FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY MALAYA

SCHOOL FACILITIES MANAGEMENT SYSTEM (SFMS)

BY
GAN HUI HOON

Under the supervision of

Puan Abrizah Abdullah

Session 2000/2001

Dissertation submitted in partial fulfillment of the requirement for
Degree of Bachelor of Computer Science
University Malaya

ABSTRACT

Computer information system and computer applications are very important to all the organizations. Thus the government is introducing the MSC or Multimedia Super Corridor. It is meant to develop Malaysia into a regional and international technology and telecommunications hub in 2020. The MSC will propel the transfer of technology and become the test bed for R&D in high-tech industries. There are seven-flagship applications in MSC, and Smart School is one of the seven flagship applications in MSC.

School Facilities Management System will be an example of what the function of facilities management in Smart School will be like. The facilities management function is one of the ten functions of Smart School implementation in Malaysia.

This report is about the development **School Facilities Management System, SFMS**. The system provides the facility to keep track of the usage of facilities in schools. It includes an inventory module that will inventories special equipments and apparatus in special rooms. The system will be able to remain the user to do maintenance for the needed equipments.

There are seven chapters in this report, which includes Introduction, Literature Review, System Analyst, System Design, System Implementation, Testing, and System Evaluation and Conclusion. The first four chapters are considered as phases that a system analysts need to go through before one can come out with clear and concise definitions of all the functional and non-functional requirements that are needed to develop a good application later in the future. The last three chapters are the remaining phases of system development, which uses all findings and information, gather in the earlier four chapters.

ACKNOWLEDGEMENT

First of all, I would like to thank Puan Abrizah Abdullah, my supervisor for giving me this opportunity to develop this project. Secondly, I would like to thank her for her constructive advice, generous guidance, encouragement, support and supervision along the project. Her diligence and kindness in helping me throughout the project is deeply appreciated.

Thanks to Mr. Teh Kang Hai, as the project moderator who contributed suggestions and ideas to further enhance value of this project.

I want to express my gratitude to my teammate Miss. Lok Farn who helped me and give me support and assistance to me throughout the project. Thanks to all my course mates especially Mr. Ng Tuck Wei, Miss Cheah Sooi Ping and Miss Fong Teng Heng for their valuable ideas. Not forgetting my housemates Miss Tey Su Ling and Mr. Mok Wai Hon who have given me ideas on my questionnaire.

Thank you very much!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 PROJECT BACKGROUND

This project is about the development of application software designed to assist in management of the school's facilities. This application or system will be called School Facilities Management System (SFMS). SFMS is specially designed to meet the requirements of the School Administrative Staffs in managing the school's facilities, so that the facilities will not be misuse and wasted.

Strictly stated here is that SFMS is not designed to replace the talents of the School's Administrative Staffs, but will provide all the power of the latest technology available to them in order to facilitate their work and reduce the repetitive work. A computer has never been able to replace an expert in any domain because the role of computers is just to provide a powerful management tool to all who needed it.

The system will have a database to keep all the information and data needed on facilities usage as well as inventory. A database is now such an integral part of our daily life that often we are not aware that we are using one. SFMS will be aimed at developing a computerized database system where it will handle data entry, file maintenance and generation of a fixed set of specific reports. SFMS also include system administrator control that only allows user to view certain subsystem. It will collect and preserved information on facilities in which in the past has been available only on paper.

SFMS will be an example of what the function of facilities management in Smart School will be like. The facilities management function is one of the ten functions of Smart School implementation in Malaysia, which will only be introduced to schools in the year 2001. Smart School is one of the seven flagship applications in MSC. The MSC or Multimedia Super Corridor is meant to develop Malaysia into a regional and international technology and telecommunications hub in 2020. The MSC will propel the transfer of technology and become the test bed for R&D in high-tech industries. The flagship will support the government's plans to obtain the status of a developed nation by the year 2020 and to gain a competitive edge with other developed countries in the global economy.

1.2 STATEMENT OF THE PROBLEM

The decision to build the application software to manage the school facilities is indeed the consequences from the difficulties and problem arise from the manual filing system.

The manual filing system is too time consuming as well as it involves a lot of clerical work. A lot of school facilities are not probably used due to poor maintenance and the human inability to keep up with the increasing facilities in a particular school.

Therefore, a management information system is the best solution for all the difficulties and troubles faced. But as stated earlier the system is not meant to replace the Administrative Staffs but it will facilitate their work in tracking the usage of every facility in the school. SFMS will also allow the user to keep information about particular equipment in the school, its location, its vendor, and any other information pertaining to inventory.

1.3 AIMS OF THE PROJECT

The aims of developing School Facilities Management System are as followed:

i. Preventing the misuse of school facilities
ii. Reducing School Administrative Staff's workload
iii. Keeping the inventory records.

1.4 OBJECTIVE OF THE PROJECT

The objectives for developing Scholl Facilities Management System are as followed:

i. To provide at the desktop all graphics and textual information needed for the management of facilities information in a particular school
ii. To help in tracking the usage of facilities in the school
iii. To inventories fixed assets in the school
iv. To store information about equipments in school for easier and faster retrieval.
v. To generate inventory number.

1.5 SCOPE AND LIMITATION OF SFMS

The SFMS project has its own scope and limitation. This project is a management information system that will be installed into one computer and the computer will act as the host of the whole system. Due to time limitation in developing the SFMS system, the system can only be accessed locally. The system will only keep track of the fixed assets in the school, such as tables and chairs, as well some of special equipment in special rooms. For further enhancement of the project, in the future, the system is hoped to maintain the entire inventory in the school, which include laboratory apparatuses, library and etc.

The project will be divided into modulus as below:

Module 1: Method of access

The system is designed to allow authorized users to access School Facilities Management System (SFMS) by using a user ID and password. Only valid user is allowed to access into SFMS. Not every user has the same authorization to the system. The school Principle

will have the full authorization in changing all the records in the database, so as the teacher who will be in charging of maintaining the school facilities. (will be develop by Lok Farn)

Module 2: Tracking Usage

The system is designed to track all the usage of every facility in the particular school. User will be able to view possible usage of a particular room. The user also can obtain information on types of equipments stored in the room. Thus, teachers can make decision whether to make use of the room for their teaching.

Module 3: Request For Service

This module will enable users of a particular facilities or equipment to report the malfunction of the equipment to the school clerk so that the clerk will assign the equipment to be fix as soon as possible.

Module 4: Maintenance

The system will give appropriate message regarding when will be the best time to do maintenance of needed assets. A pop up window will appear when the user run the system.

Module 5: Inventory

The system will keep track of all the fixed assets in the school, where it will keep information on the types of furniture available, their quantity, their quality and their

vendors. The system will also keep track of equipments and apparatus in special rooms such as the TV room.

Module 6: Booking

SFMS will be designed to be able to allow users especially Teachers to reserve rooms and equipments. By booking the rooms or equipments earlier, teachers will be able to make sure that the particular facilities will be ready when the time comes. (will be develop by Lok Farn)

Module 7: Report Generation

SFMS will be designed to generate report needed for the School Administrative Staffs. The report can be printed ac-hoc or it can be set to print the require report on periodic basic.

1.6 SYSTEM REQUIREMENT

1.6.1 Hardware Requirement

The hardware requirements for developing SFMS are:

Component	*Description*
Microprocessor	IBM compatible machine with Pentium 133MHz processor or higher
RAM	> 16 MB RAM
Storage	40 MB of hard disk
Input Device	Mouse and keyboard
Video Monitor	EGA, VGA or compatible display

Table 1.1 Hardware Requirement for developing SFMS

1.6.2 Software Requirement

The software requirements for developing SFMS are:

Component	*Description*
Operating System	Windows 95 or later in standard
Programming Language	Visual Basic Enterprise Edition 6
Database	Microsoft Access 2000

Table 1.2 Software Requirement for developing SFMS

1.7 PROJECT SCHEDULE

SCHOOL FACILITIES MANAGEMENT SYSTEM

ID	Task Name	Start	End
1	Understanding	6/5/00	6/6/00
2	Literature Study	6/11/00	7/8/00
3	Methodology & System Analysis	7/5/00	7/30/00
4	VIVA Preperation	7/27/00	8/3/00
5	Interface Design	8/5/00	8/5/00
6	Documentation	7/1/00	9/5/00
7	Submitting Report	9/16/00	9/16/00

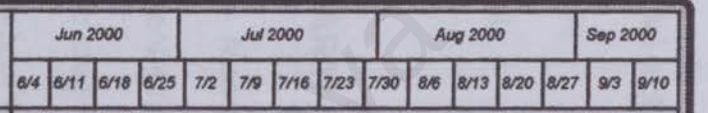

Figure 1.1 Gantt Chart For SFMS

1.8 RESEARCH PLAN

The research plan to gather information for developing SFMS is as followed:

- Questionnaire
- Printed material such as previous thesis
- Internet surfing
- Brainstorming

1.9 SUMMARY

School Facilities Management System is a computer based information system that manages the school's facilities. The system is going to help the School Administrative Staffs to track the usage of every facility listed in the database. It also enables the user to inventories the school's fixed assets as well as the equipments and apparatus in every special rooms in a particular school.

In the next chapter, the literature reviews are highlighted. It will show how other people develop application(s) that is similar to this one.

CHAPTER 2 LITERATURE REVIEW

2.1 INTRODUCTION TO FACILITIES MANAGEMENT SYSTEM

The Cambridge International Dictionary of English defines facilities as buildings, equipment and services provided for a particular purpose. In my case, facilities are restricted to those found in school, especially secondary school in a Malaysian environment. Facilities in schools include classrooms, laboratories, meeting rooms and others. Special apparatus and equipments such as projector and television are also regarded as school facilities. These facilities will help to give a better environment for students to learn and study. So, the government has invested a great deal of money to build and buy facilities for schools. Every year more and more facilities are being build to meet the increasing number of students in every school. Due to increasing facilities, computerized facilities management system is indeed a must so that facilities are not wasted and misuse.

The Facilities Management System will acts as the central data source for comprehensive and detailed information about facilities, equipment and assets available to the school. School administrative staffs will identify the facility components and assets that are needed to be manage. Information tracked about these elements in conjunction with built-in analysis and reporting then become the basis for good, sound facility decisions.

2.2 SCHOOL FACILITIES MANAGEMENT SYSTEM AS A MANAGEMENT INFORMATION SYSTEM

School Facilities Management System is a Management Information System that can evolve into a Decision Support System that can assist the school administrative staffs in decision-making.

Management Information System is a computer-based system that generates timely and accurate information for the top, middle, and lower levels of management of an organization. School Facilities Management System is a MIS where it will help the school administrative staff to manage and control the usage of school facilities. It will collect and preserved information on facilities in the past has been available only on paper.

2.3 RESEARCH ON EXISTING PACKAGES

Hospital's Inventory System (Fadzil 1998/1999)

This particular system is developed to computerize the inventory system in a hospital. The system includes the geographical location for each inventory item so that the hospital's staff could know where each item is located, so that they can easily access each item to provide a better service to patients. The system provides the capabilities to access, records and manage the entire inventory item in the database. It also includes the picture of each device. However the system was developed for installation in a PC not for the network.

This system has good way of managing equipment in hospital, but it should be installed in the hospital's network, so that it can be access anywhere in the hospital, as most hospital has a wide compound.

FSKTM Asset Management System (Faridah 1996/1997)

The system was developed using GIS, Geographic Information System (Mapinfo) with Window 3.11 interface. The system is capable to give information for each asset owned by FSKTM including information on which room the asset is located and who is responsible to the particular asset. The system also able to give the value in term of total cost of assets in each floor and room in FSKTM. It also provides the facility to give data or information of asset according to floor, room and usage.

However, the system does not have the capability that enables the user to reserve the equipment. Application written in Maphinfo does not allow the used of icon in buttons. So, the interface of the application is quite boring and not attractive. Another weakness of the application written in Maphinfo is that it does not change to a executable file after running and compiling. So, without Maphinfo installed in the PC, the application cannot be run.

Faculty Information System (Azny 1996/1997)

The system was developed using PowerBuilder 5.0. with Window 95 interface. There are 3 functional modules, which was divided by its user. The modules are Students, Staff and Asset.

The Asset module is being analyst because it is more similar to SFMS. In this particular module, it allows the user to do Asset Data Entry, New Item Purchase

Detail, Malfunction Asset Info, Asset Viewing, View by Location, Complaint, and Report

The graphical user interface is relatively simple and easy to use.

2.4 FACILITIES MANAGEMENT SYSTEM AVAILABLE COMMERCIALLY

Intergrow FMS (Facilities Management System) [2]

Integrow FMS is a fully integrated Facilities Management System developed by Higher Education Technology Company. The modules included in INTEGROW FMS are:

a) Facilities Management

The Facilities Management module is able to:

i) Maintain an Inventory Database for All Physical Facilities
ii) Provide Reports Conforming to the State Mandate
iii) Provide Facility to Produce Ad-Hoc Reports
iv) Provide On-Line Validation for Date Entered
v) Schedule Facilities for Usage by Students
vi) Link into Student Facility to Provide Information on classrooms, laboratories, meeting rooms, conference facilities, time between facilities, and its locations
vii) Update Facilities from Student with Usage
viii) Provide ability to Schedule Facilities Not Used by Students such as theaters, sports fields, and conference centers, parking areas, swimming pools, and equipment.

ix) Provide Roster of Equipment with Details such as Equipment Description, Location, Condition, Availability, and Status.

The interface in this facilities management module has the capabilities to provide:

i) Interface to Student System with Details on Classrooms, Laboratories, Meeting Rooms and Study Facilities

ii) Update the Above Schedules with Student Information

iii) Schedule All Other Facilities Conference Centers, Theaters, Sporting, Canteens, and Parking.

iv) Additional Feature to Allow Seat Booking for Theaters

b) Inventory

The inventory modules is able to:

i) Maintain Inventory Database

ii) Produce Reports Conforming to State Mandate

iii) Flexible Reporting Capability

iv) On-Line Validation of Data Entered

v) On-Line Help for Date Entered

The interface in this module is able to:

i) Provide an Interface to Student System with Information on Class, Lecture Halls, Meeting Rooms, Seating, Equipment, Covered Walkway, Distance Between Rooms, Maps, On-Site/Off-Site, Size

ii) Update Facilities with Information from Student System

Computer-Integrated Facilities Management in University of Florida Health Science Center [8]

The University of Florida Health Science Center has implemented a Computer-Integrated Facilities Management System to provide accurate, up-to-date information about university facilities.

Current facilities available to the system are as below:

View statistics on University of Florida Health Science Center space, including building and room sizes, uses, assignments, and occupants:

i) By Building

ii) By College and Department

iii) Enter a Service Request for the Health Center Physical Plant Department. *(Authorized users only - Password Validation Required)*

Facilities Information Management System (FIMS) in Indiana University [3]

The Facilities Information Management System (FIMS) will use technologies including imaging and image processing, relational databases, graphic user interface capabilities, and geographic information systems to collect, store, retrieve, maintain, and analyze facilities and infrastructure information across Indiana University.

Objectives:

- Provide at the desktop all graphic and textual information needed for the planning, design, construction, maintenance, or management of facilities and infrastructure
- Improve the operating effectiveness of facilities and infrastructure to support research and instruction

- Effect cost savings by eliminating duplication of effort and information, streamlining the design process, providing accurate planning data to eliminate costly construction mistakes, increasing operating efficiencies, and preserving deteriorating paper-based information
- Implement an improved work order system that will give customers on-line access to project status
- Provide data and data management tools for capital planning and for analysis of facilities and infrastructure

Main modules:

The three main components of FIMS are:

i) CADD (Computer aided design and drafting) - electronic drawings of facilities

ii) GIS (Geographic Information System) - a system to capture, store, update, manipulate, display and analyze geographically referenced information such as a campus site plan

iii) Relational Database - a database in which some data items in one type of record (building and room inventory) refer to records of a different type (equipment inventory of Physical Plant work orders), giving the user flexibility to link, join or create a relationship between data stored in many fields. This allows interchange and cross-reference between different types of records

Facilities Management System in Shadyside Hospital in Pittsburgh [7]

Facilities Management System in Shadyside Hospital is a system that will list costs associated with outpatient services, including the use of equipment and rooms. The report generated by the system will be able to allocate depreciation and distribute overhead expenses. It runs on one of the hospital's 26 client servers.

Main modules of the system:

i) Inventory

This module will allow each department in the hospital to tracks its inventory and to do the necessary new purchases of equipment.

ii) Facility Management

The system will be able to track the usage of each equipment in every department of the hospital as opposed to doing a walk-through to gather information. It will also allow space allocation and to computerized its facility plan.

Precision Inventory Management System (PIMS) [5]

PIMS is designed for use in public and private school systems that run on MAC System 6 or Mac System 7. It provides a fast, organized method for maintaining current records relating to the managing of inventory and fixed asset operations.

Main module:

i) Inventory

The system will be able to inventories inventory item for a fiscal school year. Stock items may be received, shipped and adjusted in each district's location.

ii) Tracking Usage

This module will enable the user to track all the usage of fixes assets available in the school. The user will just have to choose the assets in the list provided by the database, and it will automatically list out the current usage of the asset.

Fixed Asset Inventory System

Fixed Asset Inventory System provides asset-tracking capabilities for school district equipment administration. It provide information on asset condition, purchasing

account history, location of asset, tracking of movement, salvage value, and repair history.

Main modules in the system are:

i) Tracking Usage

ii) Inventory

iii) Bar code data entry

Rhodes Facilities Management [6]

Rhodes Facilities Management is a system that can be access through the Internet. This system provides an example how an online facilities management system should be like. The system has the following modules:

i) View Availability of a Space

ii) View Availability of an entire Building

iii) Info. About Campus Facilities

iv) Request a Space Reservation

v) Instructions for Students

vi) Search Calendar of Events

vii) Instructions for Faculty/Staff

viii) Check Status of a Request

The system provides detail information in each space (room) in each building available in the campus. The comments given by the system really help the user to make decision regarding on the usage of room.

2.5 TECHNOLOGY REVIEW

2.5.1 Visual Basic 6.0

Visual Basic is a simple, easy to learn language and programming environment, which can be used to build real application for window. It is widely used in business industry and IT industry for developing rapid prototypes of new applications. The features are show next [4]:

Visual Basic's Technical Strength

a) Event-Driven Programming

An event is an action of some function, fore example moving the mouse, selecting a item in a list, or clicking a button are events. These events drive what happens in Visual Basic program. This idea is quite different from traditional programming, in which we write lines of code that are executed sequentially. In event-driven programming, the code to run is determined by the type of event.

b) Visual Basic Control

Visual Basic is designed, in such a way that we can immediately design the windows that we wish. It has the ability to create and use self-contained components, or objects.

Controls are elements you can use when designing a user-interface, just like the real life control. These controls can be used to display information or to take action.

Visual Basic control enables us to add features to our programs without having us involved in the details of simple drawing a control that accepts input.

c) **Data Access Objects (DAO) and ActiveX Data Objects (ADO)**

DAO can create complete, robust data management application. DAO act as a visual program internal representation of physical data stored in some type of database or data management engine. Think of Data Access Objects as special types of variables. These variables, however, represent data stored outside the program rather than information stored in the computer's memory while the programs running.

ActiveX Objects, or ADO, is most recent method of data access that Microsoft has introduced. ADO is intended to replace DAO, the original method of Visual Basic database access.

2.5.2 PowerBuilder

PowerBuilder is a graphic PC-based client/server application development environment. Using PB one can develop front-end applications which access RDBMs (Relational Database Management Systems) without coding in a 3GL(3rd Generation Language) such as C or C++. PowerBuilder uses it's own Powerscript, that is a basic-like language, that uses screens called painters to graphically put together applications. Powerscript is a 4GL (4th Generation Language) [1].

PowerBuilder's Technical Strengths

a) **DataWindow**

The primary strength of PowerBuilder is its proprietary device called DataWindow. Some people have called it a product within a product. Others are still discovering new uses for it. It is the primary means by which a PowerBuilder application talks to the database. It has built-in features to format data for display,

allow different edit-styles, validate data entered by user, generate appropriate SQL based on the changes made by a user and also the RDBMS it is talking to and scores of other such invaluable features.

b) Object-Oriented (OO)

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

c) Native Drivers

Though ODBC (Open Database Connectivity) is good for accessing multiple databases through a common gateway, it covers only the common minimum features of these databases. PB provides native drivers for all the major RDBMSs, such as Oracle, Sybase, Informix, DB2, MS SQL Server...etc., so that you can take advantage of the power of these.

d) Cross Platform

You can write code once and run that application on all the flavors of Windows, namely Windows 3.1, Windows for WorkGroups, Win95 and Windows NT. You can also use the same code to run the application on Mac and Sun Solaris Unix.

e) Web-enabled

With PowerBuilder 5.0 and the Internet add-ons, you can build an application,

which can access data in an RDBMS through a browser, whether it is on the corporate intranet or on the Internet.

Minimum configuration programming with PB

The minimum configuration recommended for PB is: 486 with 8 MB of RAM. The windows permanent swap file should be set to 12 MB. And don't have any other application running on the background [1].

2.5.3 ODBC

ODBC (Open Database Connectivity) provides a way for client programs (eg Visual Basic, Excel, Access, Q+E etc) to access a wide range of databases or data sources.

ODBC is a standardized API, developed according to the specifications of the SQL Access Group, which allows one to connect to SQL databases. It defines a set of function calls, error codes and data types that can be used to develop database independent applications. [9]

ODBC is usually used when database independence or simultaneous access to different data sources is required.

2.5.4 Microsoft Access 2000

Microsoft Access 2000 is included in Microsoft Office 2000 pack. Using Microsoft Access, one can manage all the information from a single database file. Within the file, data is divided into separate storage containers called tables. User can view, add and update table data by using online forms in Access. User also can find and retrieve

just the data needed by using queries. They also can analyse or print data in a specific layout by using reports, a function in Access. [4]

Microsoft Access 2000 is the Microsoft Office Database Management System available when one chooses to install the typical Microsoft Office

Access 2000 System Requirements

Here are the requirements to run Microsoft Access 2000:

- PC with a Pentium 75 megahertz (MHz) or higher processor
- Microsoft Windows® 95 or later operating system, or Microsoft Windows NT® Workstation operating system version 4.0 Service Pack 3 or later
- For Windows 95 or Windows 98: - 16egabytes (MB) of RAM for the operating system, plus an additional 8 MB of RAM for Access.
- 161 MB of available hard-disk space (Number indicates typical installation; your hard-disk usage will vary depending on configuration. Choices made during custom installation may require more or less hard-disk space.)
- CD-ROM drive
- VGA or higher-resolution monitor; Super VGA recommended
- Microsoft Mouse, Microsoft IntelliMouse®, or compatible pointing device

Additional items or services required to use certain features:

- 9600 baud modem; 14,400 or higher-baud modem recommended
- Multimedia computer required to access sound and other multimedia effects

- Microsoft Outlook 2000 or Microsoft Outlook Express 5.0 or later required to run Office E-mail
- 8 MB of additional memory required to run Office E-mail
- Some Internet functionality may require Internet access and payment of a separate fee to a service provider

2.6 DEVELOPMENT METHODOLOGIES

The system development methodology is important during the development of software as it forms a common understanding of activities, researches, resources and constrains involved in a software development. When a methodology is identified, it can help us to find the inconsistencies, redundancies and omission in the process. There are many types of methodology available such as Waterfall Model, Prototyping, Transformation Model, Spiral Model and others. For my literature review, I have studied through the below methodology:

(1) Waterfall Model

One of the first models to be proposed is the waterfall model where stages are depicted as cascading form one another. One development stage will be completed before the next begins. Thus, when all the requirements are elicited from the user, analyzed for completeness and consistency, and documented in a requirement document, then the development team can go to the system design activities. The Waterfall model presents a very high level view of what goes on during development, and it suggests to developers the sequence of events they should expect to encounter.

The waterfall model can be very useful in helping developers lay out what they need to do. Its simplicity makes it easy to explain to users who are not familiar with software development.

During design, problems with requirements are identified, during coding, design problems are found and so on. The process is not a simple linear model but involves a sequence of iterations of the development activities.

Unfortunately, a model, which includes frequent iteration, makes it difficult to identify management checkpoint for planning and reporting. Therefore, after a small number of iteration, it is normal to freeze parts of the development, such as the specification, and to continue with the later development stages. Problems are left for later resolution, ignored or are programmed around. This premature freezing of requirements may mean that the system won't do what the users wants. It may lead to badly structured systems as designed problems are circumvented by implementation tricks.

The problem with waterfall model is its inflexible partitioning of projects into these distinct stages. Delivered system is sometimes unusable, as they do not meet the uses real requirements. Nevertheless, the waterfall model reflects engineering practice. Consequently, it is likely that software process models based on this methodology will remain the norm for large hardware-software development [15]

(2) **Prototyping Model**

A prototype is a partially developed product that enables users and developers to examine some aspect of proposed system and decide if it is suitable or appropriate for the finished product. In other words, prototyping means building a small version of a system, usually with limited functionality that can be used to help user or customer identify the key requirement of a system and demonstrate feasibility of a design or approach.

Prototyping is often used to design a good user interface: the part of the system with which the user interacts. Since the prototyping models allows all or part of a system to be constructed quickly to understand or clarify issues, it has the same objective as engineering prototype, where requirements or design require repeated investigation to ensure that the developer, user, and the customer have a common understanding both of what is needed and what is proposed. One or more of the loops for prototyping requirements, design or the system may be eliminated, depending on the goals of the prototyping. However, the overall goal remains the same, reducing risk and uncertainty in development [15].

Application prototyping, the process of developing and using the prototype, has five characteristics:

i) The prototype is a live, working application.

ii) The purpose of prototyping is to test out assumption made by analysts and users about required system features.

iii) Prototypes are created quickly.

iv) Prototypes evolved through an iterative process.

v) Prototypes are relatively in expensive to build.

Application prototyping has two primary uses. On the one hand, it is an effective device for clarifying user requirements. Written specifications are typically created as a vehicle for describing application features and the requirements that must be met. Developing and actually using a prototype can be very effective way of identifying and clarifying the requirements an application must meet.

A second use of application prototyping is to verify the feasibility of a system design. Analyst can experiment with different application characteristics, evaluating user reaction and response. For instance, on method of interaction, whether through menus, special keys, or entry of keywords may be better than others for particular type of application than others. Processing procedures may change, leading to a more effective design. Creating a prototype and evaluating its designs through use will prove design feasibility or suggest the need to find other alternatives [15].

2.7 SUMMARY

There are many available software and application available in the market that provides the function to track facilities usage as well as asset inventory. However, only few of the systems developed for the use in school. It can be said that none of the system review here are for the use in school in Malaysia.

The following chapter is where the analysis of the system needs is being done.

CHAPTER 3 SYSTEM ANALYSIS

3.1 INTRODUCTION

The main purpose of the system analysis phase is to learn exactly what takes place in the current system, and to determine and fully document in detail what should take place. The result of this process will be used to recommend improvement to the system. [11] Through system analysis, the programmer may add, delete and modify system components toward the goal of improving the overall system. The information gathered during this phase has provided alternative strategies to develop this system. Through this phase also, the programmer can determine types of functional requirements and non-functional requirements for the system.

3.1.1 Objectives Of System Analysis

Following are some of the objectives of the analysis

i) To study the problem faced by the user

ii) To study the problem and find out the best solution to reduced it.

iii) To study how the new system will improve the current facilities management in a school.

iv) To acquire knowledge on how this system will be developed with the new emerging technology

v) Tools to develop the new system will be chosen among different types of new tools that have been studied and stated in chapter 2.

vi) To identify the major modules to be included in the system

vii) To identify what are the modules that are feasible to develop and the knowledge and tools need to have in order to develop them.

3.2 SYSTEM DEVELOPMENT METHODOLOGY

Before it is chosen, a methodology should reflect the goals of the development. Thus, after thorough studies of methodologies involved in Chapter 2, the proposed methodology chosen for this School Facilities Management System is Waterfall Model. Figure 3.1 shows the stages included in the model.

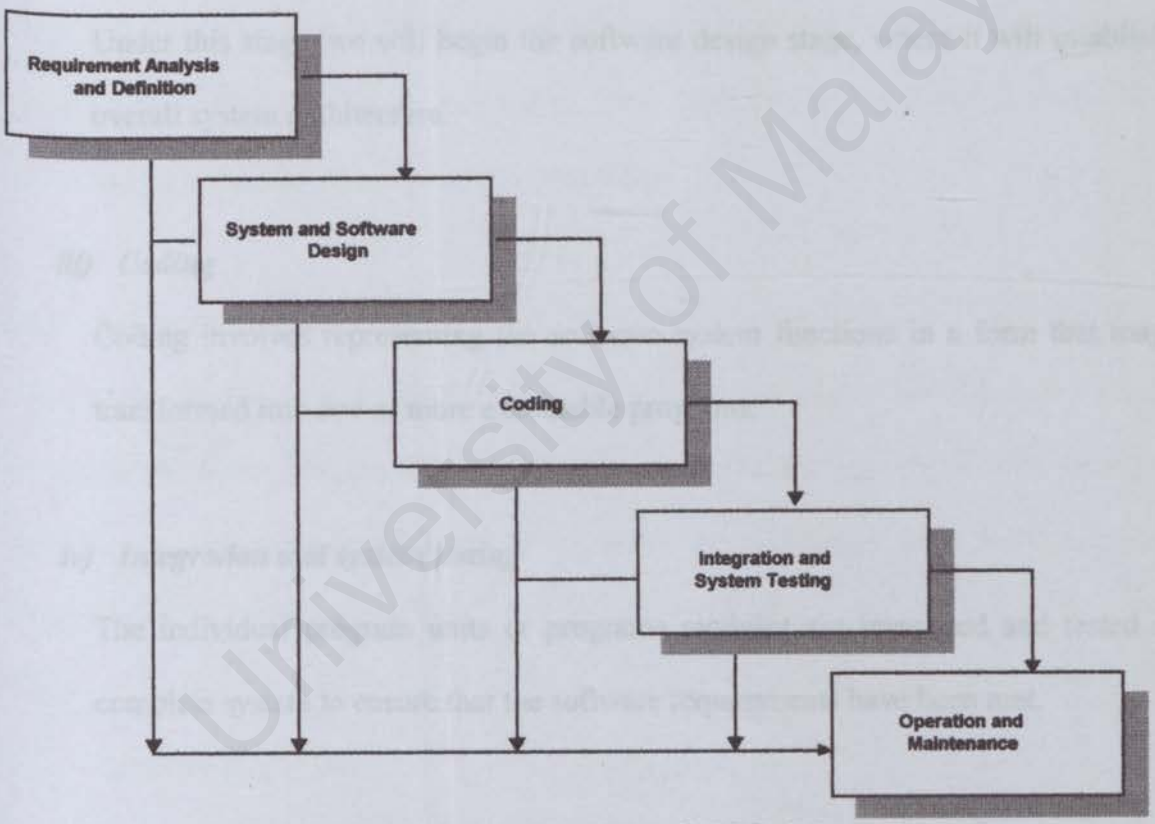

Figure 3.1 The Waterfall Model

The 5 stages of the Waterfall Model are discussed below:

i) Requirement Analysis and Definition

The concept, purpose and functionality of SFMS are identified and defined. During this stage, we have to study the existing system that is available in the market, and do planning for the new system. After finish this stage, it comes to system design.

ii) System and Software Design

Under this stage, we will begin the software design stage, where it will establish an overall system architecture.

iii) Coding

Coding involves representing the software system functions in a form that may be transformed into one or more executable programs.

iv) Integration and system testing

The individual program units or programs modules are integrated and tested as a complete system to ensure that the software requirements have been met.

v) Operation and maintenance

After testing, the system can be installed to its site to be fully utilized.

3.3 FACT FINDING TECHNIQUES

Facts finding is needed in order to have a better understanding of the system's needs and requirements. There are many sources that provide information in my research. The information gathering techniques involved are:

3.3.1 Questionnaire

Questionnaires provide an alternative to interviews for finding out information about a system. Questionnaire are chosen to be the main information gathering technique for this system because questionnaire can be widely distributed to many different schools, in a wider area. It is a quick way to gather massive amounts of data about how the users in a typical school in Malaysia in managing their facilities in the schools [12]. The questionnaire involved can be found in Appendix A.

3.3.2 Study On Previous Projects

Researches are done by studying through some of the similar system done by other seniors in the faculty. Through this method, the programmer can collect data on how previous system were developed, what were the functional and non-functional requirements, and other related data.

3.3.3 Internet Research

Internet is used as the main resource for referring any ambiguities that arise during the entire development period. Through the Internet, the programmer not only collect some idea from the similar system, but they can also find out some interesting web design and

feedback from the other remote users. Table 3.1 lists the keywords and search engine that I have used to gather information for this system:

KEYWORDS USED	SEARCH ENGINE
Facilities Management System	www.infoseek.com
School Facilities Management System	www.altavista.com
Inventory	www.yahoo.com
Inventory System	www.britannica.com
School Inventory System	www.ask.com
Sistem Pengurusan Kemudahan Sekolah	www.mol.com.my
Smart School	www.catch.com.my
Sekolah Bistari	

Table 3.1 Keywords and Search Engines

3.4 REQUIREMENTS SPECIFICATION

After completing the above fact-finding technique, the requirement is separated into three categories:

i) **Requirements that absolutely must be met**

a. Only the authenticate user is allow to use the system. Password and login ID need to be entered to gain access to the system.

b. The system should provide the functionality for the clerk to add in inventory items or new facilities in the school together with its detail as well.

c. The system should let the clerk to print out the report ac-hocly.

d. The system should let the School Principle view the attendance according to different criteria.

ii) Requirements that are highly desirable but not necessary

a. Tool Tips should have for every control inside the system.

b. Help files should be provided in every window.

c. Error messages with some guidelines should be provided to the users when illegal operations happen.

iii) Requirements that are possible but could be eliminated

a. The style of the words for the report can be changed according to the users.

b. The user can change the graphical design easily.

3.5 FUNCTIONAL REQUIREMENTS

Following are some of the functional requirement of the system.

i) Log In

a. Before users can access into the system, the user need to provide login ID and password for verification

b. The main screen will be difference based on the user right for the users.

ii) Password Setting

This button let the users to change their password.

iii) Error Message

Display error message to guide users along the operation of the system.

iv) Data Manipulation

Allow user to manipulate the data entered earlier. For example to delete or update records.

v) View Summary and Report

This button lets the administrator or clerk to print out the daily and monthly report. The administrator can view other record based on different criteria as request.

3.6 NON-FUNCTIONAL REQUIREMENTS

Following are some of the non-functional requirement of the system:

i) **Reliability**

Reliability is the extents to which a system can be expected to perform its intended function with required precision and accuracy. Thus, the system should be reliable in performing its daily functions and operations. For example, whenever a button is clicked, the system should be able to perform some functionality or generate some message to inform the user what is happening.

ii) **Scalability**

The scalability is to promise the capability of the system to migrate as a client or server to machines of greater or lesser power, depending upon requirements, with little or no change to underlying components. Database scalability issues can be resolved using distributed database architecture whereas web application scaling can be addressed by increasing bandwidth or by additional web servers.

iii) **Consistency**

Refers to any similar screen design or interface requires or process a similar action.

iv) **Usability**

The system should be developed in such as way that it is easy to use. It will enhance and support rather than limit or restrict the office processes.

v) **Security**

The system should be equipped with sufficient security. Each access by the user should be authenticated and validated by the system. The system should not show any potential of leakage of information. The password should be encrypted.

vi) **Data and Services Backup**

The system should be able to restore to its normal operation from any potential disaster. There should be a second backup for data and services to ensure the continuous of operation.

vii) **On-Time**

The system should be developed within the given time frame. In this period of time, all the requirement and also testing should be completed.

viii) **Flexibility**

The system should have the capability to take advantage of new technologies and resources. The system should be able to implemented in the changing environment.

3.7 DEVELOPMENT ANALYSIS

An analysis was carried out on the development tools to find out the most suitable tools for the system. These tools include the entire platform, servers, development software and programming language. Besides considering the suitability of the tools to the requirement, the tools used must be able to support each other. The following are the tools used in the system:

3.7.1 Visual Basic 6.0

Visual Basic 6.0 (VB6) is chosen as the development tool in this project because of the following reason:

a) VB6 is one of the most popular programming tools in windows environment due to its RAD (Rapid Application Development) capabity that associated with it.

b) VB6 is embedded with search engine (JET engine 1.0) that comes from the family that is similar to the internal search engine (JET engine 2.0) of DBMS used. Example, Microsoft Access. Both JET engines share the same database format and hence, an intermediate conversion program such as ODBC is not required.

c) Furthermore, VB6 uses an event driven approach to program the system and is not a procedural language. An application developed with an event-driven model response to event that happens in the computer environments. Such events include the pressing of mouse button or call function from another application that running concurrently.

d) Lastly, most important here is, mostly schools in Malaysia use Microsoft products that support VB6.

3.7.2 ODBC

Abbreviation of Open Database Connectivity, a standard database access method developed by Microsoft Corporation. The goal of ODBC is to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data. ODBC manages this by inserting a middle layer, called a database driver, between an application and the DBMS. The purpose of this layer is to translate the application's data queries into commands that the DBMS understands. For this to work, both the application and the DBMS must be ODBC-compliant -- that is, the application must be capable of issuing ODBC commands and the DBMS must be capable of responding to them. Since version 2.0, the standard supports SAG SQL.

3.7.3 Relational Database

Relational database is chosen because:

a) The data structure used by SFMS is relatively simple and thus, easy to be mapped into tables.

b) Since it is conceptually independent, no pointer or links are risible to the programmer, this ease the coding task.

c) The query language (SQL) is relatively simple and sufficient to implement SFMS system.

d) Queries may be expressed without the use of iteration or recursion. A block of data is directly return to DBMS.

Figure 3.2 shows the structure of a relational database.

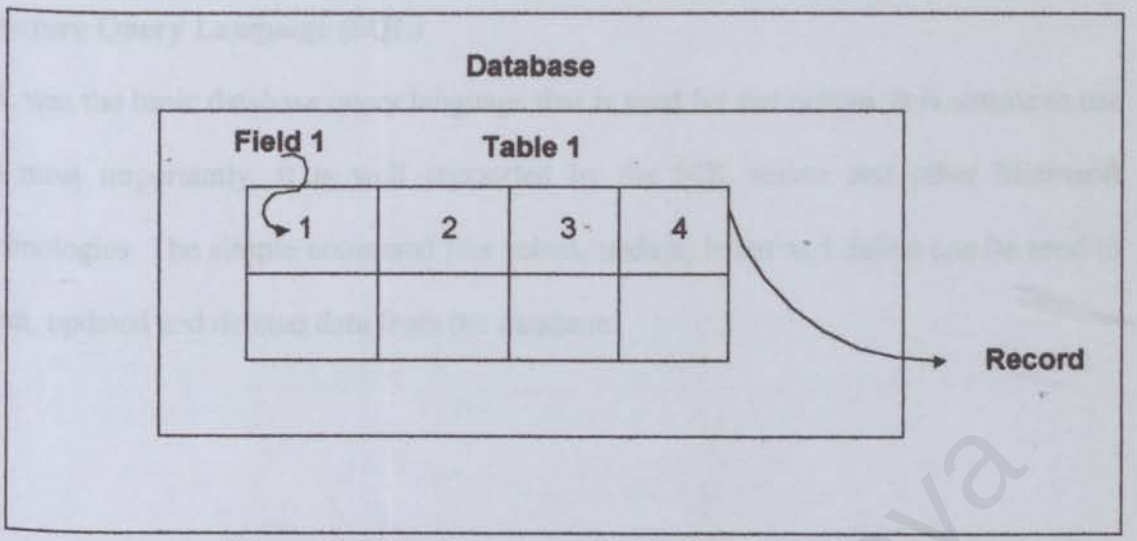

Figure 3.2 The Structure of Relational Database

3.7.4 Microsoft Access 2000

From the development technologies reviewed in Chapter 2, Microsoft Access 2000 is chosen as the database to store SFMS data. Features in Access 2000 is stated below:

a) Tables are the basic building blocks of databases, thy are where the actual data resides.

b) Forms create framework for representing of entering data in one or mire tables. In Access, forms also have special abilities for manipulating and verifying data not available at the table level.

c) Queries search and retrieve data from one or more tables based on entered criteria.

d) Reports are a way to output data from tables or queries. Reports can summarize data.

e) Macro are a simple way to coordinate operations in Access.

3.7.5 Others Concept Used

Structure Query Language (SQL)

SQL was the basic database query language that is used for the system. It is simple to use and most importantly, it is well supported by the SQL server and other Microsoft Technologies. The simple command like select, update, insert and delete can be used to select, updated and deleted data from the database.

3.8 RUNTIME ENVIRONMENT

The School Facilities Management System will be a stand-alone system that needed the following runtime environment:

3.8.1 Hardware Requirement

The hardware requirements for installing SFMS are summarized in Table 3.2. :

Component	*Description*
Microprocessor	IBM compatible machine with Pentium 133MHz processor or higher
RAM	> 16 MB RAM
Storage	40 MB of hard disk
Input Device	Mouse and keyboard
Video Monitor	EGA, VGA or compatible display

Table 3.2 Hardware Requirements

3.8.2 Software Requirement

The software requirements for installing SFMS are summarized in Table 3.3:

Component	*Description*
Operating System	Windows 95 or later in standard
Programming Language	Visual Basic Enterprise Edition 6
Database	Microsoft Access 2000

Table 3.3 Software Requirements

3.9 SUMMARY

This chapter includes all the analysis done to gather information about the functional and non-functional requirements for School Facilities Management System. In the following chapter, the database structure as well as the interface design of the system will be highlighted.

CHAPTER 4 SYSTEM DESIGN

4.1 INTRODUCTION

Design is the creative process or transforming the problem into a solution; the description of a solution is also called design. The goal of system design is to translate the requirements defined during the system analysis phase into a model or representation of an entity that will be built later. During this phase, quality is fostered.

The design of SFMS is mainly based on the 'top down' approach. Firstly, the database is designed, followed by the program design and finally the user interface design. Figure 4.1 shows the SFMS's context diagram. A context diagram is a top-level diagram that contains a single process where input sources and the output destination are acknowledged. This diagram is really a bird's eye view of data movement in the system and the broadest possible conceptualisation of the system. [13].

4.2 SYSTEM FUNCTIONALITY DESIGN

Diagram O

More detail than the context diagram permits is achievable by "exploding the diagrams". Input and outputs specify in the context diagram remain constants in this diagram. It may include more than 9 processes. The effect of drawing this diagram is that of taking a magnifying glass to view the original data flow diagram. Figure 4.2 shows the Level 0 of DFD for SFMS.

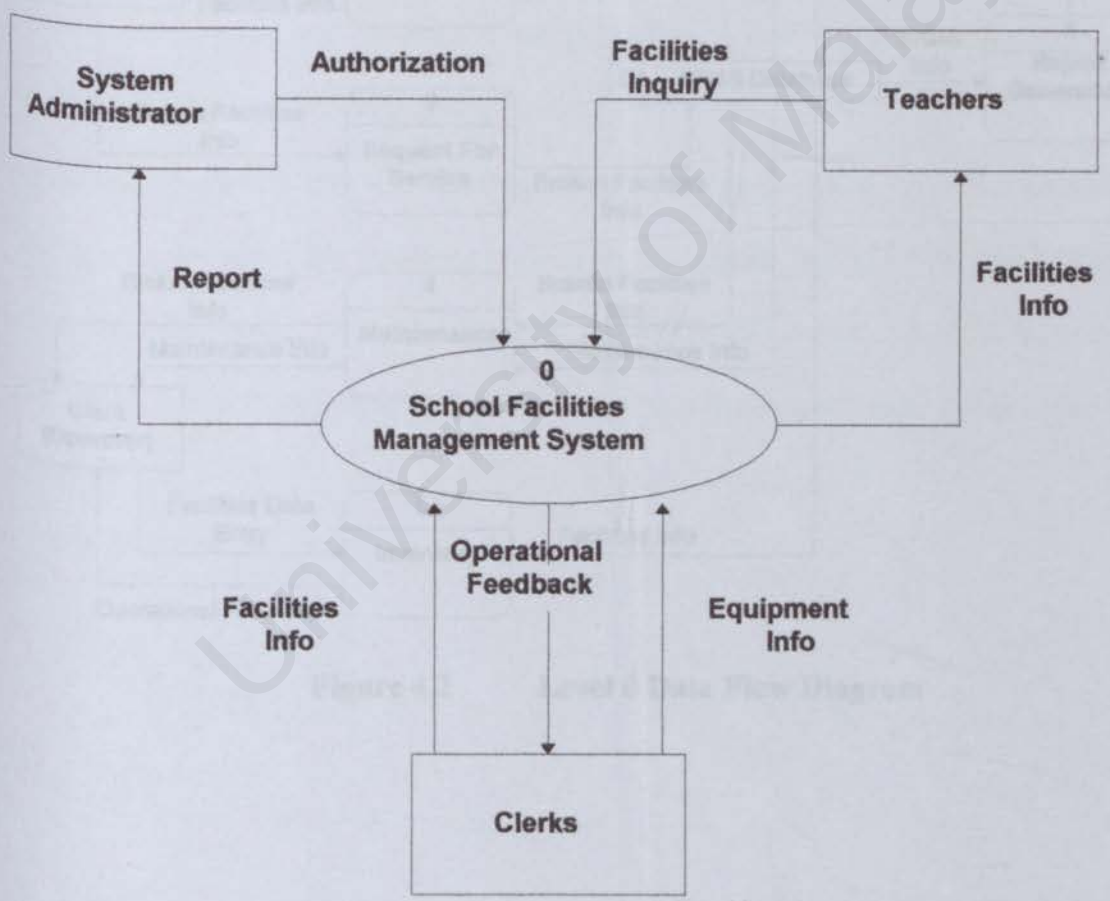

Figure 4.1 SFMS Context Diagram

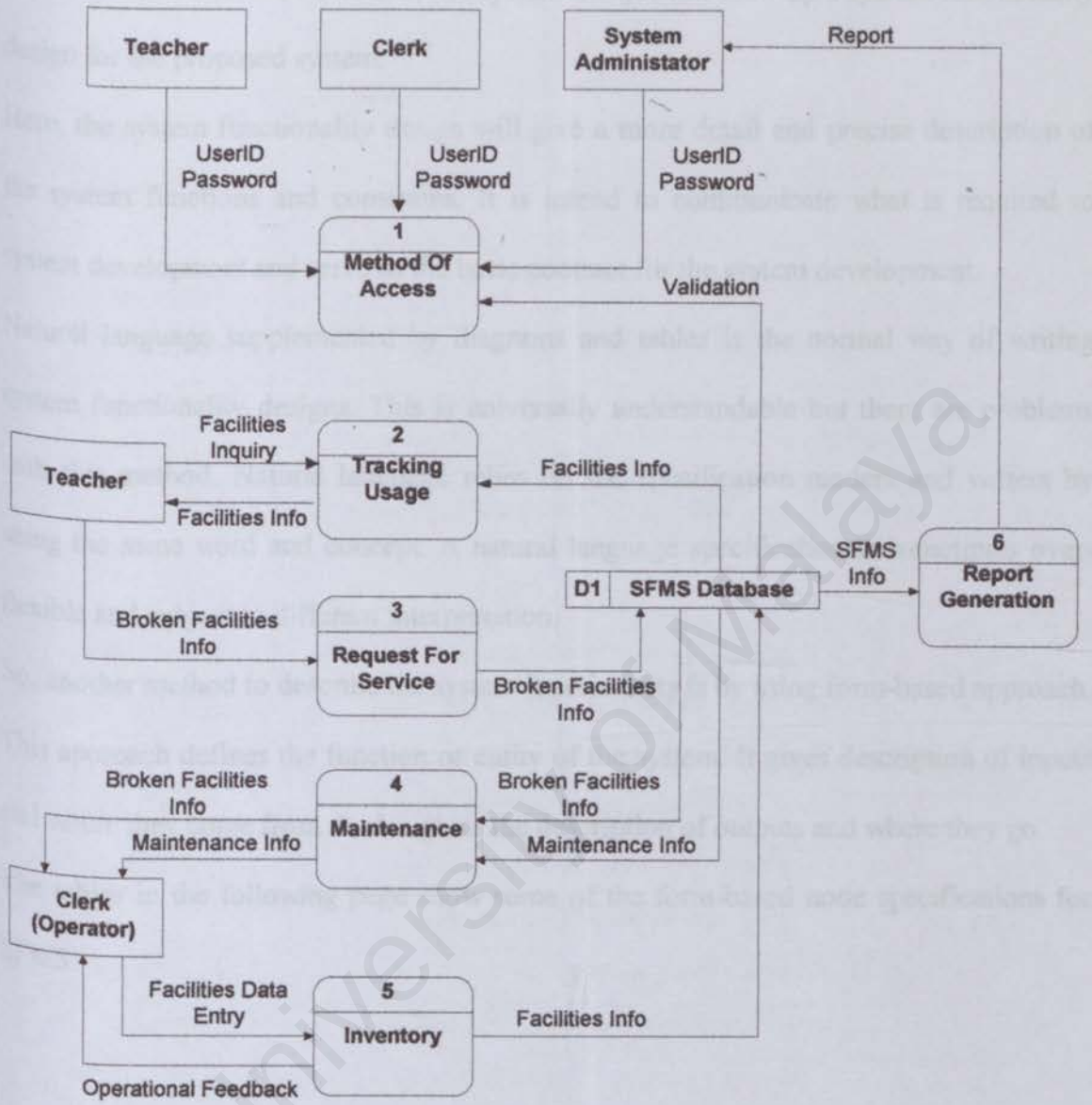

Figure 4.2 Level 0 Data Flow Diagram

System Functionality

Based on the Level 0 DFD above, the system analyst can draw up a system functionality design for the proposed system.

Here, the system functionality design will give a more detail and precise description of the system functions and constrains. It is intend to communicate what is required to system development and serve as the basic contract for the system development.

Natural language supplemented by diagrams and tables is the normal way of writing system functionality designs. This is universally understandable but there are problems with this method. Natural language relies on the specification readers and writers by using the same word and concept. A natural language specification is sometimes over-flexible and subject to different interpretation.

So, another method to describe the system functionality is by using form-based approach. This approach defines the function or entity of the system. It gives description of inputs and where they come from. It also gives the description of outputs and where they go.

The tables in the following page show some of the form-based node specifications for SFMS.

Function	**Method Of Access**
Description	Only valid user will be allowed to access SFMS.
Input	UserID and Password
Source	**cpr** Table
Output	Verification to access the modules in the system.
Destination	None
Requirement	User has been register as a valid user.
Pre-requisite	Splash Screen
Post-requisite	Other screens.

Table 4.1 Module Functionality for Method Of Access

Function	**Tracking Usage – By Building**
Description	To track usage of facilities in the school.
Input	Usage_name or Room_name respectively.
Source	**Building** Table, **Room** Table
Output	Building_Code, Building_Name, Room_Code, Room_Name
Destination	A screen to show the respective result.
Requirement	Data in the respective tables.
Pre-requisite	Tracking Usage Button
Post-requisite	Records already in the respective tables.

Table 4.2 Module Functionality for Tracking Usage-By Building

Function	**Tracking Usage – By Room**
Description	To track usage of facilities in the school.
Input	Room_Name respectively.
Source	**Room** Table, **Equipment** Table
Output	Room_Code, Room_Name, Equip_Code, Equip_Name, Equip_Type, Inven_No
Destination	A screen to show the respective result.
Requirement	Data in the respective tables.
Pre-requisite	Tracking Usage Button
Post-requisite	Records already in the respective tables.

Table 4.3 Module Functionality for Tracking Usage-By Room

Function	**Request For Service**
Description	To report broken equipment by the user.
Input	Building_Code or Room_Code or Equip_Code respectively.
Source	Report Table
Output	An operational message and a pop up message to tell the clerk of the malfunction.
Destination	None.
Requirement	The user as well as the equipment or the room or the building is already registered in the database.
Pre-requisite	None.
Post-requisite	None.

Table 4.4 Module Functionality for Request For Service

Function	**Maintenance**
Description	To keep information regarding when will be the best time to do maintenance for needed facilities.
Input	Building_Code or Room_Code or Equip_Code and Maintain_Desc, Maintain_Dur, Maintain_Date
Source	**Building** Table **orEquipment** Table or **Room** Table and **Maintenance** Table.
Output	A pop up message box to tell user it is the time to do maintenance.
Destination	**Maintenance** Table
Requirement	None.
Pre-requisite	The particular record is in the database.
Post-requisite	None.

Table 4.5 Module Functionality for Facilities Maintenance

Function	**Inventory**
Description	To keep inventory record on building, room and equipments in the school.
Input	Respective details.
Source	Respective details.
Output	An operation message.
Destination	**Building** Table, **Room** Table, **Equipment** Table
Requirement	The need to add new record.
Pre-requisite	None.
Post-requisite	None.

Table 4.6 Module Functionality for Inventory

4.3 DATABASE DESIGN

A database is an integrated collection of logically related data stored in different types of records, and in a way that makes them accessible for multiple applications.

Database design involves identifying the user data requirements and determining how these data should be structured from these requirements. It transformed the unstructured information and the processing requirements of an application into representations that define the functional specifications [14]. The database model used for this system is the relational model.

In the following section, the normalisation process in designing the database will be highlighted.

4.3.1 Normalisation

Normalisation is the technique used either to convert a large database into a relational database or create a stable, well-formatted database from scratch. Data normalisation provides rules on how to break tables or field into several tables that has references to each other. Data normalisation seeks to minimize duplication of data within the database by logically dividing a large table into several smaller tables. This process of normalisation makes information is more differentiated and usable. The main purposes of normalisation are to reduce data redundancy and to eliminate data inconsistency. A properly normalised database will not only save storage but also minimize the need to modify data relations if the system is later expended or upgraded.

The steps below describe the basic normalisation process.

Step 1: First Normal Form (INF)

This step involves removing the entire repeating group and identifying the primary key(s).

Step 2: Second Normal Form (2NF)

The second step involves removing all partial functional dependencies by splitting the original relation into more relations. The relation is said to be in 2NF is the relation that is in 1NF and every non-key attribute is fully functional dependent on the primary key.

Step 3: Third Normal Form (3NF)

The third involves removing any transitive dependencies where non-key attributes

are dependent on another non-key attributes. The relation is said to be in 3NF if the relation is in 2NF and there is no transitive dependency.

4.3.2 Data Dictionary

As most of us know, the volume of data in most of applications is substantial (more than a single analysts can easily keep track of). When teams of analysts work on a system, the task of coordinating data definition becomes more complex. Therefore, a data dictionary has to be developed in order to let system analysts and programmers to keep track of data definition used in the system. Individuals depend on the definitions other established and the assumption they made about data specification. A data dictionary is a repository of elements in a system. As the name suggest, these elements centre on data and the way they are structured to meet user requirements and organisational needs. In a data dictionary, a list of all the elements composing the data following a system can be found. The major elements are data flows, data stores and process. The data dictionary stores details and description of these elements [10].

There are 5 reasons why data dictionary are important.

i) To manage the detail in a large system.

ii) To communicate a common meaning for all system elements.

iii) To document the features of the system.

iv) To facilitate analysis of the details, in order to evaluate characteristic and determine where system changes should be made.

v) To locate errors and omission in the system.

The following are the full specification of each table in SFMS database.

Table Name :	Building	
Description :	Contains information on the buildings in school.	
Column Name	**Data Type**	**Description**
*Building_Code	Text	Building code.
Building_Name	Text	Actual name of the building.
Building_Desc	Text	Special description of the building.
Schedule_Flag	Yes/No	To see whether the Building already been schedule for its maintenance.

Table 4.7 Building Table

Note: * Primary Key

Table Name :	Room	
Description :	Contains information on rooms or special rooms in the school.	
Column Name	**Data Type**	**Description**
*Room_Code	Text	Room code
Room_Name	Text	Actual name of he room.
Room_Capacity	Number	Sitting capacity of the room.
Building_Code	Text	Building where the room are located.
Room_Desc	Text	Special description of the room.
Usage_Code	Text	Usage code.
Schedule_Falg	Yes/No	To see whether the Building already been schedule for its maintenace.

Table 4.8 Room Table

Note: * Primary Key

Table Name : Equipment		
Description : Contains information on equipments in school.		
Column Name	**Data Type**	**Description**
*Equip_Code	Text	Equipment code.
Equip_Name	Text	Equipment name.
Inven_No	Text	Inventory code given to the equipment.
Type_Name	Text	Name of the Equipment Type
Room_Code	Text	Location of the equipment.
Date_Purchase	Date/Time	Date of purchase.
Equip_Price	Currency	Price of the Equipment
Ven_code	Text	Vendor code
Schedule_Falg	Yes/No	To see whether the Equipment already been schedule for its maintenance.
Deleted_Flag	Yes/No	To see whether the record of the Equipment already been deleted r not.
Usage Code	Text	To see whether the Equipment is common or not, B for typical equipment like tables and chairs and K for special equipment like projektor.

Table 4.9 Equipment Table

Note: * Primary Key

Table Name :	Vendor	
Description :	Contains information on the vendors who sell equipments to the school.	
Column Name	**Data Type**	**Description**
*Ven_Code	Text	Vendor's code
Ven_Name	Text	Vendor's name
Con_First_Name	Text	Vendor's first name.
Con_Last_Name	Text	Vendor's last name.
Con_Title	Text	Position of the contact person, for example Manager, Sales Person, etc.
Ven_Addr1	Text	Vendor's address 1
Ven_Addr2	Text	Vendor's address 2
Ven_Addr3	Text	Vendor's address 3
Ven_Postcode	Number	Postcode
Ven_Telno	Number	Telephone number.
Ven_Faxno	Number	Fax number.
Ven_Email	Text	E-mail address.
Ven_Desc	Text	Other's description about the vendor

Table 4.10 Vendor Table

Table Name :	Equip_Type	
Description :	Contains information on types of equipment in school.	
Column Name	**Data Type**	**Description**
*Type_Code	Text	Type code that will be use to generate inventory no.
Type_Name	Text	Equipment type name.

Table 4.11 Equip_Type Table

Note: * Primary Key

Table Name :	Usage	
Description :	Contains information on the usage of every room in school.	
Column Name	**Data Type**	**Description**
*Usage_Code	Text	Usage code.B, K
Usage_Name	Text	Name of the usage,B-Biasa, K-Khas-

Table 4.12 Usage Table

Table Name :	Report	
Description :	Contains information on request for service for broken facilites..	
Column Name	**Data Type**	**Description**
Report_ID	Auto Numbew	An ID to uniquely identify the report done.
Report_Date	Date/Time	Date the facility was reported malfunction
UserID	Text	To keep track of who report the malfunction
Report_Desc	Text	Description of the malfunction of the particular equipment
Location_Name	Text	Place of the equipment.
Fixed	Yes/No	To see whether this record has been fixed or not.
System_time	Date/Time	To keep the current time the report being made.
Fac_Type	Text	To see whether the facility is type of Building, Room, or Equipment
Report_Code	Text	Code of the facility.
Spoiled	Yes/No	Keep see whether the facility is permanently spoiled.

Table 4.13 Report Table

Note: * Primary Key

Table Name :	Maintenance	
Description :	Contain information on scheduled maintenance.	
Column Name	**Data Type**	**Description**
*Maintenance_code	Text	Code of the facility that is being maintained.
Maintain_Date	Date/Time	Date of the future maintenance.
Location_Name	Text	Location of the facility
Maintenance_desc	Text	Description of the particular maintenance.
Fac_Type	Text	To see whether the facility is type of Building, Room, or Equipment
Maintenance_Dur	Text	How frequent the maintenance need to made.

Table 4.14 Maintenance Table

Table Name :	Duration	
Description :	Contain information on duration of maintenance	
Column Name	**Data Type**	**Description**
*Maintenance_Dur_ID	Auto Number	ID to uniquely identify the duration.
Maintenance_Dur	Text	Name of the duration i.e. Once a year, once a month, etc.

Table 4.15 Duration Table

Table Name :	cpr (done by Lok Farn)	
Description :	Change Password Record	
Column Name	**Data Type**	**Description**
* cpruid	Text	User identity key
cprupw	Text	User password

Table 4.16 cpr Table

Note: * Primary Key

Table Name :	upf (done by Lok Farn)	
Description :	User Personal File - store user details	
Column Name	**Data Type**	**Description**
* upfuid	Text	User identity key
upffnm	Text	User full name
upfuic	Text	User identity card number
sexcod	Text	User gender
upfhpn	Text	User phone number
upfadd	Text	User address
utfutn	Text	User type name
upftdd	Date/Time	The year the user register
upfdel	Yes/No	Deleted flag
poscod	Text	User post in school
depcod	Text	User department

Table 4.17 upf Table

Table Name :	utf (done by Lok Farn)	
Description :	User Type Facility Table - store the user type of facility.	
Column Name	**Data Type**	**Description**
*utfutc	Text	User type code, A, T, S
utfutn	Text	User type name A- Administrator, T-Teacher, S-Staff

Table 4.18 utf Table

Note: * Primary Key

4.4 GRAPHICAL USER INTERFACE DESIGN

A user interface is the part of an application that the user sees and interacts with. It relates to, but not as same as, the underlying structure, architecture, and code that makes the software works [2]. The interface includes screens, windows, control, menu, online help, and others. Anything that user sees and interacts with is part of interface.

So, the goal of interface design is to help users get information they need in and out of the system by addressing the following objectives:

i) Effectiveness as achieve through design interfaces that allow users to access the system in a way that is congruent with their individual needs.

ii) Efficiency as demonstrated through interfaces that increase speed of data entry and reduce errors.

iii) User consideration as demonstrated in designing suitable interfaces and providing appropriate feedback to users from the system.

Frustration and anxiety are part of daily life for many users of computerized information system. Users struggle to learn command language or menu selection systems that are supposed to help them do their job. Some people encounter such serious cases of computer shock, terminal error, or network neurosis that they avoid using computerised systems. After suffering from these frustrations, they are some users who finally decided to go back to their manual method.

The above statements show that user interface which is the doorway into an interactive software application, demand an understanding of human factors and

interface technology. Therefore, the system being designed should take into consideration the skill level and behaviour of users.

There are two categories of the user interface design guidelines that can be followed. They are general interaction and information display.

4.4.1 General Interaction

i) Be consistent

Used a consistent format for menu selection, command input, data display and the myriad other functions that occur in an user interface.

ii) Offer meaningful feedback

Provide the user with visual feedback to ensure that two-way communication (between the user and interface) is established. For instance, the mouse pointer will chance its shape to an hourglass shape to visually inform the user that the system is now busy in process.

iii) Ask for verification of any non-trivial destructive action

Always prompt user for actions such as deletion of a record before the action is carried out. Before an item or user is deleted from the database, a message box will appear in front of the user for confirmation.

iv) Forgive mistake

The system should protect itself from user's errors that might cause it to fail.

4.4.2 Information Display

i) Display only information that is relevant to the current context

The user should not have to wade through extraneous data, menu and graphics to obtain information to a specific system function. For example, only needed user information is displayed in the user list screen.

ii) Use consistent labels, standard abbreviation and predictable colours

The meaning of a display should be obvious without reference to some outside source of information. In SFMS, a standard and consistent command button for closing a screen is used. This command button has the same caption and icon on it in every screen that it appears.

iii) **Inactivated commands that are inappropriate in the context of current**

This is to prevent the user from attempting some action that could result in error. If the user does not have access rights to some commands or buttons, the system will hide it from their view.

4.4.3 Menu Design

A menu interface provides the user with an on-screen list of available selection. In responding to the menu, user is limited to the options displayed. The user need not know the system but does need to know what task should be accomplished.

The following figure (Figure 4.3) and table (Table 4.16) gives the description of the menu bar in SFMS.

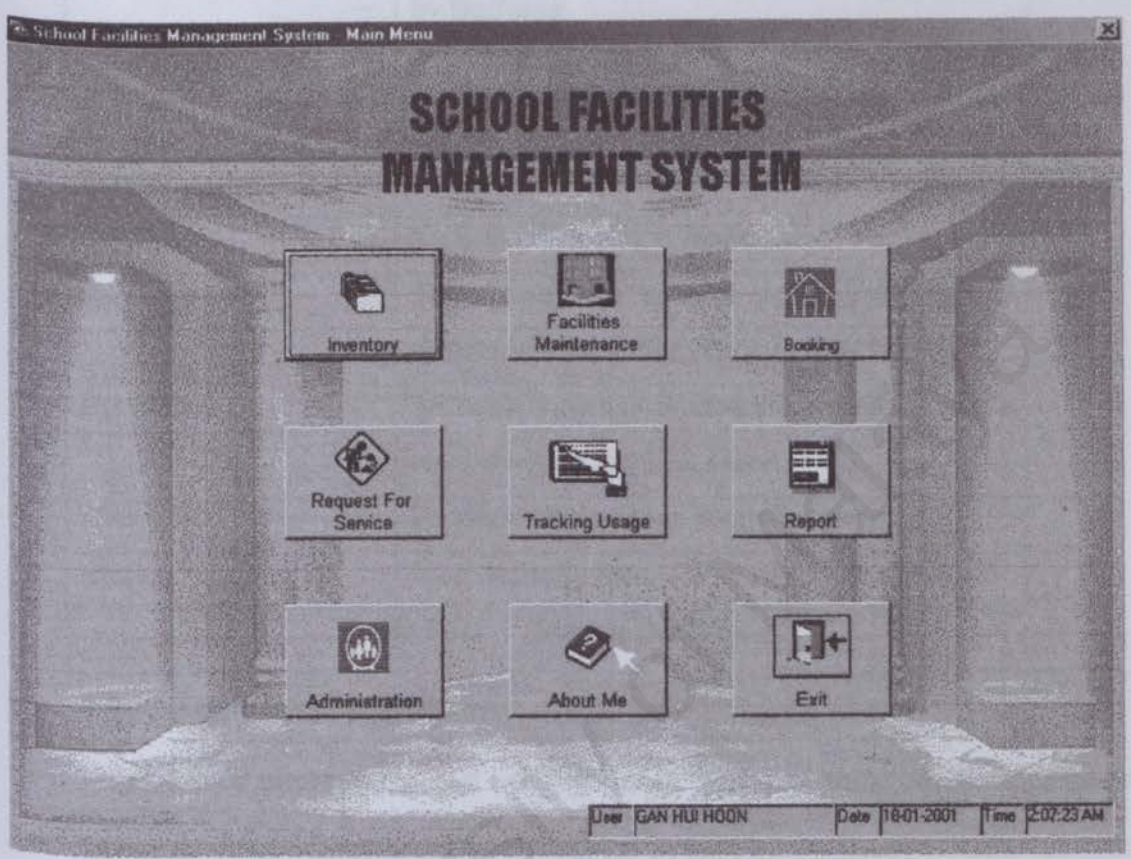

Figure 4.3 Main menu interface

Button	Description
Tracking Usage	To track usage of facilities either by Room or by Building.
Facilities Maintenance	To scheduled maintenance of facilities.
Request For Service	To report broken facilities..
Report	To generate report for school administrators.
Inventory	To keep inventory records.
Administration	To add in new user or delete user by the system administrator (done by Lok Farn).
Booking	To book rooms or equipments by the school's teachers (done by Lok Farn).
About Me	To show information about the system.
Exit	To exit the system.

Table 4.16 Button Description

4.4.4 Login Screen Design

Login screen is where the user need to key in their user name and their password so that they can gain accesses to the system as well the database.

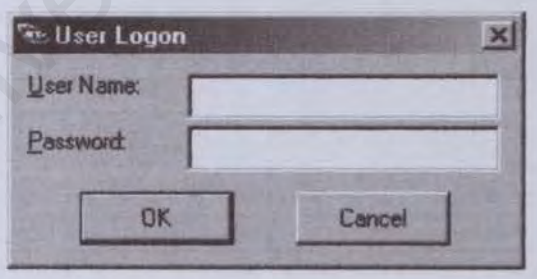

Figure 4.4 Login screen

4.4.5 Miscellaneous Screen Designs

Below are others screen designs for the SFMS. They are generated based on the guidelines and description above.

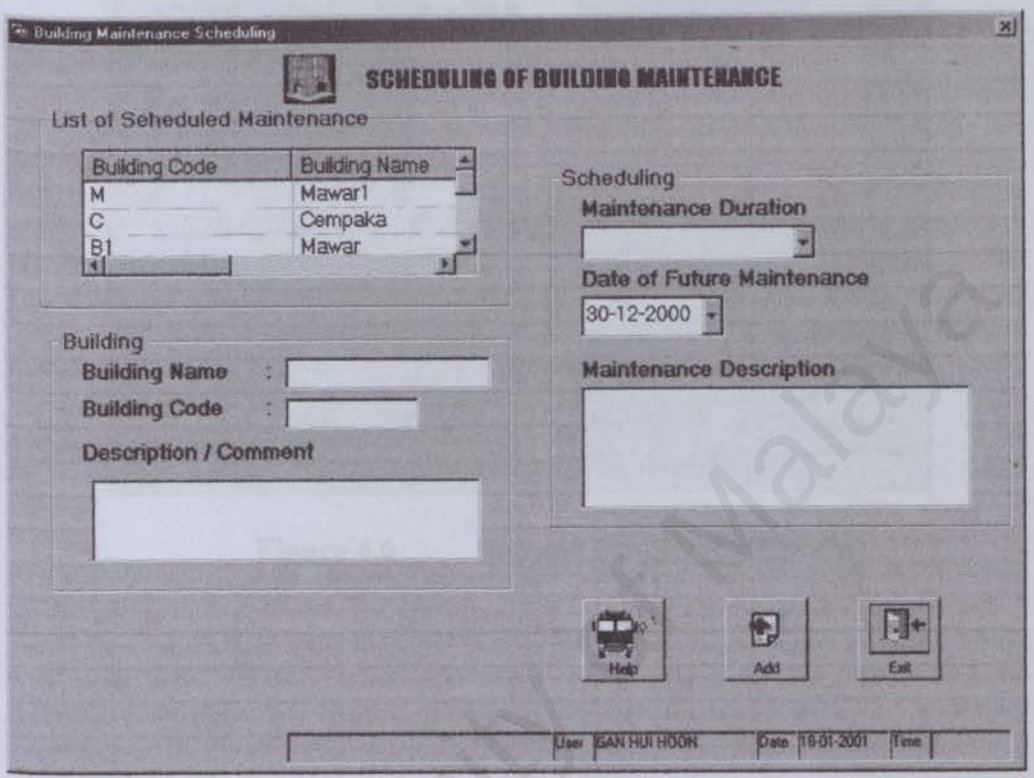

Figure 4.5 Facilities Maintenance For Building Screen

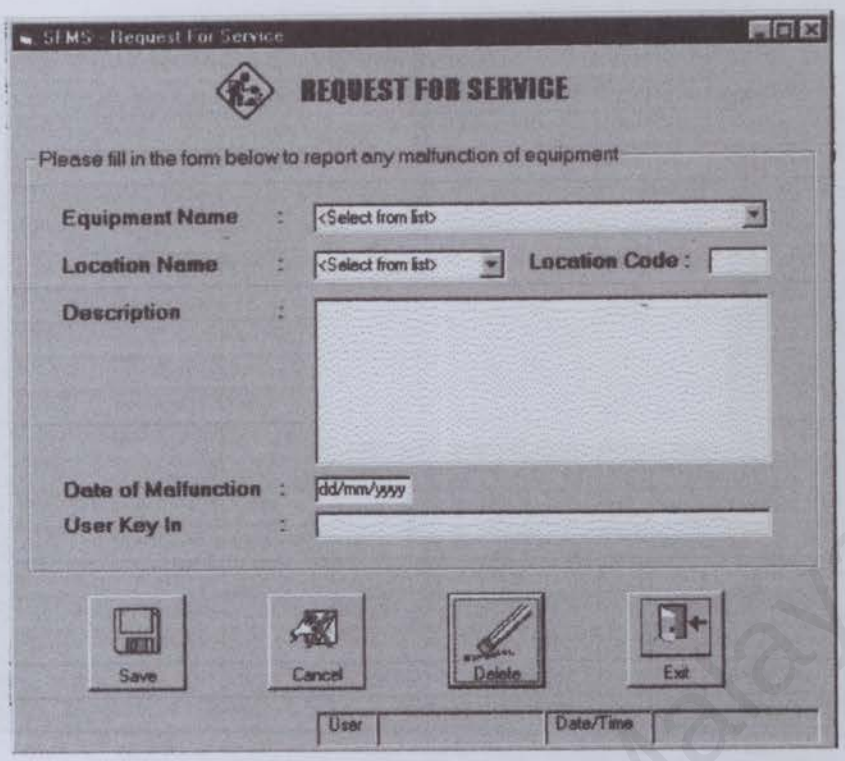

Figure 4.6 Request For Service Screen

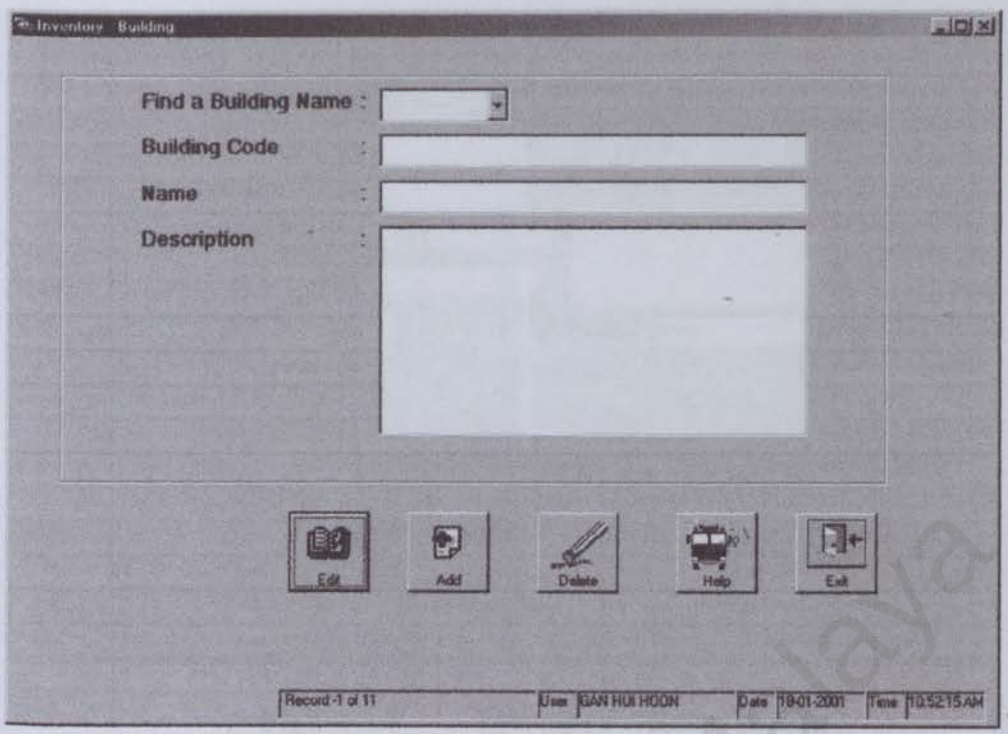

Figure 4.7 Inventory – Building Screen

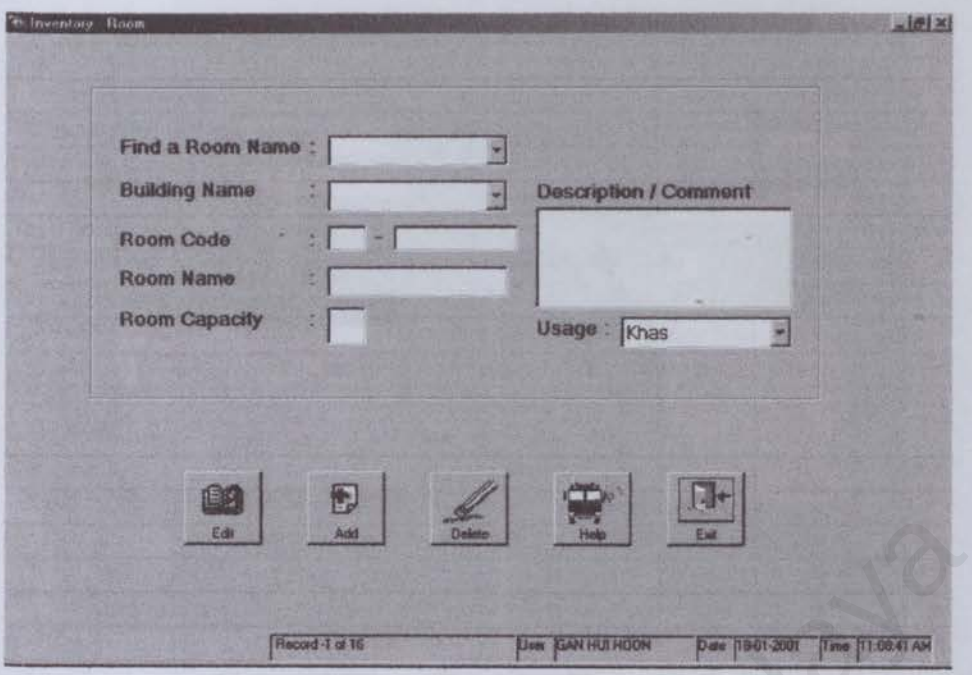

Figure 4.8 Inventory – Room Screen

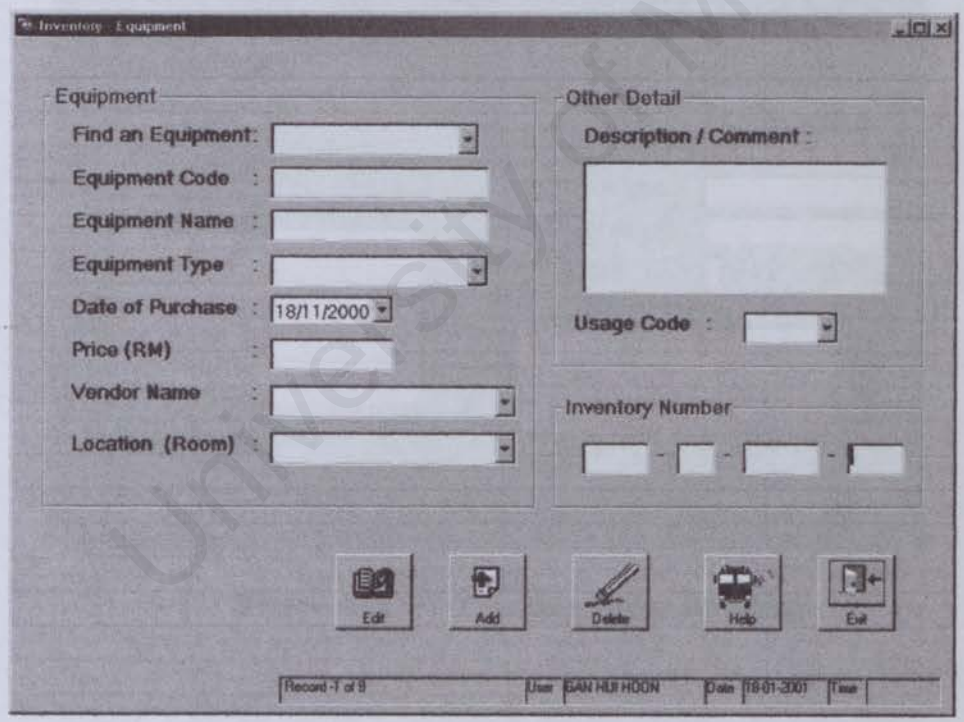

Figure 4.9 Inventory – Equipment Screen

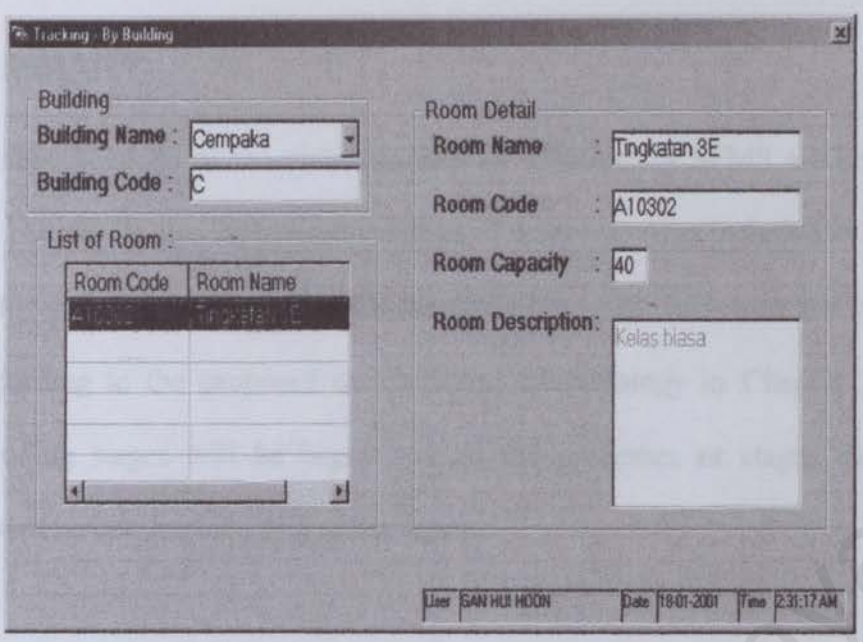

Figure 4.10 Tracking Usage by Building

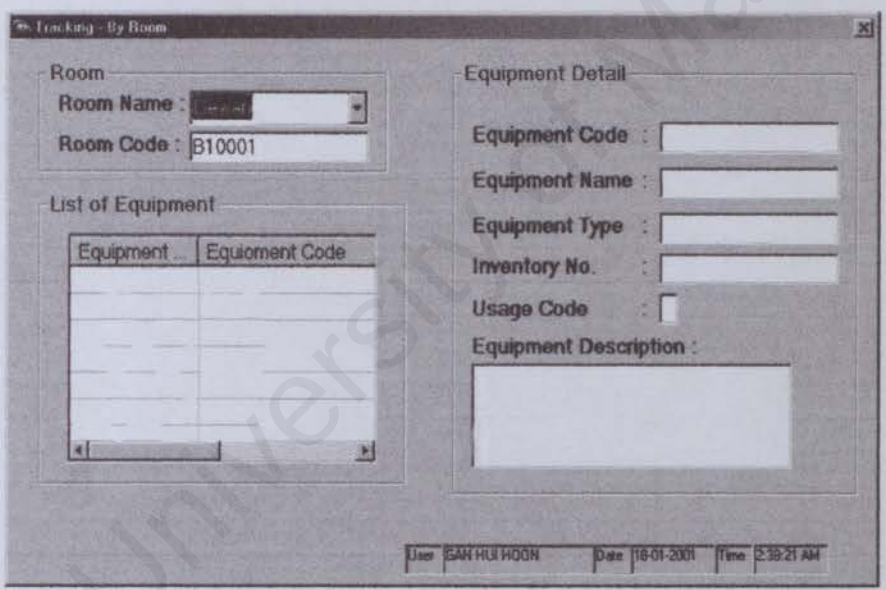

Figure 4.11 Tracking Usage By Room

4.5 SUMMARY

Functional specifications of the main modules or function for SFMS are being defined here in this chapter. It also includes the design of database. Also included in this chapter is the graphical user interface of the main functions. And this ends the proposal for SFMS. According to the proposed development methodology in Chapter 2, later this year, the coding stages will be begin and all the processes or stages that followed afterwards will be documented in another report.

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

System implementation is the physical realization of the database and application design. On completion of the design stages (which may or not may not have involved prototyping), here comes the stage where the database and the application have to be implemented. The transaction process continues when a compiler accepts source code as input and produces machine-dependent object-code as output. Compiler output is further translated into machine code – the actual instruction that drives micro-coded logic in the central processing unit (CPU).

Coding and debugging are the major works involved in the implementation phrase. So, coding mythology, documentation and testing will later be covered in this chapter or in later chapters.

5.2 CODING

Program written must be able to implement the design. This task can be daunting, for several reason. First, the designers may not have addressed all of the idiosyncrasies of the platform and programming environment; structure and relationship that are easy to describe with charts and tables are not always straightforward to write as code. Second, a programmer must write out code in a way that is understandable. Third, the programmer must take advantage of the characteristic of the design's organization, the data structure, and the programming language's constructs while still creating code that is easily reusable.

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

System implementation is the physical realization of the database and application design. On completion of the design stages (which may or not may not have involved prototyping), here comes the stage where the database and the application have to be implemented. The transaction process continues when a compiler accepts source code as input and produces machine-dependent object-code as output. Compiler output is further translated into machine code – the actual instruction that drives micro-coded logic in the central processing unit (CPU).

Coding and debugging are the major works involved in the implementation phrase. So, coding mythology, documentation and testing will later be covered in this chapter or in later chapters.

5.2 CODING

Program written must be able to implement the design. This task can be daunting, for several reason. First, the designers may not have addressed all of the idiosyncrasies of the platform and programming environment; structure and relationship that are easy to describe with charts and tables are not always straightforward to write as code. Second, a programmer must write out code in a way that is understandable. Third, the programmer must take advantage of the characteristic of the design's organization, the data structure, and the programming language's constructs while still creating code that is easily reusable.

The coding methodology used in the development of this system is the top-down and bottom-up approach. By combining both approaches at different stages of coding, testing could be done on those completed modules while others are still being coded. Example of the coding can be found in Appendix C. Comments are added in the code to provide a better understanding to the code.

5.2.1 Top-down approach

This approach allows the higher-level modules to be coded first before the lower level modules. The codes in the lower modules contain only an entry and an exit. A module with such characteristic is called a shell. The higher-level modules will reference the lower ones if they are coded and available. Reference to a shell will result in an empty action.

This approach will ensure that the most important modules will be developed and tested first. It also gives a preliminary version of the system sooner.

Below is a diagram that describes the top-down approach for SFMS.

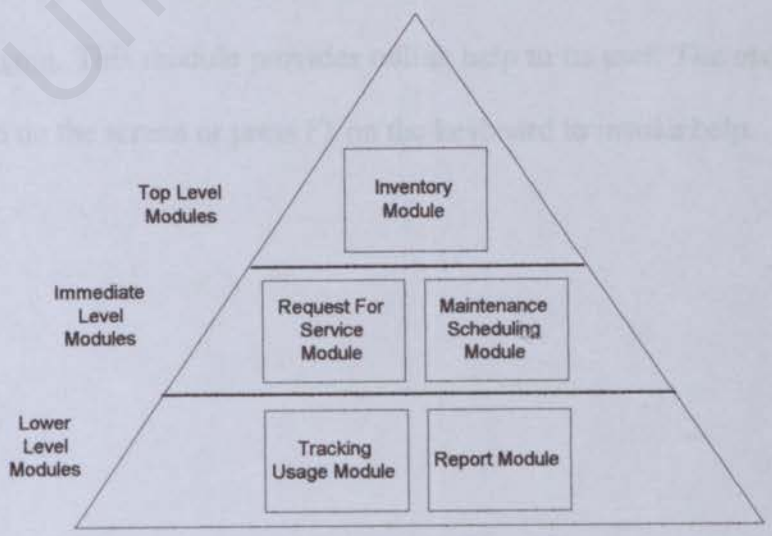

Figure 5.1 Top-down approach in SFMS

5.2.2 Bottom-up approach

As oppose to the top-down approach, the bottom-up approach begins with the coding of the lower level modules first before the higher level modules. However, the higher modules are just skeletons that call the lower modules. This approaches is used if the critically of lower level modules is high and need to be completed first.

5.3 MODULE IMPLEMENTATION

The School Facilities Management System is divided into 7 main modules, which are the Inventory Module, Facilitis Maintenance Scheduling Module, Request For Service Module, Booking Module, Report Module, and Administrator Module. Among these modules, Booking and Adminitrator are being develop by Lok Farn. Each module is developed using Visual Basic 6.0. (Chapter 3 provide more explanation of Visual Basic 6.0)

However after finishing coding of these module, another module is added that is the Help Module. This particular module is developed using ForeHelp Premier 2001 Demo Version by ForeFront Boulder. The software was dowmloaded from www.forehelp.com. This module provides online help to its user. The user may press the Help button on the screen or press F1 on the keyboard to invoke help.

5.4 SUMMARY

Under this stage, the design model of the School Facilities Management System are being into a workable product. Stages involved are coding and modules implementation.

In the next chapter, the testing of the final product are highlighted. It will show types of testing approaches used.

CHAPTER 6 TESTING

6.1 INTODUCTION

Testing is the process of executing the application programs with the intent of finding errors. Before going live, the newly developed database application system should be thoroughly tested. This is achieved using carefully planned test strategies and realistic data so that the entire testing process is methodicaclly and rigorously carried out. In fact, testing cannot show the absences of faults, it can show only that software fault. To quote Deutsch [DEU79],

The development of software system involved a series of production activities where opportunities for injection of human fallibilities are enormous. Errors may begin to occur at the very inception of the process where the objectives... may be erroneous or imperfectly specified, as well as [in] later design and development stages... Because of human inability to perform and communicate with perfection, software development is accompanied by a quality assurance activity.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation.

In this chapter, software testing fundamentals, testing strategies and software debugging methods will be presented.

Following are some of the objectives of software testing:

i) Testing is a process if executing a program with the intent of finding an error.

ii) A good test case is noted that has a high probability of finding an as-yet-undiscovered error.

iii) A successful test is one that uncovers as-yet undiscovered error.

6.2 Testing Technique

To test a component, a range of inputs and conditions are chosen. The component of the software will be allowed to manipulate the data, and the output will be observed. A particular input is chosen will demonstrate the behavior of the code behind the entire GUI. A test point or a test case is a particular choice of input data to be used in testing program. However, the data are entered with the express intent of determining whether the system will process them correctly.

Different test cases are needed on different type of testing strategies. As there is no available end-user, artificial test data that resembles the actual facilities in school are created. There are three categories of test cases that are created for testing purposes namely erroneous test, normal test data, and extreme test data. These categories are further explained in the following section.

6.2.1 Erroneous test

Using test data that are erroneous is a good way to determine how the system handles such errors and how it behaves under such situation. For example, an invalid user will not be allowed to enter SFMS until he or she gives a right combination of username and password. For this case, the system will prompt the user "invalid username or password". Therefore, an invalid username or password is use as erroneous test data.

6.2.2 Normal test data

The normal test case is use to check whether the system will work well under normal condition. One example of the normal test data is shown below.

Assume there are only three equipments in a room. There are another four equipments in another room. So, if a user want to print a report on total equipments in the school, the system will pick up seven records in two rooms. The expected results can be compared both on the screen as well as on the hard copy.

This type of test data serves as a preliminary test of the system.

6.2.3 Extreme test data

The extreme test data is use for exhaustive testing. The test data can be very huge or it can be at or beyond the boundary value. Lets take another example.

The system does not allow the user to choose the date of a malfunction of a facility bigger than the current system date. This is because, how can a user know that the facility will be malfunction on a date in the future?

6.3 TESTING STRATEGY

A strategy use to test this system (SFMS) is actually a series of four steps that are implemented sequentially. The steps are shown in figure 18.2. Initially, tests focus on each component individually, ensuring that it functions properly as a unit. Hence, the name *unit testing*. Unit testing makes a heavy use of white – box testing technique, exercise specific paths in a module's control structure to ensure complete average and maximum error detection. Next, components must be assembled or integrated to form the complete software packages. Integration testing addresses the issues associated with the dual problems of verification and program construction. Black-box test case design techniques are the most prevalent during integration, although limited amount of white-box testing maybe use to ensure coverage of major control path. After the software has been integrated (constructed), a set of high-order tests are conducted.

Validation criteria (established during requirement analysis) must be tested. Validation testing provides final assurance that software meets the functional, behavioral, and performance requirement. Black-box testing techniques are used exclusively during validation.

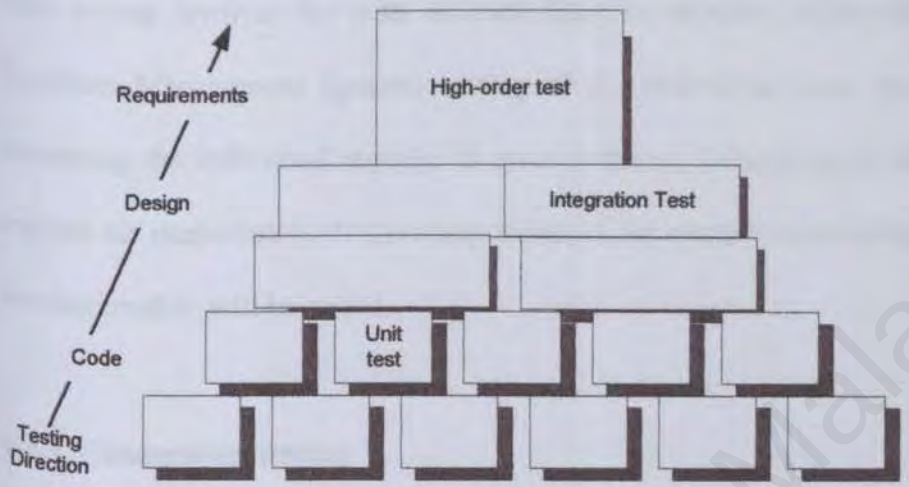

Figure 6.1 Software testing steps

The last high-order testing step falls outside the boundary of software engineering and into the broader context of computer system engineering. Software, once validated, must be combined with other system elements (e.g., hardware, people, databases). System testing is an example of high-order test, verifies that all system elements mesh properly and that overall system function/perfomance is achieved.

6.3.1 Unit testing

Historically, quality software is relied on testing each function or module. Unit testing is sometimes referred as Module Testing or Component Testing, which is extremely time-consuming. For SFMS, unit testing was done during the coding phrase. The first step is to examine the program code by reading through it, trying to faulty algorithms or syntax faults.

The process is followed by comparing the code with specifications and with the design to make sure that all relevant cases have been converted into the desired output.

Unit testing involves the tests on each function module independently. In School Facilities Management System, testing of the individual class module is merely compiling the individual module. If error is found, debugging of the codes will be carried out immediately. If the compilation of the module is completed successfully, another module will be coded.

6.3.2 Integration testing

Testing a specific feature together with other newly developed feature is known as integration testing. In other words, when the individual components are working correctly and meet the objectives, these components are combined into a working system. Testing the interface of 2 components explores how components interact with each other.

Incremental integration approach was applied during the developments of the system. The system was constructed and tested in small arguments, where errors were easier to isolate and correct. Error will be corrected before processing to the next integration.

If all individual modules passed the Module Testing successfully, there will be no bugs in the Module Integration Testing. The motive behind this testing is to make certain that all modules can be executed as a complete module. As mentioned earlier, an individual module calls other module to perform certain tasks. Parameters will be

passes among these modules and if not tested, then parameter may be passed incorrectly.

6.3.3 System testing

System testing is a series of different tests conducted to verify that all system elements such as hardware, software and information, have been properly integrated and performed allocated function. Below are some of the testing done:

i) Performance Testing

System performance is measured against the performance objectives set by the user as expressed in the non-functional requirement. Performance testing examines the speed of response to user command, accuracy of the result, and accessibility of the data are checked against user's performance prescription.

ii) Platforms testing

Platform testing involves the testing of the School Facilities Management System on the different platforms. Different platforms mean different operating system (Windows 98, Windows NT, Windows 2000 and Windows ME). These tests are done in of the laboratory in FSKTM.

6.4 SUMMARY

From the testing process that has been carry out, it can summarized the test result as follow:

i) Achieve the main objective of the project

Generally, the main objectives of the project as described earlier have been achieved. The system can handle and maintain the facilities in a particular school.

ii) Syntax of language

The language use in this system to deliver information to the user should me more descriptive and not too technical. Unlike the programmer, the user of SFMS will not have much computer knowledge, so they would not be able to understand computer syntax as programmer do.

CHAPTER 7
SYSTEM EVALUATION AND CONLUSION

7.1 INTRODUCTION

After all the handwork of designing and developing as well as implementing SFMS, the end product of the project is brought up for evaluation. They are many evaluation techniques that use to evaluate the final system. The following section will explain in detail about the system strength and its limitation.

7.2 PROBLEMS FACED AND SOLUTIONS

The following are the major problems encountered from the beginning of the project through the end of the system development process.

i) **Difficulties in choosing a programming language and tools**

There are some well-known programming tools available in the market that can be use to develop a similar database management system. (Please refer to Chapter 2). Choosing a suitable tool was a critical process as all tools have tier strength and weakness. In addition, the availability of the required tool for development was also a major concern.

Solution:

There is no latest version of Power Builder 7.0 available in FSKTM. So, I have to choose Microsoft Visual Basic 6.0.

ii) **Difficulties in gathering information**

Not much information could be gathered from the questionnaires that were distributed to the school. This is due to from 5 questionnaires sent to schools (with the help from Puan Abrizah), only one responded.

Solution:

Obtain more information and ideas through brainstorming with project supervisor, who is a user of facilities herself. Ideas are also gathered from other course mates. Their ideas are also useful because we all used to be users of facilities in schools.

iii) No end user evaluation

Since there is no really end user to test SFMS except evaluation from Puan Abrizah, I do not really know how end-user will react to SFMS.

Solution:

Tests were done on other course mates.

iv) Slow processing time

As very 4GL programmer knows, Visual Basic 6.0 is a graphical-oriented programming language. As such, application created using this language is also graphical-oriented and thus more memory is required to compile and execute the application.

Solution:

Although it is documented that the minimum memory requirement in developing SFMS is 32 MB, but personally a memory of 128 MB is much preferred.

v) Transferring SQL Statement from VB to Crystal Report

Syntax the SQL Statements in VB6 and the syntax in Crystal Report differ from each other. So, I cannot use the knowledge of my SQL statement that I learn from VB to Crystal report. This make quite hard for me to pass the selection criteria from VB6 to Crystal Report.

Solution:

Referring to online tutorial about Crystal Report and read about Crystal Report from e-books and others web-publish tutorials.

7.3 SYSTEM STRENGTHS

i) Security features

There are 3 types of users in SFMS: the Administrator, the Teacher, and the Staff. Each user type is allocated certain access rights to the functions in SFMS. For example, only the Administrator is allowed to create others user. Therefore, functions that are restricted to the user will be disabled. By incorporating these security features in SFMS, the possibility of an unauthorized access will be greatly reduced.

ii) Friendly user interface

SFMS is specially designed on the principle for ease to use. As such, GUI features have been integrated into the system. The inclusion of GUI has contributed vastly to aid users. Users can easily capture the overview of the system, without even referring to the Help provided.

iii) Request For Service

SFMS provides function to the user to report any malfunction or broken facilities to the school staff so that action to repair the particular facility can be done as soon as possible.

iv) Maintenance Scheduling

SFMS provides function to the user to schedule maintenance of facilities. A message box will prompt the user to inform him that the time to do maintenance for a particular facility has arrived.

v) Online Help

Help is provided to every main screen in SFMS. To invoke help, a user just needs to press F1 function key or to click Help button on the screen. This online help makes SFMS easier to use for inexperience users.

7.4 SYSTEM CONSTRAINS AND LIMITATIONS

i) Maintenance scheduling prompting

The function that inform the user that the time to do maintenance for a particular facility is done by comparing the current system date with the scheduled maintenance date in the database. However, the user can change the date and time of the system through Control Panel. SFMS unable to sense changes of system date and time.

ii) Single host

SFMS is a stand-alone system and therefore does not support the multi-user environment. In order to use the system, user has to install the SFMS into their PC

iii) Limited Help Topic

Due to the use of free download software (ForeHelp Premier 2001 Demo Version), there is a limited Help Topics (20 topics). So, when the user press F1 to invoke Help, it is difficult to provide a context-sensitive to the user. So, user needs to choose the help topic by himself.

A good help is able to provide help to help user directly. However in SFMS, the user needs to look for the relevant help topic by himself.

7.5 FUTURE ENCHANCEMENTS

i) Dual lingual support

Due the time constrain in developing SFMS, the language use for the entire interface is in English. If the system could provide Bahasa Malaysia support, it would be more fascinating because most of the administrative works in schools are done in Bahasa Malaysia.

ii) Interactive help (in demo or video from)

If a system could provide a demo package or a short tutorial session, it will definitely help the users learn up the system in a shorter period.

iii) Networking

To really benefit all the users, SFMS shall be modified to enable network accessing. This can be achieved by using the LAN (Local Area Network) environment. In a multi-user environment, SFMS only needed to be install in the server machine and other client machines can gain access to SFMS, simultaneously.

7.6 KNOWLEDGE AND EXPERIENCE GAINED

i) ForeHelp

ForeHelp Premier 2001 Demo version is use to create Help file for SFMS. I have the opportunity to learn to use this software to create help file that uses jump, pop-up and context-sensitive help. Now I can even create or customized my own Windows help file to assist me.

ii) Microsoft Visual Basic 6.0 (Database Programming and Event-Driven Programming)

Developing this system has given me the opportunity to learn VB6. Reading about VB6 is not enough to understand its characteristic. By developing this system, I learn more about VB6 characteristics and these characteristics cannot be gain from reading books.

iii) Microsoft Access 1997

The database in SFMS is developed using Access 97. I have the opportunity to learn more about Access and how to connect to VB6. Only through developing this system that I learn that VB6 cannot be connected to Microsoft Access 2000. So, I have to convert the database to Access 97 format.

iv) Self Expression

Developing SFMS has really given me a great change to express myself in designing and coding of the system. Finally, before graduating, I have the change to build application software by myself. Doing this thesis has greatly improved my self-esteem and self-confidence.

7.7 SUMMARY

Evaluation of a system is indeed needed to ensure its objectives and intended functions have been achieved. This chapter covers all the aspect of evaluating application software. At the end of evaluation, comes the conclusion of this thesis project.

7.8 CONCLUSION

The School Facilities Management System is a start to a computerized school environment in Malaysia. SFMS will be an example of what the function of facilities management in Smart School will be like.

To be given a change to develop this system has been both enjoyable and frustrating to me. It is enjoyable because finally I can really manipulate and use all the knowledge gathered during the three-year course in FSKTM, particularly on the use of software engineering, system analysis and design as well as database management.

It is frustrating because I have to learn the tools that I have chosen (VB) by myself. Although there is a lot of online VB tutorials, however choosing a website that is really suitable is a task for me. So, I would like to take this opportunity to suggest to the faculty to offer a course to teach the students one of the 4^{th} generation languages in the market for example Visual Basic, Power Builder or others. This course will definitely help the student later in their studies or in their work.

Last but not least, the SFMS can be said to have achieved its objectives as well as it requirement as planned earlier. It will serve well as a computerize facilities management in both primary and secondary schools in Malaysia.

GLOSSARY

Application – A business computer system that processes a portion of a database to meet a user's information needs. It consists of menus, forms, reports, queries, and an application program.

API – See Application Program Interface.

Application Program Interface – A set of program procedures or functions that can be called to invoke a set of services. The API includes the names of procedures and functions and a description of the name, purpose, and data type of parameters to be provided. For example, a DBMS product could provide a library of functions to call for database services. The names of procedures and their parameters constitute the API for that library.

Black-Box Testing – Black box testing relies on the specification of the system or component, which is being tested to derive test cases. The system is considered as a 'black-box' whose behaviour can only be determined by studying its inputs and the related activities.

Candidate Key – An attribute or group of attributes that identifies a unique row in a relation. One of the candidate keys is chosen to be the primary key.

CBIS - Computer Based Information System

Database - A collection of integrated records that are kept in files for future reference.
Hardware - The mechanical devices that comprise a computer system, such as the central processing unit, monitor, keyboard, and mouse, as well as other equipment like printers and speakers.

DBMS – Database Management System. A set of programs used to define, administer, and process the database and its application.

Foreign Key – An attribute that is the key of one or more relations other than the one in which it appear.

HTML (Hypertext Markup Language) - The coded format language used for creating hypertext documents on the World Wide Web and controlling how Web pages appear.

HTTP (Hypertext Transfer Protocol) - The standard language that computers connected to the World Wide Web use to communicate with each other.

Internet - A global connection of computer networks, also referred to as the "Net," which share a common addressing scheme. (See also "World Wide Web")

MIS - Management Information System

Modem - A hardware device that allows computers to communicate with each other by transmitting signals over telephone lines, enabling what is called "dial-up access." Modems come in different speeds. The higher the speed, the faster the data are transmitted. The fastest widely available modems are "56K" (or 56 kilobits per second).
Multimedia - Information presented in more than one format, such as text, audio, video, graphics, and images.

Normal Form – A rule or set of rules governing the allowed structure of relations. The rules apply to an attribute, functional dependencies, multivalued dependencies, domains, and constrains. The most important normal forms are 1NF, 2NF and 3NF.

Normalisation – The process of evaluating a relation to determine whether it is in a specific normal form and, if necessary, of converting it to relations in that specific normal form,

ODBC - Open Database Connectivity, a standard database access method developed by Microsoft Corporation. The goal of ODBC is to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data.

PB - Power Builder is a graphic PC-based client/server application development environment.

Primary Key – A candidate key selected to be the key of relation.

Search engine -A tool that enables users to locate information on the World Wide Web. Search engines use keywords entered by users to find Web sites, which contain the information sought. Some search engines are specifically designed to find Web sites intended for children.

Server - A special computer connected to a network that provides (serves up) data. A Web server transmits Web pages over the Internet when it receives a Web browser's request for a page. A server can also be called a host or node.

SFMS - School Facilities Management System

Software - A computer program, which provides the instructions, which enable the computer hardware to work. System software, such as Windows or Mac OS, operate the machine itself, and applications software, such as spreadsheet or word processing programs, provide specific functionality.

VB6 - Visual Basic 6.0 an event-driven programming language that are going to be used in developing SFMS.

Web - The World Wide Web. An Internet system to distribute graphical, hyper-linked information, based on the hypertext transfer protocol (HTTP). The World Wide Web is also known as WWW or W3. The Web is not synonymous with the Internet; rather, it is just one service on the Internet. Other services on the Internet include Internet Relay Chat and Newsgroups. The Web is accessed through use of a browser.

Web site - A collection of "pages" or files linked together and available on the World Wide Web. Web sites are provided by companies, organizations and individuals.

WWW - The World Wide Web. (See also "Web")

White-Box Testing – Testing strategies objective is to know every independent execution path through the component. If every independent path is executed then a state in the program must have been executed at least once. Furthermore, condition statements are also tested for both true false cases.

4GL - 4th Generation Language

SISTEM PENGURUSAN KEMUDAHAN SEKOLAH

Nama sekolah :
Nama responden :
Jawatan :

Perhatian : *Segala maklumat yang diberikan dalam soal selidik ini adalah sulit dan hanya akan digunakan untuk tujuan analisis dan pembangunan Sistem Pengurusan Kemudahan Sekolah.*

Anda dibenarkan untuk memilih lebih daripada satu jawapan.

Kemudahan

1. Apakah status sekolah anda?
   - ☐ **Sekolah Bistari**
   - ☐ **Sekolah Dalam Bandar**
   - ☐ **Sekolah Luar Bandar**
   - ☐ **Sekolah Berasrama Penuh**
   - ☐ **Lain-lain, sila nyatakan**_____

2. Apakah sistem pengendaliaan yang digunakan oleh komputer di sekolah anda?
   - ☐ **Window 98**
   - ☐ **Window 95**
   - ☐ **Window yang lebih lama dari Win 95, sila nyatakan**_____
   - ☐ **Lain-lain, sila nyatakan**_____

3. Adakah anda berpuas hati dengan cara pengurusan kemudahan di sekolah anda?
   - ☐ **Tidak puas hati**
   - ☐ **Sedikit puas hati**
   - ☐ **Puas hati**
   - ☐ **Sangat puas hati**
   - ☐ **Tiada komen**

4. Pada pendapat anda, efektifkah sistem pengurusan kemudahan di sekolah anda?
   - ☐ **Ya** ☐ **Tidak**

5. Adakah sekolah ini mempunyai sistem berkomputeran yang mengendalikan kemudahan sekolah?
   - ☐ **Ya** ☐ **Tidak**

6. Jika ya, apakah sistem pengendalian yang digunakan oleh sistem berkenaan?
   - ☐ **Window 98**
   - ☐ **Window 95**
   - ☐ **Window yang lebih lama dari Win 95, sila nyatakan**_____
   - ☐ **Lain-lain, sila nyatakan**_____

7. Nyataakan perisian yang digunakan oleh sistem tersebut.

_____

8. Sila berikan takrifan bagi kemudahan dalam konteks sekolah.
   - ☐ **Segala bentuk peralatan yang digunakan untuk membantu pengajaran dan pembelajaran.**
   - ☐ **Segala jenis bilik-bilik khas termasuk makmal.**
   - ☐ **Segala bentuk peralatan dan bilik khas.**
   - ☐ **Lain-lain, sila nyatakan**_____
     _____

9. Siapakah yang menentukan penggunaan sesuatu bilik khas di sekolah?
   - ☐ **Guru Besar**
   - ☐ **Penolong Guru Besar I**
   - ☐ **Penolong Guru Besar II**
   - ☐ **Lain-lain, sila nyatakan**_____

10. Siapakah yang bertanggungjawab terhadap keselamatan alatan dalam setiap bilik khas di sekolah?
    - ☐ **Guru Besar**
    - ☐ **Penolong Guru Besar I**
    - ☐ **Penolong Guru Besar II**
    - ☐ **Pekerja Am**
    - ☐ **Lain-lain, sila nyatakan**_____

11. Siapakah yang bertanggung jawab terhadap penjadualan penggunaan kemudahan?
    - ☐ **Guru Besar**
    - ☐ **Penolong Guru Besar I**
    - ☐ **Penolong Guru Besar II**
    - ☐ **Lain-lain, sila nyatakan**_____

12. Bolehkah kemudahan sekolah digunakan selepas waktu sekolah?
    ☐ **Ya** ☐ **Tidak**

13. Selain daripada para pelajar dan kakitangan sekolah, siapakah yang turut menggunakan kemudahan sekolah.
    - ☐ **Pelajar sekolah lain**
    - ☐ **Ibu bapa atau penjaga**
    - ☐ **Orang ramai selain ibu bapa dan penjaga**
    - ☐ **Lain-lain, sila nyatakan**_____

14. Apakah kemudahan yang digunakan oleh mereka?
    - ☐ **Padang sekolah**
    - ☐ **Dewan**
    - ☐ **Gelanggang Badminton**
    - ☐ **Gelanggang bola keranjang**
    - ☐ **Gelanggan tennis**
    - ☐ **Kolam renang**
    - ☐ **Gymnasium**
    - ☐ **Alatan sukan**
    - ☐ **Pakaian khas**
    - ☐ **Lain-lain, sila nyatakan**_____

15. Apakah penggunaan yang mungkin bagi dewan serbaguna sekolah (dalam konteks sekolah)?
    - ☐ **Dewan perhimpunan**
    - ☐ **Dewan peperiksaan**
    - ☐ **Dewan makanan (jamuan sekolah)**
    - ☐ **Pentas persembahan**
    - ☐ **Dewan latihan sukan dalamam sepertu gynantic**
    - ☐ **Gelanggang badminton**
    - ☐ **Gelanggang bola keranjang**
    - ☐ **Gelanggang bola jaring**
    - ☐ **Gelanggang ping pong**
    - ☐ **Lain-lain, sila nyatakan**_____

16. Apakah penggunaan yang mungkin bagi dewan sekolah yang disewakan kepada orang ramai?
    - ☐ **Jamuan perkahwinan**
    - ☐ **Jamuan harijadi**
    - ☐ **Pameran**
    - ☐ **Latihan sukan**
    - ☐ **Lain-lain, sila nyatakan**_____

17. Apakah peralatan di sekolah yang perlu diselenggarakan?
    - ☐ **Alat pemadam api**
    - ☐ **Komputer**
    - ☐ **Alatan makmal sains**
    - ☐ **Alatan makmal elektrik dan elektronik**
    - ☐ **Alatan makmal jahitan**
    - ☐ **Alatan bilik pertukangan**
    - ☐ **Dewan**
    - ☐ **Pentas sekolah**
    - ☐ **Bilik darjah**
    - ☐ **Bilik-bilik khas**
    - ☐ **Sistem PA**
    - ☐ **Lain-lain, sila nyatakan**_____

Inventori

1. Sila terangkan dengan ringkas bagaimana inventori sekolah dijalankan?
   - ☐ **Penggunaan kad inventori**
   - ☐ **Pengiraan peralatan serta perkakasan sekolah dan guru atau pekerja yang bertanggungjawab.**
   - ☐ **Lain-lain, sila nyatakan**_____

   _____

2. Berapa kerap inventori tersebut dijalankan?
   - ☐ **Sekali sebulan**
   - ☐ **Sekali setahun**
   - ☐ **Dua kali setahun**
   - ☐ **Lain-lain, sila nyatakan**_____

3. Siapakah yang bertanggungjawab dalam inventori sekolah?
   - ☐ **Guru Besar**
   - ☐ **Penolong Guru Besar I**
   - ☐ **Penolong Guru Besar II**
   - ☐ **Pekerja Am**
   - ☐ **Lain-lain, sila nyatakan**_____

4. Siapakah yang mempunyai autoriti dalam menempah peralatan baru di sekolah selain daripada pengetua.
   - ☐ **Penolong Guru Besar I**
   - ☐ **Penolong Guru Besar II**
   - ☐ **Guru matapelajaran berkenaan**
   - ☐ **Guru panitia**
   - ☐ **Pekerja Am**
   - ☐ **Lain-lain, sila nyatakan**_____

5. Adakah peralatan dalam bilik darjah seperti penyapu dan pemadam turut diambil kira dalam inventori?
   - ☐ **Ya** ☐ **Tidak**

6. Adakah inventori yang dilakukan termasuklah perisian komputer serta lesen yang dimiliki oleh sekolah?
   - ☐ **Ya** ☐ **Tidak**

7. Apakah perisian yang dimiliki oleh pihak sekolah?
   - ☐ **Window 3.1 / atau versi yang lebih tinggi**
   - ☐ **Perisian pemprosesan perkataan seperti Microsoft Word**
   - ☐ **Perisian perakaunan seperti Microsoft Excel**
   - ☐ **Perisian persembahan seperti Microsoft PowerPoint**
   - ☐ **Pakej pembelajaran berkomputer**
   - ☐ **Lain-lain, sila nyatakan**_____

Maklumat-maklumat yang ingin saya kumpulan untuk pangkalan data.

1. Senarai kemudahan sekolah.
2. Senarai bilik-bilik khas di sekolah termasuklah bilangannya.
3. Senarai penggunaan yang mungkin untuk setiap bilik tersebut.
4. Senarai alatan dalam setiap bilik khas tesebut.
5. Senarai alatan yang dijalan inventori.
6. Senarai pembekal peralatan sekolah.
7. Susut nilai bagi setiap peralatan tersebut.
8. Jadual penyelenggaraan peralatan termasuklah peralatan Bomba.

QUESTIONNAIRE ANALYSIS

Out of 5 questionnaires distributed to schools (by Puan Abrizah), only one reply. So, this analysis is done from the answer given by the particular school. So, it cannot be concluded that this analysis represents the management of facilities in the schools.

Facilities

From the answer given by respondent, it can be concluded that the school is using Window 98 with Access 97 to keep track of facilities in the school specially to keep track of inventory in the school.

Facilities in the school is manage in such a way that different teachers are assigned to in charge of the usage of special rooms and special equipments in the school. The particular person who is in charge of the room will also be in responsible to the safety of equipments stored inside the room. The facilities in the school cannot be use after normal school hour.

There are no other people (outsiders) who use the facilities in the school. So, they do not practice renting facilities of outsider.

Among all the facilities in the school, only few need maintenance periodically, namely fire extinguishers, computer, the Hall, and the Public Address (PA) System.

Inventory

In doing inventory, the respondent uses inventory card. They do their inventory once a year, and there is only one teacher who is responsible of carrying out the inventory. Counting all the equipments in the school, room by room, does inventory in the particular school, which include everything (CDs, licenses, brooms, chairs, tables, etc).

CODING SAMPLE

AdoDB Module:

```
Global tempName As String
Global track_build As Boolean
Global track_room As Boolean
Global From_date
Global Until_date
Public adoCn As ADODB.Connection

Function openDbCn() As Boolean
   Dim connectionString As String
   Set adoCn = New ADODB.Connection

   On Error GoTo dbError
   adoCn.Open "dsn=SFMS"
   openDbCn = True
   Exit Function

dbError:
   MsgBox (Err.Description)
   openDbCn = False
End Function
```

FrmTrack.frm

```
Option Explicit

Private Sub cmdCancel_Click(Index As Integer)
   Unload Me
End Sub

Private Sub cmdOK_Click()
   If optBuild.Value = True Then
      frmTrack_Build.Show
      Unload Me
   Else
      If optRoom.Value = True Then
      frmTrack_Room.Show
      Unload Me
   Else
      MsgBox "You have to select a type of tracking!"
      End If
   End If
End Sub

Private Sub Form_Load()
   App.HelpFile = App.Path & "\Help\Win95\Sfms.hlp"
   optRoom.Value = False
   optBuild.Value = False
End Sub
```

FrmTrack_Builld.frm

```
Option Explicit
Dim temp As String
Dim Data As Boolean
Dim ChildData As Boolean
Dim Active As Boolean
Dim rsUser As New ADODB.Recordset
Dim rsBuild As New ADODB.Recordset
Dim rsRoom As New ADODB.Recordset
Dim rsAllRoom As New ADODB.Recordset

Private Sub cboBuild_Name_Click()
   If Data = True Then
      ClearRoomDetail
      lvw.ListItems.Clear
      LoadData
   Else
      If Data = False Then
         LoadData
      End If
   End If
End Sub


Private Sub Form_Load()
   App.HelpFile = App.Path & "\Help\Win95\Sfms.hlp"
   Dim MyDate
   MyDate = Date
   lblDate.Caption = MyDate
   Data = False
   If (Not openDbCn()) Then
      MsgBox "Database failed..."
   Else
      rsUser.Open "select * from upf where upfuic = '" & tempName & "'", adoCn,
adOpenKeyset, adLockOptimistic
      If Not rsUser.EOF Then
         lblUser = rsUser("upffnm")
         rsUser.Close
      Else
         MsgBox "There is no Name for this user!"
         rsUser.Close
         Exit Sub
      End If
   End If

   rsBuild.Open "select * from building", adoCn, adOpenKeyset, adLockOptimistic
   rsBuild.MoveFirst
   Dim X
   Do Until rsBuild.EOF
     cboBuild_Name.AddItem rsBuild.Fields("building_name").Value, X
```

```
      rsBuild.MoveNext
      X = X + 1
    Loop
End Sub


Private Sub Form_LostFocus()
'   Active = False
End Sub

Private Sub Form_Unload(Cancel As Integer)
   rsBuild.Close
   adoCn.Close
End Sub



Private Sub lvw_BeforeLabelEdit(Cancel As Integer)
   Cancel = 1
End Sub

Private Sub lvw_ColumnClick(ByVal ColumnHeader As
MSComctlLib.ColumnHeader)
  ' When a ColumnHeader object is clicked, the ListView control is
  ' sorted by the subitems of that column.
  ' Set the SortKey to the Index of the ColumnHeader - 1
  lvw.SortKey = ColumnHeader.Index - 1

  ' Set Sorted to True to sort the list.
  lvw.Sorted = True
End Sub


Private Sub lvw_ItemClick(ByVal Item As MSComctlLib.ListItem)
      rsAllRoom.Open "select * from room where room_code = '" & Item & "'",
adoCn, adOpenDynamic, adLockOptimistic
      If rsAllRoom.EOF Then
         MsgBox "No Room!"
      Else
         RoomDetail
      End If
End Sub


Private Sub Timer1_Timer()
 If lblTime.Caption <> CStr(Time) Then
    lblTime.Caption = Time
  End If
End Sub
```

```
Sub LoadData()
   rsBuild.MoveFirst
   rsBuild.Find "building_name = '" & cboBuild_Name.Text & "'", , , 0
   temp = rsBuild("building_code")
   lblBuild_Code = temp
   rsRoom.Open "select room_code, room_name from room where building_code = '"
& temp & "'", adoCn, adOpenDynamic, adLockOptimistic

   With rsRoom
      If Not .EOF Then
      .MoveFirst
      Do Until .EOF
         lvw.ListItems.Add 1, rsRoom("room_code"), rsRoom("room_code")
         lvw.ListItems(1).SubItems(1) = rsRoom("room_name")
         .MoveNext
      Loop
         Else
            MsgBox "There is no room in this building!"
            lvw.ListItems.Add , , ""
            lvw.ListItems.Clear
            ChildData = True
            .Close
            Exit Sub
         End If
   .Close
   Data = True
   End With
End Sub

Sub RoomDetail()
   If IsNull(rsAllRoom("room_desc")) Then
      txtRoom_Desc = ""
   Else
      txtRoom_Desc = rsAllRoom("room_desc")
   End If
   lblRoom_Capacity.Caption = rsAllRoom("room_capacity")
   lblRoom_Name.Caption = rsAllRoom("room_name")
   lblRoom_Code.Caption = rsAllRoom("room_code")
   rsAllRoom.Close
End Sub

Sub ClearRoomDetail()
   txtRoom_Desc.Text = ""
   lblRoom_Capacity.Caption = ""
   lblRoom_Name.Caption = ""
   lblRoom_Code.Caption = ""
End Sub
```

FrmTrack_Room.frm

```
Option Explicit
Dim temp As String
Dim Data As Boolean
Dim bActive As Boolean
Dim ChildData As Boolean
Dim myBookmark As Variant
Dim rsUser As New ADODB.Recordset
Dim rsRoom As New ADODB.Recordset
Dim rsEquip As New ADODB.Recordset
Dim rsEquipDetail As New ADODB.Recordset


Private Sub cboRoom_Code_Click()
   If Data = True Then
      ClearEquipDetail
      lvw.ListItems.Clear
      rsRoom.MoveFirst
      rsRoom.Find "room_name = '" & cboRoom_Code.Text & "' ", , , 0
      myBookmark = rsRoom.Bookmark
      LoadData
   Else
      If Data = False Then
         LoadData
      End If
   End If
End Sub


Private Sub Form_Load()
   App.HelpFile = App.Path & "\Help\Win95\Sfms.hlp"
   Dim MyDate
   MyDate = Date
   lblDate.Caption = MyDate
   Data = False
    If (Not openDbCn()) Then
      MsgBox "Database failed..."
    Else
      rsUser.ActiveConnection = adoCn
      rsUser.LockType = adLockOptimistic
      rsUser.CursorType = adOpenKeyset
      rsUser.Open "select * from upf where upfuic = '" & tempName & "'"
      If Not rsUser.EOF Then
         lblUser = rsUser("upffnm")
         rsUser.Close
      Else
         MsgBox "There is no Name for this user!"
         rsUser.Close
         Exit Sub
      End If
```

```
    End If

    rsRoom.Open "select room_code, room_name from room", adoCn, adOpenKeyset,
adLockOptimistic
    rsRoom.MoveFirst
    Dim X
    Do Until rsRoom.EOF
        cboRoom_Code.AddItem rsRoom.Fields("room_name").Value, X
        rsRoom.MoveNext
        X = X + 1
    Loop
End Sub

Private Sub Form_Unload(Cancel As Integer)
    rsRoom.Close
    adoCn.Close
End Sub

Private Sub Timer1_Timer()
  If lblTime.Caption <> CStr(Time) Then
      lblTime.Caption = Time
   End If
End Sub


Private Sub lvw_BeforeLabelEdit(Cancel As Integer)
    Cancel = 1
End Sub

Private Sub lvw_ColumnClick(ByVal ColumnHeader As
MSComctlLib.ColumnHeader)
    ' When a ColumnHeader object is clicked, the ListView control is
    ' sorted by the subitems of that column.
    ' Set the SortKey to the Index of the ColumnHeader - 1
    lvw.SortKey = ColumnHeader.Index - 1

    ' Set Sorted to True to sort the list.
    lvw.Sorted = True
End Sub


Private Sub lvw_ItemClick(ByVal Item As MSComctlLib.ListItem)

      rsEquipDetail.Open "select * from equipment where equip_name = '" & Item &
"'", adoCn, adOpenDynamic, adLockOptimistic
      If rsEquipDetail.EOF Then
          MsgBox "No equipment in this Room!", vbInformation
      Else
          EquipDetail
      End If
```

```
End Sub

Sub EquipDetail()
   lblEquip_Code.Caption = rsEquipDetail("equip_code")
   lblEquip_Name.Caption = rsEquipDetail("equip_name")

   If IsNull(rsEquipDetail("equip_desc")) Then
      txtEquip_Desc.Text = ""
   Else
      txtEquip_Desc.Text = rsEquipDetail("equip_desc")
   End If
   lblEquip_Type.Caption = rsEquipDetail("type_name")
   lblInven_No.Caption = rsEquipDetail("inven_no")
   'ChildData = True
   '03/01/2000
   rsEquipDetail.Close
End Sub

Sub LoadData()
   Dim i As Integer
   Dim rs As New ADODB.Recordset
   rs.Open "select room_code from room where room_name = '" &
cboRoom_Code.Text & "'", adoCn, adOpenKeyset, adLockOptimistic
   temp = rs.Fields("room_code")
   lblRoom_Name = temp
   rsEquip.Open "select room_code, equip_code, equip_name, inven_no, type_name
from equipment where room_code = '" & temp & "'", adoCn, adOpenDynamic,
adLockOptimistic

   With rsEquip
   If Not .EOF Then
      .MoveFirst
      Do Until .EOF
         lvw.ListItems.Add 1, rsEquip("equip_name"), rsEquip("equip_name")
         lvw.ListItems(1).SubItems(1) = rsEquip("equip_code")
         lvw.ListItems(1).SubItems(2) = rsEquip("type_name")
         lvw.ListItems(1).SubItems(3) = rsEquip("inven_no")
         .MoveNext
      Loop
         Else
            MsgBox "There is no equipment in this room!"
            lvw.ListItems.Add , , ""
            lvw.ListItems.Clear
            .Close
            Exit Sub
         End If
   .Close
   Data = True
   End With
End Sub
```

```
Sub ClearEquipDetail()
    lblEquip_Code.Caption = ""
    lblEquip_Name.Caption = ""
    lblEquip_Usage.Caption = ""
    lblEquip_Type.Caption = ""
    txtEquip_Desc.Text = ""
End Sub
```

SCHOOL FACILITIES MANAGEMENT SYSTEM
USER MANUAL
(FOR SYSTEM ADMINISTRATOR ONLY)

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

INTRODUCTION

School Facilities Management System (SFMS), is a system specially designed to meet the requirements of the School Administrative Staffs in managing the school's facilities. The system incorporates many functions needed to properly manage facilities in schools.

SFMS is designed to helps its users to better perform their responsibility in properly manage the ready available facilities in school. It will also assist the user in their management of school as well as assisting them in decision making.

However, it is you, the user to SFMS who is really bringing the system to life. The system is created to serve you. But in order to serve you well, accurate and timely input of data to SFMS is vital.

This user manual is specially written for the use of System Administrator only. For other users, please refer to another copy of user manual.

CHAPTER 1 SYSTEM REQUIREMENT

Hardware Requirement

The hardware requirements for installing SFMS are summarized in Table 1.1:

Component	*Description*
Microprocessor	IBM compatible machine with Pentium 133MHz processor or higher 
RAM	> 16 MB RAM
Storage	40 MB of hard disk
Input Device	Mouse and keyboard
Video Monitor	EGA, VGA or compatible display

Table 1.1 Hardware Requirements

Software Requirement

The software requirements for installing SFMS are summarized in Table 3.3:

Component	*Description*
Operating System	Windows 95 or later in standard
Programming Language	Visual Basic Enterprise Edition 6
Database	Microsoft Access 1997

Table 1.2 Software Requirements

CHAPTER 2 INSTALLING SFMS

Steps:

1. To install SFMS, insert the **SFMS CD** into the **CD-ROM**.

2. Starting the set-up with two different ways:
   i) Through **Window Explorer**
   Open the Window Explorer and go to the **CD-ROM** drive. Select and double-click the **setup.exe** file.
   ii) Through the **Run** function
   From the **Start Menu**, choose **Run** (see figure 2.1). Click the **Browse** button and search the **setup.exe** file from the CD-ROM drive. Click ***Open*** to return to the Run window. Click **Ok** to continue. (See Figure 2.2)

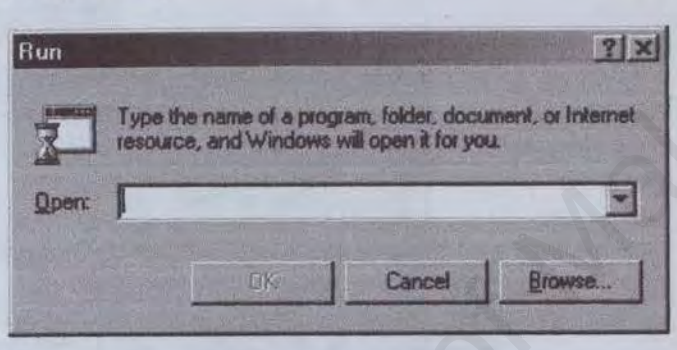

Figure 2.1 Run Function

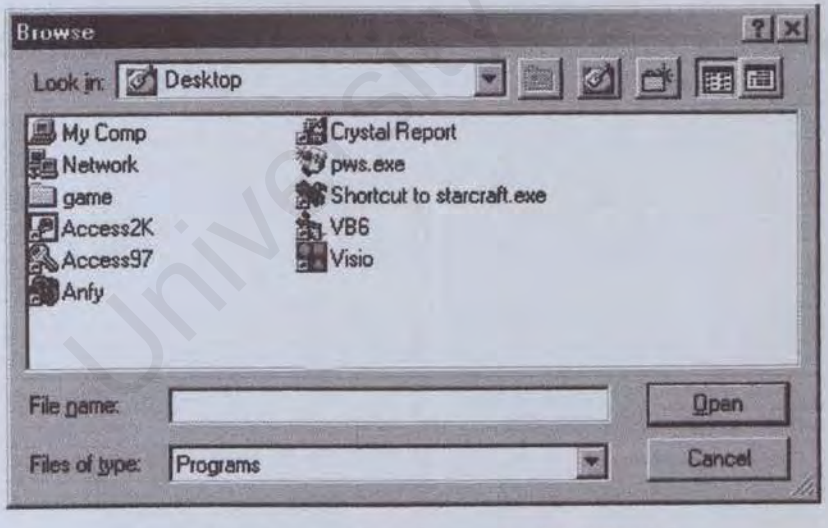

Figure 2.2 Browse For setup.exe in CD-ROM Drive

3. Now the installation process will continue.

4. By default, SFMS will be install to **C:\Program Files\SFMS**. To change directory please click **Change Directory**.

CHAPTER 3 SETTING UP ODBC

After you have successfully install SFMS into your PC, you have setup the **ODBC Data Source** before you can start using SFMS.

Steps:

1. Click the **Start Menu** and choose **Setting** and double-click on **Control Panel**. (See Figure 3.1)

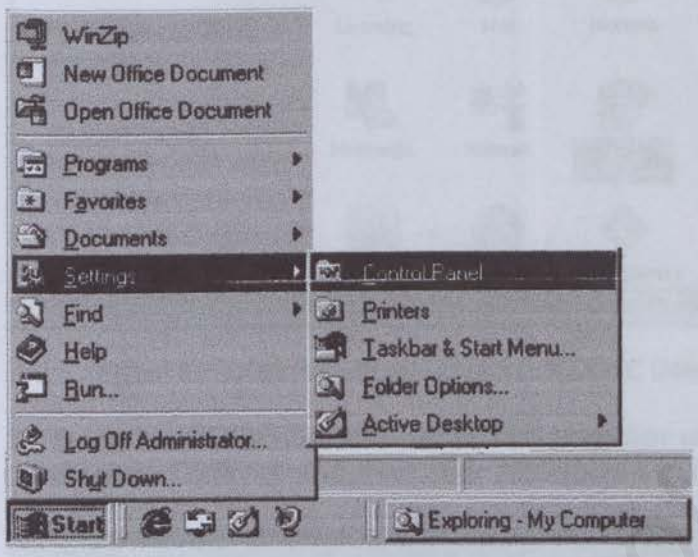

Figure 3.1 Control Location Of Control Panel in Start Menu

2. On **Control Panel**, choose **ODBC Data Sources icon**, and double-click on it. (See Figure 3.2)

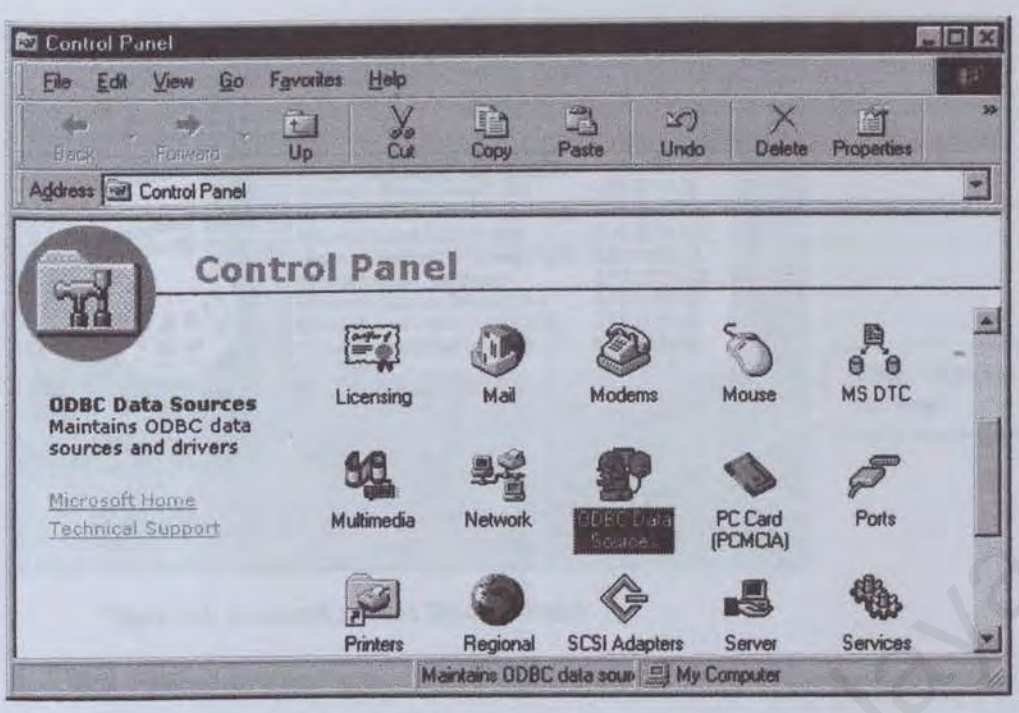

Figure 3.2 Control Panel and Location of ODBC Data Sources

3. Once the **ODBC Data Source Administrator** window appear, Choose the **System DSN** tab and then click **Add** (See Figure 3.3).

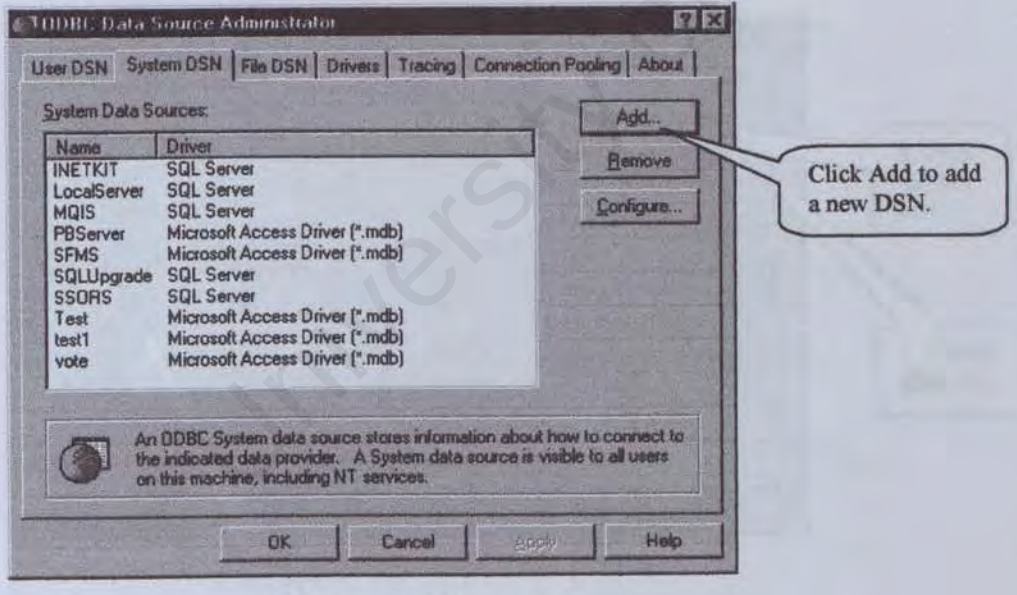

Figure 3.3 System DSN

4. A window to enable you create a **New Data Source** will appear as in Figure 3.4. Choose **Microsoft Access Driver (*.mdb)**

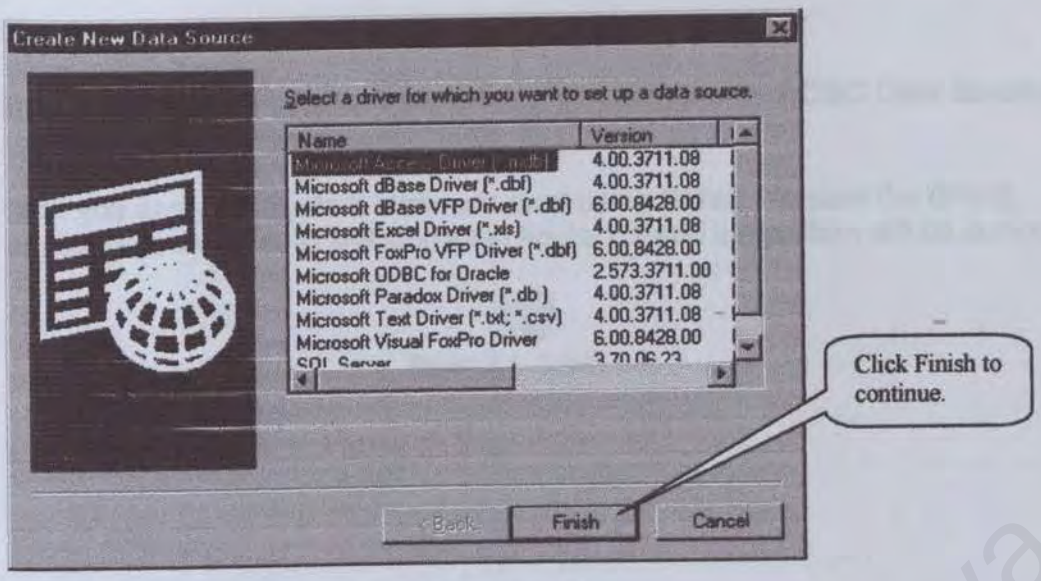

Figure 3.4 Microsoft Access Driver (*.mdb)

5. Click **Finish** to continue. (See Figure 3.4)

6. Type **SFMS** in the **Data Source Name**, and click **Select** to select the **97fms.mdb** database in **SFMS Directory** in your hard drive. (See Figure 3.5). By default the 97fms.mdb file will be in your **C:\Program Files\SFMS\Database** directory.

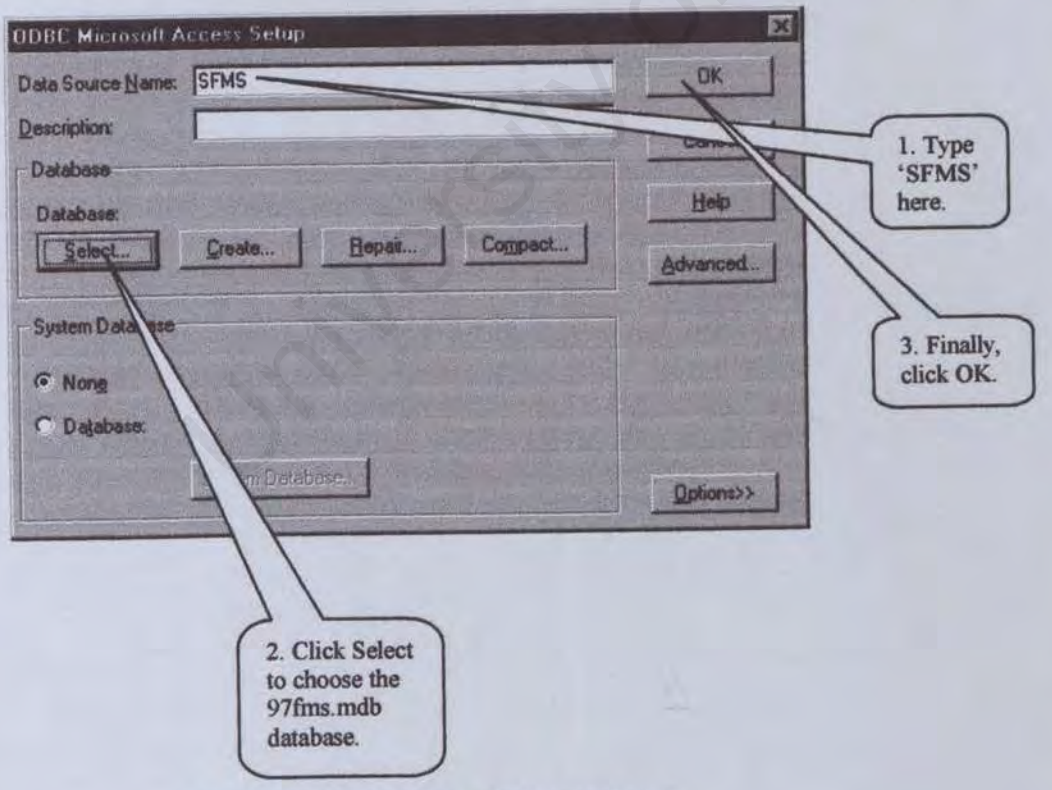

Figure 3.5 ODBC Microsoft Access Setup

7. After you have chosen the **97fms.mdb**, click **OK**.

8. Congratulation, you have successfully setting up the **ODBC Data Source** for SFMS.

Now you and your System User can start using SFMS. To start the SFMS application, click the SFMS icon as in figure 1.1 and the system will be launch.

Figure 3.6 SFMS icon

SCHOOL FACILITIES MANAGEMENT SYSTEM
USER MANUAL

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

INTRODUCTION

School Facilities Management System (SFMS), is a system specially designed to meet the requirements of the School Administrative Staffs in managing the school's facilities. The system incorporates many functions needed to properly manage facilities in schools.

SFMS is designed to helps its users to better perform their responsibility in properly manage the ready available facilities in school. It will also assist the user in their management of school as well as assisting them in decision making.

However, it is you, the user to SFMS who is really bringing the system to life. The system is created to serve you. But in order to serve you well, accurate and timely input of data to SFMS is vital.

Having said that, I wish you a good time using the system.

CHAPTER 1 LOGIN TO SFMS

1. To start the SFMS application, click the **SFMS icon** as in figure 1.1 and the system will be launch.

Figure 1.1 SFMS icon

2. When a user press the **SFMS icon**, the splash screen as in Figure 1.2 will pop-up. The splash will show basic information about SFMS to the user. Information on the splash screen is as below:

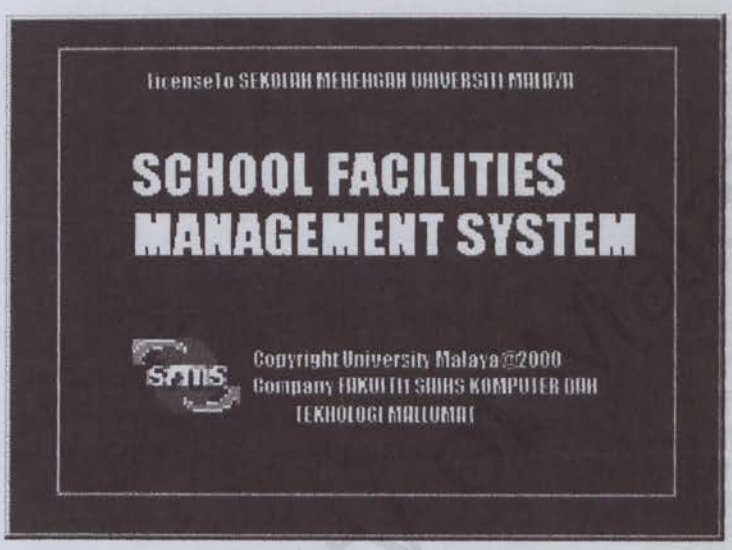

Figure 1.2 SFMS Splash Screen

3. After the splash screen, the program authentication screen will appear such as in Figure 1.3.

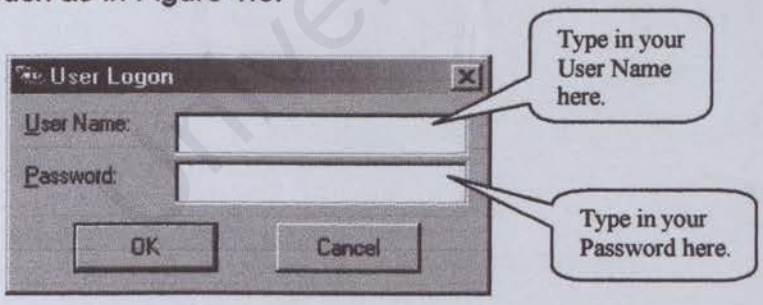

Figure 1.3 SFMS Program Authentications

4. Please enter the appropriate **Username** and **Password**. (By default your username will be your IC number).

5. If you forgot you password, please see the **System Administrator**.

6. After passing the authentication check, the SFMS main menu will appear as in Figure 2.1.

CHAPTER 2 MAIN MENU

1. There are 7 buttons to access each function in SFMS. The buttons are as followed:
    i) Inventory
    ii) Request for service
    iii) Maintenance
    iv) Report
    v) Booking – Please refer to Lok Farn's user manual
    vi) Tracking usage
    vii) Administrator– Please refer to Lok Farn's user manual
    viii) About me
    ix) Exit

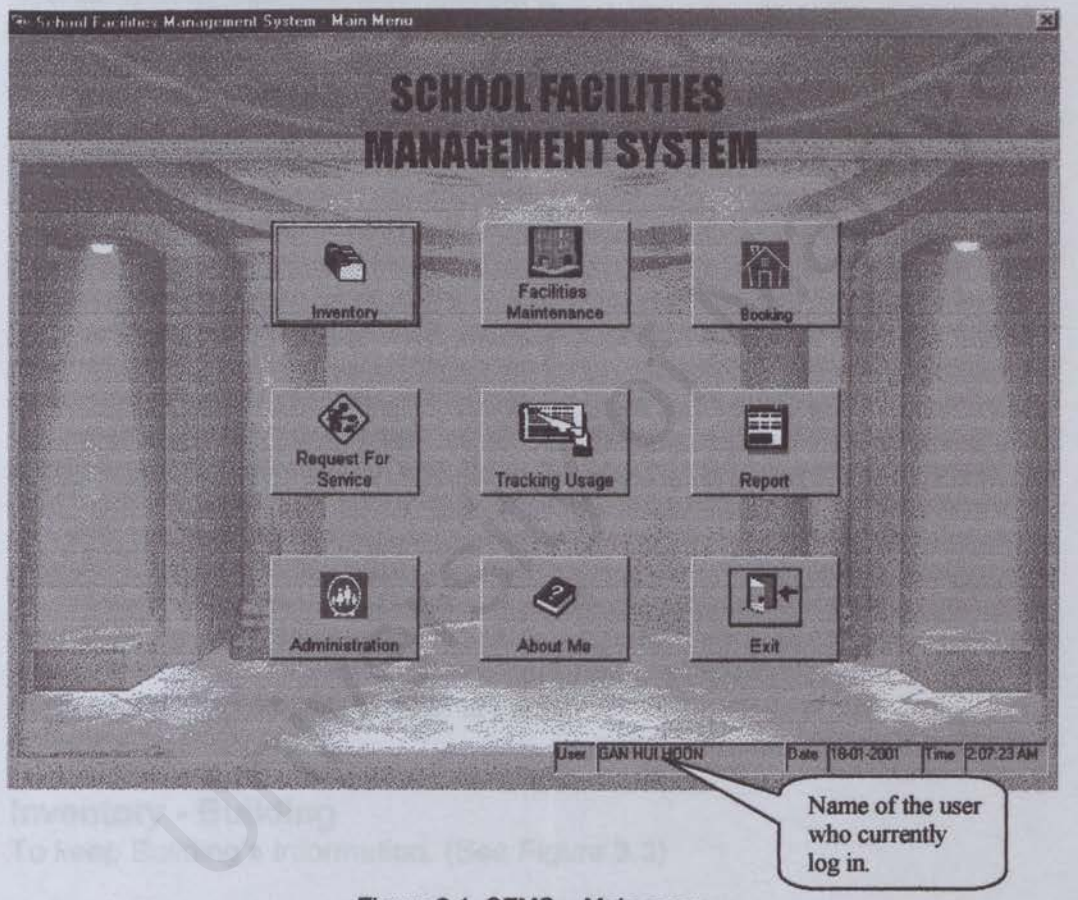

Figure 2.1 SFMS – Main menu

2. Explanation for each functionality (buttons) will be presented in later chapters.

CHAPTER 3 INVENTORY

1. To do inventory for the school, please inventory button

Figure 3.1 Inventory Button

2. After clicking the Inventory button, a screen like Figure 3.1 wills pop-up. You have to choose the type of facilities that you want to do **Inventory** for.

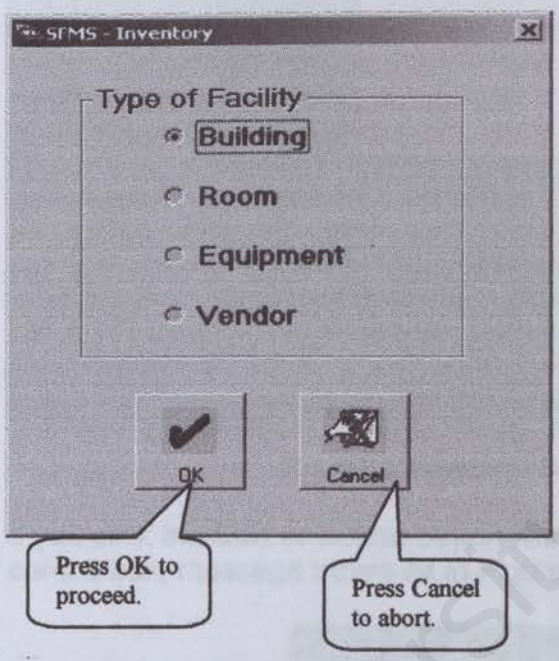

Figure 3.2 SFMS - Inventory

3. Press **OK** to proceed, **Cancel** to abort.

Inventory - Building

To keep Building's Information. (See Figure 3.3)

Edit - To edit the current record
Prerequisite - User has to choose a Building Name from the combo box.

Add - To add a new Building record.
Prerequisite - User has to choose a Building Name from the combo box.

Delete - To delete the current record.

Help - Help on current topic

Exit - Exit from current screen.

Save - To save the changes or the new added record.
Prerequisite - Edit or Add button has been clicked.

Cancel - To cancel your current operation.
Prerequisite - Edit or Add button has been click.

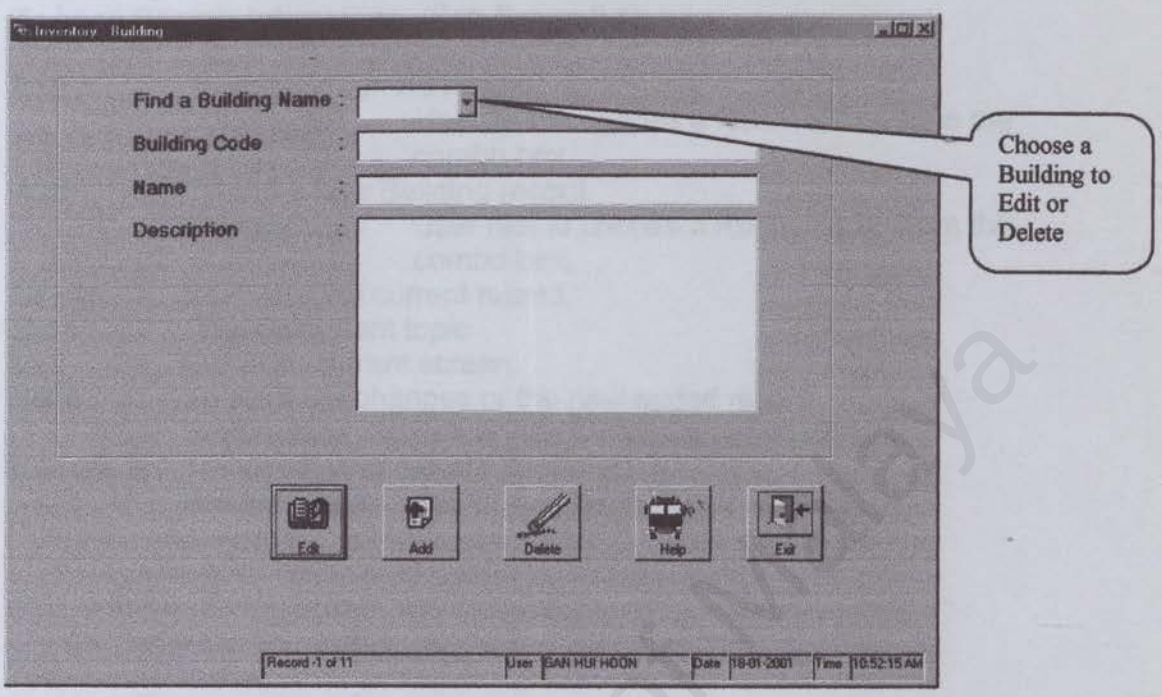

Figure 3.3 Inventory - Building

If you click the **Edit** or **Delete** button without choosing a Building from the combo box, message boxes as in Figure 3.4 or Figure 3.5 will pop-up.

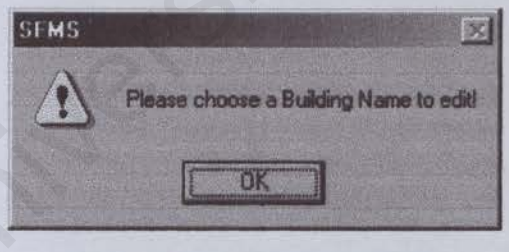

Figure 3.4 Error in Editing Record

Figure 3.5 Error in Deleting Building Record

If any of the message boxes do appear, you just have to press **OK** and return to the Inventory screen to select a Building record before you continue.

Inventory – Room

To keep Room's Information. (See Figure 3.6)

Edit - To edit the current record
Prerequisite - User has to choose a Room Name from the combo box.

Add - To add a new Building record.
Prerequisite - User has to choose a Room Name from the combo box.

Delete - To delete the current record.

Help - Help on current topic

Exit - Exit from current screen.

Save - To save the changes or the new added record.
Prerequisite - Edit or Add button has been clicked.

Cancel - To cancel your current operation.
Prerequisite - Edit or Add button has been click.

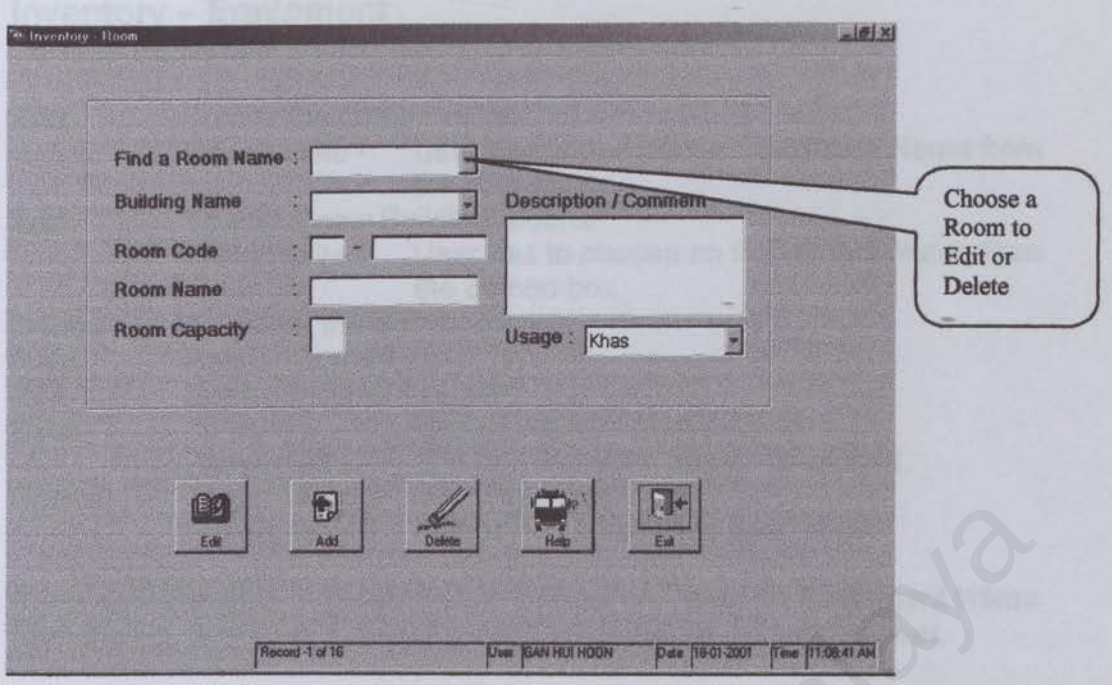

Figure 3.6 Inventory - Room

If you click the **Edit** or **Delete** button without choosing a Room from the combo box, message boxes as in Figure 3.7 or Figure 3.8 will pop-up.

Figure 3.7 Error in Editing Record.

Figure 3.8 Error in Deleting Record

If any of the message boxes do appear, you just have to press **OK** and return to the **Inventory** screen to select a Room record before you continue.

Inventory – Equipment

To keep Equipment's Information. (See Figure 3.9)

Edit - To edit the current record
Prerequisite - User has to choose an Equipment Name from the combo box.

Add - To add a new Building record.
Prerequisite - User has to choose an Equipment Name from the combo box.

Delete - To delete the current record.

Help - Help on current topic

Exit - Exit from current screen.

Save - To save the changes or the new added record.
Prerequisite - Edit or Add button has been clicked.

Cancel - To cancel your current operation.
Prerequisite - Edit or Add button has been click.

If you click the **Edit** or **Delete** button without choosing an Equipment Name from the list, a message as in Figure 3.10 or Figure 3.11 will appear.

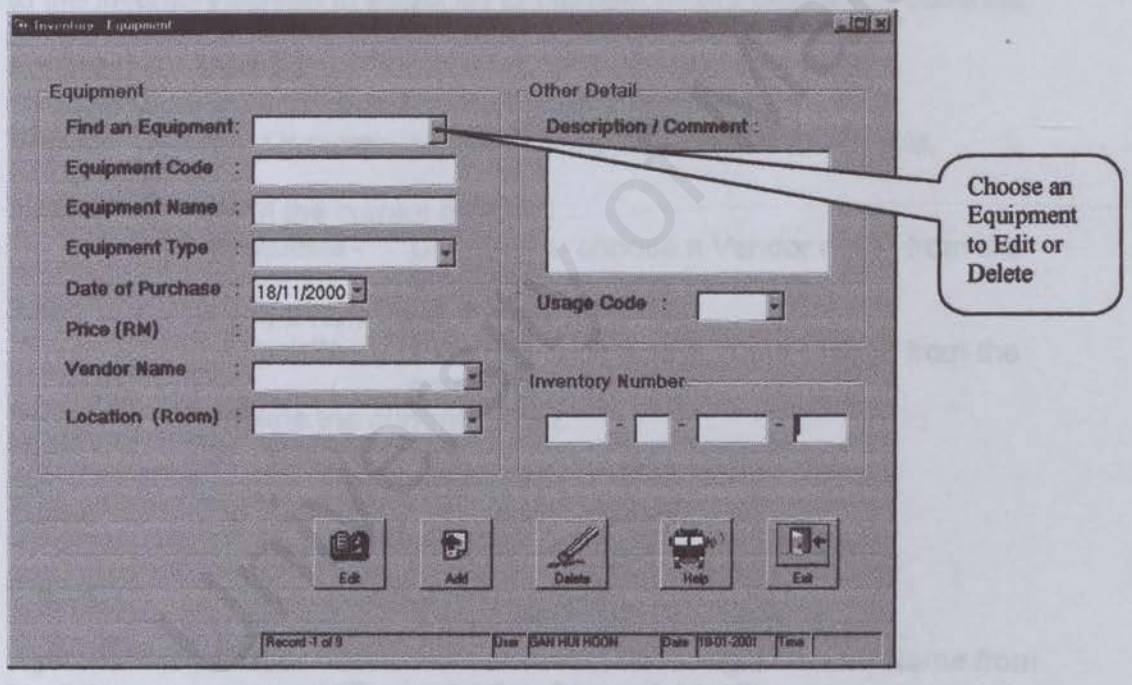

Figure 3.9 Inventory - Equipment

Figure 3.10 Error in Editing Record

Figure 3.11 Error in Deleting Record

If any of the message boxes do appear, you just have to press **OK** and return to the Inventory screen to select an Equipment record before you continue.

Inventory - Vendor

To keep Vendor's information. (See Figure 3.12)
Vendor – Person, Organisation that provide equipments to schools.

Edit - To edit the current record
Prerequisite - User has to choose a Vendor name from the combo box.
Add - To add a new Vendor record.
Prerequisite - User has to choose a Vendor name from the combo box.
Delete - To delete the current record.
Help - Help on current topic
Exit - Exit from current screen.
Save - To save the changes or the new added record.
Prerequisite - Edit or Add button has been clicked.
Cancel - To cancel your current operation.
Prerequisite - Edit or Add button has been clicked.

If you click the **Edit** or **Delete** button without choosing a Vendor Name from the list, a message as in Figure 3.13 or Figure 3.14 will appear.

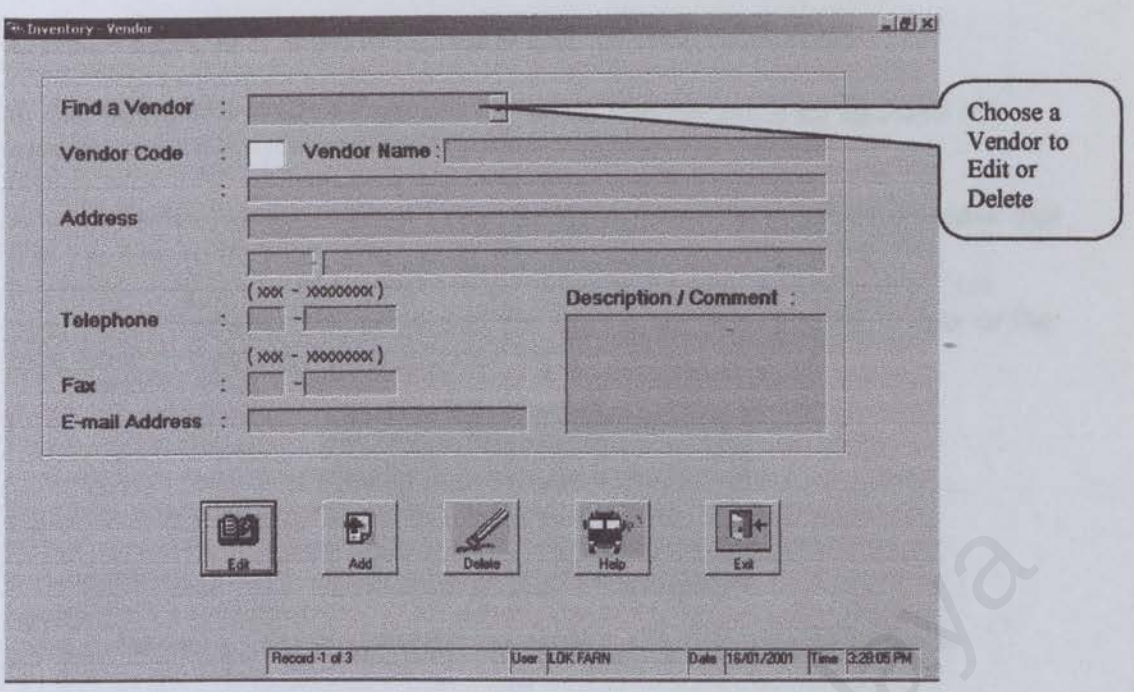

Figure 3.12 Inventory - Vendor

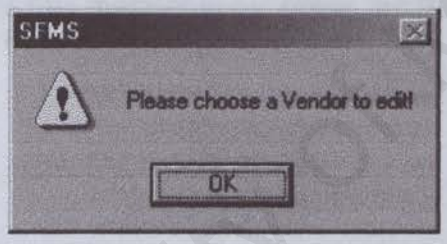

Figure 3.13 Error in Editing Record

Figure 3.14 Error in Deleting Record

If any of the message boxes do appear, you just have to press **OK** and return to the **Inventory** screen to select a Vendor record before you continue.

CHAPTER 4 MAINTENANCE SCHEDULING

In this module, you will be able to schedule maintenance for facilities in school.

After the scheduling has been done, SFMS will be able to inform the user that it is the time to do maintenance for the scheduled maintenance. The information given by the system is a message box as in Figure 4.1. This operational message will appear at the start-up of SFMS, when a user of the type **Staff** or **System Administrator** login.

Figure 4.1 Operation message indications it is the time to do maintenance.

Steps:

1. In order to schedule this maintenance; press the **Facilities Maintenance** button in the **Main Menu** (See Figure 4.2).

Figure 4.2 Facilities Maintenance Button

2. You have to choose the type of facilities to schedule its maintenance date. (See Figure 4.3)

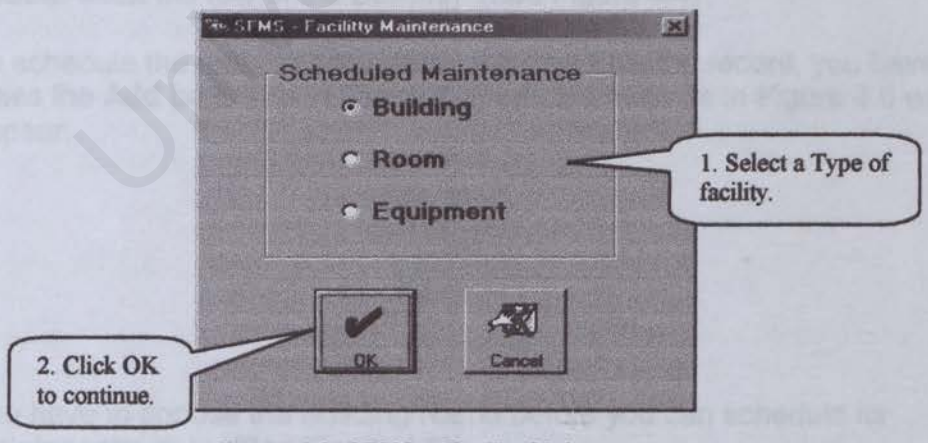

Figure 4.3 SFMS – Facility Maintenance

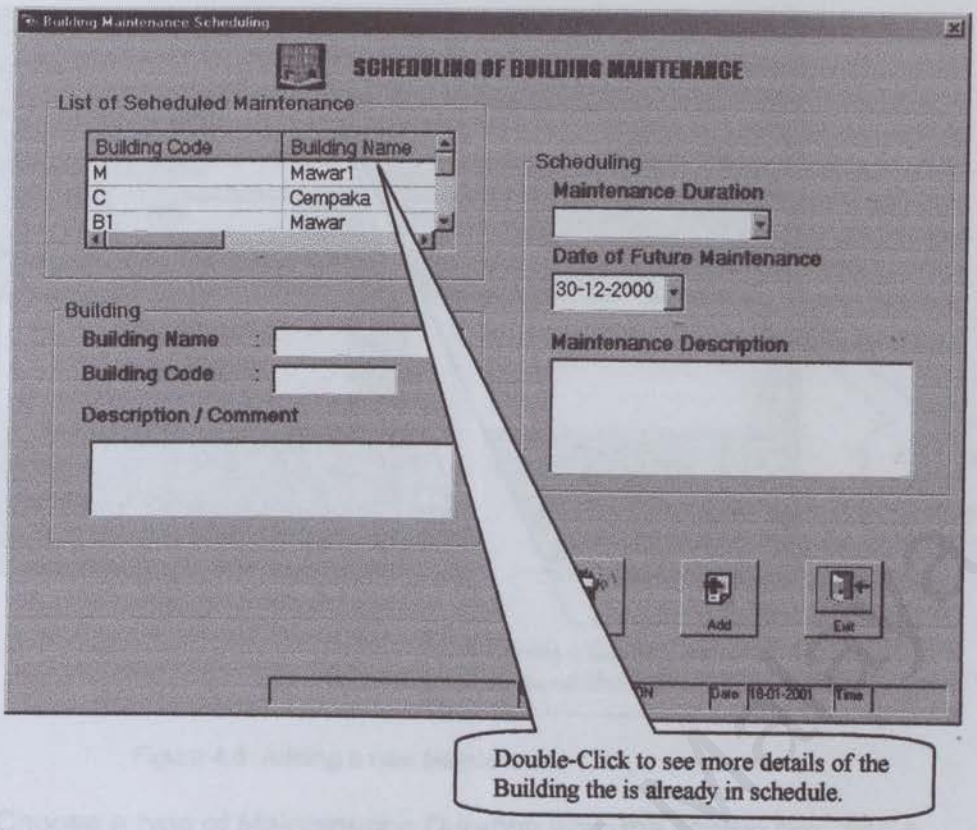

Figure 4.4 Scheduling of Building Maintenance

Steps (Facilities Maintenance – Building):

1. After you have chosen the type of facilities as Building, a screen as in Figure 4.4 will appear. Figure 4.4 shows all the Building that is already been scheduled previously.

2. To see full maintenance details for a particular building, you have to **double-click** the row of the Building. (See Figure 4.4)

3. To schedule the maintenance date for a new Building record, you have to press the **Add** button as in Figure 4.5, and a screen as in Figure 4.6 will appear.

Figure 4.5 Add Button

4. You have to choose the Building Name before you can schedule its maintenance date. (See Figure 4.6)

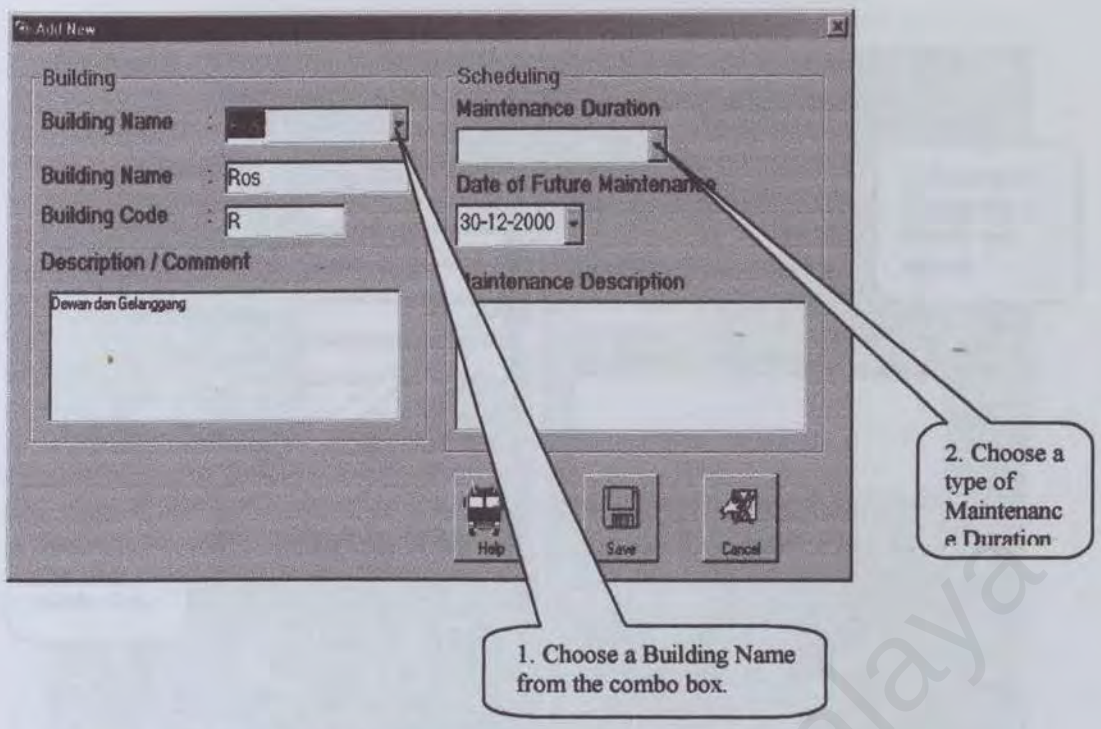

Figure 4.6 Adding a new Maintenance

5. Choose a type of Maintenance Duration from the combo box, and a corresponding date of future Maintenance will automatically appear. If you do not like the date generated, choose others from the combo box, and you can manually select the date of the Maintenance.

6. Press **Save** to save the new Maintenance record or **Cancel** to abort.

7. Press **Cancel** also, if you would like to go back to the **Scheduling of Building Maintenance** screen.

Steps (Facilities Maintenance – Room):

1. After you have chosen the type of facilities as **Room**, refer to **Steps (Facilities Maintenance – Building)** for more explanations, as scheduling for both **Building** and **Room** are the same in nature.

Steps (Facilities Maintenance – Equipment):

1. After you have chosen the type of facilities as **Equipment**, a screen as in Figure 4.7 will appear.

2. The volume of Equipments in school and the need to do maintenance for equipments are much greater. So, SFMS is designed to ease the user to do maintenance scheduling for these equipments. Please refer to Figure 4.7 for the steps.

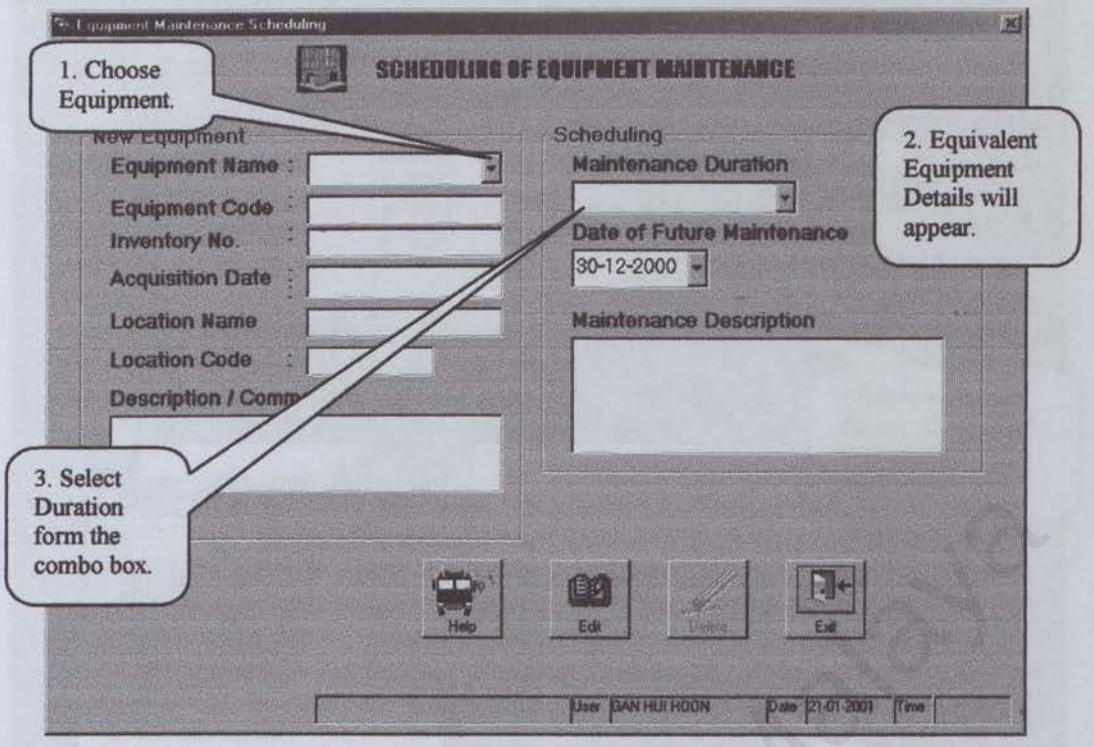

Figure 4.7 Scheduling of Equipment Maintenance

3. Once you select the Maintenance Duration from the combo box, SFMS will auto generate the Date of Future Maintenance.
   However, if you dislike the date generated, feel free to select a new date.

4. **Save** and **Cancel** button will be visible to your eye.

5. Provide **Description** for the maintenance.

6. Click **Save** to save the new Maintenance.

Steps (Editing Maintenance Records)

1. To edit the records in the Maintenance table, click **Edit** button as in Figure 4.8.

Figure 4.8 Edit Button

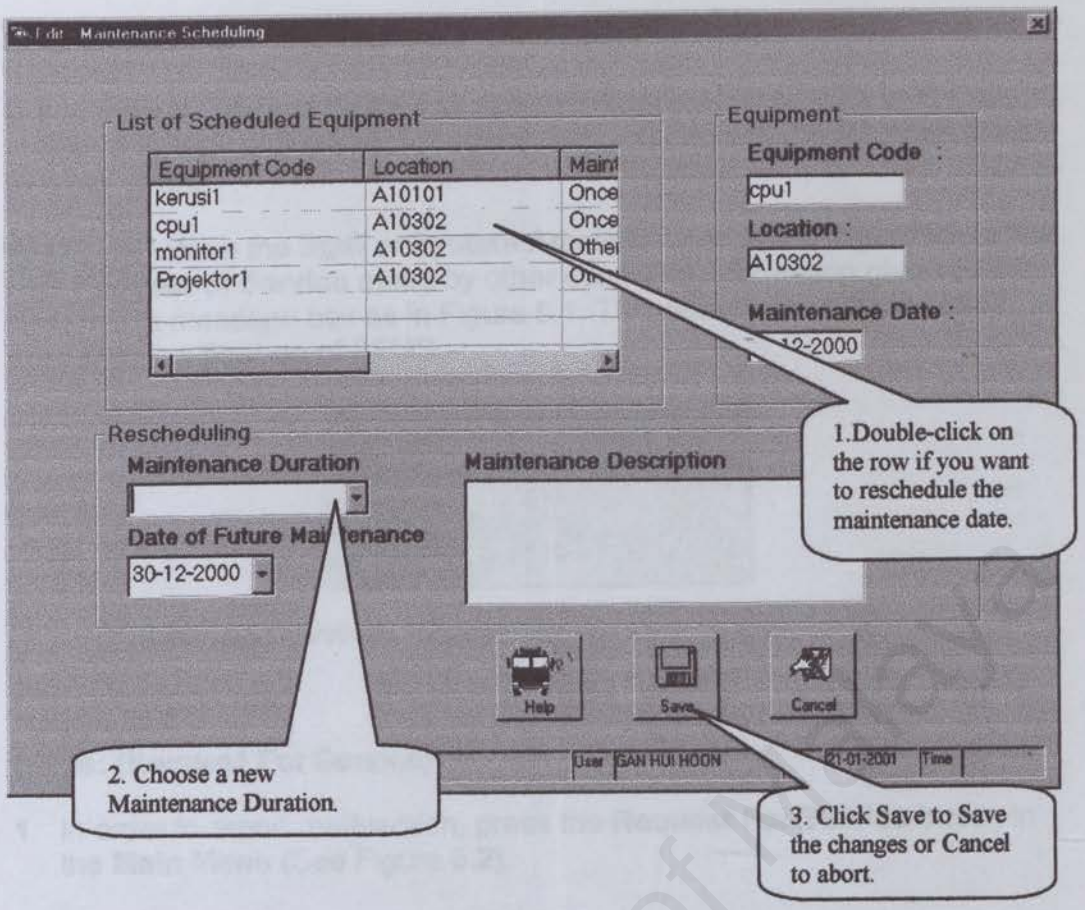

Figure 4.9 Editing Maintenance Scheduling - Building

2. After you have **double-clicked** on the record that you wish to edit (reschedule), equivalent maintenance details will appear.

CHAPTER 5 REQUEST FOR SERVICE

In this module, you will be able to report malfunction of facilities in the school and for the **Staff** or **System Administrator**, you can edit the **Request** made by other users.

SFMS will inform the **Staff** or **System Administrator** whom login if there is a new **Request of Service** made by other user. The information given by the system is a message box as in Figure 5.1. This operational message will appear at the Start-up of SFMS.

Figure 5.1 Operational Message indicates a new Request For Service

Steps: (Request For Service)

1. In order to report malfunction, press the **Request For Service** button in the **Main Menu** (See Figure 5.2).

Figure 5.2 Request For Service Button

2. As in the **Inventory** module, before you can proceed, you have to choose type of facilities.

3. Click **OK** to continue, **Cancel** to abort.

Request For Service - Building

1. If you choose **Building** as you facility type, you will come to screen as in Figure 5.3.

2. Choose a Building Name from the combo box. The equivalent Building Code will appear.

Steps: (Edit Request For Service)

Note: This process can only be done after the **Request** has been entertain.

1. To edit the Request For Service, click the **Edit** button on Figure 5.3

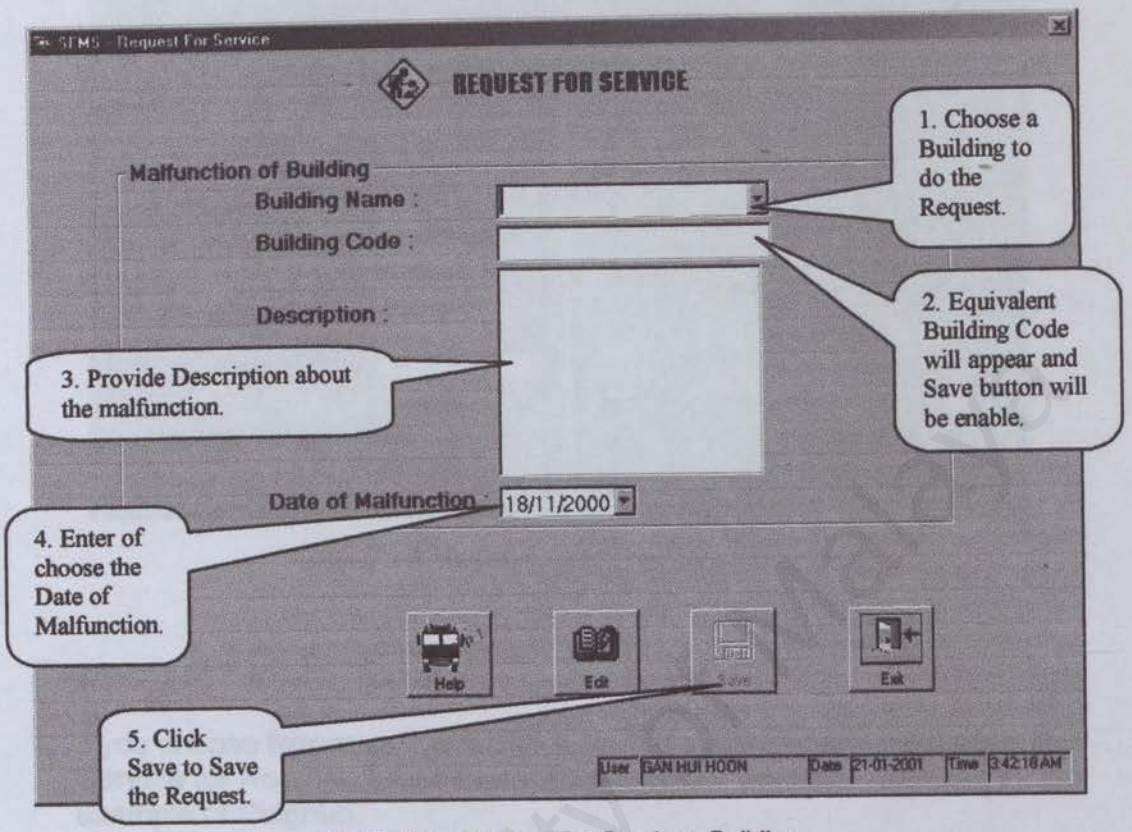

Figure 5.3 Request For Service - Building

2. **Double-click** on the List of Request For Service and the equivalent Request Detail will appear.

3. Choose either **Fixed**, or **Cannot Be Fixed**.

4. Press **Save** to save the editing.

5. After these processes, the operational message as in Figure 5.1 will not appear in the start-up of SFMS.

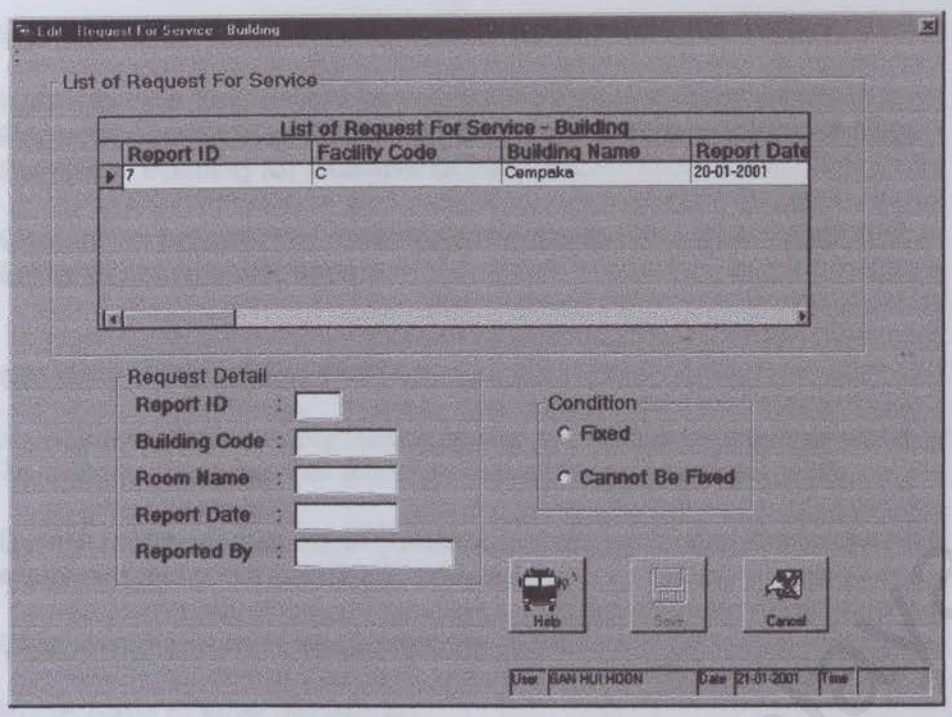

Figure 5.4 Editing Request For Service For Building.

Request For Service - Room

1. If you choose **Room** as you facility type, the steps to make a **Request For Service** for Room(s), please refer **Request For Service** – Building for example of Diagram.

2. Choose a Room Name from the combo box. The equivalent Room Code will appear.

Steps: (Edit Request For Service)

Note: This process can only be done after the Request has been entertain.

1. To edit the **Request For Service**, click the **Edit** button.

2. **Double-click** on the List of Request For Service - Room and the equivalent Request Detail will appear.

3. Choose either **Fixed**, or **Cannot Be Fixed**.

4. Press **Save** to save the editing.

5. After these processes, the operational message as in Figure 5.1 will not appear in the start-up of SFMS.

Request For Service - Equipment

1. If you choose **Equipment** as you facility type, the steps to make a **Request For Service** for Equipment(s), please refer **Request For Service** – Building for example of diagram.

2. Choose an Equipment Name from the combo box. The equivalent Equipment Code will appear.

Steps: (Edit Request For Service)

Note: This process can only be done after the Request has been entertain.

1. To edit the **Request For Service**, **click** the **Edit** button.

2. **Double-click** on the List of Request For Service - Equipment and the equivalent Request Detail will appear.

3. Choose either **Fixed**, or **Cannot Be Fixed**.

4. Press **Save** to save the editing.

5. After these processes, the operational message as in Figure 5.1 will not appear in the start-up of SFMS.

CHAPTER 6 TRACKING USAGE

In this module, the user will be able to track usage for each building or room in a school

1. When you **click Tracking Usage** button, you will see figure 6.2. You have to choose whether you want to track usage by Building or by Room.

Figure 6.1 Tracking Usage Button

3. **OK** to proceed, **Cancel** to abort.

Figure 6.2 SFMS – Tracking Usage

By Building

If your choice is By Building, the screen as in Figure 6.3 will appear.

1. To track building, choose a Building Name from the combo box.
2. The corresponding Building Code as well as the List of Room will appear automatically after you have chosen the Building Name.

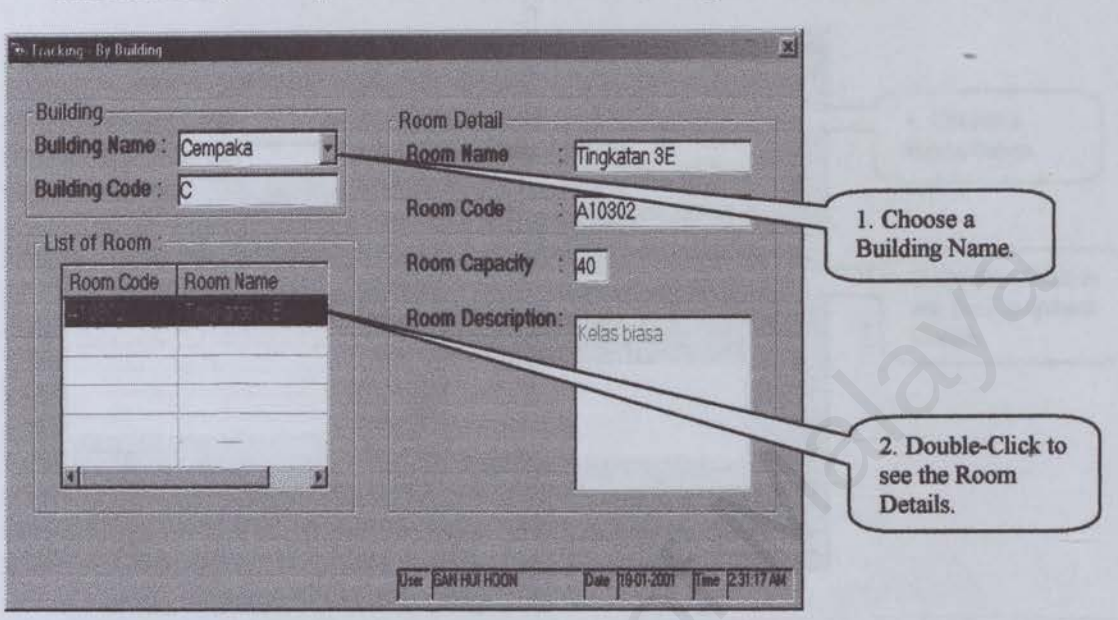

Figure 6.3 Tracking – By Building

3. If there is no Room record in this particular Building, as message box as in Figure 6.4 will appear.

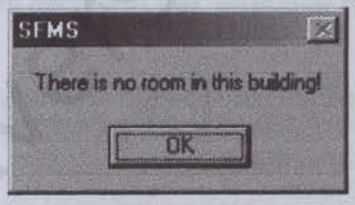

Figure 6.4 Message indicating no Room.

By Room

If your choice is **By Room**, the screen as in Figure 6.5 will appear.

1. To track Room, choose a Room Name from the combo box.

2. The corresponding Room Code as well as a list of all the Equipments located in the Room will appear after you have chosen the Room Name.

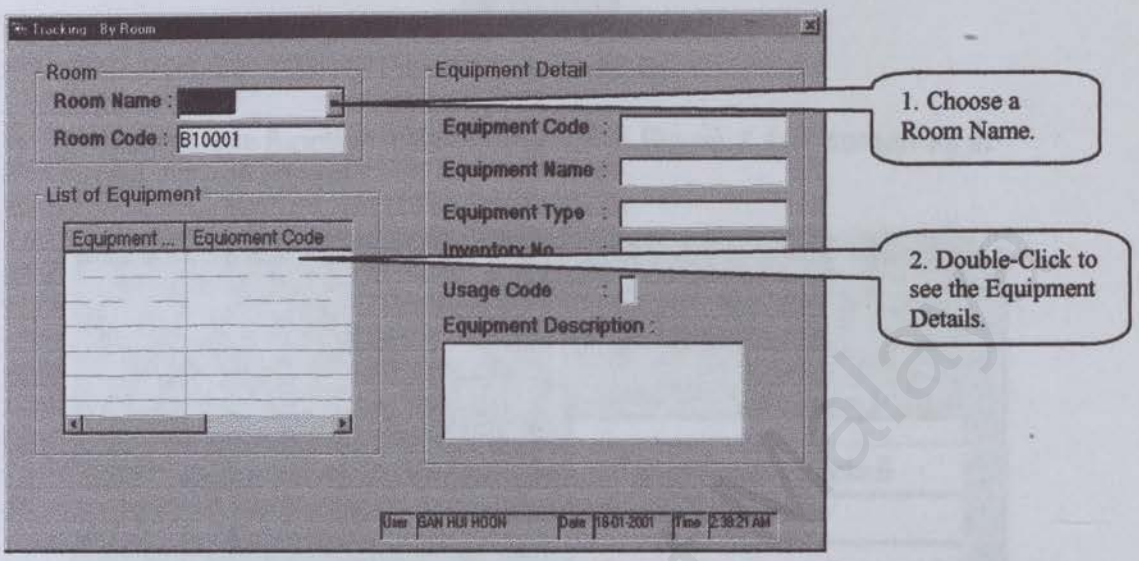

Figure 6.5 Tracking Usage – By Room

3. If you would like to see more details particular Equipment, **double-click** anywhere at the row of the List of Equipment, and automatically, the corresponding Equipment Detail will appear.

4. If there is no Equipment record in the particular room, a message box as in Figure 6.6 will appear.

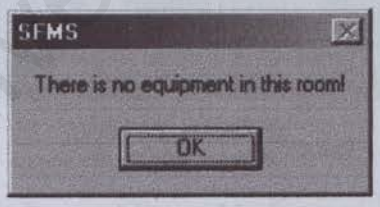

Figure 6.6 Message indicating no Equipment.

CHAPTER 7 REPORT

This module can generate and print report for School Administrator. The report printed out can help the management people to plan the usage of facilities in the future.

Figure 7.1 Report Button

1. To access this function, **click** button as in Figure 7.1. A screen as in Figure 7.2 will appear.

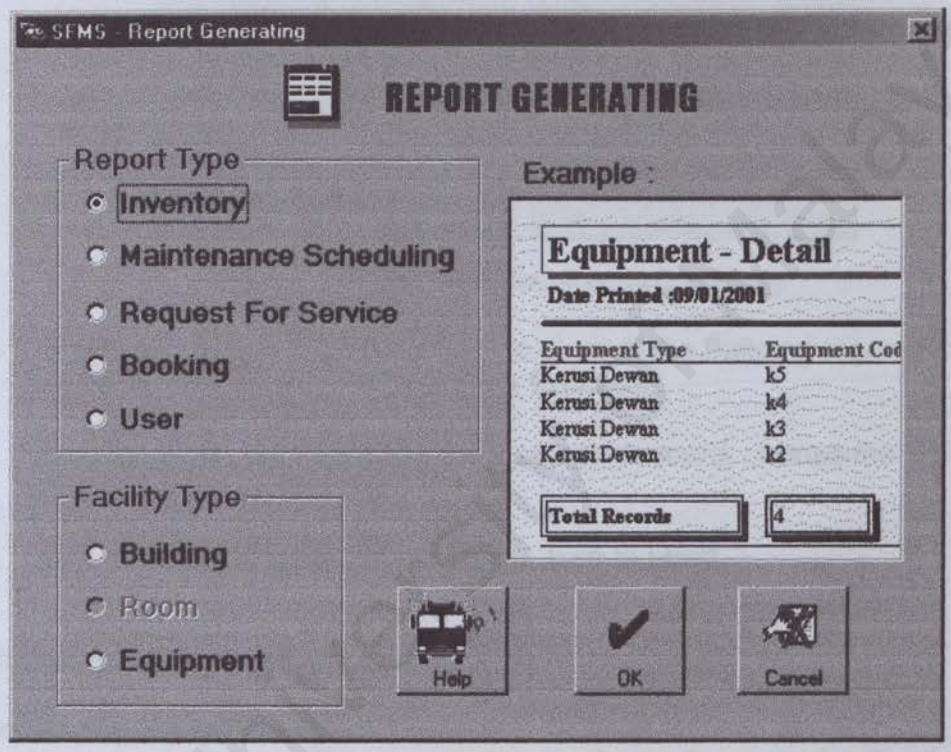

Figure 7.2 Generate Report

1. You have to choose a Type of Report before you can proceed.

2. After choosing a Type of Report, you have to choose the Type of Facility. Press **OK** to continue, or **Cancel** to abort.

Inventory – Building Report

Generating Building Record

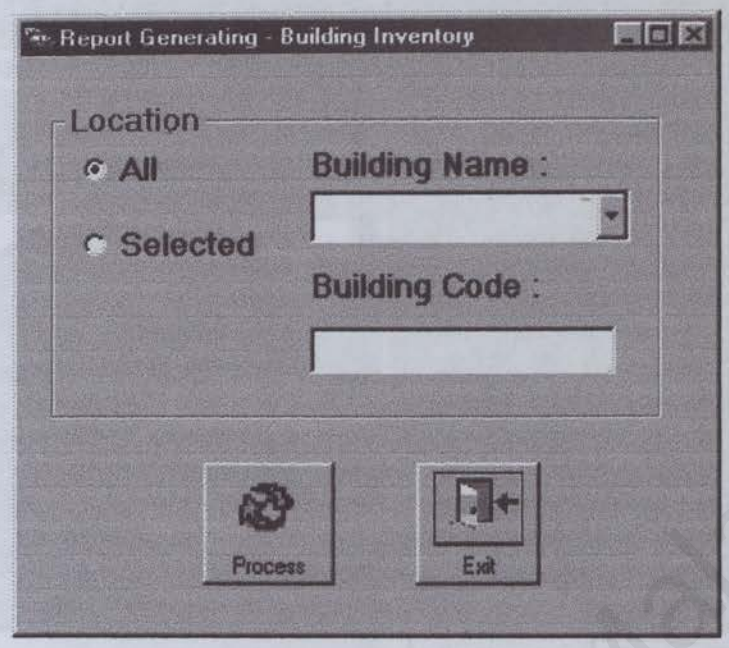

Figure 7.3 Report Generating – Building Inventory

1. Referring to Figure 7.3, if you want to view all the Building records in the database, choose **All** and then **click Process**.

2. However, if you only want to view a selected record, choose the Building Name from the combo box.

3. **Click Process** to generate the report, **Exit** to abort.

Inventory – Equipment Report

Generating Equipment Record.
Equipments records can be generated according to 3 criteria. (i.e. **Equipment Type**, **Location**, **Date of Purchase**)

For example of generating Equipment report according to **Equipment Type**, refer 7.4 for the steps.

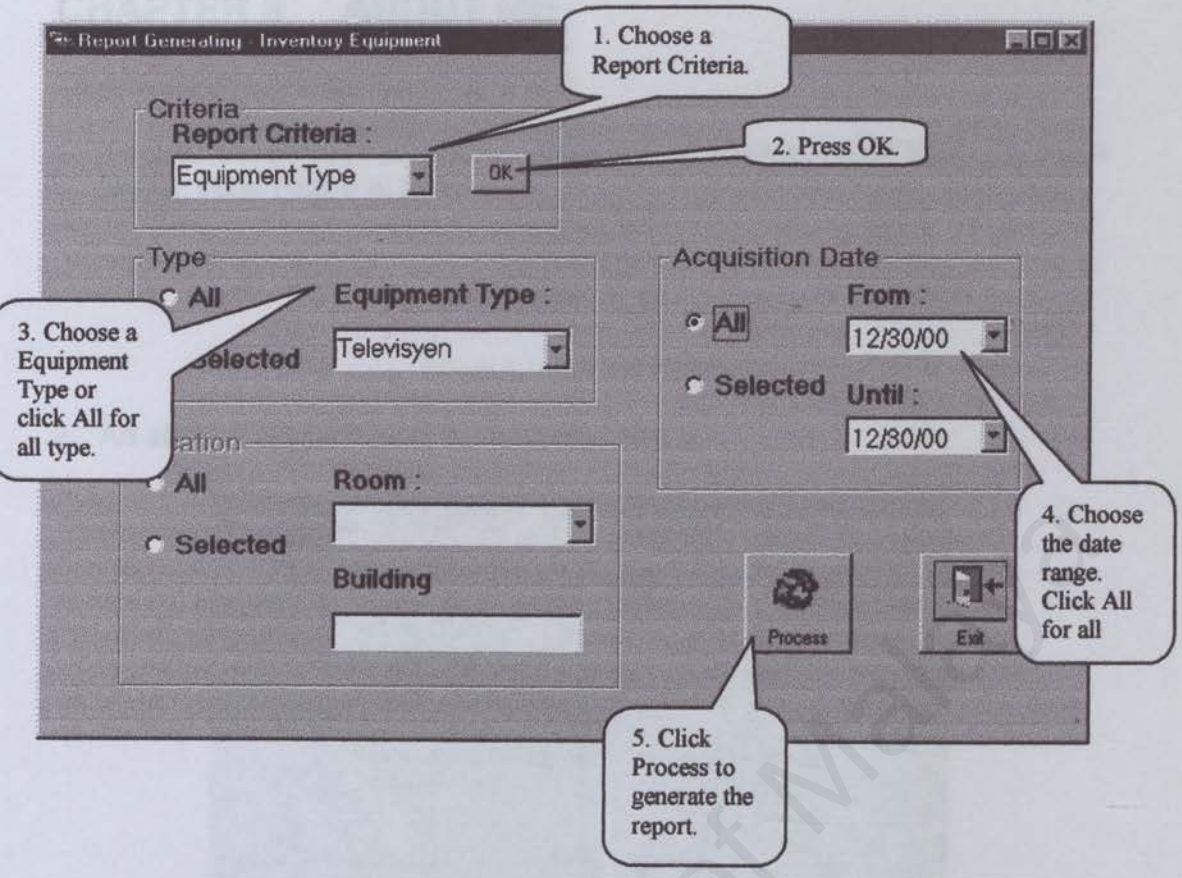

Figure 7.4 Generating Report For Building Inventory

Note: For other reports, please refer to the Help in SFMS.

CHAPTER 8 ABOUT ME

To provide information about SFMS.

1. To gain more information about SFMS, please **click About Me** in the **Main Menu**. (See Figure 8.1).

Figure 8.1 About Me Button

2. An about dialog will appear as in Figure 8.2.

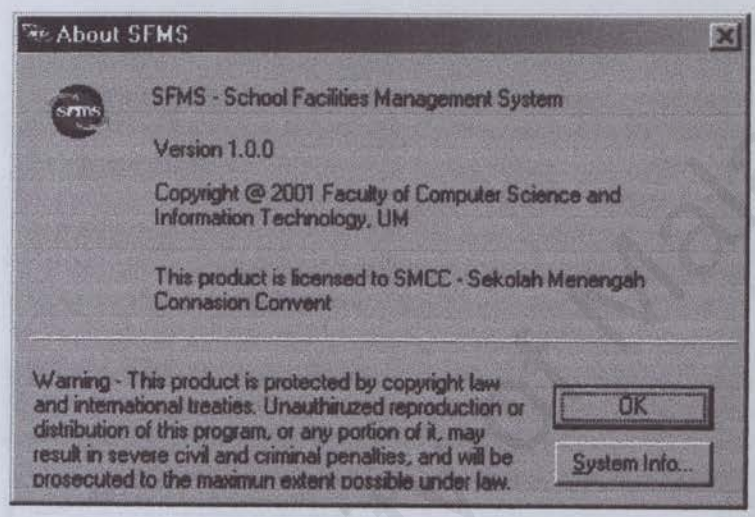

Figure 8.2 About SFMS

CHAPTER 9 HELP

SFMS provide Online Help to its users.

1. If you needed help in doing your transaction, please press **F1** or **Click** the **Help** button as in Figure 9.1.

Figure 9.1 Help Button

2. A menu like in Figure 9.2 will appear.

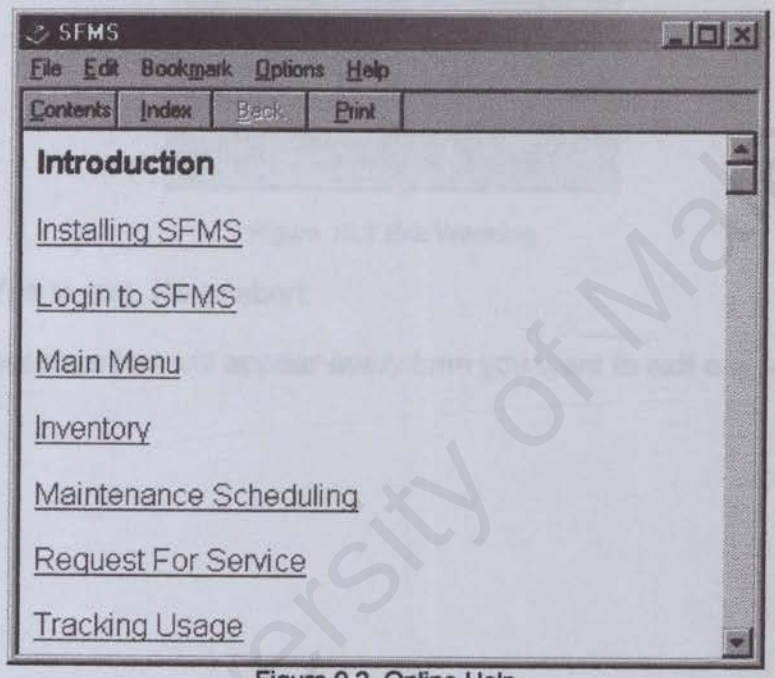

Figure 9.2 Online Help

CHAPTER 10 EXITING

1. If you would like to exit this program, you just have to **click** the **Exit** button in the **Main Menu**.

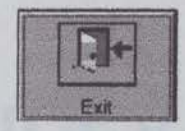

Figure 10.1 Exit Button

2. A message box as in Figure 10.1 will appear.

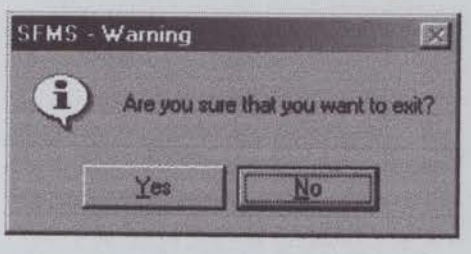

Figure 10.1 Exit Warming

3. Click **Yes** to exit, **No** to abort.

4. This message box will appear every time you want to exit any window in SFMS.

BIBLIOGRAPHY

Internet Resources

[1] **Ashok Ramachandran**, http://www.ashok.pair.com/

[2] **Intergrow FMS**, http://www.hetco.com/faciliti.htm

[3] **Indiana University,** http://www.iupui.edu

[4] **Microsoft Cooperation,** http://www.microsoft.com

[5] **Precision Inventory Management System,** http://education.software-directory.com

[6] **Rhodes Facilities Management**, http://calendar.rhodes.edu/]

[7] **Shadyside Hospital in Pittsburgh,** http://www.aperture.com

[8] **University of Florida Health Science Center,** http://www.hsc.ufl.edu

[9] **Webopedia,** http://webopedia.internet.com

Books And Publications

[10] Connolly, Thomas M.; Begg, Carolyn E.; Strachan, Anne D.; "**Database System**", Harlow, England, 1996.

[11] Fertuck, Len, **"System Analysis & Design"**, Wm. C.Brown Communications Inc, 1995.

[12] Hawryszkiewycz, Igor; **"Introduction to System Analysis and Design (4th Edition)"** Prentice Hall Inc., 1997.

[13] Kendall, Kenneth E. and Kendall, Julie E. **"System Analysis and Design 4th Edition"**, Prentice Hall International Inc. 1996.

[14] Kroenke, David M.; **"Database Processing Fundamentals, Design, and Implementation 6th Edition"**, 1998.

[15] Pfleeger, Shari Lawrence **"Software Engineering Theory and Practice"**, Prentice Hall, Inc. 1998.

[13] Senn, James A.; **"Analysis and Design of Information System, 2nd Edition."** McGraw-Hill Publishing Company, 1989.

[11] Sommerville, Ian; **"Software Engineering (5th Edition)"**, Addison Wesley USA, 1995.

[2] Weinschenk, Susan; Jamar, Pamela; and Yeo, Sarah C.; **"GUI Design Essential for Windows® 95, Windows® 3.1, World Wide Web"**; John Wiley & Sons, Inc. 1997.

[DEU79] Deutsch, M, "Verification and Validation, "in Software Engineering (R. Jensen and C. Tonies, eds), Prentice Hall, 1979, p.p 329-408.