "Doctor is not entered", vbExclamation, "HIS"
   txtDoctor.SetFocus
   Exit Sub
End If

On Error Resume Next

With Adodc2.Recordset
   .Fields("Doctor").Value = Left(txtDoctor.Text, 20)
   .Fields("Remarks").Value = Left(txtRemarks.Text, 20)
   .Fields("Diagnosis").Value = Left(txtDiagnosis.Text, 20)
   .Update
   End With

ListView1.Enabled = True
fraPrescription.Refresh
frmControls
End Sub

Private Sub ListView1_Click()

frmControls
txtID.Locked = True

Dim Name As String

If ListView1.ListItems.Count = 0 Then Exit Sub
Name = ListView1.SelectedItem.Text
showRecord Name, "medicine"

cmdSave.Visible = False
cmdCancel.Visible = False
cmdUpdate.Visible = False
cmdClear.Visible = True
End Sub

Private Sub ListView1_DblClick()
frmControls
```

```
ListView1_Click
Me.Width = 9420
cmdAdd.Enabled = False
cmdExit.Enabled = False
End Sub

Public Sub showRecord(Name As String, loadby As String)
If loadby = "medicine" Then

   With Adodc2.Recordset
     If .RecordCount = 0 Then Exit Sub
     .MoveFirst
       Do While Not .EOF
         If !ID = Name Or !Doctor = Name Then
         GoTo endOfSearch
         Else
         .MoveNext
         End If
       Loop
     Exit Sub

endOfSearch:
txtID = !ID
txtPID = !PID
txtName = !Name
txtDoctor = !Doctor
txtDate = !Date
txtTime = !Time
txtDiagnosis = !Diagnosis
txtRemarks = !Remarks

   End With
End If
End Sub

Private Sub frmControls()
txtID.Locked = True
txtPID.Locked = True
txtName.Locked = True
txtDoctor.Locked = True
txtDate.Locked = True
txtDiagnosis.Locked = True
txtTime.Locked = True
txtRemarks.Locked = True

cmdSave.Visible = False
```

```
cmdCancel.Visible = False
cmdUpdate.Visible = False
End Sub

Private Sub frmUnlocked()
txtDoctor.Locked = False
txtDiagnosis.Locked = False
txtRemarks.Locked = False
End Sub

Private Sub clearFields()
txtID.Text = ""
txtPID.Text = ""
txtName.Text = ""
txtDoctor.Text = ""
txtDate.Text = ""
txtTime.Text = ""
txtDiagnosis.Text = ""
txtRemarks.Text = ""
End Sub
```

frmBilling - Add, Delete, Edit and View company payment information

```
Private Declare Function DeleteMenu Lib "user32" (ByVal hMenu As Long, ByVal
nPosition As Long, ByVal wFlags As Long) As Long
Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal
bRevert As Long) As Long

Private Const MF_BYPOSITION = &H400&

'Function to remove the windows close button ('x')
Private Sub RemoveMenus()
   Dim hMenu As Long

   hMenu = GetSystemMenu(hWnd, False)
   DeleteMenu hMenu, 6, MF_BYPOSITION
End Sub


Private Sub Form_Load()
cmdEdit.Enabled = False
cmdDelete.Enabled = False
Me.Move 0, 0, 4230
frmControls
RemoveMenus
End Sub

'Display billing records in the listview
Private Sub cmdInfo_Click()
ListView1.ListItems.Clear

If txtCID = "" Then
   MsgBox "Please enter Company ID to view", vbCritical, "HIS"
   txtCID.SetFocus
   Exit Sub
End If

With Adodc3
   .Refresh
   .RecordSource = "select * from Billing where CID = '" & txtCID.Text & "'"
   .Refresh
End With

With Adodc3.Recordset
   If .RecordCount = 0 Then
   MsgBox " There are no billing records available"
   Exit Sub
   Else
```

```
  .MoveFirst
    While Not .EOF
    ListView1.ListItems.Add , , !ID
    .MoveNext
  Wend
End If
End With

DataGrid1.clearFields

With Adodc2
  .Refresh
  .RecordSource = "select [PID],[Name],[Charge] from Payment where Company = '"
& txtCID.Text & "'"
  .Refresh
End With

ListView1.Enabled = True
cmdOK.Visible = True
cmdSave.Visible = False
cmdCancel.Visible = False
cmdUpdate.Visible = False
cmdEdit.Enabled = True
cmdDelete.Enabled = True
End Sub

'Add billing records
Private Sub cmdAdd_Click()
If txtCID = "" Then
  MsgBox "Please enter Company ID to view", vbCritical, "HIS"
  txtCID.SetFocus
  Exit Sub
End If

With Adodc2
  .Refresh
  .RecordSource = "select * from Payment where  Company = '" & txtCID.Text & "'"
  .Refresh
End With

On Error Resume Next
With Adodc2.Recordset
  If .RecordCount = 0 Then
    MsgBox "There are no billing record available"
    Exit Sub
  Else
```

```
         .MoveFirst
         Do While Not .EOF
            If !CID = txtCID.Text Then
            txtCharge = !Charge
            Dim sum As Long
            sum = sum + txtCharge
            .MoveNext
            End If
         Loop
      End If
   End With

   txtCharge = sum

   If sum = 0 Then
      MsgBox "Current charges is 0"
   Else
      txtPayer.Text = ""
      txtPay.Text = ""
      txtChange.Text = ""
      txtPayer.SetFocus
      txtPayer.Locked = False
      txtPay.Locked = False
      txtCharge.Locked = False

      ListView1.Enabled = False

      cmdEdit.Enabled = False
      cmdDelete.Enabled = False
      cmdExit.Enabled = False
      cmdSave.Visible = True
      cmdCancel.Enabled = True
      cmdCancel.Visible = True
      cmdSave.Enabled = True
      cmdUpdate.Visible = False
      cmdOK.Visible = False
      cmdClear.Visible = False

      txtDate = Date

      DataGrid1.clearFields

      With Adodc2
         .Refresh
         .RecordSource = "select [PID],[Name],[Charge] from Payment where Company =
'" & txtCID.Text & "'"
```

```
    .Refresh
  End With
  With Adodc3
    .Refresh
    .RecordSource = "select * from Billing"
    .Refresh
  End With

  Dim X As Long
  On Error Resume Next
    Adodc3.Refresh
    Adodc3.Recordset.MoveLast
    X = Adodc3.Recordset.Fields(0)
    Adodc3.Recordset.AddNew
    txtID.Text = X + 1

    Me.Width = 9420

End If
End Sub

'Edit Billing Records
Private Sub cmdEdit_Click()
Dim toedit As String
toedit = ListView1.SelectedItem.Text

If toedit = "" Then
  MsgBox "Please choose a Record to edit", vbCritical, "HIS"
  Exit Sub
End If

Me.Width = 9420

frmUnlocked

cmdUpdate.Visible = True
cmdCancel.Visible = True
cmdSave.Visible = False

txtPayer.SetFocus
End Sub

'Delete billing records
Private Sub cmdDelete_Click()
On Error Resume Next
```

```
'Delete confirmation message
Dim del
del = MsgBox("Are you sure you want to delete this record?", vbYesNo)
If del = vbYes Then
cmdOK.Visible = False

Dim todelete As String
todelete = ListView1.SelectedItem.Text

With Adodc2.Recordset
.MoveFirst
  While Not .EOF
      If todelete = !ID Then
      .Delete
      ListView1.ListItems.Remove ListView1.SelectedItem.Index
      Exit Sub
      Else
      .MoveNext
      End If
  Wend
End With
''Not deleted confirmation message
ElseIf del = vbNo Then
  MsgBox "Payment Information is not deleted", , "HIS"
Exit Sub
End If

Me.Width = 4230
End Sub

'Exit Form
Private Sub cmdExit_Click()
Unload Me
End Sub

Private Sub cmdOK_Click()
Me.Width = 4230
txtID = ""
ListView1.Enabled = True
cmdAdd.Enabled = True
cmdDelete.Enabled = True
cmdEdit.Enabled = True
cmdExit.Enabled = True
cmdOK.Visible = False
End Sub
```

```
'Clear Form
Private Sub cmdClear_Click()
txtID.Locked = False

ListView1.Enabled = True
cmdAdd.Enabled = True
cmdDelete.Enabled = True
cmdEdit.Enabled = True
cmdExit.Enabled = True

clearFields
Me.Width = 4230
txtCID.SetFocus
frmUnlocked
End Sub

'Cancel process
Private Sub cmdCancel_Click()
cmdAdd.Enabled = True
cmdEdit.Enabled = True
cmdDelete.Enabled = True
cmdExit.Enabled = True

clearFields
Me.Width = 4230
ListView1.Enabled = True
cmdCancel.Visible = False
Adodc2.Recordset.CancelUpdate
End Sub

Private Sub cmdSave_Click()
'Check fo empty fields
If Len(txtPayer.Text) = 0 Then
   MsgBox "Payer is not entered", vbExclamation, "HIS"
   txtPayer.SetFocus
   Exit Sub
ElseIf Len(txtPay.Text) = 0 Then
   MsgBox "Pay is not entered", vbExclamation, "HIS"
   txtPay.SetFocus
   Exit Sub
End If
'Save data in the database
On Error Resume Next
With Adodc3.Recordset
  .AddNew
  .Fields("ID").Value = Left(txtID.Text, 50)
```

```
    .Fields("CID").Value = Left(txtCID.Text, 50)
    .Fields("Payer").Value = Left(txtPayer.Text, 200)
    .Fields("Date").Value = Left(txtDate.Text, 200)
    .Fields("Charge").Value = Left(txtCharge.Text, 20)
    .Fields("Pay").Value = Left(txtPay.Text, 20)
    .Fields("Change").Value = Left(txtPay.Text - txtCharge.Text, 20)
    .Update
End With

With Adodc2
    .Refresh
    .RecordSource = "select * from Payment where Company = '" & txtCID.Text & "'"
    .Refresh
End With

On Error Resume Next
With Adodc2.Recordset
   If .RecordCount = 0 Then Exit Sub
   .MoveFirst
      Do While Not .EOF
         If !Company = txtCID.Text Then
         !Charge = "0"
         .MoveNext
         End If
      Loop
End With

frmControls
cmdOK.Visible = True
cmdCancel.Enabled = False
cmdSave.Enabled = False
ListView1.Enabled = True
ListView1.ListItems.Add 1, , txtID.Text
End Sub

Private Sub cmdUpdate_Click()
'Check fo empty fields
If Len(txtPayer.Text) = 0 Then
   MsgBox "Payer is not entered", vbExclamation, "HIS"
   txtPayer.SetFocus
   Exit Sub
ElseIf Len(txtPay.Text) = 0 Then
   MsgBox "Pay is not entered", vbExclamation, "HIS"
   txtPay.SetFocus
   Exit Sub
End If
```

```
'Update fields
On Error Resume Next
With Adodc3.Recordset
   .Fields("Payer").Value = Left(txtPayer.Text, 20)
   .Fields("Pay").Value = Left(txtPay.Text, 20)
   .Fields("Charge").Value = Left(txtCharge.Text, 20)
   .Fields("Change").Value = Left(txtPay.Text - txtCharge.Text, 20)
   .Update
End With

frmControls
fraBilling.Refresh
ListView1.Enabled = True
cmdUpdate.Enabled = False
End Sub

Private Sub ListView1_Click()
frmControls
txtID.Locked = True

Dim Name As String

If ListView1.ListItems.Count = 0 Then Exit Sub
Name = ListView1.SelectedItem.Text
showRecord Name, "billing"

cmdSave.Visible = False
cmdCancel.Visible = False
cmdUpdate.Visible = False
cmdClear.Visible = False
End Sub

Private Sub ListView1_DblClick()
frmControls
ListView1_Click
Me.Width = 9420
cmdAdd.Enabled = False
cmdExit.Enabled = False
cmdOK.Visible = True
End Sub

'Show record when listview is clicked
Public Sub showRecord(Name As String, loadby As String)
If loadby = "billing" Then
```

```
   With Adodc3.Recordset
     If .RecordCount = 0 Then Exit Sub

       .MoveFirst
       Do While Not .EOF

       If !CID = Name Or !Payer = Name Then
       GoTo endOfSearch
       Else
       .MoveNext
       End If
      Loop
     Exit Sub

endOfSearch:
txtID = !ID
txtCID = !CID
txtPayer = !Payer
txtDate = !Date
txtCharge = !Charge
txtPay = !Pay
txtChange = !Change

   End With
End If
End Sub

'Lock important fields
Private Sub frmControls()
txtID.Locked = True
txtPayer.Locked = True
txtDate.Locked = True
txtCharge.Locked = True
txtPay.Locked = True
txtChange.Locked = True
End Sub

'Unlock fields for bill payment purposes
Private Sub frmUnlocked()
txtPayer.Locked = False
txtPay.Locked = False
txtCharge.Locked = False
End Sub

'Function to clear text boxes
Private Sub clearFields()
```

```
txtID.Text = ""
txtCID.Text = ""
txtPayer.Text = ""
txtDate.Text = ""
txtCharge.Text = ""
txtPay.Text = ""
txtChange.Text = ""
End Sub
```

frmPharmacy – Add, Edit, Delete or view information regarding medicine

```
Private Declare Function DeleteMenu Lib "user32" (ByVal hMenu As Long, ByVal
nPosition As Long, ByVal wFlags As Long) As Long
Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal
bRevert As Long) As Long
Private Const MF_BYPOSITION = &H400&

'Function to remove the windows close button ('x')
Private Sub RemoveMenus()
  Dim hMenu As Long
  hMenu = GetSystemMenu(hWnd, False)
  DeleteMenu hMenu, 6, MF_BYPOSITION
End Sub

Private Sub Form_Load()
Me.Move 0, 0, 5400
frmControls
frmFill
Adodc1.Visible = False
RemoveMenus
'List all item in database in listview
With Adodc1.Recordset
  If .RecordCount = 0 Then Exit Sub
  .MoveFirst
    While Not .EOF
    ListView1.ListItems.Add , , !Name
    .MoveNext
    Wend
  End With
End Sub

'Add new medicine
Private Sub cmdNew_Click()
Me.Width = 9900
clearFields
frmEnabled
txtName.SetFocus

txtDistributed = 0

cmdSave.Visible = True
cmdCancel.Visible = True
cmdOK.Visible = False

cmdUpdate.Visible = False
```

```
cmdEdit.Enabled = False
cmdInfo.Enabled = False
cmdDelete.Enabled = False
cmdExit.Enabled = False

txtCode.Enabled = False
txtName.Locked = False
txtDosage.Locked = False
txtExpiry.Locked = False
cboType.Locked = False
txtPrice.Locked = False
txtStock.Locked = False

'Generate ID
On Error Resume Next
Adodc1.Refresh
Adodc1.Recordset.MoveLast
X = Adodc1.Recordset.Fields(0)
Adodc1.Recordset.AddNew
txtCode.Text = X + 1
End Sub

'Edit medicine information
Private Sub cmdEdit_Click()
Dim toedit As String
toedit = ListView1.SelectedItem.Text
If toedit = "" Then
MsgBox "Please choose a Medicine to edit", vbCritical, "HIS"
Exit Sub
End If

ListView1.Enabled = False
cmdNew.Enabled = False
cmdInfo.Enabled = False
cmdDelete.Enabled = False
cmdExit.Enabled = False

Me.Width = 9900

txtCode.Locked = False
txtName.Locked = False
txtDosage.Locked = False
txtExpiry.Locked = False
cboType.Locked = False
txtPrice.Locked = False
```

```
cmdUpdate.Visible = True
cmdCancel.Visible = True
cmdSave.Visible = False
cmdOK.Visible = False

frmEnabled
txtName.SetFocus
End Sub

'Delete selected medicine record
Private Sub cmdDelete_Click()
Dim todelete As String
todelete = ListView1.SelectedItem.Text

If todelete = "" Then
  MsgBox "Please Medicine to delete", vbCritical, "HIS"
  Exit Sub
End If

cmdNew.Enabled = False
cmdInfo.Enabled = False
cmdEdit.Enabled = False
cmdExit.Enabled = False

On Error Resume Next

Dim del
del = MsgBox("Are you sure you want to delete this record?", vbYesNo)
If del = vbYes Then
cmdOK.Visible = False

With Adodc1.Recordset
.MoveFirst
  While Not .EOF
    If todelete = !Name Then
    .Delete
    ListView1.ListItems.Remove ListView1.SelectedItem.Index
    Exit Sub
    Else
    .MoveNext
    End If
  Wend
End With

ElseIf del = vbNo Then
MsgBox "Medicine Information is not deleted", , "HIS"
```

```
    cmdNew.Enabled = True
    cmdInfo.Enabled = True
    cmdEdit.Enabled = True
    cmdExit.Enabled = True
Exit Sub
End If

Me.Width = 5400
End Sub

'View medicine information
Private Sub cmdInfo_Click()
ListView1.Enabled = False
cmdNew.Enabled = False
cmdDelete.Enabled = False
cmdEdit.Enabled = False
cmdExit.Enabled = False

Me.Width = 9900

cmdOK.Visible = True
cmdSave.Visible = False
cmdCancel.Visible = False
cmdUpdate.Visible = False
frmDisabled
End Sub

'Exit form
Private Sub cmdExit_Click()
Unload Me
End Sub

Private Sub cmdOK_Click()
ListView1.Enabled = True
cmdNew.Enabled = True
cmdDelete.Enabled = True
cmdEdit.Enabled = True
cmdExit.Enabled = True
cmdInfo.Enabled = True
cmdOK.Visible = False

Me.Width = 5400
cmdOK.Visible = False
End Sub

'Update record in database
```

```
Private Sub cmdUpdate_Click()
ListView1.Enabled = True
cmdNew.Enabled = True
cmdDelete.Enabled = True
cmdInfo.Enabled = True
cmdExit.Enabled = True

With Adodc1.Recordset
  While Not .EOF
    If .RecordCount = 0 Then Exit Sub
    If !Name = ListView1.SelectedItem.Text Then
    !Code = txtCode.Text
    !Name = txtName.Text
    !Dosage = txtDosage.Text
    !Expiry = txtExpiry.Text
    !Type = cboType.ListIndex
    !Price = txtPrice.Text
    !Stock = txtStock.Text
    !Distributed = txtDistributed.Text
    .Update
    .MoveNext
  Else
    .MoveNext
  End If
  Wend
End With

Me.Width = 5400
cmdCancel.Visible = False
cmdUpdate.Visible = False
End Sub

'Cancel action
Private Sub cmdCancel_Click()
frmControls
Me.Width = 5400

cmdCancel.Visible = False
cmdUpdate.Visible = False

cmdNew.Enabled = True
cmdEdit.Enabled = True
cmdInfo.Enabled = True
cmdDelete.Enabled = True
cmdExit.Enabled = True
```

```
ListView1.Enabled = True

Adodc1.Recordset.CancelUpdate
Adodc1.Recordset.MoveFirst
End Sub

Private Sub cmdSave_Click()
'Check for empty fields
If Len(txtCode.Text) = 0 Then
   MsgBox "Code is not entered", vbExclamation, "HIS"
   txtCode.SetFocus
   Exit Sub
ElseIf Len(txtName.Text) = 0 Then
   MsgBox "Name is not entered", vbExclamation, "HIS"
   txtName.SetFocus
   Exit Sub
ElseIf Len(txtDosage.Text) = 0 Then
   MsgBox "Dosage is not entered", vbExclamation, "HIS"
   txtDosage.SetFocus
   Exit Sub
ElseIf Len(txtExpiry.Text) = 0 Then
   MsgBox "Expiry date is not entered", vbExclamation, "HIS"
   txtExpiry.SetFocus
   Exit Sub
ElseIf Len(txtPrice.Text) = 0 Then
   MsgBox "Price is not entered", vbExclamation, "HIS"
   txtPrice.SetFocus
   Exit Sub
ElseIf Len(txtStock.Text) = 0 Then
   MsgBox "Stock is not entered", vbExclamation, "HIS"
   txtStock.SetFocus
   Exit Sub
ElseIf Len(txtDistributed.Text) = 0 Then
   MsgBox "Distributed is not entered", vbExclamation, "HIS"
   txtDistributed.SetFocus
   Exit Sub
End If

'check if stock is numeric
If Not IsNumeric(txtStock.Text) Then
   MsgBox "Stock must consists of integers", vbExclamation, "HIS"
   txtStock.SelStart = 0
   txtStock.SelLength = Len(txtStock.Text)
   txtStock.SetFocus
   Exit Sub
```

```
End If
With Adodc1.Recordset
   .Fields("Code").Value = Left(txtCode.Text, 20)
   .Fields("Name").Value = Left(txtName.Text, 30)
   .Fields("Type").Value = Left(cboType.ListIndex, 30)
   .Fields("Dosage").Value = Left(txtDosage.Text, 20)
   .Fields("Expiry").Value = Left(txtExpiry.Text, 20)
   .Fields("Price").Value = Left(txtPrice.Text, 20)
   .Fields("Stock").Value = Left(txtStock.Text, 20)
   .Fields("Distributed").Value = Left(txtDistributed.Text, 20)
   .Update
   frmControls
   ListView1.ListItems.Add 1, , txtName.Text
End With

fraMedicine.Refresh
Me.Width = 5400

ListView1.Enabled = True
cmdEdit.Enabled = True
cmdInfo.Enabled = True
cmdDelete.Enabled = True
cmdExit.Enabled = True
End Sub

Private Sub ListView1_Click()
Dim Name As String

If ListView1.ListItems.Count = 0 Then Exit Sub
Name = ListView1.SelectedItem.Text
showRecord Name, "medicine"

cmdSave.Visible = False
cmdCancel.Visible = False
cmdUpdate.Visible = False
cmdOK.Visible = True
End Sub

Private Sub ListView1_DblClick()
ListView1_Click
Me.Width = 9900
cmdOK.Visible = True
cmdNew.Enabled = False
End Sub
```

```
'Display medicine fields information
Public Sub showRecord(Name As String, loadby As String)
If loadby = "medicine" Then

  With Adodc1.Recordset
    If .RecordCount = 0 Then Exit Sub
    .MoveFirst
      Do While Not .EOF
      If !Name = Name Or !Code = Name _
      Or !Dosage = Name Then
      GoTo endOfSearch
      Else
      .MoveNext
      End If
     Loop
    Exit Sub

endOfSearch:
frmPharmacy.txtCode = !Code
frmPharmacy.txtName = !Name
frmPharmacy.txtDosage = !Dosage
frmPharmacy.txtExpiry = !Expiry
frmPharmacy.cboType.ListIndex = !Type
frmPharmacy.txtPrice = !Price
frmPharmacy.txtStock = !Stock
frmPharmacy.txtDistributed = !Distributed

  End With

End If
End Sub

'Clear all fields
Private Sub clearFields()
txtCode.Text = ""
txtName.Text = ""
txtDosage.Text = ""
txtExpiry.Text = ""
txtPrice.Text = ""
txtStock.Text = ""
txtDistributed.Text = ""
cboType.ListIndex = 0
End Sub

'Lock fields
Private Sub frmControls()
```

```
txtCode.Locked = True
txtName.Locked = True
txtDosage.Locked = True
txtExpiry.Locked = True
cboType.ListIndex = True
txtPrice.Locked = True
txtStock.Locked = True
txtDistributed.Locked = True
cboType.Locked = True

cmdSave.Visible = False
cmdCancel.Visible = Fase
cmdUpdate.Visible = False
End Sub

Private Sub frmFill()
'Type Dropdown List'
cboType.AddItem ""
cboType.AddItem "Tablet"
cboType.AddItem "Liquid"
cboType.AddItem "Capsule"
cboType.AddItem "Soft Gels"
cboType.ListIndex = 0
End Sub

'Disable fields
Private Sub frmDisabled()
txtCode.Enabled = False
txtName.Enabled = False
txtDosage.Enabled = False
txtExpiry.Enabled = False
cboType.Enabled = False
txtPrice.Enabled = False
txtStock.Enabled = False
txtDistributed.Enabled = False
End Sub

'Enable fields
Private Sub frmEnabled()
txtName.Enabled = True
txtDosage.Enabled = True
txtExpiry.Enabled = True
cboType.ListIndex = True
txtPrice.Enabled = True
txtStock.Enabled = True
txtDistributed.Enabled = True
```

frmInventory – Add or Remove stock from storage

```
Private Declare Function DeleteMenu Lib "user32" (ByVal hMenu As Long, ByVal
nPosition As Long, ByVal wFlags As Long) As Long
Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal
bRevert As Long) As Long

Private Const MF_BYPOSITION = &H400&

'Function to remove the windows close button ('x')
Private Sub RemoveMenus()
  Dim hMenu As Long

  hMenu = GetSystemMenu(hWnd, False)
  DeleteMenu hMenu, 6, MF_BYPOSITION
End Sub

Private Sub Form_Load()
Me.Move 0, 0
RemoveMenus
lstInventori.Clear
Call ViewAlgorithm
lstInventori.Selected(0) = True
Call lstInventori_Click
End Sub

'Add stock to available stock
Private Sub cmdAdd_Click()
Dim totambah As String
totambah = lstInventori.Text
On Error GoTo err
If totambah = "" Then
  MsgBox "Please choose from the medical list", vbCritical, "HIS"
  Exit Sub
End If

Dim box As Long
box = InputBox("Amount of " & lstInventori.Text & " added into stock", "HIS")
With Adodc1.Recordset
.MoveFirst
  While Not .EOF
    If !Name = lstInventori.Text Then
    MsgBox "" & box & " Stock has been added", vbInformation, "HIS"
    !Stock = !Stock + box
    .Update
    txtStock.Text = !Stock
```

```
        .MoveNext
        Else
        .MoveNext
        End If
    Wend
End With
Exit Sub

err:
MsgBox err.Description, vbCritical, "HIS"
Exit Sub
End Sub
'Remove stock from available stock
Private Sub cmdRemove_Click()
Dim totolak As String
totolak = lstInventori.Text
On Error GoTo err
If totolak = "" Then
    MsgBox "Sila pilih pilihan daripada list", vbCritical, "HIS"
    Exit Sub
End If

Dim box As Long
box = InputBox("Amount of " & lstInventori.Text & " removed from stock", "HIS")
With Adodc1.Recordset
.MoveFirst
    While Not .EOF
        If !Name = lstInventori.Text Then
        If (!Stock - box) < 0 Then
        MsgBox "Quantity of stock exceeds available stock", vbInformation, "HIS"
        Exit Sub
        Else
        !Stock = !Stock - box
        MsgBox "" & box & " Stock has been removed", vbInformation, "HIS"
        End If
        .Update
        txtStock.Text = !Stock
        .MoveNext
        Else
        .MoveNext
        End If
    Wend
End With
Exit Sub
```

```
err:
MsgBox err.Description, vbCritical, "HIS"
Exit Sub
End Sub

'Exit form
Private Sub cmdExit_Click()
Unload Me
End Sub

'Find records based on selected text in inventory's list
Private Sub Search(ByVal Name As String)
Adodc1.Recordset.MoveFirst
Do
   If (Adodc1.Recordset.Fields("Name") = Name) Then Exit Sub
   Adodc1.Recordset.MoveNext
   DoEvents
Loop While Adodc1.Recordset.EOF = False
End Sub

'View all medicine name from the Medicine table
Private Sub ViewAlgorithm()
Do
   lstInventori.AddItem Adodc1.Recordset.Fields("Name").Value
   Adodc1.Recordset.MoveNext
   DoEvents
Loop While Adodc1.Recordset.EOF = False
End Sub

'View stock amount in the textbox
Private Sub lstInventori_Click()
   Call Search(lstInventori.Text)
   txtStock.Text = CStr(Adodc1.Recordset.Fields("Stock").Value)
   txtDistributed.Text = CStr(Adodc1.Recordset.Fields("Distributed").Value)
End Sub
```

frmReminder – Remind the user if medicine stock amount is below 50

```
Private Declare Function DeleteMenu Lib "user32" (ByVal hMenu As Long, ByVal
nPosition As Long, ByVal wFlags As Long) As Long
Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal
bRevert As Long) As Long
Private Const MF_BYPOSITION = &H400&

'Function to remove the windows close button ('x')
Private Sub RemoveMenus()
Dim hMenu As Long
hMenu = GetSystemMenu(hWnd, False)
DeleteMenu hMenu, 6, MF_BYPOSITION
End Sub

'Display medicine ID, name and stock when it is below 50
Private Sub Form_Load()
Me.Move 0, 0
Dim txt As Integer
txt = 50
RemoveMenus

With Adodc1
   .Refresh
   .RecordSource = "select * from Medicine"
   .Refresh
End With

With Adodc1.Recordset
   If .RecordCount = 0 Then Exit Sub
   .MoveFirst
      While Not .EOF
      Dim low As Integer
      low = !Stock
      If low < 50 Then
      ListView1.ListItems.Add , , "Medicine ID:" & !Code & " - " & "Name:" & !Name
& " - " & "Balance:" & low & ""
      .MoveNext
      Else
      .MoveNext
      End If
      Wend
End With
End Sub

Private Sub cmdOK_Click()
```

```
Unload Me
End Sub
```

frmSearch – Find information regarding patients

```
Private Declare Function DeleteMenu Lib "user32" (ByVal hMenu As Long, ByVal
nPosition As Long, ByVal wFlags As Long) As Long
Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal
bRevert As Long) As Long
Private Const MF_BYPOSITION = &H400&

'Function to remove the windows close button ('x')
Private Sub RemoveMenus()
Dim hMenu As Long
hMenu = GetSystemMenu(hWnd, False)
DeleteMenu hMenu, 6, MF_BYPOSITION
End Sub

Private Sub Form_Load()
Me.Move 0, 0
frmFill
RemoveMenus
End Sub
'Function to search exact match
Private Sub cmdSearch_Click()
If txtSearch = "" Then
   MsgBox "Search box is empty"
Else
   lstResults.Clear
   lstResults2.Clear

'Search by ID
If cboSearch.ListIndex = 0 Then
With Adodc1
.Refresh
.RecordSource = "select * from Patient where ID = '" & txtSearch.Text & "'"
.Refresh
End With

display

'Search by NRIC
ElseIf cboSearch.ListIndex = 1 Then
With Adodc1
   .Refresh
   .RecordSource = "select * from Patient where NRIC = '" & txtSearch.Text & "'"
   .Refresh
End With
```

```
display
'Search by Company
ElseIf cboSearch.ListIndex = 2 Then
With Adodc1
  .Refresh
  .RecordSource = "select * from Patient where Company = '" & txtSearch.Text & "'"
  .Refresh
End With

display
End If
End If
frmSearch.Refresh
End Sub

Private Sub cmdExit_Click()
  Unload Me
End Sub

'Display the results in the list box
Private Sub display()
With Adodc1.Recordset
  If .RecordCount = 0 Then Exit Sub
  .MoveFirst
    While Not .EOF
    lstResults.AddItem Adodc1.Recordset.Fields("Name").Value
    lstResults2.AddItem Adodc1.Recordset.Fields("ID").Value
    .MoveNext
    Wend
End With
End Sub

'Fill the category list item
Private Sub frmFill()
cboSearch.AddItem "ID"
cboSearch.AddItem "NRIC"
cboSearch.AddItem "Company"
cboSearch.ListIndex = 0
End Sub

Private Sub lstResults2_Click()
lstResults.Selected(lstResults2.ListIndex) = True
End Sub

Private Sub lstResults_Click()
```

```
lstResults2.Selected(lstResults.ListIndex) = True
End Sub

'Show patient form if double clicked on the results
Private Sub lstResults2_DblClick()
Dim name1 As String
name1 = lstResults2.Text

frmPatient.Show
frmPatient.showRecord name1, "patient"
frmPatient.Width = 10590
End Sub
```