

Nama Pelajar : Lok Farn
Kad Matrik : WEK 98098

Penyelia : Pn. Abrizah Abdullah
Moderator : En. Teh Kang Hai

Kod Subjek : WXES 3182
Tajuk : Sistem Pengurusan Kemudahan
Sekolah

Tarikh : 10 January 2001

Declaration

Abstract

I declare this project is my own work and has no submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

1.1	Introduction to FMS	1
1.2	Purpose of project	1
1.3	Objectives of project	2
1.4	Scope of project	2
1.5	Timeline of project	3
1.6	Software requirement	4
1.7	Hardware requirement	4
1.8	Project scheduling	5
1.9	Summary of chapter 1	6
Chapter 2: Literature review		
2.1	Overview of Booking System In SPM	7
2.1.1	What is a Computerized Booking System?	7
2.1.2	Overview of Booking System	7
2.2	Web-based / Online Booking System (WABS)	8
2.2.1	Service Provided by WABS	8
2.2.2	Operating Hours	9
2.2.3	Conditions On Using the WABS	10
2.2.4	Feedback on Using	10
2.3	Comparison Between FMS and WABS	10
2.4	Specific Online Booking System at Chinese University of Hong Kong (CUHK)	11
2.4.1	System Objectives	11
2.4.2	Registration	12
2.4.3	Booking Procedures	12
2.4.4	Operating Hours	13
2.4.5	Time Schedule	13

Content

Abstract	(vi)
Acknowledgment	(vii)
Table of Tables	(vii)
Table of Figures	(viii)
Chapter 1 : Introduction	16
1.1 Introduction to FMS	1
1.2 Purpose of project	1
1.3 Objectives of project	2
1.4 Scope of project	2
1.5 Importance of project	3
1.6 Software requirement	4
1.7 Hardware requirement	4
1.8 Project scheduling	5
1.9 Summary of chapter 1	6
Chapter 2 : Literature review	7
2.1 Overview of Existing Systems In Schools	7
2.1.1 What is a Computerized Based Management System?	7
2.1.2 Overview of Booking Procedures	7
2.2 Workstation Advance Booking System (W.A.B.S.)	8
2.2.1 Services Provided By W.A.B.S.	8
2.2.2 User Booking Limits	9
2.2.3 Guidelines On Using the W.A.B.S.	10
2.2.4 Feedback	10
2.3 Comparison Between FMS and W.A.B.S.	10
2.4 Sports Facilities Booking System of Chinese University of Hong Kong (CUHK)	11
2.4.1 Sports facilities	11
2.4.2 Regulations	12
2.4.3 Booking Procedures	12
2.4.4 Opening Hours	13
2.4.5 Time Schedule	13

Content

2.5	Comparison Between FMS and Facilities Booking System (FBS) of Chinese University of Hong Kong (CUHK)	14
2.2	Summary of chapter 2	15
Chapter 3 : Methodology		
3.1	Methodology of System Development.....	16
3.2	Fact-Finding Techniques.....	17
3.2.1	Problems and Solutions.....	18
3.3	Requirement Analysis.....	20
3.3.1	Functional Requirement.....	20
3.3.2	Non-Functional Requirement	21
3.4	Selection of Programming Language.. ..	23
3.4.1	Programming Languages	23
3.4.2	Database	27
3.4.3	Database Connectivity	29
3.5	System requirement	30
3.6	Summary of chapter 3	30
Chapter 4 : System Design		
4.1	Introduction of System Design	32
4.2	Design of Program	32
4.2.1	Data Flow Diagram (DFD)	32
4.3	Design of User Interface	36
4.4	Design of Database	47
4.4.1	Conceptual Design.....	47
4.4.2	Table Structure	49
4.6	Expectation of Outcome.....	55
4.7	Summary of chapter 4.....	55
Chapter 5 : System Implementation		
5.1	Introduction.....	56
5.1.1	Visual Basic Data Access Techniques	56
5.1.1.1	Data Control.....	56

Content

5.1.1.2	Remote Data Control.....	57
5.1.1.3	Data Access Object (DAO).....	57
5.1.1.4	ActiveX Controls.....	57
5.1.1.5	Remote Data Object (RDO).....	57
5.1.1.6	ODBC API.....	58
5.1.1.7	VB SQL (Visual Basic Library for SQL Server)	58
5.1.1.8	ODBC Direct.....	58
5.1.1.9	OLE DB / ADO.....	58
5.1.1.10	Summary	59
5.2	Coding.....	59
5.2.1	Code documents	60
5.2.2	Data integrity	60
5.2.3	Modularity	60
5.2.4	Chronology.....	60
5.2	User Interface	61
5.4	Generating FMS.....	61
5.4.1	User.....	61
5.4.2	Administrator	61
5.4.3	Staff / Teacher	62
5.4.4	System Rules	62
5.4.5	Password	62
5.4.6	Attempt of login.....	63
5.4.7	Maximum booking for user	63
5.4.8	Deleting user	63
5.4.9	Attempt of login.....	63
5.4.10	Viewing user.....	63
5.4.11	Change password.....	63
5.4.12	Editing user	64
5.4.13	Editing / adding booking.....	64
5.4.14	Customizing new / old facility	64

Content

5.4.15 Security in editing / adding mode	64
5.4.16 Cancel changes in editing / adding mode	64
5.4.17 Deleting record.....	65
5.4.18 Expiration of booking.....	65
5.4.19 Record navigation.....	65
Chapter 6 : System Testing	
6.1 Introduction	66
6.2 Approaches to testing.....	66
6.2.1 Unit testing.....	66
6.2.2 Module testing.....	66
6.2.3 Integration testing.....	67
6.2.4 System testing.....	67
6.2.5 Security testing.....	68
6.3 Testing and debugging	68
6.3.1 Test data	68
6.3.2 Test procedures	69
6.3.3 Testing techniques	69
6.3.3.1 White box testing	69
6.3.3.2 Black box testing	69
6.4 Summary	70
Chapter 7 : System Evaluation	
7.1 Introduction	71
7.2 Project Problems and Solutions	71
7.2.1 Data gathering and analysis stage	71
7.2.1.1 Difficulty in choosing appropriate development tools	71
7.2.1.2 Scope of system	71
7.2.1.3 Time slot in facility booking	72
7.2.1.4 Scope of system	72
7.2.2 Implementation and testing stage	72
7.2.2.1 Development tools	73

Content

7.2.2.2	Resources sharing	73
7.2.2.3	Database integrating	73
7.2.2.4	Debugging message.....	74
7.2.2.5	Log files	74
7.3	System Strength	74
7.3.1	Simple and user friendly interface	74
7.3.2	User ID, password and security.....	75
7.3.3	Reliable system with effective error recovery	75
7.3.4	Query based mode for record retrieval.....	75
7.3.5	System transparency	75
7.4	System Limitations	76
7.4.1	Operating system limitations.....	76
7.4.2	Minimal system rules.....	76
7.5	Future Enhancement	76
7.5.1	Encryption of password	76
7.5.2	Cater to multi-platform operating system	76
7.5.3	More administration task	77
7.5.4	Integrating mailing capabilities.....	77
7.5.5	Enable web-based functionality	77
7.5.6	Enable map-based booking	77
7.6	Knowledge and experience gained.....	78
7.7	Overall conclusion.....	78
Glossary	80
References	81
User Manual	85
Appendix		

Facility Management System (FMS) is a window-based system developed with the objectives of implementing the task of daily operation needs of small to medium size secondary school. It supports the inventory, usage and maintenance tracking, booking functionality of a typical secondary school. To maximize the effectiveness, FMS provides a computerized interface that greatly simplify information generation which enables decision maker to creatively approach, analyze and uncovering transaction problems and opportunities.

Since the local architecture forms the core of the system, research was conducted into the local computing concepts. Through the course of FMS development, it is proven that management rules can be encapsulated into the local information system.

The prototyping approach is selected for the development process of FMS since it supports rapid application development and reduces the unforeseen risks. Software engineering principles are applied throughout the development phases of system analysis, design, testing and implementation. FMS is developed using Visual Basic 6.0, an object oriented and visual-programming language on the Windows platform utilizing data base created and stored by Microsoft Access 97. It is believed that this system can be migrated to on-line data base information system in future.

Finally, problems are inevitably encountered due to the complexity and fast changing of various aspects of system requirement, techniques and solutions are used to address these problems. Although, FMS limited in certain aspects, it serves as a foundation for the future development of similar systems.

Acknowledgment

First of all, I want to express my special thanks to my supervisor, Pn. Abrizah Abdullah for her patience and guidance within this period.

Thank you to the teachers who willing to spend their time to fill in the questionnaire. Their assistance is very much appreciated.

To my family and friends, thank you for your advice.

Table of Tables

Table 1.1	:	Software Lists	4
Table 2.1	:	Facility Available in User Area.....	10
Table 4.1	:	Components of Data Flow Diagram	33
Table 4.2	:	Principles of HCI.....	36
Tables of FMS:			
Figure 4.4		cpr - change password records	49
Figure 4.5		dept – department table.....	49
Figure 4.6		fbr - facility booked record.....	49
Figure 4.7		fhf - facility history file	50
Figure 4.8		fpf - facility personal file.....	50
Figure 4.9		ftf - facility type file	50
Figure 4.10		pat - power allocation table.....	51
Figure 4.11		pos – position table.....	51
Figure 4.12		sex – gender table.....	51
Figure 4.13		upf - user personal file.....	52
Figure 4.14		utf - user type file	52
		Building.....	53
		Equipment.....	53
		Room.....	54

Table of Figures

Figure 1.1	:	Progress Chart.....	5
Figure 3.1	:	SDLC Proposal for FMS.....	17
Figure 4.1	:	DFD for booking module.....	34
Figure 4.2	:	DFD for Registration module.....	35
Figure 4.3	:	Interface for logon.....	37
Figure 4.4	:	Interface for Add New User.....	37
Figure 4.5	:	Interface for Delete User.....	38
Figure 4.6	:	Interface for View User.....	39
Figure 4.7	:	Interface for User Change Password.....	40
Figure 4.8	:	Interface for Edit User.....	41
Figure 4.9	:	Interface for General Facility Booking.....	42
Figure 4.10	:	Interface for Special Facility Booking.....	43
Figure 4.11	:	Interface for Equipment Booking.....	44
Figure 4.12	:	Interface for Facility Customizing.....	45
Figure 4.13	:	Interface for School Map Viewing.....	46
Figure 4.14	:	ER Diagram of Facility Booking Module.....	48

1.1 Introduction to FMS

This is a thesis project titled *Facilities Management System (FMS)*. The systems primarily handle the daily transactions that involved the uses of the facilities in a secondary school, such as inventory managing, facilities booking and tracking and etc.

The database is now such an integral part of our day-to-day life. FMS is aimed at developing a computerized database system to manage the facilities in a school. Through the application program written it can handle data entry, file maintenance and generation of a fixed set of specific reports.

The database is one of the key components in the success of the project. Full-featured relational database management system is used such as Access.

However, computerized school is still at its infantile stage in Malaysia. Since most of the schools are expanded from year to year, it is need to set up a computerized management system to manage the school. Furthermore the manual filling system had significant problems, mainly the amount of data redundancy present and program data dependence.

It will be used to track the usage and maintenance of all kinds of a school's facilities such as hall, classrooms, stores, hostel, workshops, laboratories and some special rooms. The system will also generate the maintenance and servicing schedule of hardware like computer, printer and etc. Besides, the system will manage the inventory, fixed asset and other facilities with its built-in reporting system. It provides the facilities booking service to academic and support staff, as well as students.

1.2 Purpose of the Project

The main purpose of this project is to provide secondary school with a computerized facilities management system. The implementation of this project can widely reduce the redundant use of paper.

Besides, the project is able to identify the facility components and assets that are important to their ongoing effective operation.

1.3 Objectives of the Project

The objective of this project is to create a window-based FMS to provide timely and useful information for facility inventory control, usage tracking and booking to improve operational efficiency. The project attempts to design a management system that advances the computer technology to the average school practice at a much-reduced cost. Specifically the objectives are to:

- (I) Make the school's management smoother.
- (II) Make the management of facilities relatively more effective and faster.
- (III) Meet the need of the school in managing transactions involving facilities.
- (IV) Meet the need of the users either he is a teacher, a student or a clerk.
- (V) Reach the demand of the users on the system's functions.
- (VI) Provide clearer and nicer view of reports.
- (VII) Reduce staff's workload and redundant jobs in respect of processing the school management.
- (VIII) Enable user check and book desired facilities easily.
- (IX) Reduce the time and manpower as well as other resources spent on old fashion paper work.
- (X) Eliminate misunderstanding between users since the system provides a view of current facilities booking states.
- (XI) Manage facilities' information efficiently and effectively.

1.4 Scope Of Project

Typically, the implementation of the FMS covers the entire school, since the applications within it are school-wide solutions and should therefore extend across the whole school.

Furthermore, the FMS is a host-based system, which means it does not required super-equipped hardware to implement. This is also preventing the school from disposing the existing equipment. This surely will save the expenditure of school budget on administration and buying new hardware.

1.6 Hardware Requirements :

The hardware recommended to develop FMS are :

- 166 MHZ Pentium Processor (Pentium class recommended)
- 512K Pipeline Burst Cache
- MB RAM minimum (64MB recommended)
- 12X CD-ROM Drive
- 50MB available disk space
- Other standard desktop PC components

Remark: Any hardware other that stated above is acceptable if it is over the qualification as mentioned.

1.7 Software Requirements

During the process of FMS development, a vast array of software tools will be used.

Table 1.1 depicts all of the software used to develop the system :

Software	Module	Description
Microsoft Windows 95	System requirements	Operating system
Microsoft Access 97	Database	DBMS
Microsoft Visual Basic 6.0	System development	To develop the client application
Microsoft Word 97	System documentation	To create rich text format file used for help file compilation and generation

Table 1.1: Software Lists

Remark: Any higher version of Windows® 95 and Word® 97 is acceptable.

1.7 Project Scheduling

To archive the project objective, a project schedule was planned to manage the time and activities needed to be accomplished. The overview of the schedule is show as below :



1.8 Summary Of Chapter 1

A summary of the whole chapter in the document is listed:

Chapter 1 :

This chapter summaries the aims and objectives of this project. It contains the overview of the project introduction, objective, scope and the project schedule.

Chapter 2 :

Explanation of the researches involved during the development of the FMS and related technology studies to implement the system.

Chapter 3 :

Description of the development methodology, information gathering technique, the system requirement and analysis.

Chapter 4 :

Concern with the design involved in building FMS. They include system functionality design, database design, process design as well as the user interface design.

2.1 Overview of Existing Systems in Schools

Initially, reading on the similar topic is being carried out to help build up a general understanding and knowledge on the work area. However, most of the information regarding the system is obtained through reference to similar facility management system found on the Internet) and questionnaire with the appropriate personnel. Next observation and record inspection are conducted to verify what was revealed in the interview.

Nowadays, most of the schools in Malaysia are still using traditional file-based system in managing the school facility booking service. The general trend is where computer might be used only to create a simple database for inventory.

Therefore, this chapter will be carried out by comparing the facility management systems found on the Internet with FMS; extracting the gist and summarizing; and rewriting them in a more structure way.

2.1.1 What is a Computerized Based Management System?

Computerized based management system (CBMS) is a management system that uses computer systems, devices and technology. A CBMS consists of hardware, software, a database, people and procedures. Below are the elements contained in a CBMS:

- Hardware consists of computer equipment to perform input, processing and output activities.
- Software consists of program and instructions given to the computer
- Database is an organized collection of facts and information.
- People are the most important element in most computer-based management system. Management system personnel include all the people who manage, run, program and maintain the computer system.

2.1.2 Overview of Booking Procedures:

In general, below is how the facility being managed based on observation and discussion with teachers:

1. Teacher, staff and administrator make a booking.
2. System will check if the facility is available to be booked.

3. System is either rejects or accepts the booking.
4. After a certain period, teacher or student returns the key to the clerk.

Remark: Process 1 is subject to the school. Some of the school might not practise this process in their school.

2.2 Workstation Advance Booking System (W.A.B.S.)

This section will cover the W.A.B.S. in four sub-sections, which are user booking limits, services provided, guidelines on using the system and feedback. Through the Workstation Advance Facility Booking System (W.A.B.S.), users can on-line reserve the User Area facility without paying a visit to the laboratory. One can reserve the following facilities by using the system:

- PCs
- Macintoshes
- Text/Image Scanner
- Optical Mark Reader
- Tape Processing Workstation

The Workstation Advance Booking System is composed of three sub-systems, which performs the following functions:

- Signon sub-system - validates user information against a master database and refuses signon request from others if the workstation has already been reserved.
- Booking sub-system - provides a user-friendly interface to assist you in booking of workstations and determines validity of your booking based on the booking criteria (refer to “user booking limits” in section 2.1.2)
- Database sub-system - stores the booking records.

2.2.1 Services Provided By W.A.B.S.

Besides, one can access the Workstation Advance Booking System and make the following requests on-line:

- Listing out available workstations;

- Making a booking request;
- Enquiring reserved workstations; and
- Cancellation of booking.

Here, are the functions and details of each of them:

i. Listing out Available Workstations

- Select the location, type of workstation, date and time you want to reserve;
- Click the "Query" button to submit the query.

ii. Making a Booking Request

- Select the location, type of workstation, date, time and workstation number you want to reserve;
- Type in your computing id and password on 'Mailserv' for verification;
- Click the "Submit Booking" button to submit the request.
- Details of the booking will be return if your booking is successful.

iii. Enquiring Reversed Workstations

- Type in your computing id and password on 'Mailserv' for verification;
- Click the "Query Booking" button to submit the request; and
- Details of all your booking will be displayed.

iv. Cancellation of Booking

- Select the location, type of workstation, date, time and workstation number you have reserved;
- Type in your computing id and password on 'Mailserv' for verification;
- Click the "Cancel Booking" button to cancel the reservation.

2.2.2 User Booking Limits

At this section, the original schedule is truncated to appropriate level due to its excessive length. It is about limitation on the usage of each workstation.

One can reserve a workstation based on the criteria set forth below:

<i>Workstation</i>	<i>Operating hours</i>	<i>Session Time (min.)</i>	<i>Maximum Sessions per day</i>
PC	24 hours	60	1
Tape Processing Workstation	10:00am-4:00pm*	60	1

Table 2.1: Facility Available in User Area

* Operating hours will be extended when there is enough staff.

2.2.3 Guidelines On Using the W.A.B.S.

One's booking request will not be honored if the number of booking request exceeds the limit of that workstation (Refer to "User Booking Limits" in section 2.1.2).

If one has reserved a workstation in advance, other user cannot access this workstation in the first 10 minutes of your booked session. However, if you have not accessed this workstation during this period, it will be re-assigned for other user.

2.2.4 Feedback

The W.A.B.S. also provides the service of feedback for users. Besides, anyone can make enquiry to the help desk on-line.

2.3 Comparison Between FMS and W.A.B.S.

The primary feature that distinguishes W.A.B.S. from FMS is it is a workstation booking system whereas FMS is a facility booking system.

W.A.B.S. is an online booking system without call in to the laboratory. It consists of three sub-systems. In FMS, facility booking is one of the main modules. Facility booking module contains sub-modules, which perform different tasks (Refer to section 2.1).

W.A.B.S. provides several services such as listing out available workstations for booking, making a booking request, enquiring reserved workstations and cancellation of booking. As can be seen from the list mentioned, it does not allow users modify a booking. However, in addition provides the basic services (e.g.: make, cancel and check a booking), booking system in FMS also provides the service where users can modify a booking.

Besides, time limit for user to cancel a booking is not mentioned. This might cause inconvenience if user cancels a booking at the very last minute. The system administrator does not take care of this point might because of it is just a workstation booking system. The facilities offered for booking are small and easy to manage. This could be one of the reasons why no severe regulations are established to control the use of these facilities. On the other hand, FMS need proper regulations to manage the booking of facilities. This is due to the huge size of facilities are not easy to take care and to avoid inconvenience caused by any last minute changes of booking.

User able to give feedback to system after they have used it. Any comments and suggestions would be considered to improve the performance of system.

2.4 Sports Facilities Booking System of Chinese University of Hong Kong (CUHK)

This section only analyzes a portion of the system, which contains sports facilities, regulations, booking procedures, opening hours and time schedules. The vital part of the system that enables users to make booking through net is limit to the system administrator. Refer to [www. http://www.peu.cuhk.edu.hk/www_peu/gym_top.htm](http://www.peu.cuhk.edu.hk/www_peu/gym_top.htm) for further information.

2.4.1 Sports facilities

This section is divided into two parts, which are indoor sports facilities and outdoor sports facilities. Indoor sports facilities includes description of gymnasium, weight training room, squash courts and recreation room. Outdoor sports facilities includes description of sports field, stadium, tennis courts and swimming pool.

2.4.2 Regulations

This section is about the regulations governing the use of sports facilities. It consists of four sections, which are general information, regulations governing the use of

different sports facilities and opening of part of sports facilities on Sunday and public holidays on trial. The Physical Education Unit reserves the right to alter or amend any of these regulations as and when necessary, without prior notice.

Eligible users mentioned in this system not only include students, staff and alumni with valid I.D. card but also guests as well as children. Special restrictions have been applied to guests and children.

Priority in the use of sports facilities is also stated clearly. Training and activities organized by different organizations vary in priority given.

This section does not show all the regulations as listed on the site due to its excessive length.

2.4.3 Booking Procedures

Types of booking distinguish booking procedures of various facilities. Group booking, which book sports facilities for competition or training course. For fair allocation, each organization may apply for a maximum of three bookings for the same type of facility per school term. Casual booking means booking of any one type of facility for one hour or less per week.

Booking will be considered cancelled if the facilities have not been taken up within the first 15 minutes of the reserved time and the facilities would be opened for booking thereafter. For cancellation of booking, notification must be made 48 hours in advance of the booked time. Otherwise, the Physical Education Unit would reserve the right to suspend the organization or individual from further bookings. All reserved facilities are not transferable (except tennis courts and squash courts). The Physical Education Unit reserves the right to cancel the booking or close any facilities that are unsuitable for use.

There are various facilities listed on this site, which can be booked by the users such as tennis courts, squash courts and etc. Each of them has included the time for booking, booking procedures and the charges.

Time for booking is valid from Monday to Friday. It may differ at time. Each facility may have its own time opened for booking.

Booking procedures stated out clearly all the procedures and restrictions. It can be an answer to questions of who, what, where, when and how about each facility booking.

Charges for the usage of facility are calculated in each one-hour session. The charges are made to guest (non-student), students and staff. Each person is charged differently following their classes and restricted to book minimum one session. Several of facilities may have different charges and different definition of one session. Besides, facility like tennis court is opened at night. The night court charges more than day court. Booking of vacant courts at the time of use may be made at the service counter of the University Sports Center.

2.4.4 Opening Hours

This section defines the opening hours for general facilities and charged facilities on weekdays, Sunday and public holidays.

2.4.5 Time Schedule

Time schedules of the facilities are just for reference only. For booking of facilities, please follow booking procedures (Refer to "Booking Procedures" at section 2.2.3). Users can choose a desired facility with particular name and view the schedule. All of the schedules are only prepared for the present one-week time.

2.5 Comparison Between FMS and Facilities Booking System (FBS) of Chinese University of Hong Kong (CUHK)

FBSCUHK has many similarities with facility booking of FMS. This instance of facility booking plays an important role as a reference to FMS.

In FBSCUHK, facilities opened for booking are mainly sports facilities such as field, gymnasium, weight training room and the others mentioned above. However, facilities defined in FMS covers not only sports facilities but also other facilities like art rooms, workshops, laboratories and etc. The coverage of booking system in FMS is greater than FBSCUHK.

The Physical Education Unit of FBSCUHK reserves the right to alter or amend any regulations as and when necessary, without prior notice. Regulations govern the use of sports facilities. Besides, rules have stated clearly on precautions, dress, security, safety, behavior and the others. Users must follow the regulations. Many of the rules can be used for booking system of FMS. This is because the full set of regulations are very details and make sense to booking system of FMS. They can be elicited to be one part of the handbook to use FMS. This step taken will save time in the development of FMS. Appropriate changes will be made on these regulations such as the payment charged to users in order to match with the happenings in reality world.

FBSCUHK provides information of general booking procedures. It has touched about the fair allocation of facilities to users, time for booking, charges, procedures and etc. User who wants to make booking has to follow the procedures as well as the regulations (Refer to "Regulations" at section 2.2.2). It also gives service for user to view the time schedule. Although its booking procedures does not fulfill the requirement of booking system of FMS, a big portion of it will still be used in the latter booking system.

FBSCUHK allows non-students or non-staff users rent the facilities. The rental will be charged regarding to their categories.

It is apparent that FBSCUHK is a web-based facilities booking system. The part, which allows booking on web, is limit to system administrator. This fact distinguishes it with FMS, which is not web-based.

2.5 Summary Of Chapter 2

This chapter discussed and analyzed the existing facilities booking system in the market. Comparison has been made between these systems and FMS, the system proposed by this subject. The advantages and disadvantages have been discussed as well. For the elements, which are suitable, will be adopted to the booking system of FMS.

Chapter three will lead you the system development methodology, fact-finding techniques, requirements analysis including those functional and non-functional one; and selection of programming languages.

Chapter 3 Methodology

University of Malaya

In order to understand the current operating environment of an office in school, a series of studies was conducted to obtain information on the existing system, which perform similar with desired facility management system. These studies were conducted with the objective of obtaining understanding of the administrative processes that the school office performs.

3.1 Methodology of System Development

The development strategy used in FMS is the software prototyping methodology due to the fact that uncertainty in several areas in the system requirement may lead to changes and design problems later in the system development. The emphasis is on trying out ideas and providing assumptions about the requirements, which allows the functionality and the system design to be merged.

Among the most popular alternatives in developing the system:

- a) Prototyping
 - b) ETHICS
 - c) Waterfall model
- a) Prototyping was offered as a response to the long development times associated with the systems development life cycle approach and to the uncertainty often surrounding user requirements.
 - b) ETHICS was introduced as a social technical methodology combining social and technical solutions.
 - c) Waterfall model sometimes also calls as System Development Life Cycle (SDLC). The model provides an overview of the phrase and then following with a discussion of specific considerations for the phase, similar to a series of waterfalls (Kenneth E., 1998).

The System Development Life Cycle (SDLC) throughout the whole development of FMS is shown in Flow Chart 3.1.

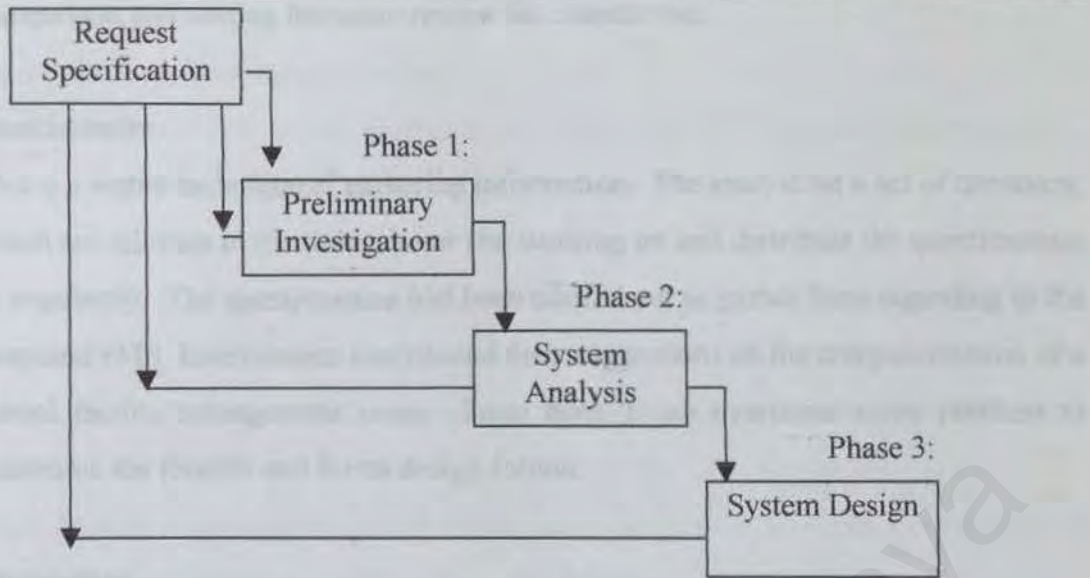


Figure 3.1: SDLC Proposal for FMS

3.2 Fact Finding Techniques

Information regarding the needs, views and suggestions of possible FMS are collected for further analysis. The fact-finding might extract from some old or proposed system, but it is necessary to gain understanding of the current state and future request of the system.

3.2.1 Preliminary and System

Before performing the detail analysis of the requirement for the FMS, facts have been gathered from:

a) Reading

Reading had been done to look for current existing FMS. Reading had been done not only to management books but also to programming reference books. Besides, I have refer to The Malaysian Smart School A Conceptual Blueprint. Articles are one of the reading materials that I found in library.

b) Internet

It is the largest information warehouse I the world. As such, it is used to get information on the same or similar system. Primarily, the Internet played an important role during fact gathering. Literature research needs to be done among the existing facilities management system especially the booking part with FMS. Searching through the Net increased the numbers of choices when making

comparison and writing literature review for chapter two.

c) Questionnaire

This is a viable technique of gathering information. The analyst set a set of questions, which are relevant to the fields, he or she working on and distribute the questionnaire to popularity. The questionnaire had been carried out to gather facts regarding to the proposed FMS. Interviewees contributed their suggestions on the computerization of a school facility management issue. From here, I can overcome some problem to determine the records and forms design format.

d) Observation

In the process of observation, the analyst is the observer of the daily operation of the management. In such way, the analyst gains first hand experience. The six-years' life in secondary school enabled me to observe the common procedures in a school office. Observation gained from past experience in secondary school enabled me to understand more about the facilities booking procedures.

3.2.1 Problems and Solutions

(i) Fault avoidance

The most important strategy in which the design and implementation process should be organized with the objectives of producing fault-free system. Thus, in the FMS development: Structure programming is applied in which the program is written in disciplined manner without using 'goto' statement.

The use of programming construct, pointer, is restricted in the coding since its inherent error-prone.

Data typing facilities in Visual Basic programming language are used to avoid the introduction of software faults, ROD, ADO and class declaration have been used to promote information hiding and encapsulation to make the system more understandable and maintainable in future. Besides that, Visual Basic is a strongly typed programming language, which help to detect any possible errors in compilation.

(ii) Incomplete specification expectation

One of the reason that people disappointed by the result of IT projects is they expect too high at beginning (The Computer Bulletin, May 1998).

a) At the very beginning of the project, specification is set as guideline to be followed in the future. The specification might not be matched totally at starting.

b) Besides, unrealistic user and programmer expectations are avoided (The Computer Bulletin, July 1998).

c) Though, proper requirement and specification must be done to increase the chance of project success (The Computer Bulletin, July 1998).

(iii) Include user involvement

The first step is to find out what results the system users want. Most of the time, programmer might forget about this during a project. The only chance of getting the message across is to write down the use requirement and make sure the use know that they want those requirements to be implemented.

(iv) Requirement engineering

a) Always look back (to the project's goals and requirements) as well as forward (to the next deadline)

b) Keep the specification traceable, which means making sure that each requirement is satisfied by the design, and vice versa: that each element of the design is called for by the requirements.

(v) Interaction between group members

Two members develop the whole FMS, where each of us (Hui Hoon and I) works on our modules respectively.

However, irregular meetings and discussions are carried out to make sure system further development, individual progress as well as the periodic changing goal.

3.3 Requirements Analysis

This section consists of two sub-sections, which are the functional requirement and non-functional requirement.

3.3.1 Functional Requirement:

This section is focus on the topic of how FMS application interacts with its operating environment. The best way to develop and maintain a large program is to construct it from smaller pieces each of which is more manageable than the original program. The technique is called "divide and conquer". FMS consists of four primary modules namely:

a) Tracking and Maintenance System:

- Track the usage and maintenance of school facilities.

b) Inventory System:

- Manage the inventory, fixed assets and other built-in reporting system.

c) User Authority System:

- Allocate a group of user with certain authority for security purposes.

d) Booking System:

- Manage and keep track of the booking transaction.

The functional requirement for integration of all these modules is where all of them should provide services in the integrated environment. The integration covers from front-end interaction with user to back-end database updating and it involved modules from another group member. The combination of each member's work will be done at the final stage of the system development, where certainly will increase the usefulness and completeness of FMS.

This project will merely concentrate on the third and fourth modules.

i) Third Module: User Authorization

For the third module, users within a school can be the support staff, teachers and students. A typical medium-sized school might have thousand students. This certainly will burden the system if all of them are given the authority to make booking. Besides, many

constraints need to be setup for this purpose, this might decrease the system performance.

In this module, users may be divided into groups with different level of authority power to access the system.

For user who attempts to make a booking, cancellation or modification, he or she needs to enter password to do so. This step is necessarily to protect the original 'booker' from being deleting by other immoral users.

Besides, system administrator has to take care of resignation of old staff and cancel their identity as eligible users.

ii) Forth Module: Booking System

This module plays the roles of providing the booking services to authorized users. There are several functionality such as make a booking, modify a booking, cancel a booking and check the schedule for booking information.

Authorization is required to modify or cancel a particular booking.

3.3.2 Non-Functional Requirement:

The system development is constraint by the non-functional requirements.

a) Capability: -

The system supports the Access database environment to manage the underlying data structures. This database provides fast data access, stability and the efficient handling of large volumes of data. The number of facilities, booking work, etc, records is a function of native database and available disk space.

b) Portability: -

Portability is the capability of a system to operate on different platforms. At the moment, the FMS can only support Windows platform.

c) Scalability: -

The ability of the system to migrate to other machines, which is greater or lesser power without many changes to the system itself.

d) User-friendly: -

User may found that it is difficult to operate the system if the design of the interface is not user-friendly enough. At the point of view as the interface designer, I put full effort during the development stage of the user interface in order to make it user-friendly. Besides, opinion from the interviewee has also been used in creating the interface.

e) Flexibility: -

This feature is regarding to the capability to adopt new technology and resources as well as implementation in changing environment.

f) Unity of system: -

The entire FMS appear as a single working environment so that it would not give user bad impression, unlike some system with separate units.

g) Reliability: -

Measurement of a disaster could affect the system. A system considered reliable as long as it does not bring dangerous or unrecoverable damage. FMS is a system that provide back up service to increase its reliability.

h) Robustness: -

A quality of a system in avoids disaster in unexpected circumstances. FMS is developed in such a way where logical of the program has include the probability an input error might occurred.

i) Maintainability: -

FMS is developed by means of modular or top-down approach. A large or complex routine was broke into smaller and manageable modules. This approach was done due to the easy maintainability in the future.

j) Understandability: -

- Coding in FMS program is considered clear and understandable to other programmers.
- The program contains adequate comments, which are written by the programmer.

3.4 Selection of Programming Languages

The main objective here is to identify a suitable programming technology and programming language to build the system. Consideration on the availability of related development tools is also an important factor. Some key issues that need to be addressed for building the system are flexibility, scalability and extensibility. The ideal solution is a system that is easy to build and deploy, easy integration with the newest emerging technologies. The chosen programming technology and language must fulfill the above-mentioned requirements.

3.4.1 Programming Languages

There are few kinds of programming languages available in the market such as C, C++, Lotus Notes, Visual Basic and etc. After the analysis on some programming tools, I have made up my mind to choose the most ideal development tool which is Microsoft® Visual Basic 6.0 for main program and Structured Query Language (SQL) for information retrieval from database. Below is some analysis of some programming languages in previous consideration.

Below is the analysis of some of the programming tools, the features available and the reason why I choose Visual Basic as my front-end tool in the development of FMS.

a) Lotus Notes:

Lotus Notes was invented by IBM co. and acts as a group communication environment that let users access and create their information. It can be defined as a groupware. It uses the Windows interface in networked client/server architecture. It supports graphic user interface and provides some graphic components tool such as button and etc.

It is designed to handle four main types of application. *Tracking* helps user follow what is

happening in their business. *Broadcasting* is used to deliver announcements or post information to a large user group. *References* store and help users shift through information in potentially vary corporate servers. *Team discussion* enables groups to share information and discuss issues. User may be in different locations and working at different times. There are also some specialized application categories.

Disadvantages of Lotus Notes:

Lotus Notes has its own database. The database is not a relational database system (RDBMS). If the user needs to store 100 records into the system, the user needs to save the records 100 times. It is tedious and inefficient.

Lotus Notes and Domino require more memory space for other applications. As a result, the Lotus Notes application frequently hang-up and has to be rebooted.

Lotus Notes has the OLE features to invoke all kind of application installed in the computer (Jeffrey A. Hoffer, 1996). But this OLE is inefficiently implemented because OLE in it can work perfectly by linking and embedding to certain length of Word document such as Abstract filed. When the Word document is in great length, only certain length of the document can be linked and embedded. This is the bug in Lotus Notes.

b) Visual Basic Script:

The Visual Basic script is actually derived from the Visual Basic. Normally it is used in designing web pages to make them more attractive. So it is inappropriate to be used in the FMS development.

c) Visual Basic 6.0:

By the integrated Visual Basic tools, advanced database applications can be developed to access SQL Server database, Access database or any third-party databases by using ODBC, DAO, RDO or ADO and bind the data to forms and reports which greatly reduce development time. Visual Basic provides a variety of data-aware controls for displaying and manipulating data in databases. For this purpose, two controls – the ADO (ActiveX Data Object) Data Control and the DataGrid control can be used. The ADO Data Control is one

way to manage access to a database by a Visual Basic program, and the DataGrid control displays in a table format the data requested from the database.

As indicated in (Deitel and Deitel, 1999), Visual Basic supports the concepts of encapsulation and polymorphism. These concepts are the fundamentals of Object Oriented Programming. Through encapsulation, the information about an object and the manner in which the information is manipulated are all stored within the object definition. While through polymorphism, many objects can have the same methods in which the method can perform differently for each of the objects.

Visual Basic has powerful support for GUI design, which help interface designer to enhance screen design. GUI are built from controls. Controls are reusable, predefined components used for visual programming. Visual Basic controls exist in two variety – intrinsic controls and ActiveX controls. It supports the integration of reporting tool to create and design management reports.

Besides, the use of the Visual Basic procedures instead of writing new procedures reduces program development time. Visual Basic procedures will normally execute faster than “equivalent” programmer-defined procedures that most programmers would write (Deitel and Deitel, 1999). Small procedures are easier than large procedures to test and debug.

Furthermore, Visual Basic is a rapid application development tool, which enables speed development to adapt the fast changing user’s requirements and needs. Support for the integration of powerful report generator tool, Seagates Crsitary Report Professional 5.0 makes Visual Basic an amazing tool for FMS development which enables the implementation of management reporting module. Besides, Visual Basic enables programmers to write codes that use SQL queries to access the information in relational databases.

Besides, Visual Basic 6.0 also provides developers with a interface for faster programming. The Visual Basic IDE provides many features for creating program rapidly. The IDE feature MDI with docking and linking windows to eliminate unnecessary mouse and keyboard

motions and the code editor offers IntelliSense™ technology such as Quick Tips, Data Tips™, list constants, and list members which eliminate the need to memorize syntax and learn object models. Auto list member is one such feature that automatically displays an object's properties and methods. Finally a complete object model to the Visual Basic IDE enables third-party vendors to create sophisticated add-ins to extend Visual Basic with modeling and CASE functionality.

d) C++

C++ is faster than VB because they produce compiled executables while a VB program must still be at least partly interpreted at run time. For a CPU intensive program, those languages can be as much as 5 times faster. The programs are also smaller and do not require run time libraries like a VB program does. On the other hand, if the application is limited by database access speed, file processing, displaying a lot of forms, etc. then a VB program may be as fast as the others and will probably be a lot easier to build. And these days programmer time is relatively expensive so it's often more cost effective to buy a faster computer than to write in a faster language.

e) Java

Java is an Object Oriented Programming language developed by Sun Microsystems, a company best known for its high-end UNIX workstations. Modeled after C++, the Java language was designed to be small, simple and portable across platform and Operating Systems, both at the source and the binary level, which means that Java programs (applications and applets) can run on any machine that has the Java virtual machine installed.

Java is usually mentioned in the context of the World Wide Web, where browsers such as Netscape Navigator and Microsoft Internet Explorer claim to be "Java Enabled".

Java was written as a full-fledged general purpose programming in which can accomplish the same sorts of tasks and solve the same sorts of problems that you can in other programming language, such as C or C++ (Ref. [9]).

f) Structured Query Language (SQL):

The method to access to the database has also been considered as well. Every machine using Microsoft® Visual Basic has access to database library. This interface provides efficient point-to-point access to MS SQL Server for programmers. However, the use of this technology is declining, as less programmer intense solutions are available.

Users access a database via statements in some form of database language. A user may make requests of the database in a specially designed query language that makes it easy to express requests in the context of a particular application. Such languages are referred to as host language.

The most popular style of database system on the kinds of computers that use Visual Basic is the relational database. SQL is almost universally used among relational database systems to make queries or request information that satisfies given criteria.

3.4.2 Database

To build a flexible and robust window application, an appropriate database management system need to be chosen to build the database. Selection will base on consideration for effectiveness in handling the storage required and the case of management. There are few databases, which can be used in the FMS application:

a) Microsoft® SQL Server

Microsoft® SQL Server is a scalable, high performance database management system designed specifically for distributed client/server computing. Microsoft® SQL Server handles multi-user access. SQL Server is an ideal database engine for powering web sites. Through tight integration with Internet Information Server, SQL Server can be queried and updated via popular web browsers. SQL Server's native ODBC lets it inter-operate smoothly with the Internet Database Connector interface included with Internet Information Server (Ref.[8]).

b) Microsoft® Access 97

Some popular relational database software packages include Microsoft Access, FoxPro,

dBase, Paradox, Oracle and SQL Server. In conventional non-database systems, each distinct application maintains its own files, often with considerable redundancy and a variety of physical formats. In database system, redundancy is reduced by integrating separate files.

Sharing is one of the most important benefits of database systems. Existing applications can reference the same data. Database systems not only reduce redundancy and allow data sharing, but also able to avoid inconsistency, enforced standards, maintain integrity and balance conflict.

Microsoft® Access 97 offers improved 32-bits performance, including smaller fonts, more efficient compilation and better data manipulation technology that results in quicker responses and faster data operations. Microsoft® Access 97 has an integrated development with Microsoft Visual Basic for applications and ActiveX. Its intuitive IDE features include drag-and-drop code, color-coded syntax, an improved debug window, and in-place object browsing.

Microsoft Access is a relational database. It is a logical representation of the data that allows the relationships between the data to be considered without concerning oneself with the physical implementation of the data structures. A relational database is composed of tables. Each column of the table represents a different field.

The relational database organization has many advantages over the hierarchical and network schemes (Deitel and Deitel, 1999) as the following:

1. The tabular representation used in the relational scheme is easy for users to comprehend and easy to implement in the physical database system.
2. It is relatively easy to convert virtually any other type of database structure into the relational scheme. Thus, the scheme may be viewed as a universal form of representation.
3. The projection and join operations are easy to implement and make the creation of new tables needed for particular applications easy to do.
4. Searches can be faster than in schemes requiring following a string of pointers.

5. Relational structures are easier to modify than hierarchical or network structures. In environments where flexibility is important, this becomes critical.
6. The clarity and visibility of the database improves with the relational structure. It is much easier to search tabular data than it is to unwind possibly arbitrarily complex interconnection of data elements in a pointer mechanism.

3.4.3 Database Connectivity

a) ActiveX Data Object (ADO)

Microsoft® ActiveX Data Objects enables you to write an application to access and manipulate data in a database server through an OLE database provider. ADO's primary benefits are high speed, ease of use, low memory overhead, and a small disk footprint.

ADO supports key features for building client/server and web-based applications including the following:

- Independently created objects, unlike DAO/ RDO, you no longer have to navigate through a hierarchy to create objects because most ADO objects can be independently created. This allows you to create and track only the objects you read, and also results in fewer ADO objects and thus a smaller working set.
- Batch updating helps improve performance by locally caching changes to data, then writing them all to the server in a single update.
- Support for stored procedures with in/out parameters and return values.
- Support for limits on number of returned rows and other query goals for performance tuning.
- Support for multiple record set objects returned from stored procedures or batch statements.
- Free-threaded objects for efficient web server applications.

b) Open Database Connectivity (ODBC):

ODBC is an open standard API that is fully aligned with XOPEN and ISO standards. It allows applications to access different SQL data sources at run time without recompiling the

application for each target database. ODBC is based on a concept of database drivers that perform conversion between the ODBC API and communication with other drivers, doing so through a standard interface called the SPI. ODBC is a network independent technology because it employs replaceable network libraries.

c) Data Access Object (DAO):

The Data Access Object (DAO) concept was released in 1992 as part of Microsoft® Access desktop database. It focuses on efficient management of desktop data and decision support level access to remote RDBMS data. It's based on Microsoft Jet database engine. Jet is a combination of a full function query processor data store, and functions such as a local cursor engine that provides robust functionality for use with data sources. It includes distributed database's query, update local data management and access to a variety of data including all popular Index Sequential Access Methods (ISAMs) and to all ODBC-based data. DAO's access to remote data involves the use of Jet engine's entire set of extended functionality.

d) Remote Data Object (RDO):

Because of considerations to optimize speed and control, developers ignored JET when creating transaction centric application to a RDBMS. Therefore, RDO was created to overcome this problem.

3.5 System Requirement

The FMS system is a window-based application. It will be setup on a stand-alone machine. This decision is made considered personal computer is the basic equipment which a typical school should has.

The system does not require client-server architecture, neither on-line nor web-based architecture as well.

3.6 Summary

Given the myriad of choices of technology available, it is decided FMS to be built using

Visual Basic 6.0, a development tool in Microsoft Visual Studio. The approach chosen is done due to the fact that it is simple to implement. VB is easier to get started in than other languages. It is relatively simple and straightforward. C++ is much harder. SQL has been selected as the host language in Visual Basic to update and retrieve data from database.

As for database repository, MS Access 97 was chosen due to familiarity and shorter learning curves compare to Microsoft SQL Server. As this project is designed to handle data for small to medium-size school, MS Access is sufficient to accommodate the requirements. Furthermore, database in Access can be scale up to handle future system expansion for growing school need by using MS Access Upsizing Tool to migrate data from MS Access to SQL Server. Due to the need to store images (the layout plan and time table) in database, the database that can be used is restricted because not all database application in the market gives the ability to store images. Therefore, this is one of the reason why Microsoft® Access 97 has been selected

As for long-term position of FMS being developed, ADO has been chosen for database access methods. Microsoft claims this is the way to go in future database access.

The following chapter will focus on system design including the database design, program design, interface design and illustrate the expected result.

4.1 Introduction

System design is the essential nucleus of the software development process and is applied regardless of the development model or standard that is used, the common step involved include analyzing, designing, coding, and testing the system to ensure that it conforms to the software specification and requirements. Each activity transforms information in a manner that ultimately results in validated computer software. This chapter describes in details of how this system will meet the requirement identified during system analysis.

4.2 Design Of Program

System functionality design is based on the system requirements as stated in previous chapter. The design translates the system requirements into system functionality. This functional design phase focuses on the system structured design and data flow diagram.

The system structure is based on the functionality modules. The figure 4.2 – 4.3 show the system structure for Authorization and Booking Modules.

4.2.1 Data Flow Diagram (DFD)

The overall booking module data flow diagram will show the data flow interconnections that are included in User Registration, Authorization, facility booking. The Data Flow Diagram is drawn using the Structure System Analysis and design Method (SSADM) approach. The components of the DFD are explained in the Table 4.1:


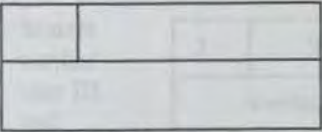


Components	Description
	<p>External entity, can be an external source or recipient, is any object (example: person or other system) which donates and receives information from the system.</p>
	<p>Process, which transform or manipulates data within the system. A process is represented by rectangle, which has three parts: identity, location or person performs the process, and the process name.</p>
	<p>Data source, where the data is held for a time within the system. It consists of two parts: identity, and the simple description of data stored.</p>
	<p>Data flow, is a directed line, which represent the information flow between two objects. The arrow denotes the direction of the data flow. Each of them is labeled with the name or details of the information represented by the data flow.</p>

Table 4.1: Components of Data Flow Diagram

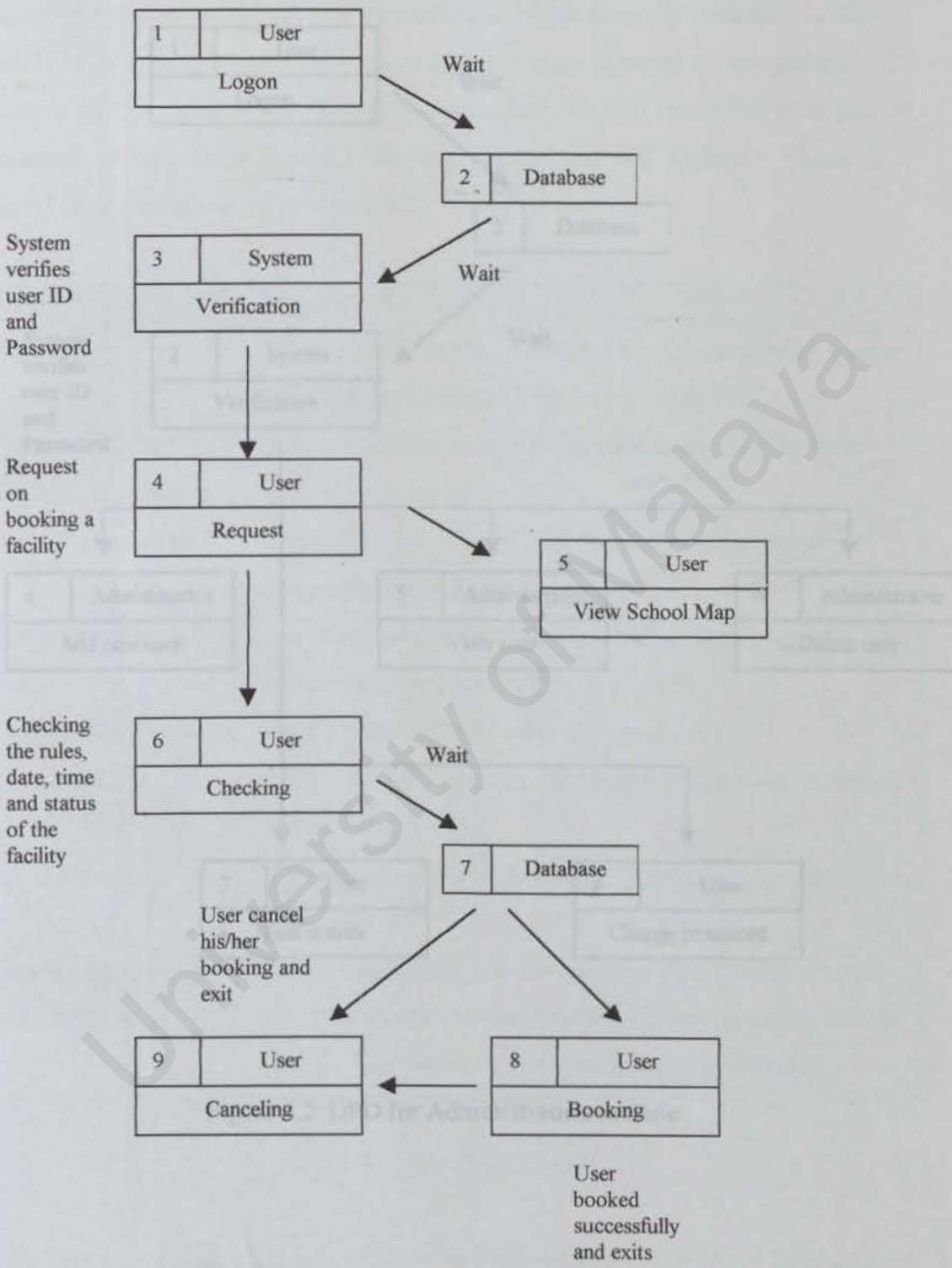


Figure 4.1: DFD for Facility Booking module

4.3 Design Of User Interface

The overall process for design of a user interface begins with the creation of different models of system. The user interface design is based on the graphic user interface (GUI). Some data on graphic user interface (GUI) general principles of designing an interactive system have been considered and applied. These HCI general principles are listed in Figure 4.2.

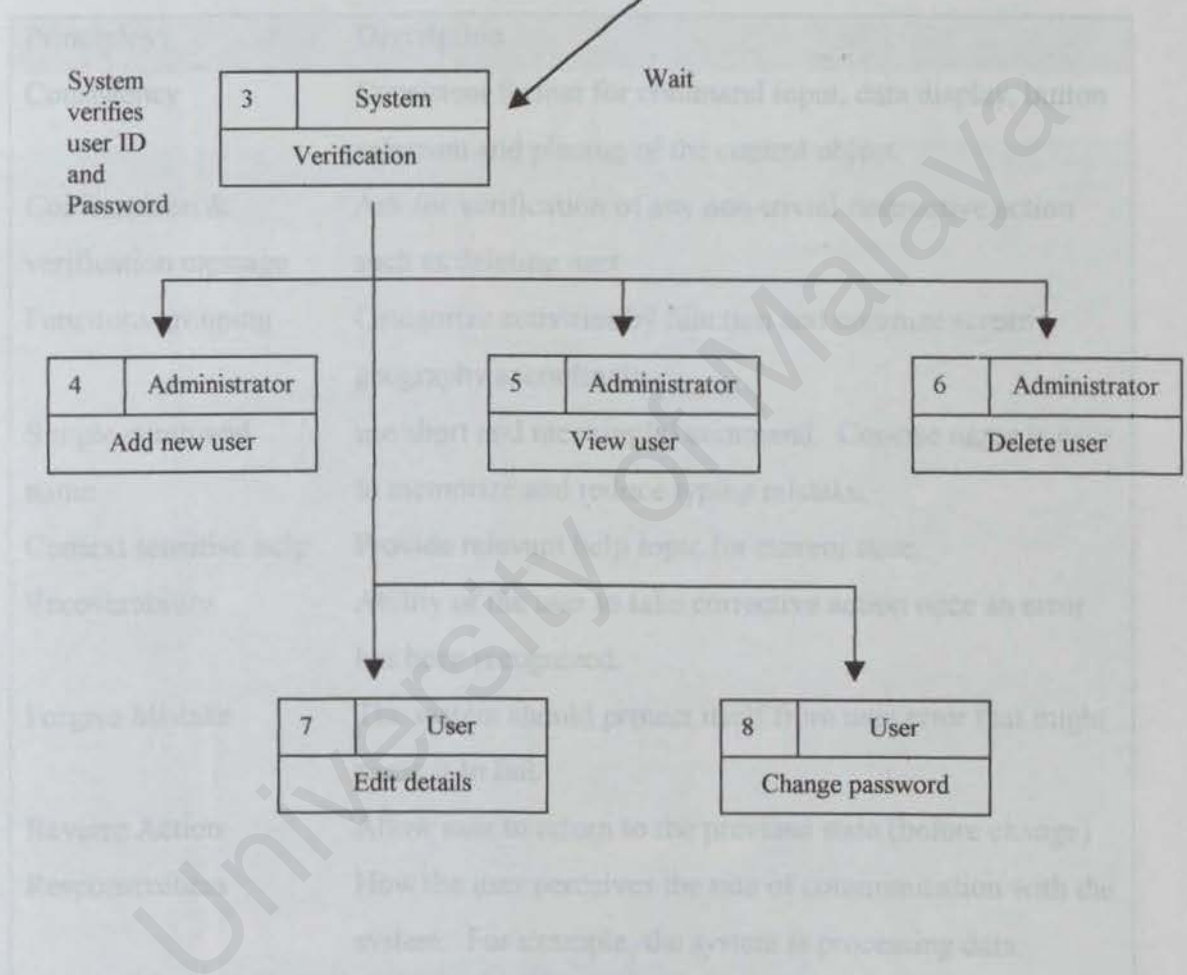


Figure 4.2: DFD for Administration module

Table 4.2: Principles of HCI

At the present stage of development, interfaces are designed as an initial requirement of the development of graphic user interface. The first draft of these interfaces is shown from figure 4.3 to 4.10. Further enhancements might be made to the features on the interfaces from the point of user.

4.3 Design Of User Interface

The overall process for design of a user interface begins with the creation of different models of system function. The user interface design is based on the graphic user interface (GUI). Some of the Human Computer Interface (HCI) general principles of designing an interactive system have been considered and applied. These HCI general principles are listed in Figure 4.2:

Principles	Description
Consistency	Consistent format for command input, data display, button selection and placing of the control object.
Confirmation & verification message	Ask for verification of any non-trivial destructive action such as deleting user
Functions grouping	Categorize activities by function and organize screen geography accordingly.
Simple command name	use short and meaningful command. Concise name is easy to memorize and reduce typing mistake.
Context sensitive help	Provide relevant help topic for current state.
Recoverability	Ability of the user to take corrective action once an error has been recognized.
Forgive Mistake	The system should protect itself from user error that might cause it to fail.
Reverse Action	Allow user to return to the previous state (before change)
Responsiveness	How the user perceives the rate of communication with the system. For example, the system is processing data.

Table 4.2: Principles of HCI

At the present stage of development, interfaces are designed as an initial beginning of the development of graphic user interface. The first draft of these interfaces is shown from figure 4.3 to 4.10. Further enhancement might be made to the features on the interfaces from the point of user.

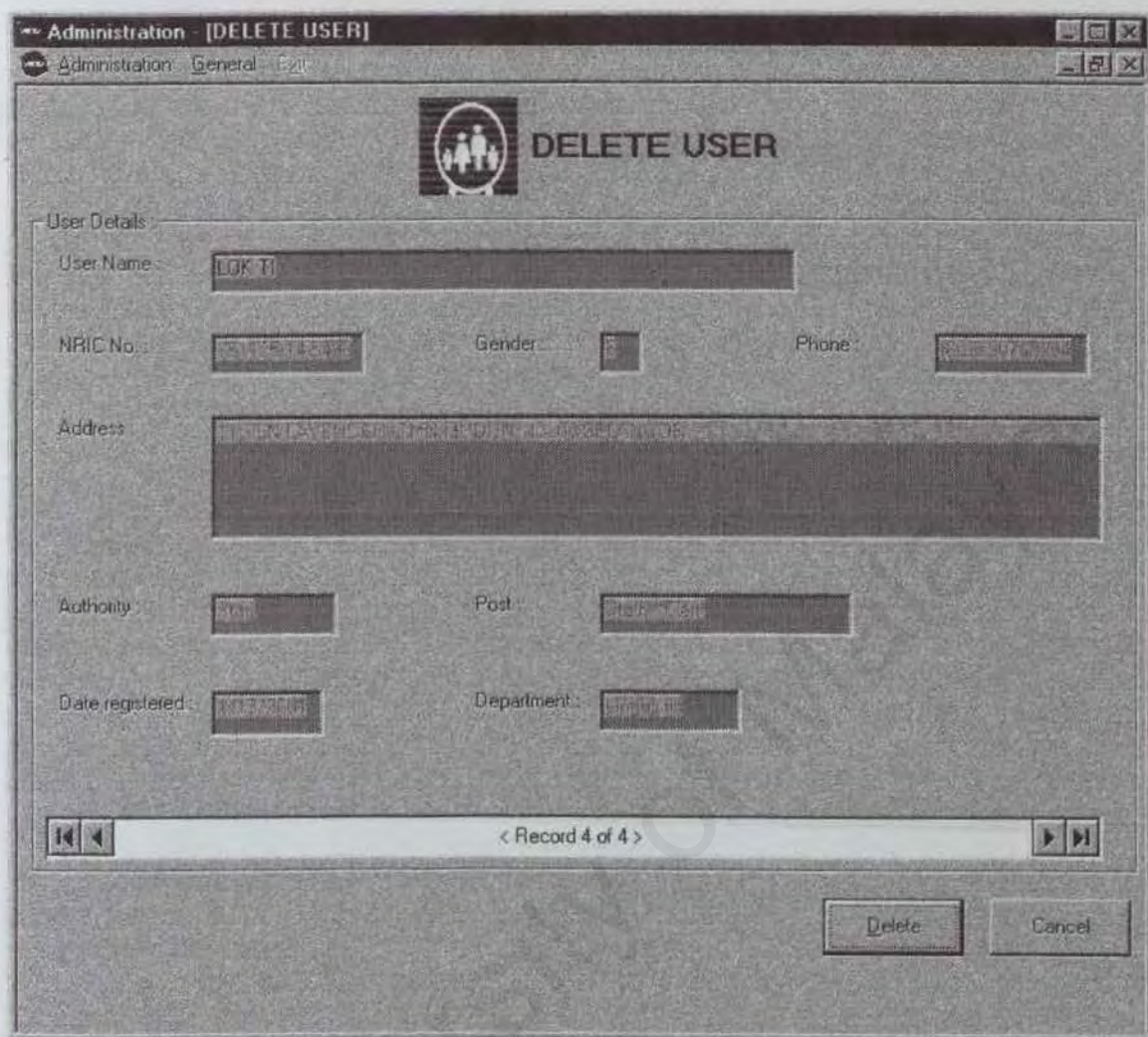


Figure 4.5: Interface for Delete User

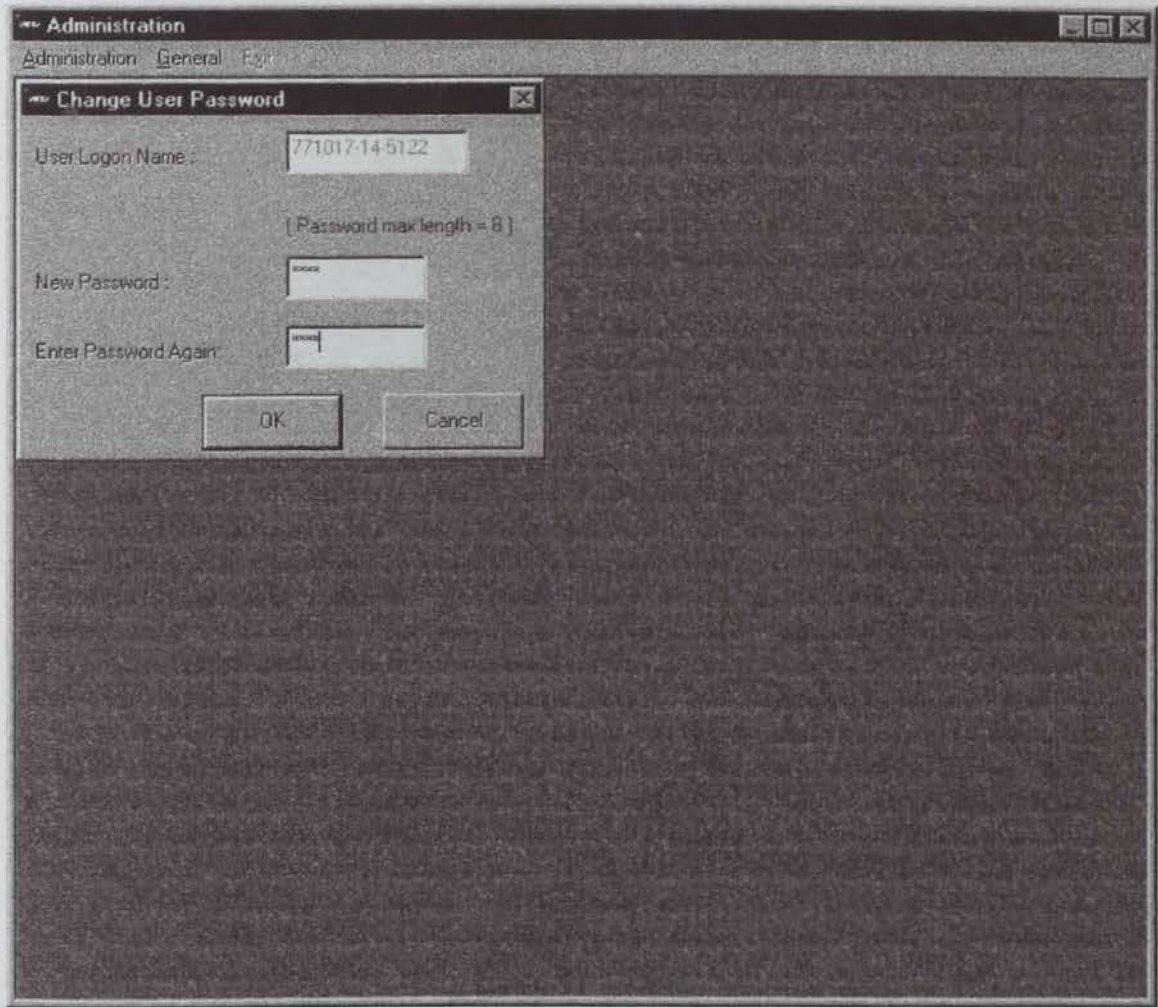


Figure 4.7: Interface for User Change Password

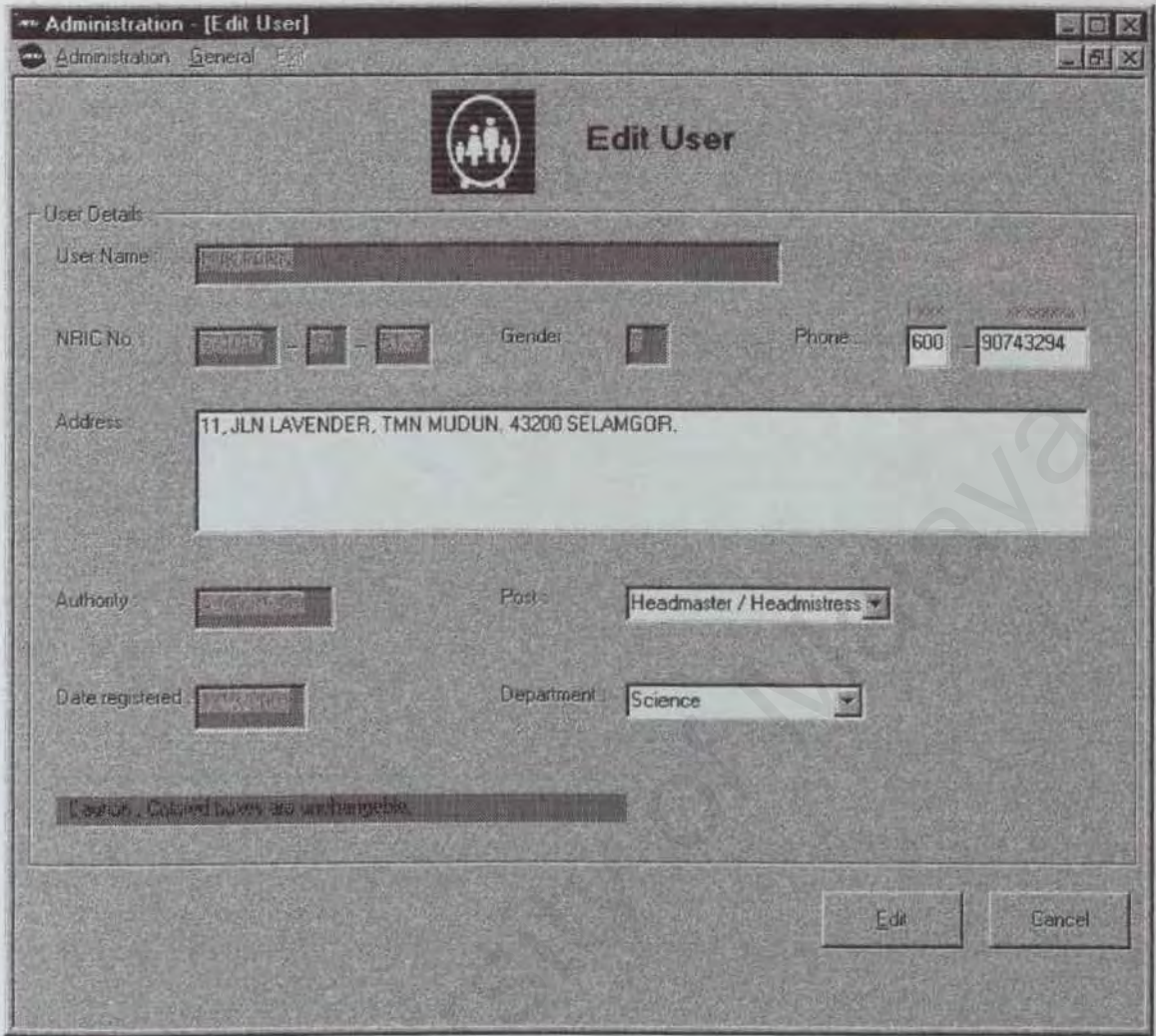


Figure 4.8: Interface for Edit User

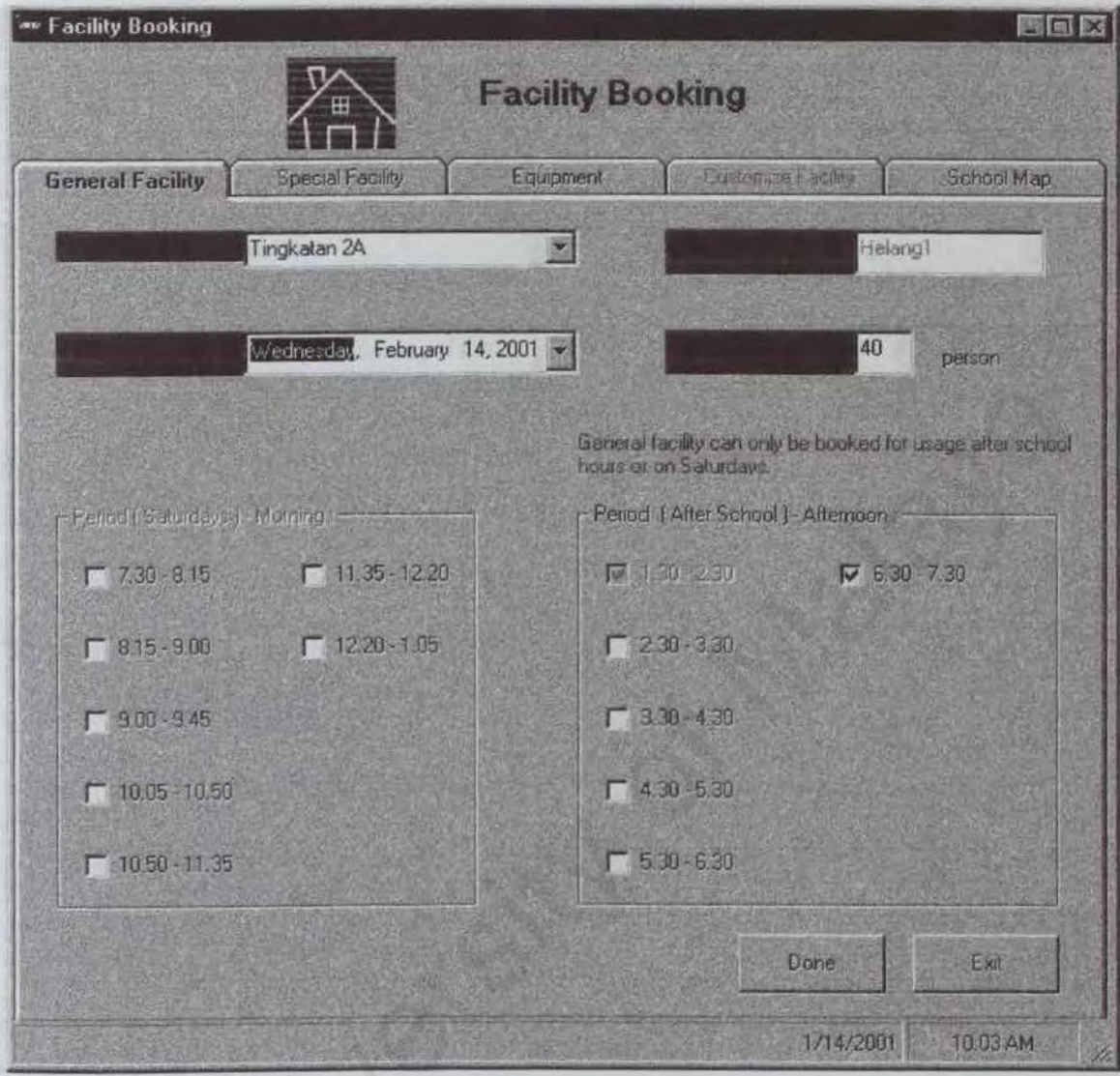


Figure 4.9: Interface for General Facility Booking

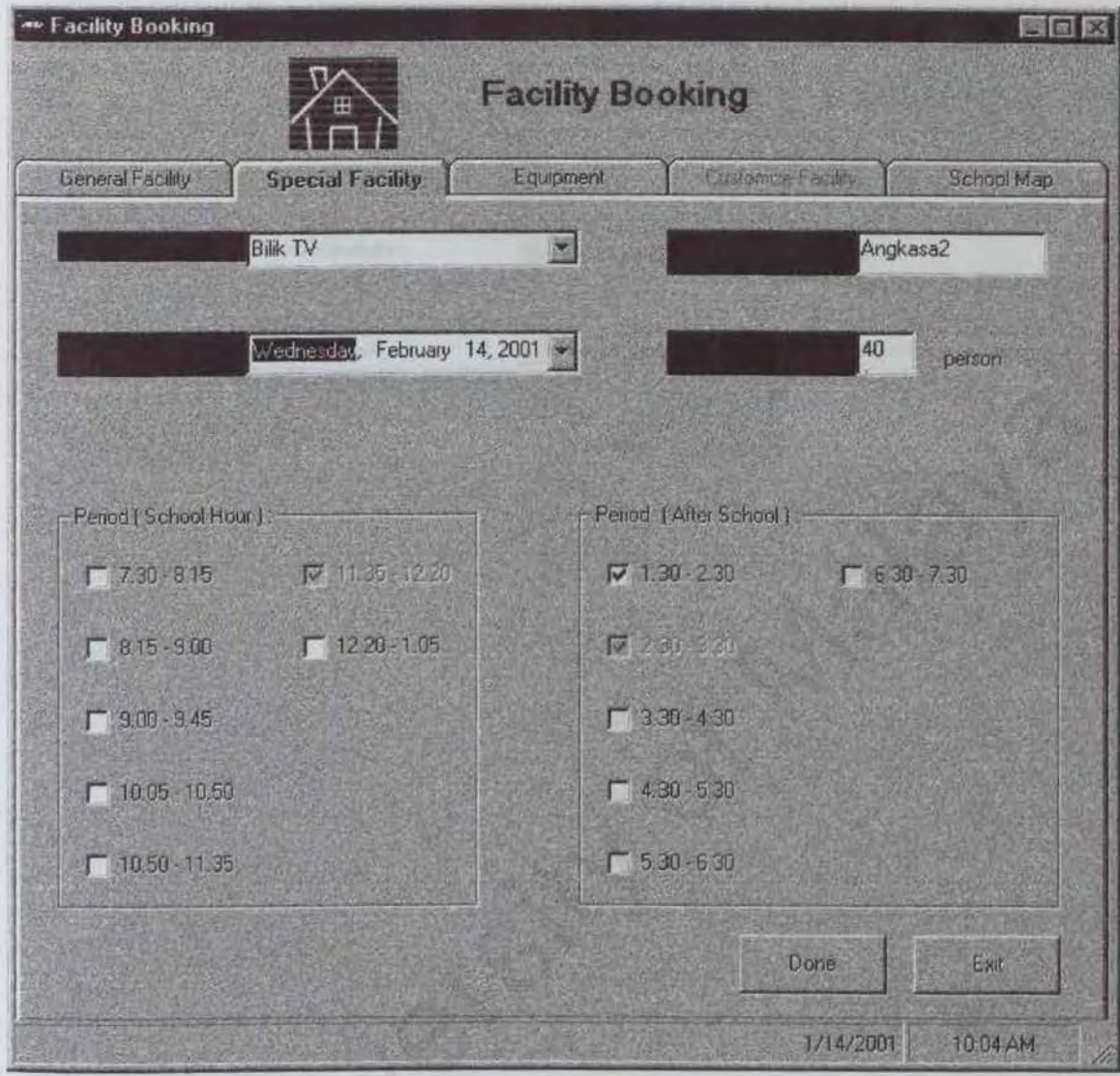


Figure 4.10: Interface for Special Facility Booking

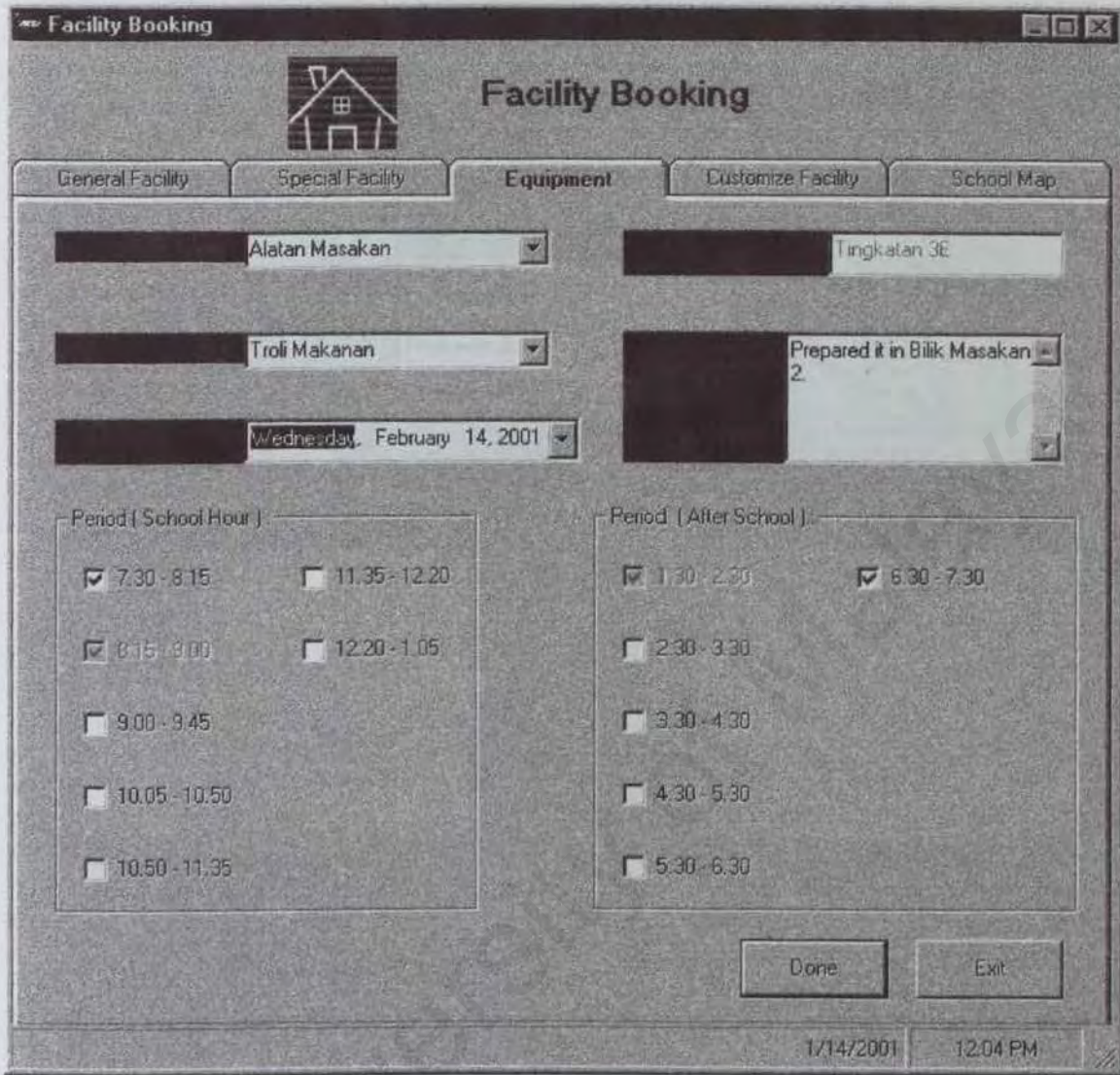


Figure 4.11: Interface for Equipment Booking

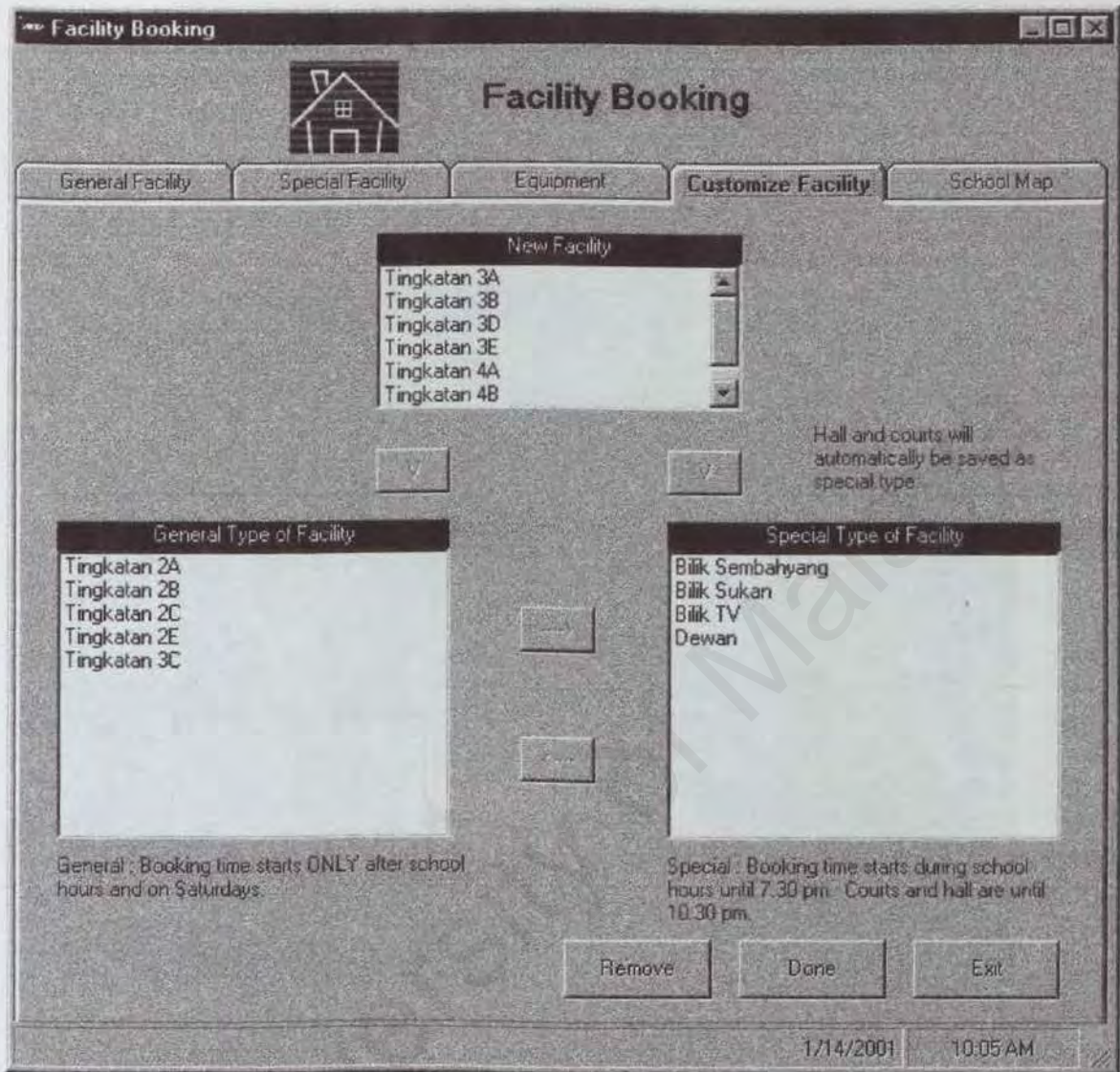


Figure 4.12: Interface for Facility Customizing

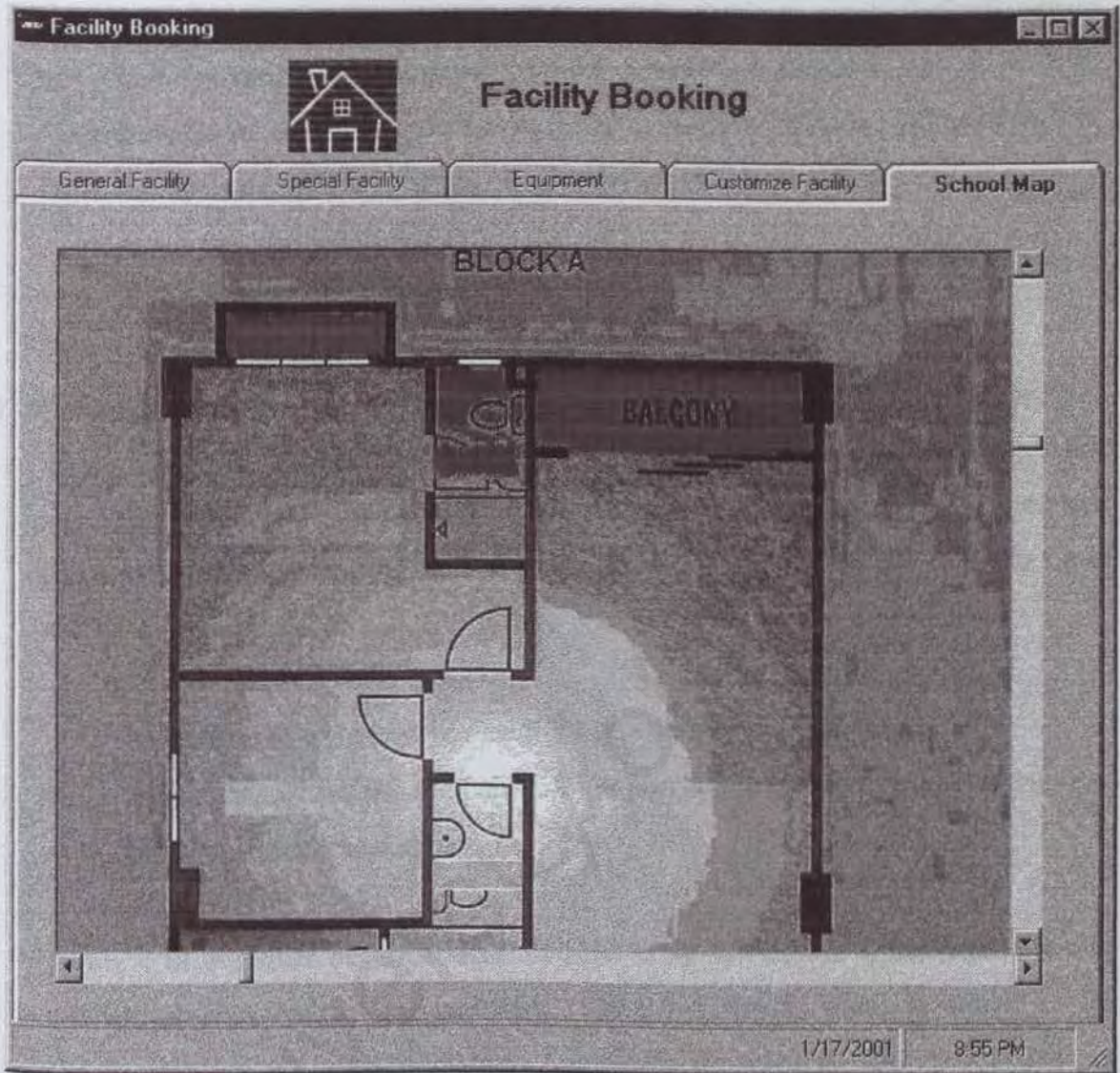


Figure 4.13: Interface for School Map Viewing

4.4 Design Of Database

Database design is one of the three design activities that are conducted during software engineering. It focuses on the design of the database model that will support the system operations and objects. In the process of database design, it must concentrate on the data characteristics required to build the database model (Ref. [9]).

There is only one database in SFMS. Key activities of this stage are:

- create the conceptual design
- designing the table structure for the database

4.4.1 Conceptual Design

In this stage, data modeling is used to create an abstract database structure that represents real world objects in the most realistic way possible. One of the steps taken in this stage is creating the entity relation (ER) model. The ER model is a tool that is commonly used to:

- translate different views of data to fit into a common framework
- define data processing and constraint requirements to help meet the differences
- help implement the database

Figure 4.14 describes the ER model for SFMS. In few tables, which were done by another group member, are included. Only the fields, which involved in the development of Facility Booking and Administration modules, are displayed in the ER diagram.

Figure 4.14 ER Diagram of Facility Booking Module

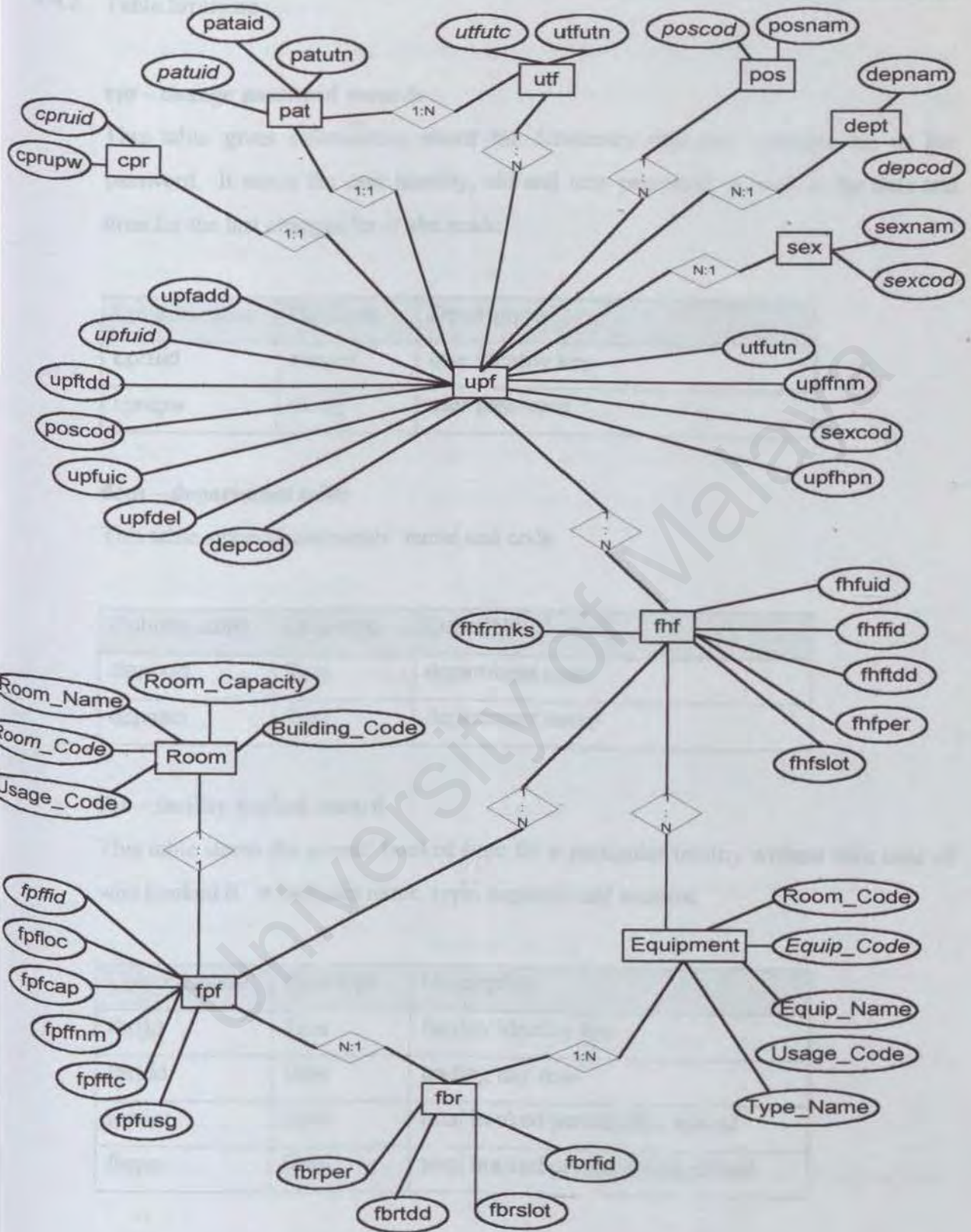


Figure 4.14: ER Diagram of Facility Booking Module

4.4.2 Table Structure

cpr - change password records

This table gives information about the frequency one user changes his or her password. It stores the user identity, old and new password, as well as the date and time for the last changes he or she made.

Column name	Data type	Description
<i>cpruid</i>	integer	user identity key
<i>cprupw</i>	string	user password

dept - department table

This table stores departments' name and code.

Column name	Data type	Description
<i>depcod</i>	Text	department code
<i>depnam</i>	Text	department name

fbr - facility booked record

This table stores the overall booked time for a particular facility without take note of who booked it. It includes name, type, capacity and location.

Column name	Data type	Description
<i>fbrfid</i>	Text	facility identity key
<i>fbrtdd</i>	Date	trading day date
<i>fbrslot</i>	Text	total booked period after school
<i>fbrper</i>	Text	total booked period during school

fhf - facility history file

This table will be used as log file. It stores information about each facility usage including the details of user, type of facility, date and time the facility was used and the administrator whom in charge of the particular facility.

Column name	Data type	Description
fhfuid	Text	user identity key
fhffid	Text	facility identity key
fhftdd	Date	trading day date
fhfslot	Text	total booked period after school
fhfper	Text	total booked period during school
fhfrmks	Text	remarks about the booking

fpf - facility personal file

This table stores the overall booked time for a particular facility booked by a user. It includes facility ID, name, type, capacity and location.

Column name	Data type	Description
fpffid	Text	facility identity key
fpffnm	Text	facility name
fpfcap	Number	facility capacity
fpfloc	Text	facility location
fpfftc	Text	facility type code
fpfusg	Text	usage code corresponding to Usage_Code

ftf - facility type file

This table is more for maintenance purpose. It describes the types of existing facility.

Column name	Data type	Description
ftfftc	Text	facility type code
ftfftn	Text	facility type name

pat - power allocation table

This table stores information about which level of power a user is allocated. It contains the administrator identity that made the allocation as well.

Column name	Data type	Description
<i>patuid</i>	Text	user identity key
<i>pataid</i>	Text	administrator identity key
<i>patutn</i>	Text	user type name

pos - post table

This table stores users' designation code and name.

Column name	Data type	Description
<i>poscod</i>	Text	designation code
<i>posnam</i>	Text	designation name

sex - gender table

This table stores gender type. It is a read-only table unless programmer changes the value by opening the database.

Column name	Data type	Description
<i>sexcod</i>	Text	gender code
<i>sexnam</i>	Text	gender name

upf - user personal file

This table stores user identity, personal details and password. A new row is inserted when a new user register to become a validate user of this system.

Column name	Data type	Description
<i>upfuid</i>	Text	user identity key
<i>upftdd</i>	Date	the date the user register
<i>upffnm</i>	Text	user full name
<i>upfuic</i>	Text	user identity card number
<i>sexcod</i>	Text	user gender
<i>upfhpn</i>	Text	user home phone number
<i>upfadd</i>	Text	user address
<i>upfdel</i>	Boolean	deleted flag
<i>utfutn</i>	Text	user type name
<i>poscod</i>	Text	user position in school
<i>descod</i>	Text	user department

utf - user type file

This table stores user type whether he/she is an administrator, staff or teacher. A new row is inserted when a new user type occurs.

Column name	Data type	Description
<i>utfutc</i>	Text	user type code
<i>utfutn</i>	Text	user type name

****Building**

To keep information about each building.

Column name	Data type	Description
<i>Building_Code</i>	Text	building code
Building_Name	Text	building name
Building_Desc	Text	building description
Schedule_Flag	Boolean	building code

****Equipment**

To store information about equipment such as printer, tables, chairs etc.

Column name	Data type	Description
Equip_Name	Text	equipment name
Room_Code	Text	room code
Date_Purchase	Date	Date of the purchase
Equip_Price	Currency	the price of the equipment
Ven_Code	Text	the vendor's code
Usage_Code	Text	room type
Equip_Desc	Text	special description for the equipment
Deleted_Flag	Boolean	Yes if the record is deleted
Type_Name	Text	Equipment type

****Room Outcome**

To keep information about each room in each building (block).

Column name	Data type	Description
Room_Code	Text	room code
Room_Name	Text	room name
Room_Capacity	Number	room capacity
Room_Desc	Text	room description
Building_Code	Text	building code
Usage_Code	Text	usage code
Schedule_Flag	Boolean	schedule flag

**This table is imported from Ms. Gan Hui Hoon database.

4.5 Expected Outcome

Brief statements are used to state the expected end product of the project. The application, which is going to be produced, will allow eligible users to make use of the basic functions of the facility booking system. A new user can become an eligible user by approaching himself or herself to the administrator.

Therefore, the application must be able to perform these two main functions to achieve the objectives of this project.

4.6 Summary Of Chapter 4

This chapter discussed about the conceptual design and technical design of the system. This includes the functionality of each module and the DFD of each module. DFD is a method to illustrate how data flows in a system. On the other hand, ER diagrams have also been used to describe the objects and relationship between them. Tables are stored in database. They can be created, deleted and updated.

5.1 Introduction

This chapter will describe how the system is implemented. From the beginning of choosing the tools and right techniques until the system is fully tested under certain testing techniques.

5.1.1 Visual Basic Data Access Techniques

Visual Basic is the programming language that has been chosen from a list of other languages. Visual Basic directly supports database files in several formats. The native format is Microsoft Access, using the Jet database engine. However, by setting just one property, VB can access files created with dBASE III, dBASE IV, dBASE 5.0, Excel, FoxPro, Lotus, Paradox, or text files. With the Professional Edition and Open Database Connectivity (ODBC), one can use many other database formats, such as SQL Server, Oracle, and DB2.

Under Visual Basic itself, there are so many concepts and techniques can be used to make standalone system. But which is the most appropriate to be used? Before I state which technique is used, I would like to briefly describe these techniques. Each of these technologies can be a suitable candidate for any given task (Dennis Kennedy, 1997).

5.1.1.1 Data Control

This is the traditional entry point for new database developers and is often used for low-volume (and sometimes even high-volume) applications. Data Control is always being used to create database applications with very little coding (Julia Case Bradley, 1999). A Data Control generally links one form with one table. More than one Data Control can be placed on a single form when someone wishes to reference more than one table. However, the Data Control is really best suited for access to desktop database engines like Jet. If developer is scaling a database from Jet to Access, the Data Control will continue to work, but for optimum performance, try to think about using another data access strategy.

engine; it expects all query processing to take place on the server. This is due to the reason of RDO is really just a thin object layer over the ODBC API, it exposes nearly all the capabilities of the database server with almost no impact on performance.

5.1.1.6 ODBC API

ODBC was designed to be a database independent server-programming interface. Cumbersome and complex to work with, it is the data access strategy for the hard-core programmer. Despite its difficulties, it might offer a minor performance advantage over RDO.

5.1.1.7 VB SQL (Visual Basic Library for SQL Server)

Built on Database Library, the native programming interface to SQL Server, VB SQL is a time-tested and reliable data access strategy. However, although other technologies have continued to advance, VB SQL has not and now has become an outdated technology. Many development shops are continuing to build new applications and update existing applications based on VB SQL despite its "lame duck" status - in part because it is still the only technology that provides access to all the features of SQL Server.

5.1.1.8 ODBC Direct

With the latest release of DAO, Microsoft unbundled the Jet engine from the object model. ODBC Direct used the DAO model but sends queries directly through to the database server using RDO (and via the RDO proxy by ODBC) by passing Jet entirely. This allows developers to use existing DAO code (with minor modifications) but gain the performance benefits of RDO. The limitations of DAO, however still apply, meaning that server database features will be unavailable or difficult to work with.

5.1.1.9 OLE DB / ADO

OLE DB is currently being touted as the new native interface to SQL Server. Flexible and powerful, this technology might meet its objective and become

universal data access technology, but at the moment it is new and unproven. Many development shops are unwilling to take risk of major client/server projects on technologies that are only a few months out of beta.

The cool thing about ADO is that it not only provides us a consistent interface but also gives us high-performance access to just about any source of data. So whether programmer need to create a front end to a local database, or a middle tier that contains business objects, or even get data from an Internet browser, ADO is the single data interface programmer need to use for solution.

The latest technology that performs this magic is OLE/DB. OLE/DB is designed to provide universal access to several relational and non-relational data sources. Programmer will communicate with OLE DB using Active Data Object (ADO). By using ADO in conjunction with OLE DB, one can talk to Access, Oracle, SQL Server, or any other data source by simply using the ADO object model.

In the future, DAO and RDO will eventually be replaced with ADO as indicated by Microsoft. OLE DB will now handle working with the standard relational data and non-relational data from just about anywhere on the planet

5.1.1.10 Summary

Data Control and ADO are chosen because it is the best option. Data Control is quite easy and powerful. ADO is a proven technology. To optimize the usability of connection object method in this system or forthcoming upgrade, ODBC driver will not be chosen because it does not support transactions involved some connection methods like "CommitTrans", "RollBack" and "BeginTrans".

5.2 Coding

Coding step a process that transforms design into a programming language. In developed FMS, it is using SQL and Visual Basic.

5.2.1 Code documents

Descriptive comments are embedded within the body of the source code and are used to describe processing functions. In FMS, the following methods are adopted:

- Describe blocks of code, rather than commenting every line unless the line of code is important enough to be described.
- Use blank lines or indentation so that comments can be readily distinguished from code.

5.2.2 Data integrity

Data integrity in FMS is enforced using the following techniques:

- Validate event occurring before inputs are added or updated to underlying table. In doing so, system will ensure that only the valid format, types of inputs are stored in the database. Besides that, system will also enforce some of the fields as mandatory fields. It means user will have to key in information into these fields before they can save the record.
- Once the user wants to delete the master – detail relationships records, FMS will make sure that all the details or related records react corresponding to the deletion.

5.2.3 Modularity

The various system components are functionally independent from each other. New modules can be added, or current modules modified, without affecting other modules in the system.

5.2.4 Chronology

For the purpose of future enhancement by author or other programmers, date, programmer name and description of enhancement should be filled at the beginning of that piece of program.

5.3 User Interface

Notice that, the user interface for all the following system is almost the same. This is to ensure a look and feel to user. By having standard interface will not only reduce user learning time, but also promote better interaction between the user and the system. Therefore, it will explicitly build user confidence in using the system. Besides, a standard environment will build a trail for later interface development of other systems.

Monitor with resolution 800 * 600 with VGA support is expected in the standalone machine to support the application. The list below shows some of the graphics tools that have been used to create the images, background and even the icon.

- IconForge™ 4.6f(Favicon Edition)
- Microsoft ® Paint

5.4 Generating FMS

5.4.1 User

In this system, user works on facility booking and administration. Basically the users are the administrator, staff or teacher. They have the right to make booking on desired facility. This system can be used during new teacher and staff intake.

Each user must logon to this system before using the features in it. There are three types of user in this system. Therefore, let the following sections describe the roles of these users:

5.4.2 Administrator

Administrator of the FMS will have the super power to operate all the functions like adding, deleting and other operations.

The purpose of adding administrator user is to help this system goes smoothly. Administrator basically provides the sources to the user. For example, administrator can group new facility, new user into the database. So later, the

users will be able to do some self-managed functionality like booking, viewing and etc.

5.4.3

Staff / Teacher

They are the primary focus on this system. Adding and editing booking are the main functions done by users in this category in a school. This will eventually build some interaction between the system and the users.

Besides, they are able to change their own password and personal details.

5.4.4

System Rules

The basic requirement for this system to work is the person who requests for booking must be a valid user. Therefore, the user record must be stored in the database. If no record found in the database, system will not go on and error message will be prompted to the screen.

To make this system more realistic, it consists of the following system rules:

5.4.5

Password

Once the system is loaded with the user information, the initial login name for accessing the system will be the user IC number itself. For example, if the user IC is 123456-14-3333, then the initial login name will be 123456143333. To ensure higher level of security, the password will be stored as case sensitive. Later, users can change their password in Change Password module.

Like any other system where user can change password. It requires user to reenter the password to confirm the changes of password. In this system, the length of the password is set to maximum of 8 alphanumeric. The password is case sensitive to ensure higher security.

- 5.4.6 **Attempt of login**
Another security issue is the number of attempt to login into a system. The maximum number of login is unlimited. Every failure of wrong password submitted to the logon page will not deduct the chances of the user from logging into the system.
- 5.4.7 **Maximum booking for user**
In this system, there is no limit for user to make booking on the same facility as long as the facility is available at that period.
- 5.4.8 **Adding new user**
When a user is adding data into database, certain field must be entered before updating can proceed. For example, adding user identity key field is compulsory. If the field is not added, error message will be prompted to user. Updating only occur when this field is filled. Otherwise, it will not make any sense to add a new address without knowing the user identity key. But the primary key of the user, user identity key is also generated once a new user is added according to the user IC number.
User is not permitted to change certain fields. For example, user identity key, user name and user IC number. Once user decided to save and update the changes, the record will be updated to database.
- 5.4.9 **Deleting user**
Administrator can browse users' records before he or she decides which record need to be deleted. This can avoid unnecessary mistake in deletion.
- 5.4.10 **Viewing user**
Administrator can browse users' records for tracking and checking purposes.
- 5.4.11 **Change password**
Users can change their own password at this module.

5.4.12 Editing user

There are some restriction rules have been setup to ensure security integrity. User is blocked from editing his or her user name, IC number as well as level of power.

5.4.13 Editing / adding booking

When an user is editing / adding booking, the prior condition is where the facility he or she want to book must be a valid type of facility in the school and this facility type can be booked. For example, booking a classroom, which is not grouped into either General or Special, is not a valid action because the classroom is only for maintenance purpose but not for booking purpose.

Firstly, the system will check whether the facility is available or charged. If the book is available, then it will check the whether other users had booked it at the same time.

User could not modify a facility booking on behalf of another person. This is due to the security issue. Users are advised not to give their password to anyone.

5.4.14 Customizing new / old facility

Administrator is allotted power to customize whether a facility should be grouped into General type or Special type. Different facility type gives different periods available for booking.

5.4.15 Security in editing / adding mode

This is a very important aspect. For example, an administrator is in editing / adding mode. If the administrator accidentally clicking the exit button, the system will cancel the transaction and changes in the form will not be saved.

5.4.16 Cancel changes in editing / adding mode

This option will cancel any changes made during in adding / editing mode. This provides the undo feature for the user when making mistake.

6.1 Introduction

Testing is performed to ensure that the programs are executing correctly and conforms to the requirements specified. It provides a method to uncover logic error and for testing system reliability. The strategies need for testing unit testing, integration testing and system testing.

6.2 Approaches to testing

6.2.1 Unit testing

Unit testing concentrates on the smallest unit of software design - the code. It basically compares the code with the specifications and with programmer's design to make sure that programmer have considered all relevant cases. The unit test is white box oriented, and the steps are conducted in parallel for multiple modules. White box testing is a test case design method that uses the control structure of the procedural design to drive test case. By using the detail design description as a guide, one tried to identify all the important control paths are tested, in order to uncover errors within the boundary of the module. There are three ways to do module testing (Thari Lawrence Pfleeger, 1998).

- i. Examining the codes
- ii. Code Walk Through
- iii. Code Inspection

After each program unit has been tested, the interaction of these components when they are put together must be tested again. Therefore, module testing is the following phase after the unit testing.

6.2.2 Module testing

In the development of FMS, module testing is done concurrently with the design or implementation of sub-modules and each sub-module contains many procedures and units. For instance, "Delete user" is one of the sub-modules defined and within this module, it can be further divided into many sub-functions. Each of these sub-functions, such as user confirmation, error message and clear operation are tested,

followed by its sub-modules. These sub-functions and sub-procedures are tested to ensure that it is error free.

In FMS, an incremental integration strategy approach is used. FMS main system is constructed and tested in small segments, where errors are easier to isolate and correct, interfaces are more likely to be tested completely.

6.2.3 Integration testing

When the module testing passed, it is the time to test integration between two or more modules. For example, in FMS it happened when a facility-booking module is integrated with the facility maintenance module, which is done by Ms. Gan Hui Hoon. Therefore, by having this type of testing, ensure that the integrated system will not fail and performs as it is expected.

Integration testing is used on FMS for constructing its program structure while at the same time constructing test to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design. This testing will ensure that the interfaces such as the module calling sequence in FMS are systematized.

Throughout the testing, the bottoms up integration approaches are used. Bottom up integration is one of the approaches for merging components to test the larger system. When this method is used, each component at the lower level of the system hierarchy is tested individually first. Then, the next components to be tested are those call the previously tested ones, these steps are repeated until all components are included in the testing.

6.2.4 System testing

This is the last testing procedure done. Once the entire system is validated, it must be combined with other system element such as hardware, end-user and database. System testing is a series of different tests which primary purpose is to full exercise the computer-based system. FMS is only one element of a larger computer-based

system and upon completion of the development process, it is incorporated into this larger system. System testing verifies that the software is properly integrated with all other system elements and that it performs its allocated functions. System testing verifies that elements are functioning properly and the overall system performance and objects are archived which in this case, FMS must achieve reliability, accuracy, flexibility and modularity.

6.2.5 Security testing

It is important to protect all the data, especially the personal documents. Security testing attempts to verify that protection mechanisms built into the system, which protect it from improper penetration.

6.3 Testing and debugging

6.3.1 Test data

Test data was used in the execution of the program. In FMS, a series of tests was conducted with data that are individually designed to represent the real environment as closely as possible. The following categories of test data were used to execute the program:

- Normal test data

Testing with normal test data is a procedure whereby the program goes through a light and simple test to determine whether the program run or not and to determine whether it is error-free.

- Missing test data

Testing with incomplete data to see whether the program display suitable error message to stop the user to proceed on.

- Erroneous test data

Testing with erroneous test data is a procedure whereby the program goes through an erroneous test. Erroneous test is a test where errors are keyed in intentionally. This test is vital to determine how the program or system can handle such errors

or incorrect data and from there, the reliability and the efficiency of the system can be predicted.

6.3.2 Test procedures

Test procedures were always tested in the execution of the program. In FMS, various combinations of procedures were conducted to test the interactions between function.

Each person has his or her thoughts about how things will be done. There will be big possibility that they add a booking according to their own procedures.

Combo boxes, list boxes and buttons were clicked in different order to ensure program proceeded smoothly.

6.3.3 Testing techniques

Two techniques were employed in the testing process: - white box testing and black box testing.

6.3.3.1 White box testing

White box testing is a testing case design method that uses the control structure of the procedural design to derive test cases. Using white box testing, test cases were derived with the purpose of (Roger S. Pressman, 2001):

- Executing at least once all independent paths within a module.
- Exercising all logical decisions on their true and false sides.
- Executing loops at their boundaries and within their operational bounds.
- Exercising internal data structures to ensure their validity.

6.3.3.2 Black box testing

Black box testing is used to demonstrate that software functions are operational, that input is properly accepted, and the output is correctly produced. It complements white box testing and is likely to uncover a different class of errors. Black box testing attempts to find errors in FMS for the following categories (Roger S. Pressman, 2001):

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Performance errors
- Initialization and termination errors

6.4 Summary

From all the sections above, the overall I have been describing, notice that certain features could not be included due to the time constraint. To understand the limitation and strength on these systems, please refer to the next chapter. Chapter six will concludes the project by elaborating on the overall achievement of the project.

7.1.1 Difficulty in choosing appropriate development tools

There are many ways and tools available to develop a standalone management system currently as stated in the project charter. Choosing a suitable language and tool proves to be an intricate process as all tools has its strength and weakness. Besides, availability of the required tool for development is also a major consideration.

To determine which approach to use, seeking advice and views from project supervisor and other project member. Finally, certain tools have been chosen due to the availability of current knowledge and division time schedule to pick up the skills.

7.1.2 Scope of system

As the name FMS applied, it is possible to build a full-scale and complete system within the given time frame. Besides that, inexperience with the present FMS technology and school information was another hindrance to implement a fine

7.1 Introduction

This chapter consists of few sections that will emphasize on the project problems and its conclusion, advantages and strength of this system, disadvantages and limitation of this system and finally suggestion and opinions on future enhancement in this system to make it a better and complete system.

7.2 Project Problems and Solutions

7.2.1 Data gathering and analysis stage

The basic knowledge needed as a foundation in building an application of this nature involves studies in fields such as the facility management system, user management system and certain security procedures. A lot of analysis has been carried out on technological and programming concepts to grasp the concept of new rising technology. The following are some of the major problems encountered during the project system studies, analysis and development.

7.2.1.1 Difficulty in choosing appropriate development tools

There are many ways and tools available to develop a standalone management system currently as stated in the earlier chapters. Choosing a suitable technology and tool proves to be an important process as all tools has its strength and weakness. Besides, availability of the required tool for development is also a major consideration.

To determine which approach to use, seeking advises and views from project supervisor and my other project member. Finally, certain tools have been chosen due to the availability of current knowledge and allowable time schedule to pick up the skills.

7.2.1.2 Scope of system

As the name FMS implied, it is possible to build a full-scale and complete system within the given time frame. Besides that, inexperience with the present FMS technology and school information was another hindrance to implement a true

workable procedures as being implemented now in the school that consists of library, gymnasium and etc.

During the initial stage of the project, discussions were held with project supervisor and the other partner to outline the scope of the project to be built. In addition, questionnaires were distributed to few teachers with the assistance of the project supervisor. The present system flow and suggested system flow have been mentioned in Chapter three.

7.2.1.3 Time slot in facility booking

Before a certain facility is ready for booking by users, it has to be assured that it is available at that period of time. But what happened is facility such as classroom; workshop, laboratory; art room, sewing room and etc are always used for teaching/learning purpose on weekdays. So, they are heavily used daily just on this purpose. There is no way that they can be booked for other usage during school hours.

- i) Therefore, few assumptions have been made to reflect the happenings in real situation:
- ii) For daily heavy usage facility, they are only opened for booking after school hours and on Saturday.
- iii) For other facility such as library, computer rooms, AVA room, hall and courts are available during and outside school hours. Besides, booking hour for hall and courts even can be extended until 11 p.m.

7.2.2 Implementation and testing stage

The problem faced during the initial project studies and analysis were not as critical compared to the problems faced during the implementation and testing period. Below are the problems encountered during these phases.

7.2.2.1 Development tools

Since there was no prior knowledge in Visual Basic 6.0 and Microsoft® Access 97, there was uncertainty in programming on how to connect a Visual Basic application to a database.

Using the Visual Basic as the programming language tool, I admitted that I have made a lot of logical as well as syntax errors. But luckily, with a good Integrated Development Environment (IDE) provided in Visual Basic, debugging was easy and it was the best way to solve many logical and syntax errors.

Though there were obstacles encountered in the early stage, choosing the right tools proves to be a wise move. The learning curve was short and most of the problems faced were manageable.

Beside, looking for the right software to design the system logo is another problem. Fortunately, a free software trial version (IconForge) was found and was able to be used to produce different effect on icon.

7.2.2.2 Resources sharing

During the development of this group project, resources sharing play an important role for coordination. For illustration, icon editor software is needed to design the system icon to meet the system needs. The output must have approval from all the members. So every member needs to possess the software to make amendment. With tolerance and understanding between the members, it was handled properly in a proper manner to avoid any inconvenience among project members.

7.2.2.3 Database integrating

At earlier stage, each of the group members has design the database according to personal needs respectively. The tables' fields have to match both members' usage especially those tables involved in more than two modules. The project members were extraordinary careful while dealing with database.

7.2.2.4 Debugging message

Unclear debugging messages did not help much in fixing the bugs. Example: "Type mismatch", "Invalid use of property" and etc. Sometimes when a problem occurred, incorrect statements were written to see the debugging message. If the message was clear enough, fixing bugs in the programs will not be a tedious way forever.

7.2.2.5 Log files

Log files inform the programmer that logical errors exist in the program. Whenever a project was opened and one of the forms was selected, log files occurred during this moment if there are any logical errors in that form.

Problems with log files even harder to be traced compared to the message generated during debugging. Log files provide ambiguous information and incorrect line of code that might be the source of problems.

7.3 System Strength

The following points illustrate the strategies in the FMS system:

7.3.1 Simple and user friendly interface

The application makes full use of Window, icon, menu and pointer (WIMP) techniques, allowing the use of visual objects to navigate through the system. The use of Multiple Document Interface (MDI) will eventually reduce the needs of interface navigation. An example is the administration system, which implements this type of interface. With this interface user will not need to go back to menu again after finish one task to perform another task again. The learning curve is foreseen to be short and a user should be able to use the system with ease within minutes. Besides that, each of the systems has almost a standard interface to help user quickly familiar to another system. For example, user interfaces for user adding and viewing are almost similar.

7.3.2 User ID, password and security

FMS is a password-protected application. Each user ID is stored in the table in database. Only those who have the correct password match with his or her user ID will be able to use the system. Not even the administrator has the power to view other users' password.

Simple encryption technique has been used in the program, which just reverse the password string entered by user and stored it in the database. This provides the most basic security where even super user also cannot view other user's password.

7.3.3 Reliable system with effective error recovery

FMS is a reliable system as it caters for almost all-possible error encountered. Input by users is validate and verify through using password matching. Appropriate error feedback will be generated to user to remind or warn the user. For example, an update failure to a record will be handled by system and a user-friendly message is generated informing the user about the error and at the same time the system would recover from the error and continues to be used.

7.3.4 Query based mode for record retrieval

The user is not given option to retrieve records based on some common criteria according to suitability. For example, in the facility booking system where user can choose the wanted facility that listed in the component. This method prevents user from entering various definition of the facility.

7.3.5 System transparency

System transparency refers to the condition where the users do not need to know where the database resides, how is the system structure, its database management system and anything related to the system built. For example, the information retrieval of records is similar to a system accessing the local database. This is to ensure not to confuse users especially students in retrieving information.

7.4 System Limitations

The following list the limitations of FMS:

7.4.1 Operating system limitations

FMS requires Window 98 or above as the operating system to perform its execution. FMS is not a portable operating application (multi-platform application). It does not support other operating system, such Macintosh system, UNIX, Linux or Java Operating System.

7.4.2 Minimal system rules

Due to time constraints, FMS does not reflect a full set of system rules for all the scenarios. At this stage only sufficient rules and procedures are implemented in the system.

7.5 Future Enhancement

A system development knows no boundaries as new requirements and better implementation methods continue to arise and evolve. There are several enhancements that could extend and expand the usability of the developed system.

7.5.1 Encryption of password

Like any issue around the computer security technology, encryption is important and is necessary to protect data. At the moment, high level and complicated encryption is not implemented due to the time limitation of the project. Therefore, this module is likely to become the next enhancement of FMS if it is continue to evolve.

7.5.2 Cater to multi-platform operating system

As stated, FMS required Window 98 or above operating system for execution. In future, FMS can be tuned to fulfill other operating system such as Linux for execution. Unlike Windows operating system, Linux does not charge a penny for its operating system license to the end user compare to Microsoft ® Windows. Eventually this will reduce the cost of buying new license for Windows. Besides that,

Linux now is becoming a rising star in the end user market. More and more user has tuned into Linux.

However, tuning of operating system from Windows to the others will only be executed based on the government policy. This definitely related to the habit and skills of the publics in using which kind of operating system.

7.5.3 More administration task

Administration task can be further enhanced to include more features to ease maintenance process. Among the features may be include are report generating, booking by means of map clicking, data mining and database backup.

7.5.4 Integrating mailing capabilities

The complete FMS system is not complete without a mail service. In future, a mail function can be added into FMS and administrated through the administration section. With mailing capability available company or workshop are easily contact by the administrator. For example, the administrator will notify the company or workshop by email if the school has any tools, books, facility, teaching materials and etc to buy and repair.

7.5.5 Enable web-based functionality

There are lot functions can be tuned into web-based system. For example, online booking through browser. Even though students are not allowed to have account to use the system, some of them might be chosen and given the account in future. For example, the monitors of each class might have the privilege to use the system compare to the others.

7.5.6 Enable map-based booking

Booking will be an easier task if 'clicking' on a dot of a map can do it. This required certain technique or may be involved some software tools other than what was used at the present.

7.6 Knowledge and experience gained

Throughout the whole process of analysis, design, implementation, testing and evaluation, technical techniques have been learned to build a complete and accurate system.

Other software tools other than the main software (Visual Basic 6.0) such as IconForge and Microsoft Help Workshop has been used to support some features in the system. Use of these supplementary tools makes the system more complete and fulfill the system requirements.

The spirit of group is important and thus has been carried through the whole process in order to complete the full system in time.

7.7 Overall conclusion

Overall, FMS has achieved and fulfilled the objectives and requirements as a FMS as determined during system analysis. FMS is not only a facility management system useful for the teachers and staffs but also includes an administration module for administrator.

There was a lot of knowledge gained throughout the development of FMS. These include knowledge gain in standalone system with integration with database technologies, programming and concepts as well as database. Programming in Visual Basic proves to be a valuable experience.

Visual Basic technology with integration to Microsoft® Access as the back end turned out to be the most suitable technologies to develop such a system. But while programming skills and technologies are important in developing the system, it must not be discounted that in any system development, good software engineering techniques must also be applied. Here theory and knowledge gained throughout the course of computer science studies like system analysis; design and software engineering were literally put into practice.

Therefore, there are still many rooms for improvement in FMS, in terms of implementing a comprehensive FMS and gearing towards online system maintenance. With the first step taken, enhancement could still made with more features added for future versions.

Finally, with project of this nature affected by the school, there exist much opportunity to be explored, especially in the field of client / server computing.

Chapter 7 System Evaluation

University of Malaya

7.1 Introduction

This chapter consists of few sections that will emphasize on the project problems and its conclusion, advantages and strength of this system, disadvantages and limitation of this system and finally suggestion and opinions on future enhancement in this system to make it a better and complete system.

7.2 Project Problems and Solutions

7.2.1 Data gathering and analysis stage

The basic knowledge needed as a foundation in building an application of this nature involves studies in fields such as the facility management system, user management system and certain security procedures. A lot of analysis has been carried out on technological and programming concepts to grasp the concept of new rising technology. The following are some of the major problems encountered during the project system studies, analysis and development.

7.2.1.1 Difficulty in choosing appropriate development tools

There are many ways and tools available to develop a standalone management system currently as stated in the earlier chapters. Choosing a suitable technology and tool proves to be an important process as all tools has its strength and weakness. Besides, availability of the required tool for development is also a major consideration.

To determine which approach to use, seeking advises and views from project supervisor and my other project member. Finally, certain tools have been chosen due to the availability of current knowledge and allowable time schedule to pick up the skills.

7.2.1.2 Scope of system

As the name FMS implied, it is possible to build a full-scale and complete system within the given time frame. Besides that, inexperience with the present FMS technology and school information was another hindrance to implement a true

workable procedures as being implemented now in the school that consists of library, gymnasium and etc.

During the initial stage of the project, discussions were held with project supervisor and the other partner to outline the scope of the project to be built. In addition, questionnaires were distributed to few teachers with the assistance of the project supervisor. The present system flow and suggested system flow have been mentioned in Chapter three.

7.2.1.3 Time slot in facility booking

Before a certain facility is ready for booking by users, it has to be assuring that it is available at that period of time. But what happened is facility such as classroom; workshop, laboratory; art room, sewing room and etc are always used for teaching/learning purpose on weekdays. So, they are heavily used daily just on this purpose. There is no way that they can be booked for other usage during school hours.

- i) Therefore, few assumptions have been made to reflect the happenings in real situation:
- ii) For daily heavy usage facility, they are only opened for booking after school hours and on Saturday.
- iii) For other facility such as library, computer rooms, AVA room, hall and courts are available during and outside school hours. Besides, booking hour for hall and courts even can be extended until 11 p.m.

7.2.2 Implementation and testing stage

The problem faced during the initial project studies and analysis were not as critical compared to the problems faced during the implementation and testing period. Below are the problems encountered during these phases.

7.2.2.1 Development tools

Since there was no prior knowledge in Visual Basic 6.0 and Microsoft ® Access 97, there was uncertainty in programming on how to connect a Visual Basic application to a database.

Using the Visual Basic as the programming language tool, I admitted that I have made a lot of logical as well as syntax errors. But luckily, with a good Integrated Development Environment (IDE) provided in Visual Basic, debugging was easy and it was the best way to solve many logical and syntax errors.

Though there were obstacles encountered in the early stage, choosing the right tools proves to be a wise move. The learning curve was short and most of the problems faced were manageable.

Beside, looking for the right software to design the system logo is another problem. Fortunately, a free software trial version (IconForge) was found and was able to be used to produce different effect on icon.

7.2.2.2 Resources sharing

During the development of this group project, resources sharing play an important role for coordination. For illustration, icon editor software is needed to design the system icon to meet the system needs. The output must have approval from all the members. So every member needs to possess the software to make amendment. With tolerance and understanding between the members, it was handled properly in a proper manner to avoid any inconvenience among project members.

7.2.2.3 Database integrating

At earlier stage, each of the group members has design the database according to personal needs respectively. The tables' fields have to match both members' usage especially those tables involved in more than two modules. The project members were extraordinary careful while dealing with database.

7.2.2.4 Debugging message

Unclear debugging messages did not help much in fixing the bugs. Example: "Type mismatch", "Invalid use of property" and etc. Sometimes when a problem occurred, incorrect statements were written to see the debugging message. If the message was clear enough, fixing bugs in the programs will not be a tedious way forever.

7.2.2.5 Log files

Log files inform the programmer that logical errors exist in the program. Whenever a project was opened and one of the forms was selected, log files occurred during this moment if there are any logical errors in that form.

Problems with log files even harder to be traced compared to the message generated during debugging. Log files provide ambiguous information and incorrect line of code that might be the source of problems.

7.3 System Strength

The following points illustrate the strategies in the FMS system:

7.3.1 Simple and user friendly interface

The application makes full use of Window, icon, menu and pointer (WIMP) techniques, allowing the use of visual objects to navigate through the system. The use of Multiple Document Interface (MDI) will eventually reduce the needs of interface navigation. An example is the administration system, which implements this type of interface. With this interface user will not need to go back to menu again after finish one task to perform another task again. The learning curve is foreseen to be short and a user should be able to use the system with ease within minutes. Besides that, each of the systems has almost a standard interface to help user quickly familiar to another system. For example, user interfaces for user adding and viewing are almost similar.

7.3.2 User ID, password and security

FMS is a password-protected application. Each user ID is stored in the table in database. Only those who have the correct password match with his or her user ID will be able to use the system. Not even the administrator has the power to view other users' password.

Simple encryption technique has been used in the program, which just reverse the password string entered by user and stored it in the database. This provides the most basic security where even super user also cannot view other user's password.

7.3.3 Reliable system with effective error recovery

FMS is a reliable system as it caters for almost all-possible error encountered. Input by users is validate and verify through using password matching. Appropriate error feedback will be generated to user to remind or warn the user. For example, an update failure to a record will be handled by system and a user-friendly message is generated informing the user about the error and at the same time the system would recover from the error and continues to be used.

7.3.4 Query based mode for record retrieval

The user is not given option to retrieve records based on some common criteria according to suitability. For example, in the facility booking system where user can choose the wanted facility that listed in the component. This method prevents user from entering various definition of the facility.

7.3.5 System transparency

System transparency refers to the condition where the users do not need to know where the database resides, how is the system structure, its database management system and anything related to the system built. For example, the information retrieval of records is similar to a system accessing the local database. This is to ensure not to confuse users especially students in retrieving information.

7.4 System Limitations

The following list the limitations of FMS:

7.4.1 Operating system limitations

FMS requires Window 98 or above as the operating system to perform its execution. FMS is not a portable operating application (multi-platform application). It does not support other operating system, such Macintosh system, UNIX, Linux or Java Operating System.

7.4.2 Minimal system rules

Due to time constraints, FMS does not reflect a full set of system rules for all the scenarios. At this stage only sufficient rules and procedures are implemented in the system.

7.5 Future Enhancement

A system development knows no boundaries as new requirements and better implementation methods continue to arise and evolve. There are several enhancements that could extend and expand the usability of the developed system.

7.5.1 Encryption of password

Like any issue around the computer security technology, encryption is important and is necessary to protect data. At the moment, high level and complicated encryption is not implemented due to the time limitation of the project. Therefore, this module is likely to become the next enhancement of FMS if it is continue to evolve.

7.5.2 Cater to multi-platform operating system

As stated, FMS required Window 98 or above operating system for execution. In future, FMS can be tuned to fulfill other operating system such as Linux for execution. Unlike Windows operating system, Linux does not charge a penny for its operating system license to the end user compare to Microsoft ® Windows. Eventually this will reduce the cost of buying new license for Windows. Besides that,

Linux now is becoming a rising star in the end user market. More and more user has tuned into Linux.

However, tuning of operating system from Windows to the others will only be executed based on the government policy. This definitely related to the habit and skills of the publics in using which kind of operating system.

7.5.3 More administration task

Administration task can be further enhanced to include more features to ease maintenance process. Among the features may be include are report generating, booking by means of map clicking, data mining and database backup.

7.5.4 Integrating mailing capabilities

The complete FMS system is not complete without a mail service. In future, a mail function can be added into FMS and administrated through the administration section. With mailing capability available company or workshop are easily contact by the administrator. For example, the administrator will notify the company or workshop by email if the school has any tools, books, facility, teaching materials and etc to buy and repair.

7.5.5 Enable web-based functionality

There are lot functions can be tuned into web-based system. For example, online booking through browser. Even though students are not allowed to have account to use the system, some of them might be chosen and given the account in future. For example, the monitors of each class might have the privilege to use the system compare to the others.

7.5.6 Enable map-based booking

Booking will be an easier task if 'clicking' on a dot of a map can do it. This required certain technique or may be involved some software tools other than what was used at the present.

7.6 Knowledge and experience gained

Throughout the whole process of analysis, design, implementation, testing and evaluation, technical techniques have been learned to build a complete and accurate system.

Other software tools other than the main software (Visual Basic 6.0) such as IconForge and Microsoft Help Workshop has been used to support some features in the system. Use of these supplementary tools makes the system more complete and fulfill the system requirements.

The spirit of group is important and thus has been carried through the whole process in order to complete the full system in time.

7.7 Overall conclusion

Overall, FMS has achieved and fulfilled the objectives and requirements as a FMS as determined during system analysis. FMS is not only a facility management system useful for the teachers and staffs but also includes an administration module for administrator.

There was a lot of knowledge gained throughout the development of FMS. These include knowledge gain in standalone system with integration with database technologies, programming and concepts as well as database. Programming in Visual Basic proves to be a valuable experience.

Visual Basic technology with integration to Microsoft Access as the back end turned out to be the most suitable technologies to develop such a system. But while programming skills and technologies are important in developing the system, it must not be discounted that in any system development, good software engineering techniques must also be applied. Here theory and knowledge gained throughout the course of computer science studies like system analysis, design and software engineering were literally put into practice.

Therefore, there are still many rooms for improvement in FMS, in terms of implementing a comprehensive FMS and gearing towards online system maintenance. With the first step taken, enhancement could still made with more features added for future versions.

Finally, with project of this nature affected by the school, there exist much opportunity to be explored, especially in the field of client / server computing.

University of Malaya

Glossary

Data. Facts concerning things such as people, objects and events.

Database. A collection of related information stored in a structured organized way. Using this structured collection, standard methods of retrieve can be used to retrieve the data.

DBMS. A set of programs that allows users to access a database in a logical way.

Design methodology. A set of design based on a concept.

Entity. A thing about which an organization chooses to record data.

E-R model. A detailed logical representation of the entities, associations and the data elements for an organization or business.

Methodology. A process to accomplish a design goal together with a set of design objects that is manipulated to support the process.

Module. A set of logically related program structures that perform specific function.

Primary key. A candidate key that has been selected as the identifier for an entity type. Primary key values may not be NULL.

Sub-system. Components of a system that are systems themselves.

System. A set of related components that work together to fulfill a purpose.

References

- Barbara C. McNurlin, Ralph H. Sprague, Jr., 1998. "Information System Management In Practice". 4th Ed. Prentice Hall, Inc..
- The Computer Bulletin July 1998, "System Development – Charting a path to project success". Pg 26-27
- The Computer Bulletin May 1998, "Project Management – Human barriers to project success". Pg 24-25
- The Computer Bulletin March 1998, "Internet". Pg27-29
- David M. Kroenke, 1998. "Database Processing - Fundamentals, design, and Implementation". 6th Ed. Prentice Hall, Inc..
- David Solomon, Roy Rankins, 1998. "Microsoft® SQL Server 6.5". SAMS Publishing.
- Deitel & Deitel, T.R. Nieto, 1999. "Visual Basic 6 How To Program". Prentice Hall.
- Dennis Kennedy, Joe Garrich, Bill Harper, Jason T.Roff, 1997. "Visual Basic 5 Database How-To", Waite Group Press.
- Jeffrey A. Hoffer, Joey F. George, Joseph S.Valacich, 1996. "Modern Systems Analysis and Design", The Benjamin/Cummings Publishing co, Inc..
- John Connell, 1998. "Beginning Visual Basic 6.0 Database Programming". Wrox Press Ltd.
- Julia Case Bradley, C. Millsbaugh, 1999. "Programming in Visual Basic 6.0". McGraw-Hill International Edition.

Kenneth E., Kendall and Julie E. Kendall, 1998. "System Analysis and Design". Prentice Hall.

[1] Physical Education Unit - CUNY

Mike Mckely, Jeff Spotts, and Brian Siler, 1997. "Special Edition Using Visual Basic 5", 2nd Edition, Que.

[2] Home of EventPro, complete and easy to use Event Management Software for all your

Peter Rob, Carlos Coronel, 1997. "Database System: design, implementation and management". Cambridge, MA: Course Technology.

[3] The Visual Basic Code Network

Raymond McLeod, Jr, 1998. "Management Information Systems". 7th Ed. Prentice Hall, Inc..

<http://www2.healthsci.vic.edu.au/~gsc/Service/Booking.asp>

Roger S. Pressman, 2001. "Software Engineering : A Practitioner's Approach". 5th Ed McGraw-Hill.

http://www.searchiv.com/sear/998_Editions_Picks_Page/0/0/1/2/2,00.html

Smart School Flagship Application, 1997. "The Malaysian Smart School A Conceptual Blueprint. Ministry of Education Malaysia 26 July 1997". Percetakan Watan Sdn. Bhd.

<http://www.aivbcode.com>

Thari Lawrence Pfleeger, 1998. "Software Engineering: Theory and Practice", Prentice Hall.

[4] Visual Basic Code Network

William Stallings, 1999. "Computer Organization And Architecture - Designing for Performance". 4th Ed. Prentice Hall, Inc..

[5] Welcome To Best Visual Basic

<http://vbcode.com>

[6] Welcome to the home of the Visual Basic

http://www.xplainer.com/~mrg/visual_basics.html

[7] Utility Programs with Visual Basic Source Code

<http://www.madeprog.com/~johncarter/visualbas.htm>

[10] <http://www.dataexperiment.net/VBcode/bugsmethods.html>

<http://www.thevisualcode.com/>

Internet Resources

- [1] Physical Education Unit - CUHK
http://www.peu.cuhk.edu.hk/www_peu/
- [2] Home of EventPro, complete and user friendly Event Management Software for all your booking and event scheduling needs
<http://www.eventpro.net/>
- [3] Faculty of Health Science – Booking Information System – View Booking
<http://www2.healthsci.utas.edu.au/scripts/Car/viewBooking.asp>
- [4] http://www.searchvb.com/searchVB_Editors_Picks_Page/0,1947,4e2,00.html
http://www.searchvb.com/searchVB_Editors_Picks_Page/0,1947,4e2,00.html
- [5] A1VBCode – free VB source code
<http://www.a1vbcode.com>
- [6] Visual Basic Code Network
<http://www.vbcn.com.ar>
- [7] Welcome To Best Free vbCode
<http://vbcity.com/vbcode>
- [8] Welcome to the home of the XPLOITER – Visual Basic
<http://www.xploiter.com/programming/vb/index.shtml>
- [9] Utility Programs with Visual Basic Source Code
<http://www.mindspring.com/~johnecarter/vbsource.htm>
- [10] <http://seduto/hypermart.net/Vbsearchegine/index.html>
<http://www.thevisualcore.com/>

- [11] Visual Basic source code
http://vbindex.virtualave.net/page/Visual_Basic.shtml
- [12] VB Source Code Search Engine
<http://seduto.hypermart.net/Vbsearchengine/index.html>
- [13] VisualBasicSource
<http://www/kather/net/VisualBasicSource/scindex.html>
- [14] SoftSeek.com - Microsoft HTML Help Workshop by Microsoft Corporation
http://softseek.zdnet.com/Programming/Help_File_Tools/Review_12864_index.html
- [15] SoftSeek.com – IcaonForge (32-bit) by CursorArts Company
http://www.softseek.com/Desktop_Enhancements/Icons/Utilities_and_Tools/Review_10597_index.html