

DIRECT ENTRY (BITI) TIMETABLE SCHEDULING USING HEURISTIC SCHEDULING TECHNIQUE

LIYANA SAFRAA BINTI AHMAD



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL: DIRECT ENTRY (BITI) TIMETABLE SCHEDULING USING
HEURISTIC SCHEDULING TECHNIQUE

SESI PENGAJIAN: 2010/2011

Saya LIYANA SAFRAA BINTI AHMAD
(HURUF BESAR)

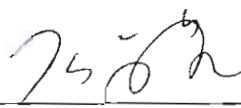
Mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)


 SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

 TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

 / TIDAK TERHAD



(TANDATANGAN PENULIS)
Alamat tetap: F1/901 KG SEPAKAT
26800, ROMPIN
PAHANG



(TANDATANGAN PENYELIA)
NUZULHA KHILWANI IBRAHIM
Nama Penyelia

Tarikh: 8/7/2011

Tarikh: 8/7/2011

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Project Sarjana Muda (PSM)

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa

DIRECT ENTRY (BITI) TIMETABLE SCHEDULING USING HEURISTIC
SCHEDULING TECHNIQUE

LIYANA SAFRAA BINTI AHMAD

This report is submitted in partial fulfilment of the requirements for the
Bachelor of Computer Science (Artificial Intelligence)

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DECLARATION

I hereby declare that this project report entitled

DIRECT ENTRY (BITI) TIMETABLE SCHEDULING USING HEURISTIC SCHEDULING TECHNIQUE

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT :  Date: 8/7/2011
(LIYANA SAFRAA AHMAD)

SUPERVISOR :  Date: 8/7/2011
(NUZULHA KHILWANI IBRAHIM)

DEDICATION

To my beloved parents and friends..

ACKNOWLEDGEMENTS

I would like to thank Miss Nuzulha Khilwani for giving me assistant to complete this project successfully..

I would also like to thank both my beloved parents and friends who have been giving me support and motivation throughout my project..

ABSTRACT

The idea of doing this project comes up when direct entry student complaining on the difficulty of scheduling their timetable since they have to face several procedures to schedule the timetable. For example they have to collect all of timetable of other class from academic assistant, head of department or faculty office. The system had been developed used to help the students by providing a tool for scheduling their timetable. In developing the system, heuristic scheduling technique or algorithm is used to have a good result of timetable generated. Heuristic scheduling consists of three approaches which are order-based, resources-based and operation-based.

ABSTRAK

Idea untuk menghasilkan projek ini adalah apabila aduan para pelajar kemasukan terus kerana mereka perlu melakukan beberapa perkara untuk menyusun jadual mereka. Sebagai contoh, mereka perlu mengambil kesemua jadual setiap kelas dalam jurusan yang sama dari pada perunding akademik, ketua jabatan atau pejabat fakulti. Sistem ini dihasilkan bertujuan untuk menyediakan satu kemudahan bagi pelajar untuk menyusun jadual mereka. Dalam menyiapkan system ini, teknik penjadualan heuristik atau algoritma heuristik ini dapat membantu menghasilkan sistem untuk menyusun jadual dengan baik. Penjadualan heuristik ini mengandungi tiga pendekatan iaitu berdasarkan susunan, berdasarkan sumber dan berdasarkan operasi.

TABLE OF CONTENT

CONTENT	PAGE
DECLARATION	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
ABSTRAK	vi
TABLE OF CONTENT	vii
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xii
LIST OF ATTACHMENT	xiii
CHAPTER I – INTRODUCTION	1
1.1 Project Background	1
1.2 Problem Statement	2
1.3 Objective	3
1.4 Scope	3
1.5 Project Significance	4
1.6 Expected Output	4
1.7 Conclusion	4
CHAPTER II – LITERATURE REVIEW AND PROJECT METHODOLOGY	5
2.1 Introduction	5
2.2 Facts and Findings	5
2.2.1 Domain	5
2.2.2 Existing System	6
2.2.1 Technique	7
2.3 Project Methodology	11
2.3.1 Structured Systems Analysis and Design Method (SSADM)	11
2.3.2 Database Life Cycle (DBLC)	14
2.4 Project Requirements	18
2.4.1 Software Requirements	18

2.4.2	Hardware Requirements	19
2.5	Project Schedule and Milestones	20
2.6	Conclusion	22
CHAPTER III – ANALYSIS		23
3.1	Introduction	23
3.2	Problem Analysis	23
3.2.1	Background of Current System	23
3.3	Requirement Analysis	24
3.3.1	Functional Requirement	25
3.3.2	Non-functional Requirement	27
3.4	Conclusion	29
CHAPTER IV – DESIGN		30
4.1	Introduction	30
4.2	High-Level Design	30
4.2.1	User Interface Design	30
4.2.3	Database Design	35
4.3	Detailed-Level Design	40
4.3.1	Software Design	40
4.4	Conclusion	41
CHAPTER V – IMPLEMENTATION		41
5.1	Introduction	41
5.2	Software Development Environment Setup	41
5.3	Software Configuration Management	42
5.3.1	Configuration Environment Setup	42
5.3.2	Version Control Procedure	42
5.4	Implementation Status	43
5.5	Conclusion	45
CHAPTER VI – TESTING		46
6.1	Introductions	46
6.2	Test Plan	46
6.2.1	Test Organization	46
6.2.2	Test Environment	47
6.2.3	Test Schedule	47
6.3	Test Strategy	47
6.3.1	Classes of Tests	48
6.4	Test Implementation	48
6.4.1	Test Description	49
6.4.2	Test Data	50
6.5	Test Result and Analysis	50
6.6	Conclusion	53

CHAPTER VII - PROJECT CONCLUSION	54
7.1 Observation on Weakness and Strength	54
7.1.1 Weaknesses	54
7.1.2 Strengths	55
7.2 Proposition for Improvements	55
7.3 Contribution	55
7.4 Conclusion	56
REFERENCES	57
BIBLIOGRAPHY	58
ATTACHMENTS	59

LIST OF TABLES

TABLE	TITLE	PAGE
	Table 2.1: List of Software Requirements	18
	Table 2.2: List of Hardware Requirements	19
	Table 2.3: Project schedule and milestone for PSM 1	20
	Table 2.4: Project schedule and milestone for PSM 2	21
	Table 4.1: Database (table name: subject)	36
	Table 4.2: Database (table name: member_login)	37
	Table 4.3: Database (table name: student)	37
	Table 4.4: Database (table name: contact)	38
	Table 5.1: Version Control for Direct Entry (BITI) Timetable Scheduling System	43
	Table 5.2: Implementation Status for Direct Entry (BITI) Timetable Scheduling System	45
	Table 6.1: List of testing during implementation	49

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 2.1:	Abstract domain model	8
Figure 2.2:	Problem space of scheduling	8
Figure 2.3:	Structured Systems Analysis and Design Method (SSADM)	14
Figure 2.4:	Database Life Cycle (DBLC)	15
Figure 3.1:	Flowchart of manual timetable scheduling.	24
Figure 3.2:	Functional Requirement	25
Figure 4.1:	Login page.	31
Figure 4.2:	Home page.	31
Figure 4.3:	Contact page.	32
Figure 4.4:	Enroll page.	32
Figure 4.5:	Timetable display page.	33
Figure 4.6:	Search Page.	33
Figure 4.7:	Print Popup Window.	34
Figure 4.8:	Save popup window.	34
Figure 4.9:	Entity Relationship Model Diagram (ERD)	35
Figure 6.1:	Question 1	50
Figure 6.2:	Question 2	51
Figure 6.3:	Question 3	51
Figure 6.4:	Question 4	52

LIST OF ABBREVIATIONS

BITC	-	Bachelor of Science Computer (Networking)
BITD	-	Bachelor of Science Computer (Database Management)
BITI	-	Bachelor of Science Computer (Artificial Intelligence)
BITM	-	Bachelor of Science Computer (Media Interactive)
BITS	-	Bachelor of Science Computer (Software Engineering)
FICT	-	Faculty of Information and Communication Technology
UTeM	-	Universiti Teknikal Malaysia Melaka

LIST OF ATTACHMENTS

ATTACHMENT	TITLE	PAGE
1.1	Example Coding	59
1.2	Questionnaire	65
1.3	User Manual	71

CHAPTER I

INTRODUCTION

1.1 Project Background

Basically, this project is for direct entry students which need to waive certain subjects and schedule the timetable. For the timetable scheduling, subjects will be scheduled according to timetable that already scheduled for other students. Since direct entry students usually do not have exactly the same subject as students who are not direct entry, their timetable will be different from other students. Subjects will be selected and scheduled according to subjects that offered for students to enroll. After subjects had been selected, timetable for current semester will be scheduled and other subjects will be listed out for students to enroll them on next semesters.

This project is under education field which will be used by university that have intake for direct entry. The problem is there is no artificial intelligent element in the existing approach. I choose to implement a Knowledge Base System (KBS) to innovate the system for the problem because study of the KBS greatly interested me, and I decided to take this project as an opportunity to study KBS more than trying to achieve a solution for the problem at hand.

1.2 Problem Statement

There are a few weaknesses in existing approach. The problems and weaknesses can be summarized as follows:

- **Timetable is manually scheduled.**

The existing approach that been used is in manually. Students need to check the timetables of every class to schedule their timetable.

- **Takes a lot of time to finish scheduling.**

Obviously, this approach takes a lot of time for student to plan their timetable. Because of this issue, student will do the scheduling without careful and they might have two classes on same time issue.

- **Big percentage of having two classes on same time.**

Two or more subject that the students enroll could have same lecture time. Whenever this matter happened, students need to reschedule their timetable. They need to start all over again to schedule the time table.

Regarding from those problems, new approach are needed to solve it. This can help avoid the students from having problem on timetable scheduling. In additional, Head of Departments also needs to spend more time to help student handle the problems.

1.3 Objective

Based on the problems from the current approach, the objectives in developing a new approach are:

- **To provide a tool for direct entry students plan their timetable.**
This project can provide tool for student to plan their timetable is easier way instead of write down the subjects on paper form and schedule by their own self.
- **Less time used to plan timetable.**
Student will spend less time to schedule their timetable. So, they did not miss any syllabus in subjects while they are busy planning their timetable.
- **To avoid having two or more classes on same time.**
Whenever there are two or more class happen to be on same time, that subject will be listed on other semester while other subject n other semester that can be enroll on that current semester will be listed in subjects need to schedule.

By using new approach, those problems can be solved and it can help students handle the subjects and timetable scheduling matter. Even Head of Departments could use less time to help student dealing with timetable issues.

1.4 Scope

This project is under education field which will be used by Faculty of Information and Communication Technology (FICT), Universiti Teknikal Malaysia Melaka (UTeM). Target users obviously will be the direct entry students of BITI course by using existing timetable from BITI. The students need to enroll in range of

more or equal to 11 until less or equal to 20 of total credit hours and with that total amount of credit hours, the students able to schedule their timetable using the system.

1.5 Project Significance

The significance of the project is to overcome planning time and manual scheduling of class timetable. It also gives the benefit to the students and the Head of Departments. This project is been done to ensure that the way of scheduling timetable will be improve and help students plan the timetable with been monitored by Head of Departments.

1.6 Expected Output

I hope this project is success and can be used to help direct entry student plan their timetable. In additional, I will get experience of working on something new using KBS.

1.7 Conclusion

This chapter had explain about the basic idea of the project such as the background of project, why I come up with this project in problem statement, the objective of this project and scope of project. On next chapter, I will be discussing about literature review and project methodology.

CHAPTER II

LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter specifies literature review and project methodology that will be opted to optimize the development of to-be application. Literature review includes study and research of published materials like journals, thesis, case studies, technical documents and online library. Generally, the purpose of a review is to analyze critically a segment of a published body of knowledge through summary, classification and comparison of prior research studies, reviews of literature, and theoretical articles. Project methodology describes a set of practices that will be carried out iteratively to produce the application.

2.2 Facts and Findings

Below are some findings in process to build the system.

2.2.1 Domain

This project, scheduling the class timetable for FICT direct entry students (BITI) using genetic algorithm is actually in educational field. This is because it

combines the technology with education purpose. It is a technology because of using the genetic algorithm technique and it change the manual approach.

2.2.2 Existing System

After doing some research, there is no exiting system for direct entry but there is a system to schedule timetable for all students which in current semester not the direct entry.

2.2.2.1 FTMK Timetabling System

Background

This system is actually a good system to schedule timetables. The system was built by team of lecturers and also people from IT department. The system will schedule the timetables for group one and two of year one, two and three for each major course which is BITI, BITM, BITC, BITD and BITS.

Problem Statement

This system has a weakness regarding the situation of student enrolling the subject. The weakness is as follow:

- **No timetable scheduling for direct entry**
Since UTeM has direct entry students, which the students may transfer some credit and start the study from any semester or year. Not like the students who start enroll as the first semester of first year student.

2.2.2.1 Direct Entry Timetable Manually schedule

Background

This manual approach is basically a way where students need to plan their timetable according to subject can be enrolled on current semester. Students need to see the Head of Department to confirm the timetable so that there is no more than one class in a time.

Problem Statement

The weaknesses are as follow:

- **Have more than one class in a time**
Because of this approach is done manually, most of student did not take it seriously. They just randomly select the class without noticing they have more than one class on the same time.
- **Using a lot of time**
Whenever students have more than one class in a time, they need to reschedule the timetable.

2.2.3 Technique

Technique that will be used to schedule the time table is as follow:

2.2.3.1 Knowledge Base System

Scheduling problem can be visualized by AND/OR tree which constructed by the combination of production requirement with the order given and scheduling

horizon. Finding a solution in scheduling is finding a solution of the root node, all successors of an AND node and OR node.

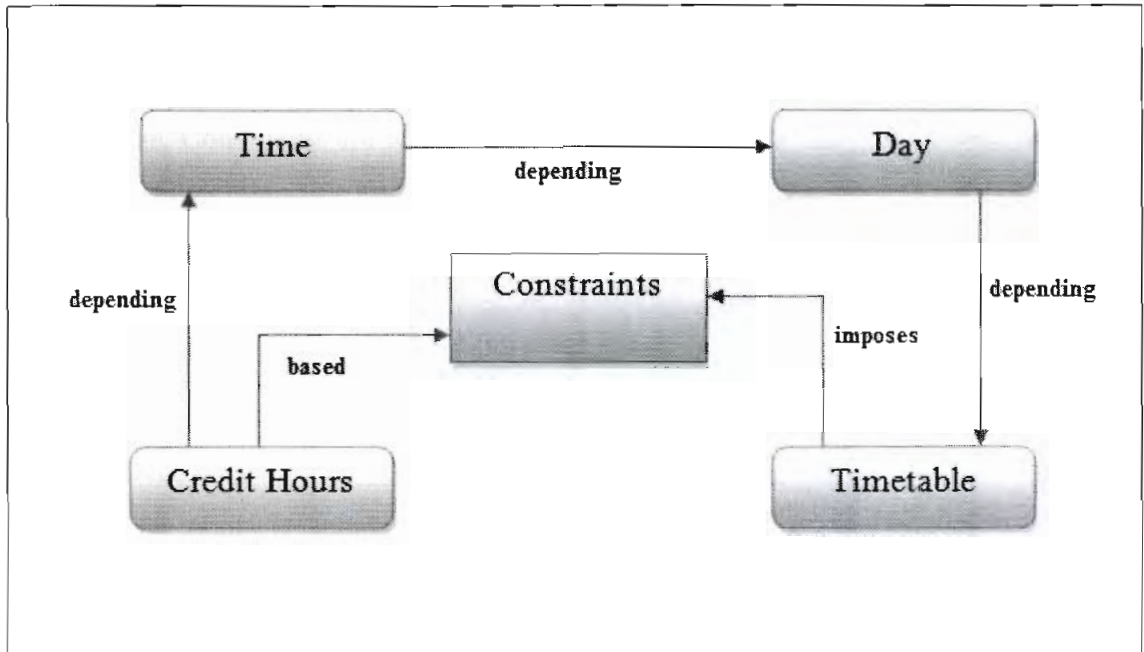


Figure 2.1: Abstract domain model

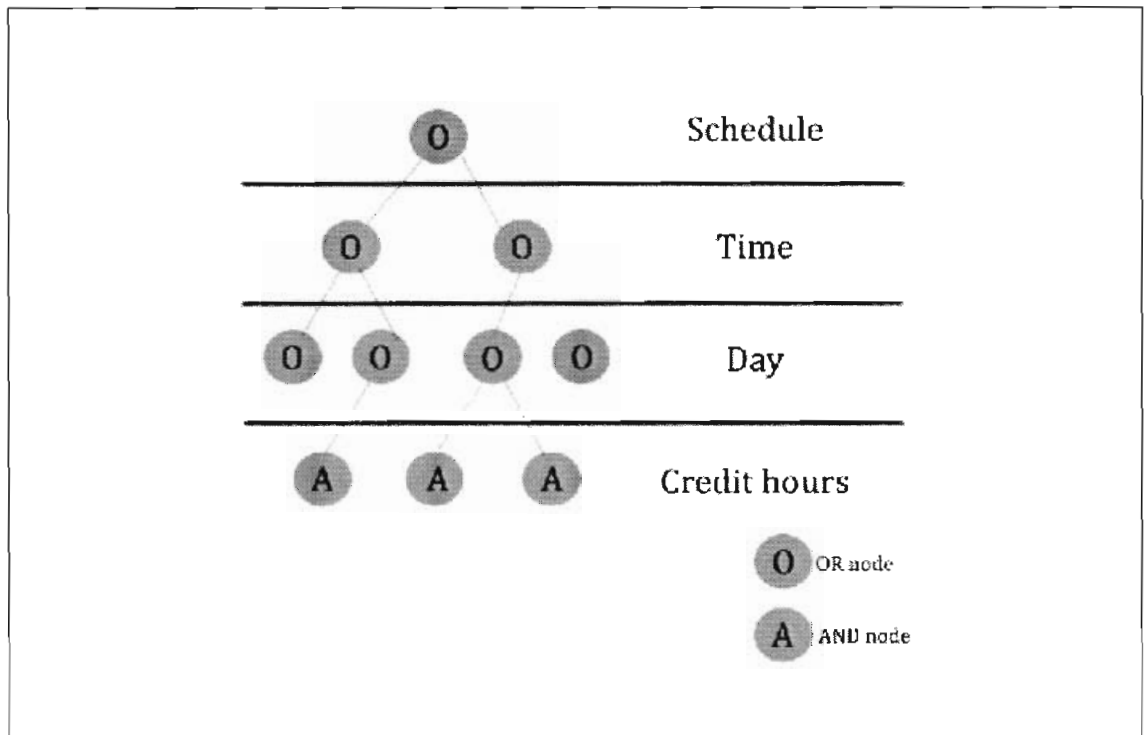


Figure 2.2: Problem space of scheduling

2.2.3.1.1 Heuristic Scheduling

This scheduling technique is one of technique that can be use for knowledge based system. Heuristic scheduling is based on heuristic search technique and general heuristic such as problem decomposition together with problem-specific knowledge. Constraints are widely used to guide the search and the representation of knowledge can be in rule or frame based. Most of the approaches used to scheduling can be categorized by the underlying perspective of the problem decomposition as following:

- **Order-based**

The order is selected from all unscheduled orders and places it as first operation that can be followed with next operation after the first operation is done. Then it will be schedule based on the time the operation takes to finish. For most subject offered for BITI students have 2 hours of class times.

- **Resource-based**

Resource is selected and suitable operation is chosen out of the set of operations. It means the timetables scheduled for direct entry students are based on existing timetable from other BITI students.

- **Operation-based**

Operation after operation is selected and scheduled and it is based on what subject that the direct entry student select to enrol and generate the timetable for them.

Example of using the three approaches can be seen in Appendix A. Following is the algorithm used in this system.

WHILE subject to schedule

Select subject

Select credit hour for subject

WHILE constraints to schedule

Select day

Select time

Select class

Schedule timetable or solve conflict

END WHILE

ENDWHILE

2.2.3.1.2 Constrains

Constrains can be divide into two type of constraints which are soft and hard. Hard constraint is requirement needed and must be consider building the system while soft constraint is also requirement needed but not as much as hard constraint. The system can be done without fulfilling the soft constraint requirement.

Hard Constrains

- **No students have more than one class at a time**
Having student with more than one class on the same time should be avoided.
- **Schedule according the subject requirement.**
Some subject should be enrolled first before enrolling the other subjects on next semester.
- **Credit hour should not more than 20 in total for a semester.**
Total of subjects' credit hour for a semester should less or equal to 20 in total.

Soft Constrains

- **Some students do not wish to have empty periods in their timetables.**
Student might do not want to have empty period in their timetable since they want to use less time in campus.
- **Some students do not wish to have classes assigned consecutively in time.**
Student might want some free time before starting the next class.

2.3 Project Methodology

Methodology is a set of methods that define the process and order of how something is to be achieved. There has been paradigm shift from instructions to constructivist and that this has, in turn, reflected a change in methodologies from breaking down the project into single step tasks to a more holistic approach. The selection of methodology that suitable will support in building quality project and will give a lot of benefit to the developer.

Ensures the consistent and reproducible approach that applies to the determination and design of business solutions are known as good methodology. In addition, a rigorous methodology serves to minimize or eliminate many of the risks and pitfalls commonly associated with taking shortcuts or making errors. So that, a sensible methodology results in a consistent and those working can quickly and easily retrieve comprehensive.

2.3.1 Structured Systems Analysis and Design Method (SSADM)

For this project, I used Structured Systems Analysis and Design Method (SSADM) as one of the methodology. SSADM is a systems approach to the analysis and design of information systems. It uses a combination of text and diagrams throughout the whole life cycle of a system design, from the initial design idea to the

actual physical design of the application SSADM uses a combination of three techniques as below:

- **Logical Data Modelling**

This model is the process of identifying, modelling and documenting the data requirements of the system being designed. The data is separated into entities which are things about which a business needs to m r d information and relationships which the associations between the entities.

- **Data Flow Modelling**

This model is the process of identifying, modelling and documenting on how data moves around the information system. It examines processes which activities that transform data h m one form to another, data stores which is the holding areas for data, external entities which what sends data into a system or receives data from a system, and data flows which is route by which data can flow.

- **Entity Behaviour Modelling**

The process of identifying, modelling and documenting the events that affects each entity and the sequence in which these events occur.

Each of these three system models provides a different viewpoint of the same system, and each viewpoint is required to form a complete model of the system being designed. The three techniques are cross-referenced against each other to ensure the completeness and accuracy of the whole application.

SSADM is suits for both large and small projects because it is scalable. In any size or type of application, only selected elements which are useful to the project are used. In additional, it will provide you with a method which is effective. SSADM is rigorous but not done by only the technique that have its own rules, but also the method which allows you to cross-check the quality and completeness of the deliverables of one technique against others. SSADM avoids user being locked in to a specific hardware or software environment since it separates the logical design from the physical implementation. Meaning that, a user can implement the solution

on any number of different hardware and software environments without changing the logical design.

SSADM application development projects are divided into five modules that are further broken down into a hierarchy of stages, steps and tasks which are shown below:

- **Feasibility Study**

This project is done to help direct entry students plan their class timetable.

- **Requirements Analysis**

The main requirement of having successful project is efficiently scheduling the timetable according to subjects that student enrolled and time of the classes.

- **Requirements Specification**

Functional requirement is the genetic algorithm technique should able to help student planning their timetable and reduce the time of scheduling compare to manually schedule the timetable. While the non-functional requirements are the time should consumed to make the system operate efficiently and the cost are needed to accomplish this project.

- **Logical System Specification**

The system should able to place the suitable classes on the timetable without having more than one class on the same time.

- **Physical Design**

This project requires data for other student group which are group one and group two of each year one, two and three of BITI student.

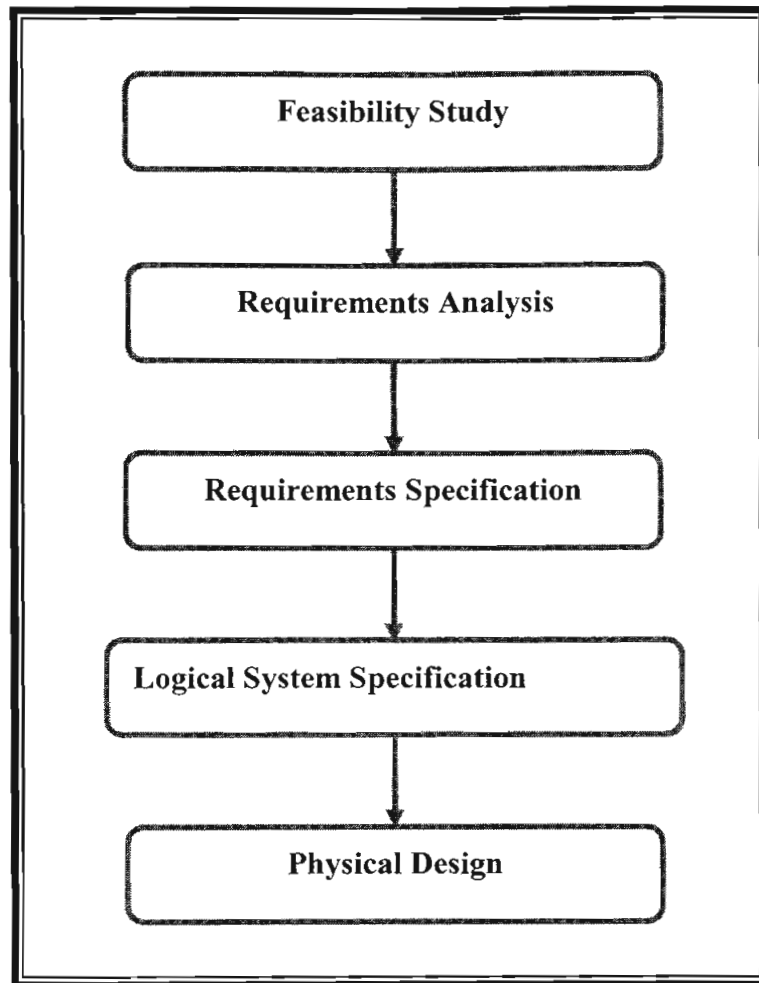


Figure 2.3: Structured Systems Analysis and Design Method (SSADM)

2.3.2 Database Life Cycle (DBLC)

Phases of Database Life Cycle (DBLC) are database initial study, database design, implementation and loading, testing and evaluation, operation, and maintenance and evolution. The process that will be used in order to implement this system is the DBLC which is available to database development process. DBLC is similar cycle applies to the databases. The database is created, maintained and enhanced. When even enhancement can no longer stretch the database's usefulness and the database can no longer perform its function adequately, it may have to be replaced. The six phase's flows of Database Life Cycle are shown as following:

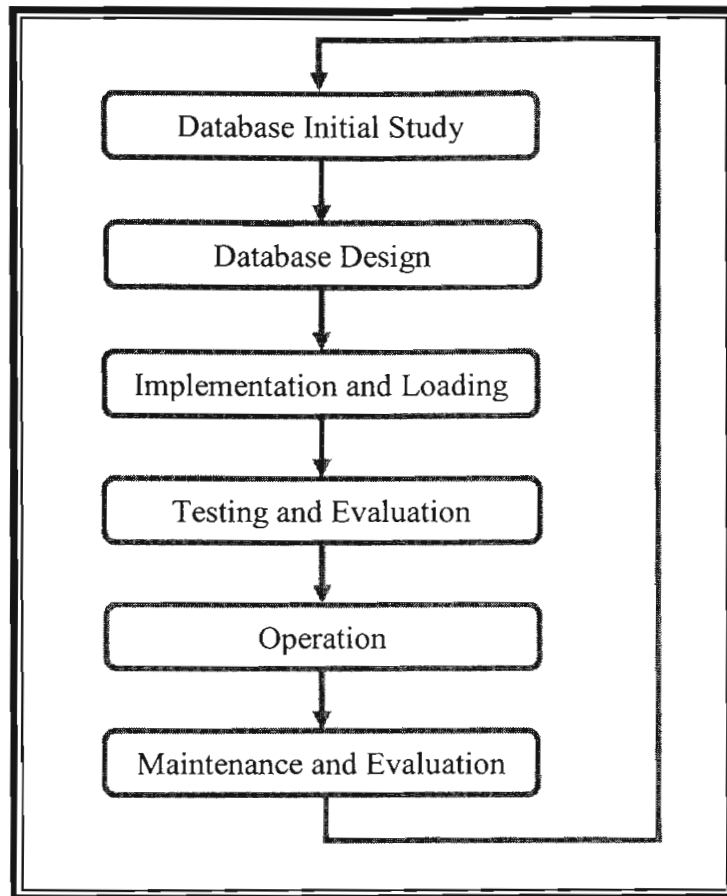


Figure 2.4: Database Life Cycle (DBLC)

Phase 1: Database Initial Study

- Analyze company situation
 - Operating environment
 - Organizational structure
- Define problems and constraints
- Define objectives

To analyse the company situation, two other components need to be considered which are the faculty environment and the organizational structure. Do the observation to obtain the problems and constraints of the current system and the constraint that facing during developing the system. Objectives and scopes of the

system need to be define so that new idea can be thought about to improve the existing system.

Phase 2: Database Design

- Most Critical DBLC Phase
- Makes sure final product meets requirements
- Focus on data requirements
- Sub phases
 - Create conceptual design
 - DBMS software selection
 - Create logical design
 - Create physical design

The system should be design according to requirement that listed down. To have a successful project, data requirement should be focused on more. Sub phases for this phase are creating the conceptual design for the database, selecting the suitable DBMS software as The database software will be used in order to develop this system is MYSQL, creating the logical design and also the physical design for the database which is data storage that will be chosen. While logical design means developing the coding of the system that related to the database system.

Phase 3: Implementation and Loading

- Data loaded into tables
- Other issues
 - Performance
 - Security
 - Backup and recovery

In this phase, data are loaded inside database and it also describes the processes installing the DBMS software that will be used in this system, creating the database and loading and converting the data.

Phase 4: Testing and Evaluation

- Database is tested and fine-tuned for performance, integrity, concurrent access, and security constraints
- Actions taken if tests fail
 - Fine-tuning based on reference manuals
 - Modification of physical design
 - Modification of logical design
 - Upgrade or change DBMS software or hardware

This phase starts when the data have been loaded into the database. Supervisor will test and fine-tuned the databases for performance, integrity, concurrent access, and security constraints while the tester will test the database whether it is working or not. Whenever the system is fail, the system will be modified.

Phase 5: Operation

- Database considered operational
- Starts process of system evaluation

In this phase, database time table will be created based on type of database needed and amount of timetable needed.

Phase 6: Maintenance and Evaluation

- Preventative maintenance
- Corrective maintenance
- Adaptive maintenance
- Assignment of access permissions
- Generation of database access statistics to monitor performance
- Periodic security audits based on system-generated statistics

While in this phase, the entire database created will be monitored and modified or update from time to time to get a suitable and good performance of database.

2.4 Project Requirements

2.4.1 Software Requirements

The software requirements must be clear and understandable to ensure the enhanced system can be implemented.

Table 2.1: List of Software Requirements

Software	Function
Microsoft Windows 7	Operating System
Adobe Dreamweaver	Main Server Side Language
Microsoft Internet Explorer (IE)	Web Browser
Wamp	Server
MYSQL	Database System
Microsoft Word 2007	Documentations
Microsoft Project 2007	Tasks Schedule and Chart
Microsoft Power Point 2007	Presentations

2.4.2 Hardware Requirements

The hardware requirement is also very important because it supports the software requirements.

Table 2.2: List of Hardware Requirements

Hardware	Specification
Processor	Any compatible processor recommended.
Monitor	15, 17 Inch (High Resolution 600 x 800)
Memory	1 GB of RAM or higher.
Hard Disk	10 GB of hard disk free space (20GB or higher recommended)

2.5 Project Schedule and Milestones

Following the schedule of project deliverables:

Table 2.3: Project schedule and milestone for PSM 1

Week	Activities	Deliverables
1	Chapter 1 & 2 Project evaluation	Introduction, literature review and methodology report.
2		Project evaluation report.
3	Chapter 3 Project evaluation	Requirement analysis report.
4		Project evaluation report.
5	SEMESTER BREAK	
6	Chapter 4 Project evaluation	Design report.
7	Full report	Project evaluation report.
8	Preparation for final presentation Full report	Full report
9	Final Presentation	Project evaluation report.

* Subjects to change

Table 2.4: Project schedule and milestone for PSM 2

Week	Activity	Deliverable
1	Chapter 4: Design Chapter 5: Implementation	Chapter 4
2	Chapter 5: Implementation Project evaluation	Project evaluation report
3	Chapter 5: Implementation Chapter 6: Testing	Chapter 5
4	Chapter 6: Testing Project evaluation	Project evaluation report
5	Chapter 6: Testing Chapter 7: Conclusion	Chapter 6
6	Chapter 7: Conclusion Draft full report	Chapter 7
7	Final Presentation	Draft report
8	Repair draft report	
9	Full Report Submission	Full report (3 copies) CD (1 copies)

* Subjects to change

2.6 Conclusion

This chapter describes fact findings that are related to the project title through literature reviews of published materials. As the conclusion, research and case study are very important to find out the ideas or facts from the current system. Besides, according to the facts from existing system there are problems that might be able to solve in new system.

Project methodology is another important element in developing a successful system based on the fact findings, appropriate approaches and project requirements are stated to carry out the development of the prototype. The milestone is scheduled to help implement the system and finish it on time. Next chapter will be discussing about the analysis of the project.

CHAPTER III

ANALYSIS

3.1 Introduction

The analysis is where the project lifecycle begins. The only exception is a situation where you have broken a large project down into smaller components. In that case, you may have an entire project that is only focused on the analysis phase, while the next project may start at the design phase. The analysis is also part of the project where you identify the overall direction that the project will take through the creation of the project strategy documents.

3.2 Problem Analysis

3.2.1 Background of Current System

Before scheduling the timetable, student will come to see the head of department to select subjects that can be transfer the credit hour. Then the students will schedule the timetable according to subject they want to enroll. All this steps are done manually. Students are given a blank printed timetable and they need to fill that table by them self.

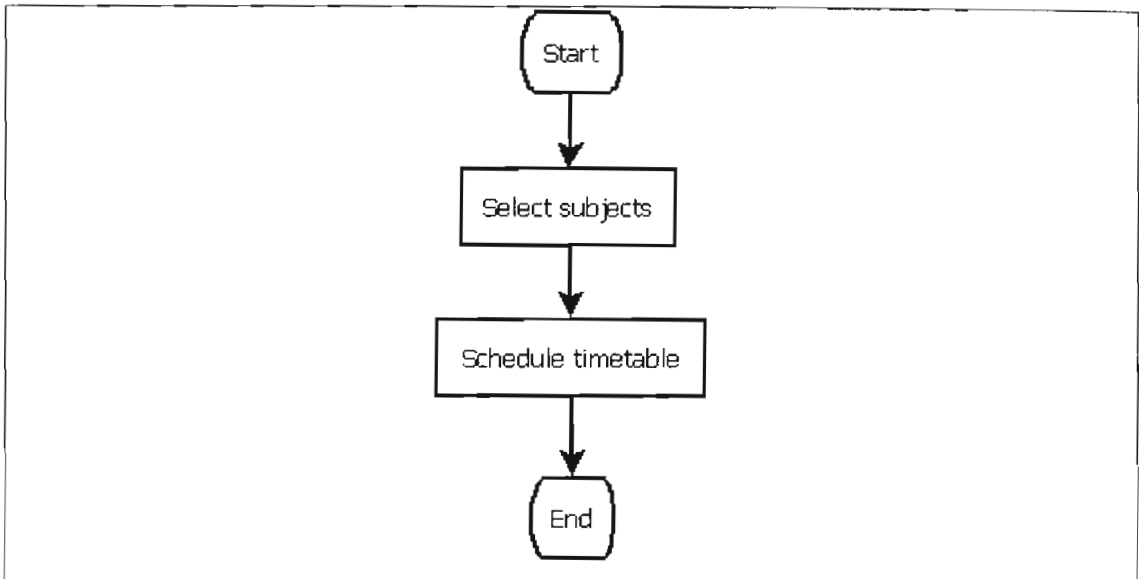


Figure3.1: Flowchart of manual timetable scheduling.

3.3 Requirement Analysis

Gathering requirements is the main attraction for analysis. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down. Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements. This formal process, which will be developed in more detail, consists of four basic steps:

- Elicitation – I ask questions, you talk, I listen
- Validation – I analyze, I ask follow-up questions
- Specification – I document, I ask follow-up questions
- Verification – We all agree

Although gathering requirements is the main focus during the analyzing, there are other important activities during this phase as well. One is to create a requirement management plan to define how the requirements will be documented, communicated, tracked and changed throughout the rest of the project lifecycle.

This plan will specifically address establishing a baseline, a change control process, and a way to track the requirements through the rest of the lifecycle. Another important activity is to set the overall direction for work that does not take place until later. This is accomplished through a series of strategy documents. For instance, once you have your requirements, you can start to set the overall direction for training in a training strategy document. The strategies are at a high-level and are later defined at a lower level before they are finally implemented toward the end of the project.

3.3.1 Functional Requirement

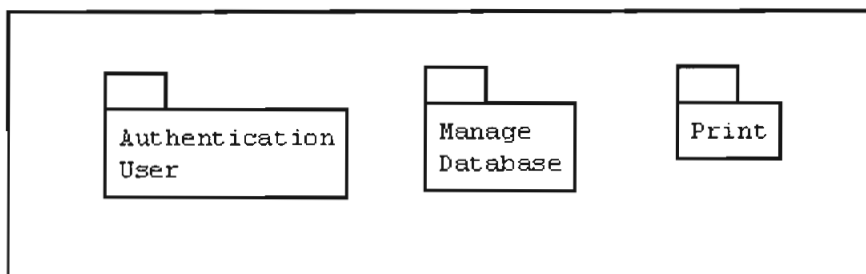


Figure 3.2: Functional Requirement

3.3.1.1 Authentication User

In this module, user authentication will be checked using the username and password given by user. Avoiding any user simply used the system without login.

Input: User need to fill the username and password field and click on login button.

Output: Displaying the home page and user able to enroll the subjects.

Pseudo code:

1. Start
2. Fill in the field (username and password)
3. Click on login button
4. Login check (error will displayed if any error)
5. End

3.3.1.2 Manage Database

Searching the database wanted by user to schedule the time table.

Input: Fill the matric number field and click on button search.

Output: Timetable that been created for that matric number displayed.

Pseudo code:

1. Start
2. Fill in the matric number
3. Click on search button
4. Searching on database
5. Displaying timetable
6. End

3.3.1.3 Print

After students select the subjects, time table will be generated and displayed. Student may view the timetable again using search module. Whenever the timetable displayed, it can be printed out.

Input: Click on button print

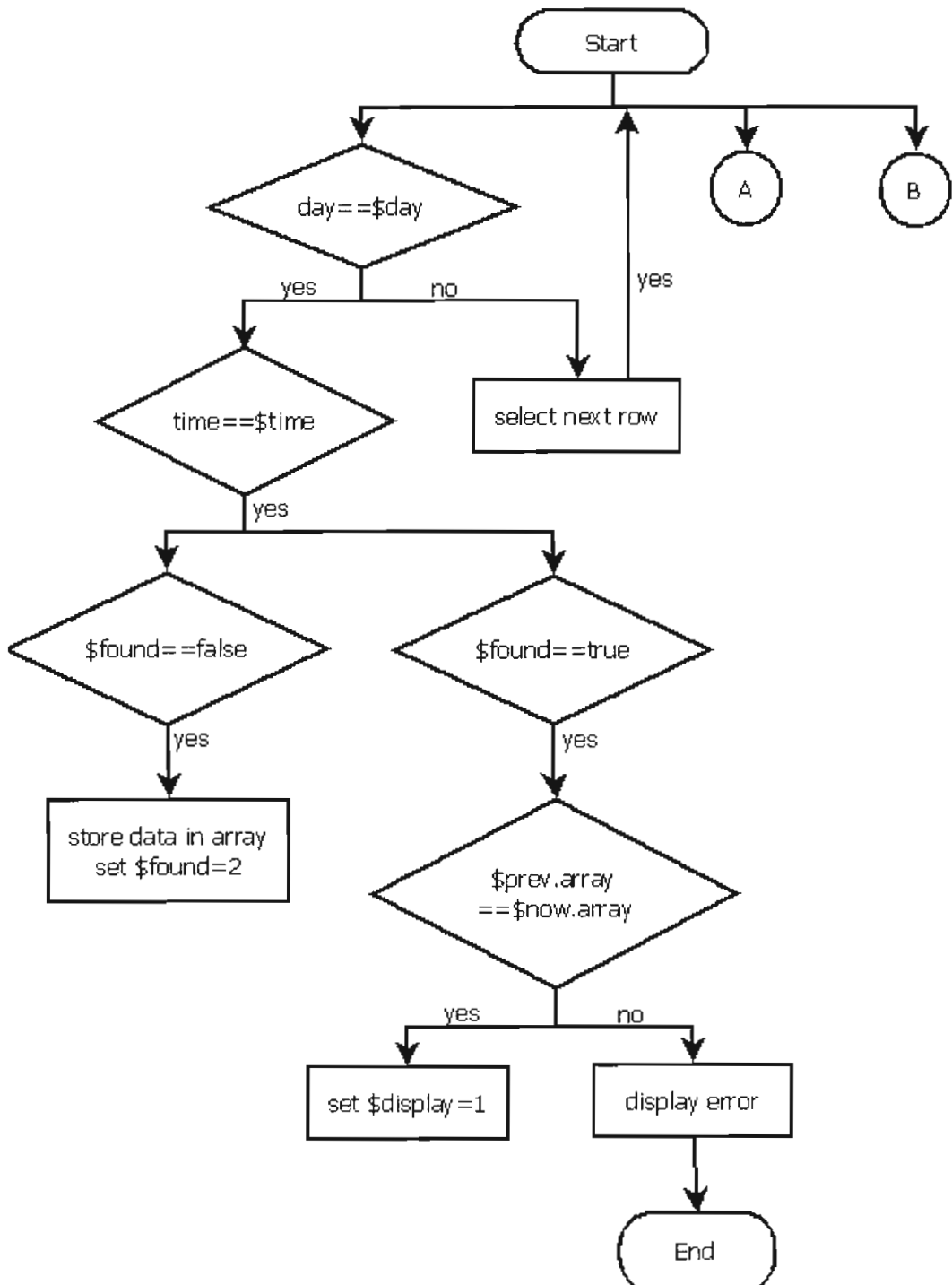
Output: Hard copy of the time table is printed out.

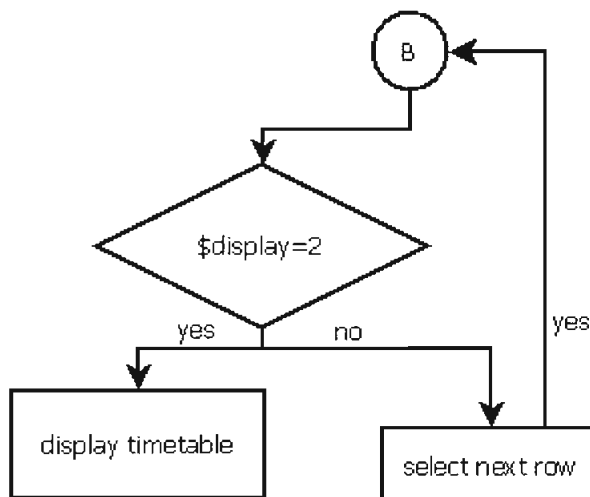
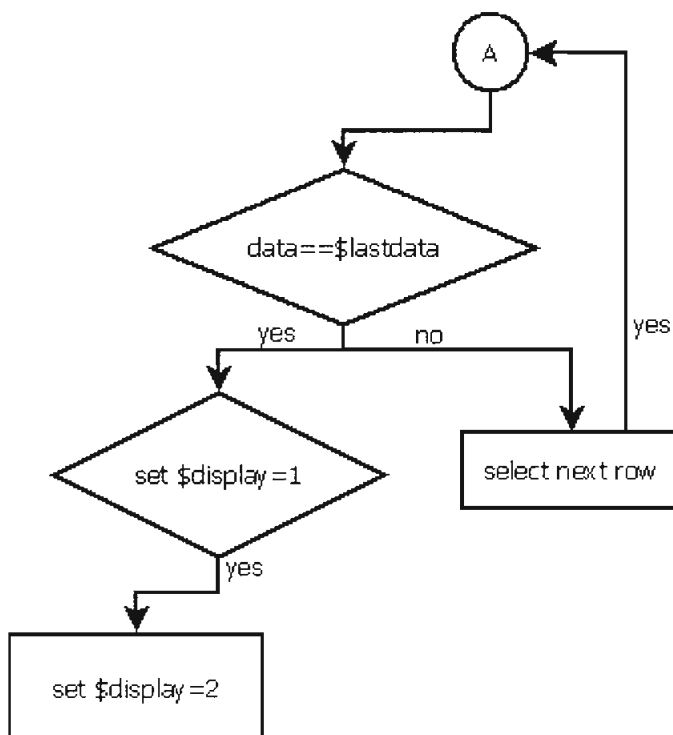
Pseudo code:

1. Start
2. Click on print button
3. Printing the time table
4. End

3.3.2 Non-functional Requirement

Heuristic scheduling technique is used to generate the time table. Following are the flow of generating the timetable for direct entry students.





3.4 Conclusion

Requirements are needed in having a good system since this chapter explaining the requirements needed for the project. In additional, a better understanding of project can help developer build the system easily. The next chapter is design. In that chapter, detailed explanations about the project interface, database and the system flow stated.

CHAPTER IV

DESIGN

4.1 Introduction

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. Based on the requirements and detailed analysis of new system, the system can be designed easily. There are two levels that needed to be considering in designing the system which are high level design and low level design.

4.2 High-Level Design

4.2.1 User Interface Design

Following are example of running system with user as admin:



Figure 4.1: Login page.

User authentication needed to let only user with access right to use the system. If the users cannot login, they can view only the contact page which they can ask anything to admin.

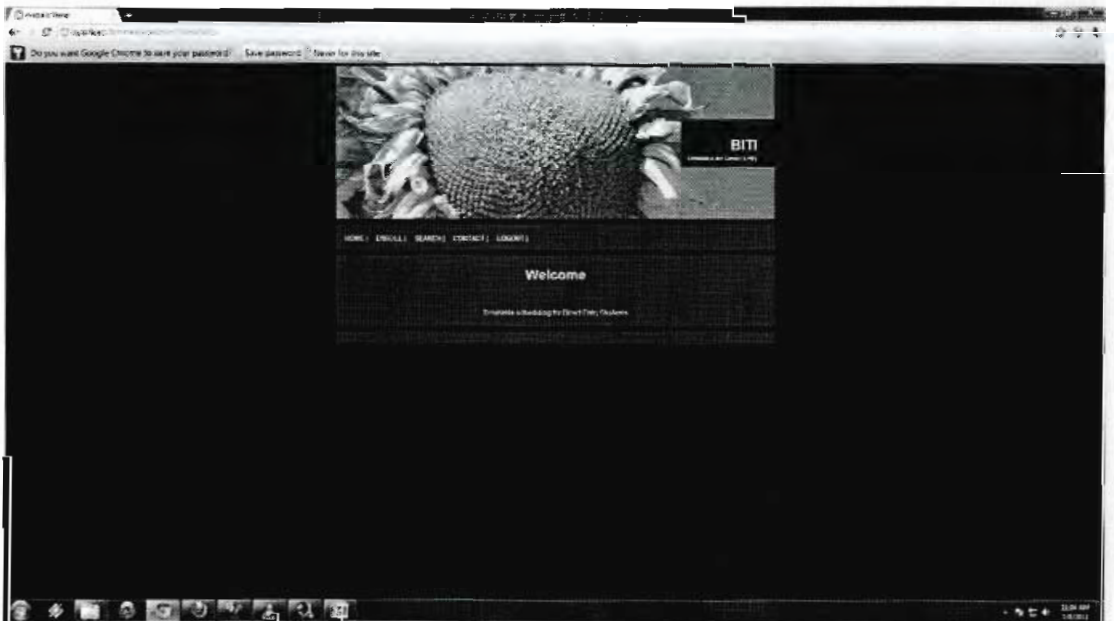


Figure 4.2: Home page.

This page only can be viewed when user is successfully login.



Figure 4.3: Contact page.

This page is for user to sent message to the administrator of the system. This page can be viewed by user which is not login and successfully login.

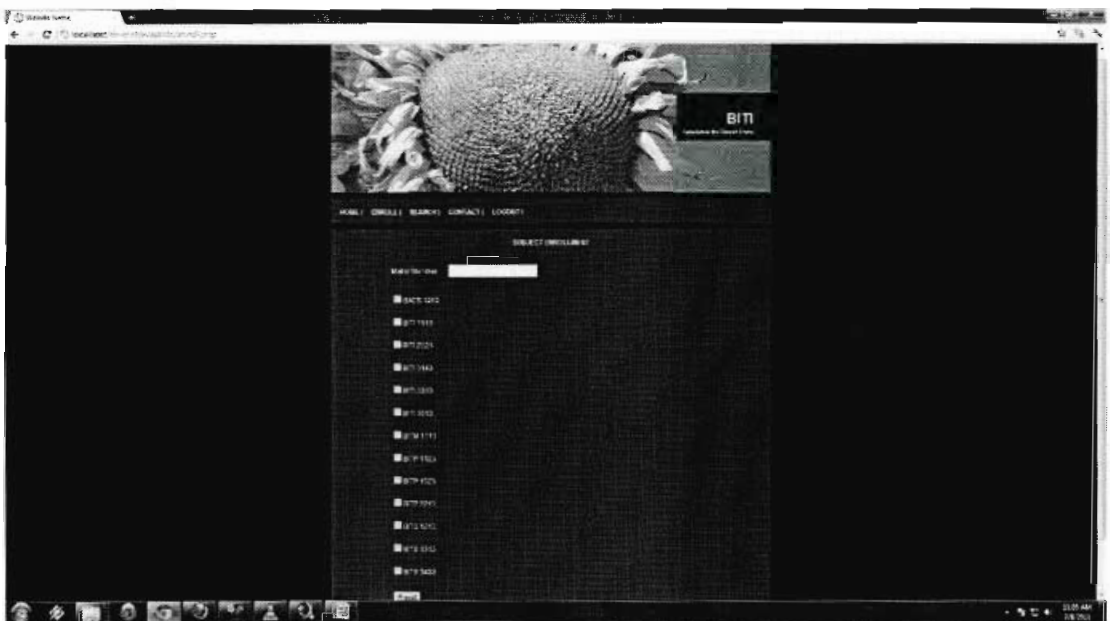


Figure 4.4: Enroll page.

Students select the subjects they want to enroll on this page. Then, timetable will be generated according to subjects selected.

TIME	MUNDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
08:30-09:30	BITP 1102 LIBRA				BITP 1103 LIBRA
09:30-10:30	BITP 1103 LIBRA	BIT 1112 LIBRA		BITP 1103 LIBRA	BITP 1103 LIBRA
10:30-11:30		BIT 1112 LIBRA		BITP 1103 LIBRA	BITP 1103 LIBRA
11:30-12:00	BITP 1103 LIBRA	BIT 1112 LIBRA	BITP 1103 LIBRA	BITP 1103 LIBRA	BITP 1103 LIBRA
12:00-12:30	BITP 1103 LIBRA	BIT 1112 LIBRA	BITP 1103 LIBRA	BITP 1103 LIBRA	
12:30-01:00					
01:00-01:30					
01:30-02:00					
02:00-02:30					
02:30-03:00					BITP 1103 LIBRA
03:00-03:30					BITP 1103 LIBRA
03:30-04:00					
04:00-04:30					

Figure 4.5: Timetable display page.

Timetable will be displayed with subjects and the classes. If there are any clash subjects, it will be displayed in the table. Table which successfully scheduled without having any constraints will be saved to database.

Figure 4.6: Search Page.

Timetable will be searched based on matric number. The timetable can only be displayed if student had created timetable before.

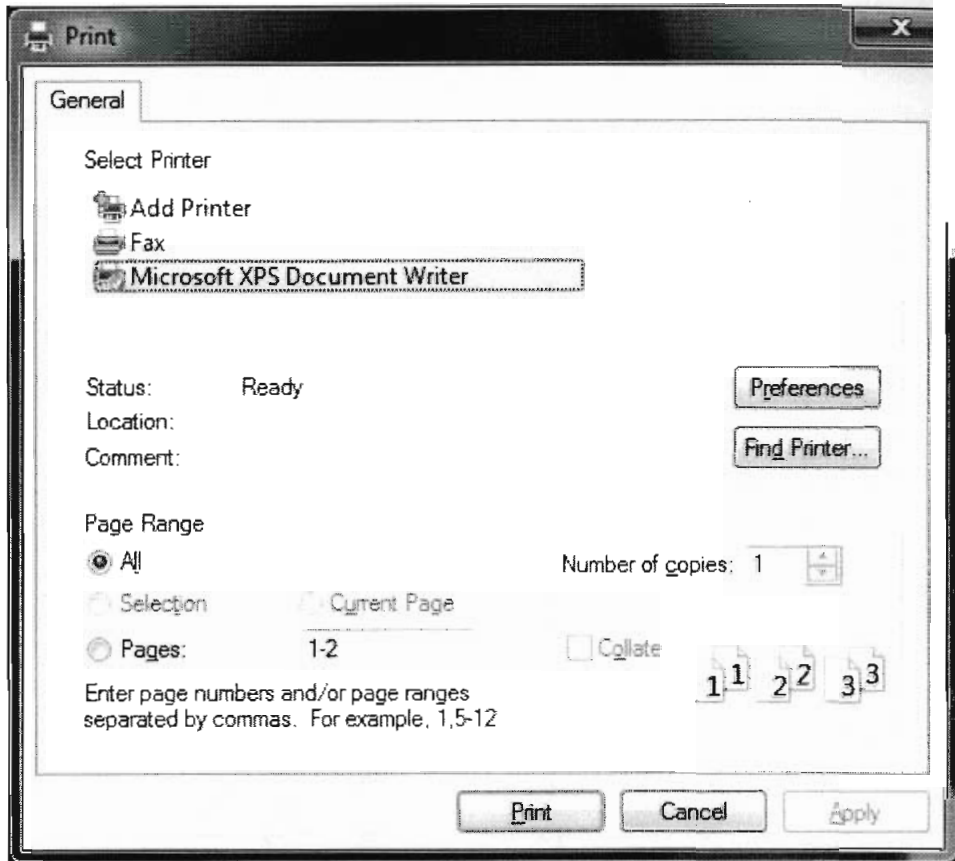


Figure 4.7: Print popup window.

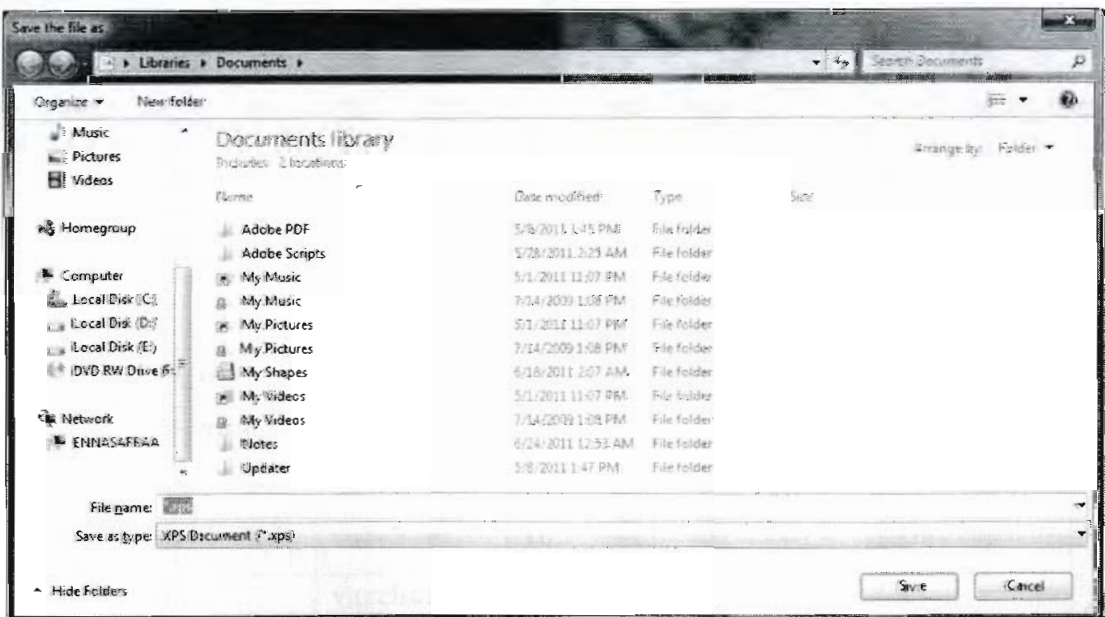


Figure 4.8: Save popup window.

If the printer cannot be read, the time table will be saved as .xps file.

4.2.3 Database Design

Figure 4.9: Entity Relationship Model Diagram (ERD)

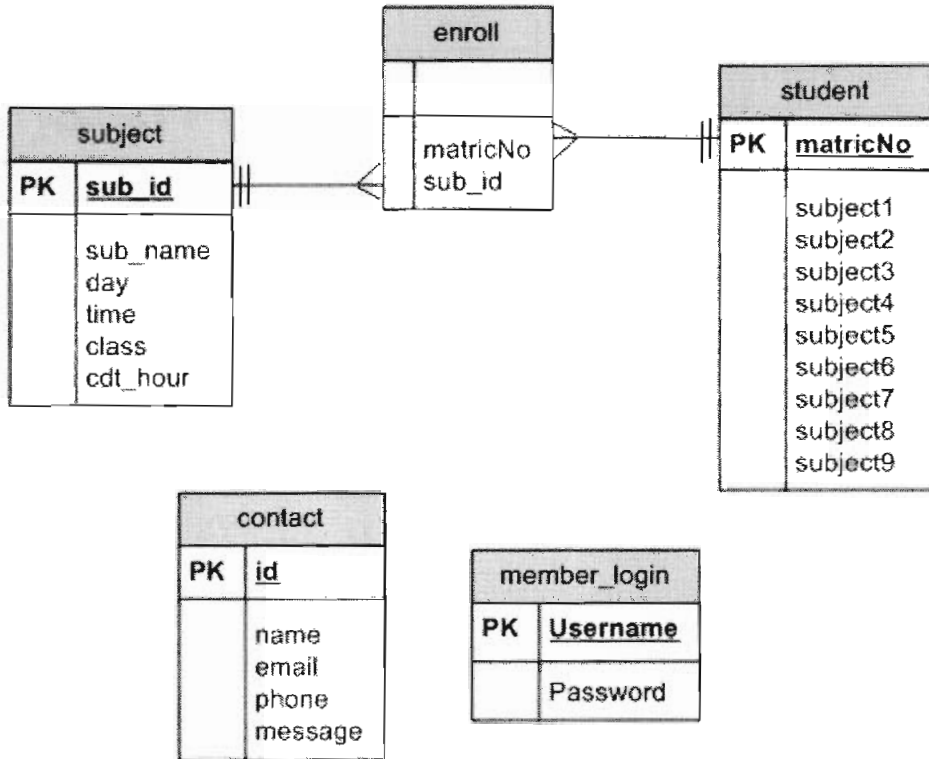


Table 4.1: Database (table name: subject)

No	Name	Data Type	Length	Primary Key / Foreign Key	Descriptions
1	sub_id	varchar	10	PK	Subject ID
2	sub_name	varchar	10		Subject name
3	day	varchar	10		Day
4	time	varchar	4		Time
5	class	varchar	10		Class
6	cdt_hour	int	1		Credit hour

Table 4.2: Database (table name: member_login)

No	Name	Data Type	Length	Primary Key / Foreign Key	Descriptions
1	Username	varchar	50	PK	Username
2	Password	varchar	50		Password

Table 4.3: Database (table name: student)

No	Name	Data Type	Length	Primary Key / Foreign Key	Descriptions
1	matricNo	varchar	10	PK	Student matric number
2	Subject1	varchar	10		Subject enrolled
3	Subject2	varchar	10		Subject enrolled
4	Subject3	varchar	10		Subject enrolled
5	Subject4	varchar	10		Subject enrolled
6	Subject5	varchar	10		Subject enrolled
7	Subject6	varchar	10		Subject enrolled
8	Subject7	varchar	10		Subject enrolled
9	Subject8	varchar	10		Subject enrolled
10	Subject9	varchar	10		Subject enrolled

Table 4.4: Database (table name: contact)

No	Name	Data Type	Length	Primary Key / Foreign Key	Descriptions
1	id	int	10	PK	Subject ID
2	name	varchar	50		Sender name
3	Email	varchar	20		Sender email address
4	phone	varchar	12		Sender phone number
5	message	varchar	400		Message received

Listed before are the table of type data with its details and description. Following are the example database which been saved in its format according to what user had select.

Table 4.5: Data saved (table name: member_login)

Username	Password
student	student
admin	admin

Table 4.6: Data saved (table name: contact)

id	name	email	phone	message
1	azza shazwani	azza128@gmail.com	0136788221	hello.
2	test	test@test.com	0171234567	testing

Table 4.7: Data saved (table name: subject)

sub_id	sub_name	day	time	class	cdt_hour
1	BITP 1323	Monday	08:00	Lecture	2
2	BITP 1323	Wednesday	11:00	Lab	1
3	BITI 1113	Tuesday	09:00	Lecture	2
4	BITI 1113	Tuesday	11:00	Lab	1
5	BITS 1213	Tuesday	14:00	Lecture	2
6	BITS 1213	Friday	10:00	Lab	1
7	BACS 1263	Thursday	09:00	Lecture	2
8	BACS 1263	Thursday	11:00	Lab	1
9	BITP 1123	Friday	08:00	Lecture	2
10	BITP 1123	Friday	15:00	Lab	1
11	BITS 1313	Monday	08:00	Lecture	2
12	BITS 1313	Wednesday	08:00	Lab	1
13	BITM 1113	Monday	11:00	Lecture	2
14	BITM 1113	Friday	10:00	Lab	1
15	BITI 2223	Tuesday	09:00	Lecture	2
16	BITP 2213	Tuesday	16:00	Lecture	2
17	BITP 2213	Thursday	14:00	Lab	1
18	BITI 2223	Thursday	09:00	Lab	1
19	BITI 3143	Monday	08:00	Lecture	2
20	BITI 3143	Friday	10:00	Lab	1
21	BITS 3423	Tuesday	09:00	Lecture	2
22	BITS 3423	Thursday	11:00	Lab	1
23	BITI 3313	Tuesday	11:00	Lecture	2
24	BITI 3313	Tuesday	14:00	Lab	1
25	BITI 3513	Thursday	09:00	Lecture	2
26	BITI 3513	Thursday	14:00	Lab	1

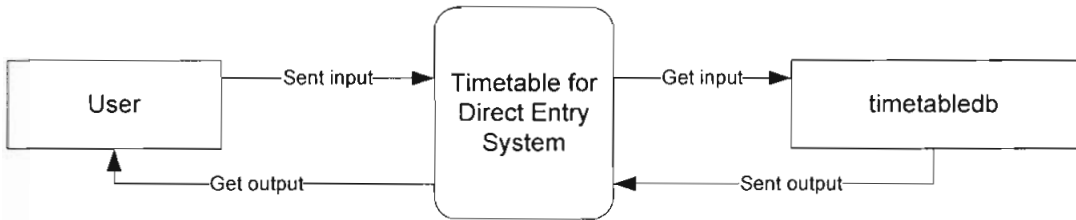
Table 4.8: Data saved (table name: student)

matricNo	Subject1	Subject2	Subject3	Subject4	Subject5	Subject6	Subject7	Subject8	Subject9
B031010196	BITS 1313	BITI 2223	BITM 1113	BITP 2213	BITI 2213				
B030810091	BITI 3143	BITS 2423	BITI 3513	BITS 3423	BITI 3313				

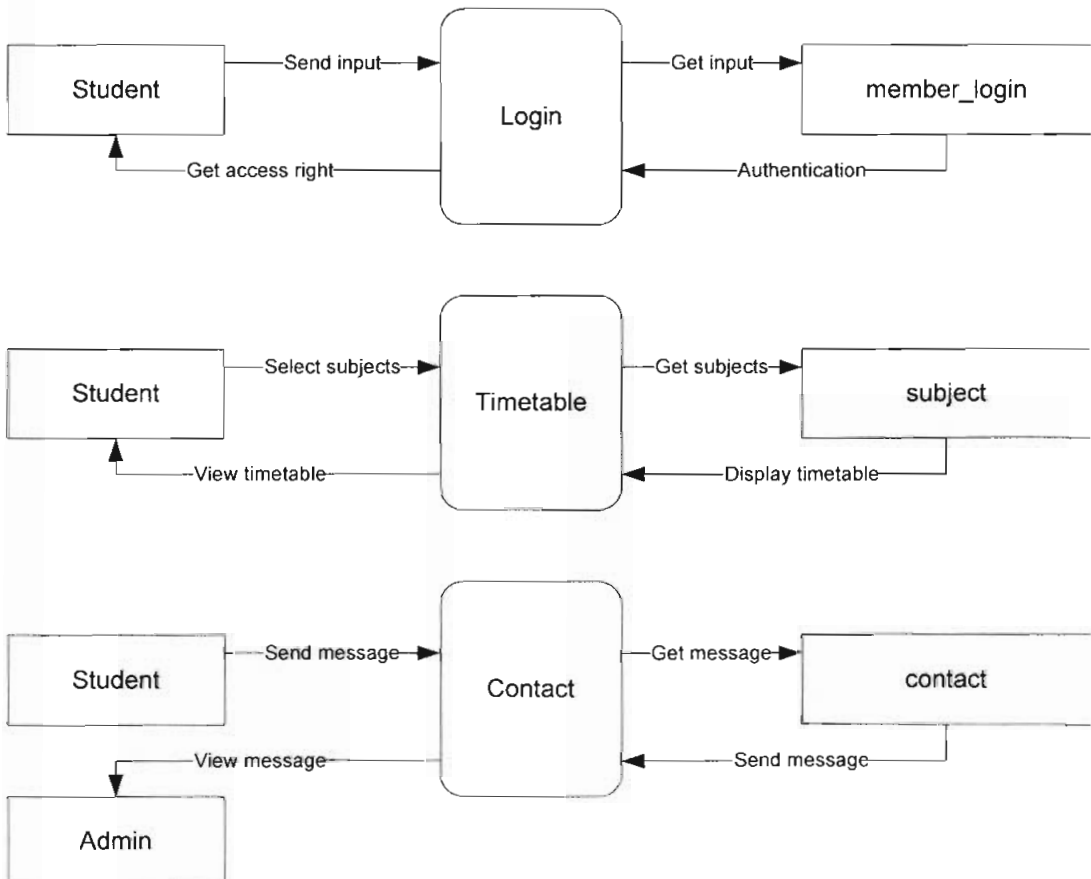
4.3 Detailed-Level Design

4.3.1 Software Design

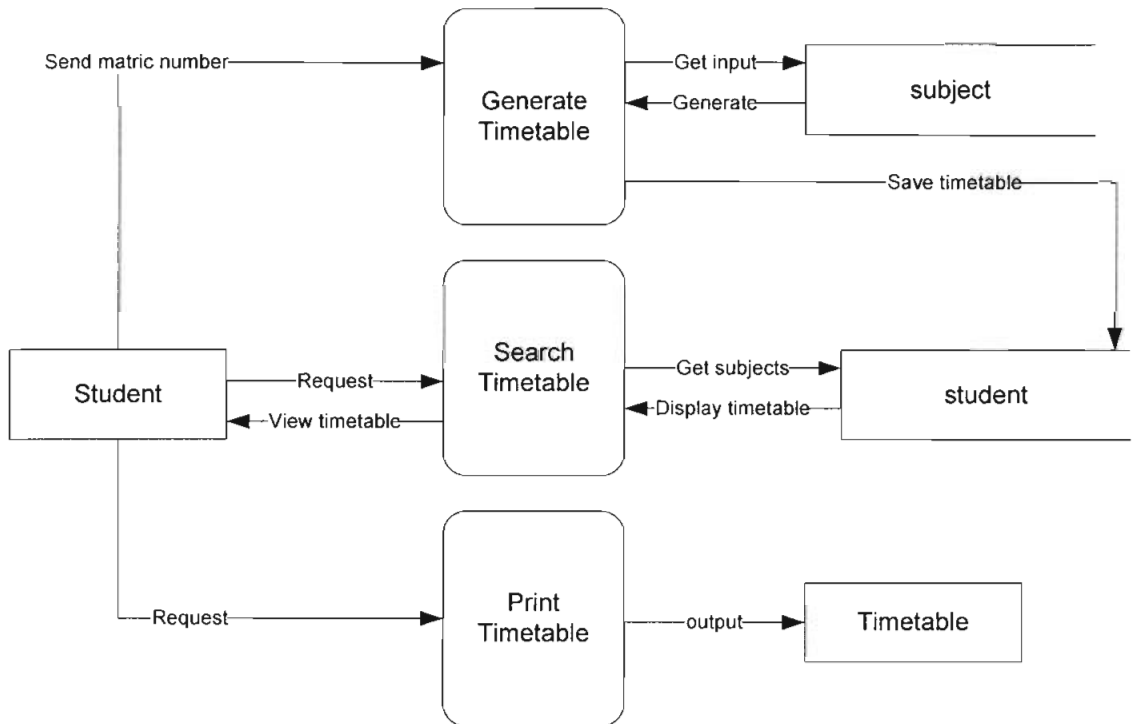
4.3.1.1 Data Flow Diagram Level 0



4.3.1.2 Data Flow Diagram Level 1



4.3.1.2 Data Flow Diagram Level 2



4.4 Conclusion

This chapter had explained in detail the design for the system including the database design, interface design and the system itself. The flow of database and how the system runs also had been explained on getting the main idea of the project. Next chapter will explain about the implementation of the project which is about the system and how to implement it.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

Implementation is the carrying out, execution, or practice of a plan, a method, or any design for doing something. As such, implementation is the action that must follow any preliminary thinking in order for something to actually happen. In an information technology context, implementation encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, running, testing, and making necessary changes. The word deployment is sometimes used to mean the same thing.

5.2 Software Development Environment Setup

In this project, there is no hardware development setup needed but the software development environment setup is needed. Adobe Dreamweaver is used to create the interfaces and implement the coding while Wamp server is used to enable the PHP coding which been applied to run the system. Other than that, web browser which is Internet Explorer is used to view the interface with the system functionality.

5.3 Software Configuration Management

Software configuration can be divided to two parts which are software configuration environment setup and version control procedure.

5.3.1 Configuration Environment Setup

To start the implementation, we need to install the Wamp server so that we can used to compile and check the error on PHP coding. After done the installation, we need to start the server and put it on online. Database system is available in the server so we may create the data tables in the server or we also may import the data tables from resource. The installation of all the software used will not be discussed in this chapter.

5.3.2 Version Control Procedure

A version control system is a combination of technologies and practices for tracking and controlling changes to a project's files, in particular to source code, documentation, and web pages. These days, everyone will expect at least your project's source code to be under version control, and probably will not take the project seriously if it doesn't use version control with at least minimal competence.

The reason version control is so universal is that it helps with virtually every aspect of running a project such as inter-developer communications, release management, bug management, code stability and experimental development efforts, and attribution and authorization of changes by particular developers. The version control system provides a central coordinating force among all of these areas. It is a communications mechanism where a change is the basic unit of information.

Following is table of all of the version control and the description created that corresponding to all modules that need to be developed in the system in this project.

Table 5.1: Version Control for Direct Entry (BITI) Timetable Scheduling System

Version	Description
Direct Entry (BITI) Timetable Scheduling System V1	Earliest version of the system without any function. Only the interface is available to be viewed.
Direct Entry (BITI) Timetable Scheduling System V1.1	Updated version than V1 which come up with login module and load the database.
Direct Entry (BITI) Timetable Scheduling System V1.2	This version contains an improvement from the previous versions. The improvements including the first phase of heuristic scheduling approach which is order-based.
Direct Entry (BITI) Timetable Scheduling System V1.3	Updated version from V1.2 which contain additional module for user to select subjects and also including the second phase of heuristic scheduling approach which is resource-based.
Direct Entry (BITI) Timetable Scheduling System V1.4	This version is updated version from V1.3 which contain additional module for user to search existing data and also including the third phase of heuristic scheduling approach which is operation-based.
Direct Entry (BITI) Timetable Scheduling System V1.5	This version is updated version from V1.3 which contain full phase of heuristic scheduling approach. It also contains several additional modules such as print and contact.

5.4 Implementation Status

Implementation status is the explanation of the project status in detailed. The description of the system development status will be described by type of module with the description about the module and the duration it take to finish the module.

Table 5.2: Implementation Status for Direct Entry (BITI) Timetable Scheduling System

Module	Description	Duration
Interface design	The interfaces need to be designed according to the system specification with all the function including in the system.	3 days
Subject Selection	Users must select the subjects they want to be scheduled.	2 day
Credit Hour Calculation	After selecting the subjects, system will calculate the total of credit hour availability.	2 day
Timetable scheduling	In this module a lot of things need to be considered to create the best timetable according to the constraints. Each phases of scheduling need to be done carefully to avoid having a result of an irrational timetable.	32 days

Printing	This module allow user to print the timetable created from the system.	2 days
User manual	This will help the users to go through each step on how to use this system efficiently and smoothly.	1 days

5.5 Conclusion

In this chapter, we had discussed the implementation phase in developing a system including the software and hardware environment setup and software and hardware configuration. This phase is basically helping me out to discover the best way and to develop the system with more efficiently and smoothly. Next chapter is the testing phase which will discuss on testing steps and the result analysis.

CHAPTER VI

TESTING

6.1 Introductions

Testing is activities involve multiple cycles that are largely executed in a serial fashion. Therefore, the testing process can occur over a large period of elapsed time. The focus of testing varies to some degree depending on the solution being delivered and the activities presented in this section are general approach for an information management project.

6.2 Test Plan

The testing plan is divided into three sections which are test organization, test environment and test schedule.

6.2.1 Test Organization

The testing will be done by direct entry students of BITI in Universiti Teknikal Malaysia Melaka. The students can be from any year of education and semester as long as they are a BITI student.

6.2.2 Test Environment

The system testing been carried out in several places such as in campus or faculty, cafeteria, hostels and UTeM bus stop. Before the testing start, laptop and simple questionnaire are provided to help the tester evaluate the system. Questionnaire given to the tester can be view on Appendix B.

6.2.3 Test Schedule

System only been tested by direct entry student after all the modules are functional. Testing process also has been done frequently during the implementation process after each module in the system is completed. During that testing process, any errors or problems occurred will be fixed instantly to avoid the error affected other function.

6.3 Test Strategy

Testing strategy can be classified to several types such as bottom-up testing, top-down testing, black-box testing, and white box testing. Choosing the right testing strategy will help a developer to develop a good system with fewer errors or any other problems.

- **Black-box testing**

This approach is to test the system without knowing any function inside the system and only focusing on the testing of the functional requirement and specification. The internal inspection is not necessary to be done.

- **White-box testing**

This approach is to test the internal or system coding testing. It is opposite to the black-box testing which focus on the system function. This kind of testing is more suitable for system with security.

- **Bottom-up testing**

This approach is focusing on integrated testing process which started with lowest component level of the module and continues with the next level of the modules. This process will continuous until the highest level of component module is tested.

- **Top-down testing**

This is the opposite testing strategy with the bottom-up strategy. System will be tested starting with highest integrated component and continuous until the lowest level of the module had been tested.

In this project, bottom-up testing had been used during the implementation of the system while the testing strategy been used during the direct entry student test the system after all the modules are functional is black-box testing.

6.3.1 Classes of Tests

In the testing process, there are several classes of testing which are output correctness, functionality test, significant test, security test, stress test and including the code debugging.

6.4 Test Implementation

This is the testing had been done during the implementation to avoid having errors or any problems occur.

6.4.1 Test Description

Following is the description of testing done during the implementation.

Table 6.1: List of testing during implementation

Test	Error or Problem	Action
1	Login module is able to use but user can use the system without login. No security on login.	Fixing the login module by adding authentication validation.
2	Subject can be chosen with more or less credit hour than university requirement.	Adding calculation to calculate the total of credit hour.
3	Subject selected but timetable cannot be generated.	Fixing the coding on generating the timetable module
4	Timetable displayed cannot detect which subjects have class on the same day and time.	Fixing the coding on generating the timetable module
5	Timetable generated only display an empty table	Fixing the coding on generating the timetable module
6	Timetable displayed all subjects is clash with each other even if they are not.	Fixing the coding on generating the timetable module
7	Timetable displayed subjects which are clash only but not the other subjects.	Fixing the coding on generating the timetable module
8	Timetable displayed correctly but timetable cannot be detected to be print out.	Fixing the coding on generating the timetable module and print module.
9	Cannot view the timetable in print view.	Fixing the coding on print module

6.4.2 Test Data

Data that been used in testing during the implementation is the real database which been collected from the faculty. Same goes to the testing during the system been tested by direct entry. Database been collected is the existing timetable of year one, two and three of BITI student in current semester. System will generate the new timetable for direct entry student according to subjects selected and existing timetable.

6.5 Test Result and Analysis

The result is based on questionnaire been done by tester after testing the Direct Entry (BITI) Timetable Scheduling System and all of students selected to test this system are BITI students. There are five questions with question 1 to 4 will are answered by rating from 1 to 3 as 1 bad and 3 is good, while question 4 needed to be answered with yes or no and question 5 asks the tester to write comment about the system. For the example of questionnaire, see Appendix B.

Question 1: The interface is user friendly?

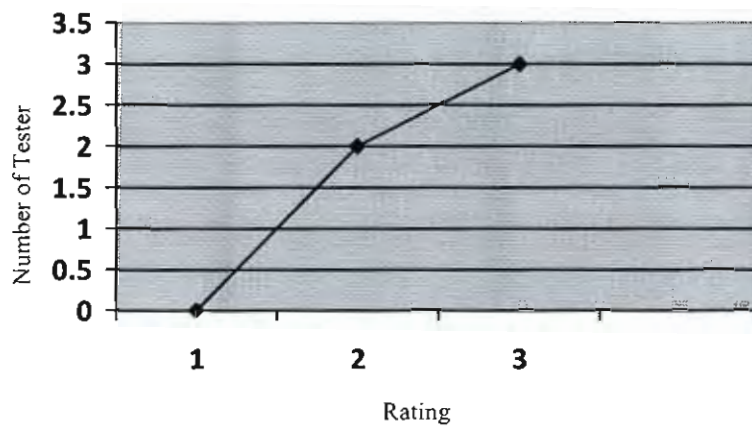


Figure 6.1: Question 1

Three people who tested the system agree that the system has interface which is user friendly, while other two people choose to rate it as 2.

Question 2: This system is helpful to schedule your time table?

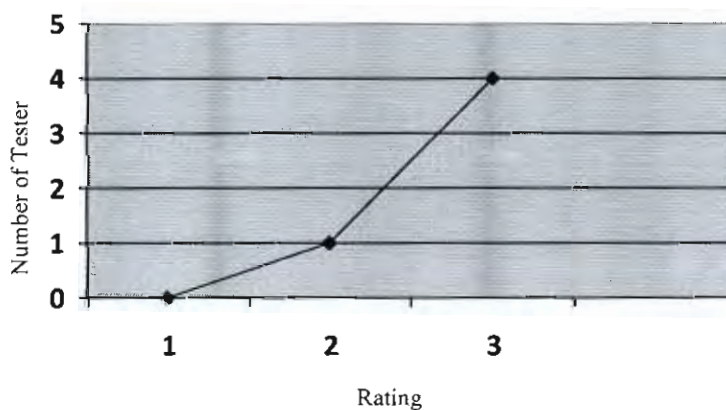


Figure 6.2: Question 2

For question two, four people agree that this system is helpful for them to schedule the timetable.

Question 3: You use less time to generate a timetable?

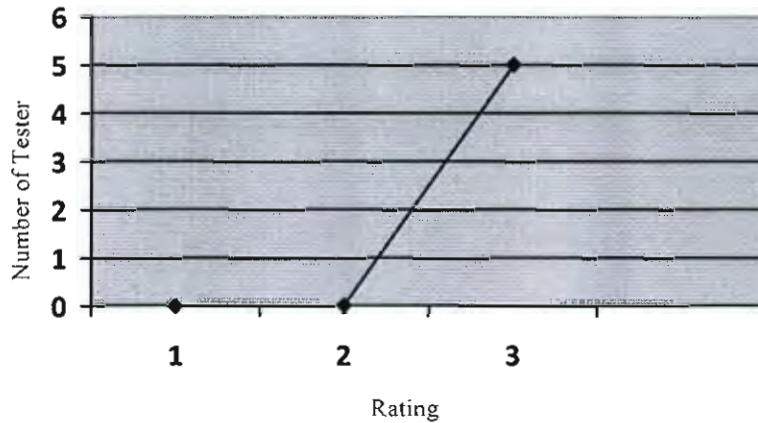


Figure 6.3: Question 3

All five people agree that this system help them use less time to schedule their timetable.

Question 4: Will you suggest this system to your friends?

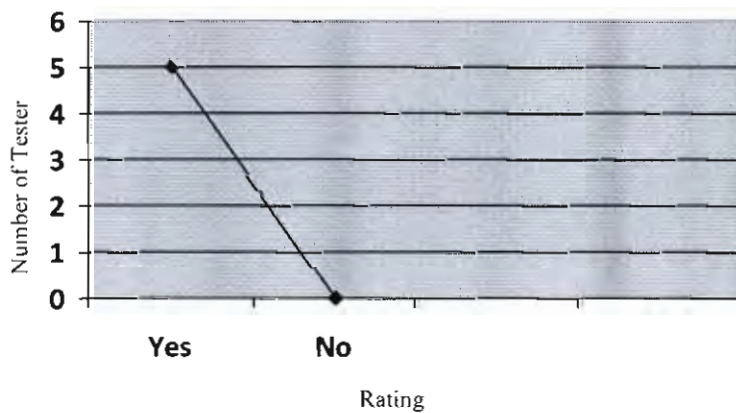


Figure 6.4: Question 4

For question number four, all five people voted YES to suggest this system to their friends.

Question 5: Comment

Four out of five commented that this system should allow them to join other class from other courses and not just BITI. Some of them also commented that they cannot enroll other subjects because the subjects are not listed because their subjects only offer for student with special case. Some of them also suggested the interface should be change.

6.6 Conclusion

In this chapter, the testing had been done during and after all the modules are functional is discuss in detailed including the test environment, test schedule, test strategy, test description, test data and test result with analysis. Next chapter discuss on the overall conclusion of the project.

CHAPTER VII

PROJECT CONCLUSION

7.1 Observation on Weakness and Strength

There are few weaknesses and strengths noticed after doing the observation on this project. Both weaknesses and strengths will be discussed in detail in this chapter.

7.1.1 Weaknesses

Few weaknesses had been noticed after running the test phase of development with the help of questionnaire provided during the testing process. The weaknesses noticed are students unable to join the class from other courses such as BITM, BITS, BITD or BITC which also offer the same subject if BITI class is unavailable for them. Other than that, students also cannot select the subject which is not listed because some student enroll certain subject which is not been offered on the current semester as they are in special case.

7.1.2 Strengths

After the testing phase and analysis the result, most of students agreed that this system is helpful for them to generate the timetable. In additional, they do not need to ask all of timetable for each class from their academic assistant or head of department or faculty. Other than that, most student also agree that this system help them to schedule the timetable in short time.

7.2 Proposition for Improvements

Further research on this problem of scheduling the timetable for direct entry student is needed since not many researchers do the research on this problem. Most of them do the research on scheduling new timetable based on class, time, day, place and lecturer. This system need to be improved on database collection. The database should be including all courses in FICT. Other than that, this system also can be implemented using other algorithm or programming language.

7.3 Contribution

This Direct Entry (BITI) Timetable Scheduling System obviously can be used by BITI students in UTeM. This system also can be used by other courses which have direct entry student. Other universities that have direct entry student also can be used this system because it is very helpful to scheduling the timetable in short time. The timetable which had been saved after it had been scheduled can be viewed again by searching the timetable based on student's matric number. Printable timetable created by this system help the student easily scheduled their timetable. User manual is attached as Appendix C.

7.4 Conclusion

As the conclusion, all of the objectives such as to provide a tool for student to schedule their timetable, use less time to schedule the timetable and avoid having more than one class at the same time is successfully fulfilled. Heuristic scheduling technique is suitable for this project and other improvement of this project will help student have a better tool to schedule their timetable.

REFERENCES

A. J. Hosios and J. M. Rousseau, (Aug., 1980). "A Heuristic Scheduling Algorithm." The Journal of the Operational Research Society. Vol. 31, No. 8 . pp. 749-753.

S. Jurgen, "Knowledge-based Systems in Scheduling." Department of Computer Science, University of Oldenburg.

T. G. Liu, (2009). "Class Timetable Scheduling System." Bachelor Degree Thesis. Universiti Teknikal Malaysia Melaka.

H. Abu Bakar, (2009) "E Dokumen Projek Diploma UTeM." Diploma Thesis. Universiti Teknikal Malaysia Melaka.

Jawatan Kuasa Projek Sarjana Muda dan Diploma, (2011). "Buku Panduan Pelaksanaan Projek Sarjana Muda (PSM)." Edisi Pertama. Desktop Publisher. 1-90.

BIBLIOGRAPHY

S. Jurgen, "Knowledge-based Systems in Scheduling." Department of Computer Science, University of Oldenburg.

Jawatan Kuasa Projek Sarjana Muda dan Diploma, (2011). "Buku Panduan Pelaksanaan Projek Sarjana Muda (PSM)." Edisi Pertama. Desktop Publisher. 1-90.

APPENDIX A: EXAMPLE CODING OF USING HEURISTIC SCHEDULING

```

        <?php
            $found = false;
            $display = 0;
            $sarr = array();
            $lastData = count($allData)-1;

            for($i=0;$i<count($allData);$i++){
                list($time,$ignore) = split(":",$allData[$i][3]);
                $day = $allData[$i][2];

                if($day == "Monday"){
                    if($time == "" || $time == ""){
                        if($found == false){
                            $sarr[0] = $allData[$i][0];
                            $sarr[1] = $allData[$i][1];
                            $sarr[2] = $allData[$i][2];
                            $sarr[3] = $allData[$i][3];
                            $found = true;
                        }
                        if($found == true){
                            if($sarr[0] == $allData[$i][0]){
                                $display = 1;
                            }
                        }
                    }
                    else{
                        echo "Subject ".$sarr[0]."  

                        clash with subject ".$allData[$i][0];
                        $slash = 1;
                        break;
                    }
                }
            }

            }
        }
        if($i == $lastData){
            if($display == 1){
                $display = 2;
            }
        }
        if($display == 2){
            echo $sarr[0]."<br />";
            echo $sarr[1]."<br />";
        }
    }
?></td>

<td align="center" valign="middle"><?php
    $found = false;
    $display = 0;

```

```

        $sarr = array();
        $slastData = count($sallData)-1;

        for($si=0;$si<count($sallData);$si++){
            list($stime,$signore) = split(":",$sallData[$si][3]);
            $sday = $sallData[$si][2];

            if($sday == "Tuesday"){
                if($stime == "" || $stime == ""){
                    if($sfound == false){
                        $sarr[0] = $sallData[$si][0];
                        $sarr[1] = $sallData[$si][1];
                        $sarr[2] = $sallData[$si][2];
                        $sarr[3] = $sallData[$si][3];
                        $sfound = true;
                    }
                    if($sfound == true){
                        if($sarr[0] == $sallData[$si][0]){
                            $sdisplay = 1;
                        }

                        else{
                            echo "Subject ".$sarr[0]."  

                            clash with subject ".$sallData[$si][0];
                            $sclash = 1;
                            break;
                        }
                    }
                }
            }
        }

        if($si == $slastData){
            if($sdisplay == 1){
                $sdisplay = 2;
            }
            if($sdisplay == 2){
                echo $sarr[0]."<br />";
                echo $sarr[1]."<br />";
            }
        }
    ?></td>

<td align="center" valign="middle"><?php
        $sfound = false;
        $sdisplay = 0;
        $sarr = array();
        $slastData = count($sallData)-1;

```

```

for($i=0;$i<count($allData);$i++){
    list($time,$ignore) = split(":",$allData[$i][3]);
    $day = $allData[$i][2];

    if($day == "Wednesday"){
        if($time == "" || $time == ""){
            if($found == false){
                $arr[0] = $allData[$i][0];
                $arr[1] = $allData[$i][1];
                $arr[2] = $allData[$i][2];
                $arr[3] = $allData[$i][3];
                $found = true;
            }
            if($found == true){
                if($arr[0] == $allData[$i][0]){
                    $display = 1;
                }

                else{
                    echo "Subject ".$arr[0]."  

                    clash with subject ".$allData[$i][0];
                    $slash = 1;
                    break;
                }
            }
        }
    }

    }

    if($i == $lastData){
        if($display == 1){
            $display = 2;
        }
        if($display == 2){
            echo $arr[0]."<br />";
            echo $arr[1]."<br />";
        }
    }
}
?></td>

<td align="center" valign="middle"><?php
    $found = false;
    $display = 0;
    $arr = array();
    $lastData = count($allData)-1;

for($i=0;$i<count($allData);$i++){
    list($time,$ignore) = split(":",$allData[$i][3]);
    $day = $allData[$i][2];

```

```

if($day == "Thursday"){
    if($time == "" || $time == ""){
        if($found == false){
            $arr[0] = $allData[$i][0];
            $arr[1] = $allData[$i][1];
            $arr[2] = $allData[$i][2];
            $arr[3] = $allData[$i][3];
            $found = true;
        }
        if($found == true){
            if($arr[0] == $allData[$i][0]){
                $display = 1;
            }

            else{
                echo "Subject ".$arr[0]."  

                clash with subject ".$allData[$i][0];
                $slash = 1;
                break;
            }
        }
    }
}

}

if($i == $lastData){
    if($display == 1){
        $display = 2;
    }
    if($display == 2){
        echo $arr[0]."<br />";
        echo $arr[1]."<br />";
    }
}
?></td>

<td align="center" valign="middle"><?php
    $found = false;
    $display = 0;
    $arr = array();
    $lastData = count($allData)-1;

    for($i=0;$i<count($allData);$i++){
        list($time,$ignore) = split(":",$allData[$i][3]);
        $day = $allData[$i][2];

        if($day == "Friday"){
            if($time == "" || $time == ""){

```

```

        if($found == false){
            $arr[0] = $allData[$i][0];
            $arr[1] = $allData[$i][1];
            $arr[2] = $allData[$i][2];
            $arr[3] = $allData[$i][3];
            $found = true;
        }
        if($found == true){
            if($arr[0] == $allData[$i][0]){
                $display = 1;
            }

            else{
                echo "Subject ".$arr[0]."  

                clash with subject ".$allData[$i][0];
                $slash = 1;
                break;
            }
        }
    }
}
}
if($i == $lastData){
    if($display == 1){
        $display = 2;
    }
}
if($display == 2){
    echo $arr[0]."<br />";
    echo $arr[1]."<br />";
}
}
?>

```

APPENDIX B: QUESTIONNAIRE

Direct Entry (BITI) Timetable Scheduling

Name: Azza Syazwani
 Semester: 4

Course: BITI
 Faculty: FTMK

Please rate from 1 to 3. 1 is bad, 3 is good.

1. The interface is user friendly? 3
2. This system is helpful to schedule your time table? 2
3. You use less time to generate a timetable? 3
4. Will you suggest this system to your friends?(yes or no) yes
5. Comment (Fill the box with your comment)

-unable to enroll subjects from other classes
 -change sunflower (interface) to computer image.

Direct Entry (BITI) Timetable Scheduling

Name: Ahmad KhalisCourse: BITISemester: 4Faculty: FTMK

Please rate from 1 to 3. 1 is bad, 3 is good.

1. The interface is user friendly? 22. This system is helpful to schedule your time table? 33. You use less time to generate a timetable? 34. Will you suggest this system to your friends?(yes or no) yes

5. Comment (Fill the box with your comment)

good

Direct Entry (BITI) Timetable Scheduling

Name: Shakir ZamriCourse: BITISemester: 4Faculty: FTMK

Please rate from 1 to 3. 1 is bad, 3 is good.

1. The interface is user friendly? 22. This system is helpful to schedule your time table? 33. You use less time to generate a timetable? 34. Will you suggest this system to your friends?(yes or no) yes

5. Comment (Fill the box with your comment)

-cannot enroll subjects from other courses

Direct Entry (BITI) Timetable Scheduling

Name: Siti RohaizaCourse: BITISemester: 2Faculty: FTMK

Please rate from 1 to 3. 1 is bad, 3 is good.

1. The interface is user friendly? 3
2. This system is helpful to schedule your time table? 3
3. You use less time to generate a timetable? 3
4. Will you suggest this system to your friends?(yes or no) yes
5. Comment (Fill the box with your comment)

- subject not listed (special case)

Direct Entry (BITI) Timetable Scheduling

Name: ZulfahimiCourse: BITISemester: 4Faculty: FTMK

Please rate from 1 to 3. 1 is bad, 3 is good.

1. The interface is user friendly? 32. This system is helpful to schedule your time table? 33. You use less time to generate a timetable? 34. Will you suggest this system to your friends?(yes or no) yes

5. Comment (Fill the box with your comment)

APPENDIX C: USER MANUAL

This is the guide to use the system.

** For User to create new timetable*

Step 1: Login as student.

Username: student

Password: student

Step 2: Choose tab 'Enrollment'

Step 3: Select subjects

Step 4: Click on 'DONE' button

Step 5: Timetable generated

Step 6: Click on 'PRINT' to print the timetable

Step 7: Logout

** For user to view timetable which had been created before*

Step 1: Login as student.

Username: student

Password: student

Step 2: Choose tab 'Search'

Step 3: Fill the matric number

Step 4: Click on 'SEARCH' button

Step 5: Timetable generated

Step 6: Click on 'PRINT' to print the timetable

Step 7: Logout