Faculty Of Computer Science
And Information Technology
University Of Malaya

FSKTM COURSE SCHEDULING SYSTEM

ONG SOH PUN
WEK 990063

Supervisor: Puan Norizan Mohd Yasin

Moderator: Puan Fariza Hanum

ABSTRACT

Timetabling the courses offered at the Faculty of Computer Science and Information Technology in University of Malaya (FSKTM, UM) is a hard problem faced continually in this faculty. The timetabling problem consists in fixing a sequence of meetings between lecturers and students in a prefixed period of time, often to satisfy a set of constraints. Currently, FSKTM is using the manual system to generate timetable. The manual solution of courses timetabling usually requires days of work and sometimes the solution may be unsatisfactory to some respect.

Subsequently, this project paper is conducted to develop an automated FSKTM Course Scheduling System (FCSS) under Web-based application. The purpose of FCSS is to reduce the time and effort needs for generating a timetable in FSKTM. FCSS could generate timetable automatically and publish the timetable on the Web.

In this system, an efficient scheduling engine will solve conflicting issues in timetabling problem. The engine will generate timetable automatically after checking the crashes between lecturers, students, rooms and other hard constraints. All the information needed such as information about rooms, courses and lecturers are stored in the database. The system will interact with the end user through the interface subsystem.

Basically, there are three modules in FCSS, which are administration module, course scheduling module and user module. Administrators could access administration module to manage and keep track of the database's information. Timetable coordinator could generate timetable using the scheduling engine in the course scheduling module. In user module, students and lecturers could view and print their timetable through the Internet.

ACKNOWLEDGEMENT

In the process of completing this project paper, many people have been very kind in lending helping hands, giving invaluable advices and encouragement that contributed to the success in the production of this system.

First of all, I would like to express my greatest gratitude and acknowledgement to my supervisor, Puan Norizan Mohd Yasin, for being so dedicated and patient in giving me advices and consultation at any time and providing me with the detail of how to complete the project. Her encouragement and kindness in helping me throughout the project is deeply appreciated.

Secondly, special thanks and appreciation are conveyed to Puan Fariza Hanum, for her kindness of being my moderator and giving me valuable advices, guidance and motivation to complete my project with success.

Thirdly, many thanks to all my family members as they have given me much needed support and encouragement on my studies.

Last but not least, not forgotten also to thank all of my fellow and helpful friends. Every time when I encountered some problems, they are very willing to assist and advise me. Thank for their full supports, advises and helps.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

University Courses timetabling is a problem to assigning times and places to many separate lectures and tutorials in order to satisfy several constraints concerning capacities and locations of available rooms, free time needs and other such consideration for lecturers, and relationships between particular courses. The most prominent constraint is that there should be no clashes where two lecturers having common students should not be scheduled simultaneously or a lecturer should not be scheduled simultaneously.

In general, there are a few algorithms used to solve timetabling problem such as Graph Coloring Algorithms, Tabu Search, Simulated Annealing, Genetic Algorithms and Evolutionary Algorithms. Each algorithm has its own way to solve the constraint in course timetabling. Because there are a lot of versions of timetabling constraints, differing from one university to the next, they used different algorithms to overcome timetabling problem.

At the start of each new semester in University of Malaya, Faculty of Science Computer and Information Technology must undertake the laborious, time-consuming process of slotting students, lecturers and courses into available rooms. Currently, the timetable was done manually where the timetable coordinator assigning a particular course to a location and time period in order to satisfy several constraints in the faculty. The manual solution of courses timetabling usually requires days of work and sometimes the solution may be unsatisfactory to some respect.

Due to the increasing student numbers, new courses introduced and shortage of lecture halls and laboratories, it is very difficult for the timetable coordinator to schedule a timetable manually. Therefore, we need an automated scheduler to overcome these problems. Subsequently, an automated timetable scheduler named **FSKTM Course Scheduling System (FCSS)** will be proposed and developed in this project paper.

In this project, I plan to develop a more capable and effective timetable scheduler that satisfies several hard and soft timetabling constraints in the faculty. The system can generate courses schedule automatically. Nevertheless, I am developing a web-based

scheduling system to facilitate students' and lecturers' work. The implementation of the system in the web-based will enable online access by the students and the lecturers through the Internet. Student could view and print schedule consists of courses selected and lecturers could view their teaching times online.

The system is divided into three subsystems, which are database subsystem, scheduling subsystem, and user interface subsystem. The database subsystem is used to store the information about courses, lecturers, rooms and other related information. The scheduling subsystem includes the algorithms used to produce the schedules. The user interface subsystem allows users to interact with the system.

Basically, there are three modules in the system: Administration module, Course Scheduling module and User module. Administration module allows the administrators to manage and update information stored in database. Course Scheduling module allows the timetable coordinators to generate timetable for the particular semester. Lastly, User module allows students and lecturers to view and print their timetable.

1.2 PROBLEM DEFINITION

Current manual course scheduling system has some problems as stated below:

1. *Unable to fulfill stakeholders' requirement and constraints*

   The main problem of the current course scheduling system is that it could not fulfill most of the stakeholders' requirements from time to time. Presently the particular courses, rooms, time periods and lecturers are inputted into the schedule manually. Usually the schedule has to be regenerated to recover some errors and problems discovered in the previous schedule. It is very time consuming for coordinator to generate a schedule that could meet the lecturers' and students' requirements and at the same time satisfy constraints in the faculty. These constraints include free time needs, room capacities, other consideration for lecturers and relationships between particular courses. Due to the increase of number of students, new courses introduced and shortage of lecture halls and laboratories, it raises the difficulty for coordinator to generate timetable manually.

2. *Resources not fully utilized*

   Faculty resources are not fully utilized due to the lack of information in resources planning. Number of students (enrollment) and room capacity are not at the same level. Some classes are too crowed, while some classes are too spaced. For example, class that consists of 200 students is undertaken in Auditorium room while DK1 is assigned to class that consists of 50 students only. Furthermore, certain course which need certain equipment such as multimedia equipment in certain room are not accessible to that room due to inefficient room assignment of current manual scheduling system.

3. *Students' class timing conflict*

   Students' class timing conflict is always a headache issue for timetabling coordinator. This issue is still unsolved although there have been many attempt from undergraduates to develop artificial intelligent computerized course scheduling system to replace the current manual system.

4. *Lecturers' daily workload not balanced*

   Timetable developed currently was unable to satisfy different time preferences among lecturer. Furthermore, the daily workload of the lecturer are not balanced or spread equally in a week period. Psychologically, this will lead to undesirable productivity and dispute among lecturer. For example, it would be impossible for the lecturer to have two or three continuous class in a day, as this will exhaust the lecturer. As a result, they might find it hard to give concentration in each class. Another example is that lecturer might want to reserve certain time for appointment, or doing their research.

As a conclusion, I will develop an automated course scheduling system and solve the problem as stated above.

1.3 TIMETABLE DEFINITION

There are two common ways to define a ***Timetable*** or ***Schedule***:

*" A **Timetable**, or **Schedule**, is a description of the movement and grouping of resources over time, often to achieve a certain aim or aims and /or subject to a set of constraints. "*

or

*" A feasible **Timetable**, or **Schedule**, is one that satisfies its associated set of constraints."*

1.4 AIM

The aim of this project is to develop an automated course scheduling system under web-based application. Courses timetable will be generated automatically for a particular semester. The administrators could view, edit and manage the rooms, courses, lecturers and other timetable information stores in database through the Internet; the timetable coordinators could generate timetable online; students and lecturers could view their timetable through the Internet.

1.5 OBJECTIVES

The objectives of this project:

1. Reducing the time and effort for generating a course schedule in FSKTM. The system will speed up the generation of course schedule in our faculty.

2. Intelligent and flexible enough to overcome variety of schedule constraints encountered in real-life problems such as students timing conflict, lecturer preference timing and rooms allocation. The system will produce schedules that are feasible and with sufficient quality to be used.

3. All timetables should be automatically generated.
4. Be user-friendly and convenient: easy to use and the users could view timetable at any places, any time through the Internet.
5. Be standard and easy enough for the administrators and timetable coordinators to be trained to handle and maintain the system in short period.

1.6 SCOPE

1.6.1 COURSE

1. Courses encompass including the Diploma, Degree, Minor IT, Master and PJJ courses offered in FSKTM.
2. It does not cover the courses offer by FSKTM to other faculty students at other faculty.
3. Scheduling to University courses like co-curriculum also not include in the system.
4. The system does not offer tutorial class scheduling.
5. Two different kinds of courses could not be taught at the same period at the same room. However, two or more different course codes but with the same course name and syllabus are allowed to be taught at the same period and at the same room.
6. One course may teach by more than one lecturer.

1.6.2 TIME PERIOD

1. Hour-based time slot. The start time and end time for each course must start or end at hour-based time such as 8:00am or 5:00pm and so on. It does not allow time slot to be held quarterly or half-hourly such as 8:15am, 5:30pm, 9:10pm and so on.
2. Start time of the day is at 8:00am and end time of the day is at 10:00pm.
3. Sunday is not included in course scheduling.

1.6.3 ROOM

1. Rooms being used are limited in FSKTM only.
2. Lab classes should be scheduled to computer labs only.

significant in order to develop an automated course scheduling system that will take over the human's effort in generating the timetable in FSKTM.

1.8 EXPECTED OUTCOME

At the end of the project, the FSKTM Course Scheduling System (FCSS) should be able to provide the following outcome:

1. Publish the timetable on the web. The timetable of each program could be shown separately and each timetable will have the time period for every course in a week.

2. Stakeholders could view the timetable grouped by years, lecturers, courses and rooms.

3. The timetable coordinators could set the courses offered and assign lecturer to the particular courses for a corresponding semester.

4. The administrators could manage and update information about rooms, courses, lecturers and users.

5. Secure the system from unauthorized people by using user ID and password. Only authorized people such as administrators and timetable coordinator could access the administration module and course-scheduling module. This is to prevent other people from modifying the database and timetable.

6. The timetable will be generated by the scheduling subsystem to ensure that the final timetable fulfills all hard constraints.

1.9 GLOSSARY OF TERMS

While every specific scheduling problem has its own associated terminology, which associated with the university courses timetabling problem is not specific and should be easily understood. The following table, included for completeness, gives a list of the terms used in this thesis in relation to course timetables.

Term	Description
Conflict / Clash	Two course conflict if they may not be scheduled concurrently under any circumstances. For example, if they have a student in common.
Constraint	A constraint places a restriction on when or where courses may be scheduled. A constraint is described as hard if it must be satisfied and soft if it is desirable but not absolutely necessary that it should be satisfied.
Course	An event or lesson involving a specific set of students and possibly other resources such as video recorders or overhead projectors.
Feasibility	A timetable is feasible if it satisfied all its hard constraints.
Period / Time Slot	A period is a fixed time slot in which courses may be scheduled.
Resource	Resources are physical entities that are referred to by the timetable. For example: student, room, and overhead projector. Time periods may also be thought of as resources.
Room / Hall	A venue where courses may be held.
Term / Semester	Period need to complete certain program taken / offered. Typically, an institution will have two or three term / semester in a year.

Table 1.1: Terms Of Timetabling Problem

1.10 PROJECT SCHEDULE

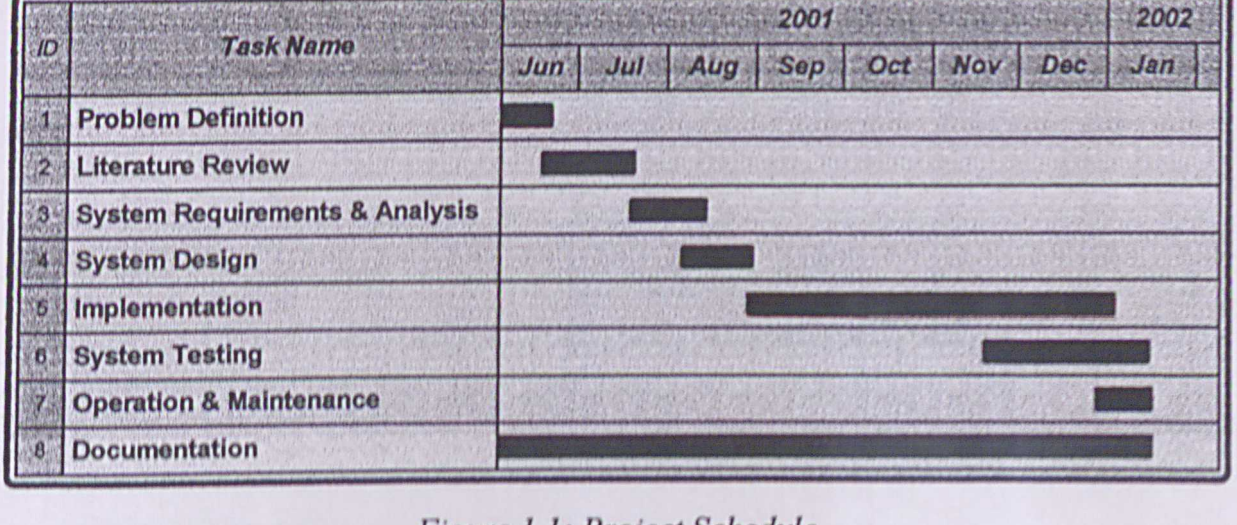


Figure 1.1: Project Schedule

CHAPTER 2

LITERATURE REVIEW

2.1 WHAT IS LITERATURE REVIEW?

Literature review is a critical look at the existing research that is significant to the work that a researcher is carrying out. It summarizes, interprets, and evaluates existing "literature" (or published material) in order to establish current knowledge of a subject. The purpose for doing so relates to ongoing research to develop that knowledge: the literature review may resolve a controversy, establish the need for additional research, and/or define a topic of inquiry.

In this project paper's context, literature review is an analysis that used to gather information about the system I intend to develop. This approach is used to evaluate existing system on the same topic so that a better product can be developed. It also includes the comparison of a few software, tools and approach to get the best outcome. Without this analysis, I would not be able to identity the strengths and weaknesses for the system I am going to develop.

2.2 SOME ISSUE IN TIMETABLE AND SCHEDULE CONSTRUCTON

Each different occurrence of the general scheduling problem carries with its own sets of jargon, rules and requirements. Very often, the problems are so different that they may hardly be classified as of the same form and more often; an efficient solution methodology for one will prove inefficient for another. Some examples of timetable/schedule are school timetable, rail timetable, job shop schedule, university course timetable, university examination timetable and so on.

Each type of the timetable has its own general aims, specific constraints and resources. These must be elicited before a scheduler may start the process of constructing a schedule. Similarly, a host of other issues must also be addressed. Some examples of the sort of question a scheduler must ask are:

- What is the purpose of the schedule?
- What constraints must the schedule satisfy?

- What qualities are desirable in the schedule?
- Does a feasible schedule exist?
- Who will be affected by the schedule?
- What resources are available?
- How much time is available to produce the schedule?

2.3 UNIVERSITY COURSE TIMETABLING SYSTEM

The university course timetabling problem consists in scheduling a set of lectures for each course within a given number of rooms and time periods. The main different with school problem is that university courses can have common students, whereas school classes are disjoint of students. Moreover, school's teachers always teach to more than one class, whereas in university, a lecturer may teach only one class. In addition, in the university problem, availability of rooms and the size plays an important role, whereas in the high school problem they are often neglected because, in most cases, we can assume that each class has its own room.

2.4 TIMETABLING FROM DIFFERENT PERSPECTIVES

Romero [1] identifies three different groups of stakeholders in the timetabling process. These are the administrators, the departments and the students.

2.4.1 THE ADMINISTRATORS

Administrator wants no conflicts in course timetabling. This may be interpreted as the stating of a minimum set of constraints (hard constraints), which describe the required properties of a feasible timetable.

It is perhaps more useful to make a further distinction within this category between the timetable coordinator and rest of the university or institution administration. It is likely that the institution will decide whether certain constraints are allowable or not. The distinction is thus that the administration is involved at a political level, perhaps even as a final arbitrator whereas the timetable coordinator is involved at a particular level performing the timetable construction itself.

2.4.2 THE DEPARTMENTS

The departments want the schedule to be consonant with the development of the subject taught and not to disturb any other department activities. Some departments have requirements for specific classrooms. A department may set requirements for particular courses and also express concern on behalf of the students of that department if they have what is considered a bad timetable.

2.4.3 THE STUDENTS

The students' view of the timetable is largely restricted to that section that directly affects them – their own personalized course schedule. The logistics and difficulty of the problem is irrelevant expecting where it directly affects their own schedule.

No research has really been carried out into the wants of students, whether they prefer early lectures or lectures spread evenly through the timetable. For example, some might prefer morning lecture whereas others prefer those scheduled in the afternoon. In this case, there is no absolutely fair way to choose where to put the course time period if it advantages one group over another. In most cases, the complexity of the problem would not allow such preferences to be made anyway.

2.5 SEARCH, OPTIMIZATION AND UNDERLYING PROBLEM

In some cases, the timetabling problem consists in finding any timetable that satisfies all the constraints. In these cases, the problem is formulated as a search problem.

In other cases, the problem is formulated as an optimization problem. That is, what is required is a timetable that satisfies all the hard constraints and minimizes (or maximizes) a given objective function, which embeds the soft constraints. As shown later, in some approaches, the optimization formulation is just a means to apply optimization techniques to a search problem. In this case, what is minimized is the so-called distance to feasibility. Even when the problem is a true optimization problem, the distance to feasibility may be

included in the objective function. This is generally done to facilitate the navigation of the search space.

In both cases (search and optimization), we define the underlying problem, which is the problem of deciding if there exists a solution in the case of a search problem, and the problem of deciding if there exists a solution with a given value of the objective function in the case of an optimization problem. When we mention the complexity of the problem, we refer to the complexity of the underlying decision problem. [2]

2.6 REVIEW ON TECHNIQUES APPLIED TO AUTOMATED TIMETABLING PROBLEM – TIMETABLING ALGORITHMS

Most of the early techniques were based on a simulation of the human way of solving the timetabling problem. Later on, researches started to apply general techniques to this problem such as Integer Programming, Network Flow and Graph Coloring. More recently, some techniques based on Artificial Intelligence appeared such as Simulated Annealing, Tabu Search, Genetic Algorithms, Evolutionary Algorithms and Constraint Logic Programming.

2.6.1 GENERAL TECHNIQUES

2.6.1.1 HEURISTICS AND GRAPH COLORING ALGORITHM

The Graph Coloring problem is one of the classical NP-complete problems on graphs. In this algorithm, each course is associated with a vertex in the graph and an edge drawn in the graph between any pair of vertices representing courses that may not be scheduled concurrently. A proper vertex coloring of the graph is where any pair of vertices with an edge between them is not colored the same. Each color used corresponds to a different period and the number of colors used gives the total number of periods required.

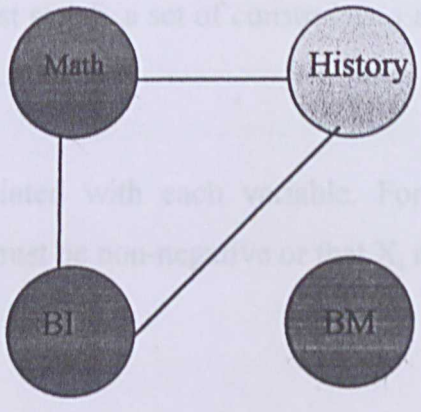


Figure 2.1: A Timetabling Conflict Graph

For example, in Figure 2.1, Math and BM may be scheduled in the same period but not Math and History as the edge indicates they may not be scheduled together. In this case the minimum length timetable must be three periods long.

In Heuristics approaches, we could use a sequential algorithm assigning courses to periods, taking one course at a time and placing it in the first available period. This was adapted for course timetabling by choosing the period with the least number of student conflicts when the numbers of periods are restricted and the course clashed with at least one other in each period.

There are many methods based on this algorithm. For example, Welsh and Powell (1967) propose a variant of the above method in which the vertices are ordered by degree. That is, the vertices with highest degree are colored first. The underlying idea of this method is that the vertices with high degree are the most difficult to be colored [3].

2.6.1.2 INTEGER PROGRAMMING

An Internet Programming problem is a Linear Programming Problem in which some or all of the variables to be optimized must be positive integers. A Linear Programming problem is an optimization problem, which has the following characteristics:

Attempt to maximize (or minimize) a linear function of the decision variables. The function that is to be maximized or minimized is called the objective function. The value

of the decision variables must satisfy a set of constraints. Each constraint must be a linear equation or linear inequality.

A sign restriction is associated with each variable. For any variable X_i , the sign restriction specifies that X_i must be non-negative or that X_i may be unrestricted in sign.

This technique define:

$Y_{ik} = 1$ if course i is scheduled in period k, or 0 otherwise,
C_{ik} to be the cost of scheduling course i in period k and
$X_{ij} = 1$ if course i conflicts with course j.

Internet programming has been tried in a few cases to solve timetabling problems but its use is certainly not widespread. This is because of the generally held belief that "when the problem is expressed mathematically, the number of variables and constraints become unmanageably large for practical size problems" [3].

2.6.1.3 NETWORK FLOW

In university timetabling, network model could be implemented as it contains three levels, plus a source and a sink vertex. The first level is the Department Level, which includes a vertex for each department, such that all of these vertices are connected to the source. The second level is the Faculty/Staff Level, which includes a vertex for each possible combination of teacher and course taught by the teacher; these vertices are connected to the vertices representing the departments to which the teachers belong. The third level is the Room Size/Time Level, which contains a vertex for each combination of room and time. Each vertex of this level is connected to a vertex of the second level only if the size of the room represented by the vertex is compatible with the number of students of the course represented by the other vertex. An edge between levels two and three represents a possible lecture.

The network model can be solved in polynomial time; however it does not prevent the solution from assigning a single teacher to multiple lectures at the same time. The

procedure therefore solves the problem and, it finds a feasible solution, the process is over, otherwise a human intervention changes some of the weights manually so as to get rid of the reason of unfeasibility. The procedure is executed several times until a feasible solution is obtained [2].

2.6.2 ARTIFICIAL INTELLIGENCE TECHNIQUES

2.6.2.1 TABU SEARCH

Tabu Search is a local search technique designed to solve optimization problems. Local search techniques are based on the notion of neighbor. Given an optimization problem P, let S be the search space of P, and let f the objective function to minimize. A function N, which depends on the structure of the specific problem, assigns to each feasible solution $s \in S$ its neighborhood $N(s) \subseteq S$. each solution $s' \in N(s)$ is called a neighbor of s.

A local search technique, starting from an initial S_{init}, which can be obtained with some other technique or generated at random, the algorithm, enters in a loop that navigates the search space, stepping from one search to one of its neighbors. The connectivity of the search space is necessary condition for the technique to work effectively.

Tabu Search, the algorithm explores a subset V of the neighborhood $N(s)$ of the current solution s; the member of V that gives the minimum value of the objective function becomes the new current solution independently of the fact that its value is better or worse than the value in s.

In order to prevent cycling, there is a so-called Tabu list, which is the list of solutions to which it is forbidden to move back. It is the list of the last k current solutions, where k is a parameter of the method, and it is run as a queue; that is, when a new solution is added, due to a move, the oldest one is discarded [4].

2.6.2.2 SIMULATED ANNEALING

Simulated Annealing is a probabilistic local search technique for finding solutions to optimization problems.

The technique starts by creating a random initial solution. The main procedure consists of a loop that generates at random at each iteration a neighbor of the current solution. Like for Tabu Search, the definition of neighbor depends on the specific structure of the problem.

Let's call Δ the difference in the objective function between the new solution and the current one and suppose to deal with a minimization problem. If $\Delta < 0$ the new solution is accepted and becomes the current one. If $\Delta > 0$ the new solution is accepted with probability $e^{-\Delta/T}$, where T is a parameter, called the temperature.

The temperature T is initially set to an appropriately high value T_0. After a fixed number of iterations, the temperature is decreased by the coding rate a, such that $T_n = a \times T_{n-1}$, where $0 \leq a \leq 1$.

The procedure stops when the temperature reaches a value very closed to 0 and no solution that increases the objective function is accepted anywhere. The system is frozen. The solution obtains when the system is frozen is obviously a local minimum.

The control knobs of the procedure are the coding rate a, the number of iterations at each temperature, and the starting temperature T. [2]

2.6.2.3 GENETIC ALGORITHM

Genetic Algorithm is a solution technique for optimization problems. Differently from Tabu Search and Simulated Annealing they are not based on local search [4]. A Genetic Algorithm starts with a set of solutions randomly chosen $\{ s^0_1, \ldots, s^0_n \}$, which is called the population at time 0.

The core procedure is a loop that creates the population $\{ s^{t+1}_1, \ldots, s^{t+1}_n \}$, at time t + 1 starting from the population at time t. To this aim the value of the objective function is computed for each solution s^t_i based on a weighted randomization, n elements of the population at time t are selected. Obviously, some solution may be selected more than once. The randomization is biased by the value of the objective function so as to assign a

higher probability to be selected to the solutions that result in a better of the objective function. In this way the best solutions get more copies, and the worse ones probably die off.

At this point, each solution is selected for recombination with a given probability (P_R). The recombination is done by the crossover operator. That is, two selected solutions are mixed by swapping corresponding segments of their representations. One of the most common ways to do the crossover is by selecting a fixed number of positions in which the swapping takes place (fixed-point crossover).

For example, if two solutions are represented by the strings **abcdef** and **uvwxyz** and we choose two crossover points after the second and the fifth character, then the new solutions would be **abwxyf** and **uvcdez**. In addition, mutation arbitrarily alters randomly some part of some solutions randomly selected, based on a given probability value (P_M).

The method terminated either when it generated a fixed number of populations, or when the best solution reaches a certain value of the objective function, or when the algorithm does not make any programs for a certain number of iterations.

The main control parameters of the method are the population size *n*, the probability of crossover P_R, and the probability of mutation P_M.

2.6.2.4 EVOLUTIONARY ALGORITHMS

Evolutionary Algorithms use computational models of evolutionary processes as key elements in the design and implementation of computer, based problem solving systems. A variety of evolutionary computational models have been proposed. They share a common conceptual base of simulating the evolution of individual structures via processes of selection, mutation, and reproduction. The processes depend on the perceived performance of the individual structures as defined by an environmental. [3]

An Evolutionary Algorithms works a set (population) of candidate solutions to a problem typically initially filled with random possible solutions. Each encodes a solution as a

string of symbols (genes). And Evolutionary Algorithms then typically proceeds by iterating through the following steps until some termination criterion is met. [5]

1. *Evaluation*

   Using some predefined problem-specific measure of fitness, evaluate every member of the current population. The idea is that fitter candidates are in some way closer to being one of the solutions being sought.

2. *Selection*

   Select candidate solutions from the current generation to be used for breeding. This may be done entirely randomly, so stochastically based on fitness or in other ways (but usually based on fitness, such that fitter individuals have more chance of being chosen).

3. *Breeding*

   Produce new individuals by using genetic operators on the individuals chosen in the selection step. There are two main kinds of operators:

   Recombination: A new individual is produced by recombining features of two or more 'parent' solutions.

   Mutation: A new individual is produced by slightly altering an existing one.

4. *Population update*

   The set is altered, typically by choosing to remove some or all of the individuals in the existing generation (usually beginning with the least fit) and replacing these with individuals produces in the breeding step. The new population becomes the current generation.

2.6.2.5 CONSTRAINT LOGIC PROGRAMMING

Constraint Logic Programming is a technique, which has much in common with the heuristic assignment algorithms. A constraint satisfaction problem (CSP) is represented by a constraint network, consisting of a set of n variables, $X_1 \ldots X_n$; their respective value domains, $R_1 \ldots R_n$; and a set of constraints. A constraints $C_i (X_{i1} \ldots X_{ij})$ is a subset of the

Cartesian product $R_{i1} x .... x R_{ij}$, consisting of all tuples of values for a subset $(X_{i1} .... X_{ij})$ of the variables which are compatible with each other. "A solution of the network is an assignment of values to all the variables such that all the constraints are satisfied". This is a more powerful representation then the graph coloring model alone as more complex constraints may be represented. The graph coloring problem is equivalent to a binary Constraint Logic Programming in which each constraint only has two variables.

This is solved by trying to assign values to each of the variables in turn and then backtracking if a dead end is reached and removing constraints if it looks like no solution may be found. The order in which variables are chosen (labeling strategies) and the order on the values assigned to the variables may be altered too try to speed up the search. [3]

2.7 ANALYSIS AND SYNTHESIS ON PREVIOUS SIMILAR PROJECTS

There are around three similar projects regarding FSKTM course scheduling system attempt by students of FSKTM. It must be clearly noted that although some of the developed system fulfill the objectives of their developer, but they fail to meet the constraints in real world which lead to their system can not be apply in faculty working environment. This further leads to the study of these course scheduling systems in this project paper.

2.7.1 REVIEW ON " INTELLIGENCE SYSTEM FOR COURSE SCHEDULING FOR FSKTM "

This system is conducted by Tan Wei Yee who was supervised by Dr. Syed Malik in session 1999/2000. She solved the scheduling problem for FSKTM using an artificial intelligent approach. The intelligent system is to automate the scheduling process by given input parameters such as course, characteristic of course, constraints of rooms, time and lecturer availability. She developed the system using Prolog language.

2.7.1.1 SYSTEM STRENGTH

1. *Speed*

   The system speed up the generation of course schedule. The generation of schedule with the system is very fast and user does not need to spent much time on clerical work involve in scheduling. The schedule can be easily got from the data files that store the schedule.

2. *Automated scheduling process*

   The scheduling process is automated after given the necessary input parameters such as the course, lecturer, time and room.

3. *Dynamic and interactive content*

   Because the user provides all the data during the execution of the application, the content of the data are dynamic and interactive where the user can have their own codes and others for their data.

2.7.1.2 SYSTEM LIMITATION

1. *No error checking*

   There are no validations on user inputs. User may enter something that may affect the output of the system.

2. *Record cannot be modified*

   The record in the database cannot be modified easily. The application creates new records every time and there is no facility to modify or delete the record stored in the data files.

3. *Re-key-in every time*

   The user has to re-key-in the data although he/she may need to generate a schedule with the same data again after he/she has exit from the program.

4. *Not web-based application*

   The application is not a web-based application. So user could not access the application via browser in any workstation.

5. *Additional query functions*

   Application should provide other query functions than scheduling only. The functions include listing out the courses, lecturers, venues and many more.

2.7.2 REVIEW ON " A TIMETABLE SCHEDULER FOR FSKTM "

This system was conducted by Tan Kay Sim who was supervised by Prof. Madya Dr. Lee Sai Peck in session 1999/2000. She had developed a web-based application for automated timetables. According to her statements, she claims that her system not only can generate timetable automatically, but it also can be modified manually. Her system is developed based on a general technique (technique she formulated). There is no artificial intelligent technique applied in her system.

2.7.2.1 SYSTEM STRENGTH

1. *Wide accessibility*

   The system is developed in web-based application. This has provided wide accessibility to users who can access the system virtually from anywhere. Client side software only requires the installation of a web browser to access the system. Furthermore, browsers are already available across all platforms.

2. *User-friendly interface*

   As a web-based system, this system employs direct manipulation interfaces to application system. It has a naturally interface with menus, buttons, scrollbars and text boxes, which make the system relatively easy for users to learn and use. Users should find the system user-friendly and interesting, without needing much skill on handling it.

3. *Produce more information in timetable*

   Existing timetable shows the course code, lecturer code and room code for particular time slot in timetable. This system provide more information about these codes where a description about this code will be shown when the user click on these codes.

4. *System reliable*

   This system caters most of the possible errors encounter in timetable coordinator section and user section. Users' input is validated and verified through using client side scripting and server side scripting.

5. *Retrieving and manipulating records*

   The application allows efficient ways in retrieving relevant and necessary records on the screen. It is easy on manipulating records found in the database, such as adding new records, editing and deleting existing records.

6. *Security*

   Only authorized users who have login authentication into the application is allowed to manipulate and make changes on the system and database. The timetable coordinator also can change password by entering the old password correctly.

7. *Produce convenient timetable*

   Students not only could view the overall timetable, they also could view their timetable by their degree program. This will reduce the response time of web server because only the information needed will be shown. Besides that, students and lecturers could view their timetable by a particular course, by lecturer and by room.

2.7.2.2 SYSTEM LIMITATION

1. *Database management limitation*

   The database in this system is not fully secured, as any one who found the database file could still manipulate on it. There is also no proper management on database file, where no functions were created in the application to backup and to compact the database.

2. *Timetable print out*

   This system does not provide facility on printing the timetable on web. This is not effective, as users have to copy the timetable manually. This is not much different with copying the timetable from the notice board.

3. *Lecturer section*

   There is no lecturer section in this system. This section is needed for lecturers to review and select their preferable and not preferable teaching time slot.

4. *Security*

   For system implemented in web-based application, the single type of access control mechanism is not enough to secure the data integrity, availability, privacy and safeguard. In the Internet environment, this system exposed to external threats like hackers, viruses and worms. It is suggested to enforce the access control policies or implement it in Intranet environment where the system is sheltered by firewall.

2.7.3 REVIEW ON " ELECTRONIC TIMETABLE SYSTEM "

The system is conducted by Mokhairi Bin Mokhtar who was supervised by Cik Hannyzzura Pal @ Affal in session 1999/2000. Mokhairi employs graph coloring and heuristic method in this interactive timetable system. The system includes an interactive window based, iconic user interface and implemented in networking environment. The main feature in his system is that his system provides flexibility in scheduling.

2.7.3.1 SYSTEM STRENGTH

1. *User-friendly interface*

   The system's display is interesting due to the using of graphical user interface (GUI) concept. The application also supports WIMP interface (Window, Icon, Menu and Pointer). It creates no difficulties to novice who is unskilled in using the keyboard and mouse.

2. *Help facility*

The system has a good quality of help feature. The help facility assists users by explaining the timetabling process, the ways of adding, editing and deleting data. Users could access to help facility by clicking the "Help" button.

3. *Validation of data input*

Before the data is stored in the database, the record validation checking process will be conducted to verify the record integrity in the database. The purpose of field check is to ensure that every field is filled correctly.

4. *Security system*

Every user (management and lecturer) is given user ID and password respectively for accessing to the system. Lecture has the right on undertaking data manipulation to his/her course assignment only. Access to other course sections is forbidden.

5. *Schedule courses by lecturer*

Lecturer is permitted to undertake scheduling of courses assigned to his/her only. It means that lecturer is allowed to choose and set their preferable teaching time.

6. *Scheduling report*

The system permit scheduling report to be printed out or displayed on the screen according to course, program, room, lecturer and time.

2.7.3.2 SYSTEM LIMITATION

1. *Not automated scheduling*

The system needs input from users for day, course, lecturer, room and time slot to undertake scheduling. The system is only responsible on conflict checking before the schedule is generated. This is very time consuming for producing a course schedule compare to an automated scheduling system.

2. *No system interaction with existing information system in FSKTM*

   This system requires coordinator and lecturer to enter all the courses and lecturers particular and data into the system. Interaction between this system and Academic Information system (consists of courses information) and Academic Staff System (consists of lecturer information) do not exist. As a result, this system could not extract the corresponding information from the existing system and the users have to input the data manually.

2.8 WHAT IS WEB-BASED APPLICATION?

Web based functionality is achieved when built on the foundations of the World Wide Web, such applications can be run anywhere in the world at any time and are completely cross platform. All data and input by the user using the application is used to update a database or document that is then made available by the system to global users in real time.

The only client side software needed to access and execute web-based applications is a web browser environment. An example of such an application would be an online store accessed via Netscape Navigator or Microsoft Internet Explorer.

Web applications provide a rich interactive environment through which the user can further define their unique online experience. Without web applications to breathe life and provide user interaction, a web page is limited to static electronic text.

Key features of a web-based application are centrally located application, database and document, information is updated in real time, and system as well as applications is accessible remotely.

2.9 REVIEW ON THE INTERNET AND WORLD WIDE WEB

2.9.1 WHAT IS INTERNET?

The Internet has been around for over three decades. It began as a Department of Defense program for enabling computers to communicate over great distances without requiring a central server to route the communications traffic. Since those early days, the Internet has grown substantially. Early on it was adopted by the academic community, and more recently it has been commercialized. The federal government no longer funds the Internet directly; leaving private and public telecommunications companies in charge of the major backbones – the major network connections of the Internet. The telecommunications companies charge Internet service provides for connections to the backbone, and Internet service providers in turn charge companies and individuals for their access to the Internet. The Internet itself is nothing more than an enormous number of networked computers all over the world. Like any computer network, the Internet has various software programs running on it, such as e-mail, newsgroup, FTP, Gopher, and the World Wide Web.

2.9.2 ADVANTAGES AND DISADVANTAGES OF INTERNET

Advantages	Disadvantages
1. Can access any computer anywhere with Internet access.	1. Limited access to intranet resources.
2. In terms of design, anything is possible such as great multimedia and video presentations.	2. In terms of security, there is a concern that unauthorized users could access proprietary information.
3. Take advantage of external resources by linking to vendor websites, etc.	

Table 2.1: Advantages And Disadvantages Of Internet

2.9.3 WHAT IS WORLD WIDE WEB (WWW)?

The World Wide Web, or Web, was born in 1989 at CERN (the European Laboratory for Particle Physics). Since then, it has grown at a phenomenal rate. Today, Web traffic accounts for somewhere between one third and one half of the total traffic on the Internet. Because the Internet consists of many other sources of traffic, many of which have been around for decades, this is an impressive feat. [6]

So, what is the Web? In simple terms, the Web is a part of the Internet that uses the Hypertext Transfer Protocol (HTTP) to display hypertext and images in a graphical environment. Hypertext refers to the ability to present text documents that are interlinked. You might click on a portion of the text in a document and be taken to another section of text in a different document. The Web is based on the concepts of hypermedia, which ia a superset of hypertext. Think of hypermedia as various forms of media (text, graphics, sound files, and so on) that are interlinked.

The process of viewing a document on the Web starts when a Web browser sends a request to a Web server. The Web browser sends details about itself and the file it is requesting to the Web server receivers and reviews the HTTP request headers for any relevant information, such as the name of the file being requested, and sends back the file with HTTP response headers. The Web browser then uses the HTTP response headers to determine how to display the file or data being returned by the Web server.

2.10 REVIEW ON CLIENT / SERVER COMPUTING

2.10.1 CLIENT/SERVER CONCEPTS

The premise of client/server computing is to distribute the execution of a task among multiple processors in a network. Each processor is dedicated to a specific, focused set of subtasks that it performs best, and the end result is increased overall efficiency and effectiveness of the system as a whole. Splitting the execution of tasks between processors is done through a protocol of service requests; one processor, the client, requests s service from another processor, the server. The most prevalent implementation of client/server processing involves separating the user interface portion of an application from the data access portion. [7]

2.10.2 CLIENT PROCESS

The client is a process (program) that sends a message to a server process (program), requesting that the server perform a task (service). Client programs usually manage the user-interface portion of the application, validate data entered by the user, dispatch requests to server programs, and sometimes execute business logic. The client-based process is the front-end of the application that the user sees and interacts with. The client process contains solution-specific logic and provides the interface between the user and the rest of the application system. The client process also manages the local resources that the user interacts with such as the monitor, keyboard, workstation CPU and peripherals. One of the key elements of a client workstation is the graphical user interface (GUI). Normally, a part of operating system i.e. the window manager detects user actions, manages the windows on the display and displays the data in the windows. [8]

2.10.3 SERVER PROCESS

A server process (program) fulfills the client request by performing the task requested. Server programs generally receive requests from client programs, execute database retrieval and updates, and manage data integrity and dispatch responses to client requests. Sometimes, server programs execute common or complex business logic. The server-based process may run on another machine on the network. This server could be the host operating system or network file server; the server is then provided both file system services and application services. Or in some cases, another desktop machine provides the application services. The server process acts as a software engine that manages shared resources such as databases, printers, communication links, or high powered-processors. The server process performs the back-end tasks that are common to similar application. [8]

2.11 REVIEW ON SYSTEM ARCHITECTURE

2.11.1 TWO-TIER ARCHITECTURE (CLASSICAL CLIENT-SERVER ARCHITECTURE)

Classical client-server application architecture (two-tier architecture) is based on client computer where services and queries are constructed and a data call connection is made to

server computer, with no intervening server. The results of the query are returned as data stored within the local cache of the client computer. A visual display will be connected to this cache for displaying the query's results and data. High-performance data retrieval technologies became an important consideration when constructing the entire architecture.

It is typically used in small environments (less than 50 users). A common error in client/server development is to prototype an application in a small, two-tier environment and then scale up by simply adding more users to the servers. This approach will usually result in an ineffective system, as the server becomes overwhelmed. To properly scale to hundreds or thousands of users, it is usually necessary to move to a three-tier architecture.

2.11.2 MULTI-TIER ARCHITECTURE

The core design of a multi-tier, or three-tier, computing is to enable application codes to be run separately from the client computer and the database server. Three-tier architecture introduces a server (or an "agent") between the client and the server. The role of the agent is many folds. It can provide translation services (as in adapting a legacy application on a mainframe to a client/server environment), metering services (as in acting as a transaction monitor to limit the number of simultaneous requests to a given server), or intelligent agent services (as in mapping a request to a number of different servers, collating the results, and returning a single response to the client.) [9]

The introduction of a server (or and "agent") between the client and the server supports the performance of large application processing in the middle tier. This application architecture is beneficial to developers as moving the data access and business logic to the middle tier denotes less application code for client's computers. It also eases the management of business process components and data access logic. Therefore, multi-tier architecture present a more scalable and flexible way for building application.

2.12 REVIEW ON PROGRAMMING LANGUAGE AND TECHNOLOGY

2.12.1 ACTIVE SERVER PAGES (ASP)

Microsoft® Active Server Pages (ASP) is a server-side scripting environment that you can use to create and run dynamic, interactive Web server applications. With ASP, you can combine HTML pages, script commands, and COM components to create interactive Web pages or powerful Web-based applications, which are easy to develop and modify. ASP enables you to run ActiveX™ scripts and ActiveX server components on the server. By combining scripts and components, developers can create dynamic content and powerful Web-based applications easily. Active Server Pages provides native support for both Microsoft JScript and VBScript.

ASP can work with any Web browser. The output of an ASP file is plain HTML, the content of which can be customized for the capabilities of the client. ASP will run on Microsoft Windows NT Server 4.0, Windows NT Workstation 4.0 with Peer Web Services, and Microsoft Windows® 95 with Personal Web Server. However, Windows NT 3.51 and Windows NT 4.0 on MIPS are not supported ASP.

ASP is a component of Internet Information Server, and thus uses Windows NT Security. ASP files can be easily restricted to just certain users through secure Windows NT authentication, basic Web authentication, or client-side certificates. For additional security, all client-to-server communications can be secured with Secure Sockets Layer (SSL).

An Active Server Pages application can integrate with any ODBC-compliant databases including Microsoft SQL Server™, Oracle, Sybase, Informix, and DB2 databases. Any OLE 2 application, such as Lotus Notes or Microsoft Excel, can also be scripted to access or process information. Users can also write components to access online data feeds and legacy mainframes.

2.12.2 JAVASERVER PAGES (JSP)

JavaServer Pages (JSP) technology provides a simplified, fast way to create web pages that display dynamically-generated content. The JSP specification, developed through an industry-wide initiative led by Sun Microsystems, defines the interaction between the server and the JSP page, and describes the format and syntax of the page. [21]

JSP pages use XML tags and scriptlets written in the Java™ programming language to encapsulate the logic that generates the content for the page. It passes any formatting (HTML or XML) tags directly back to the response page. In this way, JSP pages separate the page logic from its design and display.

JSP technology is part of the Java technology family; it uses a Java programming language-based scripting language, and JSP pages are compiled into servlets. JSP pages may call JavaBeans™ components (beans) or Enterprise JavaBeans™ components (enterprise beans) to perform processing on the server. As such, JSP technology is a key component in a highly scalable architecture for web-based applications.

JSP pages are not restricted to any specific platform or web server. The JSP specification represents a broad spectrum of industry input.

2.12.3 COMMON GATEWAY INTERFACE (CGI)

The Common Gateway Interface (CGI) is a mechanism for creating scripts on the server, which can then be used to create dynamic applications. It has been around for quite a bit longer than ASP, and right now the majority of dynamically-created pages on the web are created using CGI and a scripting language. However, it is incorrect to assume that CGI does the same job as ASP. Rather, CGI allows the user to invoke another program (such as Perl script) on the web server to create the dynamic web page, and the role of CGI is to pass the user-supplied data to this program for processing. However, it does provide the same end-results -a dynamic web application.

However, CGI has some shortcomings. The major one is that it adds extra level to the browser-server model of interaction: namely, it is necessary to run a CGI program to

create the dynamic page, before the page is processed on the server. Also, the format in which CGI receives and transmits data means that this data is not easily manipulated by many programming language, so a programming language that has good facilities for manipulating text and communicating with other software have to be used. The most able programming languages that can work on any operating system for doing this are C, C++ and Perl. While these languages can adequately do the job, they are some of the more complex languages to learn. Visual Basic does not offer sufficiently adequate text-handling facilities, and is therefore rarely used in CGI.

2.12.4 VBSCRIPT

Visual Basic Scripting Edition (VBScript) was introduced by Microsoft to allow web page developers to leverage their existing Visual Basic skills in creating client-side script. It inherits its syntax and structure from Visual Basic programming language. VBScript is an alternative to JavaScript in the client-side scripting language, however, only Microsoft Internet Explorer supports this language. Netscape Navigator users require a Netscape Plug-in called ScriptActive, developed by rd party developer NCompass in order to enable VBScript to run on the browser. VBScript provides the ability similar to JavaScript. It can be used to validate form data, displaying status bar messages, working with cookies and ActiveX control. For example, users can write a VBScript function to verify that users enter valid information into a form. Without any network transmission, an HTML page with embedded VBScript can interpret the entered text and alert the user with a message dialog if the input is invalid. Or users can use VBScript to perform an action (such as play an audio file, execute an applet, or communicate with a plug-in) in response to the user opening or exiting a page.

2.12.5 JAVASCRIPT

JavaScript is a cross-plattorm scripting language, which is simple, interpreted, and object-oriented. It can be used to add simple interactive behaviors to an HTML page by means of a script of keywords inserted into a Web page. Its originate trom LiveScript that developed by Netscape to provide a way to interface with Java. Sun, the developer of Java, helped Netscape rework LiveScript and called it JavaScript. Anyway, Java Script is not derivative of Java. It lacks of power of a full-featured programming language.

Netscape support the language since its version 2.0 and Microsoft Internet Explorer giving support since its Internet Explorer 3.0 through JScript. [24]

The main roles JavaScript play in the web pages are form validation, responding to input, dialog boxes, detecting browser characteristics, updating the browser properties, math capabilities, using cookies to keep visitor information, date and time information, Integrating with Java, basic graphics and dynamic HTML.

NetScape is working with ECMA (European Computer Manufacturers Association) to deliver a standardized, international programming language based on core JavaScript. ECMA is an international standards association for information and communication stems. This standardized version of JavaScript, called ECMAScript, behaves the same way in all applications that support the standard, Companies can use the open standard language to develop their implementation of JavaScript. The first version of the ECMA standard is documented in the ECMA-262 specification.

The current version of JavaScript is JavaScript version 1.4. It provides the some new atures and enhancements such as exception handling, additional operators in and stance of, as well as changes to LiveConnect and functions object. JavaScript 1.4 is fully compatible with ECMA-262. However the second version of the ECMA specification was not finalized when JavaScript 1.4 was released.

Microsoft's version of JavaScript is called JScript. It is Microsoft full implementation of CMA 262 language specification, plus some enhancements that take advantage of capabilities of Microsoft Internet Explorer. The JScript version implemented in Microsoft Intemet Explorer 4.0 is version 3.0 and the Microsoft Visual Studio 6.0 implements version 4.0.

2.12.6 HTML

HTML (Hypertext Markup Language) is the set of "markup" symbols or codes inserted in a file intended for display on a World Wide Web browser. The markup tells the Web

browser how to display a Web page's words and images for the user. The individual markup codes are referred to as elements (but many people also refer to them as tag).

HTML is a standard recommended by the World Wide Web Consortium (W3C) and adhered to by the major browsers, Microsoft's Internet Explorer and Netscape's Navigator, which also provide some additional non-standard codes. However, both Internet Explorer and Netscape implement some features differently and provide non-standard extensions. Web developers using the more advanced features of HTML 4 may have to design pages for both browsers and send out the appropriate version to a user. Significant features in HTML 4 are sometimes described in general as dynamic HTML.

2.13 REVIEW ON DATABASE

2.13.1 MICROSOFT SQL SERVER 2000

Microsoft SQL server 2000 is a relational database management system for Windows NT-based systems. It is a scalable, high-performance database management system designed specifically for distributed client-server computing. SQL server 2000 provides tight integration with Windows and Windows based applications helping reduce the cost and complexity of deploying sophisticated applications. SQL server 2000 is an ideal database engine for powering web sites. Through tight integration with Internet Information Server, SQL server 2000 can be queried and updated via popular web browsers. SQL server's native Open Database Connectivity (ODBC) lets it inter-operate smoothly with the Internet Database Connector interface included with Internet Information Server. [23]

SQL Server 2000 is the complete database and analysis offering for rapidly delivering scalable e-commerce, enterprise, and data warehousing solutions. It dramatically reduces the time required to bring these applications to market, while offering the scalability needed for the most demanding environments.

SQL Server 2000 provides the reliability users need to keep their business operations up and running. It can handle users workload today, and in the future as their business grows. With SQL Server 2000, users have both the flexibility to take maximum advantage of

their existing hardware investment and the agility to quickly adapt to their ever-changing business environment.

The core features and functionality in SQL Server 2000:

- *Data Warehousing.* Data Transformation Services (DTS), creating online analytical processing (OLAP) cubes, data warehousing, data mining, and more.
- *E-Commerce.* Extensible Markup Language (XML), ease of use, scalability, automated management, availability, and development.
- *Enterprise Solutions.* Including Microsoft Excel, Cognos, Knosys, and Maximal.

2.13.2 MICROSOFT ACCESS 2000

Microsoft Access 2000 is a window-based relational database management system. It provides powerful tools that help in organizing and sharing decisions. It is one of the most popular database management software as it is powerful and it is also easy to learn and use. Microsoft Access can be used to enter, store and manipulate data in a variety of ways. Other than that, it can be used to query for specific information or to produce (detailed or summary) reports based on certain criteria. [26]

Microsoft Access uses the Microsoft Data Engine (MSDE), a SQL server-compatible client/server engine designed for a single-user computer or small workgroup server. However, there are some weaknesses in Microsoft access, which is very slow transaction speed. Other than that, queries are stored locally (all data is pulled over the network instead of just the few records you want).

2.13.3 MYSQL

MySQL, the most popular Open Source SQL database, is provided by MySQL AB. MySQL AB is a commercial company that builds its business providing services around the MySQL database. MySQL is a relational database management system that stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. MySQL is Open Source Software that it is possible for anyone to use and modify. Anybody can download MySQL from the Internet and use it without

paying anything. Anybody so inclined can study the source code and change it to fit their needs. [25]

Why use MySQL?

MySQL is very fast, reliable, and easy to use. MySQL was originally developed to handle very large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Though under constant development, MySQL today offers a rich and very useful set of functions. The connectivity, speed, and security make MySQL highly suited for accessing databases on the Internet. MySQL is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and several programming interfaces.

2.13.4 DATABASE CONNECTIVITY – ADO (ACTIVE X DATA OBJECTS)

ADO is one of the built-in components delivered with Active X Server Pages. Using ADO, data from a variety of data providers can be stored and retrieved. The ADO can be used to access information stored in Microsoft Access, Microsoft SQL Server and Oracle Database Server within an ASP file. The ADO consists of seven independent objects as below:

- *Connection object*

  Represent a unique session with a data source. This object is used to open a connection to Microsoft SQL.

- *Recordset object*

  Represent record from a data provider. This object is used to alter the records contained in a Microsoft SQL Server table.

- *Field object*

  Represents an individual field in Recordset.

- *Command object*

  Represents a command. This object is used to execute a SQL procedure or a parameterized query.

- *Parameter object*

  Represents an individual parameter in a SQL stored procedure or parameterized query.

- *Property object*

  Represents data provider specific properties.

- *Error object*

  Represents ADO errors.

2.13.5 DATABASE CONNECTIVITY – ODBC (OPEN DATABASE CONNECTIVITY)

Open Data Base Connectivity, or ODBC, is a standard for accessing data. It was designed to allow the programmer to use a common set of routines to access the data stored in databases, regardless of the *type* of database in which the data was stored. This meant that once the programmer was connected to the database using ODBC, they could manipulate the data without worrying exactly where the data was stored, or which type of database was storing it. It provided interface transparency -so the programmer could access an Oracle database in the same way that they access a SQL Server database (Figure 2.2):

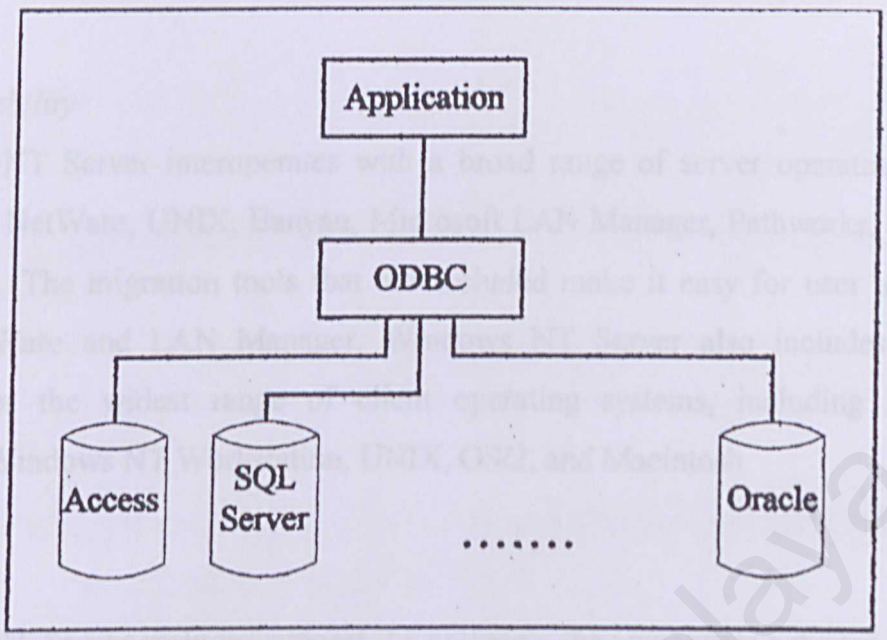


Figure 2.2: Data access using ODBC

There is no common format for saving databases to file, so the user cannot save a database using one database application and then, directly open the same file using a different database application (in the way that a word-processed file in the .txt format can be opened and read as plain text in NotePad, WordPad or Microsoft Word).

Fortunately, all sorts of database application store data using the structures such as tables, records, and keys. ODBC allows the user to get at this data without worrying about the nuts and bolts of the hosting database application.

2.14 REVIEW ON PLATFORM AND SERVER

2.14.1 MICROSOFT WINDOWS NT SERVER 4.0

Microsoft Windows NT Server 4.0 is a powerful, easy-to-use, multipurpose server operating system. Windows NT Server, including Windows NT 4.0 Option Pack and Microsoft FrontPage 98, makes the most complete platform available for building powerful business solutions.

Key features of Windows NT Server 4.0:

- *Interoperability*

  Windows NT Server interoperates with a broad range of server operating systems including: NetWare, UNIX, Banyan, Microsoft LAN Manager, Pathworks, SNA, and Macintosh. The migration tools that are included make it easy for user to upgrade from NetWare and LAN Manager. Windows NT Server also includes standard support for the widest range of client operating systems, including MS-DOS, Window, Windows NT Workstation, UNIX, OS/2, and Macintosh.

- *Protocols*

  Windows NT Server includes support for protocols than any other network operating system.

- *File sharing*

  Windows NT Server is the only network operating system that supports file sharing via NCP, X-Open SMB, and HTTP. Windows NT Server supports POSIX application interfaces, conforming to the open systems guidelines of the U.S government.

2.14.2 MICROSOFT WINDOWS 2000 SERVER

Microsoft Windows 2000 Server operating systems are the next generation in the Windows NT Server series of OS. In addition to providing a comprehensive Internet and applications platform, Windows 2000 Server builds on the strengths of Windows NT Server 4.0 by delivering increased reliability, availability, and scalability with end-to-end management features that reduce operating costs. Although Windows 2000 Server offers many new features and enhancements, arguably the most critical are those that relate to Internet capabilities. The Windows 2000 Server operating system builds on the solid Internet technologies delivered in Windows NT Server 4.0 to provide an agile, powerful Internet platform. It lets an organization not only get more out of existing applications, but also build new and versatile solutions that use the most complete set of Internet technologies available.

Main advantages new features:

- The ideal platform for building and running rich Web-based applications and services.
- Scalability and flexibility - Host lots of Web sites and more Terminal Services users while getting better use of your bandwidth and high performance even on the fastest networks.
- Security - With flexible authentication and authorization options, strong encryption services and flexible and secure network access, rest assured that only the right people have access.
- Increased server and network availability - It is resilient to application failures and allocates resources to preserve availability.
- Windows 2000 Server supports up to four CPUs on one machine.
- Supports the newest networking devices and technologies as well as the newest USB peripherals and advanced printer drivers

2.14.3 MICROSOFT WINDOWS 2000 PROFESSIONAL

Microsoft Windows 2000 Professional is built on Windows NT technology. It is easy-to-use because it has familiar Windows 98 user interface. Its integrated Web capabilities and broad support for mobile computers and hardware devices makes Windows 2000 the easy way for users to connect to the Internet anywhere and anytime. Its rock-solid reliability and improved manageability simplify desktop management for IT professionals.

The combined features of Windows 2000 Professional create the mainstream operating system for desktop and notebook computing in all organizations. The best features of Windows 98 - Plug and Play, easy-to-use user interface, and power management are integrated with the strengths of Windows NT standards-based security, manageability and reliability to form Windows 2000 professional.

2.14.4 APACHE SERVER

The Apache server is a powerful, flexible, HTTP/1.1 compliant web server. It implements the latest protocols, including HTTP/1.1 and is highly configurable and extensible with

third-party modules. Apache server can run on Windows NT/9x, Netware 5.x, OS/2, and most versions of Unix, as well as several other operating systems. [20]

Apache server implements many frequently requested features, including:

1. *DBM databases for authentication*
   Allow users to easily set up password-protected pages with enormous numbers of authorized users, without bogging down the server.

2. *Customized responses to errors and problems*
   Allow users to set up files, or even CGI scripts, which are returned by the server in response to errors and problems, e.g. setup a script to intercept 500 Server Errors and perform on-the-fly diagnostics for both users and yourself.

3. *Unlimited flexible URL rewriting and aliasing*
   Apache has no fixed limit on the numbers of Aliases and Redirects, which may be declared in the *config* files. In addition, a powerful rewriting engine can be used to solve most URL manipulation problems.

4. *Content negotiation*
   The ability to automatically serve clients of varying sophistication and HTML level compliance, with documents which offer the best representation of information that the client is capable of accepting.

5. *Virtual Hosts*
   A much requested feature, sometimes known as multi-homed servers. This allows the server to distinguish between requests made to different IP addresses or names (mapped to the same machine). Apache also offers dynamically configurable mass-virtual hosting.

6. *Configurable Reliable Piped Logs*

   You can configure Apache to generate logs in the format that you want. In addition, on most Unix architectures, Apache can send log files to a pipe, allowing for log rotation, hit filtering, real-time splitting of multiple hosts into separate logs, and asynchronous DNS resolving on the fly.

2.14.5 MICROSOFT INTERNET INFORMATION SERVER (IIS)

Microsoft Internet Information Server is a comprehensive solution that will surely help Windows NT take a bite out of the Unix-dominated Web server pie. With perks for both small and large Web sites, IIS comes with powerful extras, including Active Server Pages (ASP) for building dynamic Web pages, Crystal Reports for custom reporting, Microsoft FrontPage for site management, Index Server for advanced searching, and NetShow for on-demand multimedia.

IIS is an extremely capable performer all around, one that would suit any Web site's needs. IIS performed very well serving static pages and handling ISAPI processing on the server side. Depending on the client load, IIS held its own against or outperformed Netscape servers on any platform with static pages. IIS began to outshine its competitors as the load increases to 56 and then 60 clients with static pages, boding well for its scalability.

IIS comes with three default services: WWW, FTP, and Gopher. Its Internet Service Manager (ISM) application controls these services on this or any other IIS server on the network. ISM is run from the Windows NT Server or from a Windows NT or Windows 95 workstation. For remote administration, users can run an HTML version of ISM from a browser. Mapping logical URLs to directories is straightforward, but IIS can't map to a directory on another server, a feature Enterprise handles easily.

2.15 REVIEW ON DEVELOPMENT TOOLS

2.15.1 DREAMWEAVER ULTRADEV 4.0

Dreamweaver UltraDev 4 includes all the great new features of Dreamweaver 4 and is designed to let user quickly connect Web pages to databases, preview live data in the workspace with Live Data Preview, and easily add server-side logic, navigation, and interactivity to a Web pages. It consists of flash buttons, pop-up menus, layout view and code reference - along with an optimal design environment for ASP, JSP and CFML sites. Dreamweaver UltraDev 4 will provide users with critical tools for ensuring the quality of their design if they intend to work on a Web site with dynamic content.

UltraDev is available for Macintosh versions 8.6 and higher, and Windows 95, 98, NT, 2000. While UltraDev performs well on all supported Windows platforms, Macromedia recommends Windows 2000 Professional. UltraDev currently supports ASP, ColdFusion, and JSP servers including Microsoft IIS, Allaire ColdFusion, IBM WebSphere, Allaire JRun, Borland AppServer and Netscape iPlanet. UltraDev can also be extended to support other servers. UltraDev supports any relational database, including Oracle, mySQL, Microsoft SQL Server, and Microsoft Access that users can connect to via ODBC, JDBC, or ADO. [19]

2.15.2 VISUAL INTERDEV 6.0

Microsoft Visual InterDev 6.0 is the latest version of the leading tool for developers building dynamic Web applications. Supporting features like Active Server Pages (ASP), middle-tier components written in any language, dynamic HTML, integrated database design and programming tools make Visual InterDev the ideal tool for building dynamic, data-driven Web sites.

Visual InterDev 6.0 enables developers to build applications accessible from any platform running a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. The Visual InterDev development environment itself runs on Microsoft Windows 95 or Microsoft Windows NT 4.0 or later.

Visual InterDev 6.0 makes it easier to add server components to a project (just drag and drop them in) and deploy projects with components (automatically register components on the client or server). In addition, Visual InterDev supports the creation of COM components out of HTML and script, called Scriptlets, which are programmed just like any other COM component. Visual InterDev will also support high-performance access to virtually any database using Universal Data Access, including ADO, ODBC and OLE DB. [22]

2.15.3 MICROSOFT FRONT PAGE

Microsoft FrontPage is a member of Microsoft Office which designed for non-programmers, yet robust enough even for experienced Web site developers, Microsoft FrontPage is the fast, easy way to create and manage professional-quality Web sites. With features such as WYSIWYG editing and wizards to guide you in creating your Web site, it's never been easier to publish on the Web.

FrontPage also makes it easy for large teams to work together to create and manage sites. Its combination of flexible client/server architecture, passwords, user authentication, and other security features enable contributors in different locations to securely update different pages simultaneously on the same site.

CHAPTER 3

METHODOLOGY AND SYSTEM ANALYSIS

3.1 JUSTIFICATION OF THE PROPOSED METHODOLOGY

3.1.1 INTRODUCTION

The software process is the set of activities and associated results, which produce a software product. Different software processes decompose these activities in different ways. If the wrong process is used, this will probably reduce the quality or the usefulness of the software product to be developed. [10]

In this project, the software development process is based on combination of two software process models, "Waterfall Model" and "Prototyping".

3.1.2 HISTORICAL BACKGROUND OF WATERFALL MODEL

At first, programming software was simple and often done by one person and for engineer or scientific purposes. However, as the use of computers got more widespread, software had to be written for other people than the writers themselves, people with little or no understanding in programming. The old idea of writing a program and then fixing the bugs was no longer sufficient. In 1970 Royce proposed a model for the development of software, derived from a similar model from engineering activities. The notion at the time was that software development was an engineering discipline and that; therefore it would follow a model. This model was warmly greeted and became known as the Waterfall Model. Later it was found that it only worked well for certain classes of software. Thus, new and more complex models were developed. The original model by Royce was also slightly improved and adjusted over time.

3.1.3 WATERFALL MODEL WITH PROTOTYPING

The waterfall model is an engineering model designed to be applied to the development of software. The idea is the following: there are different stages to the development and the outputs of the first stage "flow" into the second stage and these outputs "flow" into the third stage and so on.

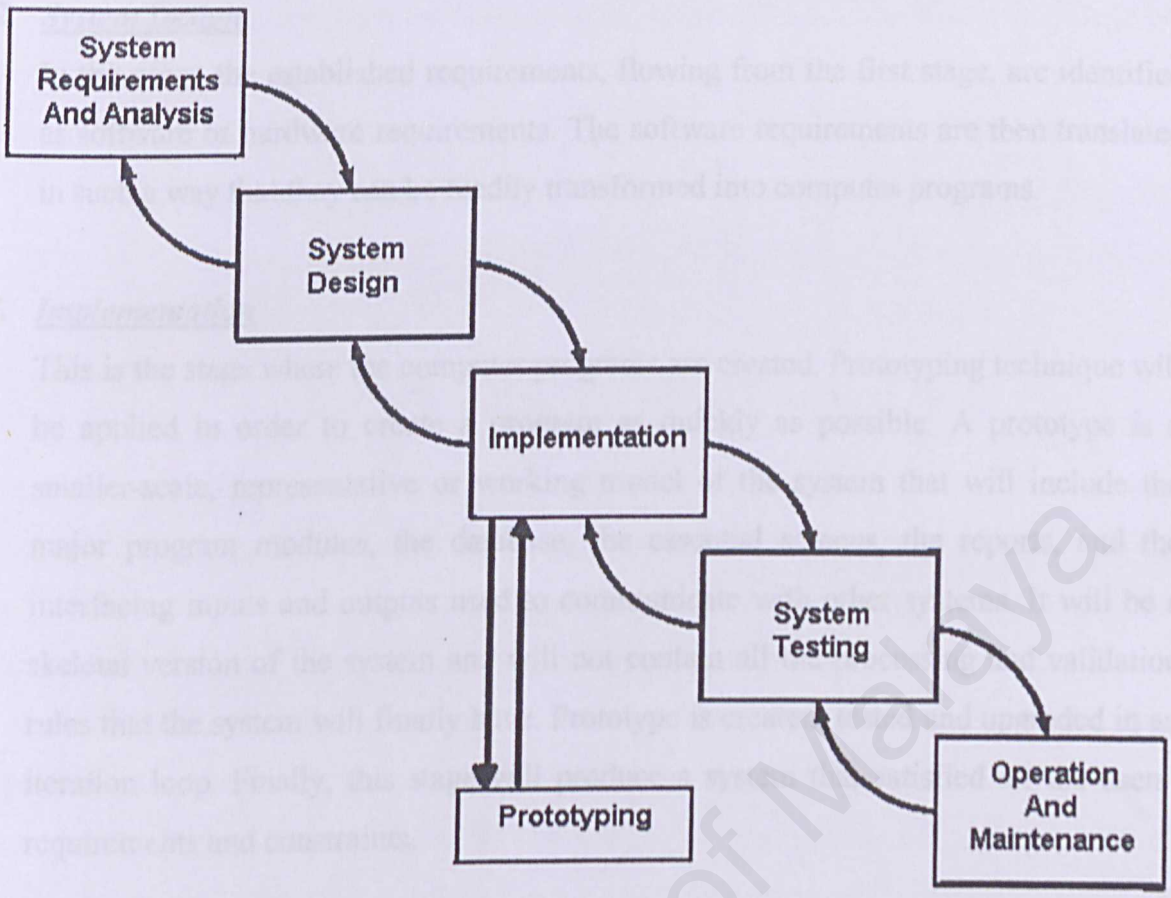


Figure 3.1: Waterfall Model with Prototyping

There are usually five stages in this model of software development:

1. ***System Requirements And Analysis***

   In this stage the requirements of the "to be developed software" are established. These are usually the services it will provide, its constraints and the goals of the software/system. Once these are established they have to be defined in such a way that they are usable in the next stage. This stage is often prelude by a feasibility study or a feasibility study is included in this stage. The feasibility study includes questions like: should we develop the software, what are the alternatives? It could be called the conception of a software product and might be seen as the very beginning of the life cycle.

2. ***System Design***

In this stage the established requirements, flowing from the first stage, are identified as software or hardware requirements. The software requirements are then translated in such a way that they can be readily transformed into computer programs.

3. ***Implementation***

This is the stage where the computer programs are created. Prototyping technique will be applied in order to create a program as quickly as possible. A prototype is a smaller-scale, representative or working model of the system that will include the major program modules, the database, the essential screens, the reports, and the interfacing inputs and outputs used to communicate with other systems. It will be a skeletal version of the system and will not contain all the processing and validation rules that the system will finally have. Prototype is created, tested and upgraded in an iteration loop. Finally, this stage will produce a system that satisfied all the users' requirements and constraints.

4. ***System testing***

The entire system is tested. Information needed is entered into the system for the testing purpose. When the combined programs are successfully tested the software product is finished.

5. ***Operation and maintenance***

Most software products include this stage of the development. It involves correcting errors that have gone undetected before, improvement and other forms of support. This stage is part of the life cycle of a software product, and not of the strict development, although improvements and fixes can still be considered as "development".

These steps are the main stages. There are also sub-stages, within each stage, but they differ from project to project. For example for management purposes the requirements stage is divided in a feasibility study, an outline requirements definition, a design study and a requirements specification stage.

It is also possible that certain software projects require the adding of an extra stage all together, or the splitting of one in two stages. However all the different waterfall models have the same underlying idea; the idea that one stage provides outputs which can be used as the input for the next stage. There thus is a linear flow amongst the stages. The progress of the software development, using the waterfall model, is thus easy to find out. A common way to look at the outputs of a certain stage and see whether or not they are finished in time, thus seeing how far the overall progress is.

There are also activities that are performed at every stage of the software development. These are documentation, verification and management. Documentation is intrinsic to the Waterfall model for it is document driven, as most of the outputs are documents. Verification, not only is a part of implementation & unit testing and system testing, but it is also part of all the other stages in the form of walk through, reviews and the like. Management involves the tailoring of the waterfall model to fit individual processes, managing the human resources (i.e. the people) and managing the rules and the protocol on how the output is formalized, who accesses what and other managing tasks.

Finally it has to be noted that the software development process is not as linear as it seems. When errors in later stages are found, they are often fed back to a previous stage and the development is set back to that stage again. Since this is a managing nightmare, it often occurs that problems are ignored, left for later or programmed around. This feed back makes for a waterfall with information flowing both ways: down through the stages when something is made, and up through the stages when something goes wrong, or feedback is given. Also many processes are frozen when it is not yet the time to deal with them.

3.1.4 STRENGTH OF THE WATERFALL MODEL

The divide and conquer approach of the waterfall process has several advantages like other top-down approaches. It enables organization to track project progress more accurately and discover possible problems at early stages of the product life cycle, especially for large or huge projects. It demands that the process generate a series of documents that can later be used to test and maintain the system. The bottom line of this

approach is to make large software projects more manageable and delivered on time without cost overrun.

In addition, the waterfall model is an important model as it is the basis for many other software development models. The Waterfall model itself can be used for single projects that are cost or time restricted because of the easy way to manage it and cost and time restrictions are often an important issue. Furthermore, this model has a clear define objective to reduce the development and maintenance cost. It is suitable for short-term program or short-life span system.

3.1.5 APPROPRIATENESS OF "WATERFALL MODEL WITH PROTOTYPING" TO THE PROJECT

This methodology is chosen in this project because it is appropriate to this project based on the reasons listed below:

- The project is relatively short (less than one year).
- The scopes and objectives of the project are well understood.
- User requirements of the system are well understood and could be defined easily.
- There is a defined change control procedure.
- End users require the entire system at one time.
- With prototyping in the implementation stage, the complexity of an error is low because the prototype enables the developer to detect any deficiency early at the process.

3.2 JUSTIFICATION OF THE PROPOSED TOOLS

This project intends to produce an automated Course Scheduling System for FSKTM. The system is a web-based application that will be developed using the following tools and technologies:

- Macromedia Dreamweaver UltraDev 4.0
- Microsoft Window 2000 Professional

- Microsoft SQL Server 2000
- Internet Information Server (IIS)
- Active Server Pages (ASP)
- VBScript
- HTML

3.2.1 WHY WEB-BASED APPLICATION?

This project will produce a web-based application because web-based application has many advantages. The advantages are stated as following:

- It allows users to log onto the system from anywhere in the world as long as they have a computer, an Internet connection and a web browser.
- Applications are resident on the server.
- All data input into the system is done so in real-time.
- Data is available in real-time.
- User accessibility to data is definable.
- System administration can be performed remotely.
- Distributed architecture provides for a stable system with feature like database mirroring and redundant servers eliminating any type of system downtime.
- Web-based solutions are based on the web principle of open architecture, which eliminates the integration problem.

3.2.2 MACROMEDIA DREAMWEAVER ULTRADEV 4.0

UltraDev 4.0 is used as the development tools because it has the following features: [19]

- *Easily identify keywords and scripts in code*

  The integrated Text Editor includes ASP, JSP, CFML, JavaScript, and HTML keyword color-coding, auto-indenting, and line numbers.

- *Effortlessly add user authentication Server Behaviors*

  Add password protection to sites without writing complex server-side code—simply validate user names and passwords against a database to password-protect site pages.

- *Easily connect to databases on your Web server*

  With Remote Database Connectivity, there's no need to set up ODBC or JDBC drivers on developer's development machine. UltraDev uses the application server to connect directly to the database.

- *Get total control over source code*

  The users could see both Code and Design Views simultaneously.

3.2.3 MICROSOFT WINDOW 2000 PROFESSIONAL

Windows 2000 Professional has the following features: [27]

- *Windows File Protection*

  Protects core system files from being overwritten by application installs. In the event a file is overwritten, Windows File Protection will replace that file with the correct version. By safeguarding system files in this manner, Windows 2000 mitigates many common system failures found in earlier versions of Windows.

- *Driver certification*

  Provides safeguards to assure you that device drivers have not been tampered with and reducing your risk of installing non-certified drivers.

- *Full 32-bit operating system*

  Minimizes the chance of application failures and unplanned reboots.

- *Microsoft Installer*

  Works with the Windows Installer Service, helping users install, configure, track, upgrade, and remove software programs correctly, minimizing the risk of user error and possible loss of productivity.

- *Windows Logo Program*

  Provides assurance that applications have met a comprehensive set of standards developed by Microsoft in cooperation with customers and third-party developers.

- *Dramatically Reduced Reboot Scenarios*

  Eliminates most scenarios that forced you to reboot in Windows NT 4.0 and Windows 9x. Many software installations also will not require reboots.

3.2.4 MICROSOFT SQL SERVER 2000

SQL Server 2000 is used as the database management system software for this project because of the following features: [23]

- *Easy access to data through the Web*

  With SQL Server 2000, you can use HTTP to send queries to the database, perform full-text search on documents stored in database, and run queries over the Web with natural language.

- *Powerful, flexible Web-based analysis*

  SQL Server 2000 Analysis Services capabilities are extended to the Internet. You can access and manipulate cube data by means of a Web browser.

- *Quick development, debugging, and data transformation*

  SQL Server 2000 features the ability to interactively tune and debug queries, quickly move and transform data from any source, and define and use functions as if they were built in to Transact-SQL. You can visually design and code database applications from any Visual Studio tool.

- *Scale out*

  Scale out distributes the database and data load across servers.

- *Availability*

  SQL Server 2000 achieves maximum availability through enhanced fail over clustering, log shipping, and new backup strategies.

3.2.5 ACTIVE SERVER PAGES (ASP)

Since this system is a web-based application, ASP is used as the programming language. This is because ASP allows developers to create dynamic web sites with database connectivity. ASP supports the use of Internet Information Server in this project. Furthermore, it is easy to learn, especially for those who are familiar with Visual Basic. Besides that, ASP is nice because it is free.

Compare to other Web application tools, ASP allows developers to quickly bring their existing skills and knowledge, data sources, components, and applications to the Web. Other tools create either static HTML or lock developers into a non-standard programming model or language. ASP is based upon the leading industry standards, making it easy to build, maintain, and evolve powerful interactive Web applications.

Active Server Pages supports the use of virtually any scripting language, with native support for VBScript and JScript. It supports ActiveX server components written in any language.

3.3 SYSTEM REQUIREMENTS AND ANALYSIS

3.3.1 FUNCTIONAL REQUIREMENTS

A functional requirement is a description of activities and services a system must provide. It defines what data should be input into the system, where to store inputted data, how the system should react to particular inputs, how the system processes inputted data, how the system should behave in particular situations and what data should be output.

In this Course Scheduling System, it must allows the administrators to input related information into the system. The information includes courses' information, lecturers' information, rooms' information and users' information. All the data will be store in database. Besides that, this system must allow the timetable coordinator to input courses

offered and assigned lecturers to those courses for a corresponding semester. Then the system will generate courses timetable automatically and publish the timetable on the web. Students and lecturers could view and print their timetable through the Internet.

The system could be divided into three modules, which are:

- Administration Module
- Course Scheduling Module
- User Module

3.3.1.1 ADMINISTRATION MODULE

This section should provide functions that allow administrators to manage and update the information storing in the database. The system should provide functions for inputting information about lecturers, courses, rooms and users. All the inputted data will then be stored in the database.

For the security purpose, the administrators need to enter their user ID and password before access is granted. The authentication process will not be continued if there are any errors in entering the user ID and password. The administrators will get a message that prompted them to re-enter their user ID and password correctly. The system also provides the capability for the administrators to change their password when necessary.

Basically, there are five functions in this module.

- Room Information Maintenance
- Lecturer Information Maintenance
- Course Information Maintenance
- User Information Maintenance
- Change Password

Room Information Maintenance

This function should allow the administrators to add, delete, update, search and view the information about rooms in FSKTM. Rooms' information should contain the following information:

Information	Example of Description
Room Name	Dewan Kuliah 1
Room Code	DK1
Room Capacity	200
Room Type	Lecture Hall / Tutorial Room / Lab ?
Room Status	Available for class ? (Y=Yes / N=No)

Table 3.1: Room Information

Lecturer Information Maintenance

This function should allow the administrators to add, delete, update, search and view the information about lecturers in FSKTM. Lecturers' information should contain the following information:

Information	Example of Decription
Lecturer Name	Puan Norizan Mohd Yasin
Lecturer Code	NMY

Table 3.2: Lecturer Information

Course Information Maintenance

This function should allow the administrators to add, delete, update, search and view the general information about courses in FSKTM. Courses' information should contain the following information:

Information	Example of Description
Course Name	Pengaturcaraan Multimedia
Course Code	WMES 3309
Class Type	Lecture / Lab ?
Course Credit Hour	3

Table 3.3: Course Information

User Information Maintenance

This function should allow the administrators to add, delete, update, search and view the information about the system users. Users' information should contain the following information:

Information	Example of Description
User Name	Puan Norizan Mohd Yasin
Post	Timetable Coordinator
User ID	Norizan
Password	*****

Table 3.4: User Information

Change Password

This function should allow the administrators to change their authentication password when necessary. In order to change their password, they must enter their old password correctly. This is the step to ensure that only the valid user can edit the information stored in database.

3.3.1.2 COURSE SCHEDULING MODULE

This is the section that will be used by timetable coordinators. The system should generate timetable automatically.

Similar to the administration section, for the security purpose, the coordinator need to enter their user ID and password before access is granted. The authentication process will not be continued if there are any errors in entering the user ID and password. The coordinators will get a message that prompted them to re-enter their user ID and password correctly. The system also provides the capability for the coordinators to change their password when necessary.

Basically, there are four functions in this module.

- Course Offered Entries
- Generate Timetable

- View Generated Timetable
- Change Password

Course Offered Entries

This function allows the coordinators to input the courses offered for a particular semester. They will assign lecturers for those offered courses. The coordinators could add, delete, update, search or view the offered courses' information. Courses offered entries should contain the following information:

Information	**Example of Description**
Course Code	WXES 1101
Course Volume	350 (Number of students)
Lab Class	Yes / No ?
Lab Hours	3
Lecturer Assigned	Dr. Roziati
Course Type	Core / Elective
Student Type	KB1 / P3 / MCS
No of Weeks	14 Weeks / 7Weeks
Assigned After 6pm?	Yes / No
Concurrent	WXET 1101
Faculty Paper	Yes / No

Table 3.5: Course Offered Information

If the course is a non-faculty course, it should contains more information as follow:

Room / Place	K1, FPP
Day	Monday
Time	10.00 – 11.50 am

Table 3.6: Course Offered Information for Non-faculty Course

Generate Timetable

This function will generate timetable on user's request. The system should generate timetable automatically. The timetable generated must satisfy all the timetabling hard constraints such as room capacity and the crash of courses, lecturers and rooms.

View Generated Timetable

This function allows the coordinator to view and check the timetable after the system generates the timetable.

Change Password

This function should allow the coordinator to change their authentication password when necessary. In order to change their password, they must enter their old password correctly. This is the step to ensure that only the valid user can edit the information stored in database.

3.3.1.3 USER MODULE

The system should allow the users to view their timetable through the Internet. This section contains the functions that allow the lecturers and students to view and print timetable in various formats.

View Overall Timetable

This function allows the lecturers and students to view and print overall timetable. The timetable for all programs in FSKTM such as Computer Science first year, PJJ or Diploma could be viewed.

View Timetable By Program / Year

This function allows the users to view and print timetable for particular program. The users must specify the program they want to view such as Computer Science third year.

View Timetable By Course

This function allows the users to view and print timetable for particular course. The users must enter the specific course code such as WXES 3204.

View Timetable By Room

This function allows the users to view and print timetable for particular room. The users must enter the specific room code such as DK2.

View Timetable By Lecturer

This function allows the lecturer to view and print their timetable. Entering the code of the lecturer will display a timetable that shows the lecturer's teaching hours.

3.3.2 NON-FUNCTIONAL REQUIREMENTS

A non-functional requirement is a description of other features, characteristics, and constraints that define a satisfactory system. These are constraints on the services or functions offered by the system. Examples of non-functional requirements include performance (throughput and response time), ease of learning and use, budgets, costs, and cost savings, timetables and deadline, documentation and training needs, quality management, and security. [11]

In order to develop a satisfactory Course Scheduling System, this system should have the characteristics of reliability, usability, security, availability and manageability.

3.3.2.1 RELIABILITY

The system shall be reliable in performing its functions and operations and shall not cause unnecessary and unplanned downtime of the overall environment. In this system, if the information provided across the Internet is not reliable, it might cause lecturers or students to miss their session.

3.3.2.2 USABILITY

The system will be easy to use because users only have to click on the hyperlink or image by using the mouse. Interface shall be intuitive and consistent with other modules in this system. The use of suitable and meaningful icons will help the users to use the system with more easily.

3.3.2.3 SECURITY

Only the authenticated users like timetable coordinator or administrator could have the access rights to modify or update the date in the database. The authorized users are allowed to change their password if they choose to do so.

3.3.2.4 AVAILABILITY AND MANAGEABILITY

The system shall be available to lecturers and students to ensure smooth operator and management services at any point of time. The system shall be capable of being managed and easy to operate, This will make the maintenance and enhancements work simpler and not too time consuming.

3.3.3 RUN-TIME ENVIRONMENT

The run time environment consists of hardware and software configuration.

3.3.3.1 SERVER SIDE

Hardware Requirements

- A server with at least Pentium 166 Mhz processor
- At least 64MB RAM
- Network Interface Card (NIC) and network connection

Software Requirements

- Windows 2000 Professional
- Internet Information Server 3.0 or higher
- Microsoft SQL Server 2000

3.3.3.2 CLIENT SIDE

Hardware Requirements

- At least 16MB of RAM
- Network connection through existing network configuration/LAN or via modem

Software Requirements

- Any Web Browsers such as Netscape Communicator
- Microsoft Internet Explorer 3.01 or above (recommended)

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE DESIGN

FSKTM Course Scheduling System is formed by three interrelated subsystems as shown in Figure 4.1. They are:

- Interface Subsystem
- Database Subsystem
- Scheduling Subsystem

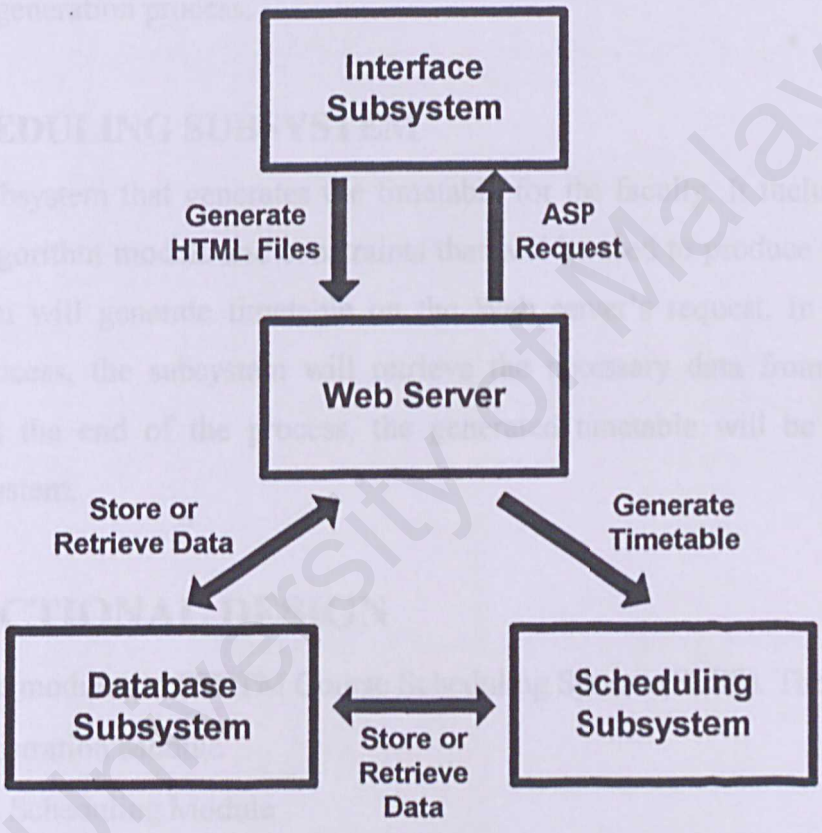


Figure 4.1: System Architecture

4.1.1 INTERFACE SUBSYSTEM

This is the subsystem that allows end users to interact with the system. As the subsystem is resides on the client side, it could be accessed from any workstation or PC that have Internet connection. This subsystem is designed to allow users to input and manage data stored in database subsystem, request the system to generate timetable and view or print timetable in different format.

When the users perform some actions on this subsystem, it will generate ASP requests that ask the Web server to react to the users action. Then, the Web server will generate corresponding HTML files and display web pages to the users.

4.1.2 DATABASE SUBSYSTEM

Database subsystem is resides in the Web server. The database is used to store all the data needed by the system. Web server has the responsibility to store or retrieve data from the database subsystem. This subsystem will also interact with the scheduling subsystem in the timetable generation process.

4.1.3 SCHEDULING SUBSYSTEM

This is the subsystem that generates the timetable for the faculty. It includes all of the timetabling algorithm models and constraints that will be used to produce the schedules. The subsystem will generate timetable on the Web server's request. In the timetable generation process, the subsystem will retrieve the necessary data from the database subsystem. At the end of the process, the generated timetable will be stored in the database subsystem.

4.2 FUNCTIONAL DESIGN

There are three modules in FSKTM Course Scheduling System (FCSS). They are:

- Administration Module
- Course Scheduling Module
- User Module

Figure 4.2 shows the overall modules in FCSS.

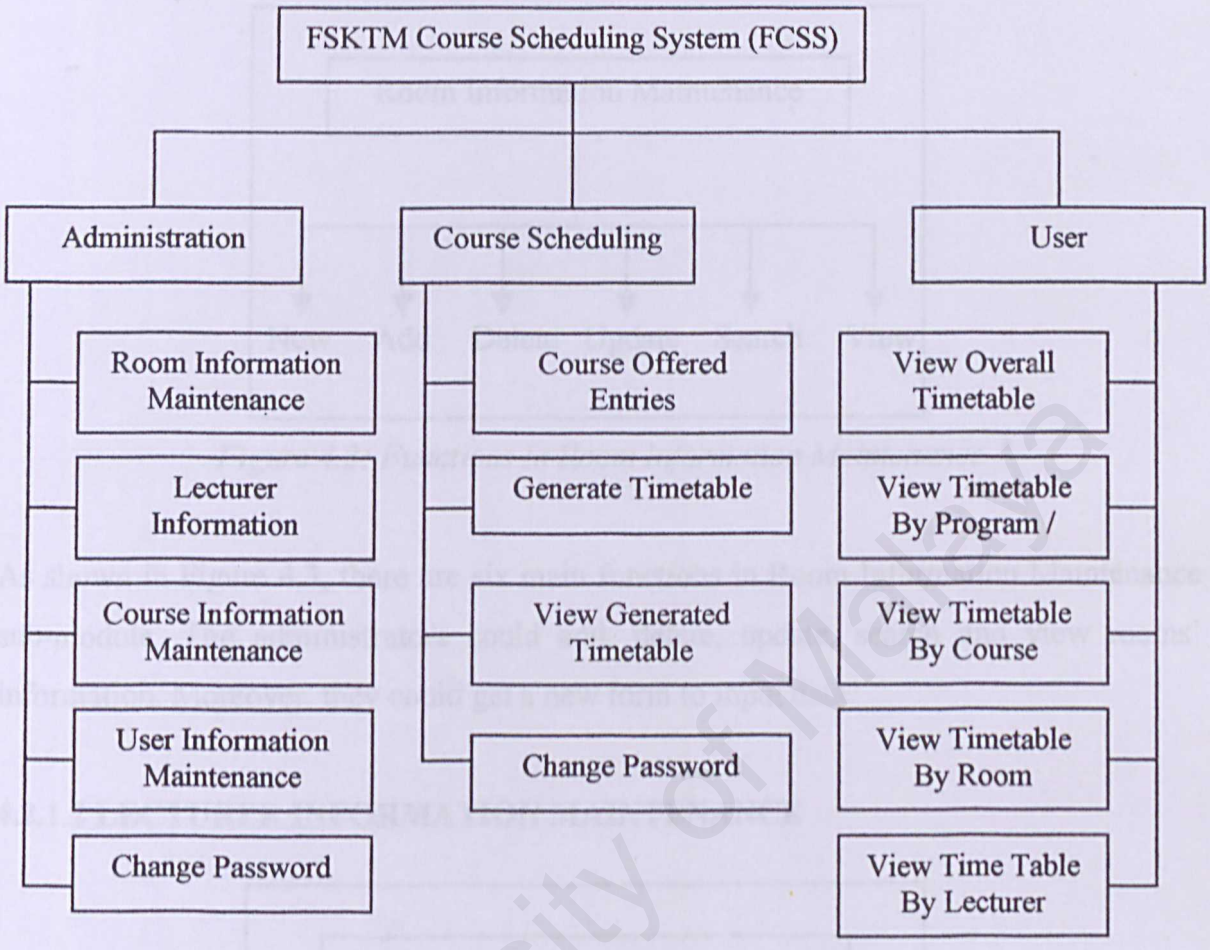


Figure 4.2: Modules in FSKTM Course Scheduling System

4.2.1 ADMINISTRATION MODULE

Administrators used this module to manage the data stored in the system. They need to enter their user ID and password in order to access this module. Basically, there are five sub-modules in this module:

- Room Information Maintenance
- Lecturer Information Maintenance
- Course Information Maintenance
- User Information Maintenance
- Change Password

4.2.1.1 ROOM INFORMATION MAINTENANCE

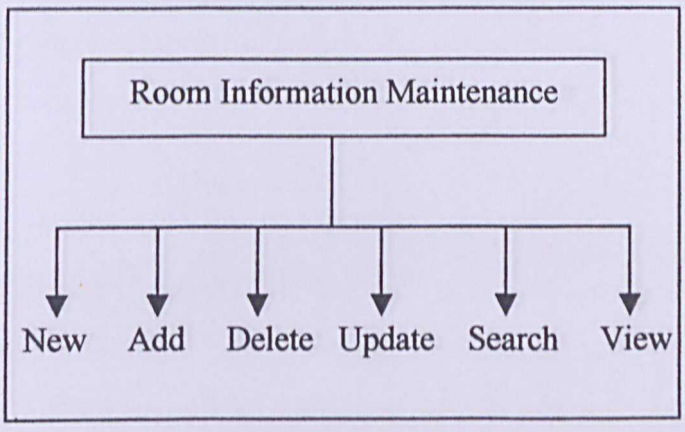


Figure 4.3: Functions In Room Information Maintenance

As shown in Figure 4.3, there are six main functions in Room Information Maintenance sub-module. The administrators could add, delete, update, search and view rooms' information. Moreover, they could get a new form to input data.

4.2.1.2 LECTURER INFORMATION MAINTENANCE

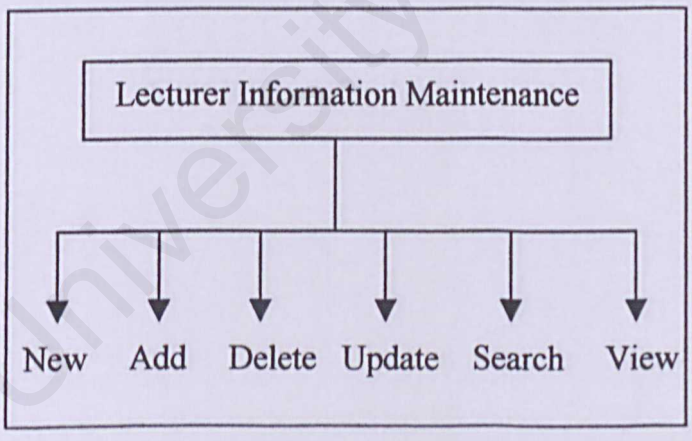


Figure 4.4: Functions In Lecturer Information Maintenance

As shown in Figure 4.4, there are six main functions in Lecturer Information Maintenance sub-module. The administrators could add, delete, update, search and view lecturers' information. Moreover, they could get a new form to input data.

4.2.1.3 COURSE INFORMATION MAINTENANCE

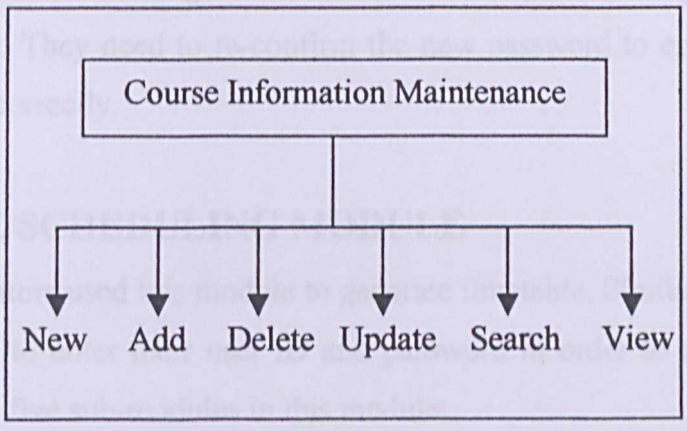


Figure 4.5: Functions In Course Information Maintenance

As shown in Figure 4.5, there are six main functions in Course Information Maintenance sub-module. The administrators could add, delete, update, search and view courses' information. Moreover, they could get a new form to input data.

4.2.1.4 USER INFORMATION MAINTENANCE

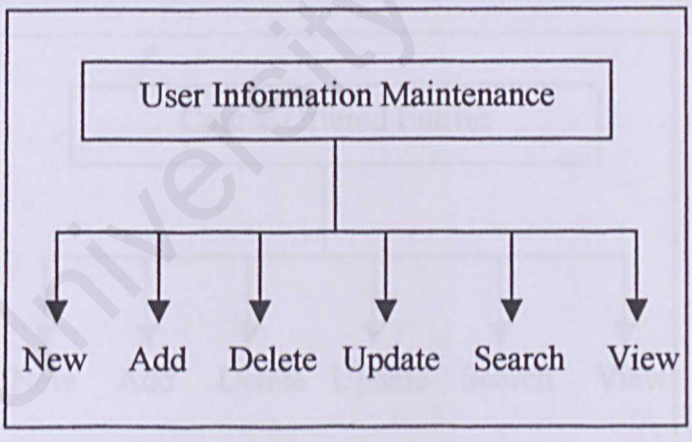


Figure 4.6: Functions In User Information Maintenance

As shown in Figure 4.6, there are six main functions in User Information Maintenance sub-module. The administrators could add, delete, update, search and view users' information. Moreover, they could get a new form to input data.

4.2.1.5 CHANGE PASSWORD

The administrators could change their authentication password by entering the old password correctly. They need to re-confirm the new password to ensure that they type the new password correctly.

4.2.2 COURSE SCHEDULING MODULE

Timetable coordinators used this module to generate timetable. Similar to Administration module, they need to enter their user ID and password in order to access this module. Basically, there are five sub-modules in this module:

- Course Offered Entries
- Generate Timetable
- View Generated Timetable
- Change Password

4.2.2.1 COURSE OFFERED ENTRIES

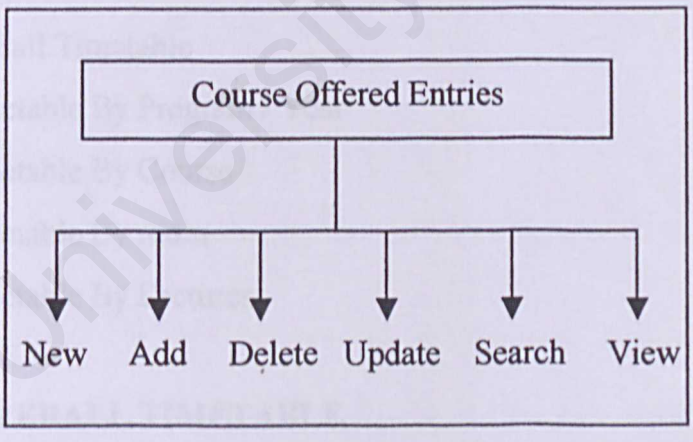


Figure 4.7: Functions In Course Offered Entries

As shown in Figure 4.7, there are six main functions in Course Offered Entries sub-module. The timetable coordinators could add, delete, update, search and view courses offered information. Moreover, they could get a new form to input data. This sub-module also allows the coordinators to assign lecturer for those offered courses.

4.2.2.2 GENERATE TIMETABLE

Timetable coordinator could generate timetable automatically after inputting the courses offered information into the system. The timetable engine will generate the timetable automatically.

4.2.2.3 VIEW GENERATED TIMETABLE

Timetable coordinators could view and check the generated timetable.

4.2.2.4 CHANGE PASSWORD

The coordinators could change their authentication password by entering the old password correctly. They need to re-confirm the new password to ensure that they type the new password correctly.

4.2.3 USER MODULE

Students and lectures used this module to view and print their personalized timetable. They could access this module directly through the Internet without any authentication process. Basically, there are five sub-modules in this module:

- View Overall Timetable
- View Timetable By Program / Year
- View Timetable By Course
- View Timetable By room
- View Timetable By Lecturer

4.2.3.1 VIEW OVERALL TIMETABLE

Students and lecturers could view overall timetable where the timetable will include all the programs offered in FSKTM. These programs are Computer Science first year, second year, third year, Information Technology first year, second year, third year, Diploma, Master and PJJ program. Each time slot in the timetable contains three types of information, which are Course Code, Room Code and Lecturer Code. The details of these codes such as Course Name will be shown in the timetable in order for the users to view the timetable in a more convenient way.

4.2.3.2 VIEW TIMETABLE BY PROGRAM / YEAR

Students and lecturers could view the timetable by choosing the program such as Computer Science first year. Each of the time slots in this timetable also contains three types of information, which are Course Code, Room Code and Lecturer Code. The details of these codes will be shown.

4.2.3.3 VIEW TIMETABLE BY COURSE

By entering the specific course code, students and lecturers could view the timetable for that particular course. The time slots, rooms and lecturers for this course will be shown.

4.2.3.4 VIEW TIMETABLE BY ROOM

By entering the specific room code, students and lecturers could view the timetable for that particular room. The time slots, courses at particular time slot and lecturers for these courses will be shown.

4.2.3.5 VIEW TIMETABLE BY LECTURER

By entering the specific lecturer code, users could view the timetable for that particular lecturer. The time slots, courses and rooms at particular time slot will be shown.

4.3 USER INTERFACE DESIGN

Normally, developers will face with two key issues when design the user interface for a computer system:

- How can information from the user be provided to the computer system?
- How can information from the computer system be presented to the user?

This system is designed to have Graphical User Interfaces (GUIs). GUIs support high-resolution color screens and interaction using a mouse as well as a keyboard. The advantages of GUIs are [12]:

- They are relatively easy to learn and use. Users with no computing experience can learn to use the interface after a brief training session.

- The user has multiple screens (windows) for system interaction.
- Fast, full-screen interaction is possible with immediate access to anywhere on the screen.

4.3.1 DESIGN PRINCIPLES

The following general principles were applied in the user interface design for this system:

- *User Familiarity*

  Users should not be forced to adapt to an unfamiliar interface. For example, a system is designed by using familiar words such as "edit", "add", "delete" and so on.

- *Consistency*

  System commands and menus should have the same format. Parameters should be passed to all commands in the same way, and command punctuation should be similar.

- *Confirmation*

  If a user specifies an action, which is potentially destructive, he or she should be asked to confirm that this is really what is intended before any information is destroyed. For instance, deletion of courses information in administration module will be asked to confirm the deletion.

- *Meaningful Error Messages*

  The error messages should describe the problem in a language that the user understands.

- *Robustness*

  The system should be able to protect itself from the user errors that might cause it to fail. For example, the system is able to recover and display an error message when the user specifies a non-numeric value for the numeric input field.

4.3.2 CHARACTERISTICS OF USER INTERFACE

Some of the characteristics of the user interface design for this system are:

- *Buttons*

  Picking a button causes a single action to be initiated.

- *Text Field*

  The user inputs a specific piece of data / information within a form.

- *Radio Button / Drop Down List*

  Provides some choices for the user to select.

4.3.3 EXAMPLES OF USER INTERFACE DESIGN

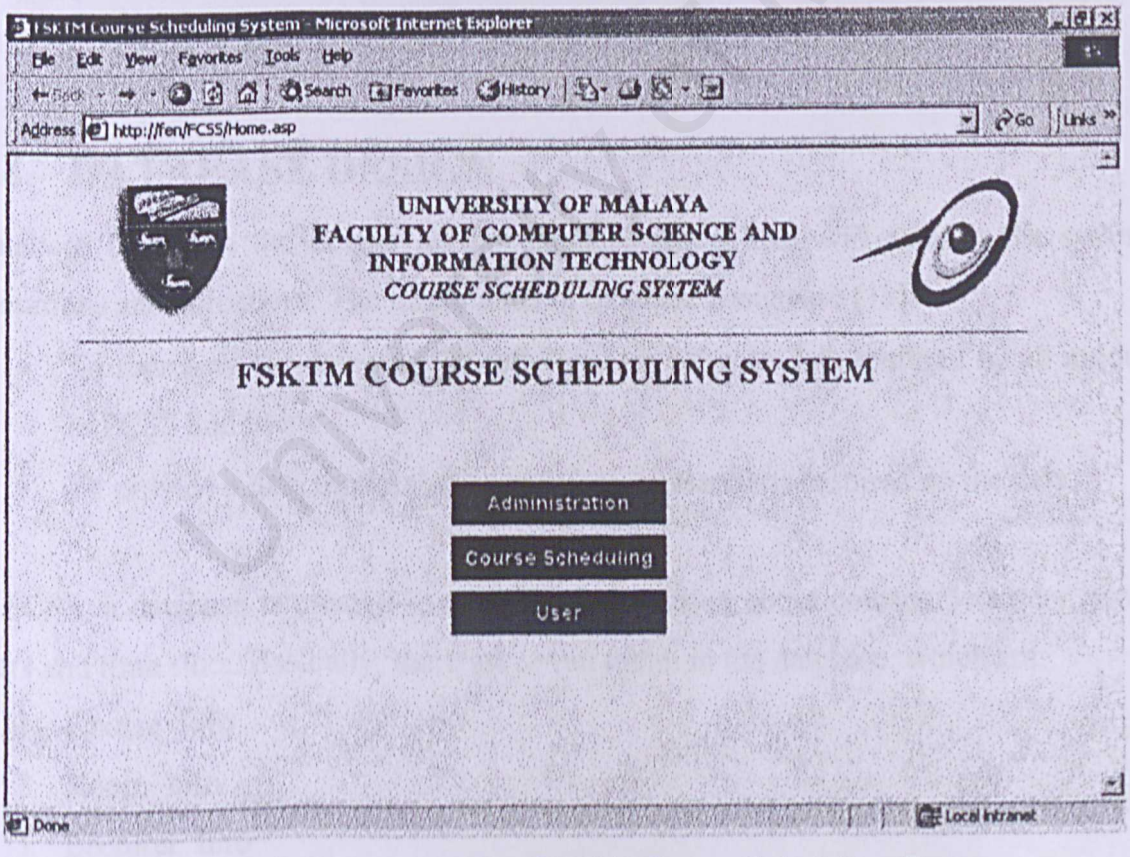


Figure 4.8: Main Page For FSKTM Course Scheduling System

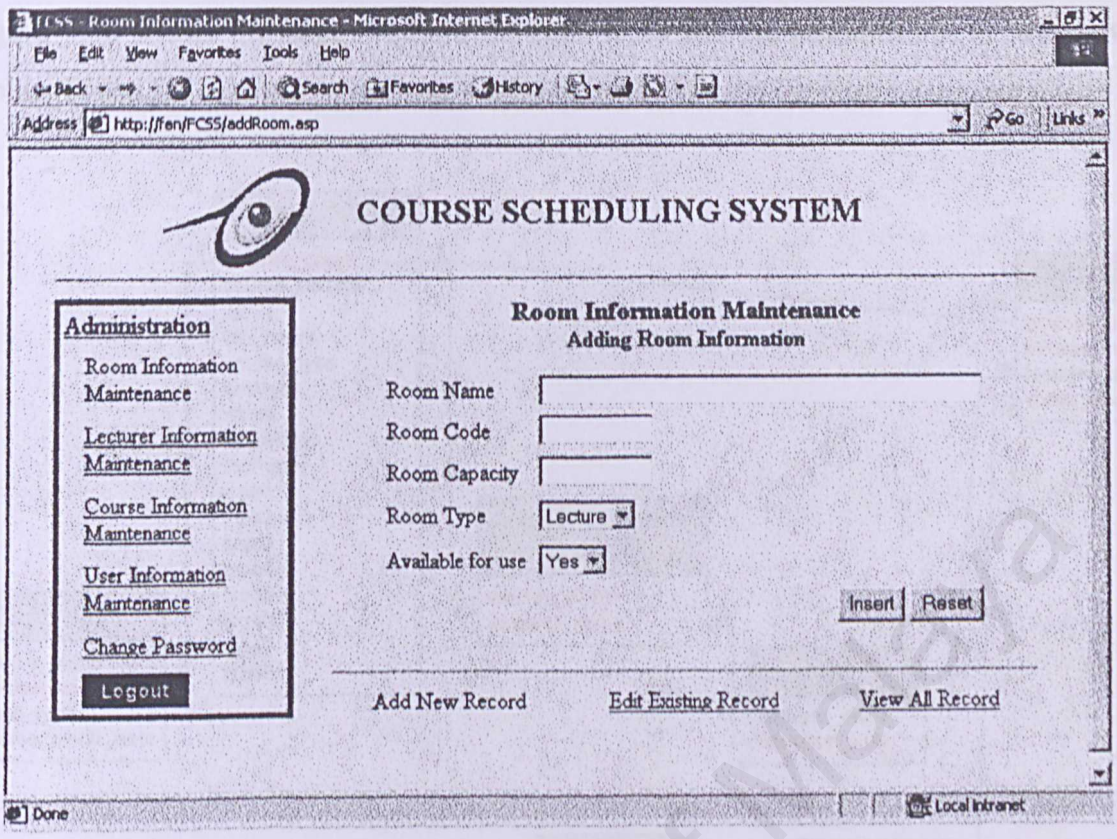


Figure 4.9: Form Of Room Information Maintenance

4.4 DATABASE DESIGN

Database design is the design of the database model that will support the system's operations and objectives. The major aims of database design are [13]:

- To represent the data and the relationships between data required by all modules in FCSS and users.
- To provide a data model that support any transactions required on the data.

In FCSS, a database is created to store the information about courses, lecturers, rooms, users and time slots. Basically, there are seven tables in the database, which are:

1. Course_Info
2. Room_Info
3. Lecturer_Info
4. Course_Offered
5. Course_Lecturer_Assigned
6. Time_Slot
7. User_Login

Figure 4.10 shows the data structure for each table and the relationships between those tables. Those field names in red color is the primary key for the particular table.

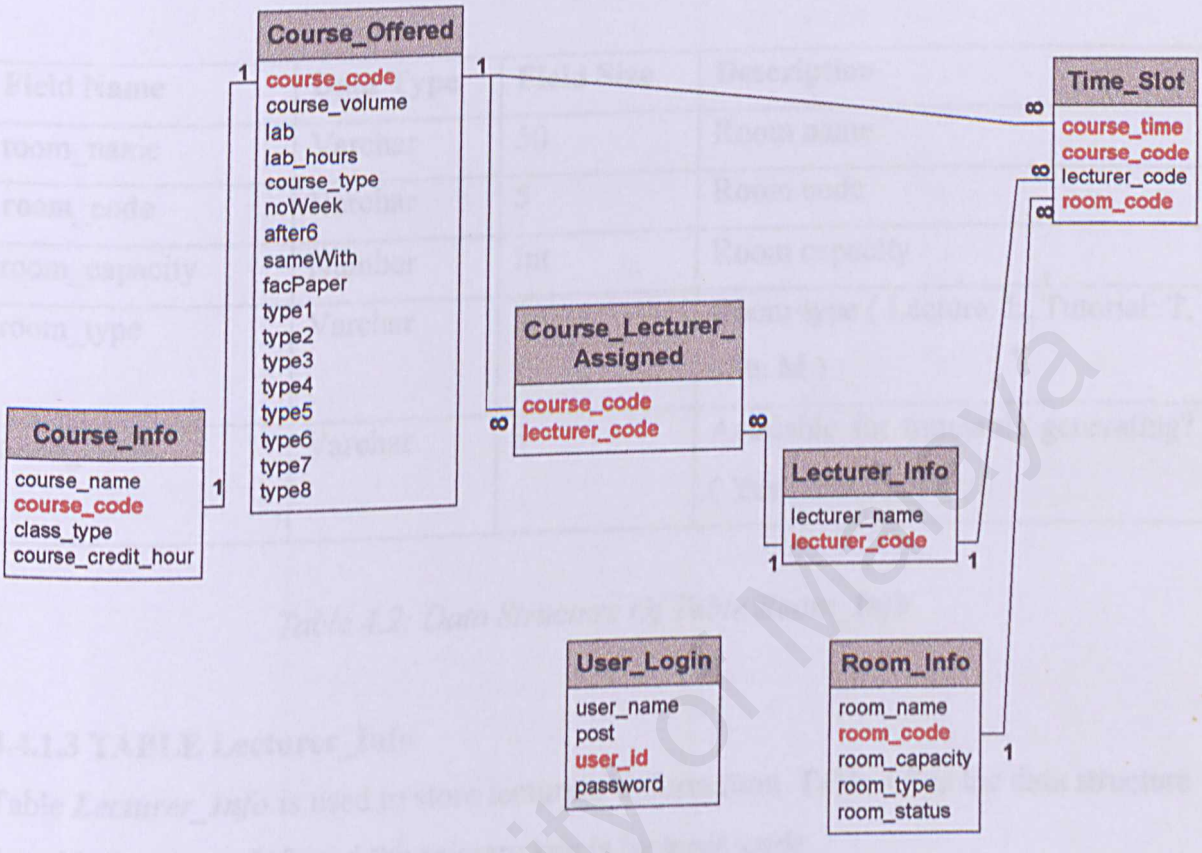


Figure 4.10: Database's Tables And Relationships

4.4.1 DATA STRUCTURE

4.4.1.1 TABLE Course_Info

Table ***Course_Info*** is used to store courses' information. Table 4.1 is the data structure for table ***Course_Info*** and the primary key is ***course_code***.

Field Name	Data Type	Field Size	Description
course_name	Varchar	50	Course name
course_code	Varchar	8	Course code
class_type	Varchar	3	Lecture (L) or Lab (M)
course_credit_hour	Number	Int	Course credit hour

*Table 4.1: Data Structure Of Table **Course_Info***

4.4.1.2 TABLE Room_Info

Table ***Room_Info*** is used to store rooms' information. Table 4.2 is the data structure for table ***Room_Info*** and the primary key is ***room_code***.

Field Name	Data Type	Field Size	Description
room_name	Varchar	50	Room name
room_code	Varchar	5	Room code
room_capacity	Number	Int	Room capacity
room_type	Varchar	3	Room type (Lecture: L, Tutorial: T, Lab: M)
room_status	Varchar	3	Available for timetable generating? (Yes: Y or No: N)

*Table 4.2: Data Structure Of Table **Room_Info***

4.4.1.3 TABLE Lecturer_Info

Table ***Lecturer_Info*** is used to store lecturers' information. Table 4.3 is the data structure for table ***Lecturer_Info*** and the primary key is ***lecturer_code***.

Field Name	Data Type	Field Size	Description
lecturer_name	Varchar	50	Lecturer name
lecturer_code	Varchar	5	Lecturer code

*Table 4.3: Data Structure Of Table **Lecturer_Info***

4.4.1.4 TABLE Course_Offered

Table ***Course_Offered*** is used to store courses offered information. Table 4.4 is the data structure for table ***Course_Offered*** and the primary key is ***course_code***.

Field Name	Data Type	Field Size	Description
course_code	Varchar	8	Course code
course_volume	Number	Int	Number of students
lab	Varchar	3	Is there any lab class? (Yes: Y or No: N)
lab_hours	Number	Int	Lab class hours
course_type	Varchar	3	C: Core or E: Elective
noWeek	Varchar	10	14 Weeks or 7 Weeks
after6	Varchar	3	Assigned after 6pm? (Yes: Y or No: N)
sameWith	Varchar	10	Concurrent course code
facPaper	Varchar	3	Faculty course or Non-faculty course (Yes: Y or No: N)
type1	Varchar	10	Student Type 1 (Eg: KB1, SPM3)
type2	Varchar	10	Student Type 2 (Eg: KB1, SPM3)
type3	Varchar	10	Student Type 3 (Eg: KB1, SPM3)
type4	Varchar	10	Student Type 4 (Eg: KB1, SPM3)
type5	Varchar	10	Student Type 5 (Eg: KB1, SPM3)
type6	Varchar	10	Student Type 6 (Eg: KB1, SPM3)
type7	Varchar	10	Student Type 7 (Eg: KB1, SPM3)
type8	Varchar	10	Student Type 8 (Eg: KB1, SPM3)

Table 4.4: Data Structure Of Table ***Course_Offered***

4.4.1.5 TABLE Course_Lecturer_Assigned

Table ***Course_Lecturer_Assigned*** is used to store information about lecturers assigned to particular course. Table 4.5 is the data structure for table ***Course_Lecturer_Assigned*** and the primary keys are combination of ***course_code*** and ***lecturer_code***.

Field Name	Data Type	Field Size	Description
course_code	Varchar	8	Course code
lecturer_code	Varchar	5	Lecturer code

*Table 4.5: Data Structure Of Table **Course_Lecturer_Assigned***

4.4.1.6 TABLE Time_Slot

Table ***Time_Slot*** is used to store time slots information for particular course. Table 4.6 is the data structure for table ***Time_Slot*** and the primary keys are combination of ***course_time***, ***course_code*** and ***room_code***.

Field Name	Data Type	Field Size	Description
Course_time	Number	Int	Time slot eg: Monday 8am: 1 Monday 9am: 2
Course_code	Varchar	8	Course code in this slot
Lecturer_code	Varchar	5	Lecturer code I this slot
room_code	Varchar	5	Room code in this slot

*Table 4.6: Data Structure Of Table **Time_Slot***

4.4.1.7 TABLE User_Login

Table ***User_Login*** is used to store users' information. Table 4.7 is the data structure for table ***User_Login*** and the primary key is ***user_id***.

Field Name	Data Type	Field Size	Description
user_name	Varchar	50	Users' full name
post	Varchar	20	Users' post
user_id	Varchar	10	ID used to identified the particular user
password	Varchar	10	Password

*Table 4.7: Data Structure Of Table **User_Login***

4.4.2 RELATIONSHIPS

As shown in Figure 4.11, there are two types of relationships between all tables in the database. The relationships are one-to-one relationship and one-to-many relationship. The following section describes the relationships between each table.

- *Table **Course_Info** And Table **Course_Offered***
  Relationship between them is one-to-one, where an offered course might has only one course details.

- *Table **Course_Offered** And Table **Course_Lecturer_Assigned***
  Relationship between them is one-to-many, where one course might teach by one or more lecturers.

- *Table **Course_Offered** And Table **Time_Slot***
  Relationship between them is one-to-many, where an offered course might be taught at more than one time slot.

- *Table **Lecturer_Info** And Table **Course_Lecturer_Assigned***
  Relationship between them is one-to-many, where one lecturer might teach more than one course.

- *Table **Lecturer_Info** And Table **Time_Slot***
  Relationship between them is one-to-many, where one lecturer might have many teaching time slots.

- *Table **Room_Info** And Table **Time_Slot***
  Relationship between them is one-to-many, where one room might be assigned to many different time slots.

4.5 DATA FLOW DESIGN

A Data Flow Diagram (DFD) is a tool that depicts the flow of data through a system and the work or processing performed by that system. Synonyms include bubble chart, transformation graph, and process model. [11]

DFD is used to represent the input, output data and processes in this system. DFD shows how the data flow into the system, how they are transformed and how they leave the system. [12]

Table 4.8 describes symbols used in Data Flow Diagram.

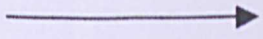
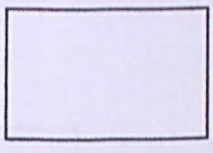
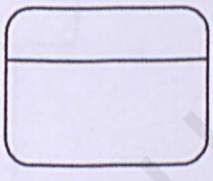
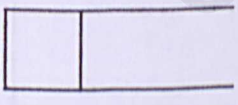
Symbol	Name	Description
	Data Flow	The direction data flow. It represents an input of data to a process or the output of data (or information) from a process.
	Source / External Agent	External source interact with the system but outside the system.
	Process	Function that transform input to output.
	Data Store	A place where multiple data elements are stored. It is an "inventory" of data.

Table 4.8: Data Flow Diagram Symbol

The following section shows the DFD design for the FSKTM Course Scheduling System.

4.5.1 CONTEXT LEVEL DFD

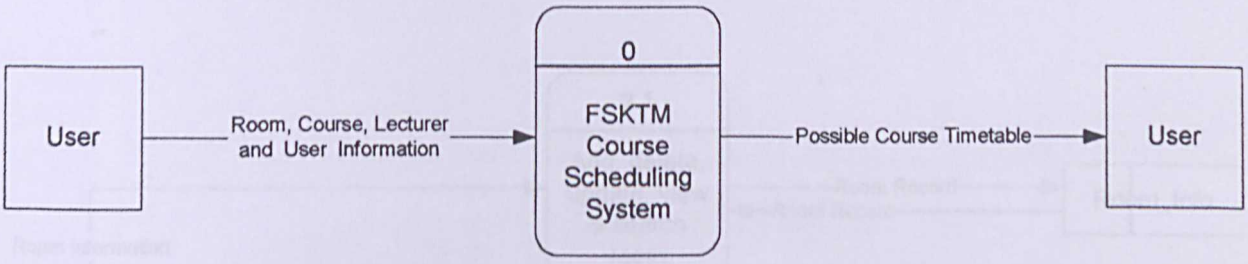


Figure 4.11: Context Level DFD

4.5.2 DIAGRAM 0 DFD

Diagram 0 DFD shows the overall processes for this system.

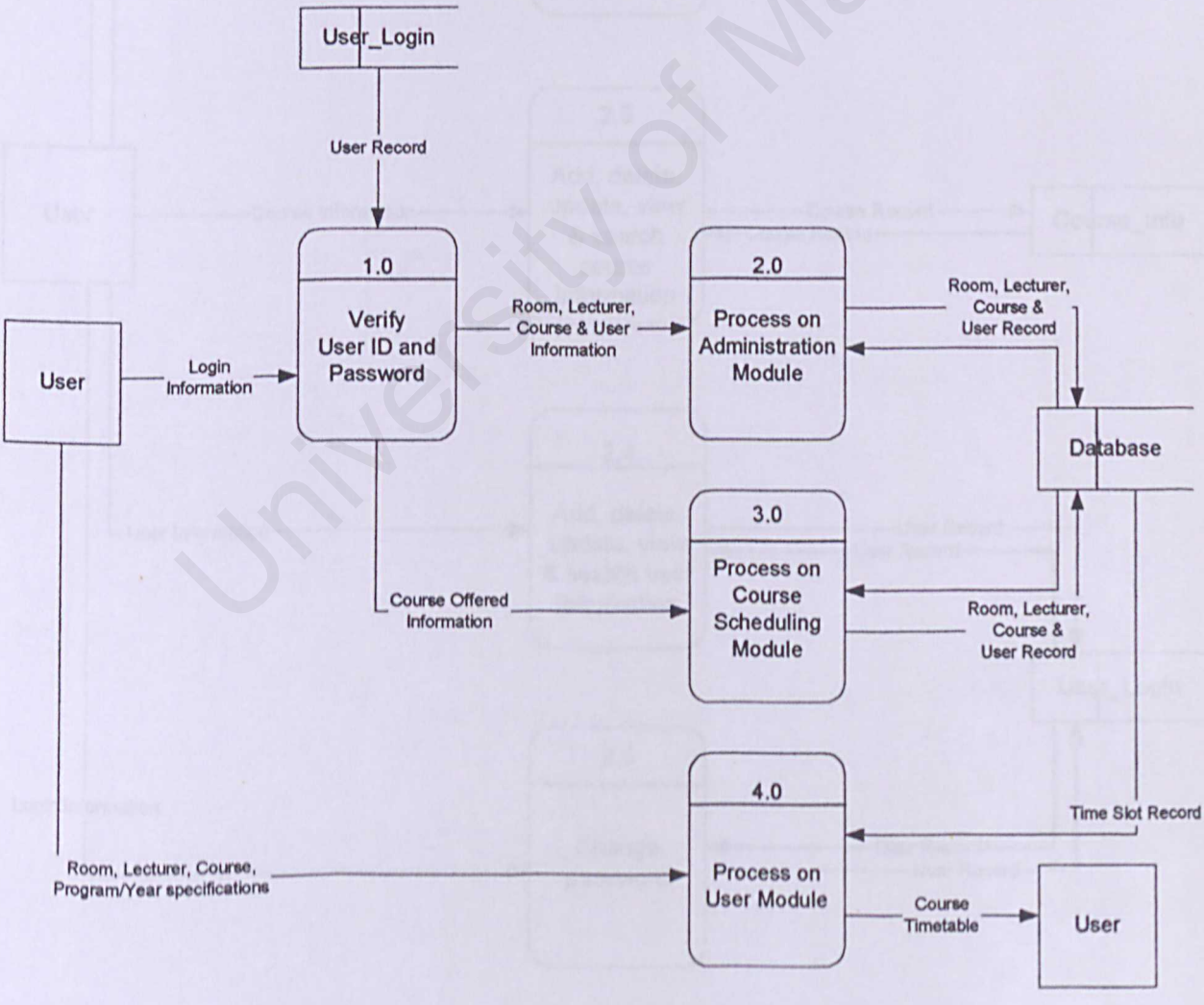


Figure 4.12: Diagram 0 DFD

4.5.3 DIAGRAM 2 DFD

Diagram 2 DFD shows the processes for the Administration Module.

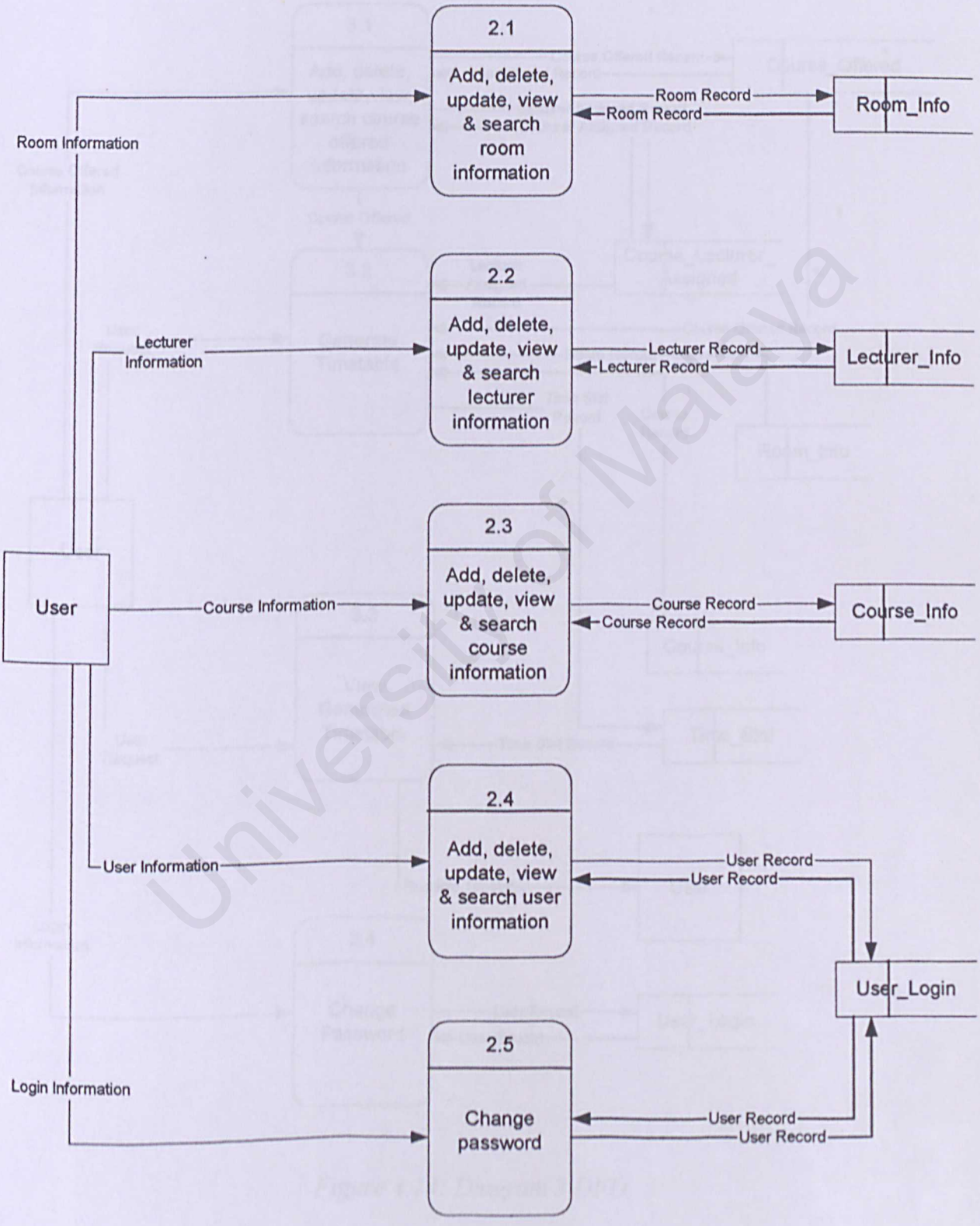


Figure 4.13: Diagram 2 DFD

4.5.4 DIAGRAM 3 DFD

Diagram 3 DFD shows the processes for the Course Scheduling Module.

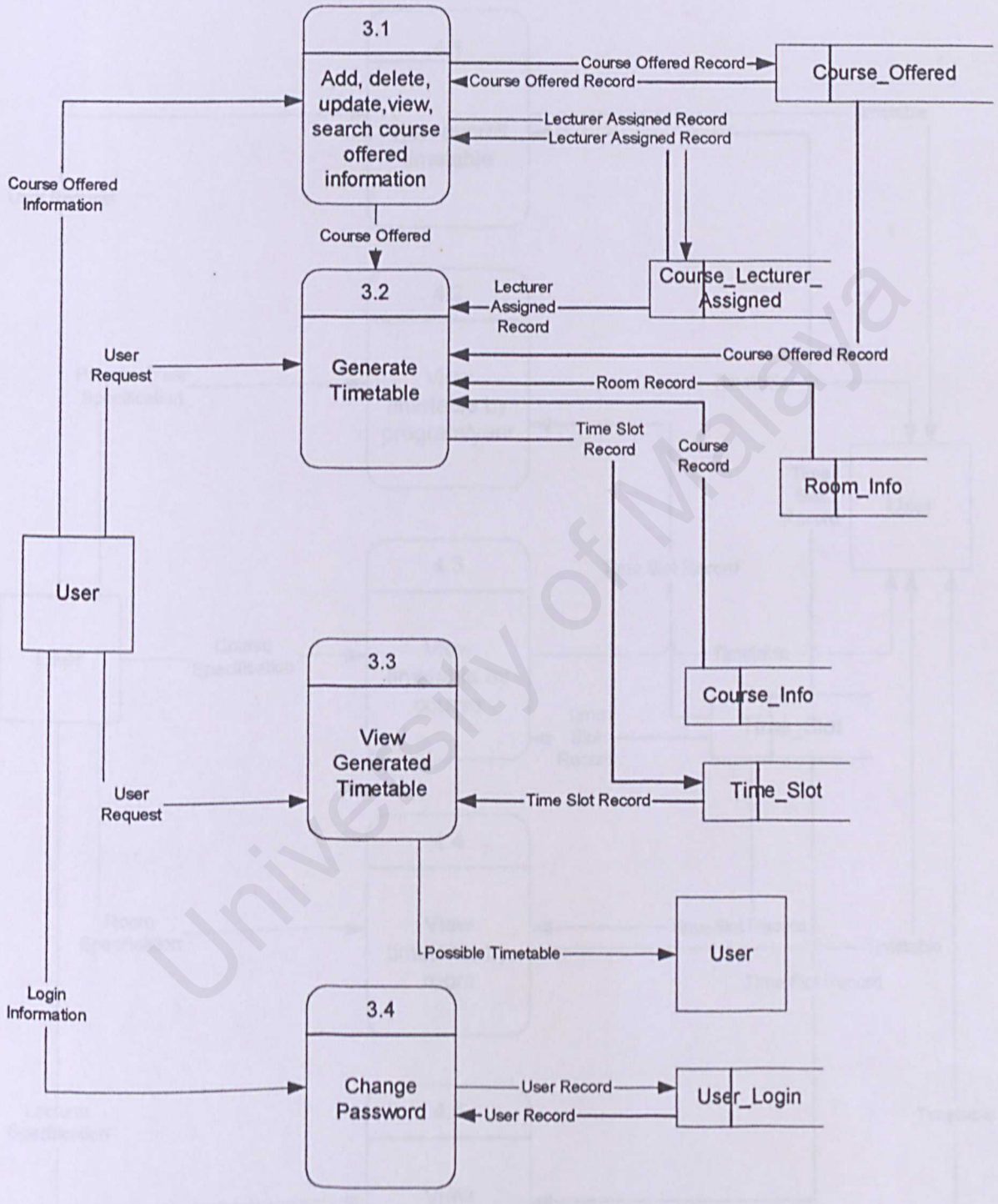


Figure 4.14: Diagram 3 DFD

4.5.5 DIAGRAM 4 DFD

Diagram 4 DFD shows the processes for the User Module.

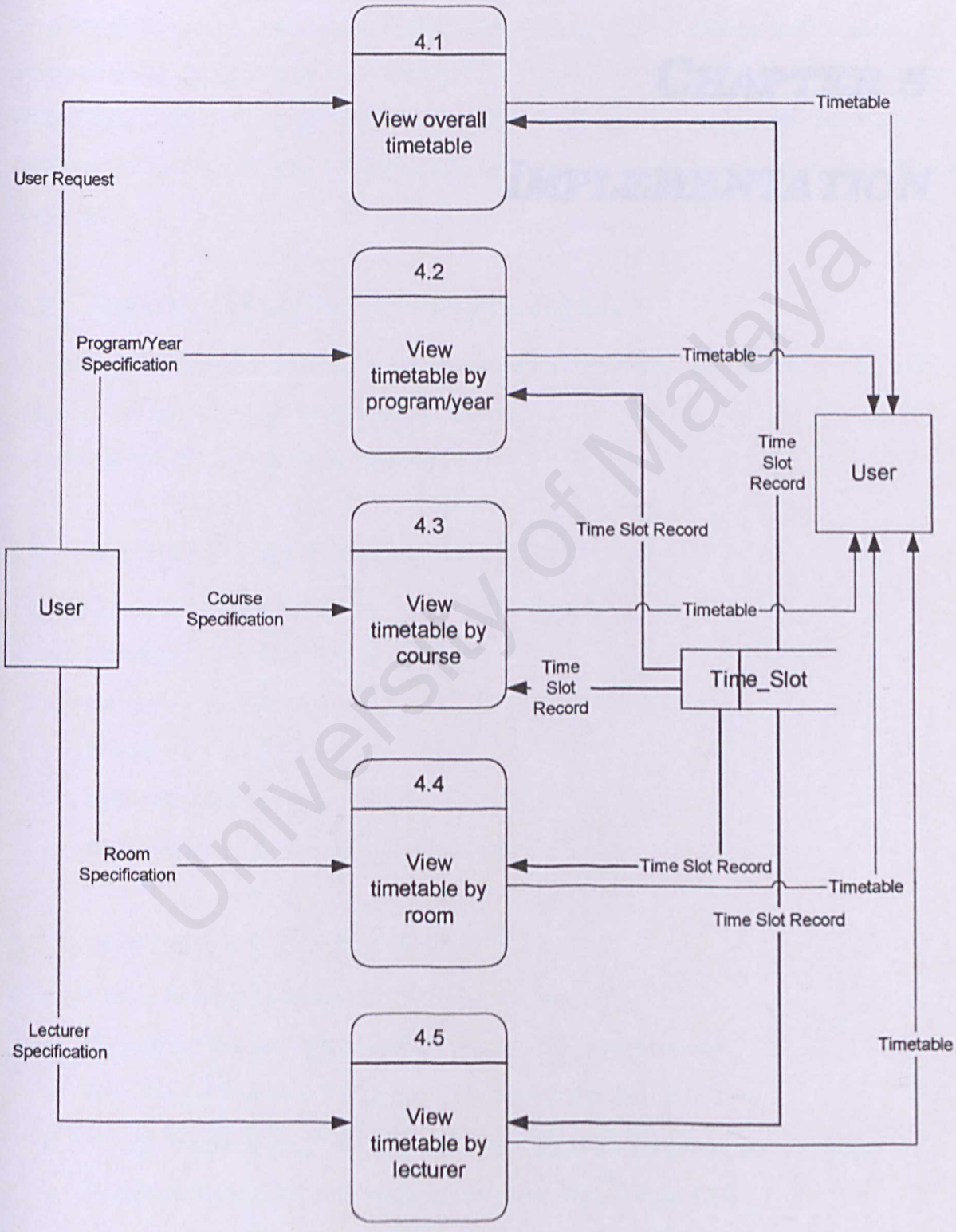


Figure 4.15: Diagram 4 DFD

CHAPTER 5

IMPLEMENTATION

5.1 INTRODUCTION

During this stage, the software design is realized as a set of program [10]. The design is implemented in the most appropriate languages. In FSKTM Course Scheduling System, Web-based interfaces are implemented using HTML. The database implementation is achieved using the Microsoft SQL Server 2000. Parts of the applications are the database transactions, which are implemented using SQL language. All server side operations are implemented using ASP and VBScript while JavaScript is used as the client side scripting language.

5.2 DEVELOPMENT ENVIRONMENT

In development of one system, using the suitable hardware and software will help to speed up the entire system development. In below is the hardware and software tools used to develop and documented the entire system.

5.2.1 HARDWARE CONFIGURATION

This system is developed using the following hardware:

- Processor – Pentium III 600 MHz
- Memory – 256 MB RAM
- Hard Disk – 10.5 GB
- Network Interface Card (NIC)
- Standard monitor, mouse and keyboard

5.2.2 SOFTWARE CONFIGURATION

This system is developed using the following software:

- Operating System – Microsoft Windows 2000 Professional
- Web Server Software – Internet Information Services 5.0 (IIS)
- Visual Programming Tool for ASP – Macromedia Dreamweaver UltraDev 4.0
- Database Management System – Microsoft SQL Server 2000
- Word Processor for system documentation – Microsoft Word 2000
- Browser – Microsoft Internet Explorer 5.0

5.3 CODING STEP AND METHODOLOGY

The bottom up approach is used as the coding methodology for this system. This approach begins with the coding of the lower level modules first before the higher level modules. The higher modules are just skeletons that call the lower modules.

5.3.1 DATABASE SUBSYSTEM

Firstly, the system's database is developed according to the database design in Chapter 4. The data structure of each table is declared, the primary key is set, and the relationship among each table is defined. Then the database is maintained from time to time to add in some necessary constraints such as accessibility level for the users of the system.

5.3.2 INTERFACE SUBSYSTEM

Secondly, the interface for the system is built. The interface is developed to have Graphical User Interfaces (GUIs). GUIs support high-resolution color screens and interaction using a mouse as well as a keyboard.

5.3.3 SCHEDULING SUBSYSTEM

Thirdly, functionality is added to the user interfaces by adding in appropriate coding. The coding part of the system is begun with the coding of Administration module. In this module, there are five sub-modules:

- Room Information Maintenance
- Lecturer Information Maintenance
- Course Information Maintenance
- User Information Maintenance
- Change Password

Each of it is coded one by one to add in the function like adding new record, editing existing record, viewing all records and changing password.

Next, the Course Scheduling module is coded. There are four sub-modules in this module:

- Course Offered Entries
- Generate Timetable
- View Generated Timetable
- Change Password

Course Offered Entries and Change Password sub-modules are similar to the coding in Administration module. A scheduling engine is developed in the Generate Timetable sub-module. A lot of efforts and times are spent to complete the coding for the scheduling engine.

Then, the User module is coded. There are five sub-modules in this module:

- View Overall Timetable
- View Timetable By Program / Year
- View Timetable By Course
- View Timetable By room
- View Timetable By Lecturer

Each of it is coded one by one to allow users to view the generated timetable for a particular semester according to their preference.

Lastly, login function is built to ensure the authorized users for the Administration and Course Scheduling modules.

5.4 CODING STYLE

Coding style and its convention rules is an important attribute to determine the maintainability and readability of the source codes and the intelligibility of a program unit. The elements of style include internal documentation, methods for data declaration and statement construction [12]. The following convention rules of a good programming style were adopted when coding the entire system:

- Proper variables or field naming that does not against reserve words.
- Meaningful and understandable function and method declarations.
- Source codes indentation by functional segments for a neater look and readability.

- Keep all complex or compound statements as simple as possible to avoid confusion.
- Important comments in the program when necessary for future references.

5.5 DEVELOPMENT TOOLS

5.5.1 MACROMEDIA DREAMWEAVER ULTRADEV 4.0

Macromedia Dreamweaver UltraDev is a Rapid Application Development (RAD) tool for developing ASP applications. Dreamweaver UltraDev is used as the editor for the entire program because it is easy to develop and debug HTML and ASP coding. It provides several tools for me to create interactive Web-based user interfaces and ASP coding could be added easily to provide functionality to the user interfaces.

5.5.2 MICROSOFT SQL SERVER 2000

SQL Server 2000 is used as the Database Management System (DBMS) for this system. It allows developer to create easy-to-use database solutions quickly. By using this DBMS, database structure, validation rules, defaults, relationships and referential integrity could be done easily. It also allows developer to create user defined functions and procedures and stored them as a part of the database.

With SQL Server 2000, the Web server could manipulate data easily and run queries over the Web with natural language. This advantage allows this application to be run faster in responding to the browser.

5.6 PROGRAMMING LANGUAGE USED

5.6.1 HTML

In this system, the Web-based interfaces are created using HTML. HTML is the lingua franca for publishing hypertext on the World Wide Web. It uses tags like <A> and </A> to structure text into tables, hypertext links interactive forms, headings, paragraphs, lists, and more. HTML is useful to create form based data entry for this application. For example, Figure 5.1 shows some HTML code to create a form in this application.

```
1) < form name="addCourseForm" method="post" action="addCourseConfirm.asp" >
   </form >
2) < input type="text" name="courseName" size="50" maxlength="60" >
3) < select name="classType">
        <option value="L" selected>Lecture</option>
        <option value="M">Lab</option>
   </select >
4) < input type="submit" name="insert" value="Insert" >
```

Figure 5.1: HTML Code for creating data entry form

Command 1) creates a form for the user to add new record in database.

Command 2) creates a text field for the user to key in Course Name.

Command 3) creates a drop down list for the user to choose the Class Type of a course.

Command 4) creates a button and the data will be sent to the server if pressing the button.

5.6.2 JAVASCRIPT

JavaScript is a programming language that allows scripting of events, objects, and actions to create Internet applications. In FCSS, JavaScript is used at client side. It is used to validate data entered by the user. If invalid data is detected or user does not enter any data, a dialog box, similar to Figure 5.2 is displayed.

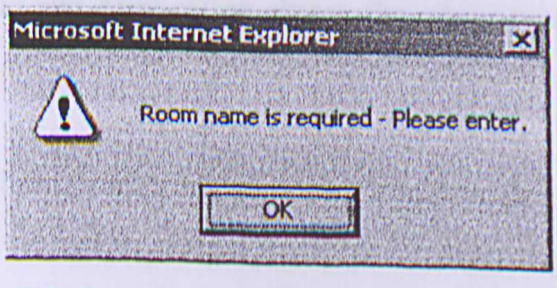


Figure 5.2: JavaScript Dialog Box for data validation

Besides that, it is also used to confirm the deletion or update from the user. For example, coding in Figure 5.3 displays a message box to ask confirmation from the user to delete or update database record.

```
<Script language="JavaScript">
<!--
        function Confirm()
        {
                if(confirm("Update / Delete this information?"))
                        return true;
                else
                        return false;
        }
//-->
</Script>
```

Figure 5.3: JavaScript Code for confirmation from the user

5.6.3 ASP SCRIPT

In FCSS, VBScript is used as server side scripting language to generate ASP coding. There are six built-in objects in ASP, which are:

- Application object
- ObjectContext object
- Request object
- Response object
- Server object
- Session object

Except for ObjectContext and Application object, the other objects are used frequently in FCSS application.

Request Object – Used to retrieve information passed to the Web server by the Web browser. For example, the below coding can retrieve input data from the user and the Web server could insert this data into database column.

```
objRS("course_code") = Request.Form("courseCode")
```

Figure 5.4: ASP Code for retrieving input data using Request object

Response Object – Sends text output to the browser and the text is displayed to the user. Coding in Figure 5.5 sends a welcome message to the browser.

```
Response.Write "Welcome to FSKTM Course Scheduling System"
```

Figure 5.5: ASP Code for sending text output to browser using Response object

Server Object – Manage server objects, HTML streams and URLs. Figure 5.6 shows how the Web server creates an connection object.

```
set objRS = Server.CreateObject("ADODB.Recordset")
```

Figure 5.6: ASP Code for creating connection object using Server object

Session Object – Store temporary information during a user session. It could store the user name and password for the user who login to the system.

```
Session("username") = rsUsers("user_name")
```

Figure 5.7: ASP Code for storing user name in Session variable

5.6.4 ADO

ADO (Active X Data Objects) is a group of objects designed to provide a simple programming interface to database. ADO has only six objects and two collections, and of those, only two main objects are used in FCSS:

- Connection
- Recordset

Connection Object – It represents the actual session established with the database. Open() and Close() are the two methods that frequently used. Figure 5.8 shows how the connection object is used in FCSS.

```
set objConn = Server.CreateObject("ADODB.Connection")
objConn.Open strConnect
.
.
objConn.Close
```

Figure 5.8: Connection Object

Recordset Object – A Recordset Object represents any group of records – whether it is the result of a query or the entire contents of a table. This is the object use for almost all data access. For example, view timetable will require this Recordset to get the information from database.

```
set objRS = Server.CreateObject("ADODB.Recordset")
objRS.Open "Time_Slot", strConnect
.
.
objRS.Close
```

Figure 5.9: Recordset Object

5.6.5 SQL

After establishing a connection with database, SQL (Structured Query Language) statements is used to insert, delete and retrieve information from database. The following coding is one of the SQL statements in FCSS, where all the rooms' information will be retrieved from table *Room_Info* in database.

```
StrQuery = "SELECT * FROM Room_Info ORDER BY room_code
```

Figure 5.10: SQL Statement for retrieving data from database

5.7 SYSTEM DOCUMENTATION

From time to time, the project status is documented using Microsoft Word 2000. Documentation is important to help developer to determine the progress of the project. The system documentation includes all of the documents describing the implementation of the system from the requirement specification to the final acceptance test plan. There are two types of documentation prepared in this system, which are important for future maintenance needs in the system. These documents are:

1. User Manual

User manual is a reference guide or tutorial for system users. The manual describes the application's purpose and teaches users on how to use the system. This preliminary information is especially helpful in reassuring users that the document contains the type of information they seek. Special terms, abbreviations, or acronyms used in the manual are included for easy reference.

2. Internal system documentation

This document contains information that directed someone who wants to read the source code of the programs. Thus, summary information is provided to identify the program and describe its data structures, algorithms, and control flow.

CHAPTER 6

SYSTEM TESTING AND MAINTENANCE

6.1 INTRODUCTION TO SYSTEM TESTING

Testing is the act of executing the system to uncover bugs or errors and ensure the system meets defined requirements. It involves exercising the program using data like the real data processed by the program. Testing may be carried out during the implementation phase to verify that the system behaves as intended by its designer and after the implementation has completed [12].

Different types of testing use different types of test data. There are two types of testing that requires different set of test data:

1. Statistical Testing

Statistical testing is used to test program's performance and reliability. Tests are designed to reflect the frequency of actual user inputs. After running the tests, an estimate of the operational reliability of the system can be made.

2. Defect Testing

Defect tests are designed to discover system defects. A successful defect test is one that reveals the presence of defects in a system. It is intended to find areas where the program does not conform to its specification. When defects have been found in a program, these must be discovered and removed. Debugging is concerned with locating and correcting these defects.

6.2 TESTING PROCESS

In FCSS (FSKTM Course Scheduling System), testing process consists of four stages as shown in Figure 6.1. In general, defects are discovered at any stage, they require program modification to correct them and this may require other stages in the testing process to be repeated.

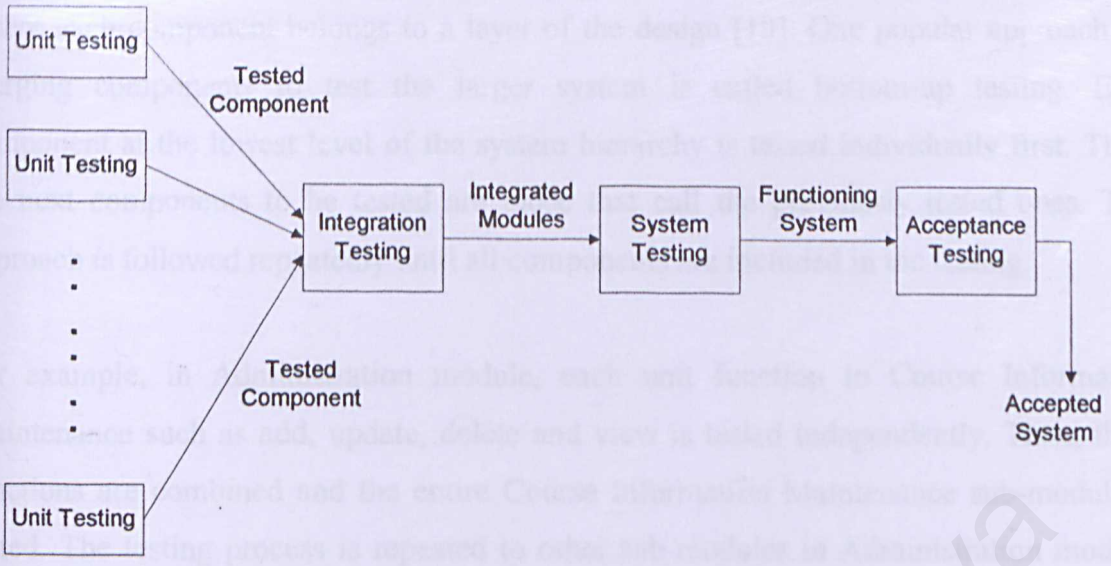


Figure 6.1 Testing Process in FCSS

6.2.1 UNIT TESTING

This stage focuses on the testing of individual subroutines and modules. Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system component [12].

In FCSS, code review had been done to make sure that the code had considered all the relevant cases. FCSS is divided into three modules and each module contains many sub-modules and functions, called units. Each of this unit is tested one by one to ensure its functionality.

For example, Course Information Maintenance sub-module consists of functions for adding record, editing existing record and viewing record. Unit testing involves finding out the fault in these units and make sure the data structure is correct. For instance, a record of course is added to check whether the record could be stored in database. Besides that, user interface is checked to ensure smooth operation.

6.2.2 INTEGRATION TESTING

When individual components are working correctly and meet the objectives, there are combined into a working system. The system is viewed as a hierarchy of components,

where each component belongs to a layer of the design [10]. One popular approach for merging components to test the larger system is called bottom-up testing. Each component at the lowest level of the system hierarchy is tested individually first. Then, the next components to be tested are those that call the previously tested ones. This approach is followed repeatedly until all components are included in the testing.

For example, in Administration module, each unit function in Course Information Maintenance such as add, update, delete and view is tested independently. Then, these functions are combined and the entire Course Information Maintenance sub-module is tested. The testing process is repeated to other sub-modules in Administration module. After testing all of the sub-modules and there are no errors found, these sub-modules are integrated as one and it is tested to ensure its functionality. (Figure 6.2)

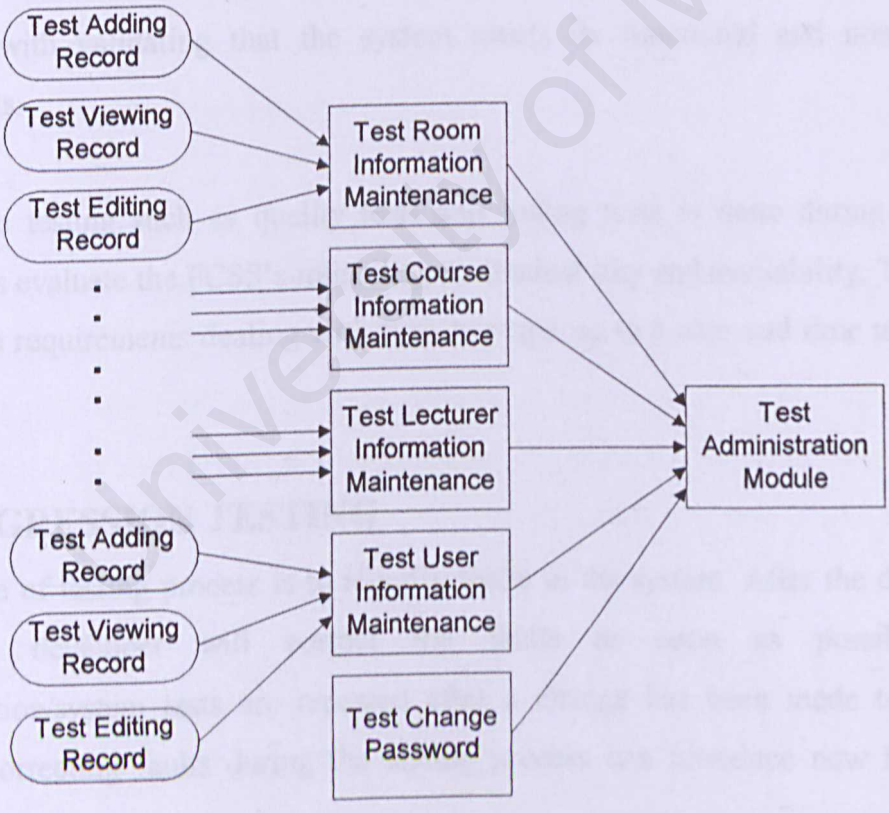


Figure 6.2 Integration Testing using bottom-up approach in Administration Module

6.2.3 SYSTEM TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work should verify that all system elements (software, hardware, interfaces) have been properly integrated and perform allocated functions. There are four types of system level testing that can be performed.

- Recovery Testing
- Security Testing
- Stress Testing
- Performance testing

In this stage, the testing process is concerned with finding errors that result from unanticipated interactions between sub-systems and system components. It is also concerned with validating that the system meets its functional and non-functional requirements.

Performance testing such as quality tests and timing tests is done during this stage. Quality tests evaluate the FCSS's reliability, maintainability and availability. Timing tests evaluate the requirements dealing with time to response to a user and time to perform a function.

6.2.4 REGRESSION TESTING

The purpose of testing process is to identify faults in the system. After the discovery of faults, the developer will correct the faults as soon as possible. Then, unit/integration/system tests are repeated after a change has been made to the code. However, correcting faults during the testing process can introduce new faults while fixing old ones.

Regression testing identifies new faults that may have been introduced as current ones are being corrected. A regression test is a test applied to a new version or release to verify that it still performs the same functions in the same manner as an older version or release.

In FCSS, faults found were corrected and the unit, integration and system testing were repeated. Finally, a fully integrated and functioning system is produced and will be test for acceptance.

6.2.5 ACCEPTANCE TESTING

This is the final stage in the testing process before the system is accepted for operational use [12]. FCSS is tested with data from FSKTM where course, room and lecturer in FSKTM are used to make sure this application could be accepted by FSKTM.

6.3 DEBUGGING THE SYSTEM

Testing process helps developers to discover failure in the system. Then, developers must correct the faults so that the system could operate correctly without failure. Debugging is the process of finding the faulty code responsible for failed tests. It helps developer to determining what fault or faults caused the failure, and then corrections are made to the system so the faults are removed.

In FCSS, debugging process was done using Microsoft Script Debugger. Normally, failure was cause by algorithmic faults that occur when a components' algorithm or logic does not produce the proper output for a given input because something is wrong with the processing steps. These faults are sometimes easy to spot just by reading through the program (call desk checking) or by submitting input data from each of the different classes of data that we expect the program to receive during its regular working. Typical algorithmic faults in FCSS include

- branching too soon
- branching too late
- testing for the wrong condition
- forgetting to initialize variables or set loop invariants
- forgetting to test for a particular condition (such as when division by zero might occur)
- comparing variables of inappropriate data types

With the help from the script debugger software, these faults are corrected eventually and the system could run as expected.

6.4 OPERATION AND MAINTENANCE

It is impossible to produce systems of any size, which do not need to be changed. Over the lifetime of a system, its original requirements will be modified to reflect changing user and customer needs. Maintenance activities are similar to those of development: analyzing requirements, evaluating system and program design, writing and reviewing code, testing changes and updating documentation.

Maintenance focuses on four major aspects of system evolution simultaneously:

- Maintaining control over the system's day to day functions.
- Maintaining control over system modifications.
- Perfecting existing acceptable functions.
- Preventing system performance from degrading to unacceptable levels.

There are four different types of software maintenance:

1. ***Corrective maintenance*** is concerned with fixing reported errors in the software.
2. ***Adaptive maintenance*** means changing the software to some new environment such as different hardware platform or for use with a different operating system.
3. ***Perfective maintenance*** involves implementing new functional or non-functional system requirements.
4. ***Preventive maintenance*** involves changing some aspect of the system to prevent failures.

6.4.1 MAINTENANCE PROCESS IN FCSS

In general, maintenance process is carried out after the system is being run and tested fully by the user in the real working environment. Since this system has just been developed and has not been fully run in the working environment in FSKTM, this phase cannot be carried out in the system.

CHAPTER 7

SYSTEM EVALUATION AND CONCLUSION

7.1 INTRODUCTION

System evaluation is the process of identifying the system strength and limitations, and possible enhancements of the FCSS application. The process also helps developer to identify problems faced and solutions to those problems during the development of the application. Through the evaluation process, developer could determine how far the system has achieved its objectives and fulfill the users requirements.

7.2 SYSTEM STRENGTH

1. Wide accessibility

FCSS is a web-based application that provides wide accessibility to the users and allows them to access the system virtually from anywhere through the Internet. Client side software only requires the installation of a web browser to access the system. Furthermore, browsers are already available across all platforms and allow users to access to the system easily.

2. User Friendly Interface

The system's interface is interesting due to the using of graphical user interface (GUI) concept. The interface uses menus, buttons, scrollbars and text boxes, which make the system relatively easy for users to learn and use. Users should find the system user-friendly and interesting, without needing much skill on handling it. It creates no difficulties to novice who is unskilled in using the keyboard and mouse.

3. Security System

Only authorized users who have login authentication is allowed to manipulate and make changes on the database information. Users must login using their password if they want to access Administration and Course Scheduling modules. The administrator and timetable coordinator could change their password by entering the old password correctly.

4. Automated scheduling process

The scheduling process in FCSS is automated where the scheduling engine could generate the timetable automatically. The scheduling engine will check for the constraints and

allocate time slot for courses. The generation of schedule with the system is very fast and user does not need to spent much time on clerical work involve in scheduling.

5. Validation of data input

Users' input is validated and verified through using client side scripting and server side scripting. The purpose of data validation is to ensure that every database field is filled correctly.

6. Various Timetable View

Students not only could view the overall timetable, they also could view their timetable by their degree program. Besides that, students and lecturers could view their timetable by a particular course, by lecturer and by room.

7. Retrieving and manipulating records

The application allows efficient ways in retrieving relevant and necessary records on the screen. It is easy on manipulating records found in the database, such as adding new records, editing and deleting existing records.

7.3 SYSTEM LIMITATION

1. Timetable print out

FCSS does not provide facility on printing the timetable on web. This is not effective, as users have to copy the timetable manually.

2. Schedule courses by lecturer

There is no lecturer section in this system. Lecturers are not permitted to undertake scheduling of courses assigned to them. Lecturer is not allowed to choose and set their preferable teaching time.

3. Help facility

The system does not provide help facility to assists users on how to use the system.

4. Browser Limitation

For fully utilize FCSS, the browser used in the client side (IE or Netscape) must at least version 3.01 or above. This is because browser with version below 3.0 may not support some of the client side functions and commands, such as client side data validation implemented using JavaScript.

5. Database management limitation

There is no proper management on database file, where no functions were created in the application to backup and to compact the database.

7.4 FUTURE ENHANCEMENT

1. Printable Timetable

FCSS could provide the functionality that allows users to print out the timetable.

2. Lecturer Section

Lecturer module might be added to give permission to lecturers to undertake scheduling of courses assigned to them. It means that lecturers are allowed to review and select their preferable and not preferable teaching time slot.

3. Help Facility

The system could provide help feature to guide user on how to use the system. The help facility could assists users by explaining the timetabling process, the ways of adding, editing and deleting data. Users could access to help facility by clicking the "Help" button.

4. Database Management

The application should give users fully control over the database used, where they can do their own maintenance such as backup and compacting the database.

7.5 PROBLEMS AND SOLUTIONS

Various problems were faced during the development of FCSS. However, these setbacks were eventually solved or reduced. The problems and solutions are described below.

7.5.1 HOW WIDE OF THE SYSTEM SCOPE TO BE BUILT

It is impossible to build a full-scale complete system because of time limitation. Therefore, I have to determine the system scope that must be encountered in order to fulfill the users requirements. A good schedule should meet all the scheduling hard constraints and maximize the fulfillment of scheduling soft constraints. It is very important to defined the system scope clearly so that a good schedule could be produce by the system.

Many discussions had been carried out with my project supervisor to outline the scope of the project to be built. After the scope has been defined, analysis of the system was done and the development of the system started.

7.5.2 DIFFICULTY IN DESIGNING A GOOD DATABASE

Due to the complicated of timetable in this faculty, it is very difficult to design a good database to store information needed in scheduling process. There are many special cases in FSKTM such as particular courses might have two lecturers. So, the database must be designed to store all information for each special case.

With the guidance from project supervisor and some examples from reference books, a suitable database design was created to manage all the information.

7.5.3 DIFFICULTY IN CHOOSING A SUITABLE SYSTEM DEVELOPMENT TOOLS

There are too many system development tools available for developing a web-based application. Choosing suitable tools is very important because the use of suitable tools could speed up the System Development Life Cycle and minimal the unexpected bug and error.

To determine the suitable tools, I have seek for advises from my course mates and senior who have experience in developing web-based application. Besides, surfing through the Internet and visiting the library help clarify some doubt. Finally, I managed to choose the suitable tools and programming language that could be use to develop this application.

7.5.4 DIFFICULTY IN CHOOSING TIMETABLING ALGORITHM

As stated in Chapter 2, there are a lot of timetabling algorithms that could be used to schedule courses. Due to the unfamiliarity with those algorithms, I faced with the problem on choosing the timetabling algorithm for FCSS.

I have visited the library and found information about those algorithms. Then, I tried to understand those algorithms. Finally, heuristics and graph coloring method is choosing.

7.5.5 TIME CONSTRAINT

There was not enough time to study, learn and produce the best solution of design in Semester 1. Mainly, this was cause by inexperience and insufficient knowledge of designing a system. Furthermore, time is needed to study and explore ASP language, HTML and SQL Server 2000 before knowing how to apply these technologies and languages in the process of developing and solving problems.

Thus, I have made a lot of reference to those who are familiar with these languages and technologies. In addition, I read reference books to catch up on how to build a good system using these language and technologies.

7.6 EVALUATION ON OBJECTIVES ACHIEVED

Basically, the developed system has achieved most of its main objective defined during the analysis phase and has met the functional and non-functional requirements.

In FCSS, it allows the administrators to input related information into the system. The information includes courses' information, lecturers' information, rooms' information and users' information. All the data will be store in database. Besides that, this system allows

the timetable coordinator to input courses offered and assigned lecturers to those courses for a corresponding semester. Then the system could generate courses timetable automatically and publish the timetable on the web. Students and lecturers could view their timetable through the Internet.

In conclusion, FSKTM Course Scheduling System has achieved the following objectives:

1. Reducing the time and effort for generating a course schedule in FSKTM. It could speed up the generation of course schedule in our faculty.

2. Intelligent and flexible enough to overcome variety of schedule constraints encountered in real-life problems such as students timing conflict and rooms allocation. The system could produce schedules that are feasible and with sufficient quality to be used.

3. All timetables are automatically generated.

4. FCSS is user-friendly and convenient: easy to use and the users could view timetable at any places, any time through the Internet.

5. It is standard and easy enough for the administrators and timetable coordinators to be trained to handle and maintain the system in short period.

7.7 PROJECT CONCLUSION

As a conclusion of this project, FCSS has fulfilled the requirement to deliver the system in time and achieved most of the objectives and requirements as determine during analysis phase.

Throughout this project, I have learned and gained a lot of valuable knowledge and experience. During the period of system development, I become clearer on how to establish the connection to database, maintaining and configuring database, how Internet technologies and ASP concepts work, and also how to configure IIS as the Web server software. Besides that, I have gained a lot of experience in system analysis, planning, design, implementation and testing. In addition, skills in using software such as Macromedia UltraDev 4.0, Visual InterDev, Microsoft FrontPage 2000 and Adobe Photoshop have been acquired.

During the project development, programming skills and good practice on software engineering techniques are essential and must also be applied in an efficiency way. Therefore, this project has provided the good chances to experience using the method, techniques, paradigms, and approaches that learned from System Analysis & Design and Software Engineering courses in the second year and third year study respectively.

The scope of this system was defined through waterfall model with prototype; this makes the system most expandable in terms of functionality. Thus, enhancement could still made to this system with more features added. Lastly, hoping that the features and benefits of this system will enable the course scheduling process to become more effective, systematically and efficiency.

REFERENCES

REFERENCES

[1] BP Romero, "Examination Scheduling In A Large Engineering School: A Computer Assisted Participative Procedure", Interface, vol. 12, no.2, pp. 17-234, The Institute of Management Sciences, April 1982.

[2] Andrea Achaerf, "A Survey of Automated Timetabling", report CS-R9567, The Netherlands, 1995.

[3] R. Weare, "Automated Examination Timetabling", University of Nottingham, UK, 1995.

[4] Andrea Schaerf, "Tabu Search Techniques for Large High-School Timetabling Problems", 13th National Conference of The American Association for Artificial Intelligence, 1996.

[5] D. Corne, "Evolutionary Timetabling: Practice, Prospects And Work In Progress", UK Planning and Scheduling SIG Workshop, 1994.

[6] Deborah Craig, "CGI Manual Of Style", Macmillan Computer Publishing, USA, 1996.

[7] Lave Singh, "Oracle Unleashed", SAMS Publishing, 1996.

[8] P. Renaud, "Introduction to Client/Server Systems", John Wiley and Son, New York, 1996.

[9] A. Benson, "Client/Server Architecture", McGraw Hill, New York, 1996.

[10] Shari Lawrence Pfleeger, "Software Engineering Theory And Practice", Prentice Hall International, Inc, 1998.

[11] Jeffrey L. Whitten, Lonnie D. Bentley, & Kevin C. Dittman, "System Analysis & Design Methods", McGraw Hill, New York, 2000.

[12] Ian Sommerville, "Software Engineering", Addison-Wesley, 1996.

[13] Thomas M. Connolly, "Database Systems A Practical Approach To Design, Implementation and Management", Addison-Wesley, September 1995.

[14] Suleiman K. Kassicieh, "Design And Implementation of a Decision Support System for Academic Scheduling", The International Journal of Information Systems Applications, Elsevier Science Publishers, 1996.

[15] Peter Ross, Dave Corne, Hsiao Lan Fang, "Successful Lecture Timetabling with Evolutionary Algorithms", University of Edinburgh, 1994.

[16] Wilhelm Erben, Jurgen Kepper, "A Genetic Algorithm Solving a Weekly Course-Timetabling Problem", First International Conference on the Practice and Theoryof Automated Timetabling, 1995.

[17] Faculty of Computer Science and Information Technology, "Buku Panduan Fakulti Sains Komputer & Teknologi Maklumat", University of Malaya, 1999.

[18] Weiying Chen, "ActiveX Programming Unleashed", Macmillan Computer Publishing, 1996.

[19] Sanjana Hettihewa, "Teach Yourself Active Server Pages 2.0 in 21 Days", SAMS Publishing, November 1998.

[20] David Buser, John Kauffman, Juan T. Llibre, Brian Francis, David Sussman, Chris Ullman & Jon Duckett, "Beginning Active Server Pages 3.0", Wrox Press Ltd., November 2000.

[21] Keith Brophy, Timothy Koets, "Teach Yourself VBScript in 21 Days", SAMS Publishing, 1996.

[22] Rick Samtell, "Teach Yourself Microsoft SQL Server 6.5 in 21 Days", Macmillan Computer Publishing, 1996.

[23] Ryan K. Stephens, "Teach Yourself SQL in 21 Days, Second Edition", Macmillan Computer Publishing.

[24] http://www.macromedia.com

[25] http://httpd.apache.org

[26] http://java.sun.com

[27] http://msdn.microsoft.com/vinterdev

[28] http://www.microsoft.com/sql

[29] http://wdvl.com/Authoring/JavaScript

[30] http://www.mysql.com

[31] http://www.microsoft.com/office/access

[32] http://www.microsoft.com/window2000

ACRONYMS

ACRONYMS

ADO	ActiveX Data Objects
ASP	Active Server Pages
CGI	Common Gateway Interface
CLP	Constraints Logic Programming
DBMS	Database Management Systems
DSN	Data Source name
EA	Evolutionary Algorithms
FCSS	FSKTM Course Scheduling System
FSKTM	Faculty of Computer Science and Information Technology
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IE	Microsoft Internet Explorer
IIS	Internet Information Server
JSP	Java Server Pages
MS	Microsoft
ODBC	Open Database Connectivity
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol/ Internet Protocol
VB	Visual Basic
VBScript	Visual Basic Scripting Edition
WWW	World Wide Web
XML	Extensible Markup Language
MSDE	Microsoft Data Engine

GLOSSARY

GLOSSARY

Active Server Pages (ASP)

Is a HTML page that includes scripts both for client-side and server-side that are processed (only server-side scripts) on the Web server before they are sent to the user. It is somewhat similar to a Common Gateway Interface (CGI) application.

ActiveX Data Objects (ADO)

It is an application program interface (API) from Microsoft. It lets programmers writing Windows applications and get access to relational and non-relational databases from both Microsoft and other database providers.

Browser

Also known as client browser. It is an application program that provides a way to look at and interact with all the information on the World Wide Web by doing the requesting connection to the web server.

Client/Server

The relationship between two computer programs in which one program, the client, makes a service request from another program, the server, which fulfills the request.

Common Gateway Interface (CGI)

CGI is a standard way (usually in UNIX platform) that defines how web servers can access external programs for execution to return or insert data in the format of web page.

Database

A collection of related information stored in a structured and organized way. Using this structured collection, standard methods of retrieval like using Structured Query Language (SQL) can be used to define and manipulate the data.

DFD (Data Flow Diagram)

Is used to represent the input, output data and processes in an application.

Evolutionary Algorithms (EA)

Use computational models of evolutionary processes as key elements in the design and implementation of computer-based problem solving systems.

Genetic Algorithms

It is a solution technique for optimization problems.

Graph Coloring

Is one of the classical NP-complete problems on graphs.

Graphical User Interface (GUI)

It is an interface between a user and the computer. GUIs, generally, requires a mouse-a type of pointing device. All GUI-based programs usually look similar, with pull-down menus, scroll bars, etc.

Hypertext Markup Language (HTML)

The set of "markup" symbols or codes inserted in a file intended for display on a World Wide Web Browser.

Hypertext Transfer Protocol (HTTP)

HTTP is a set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Integer Programming (IP)

IP is a linear programming problem in which some or all of the variables to be optimized must be positive integers.

Internet

A worldwide system of linked computer networks for data communication services such as World Wide Web and electronic mail. The Internet is a way of connecting existing computer networks that greatly extends the reach of each participating system.

Internet Information Server (IIS)

Software that provides a group of Internet services (Web or HTTP, FTP, and Gopher) and other capabilities for Microsoft Windows NT Operating System.

JavaScript

An interpreted programming or script language from Netscape. It is somewhat similar in capability to Microsoft Visual Basic, Sun's Tcl, the UNIX-derived Perl, and IBM's REXX.

Java Server Pages (JSP)

It is a HTML page that includes scripts both for client-side and server-side that are processed (only server-side scripts) on the Web server before they are sent to the user. It is somewhat similar to a Common Gateway Interface (CGI) application.

Open Database Connectivity (ODBC)

A standard or open application program interface (API) for accessing a database.

Primary Key

The field (or fields) that uniquely identify a record in a table.

Relationship

Associations between two tables. Normally, there are common field (or fields) that identify how the tables are connected.

Structured Query Language (SQL)

A standard computer language for manipulating data stored in database.

VBScript

Language that is closely related to the BASIC programming language.

Waterfall Model

A systematic sequential approach to software development modeled after a conventional engineering cycle.

Web Server

A program that, using the client/server model and the World Wide Web's Hypertext Transfer Protocol (HTTP), serves the files that forms Web pages to Web users (whose computers contain HTTP clients that forward their requests).

World Wide Web (WWW)

It a popular hypertext-based system of transmitting textual and multimedia-based information through the Internet.

APPENDIX A

USER MANUAL

CONTENTS

1.0 INTRODUCTION

1.1 ABOUT FCSS (FSKTM COURSE SCHEDULING SYSTEM)

FCSS (FSKTM Course Scheduling System) is an automated Web-based course scheduling system developed for FSKTM (Faculty of Computer Science And Information Technology). The purpose of FCSS is to reduce the time and effort needs for generating a timetable in FSKTM. It could generate timetable automatically for a particular semester and publish the timetable on the Web.

1.2 GETTING STARTED WITH FCSS

FCSS is a Web-based application that could be accessed through the Internet. The users just have to type the address of the site in the web browser to access the system. However, before using this system, users must make sure that the hardware and software requirement meet the following criteria in order to utilize this system.

1.2.1 SERVER SIDE RUN TIME ENVIRONMENT

Hardware Requirements

The hardware required for running FCSS are:

- A server with at least Pentium 166 Mhz processor
- At least 64MB RAM
- Network Interface Card (NIC) and network connection

Software Requirements

The software required for running FCSS are:

- Microsoft Windows 2000
- Internet Information Server 3.0 or higher
- Microsoft SQL Server 2000

1.2.2 CLIENT SIDE RUN TIME ENVIRONMENT

Hardware Requirements

The hardware required for accessing FCSS through the Internet are:

- At least 16MB of RAM
- Network connection through existing network configuration/LAN or via modem

Software Requirements

The software required for accessing FCSS through the Internet are:

- Any Web Browsers such as Netscape Communicator
- Microsoft Internet Explorer 3.01 or above (recommended)

1.2.3 WHO IS GOING TO USE THIS FCSS

The users of FCSS could be divided into three categories, which are:

1. ***Administrator***

   Administrator could access Administration module to manage course, room, lecturer and user information. They have to use their username and password to login to the system for this purpose.

2. ***Timetable Coordinator***

   Timetable coordinator could access Course Scheduling module to input course offered, assign lecturer for those offered course and generate timetable. They have to use their username and password to login to the system for this purpose.

3. ***Students and lecturers***

   Students and lecturers could view their timetable.

1.2.4 HOW TO ACCESS FCSS

To use FCSS, the users need to start their Web browser. Then type the FCSS's web site address on the browser's address combo box and press enter. The web address would probably sound as follow:

http://fsktm.um.edu.my/FCSS/Home.html

Figure 1.1 shows the main page for FCSS.

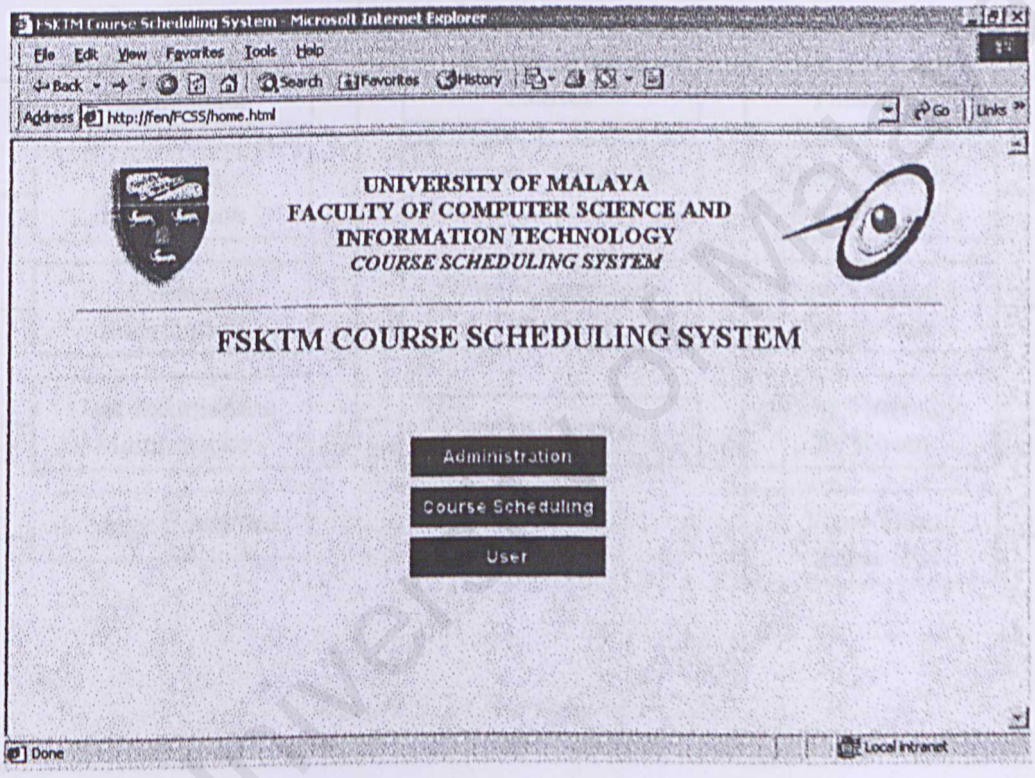


Figure 1.1 Main Page for FCSS

1.3 OVERVIEW OF FCSS

Figure 1.2 shows the overall functions of FCSS.

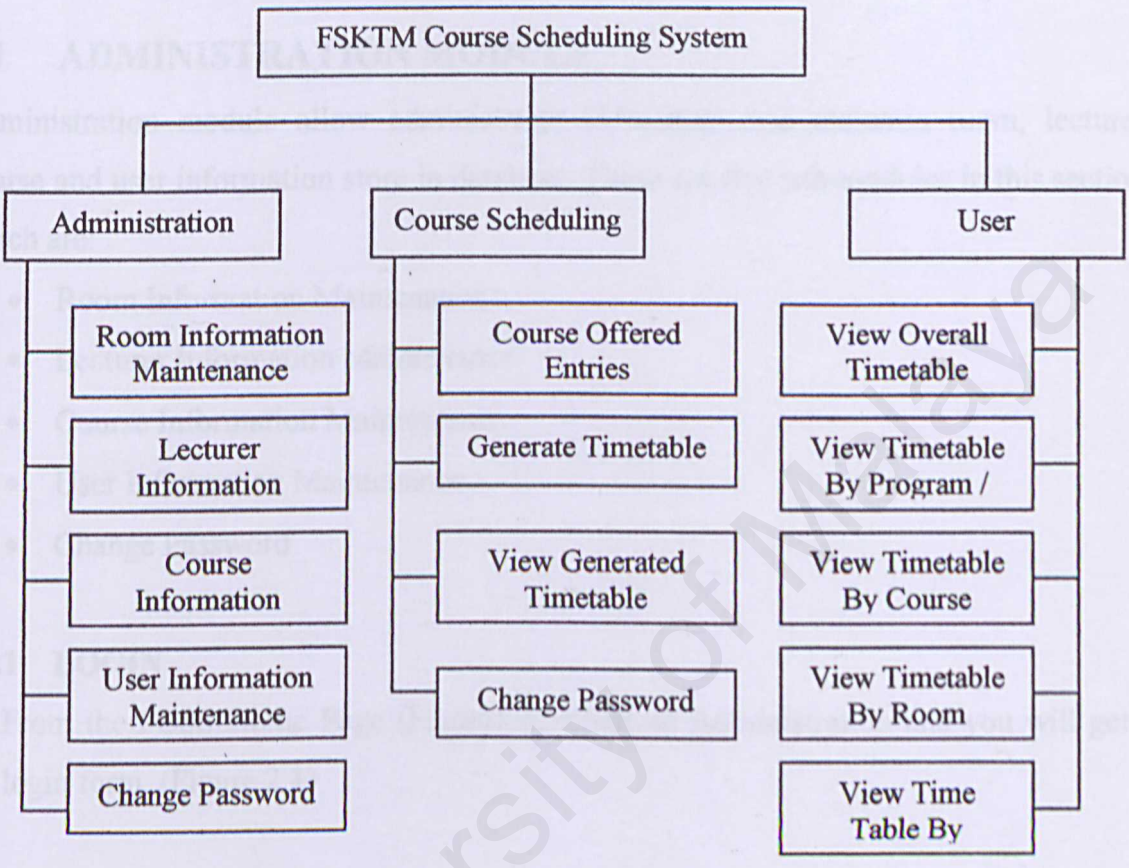


Figure 1.2 Overview of FCSS

2.0 SYSTEM FEATURES

The module in FCSS can be divided into three, which are Administration module, Course Scheduling module and User module.

2.1 ADMINISTRATION MODULE

Administration module allow administrator to manage and maintain room, lecturer, course and user information store in database. There are five sub-modules in this section, which are:

- Room Information Maintenance
- Lecturer Information Maintenance
- Course Information Maintenance
- User Information Maintenance
- Change Password

2.1.1 LOGIN

1. From the FCSS Home Page (Figure1.1), click on Administration and you will get a login form. (Figure 2.1)

2. To logon to Administration module, you have to key in your user name and password into the login form (Figure 2.1) and press Login.

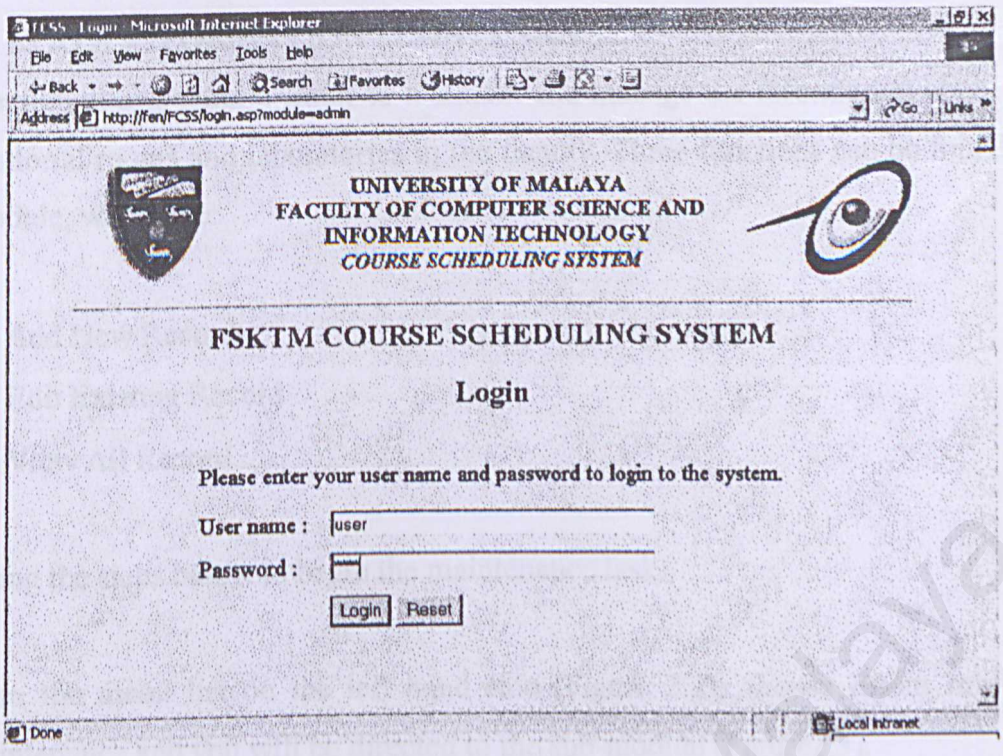


Figure 2.1 Login Form

3. A successful logon will bring you to administration module (Figure 2.2).

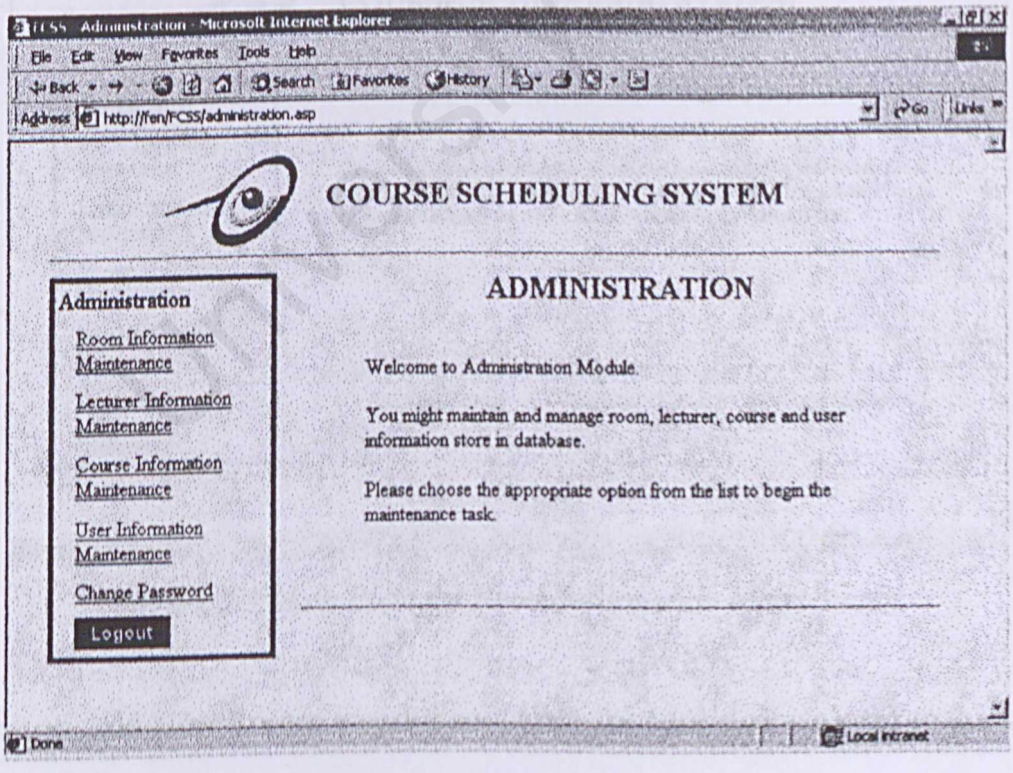


Figure 2.2 Main Page for Administration Module

2.1.2 ROOM INFORMATION MAINTENANCE

This section allows administrator to maintain and manage the information about lecture halls, tutorial rooms and laboratories in the faculty. Three functions can be found in this sub-module, which are:

- Add New Record
- Edit Existing Record
- View All Record

Following the steps below to begin the maintenance task:

1. From the menu list on the left hand side (Figure 2.2), choose Room Information Maintenance and you will be directed to the sub-module (Figure2.3).

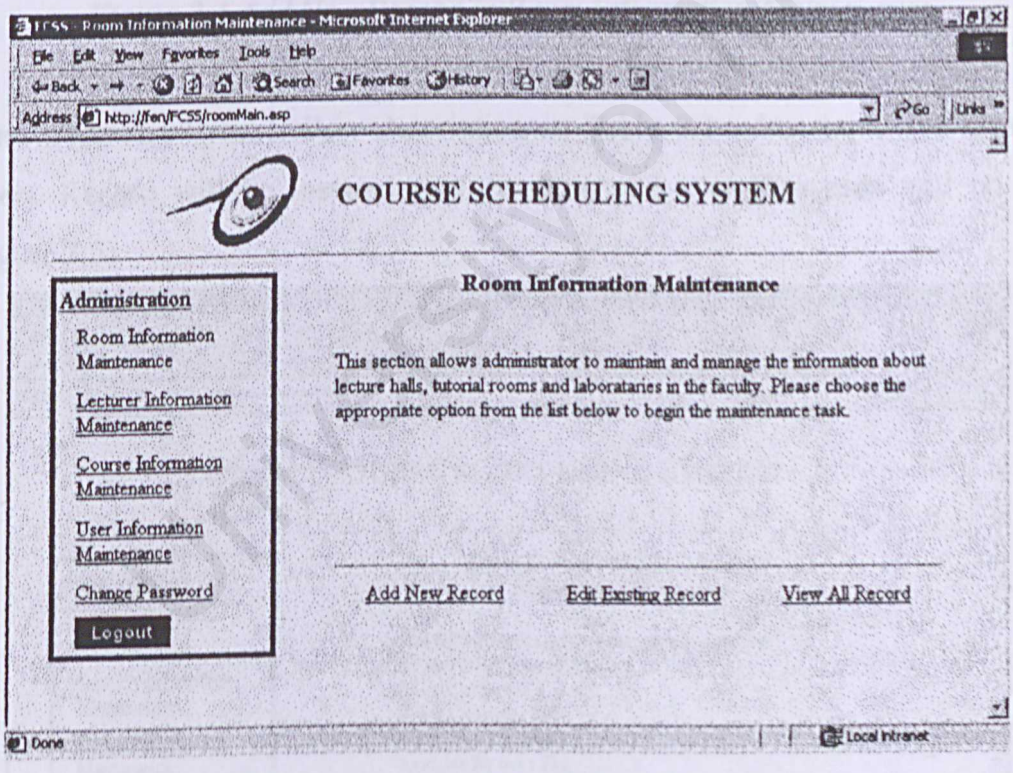


Figure 2.3 Main Page for Room Information Maintenance

2. Click on Add New Record to get a new form for adding new record to the database (Figure 2.4).

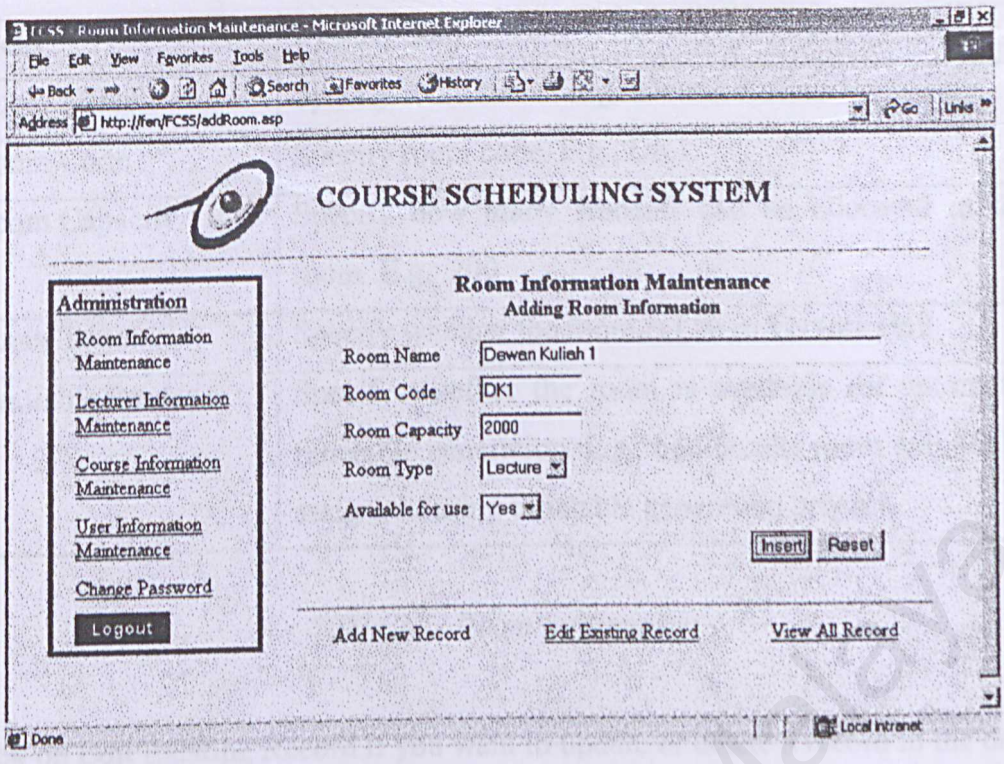


Figure 2.4 Add New Record in Room Information Maintenance

3. Input necessary data. Follow description in Table 2.1 when input the data. Press Insert. Record will be inserted into database and a confirmation will be shown. (Figure 2.5)

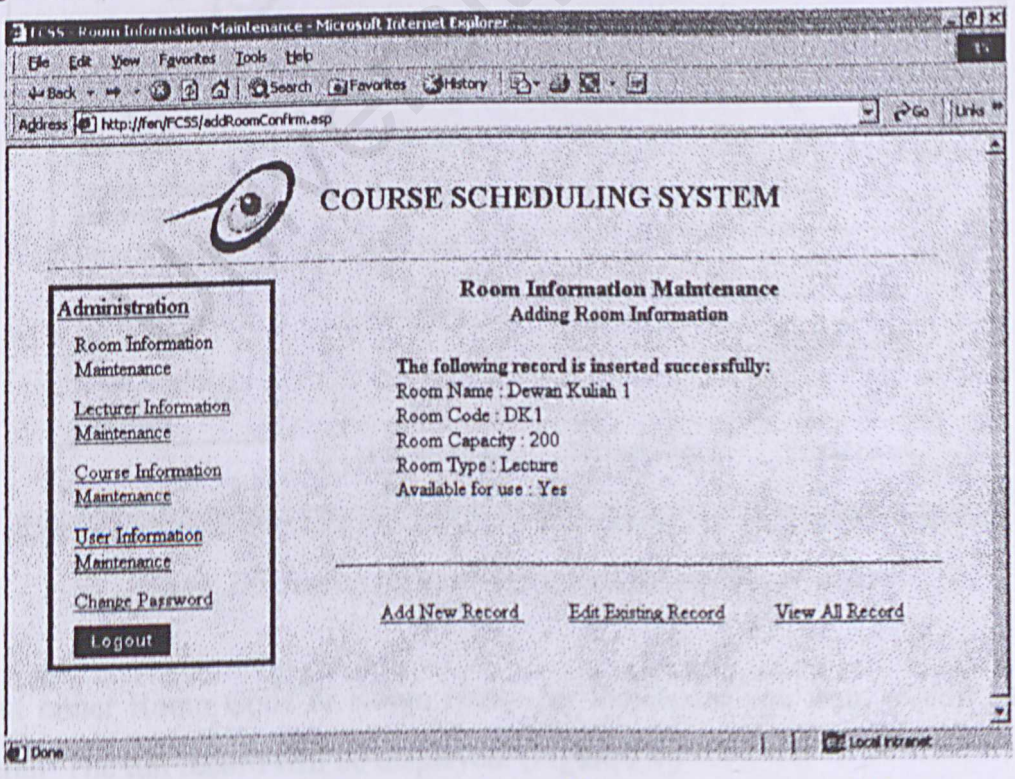


Figure 2.5 Confirmation when adding record successfully

Field	Description
Room Name	Specify room name. E.g.: Dewan Kuliah 1
Room Code	Specify room code. E.g.: DK1
Room Capacity	Specify how many students can be allocated in the room. E.g.: 200
Room Type	Specify whether the room is Lab or Lecture Hall
Available for use	Specify whether the room is available for automatic timetable generation. E.g.: non-faculty room will not be assigned during timetable generation process.

Table 2.1: Room Information

4. Click on Edit Existing Record if you want to update or delete a record in the database. You will get a form to search for the record you want to edit. (Figure 2.6)

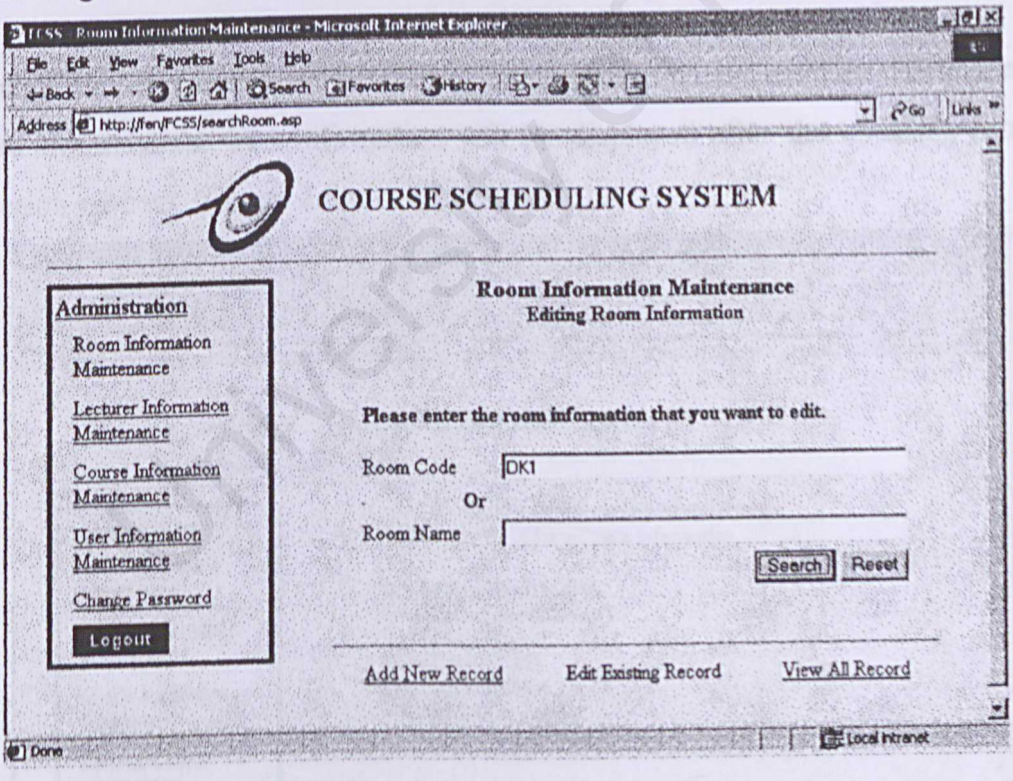


Figure 2.6 Search an existing record for update or delete

5. Input either Room Code or Room Name for the record you want to edit and press Search. The full record will be displayed. (Figure 2.7)

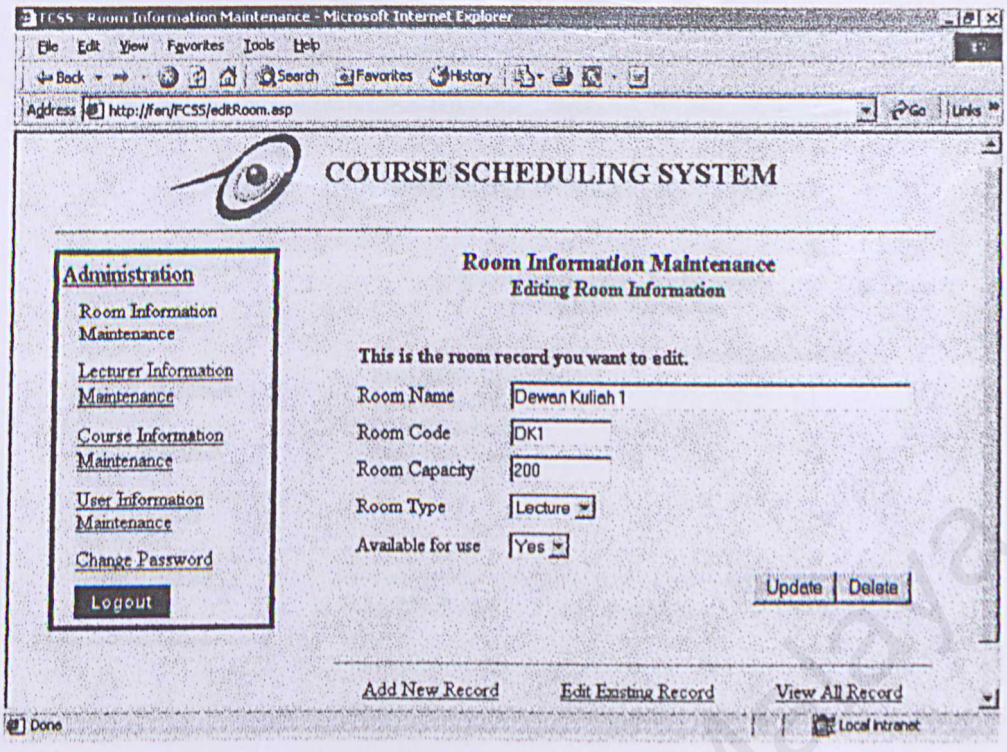


Figure 2.7 Edit Existing Record

6. Change the data and press Update or press Delete if you want to delete the record from the database. A confirmation will be displayed after the updating process or deletion process. (Figure 2.8 and Figure 2.9)

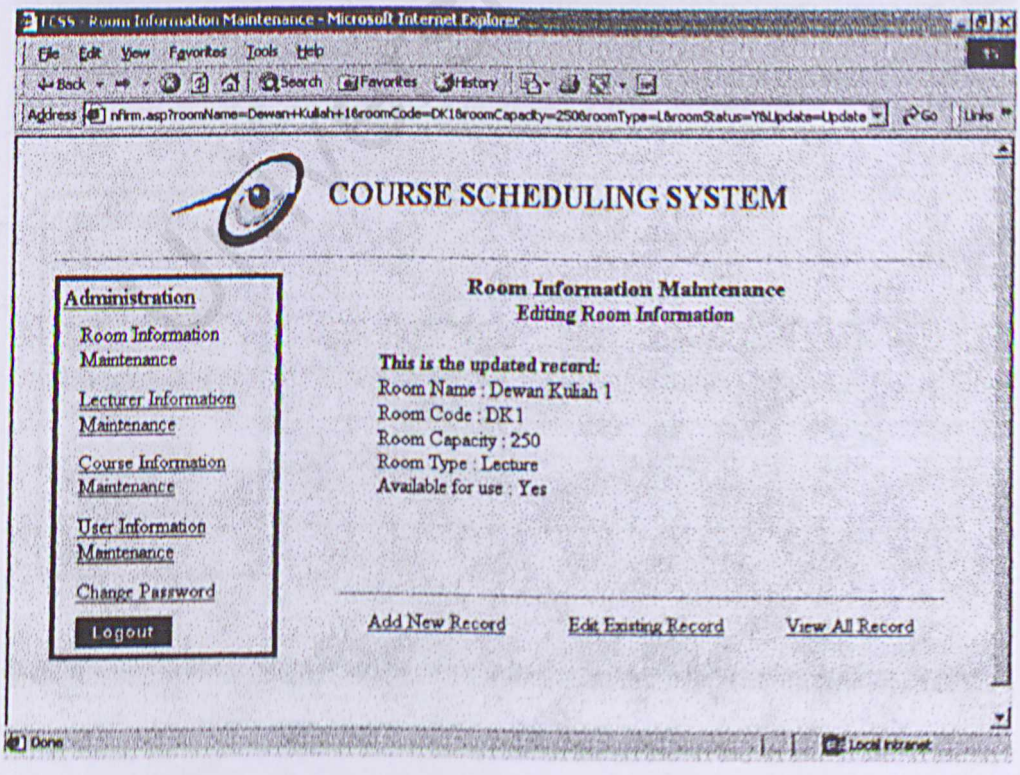


Figure 2.8 Confirmation after updating a record

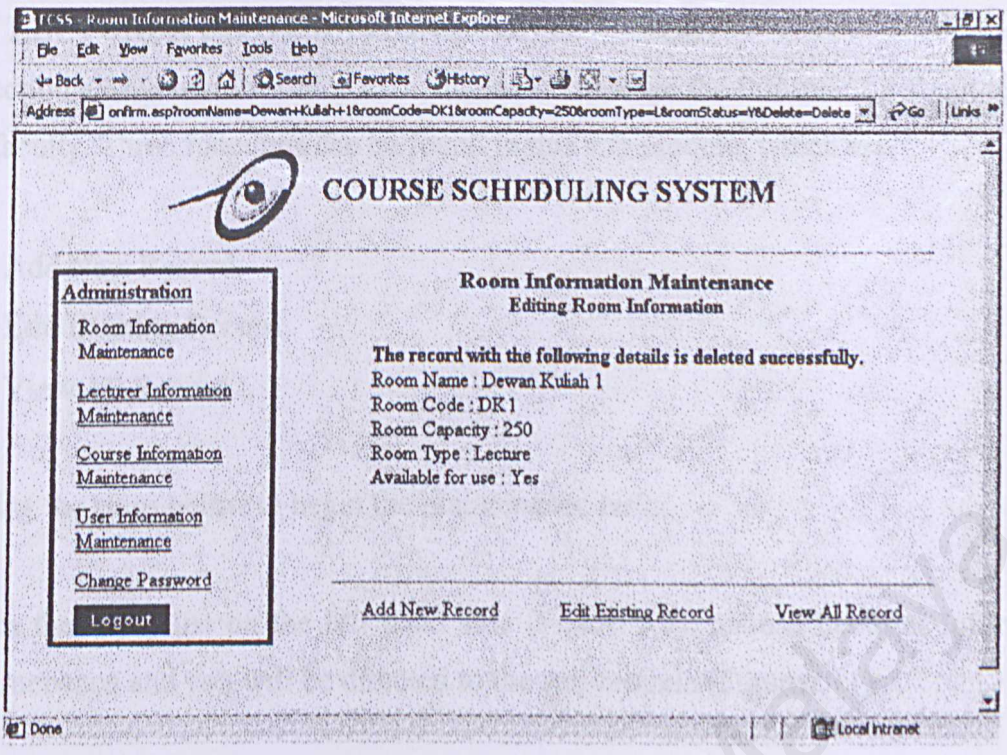


Figure 2.9 Confirmation after deleting a record

7. Click on View All Record if you want to view all records store in database. You will have the chance to view all rooms' record. (Figure 2.10)

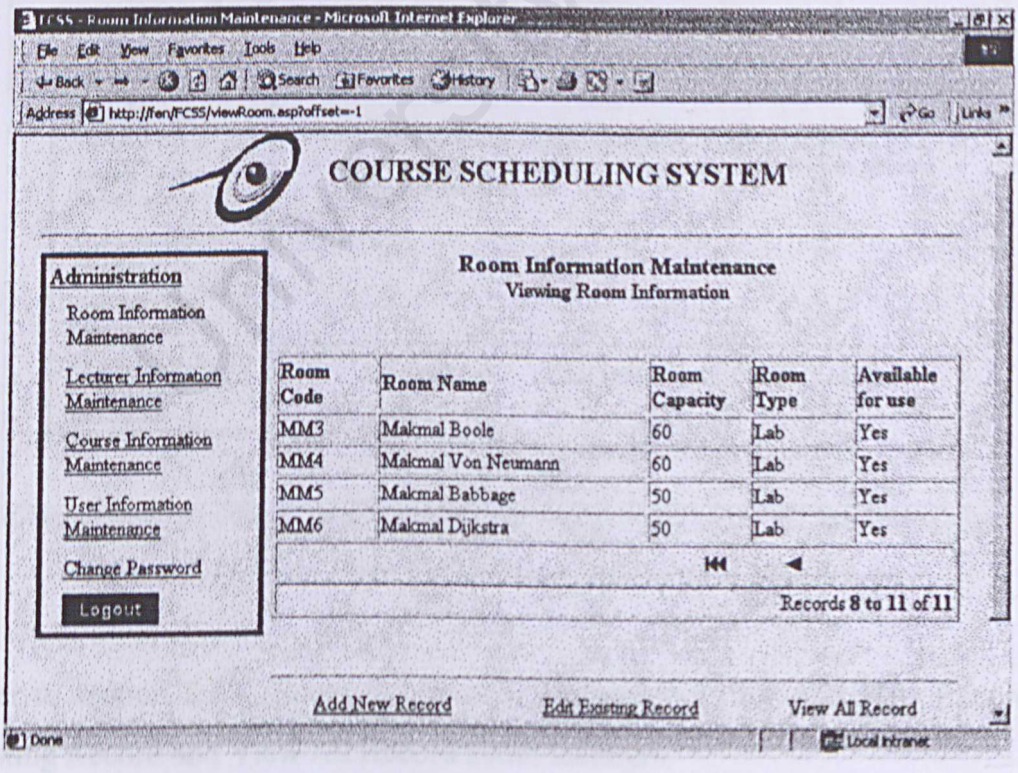


Figure 2.10 Viewing All Records

2.1.3 LECTURER INFORMATION MAINTENANCE

This section allows administrator to maintain and manage the information about lecturers in the faculty. Three functions can be found in this sub-module, which are:

- Add New Record
- Edit Existing Record
- View All Record

Following the steps below to begin the maintenance task:

1. From the menu list on the left hand side (Figure 2.2), choose Lecturer Information Maintenance and you will be directed to the sub-module (Figure2.11).

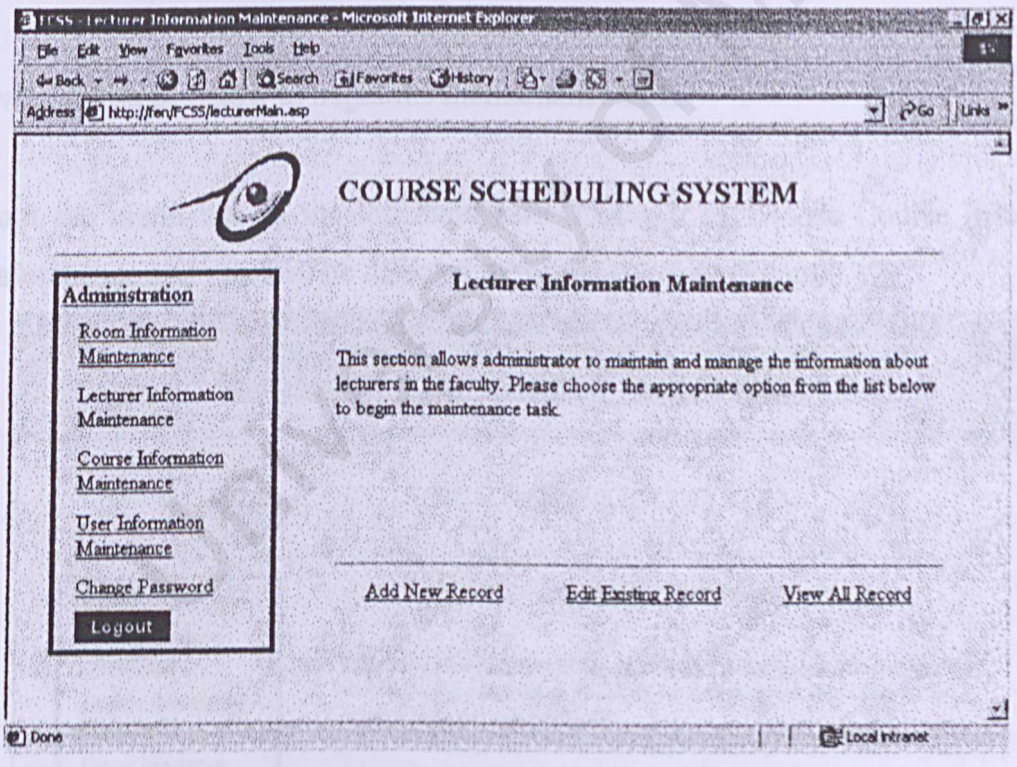


Figure 2.11 Main Page for Lecturer Information Maintenance

2. Refer to section *2.1.2 (Room Information Maintenance)* for Add New Record, Edit Existing Record and View All Record about lecturers.

3. Follow description in Table 2.2 when input the data.

Field	Description
Lecturer Name	Specify lecturer name. E.g.: Pn Norizan Mohd Yasin
Lecturer Code	Specify lecturer code. E.g.: NMY

Table 2.2: Lecturer Information

2.1.4 COURSE INFORMATION MAINTENANCE

This section allows administrator to maintain and manage the general information about courses for all program in FSKTM. Three functions can be found in this sub-module, which are:

- Add New Record
- Edit Existing Record
- View All Record

Following the steps below to begin the maintenance task:

1. From the menu list on the left hand side (Figure 2.2), choose Course Information Maintenance and you will be directed to the sub-module (Figure2.12).

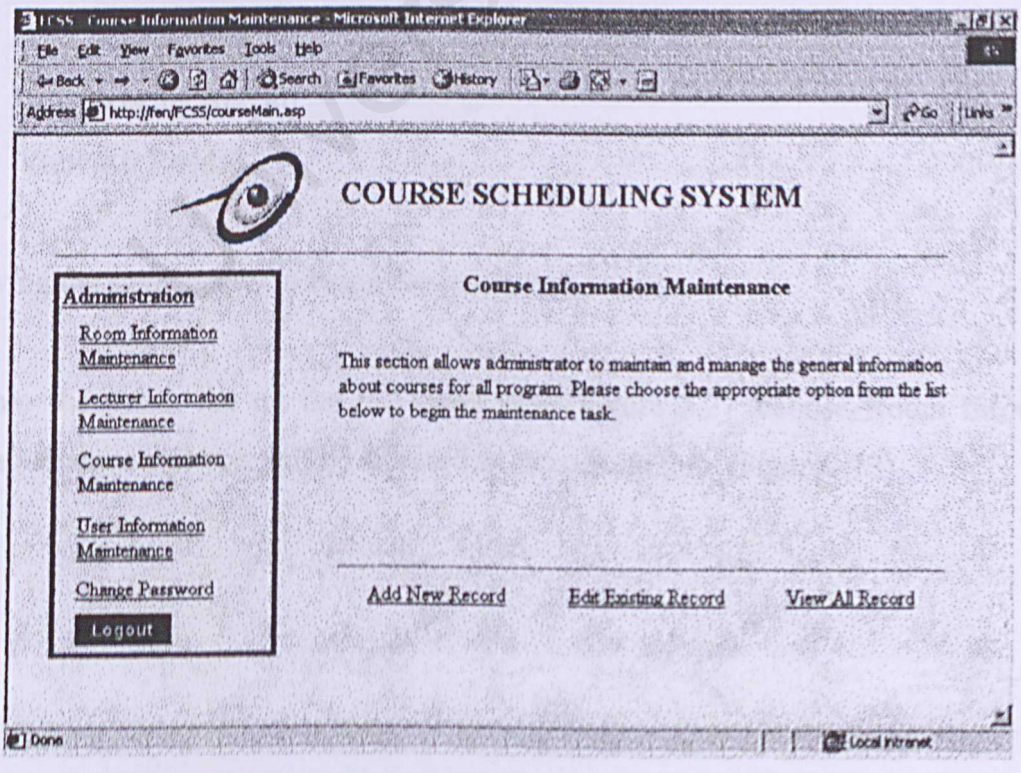


Figure 2.12 Main Page for Course Information Maintenance

2. Refer to section *2.1.2 (Room Information Maintenance)* for Add New Record, Edit Existing Record and View All Record about courses.

3. Follow description in Table 2.3 when input the data.

Field	Description
Course Name	Specify course name. E.g.: Kejuruteraan Perisian
Course Code	Specify course code. E.g.: WXES3204
Credit Hour	Specify credit hour for the course. E.g.: 3
Class Type	Specify whether the course is Lab or Lecture class.

Table 2.3: Course Information

2.1.5 USER INFORMATION MAINTENANCE

This section allows administrator to maintain and manage the information about users for Administration and Course Scheduling modules. Three functions can be found in this sub-module, which are:

- Add New Record
- Edit Existing Record
- View All Record

Following the steps below to begin the maintenance task:

1. From the menu list on the left hand side (Figure 2.2), choose Room Information Maintenance and you will be directed to the sub-module (Figure2.13).

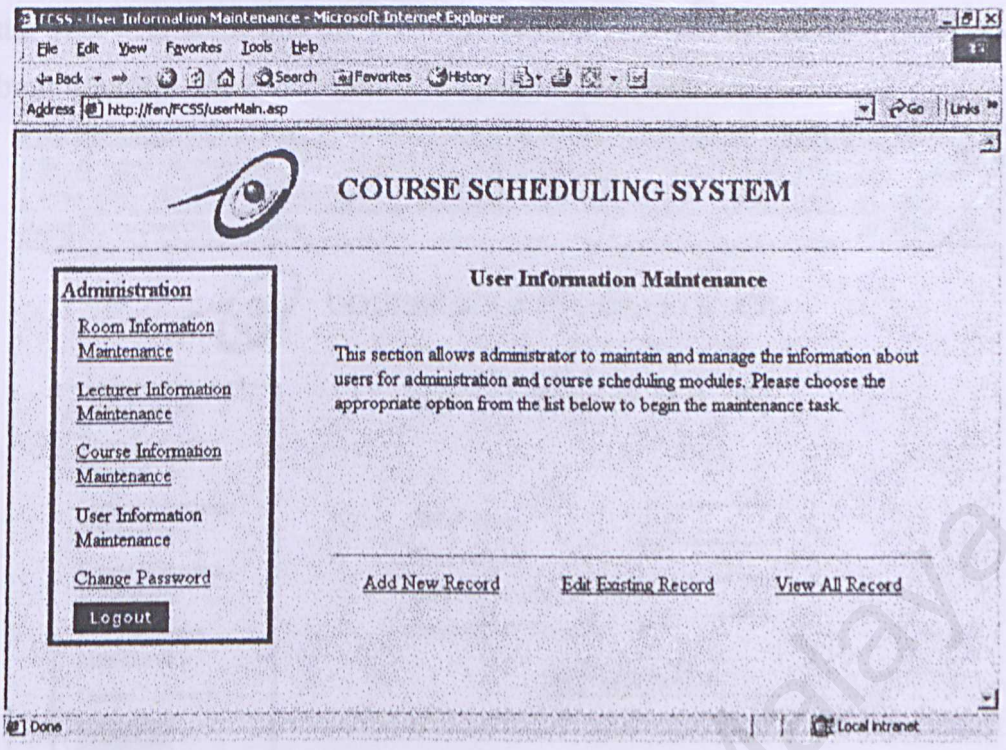


Figure 2.13 Main Page for User Information Maintenance

2. Refer to section *2.1.2 (Room Information Maintenance)* for Add New Record, Edit Existing Record and View All Record about users.

3. Follow description in Table 2.4 when input the data.

Field	**Description**
Name	Specify the full name of the user.
Post	Specify the post of the user. E.g.: administrator
User Name	Specify user name use to login to the system.
Password	Specify password use to login to the system.

Table 2.4: User Information

2.1.6 CHANGE PASSWORD

This function allows you to change your login password. Following the steps below to change your password:

1. From the menu list on the left hand side (Figure 2.2), choose Change Password and you will be directed to the sub-module (Figure2.14).

2. Enter your user name and old password. You must confirm your new password before submit the form. Press Submit to change your password. (Figure 2.14)

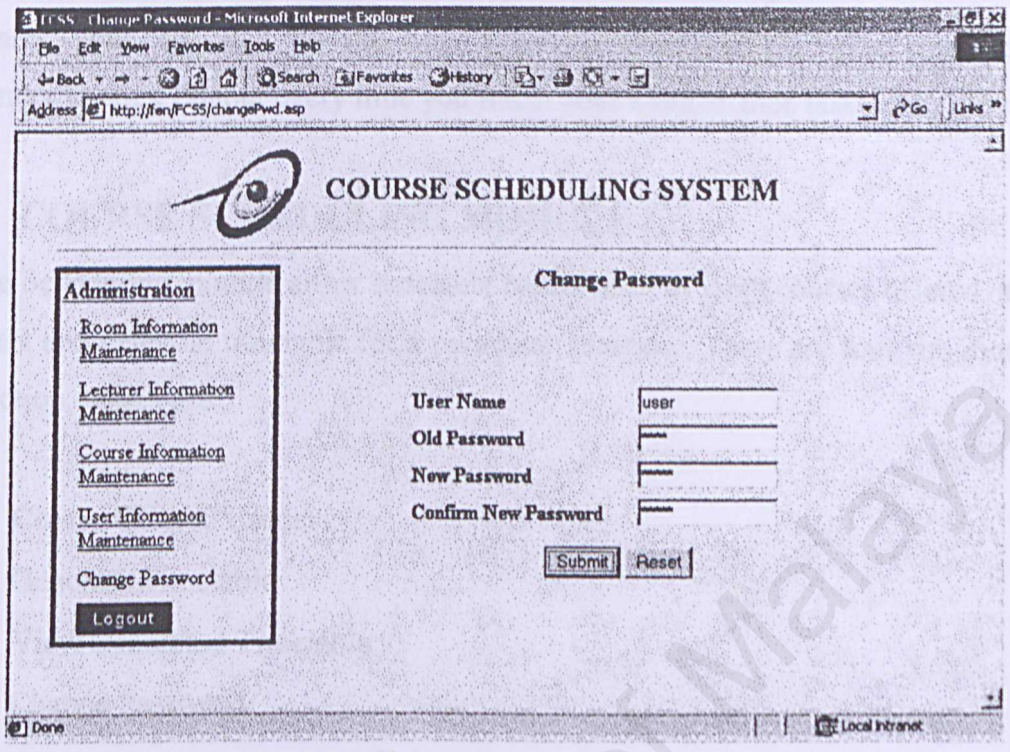


Figure 2.14 Change Password

2. You will be notified if you have changed your password successfully. (Figure 2.15)

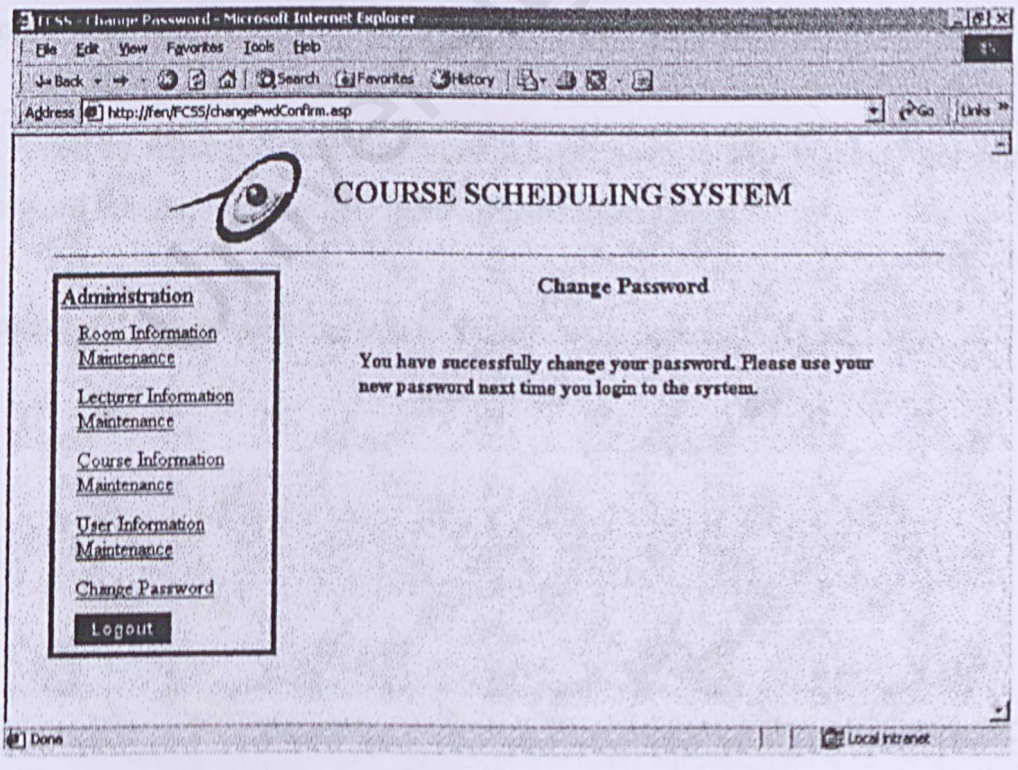


Figure 2.15 Change Password Confirmation

2.1.7 LOGOUT

A user is logout automatically from the system when closing the browser. To logout manually, you can click on the Logout button (Figure 2.2). For security purpose, remember to click Logout every time you finish your maintenance task.

2.2 COURSE SCHEDULING MODULE

Course Scheduling module allow timetable coordinator to input courses offered, assigned lecturers and generate timetable for a particular semester. There are four sub-modules in this section, which are:

- Course Offered Entries
- Generate Timetable
- View Generated Timetable
- Change Password

2.2.1 LOGIN

1. From the FCSS Home Page (Figure 2.1), click on Course Scheduling and you will get a login form. (Figure 2.1)

2. To logon to Course Scheduling module, you have to key in your user name and password into the login form (Figure 2.1) and press Login.

3. A successful logon will bring you to the following screen (Figure 2.16).

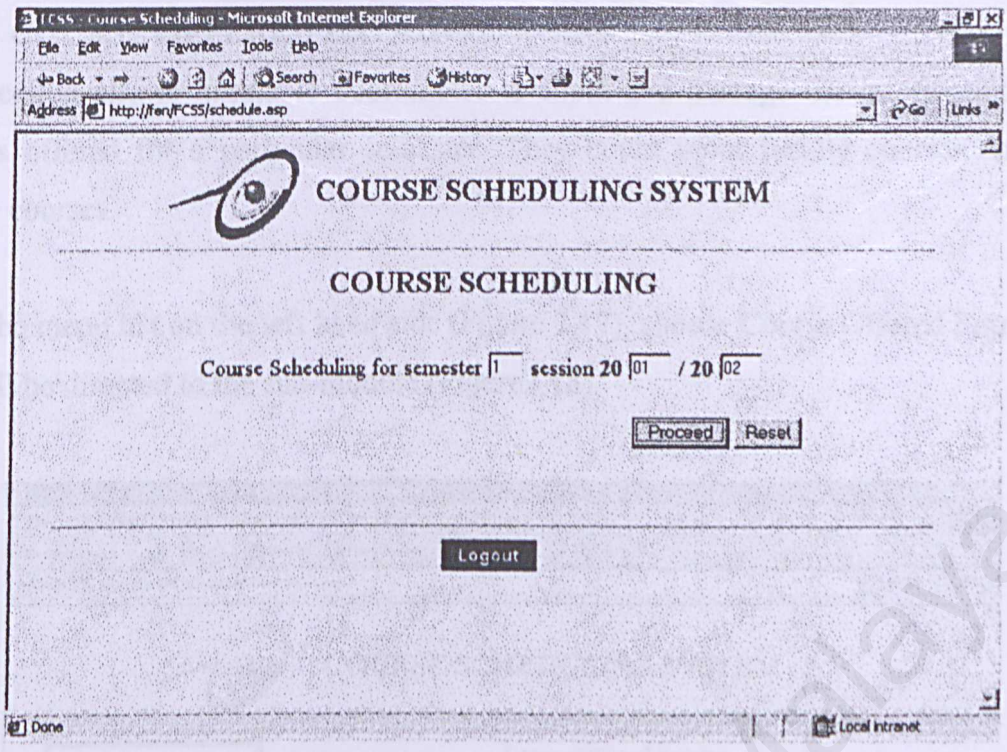


Figure 2.16 Course Scheduling for a particular semester

4. Input corresponding semester and session and press Proceed. You will be directed to Course Scheduling Module. (Figure 2.17)

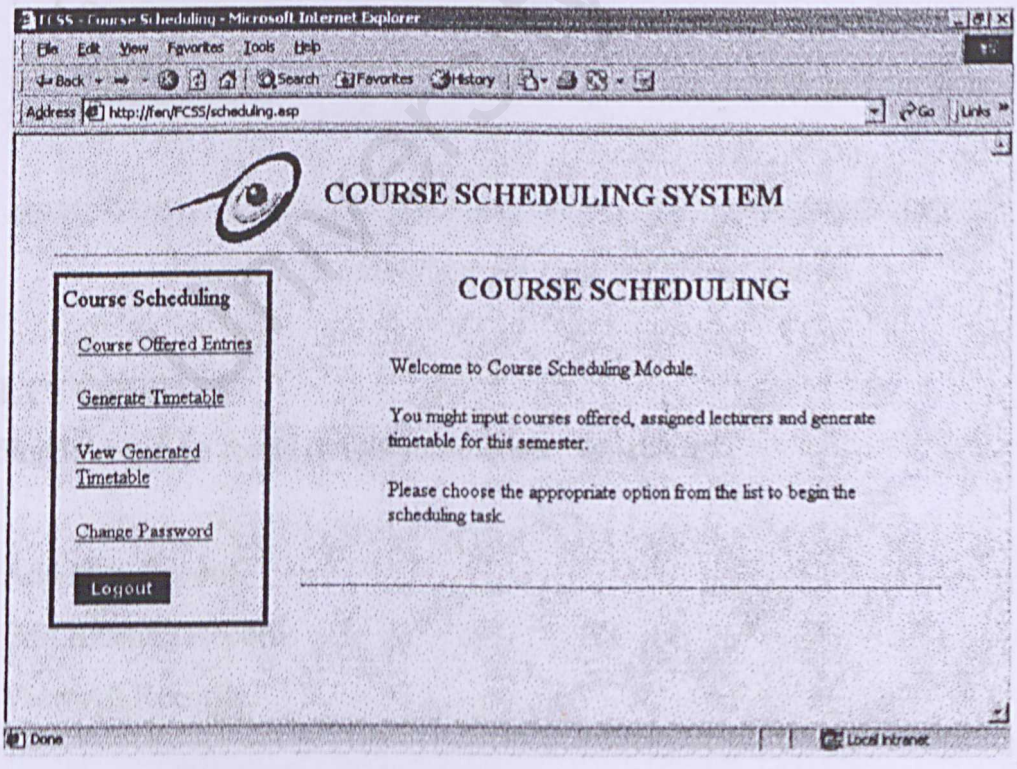


Figure 2.17 Main Page for Course Scheduling Module

2.2.2 COURSE OFFERED ENTRIES

This section allows timetable coordinator to input and manage the information about courses offered for a particular semester. They could input faculty courses and non-faculty courses.

From the menu list on the left hand side (Figure 2.17), choose Course Offered Entries and you will be directed to the sub-module (Figure2.18).

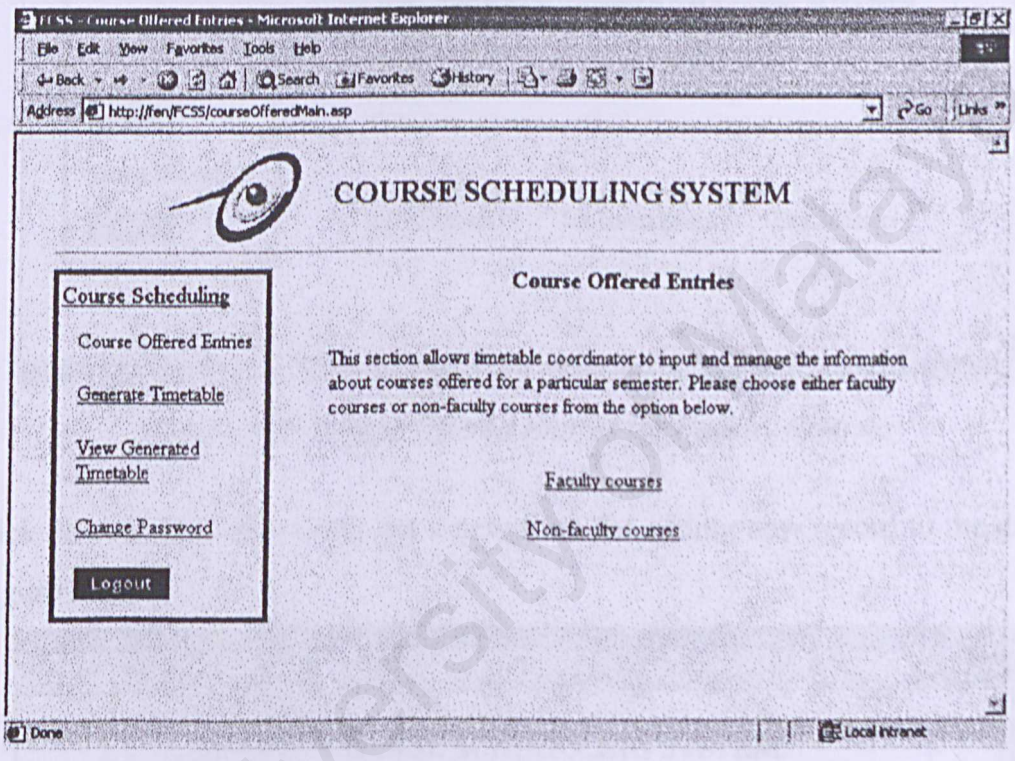


Figure 2.18 Main Page for Course Offered Entries

FACULTY COURSES

Three functions can be found in this sub-module, which are:

- Add New Record
- Edit Existing Record
- View All Record

Follow the steps below to start your work:

1. Click on Faculty courses (Figure 2.18) and you will be directed to the Faculty Courses Offered Entries. (Figure 2.19)

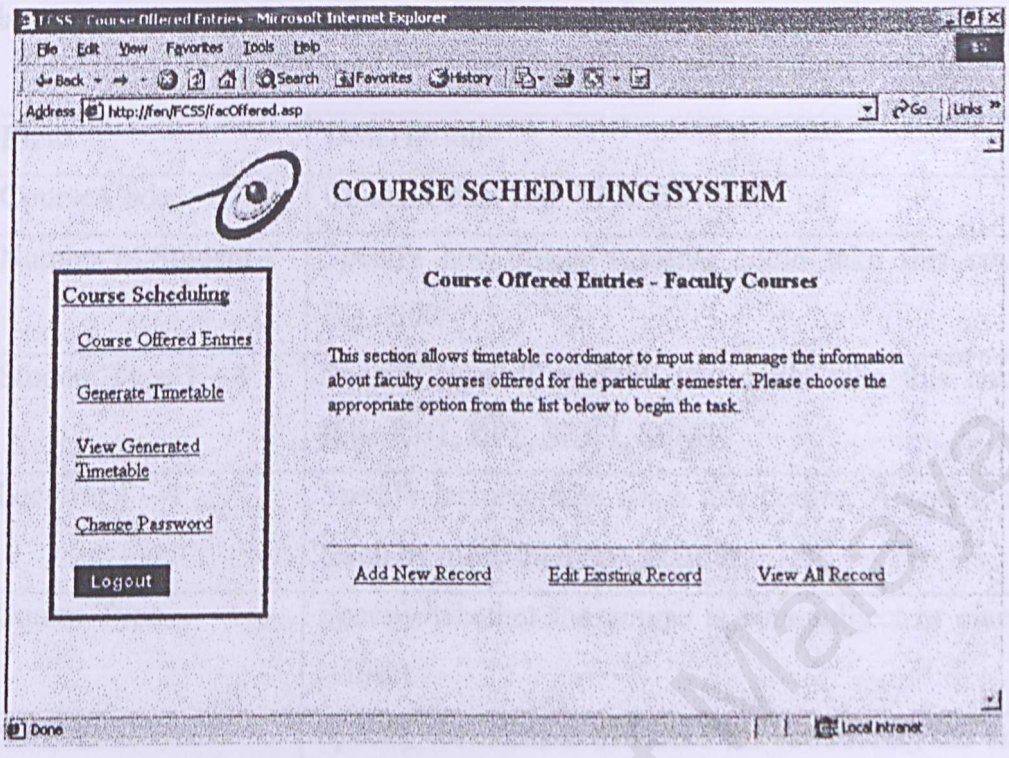


Figure 2.19 Course Offered Entries for Faculty Courses

2. Click on Add New Record to get a new form for adding new record to the database. (Figure 2.20)

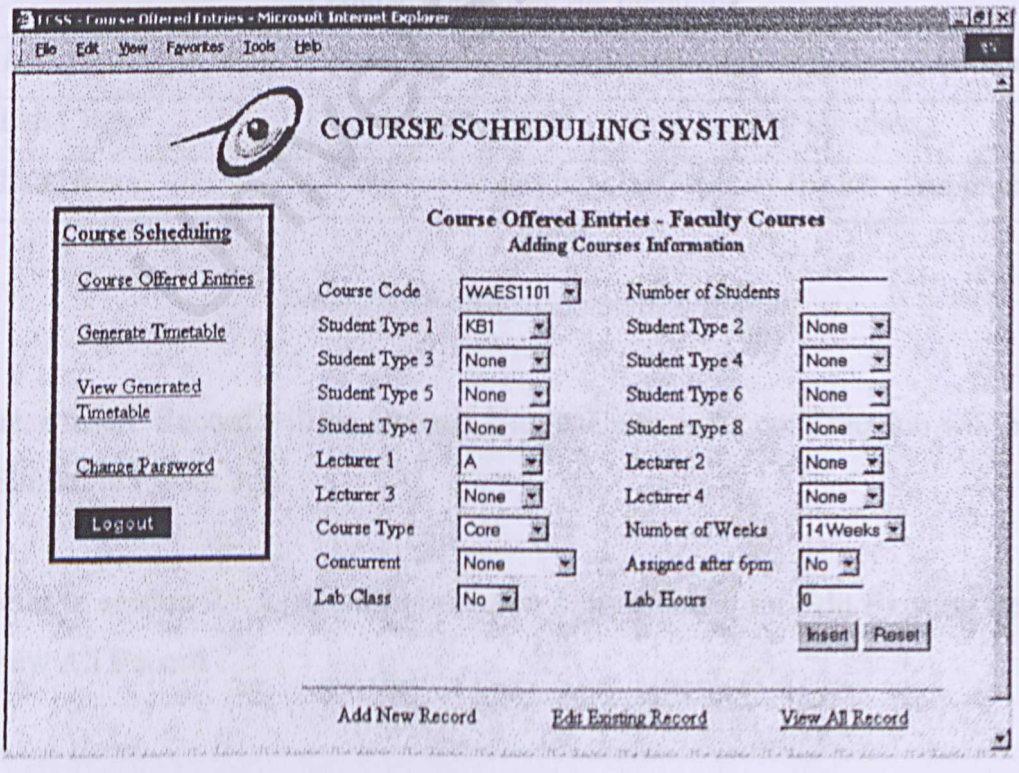


Figure 2.20 Add New Record in Course Offered Entries – Faculty Courses

3. Input necessary data into the form. Follow description in Table 2.5 when input the data.

Field	Description
Course Code	Course Code. E.g.: WXES1101
Number of Student	Specify how many students could take this course. E.g.: 300
Student Type 1 – 8	Specify type of student who could take this course. E.g.: KB1, KP1, SRK1, SPM1
Lecturer 1 – 4	Specify lecturer who teach this course. A course may teach by more than one lecturer.
Course Type	Specify whether the course is core subject or elective subject.
Number of Weeks	Specify the number of weeks for conducting the entire course. It may be 14 weeks or 7 weeks.
Concurrent	Specify whether the course must be assigned at the same time with particular course. If so, choose the course code from the menu list.
Assigned After 6pm	Specify whether the course must be assigned after 6pm.
Lab Class	Specify whether the course has any lab class.
Lab Hours	If the course has lab class, specify the lab class hours.

Table 2.5: Faculty Course Information

3. Press Insert. Record will be inserted into database and a confirmation will be shown. (Similar to Figure 2.5)

4. Refer to section *2.1.2 (Room Information Maintenance)* for Edit Existing Record and View All Record.

NON-FACULTY COURSES

Three functions can be found in this sub-module, which are:

- Add New Record
- Edit Existing Record
- View All Record

Follow the steps below to start your work:

1. Click on Non-faculty courses (Figure 2.18) and you will be directed to the Non-faculty Courses Offered Entries. (Figure 2.21)

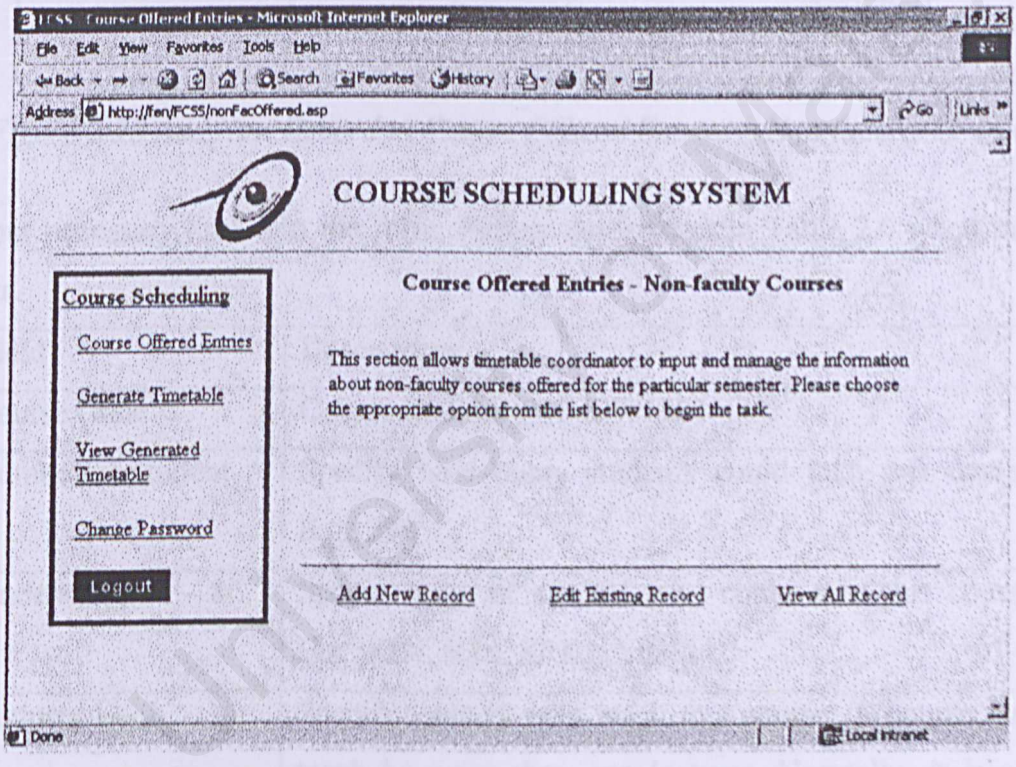


Figure 2.21 Course Offered Entries for Non-faculty Courses

2. Click on Add New Record to get a new form for adding new record to the database. (Figure 2.22)

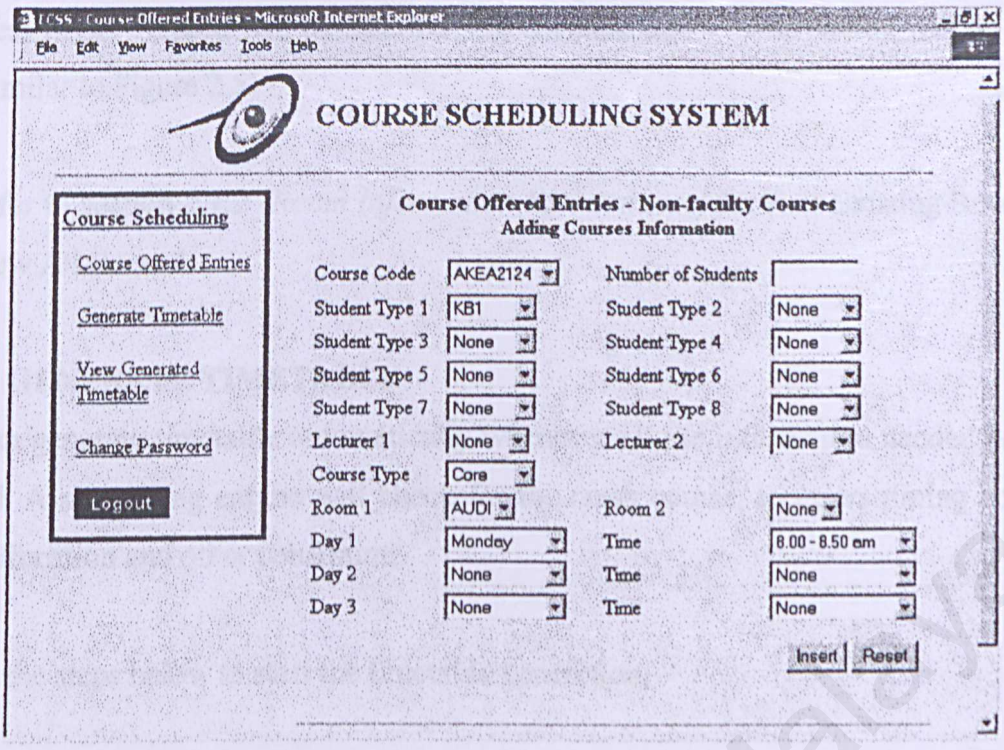


Figure 2.22 Add New Record in Course Offered Entries – Non-faculty Courses

3. Input necessary data into the form. Follow description in Table 2.6 when input the data.

Field	**Description**
Course Code	Course Code. E.g.: SJEW1301
Number of Student	Specify how many students could take this course. E.g.: 300
Student Type 1 – 8	Specify type of student who could take this course. E.g.: KB1, KP1, SRK1, SPM1
Lecturer 1 – 2	Specify lecturer who teach this course. A course may teach by more than one lecturer. Normally, it is not FSKTM's lectuer.
Course Type	Specify whether the course is core subject or elective subject.
Room 1 – 2	Specify the room / place for this course. Normally, the rooms are not in FSKTM. E.g.: K1 in FPP
Day & Time	Specify day and time for the course.

Table 2.6: Non-faculty Course Information

3. Press Insert. Record will be inserted into database and a confirmation will be shown. (Similar to Figure 2.5)

4. Refer to section *2.1.2 (Room Information Maintenance)* for Edit Existing Record and View All Record.

2.2.3 GENERATE TIMETABLE

You can generate timetable automatically whenever all the information needed has been entered. A scheduling engine will check for the crash course, lecturers timing conflicts, room allocation and other constraints.

Follow the steps below to start the timetable generation;

1. From the menu list on the left hand side (Figure 2.17), choose Generate Timetable and you will be directed to the sub-module (Figure2.23).

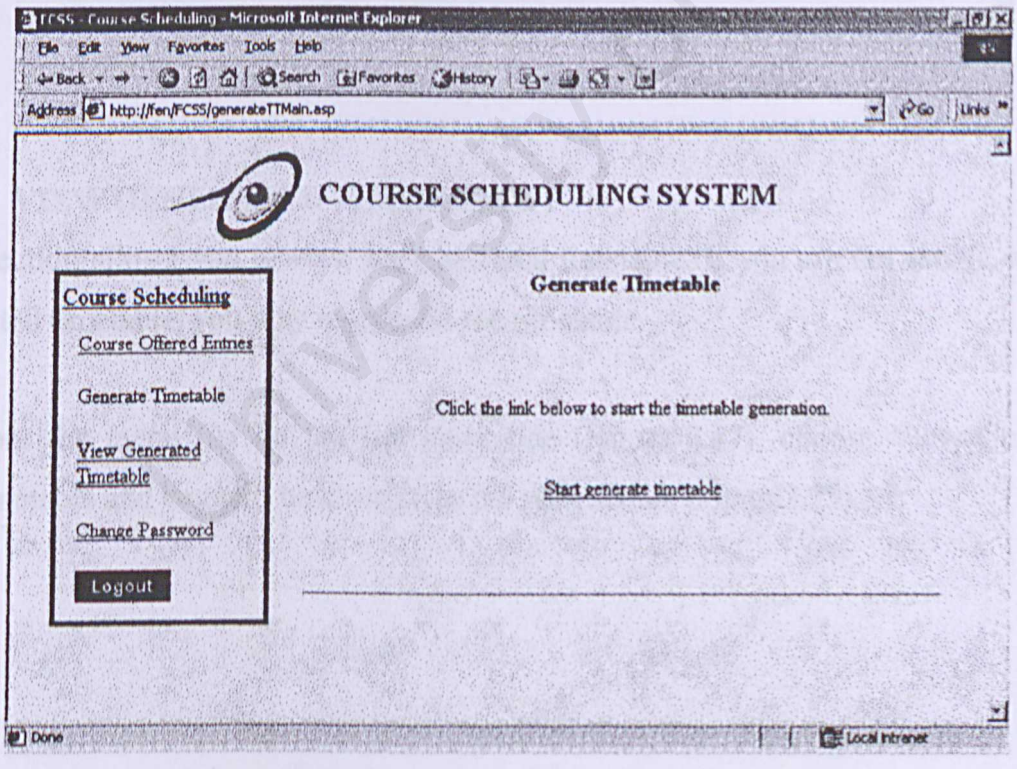


Figure 2.23 Main Page for Generate Timetable

2. Click on Start Generate Timetable. Wait for a moment. It may take some time to generate timetable. The following screen will be displayed after the generation process is successfully. (Figure 2.24)

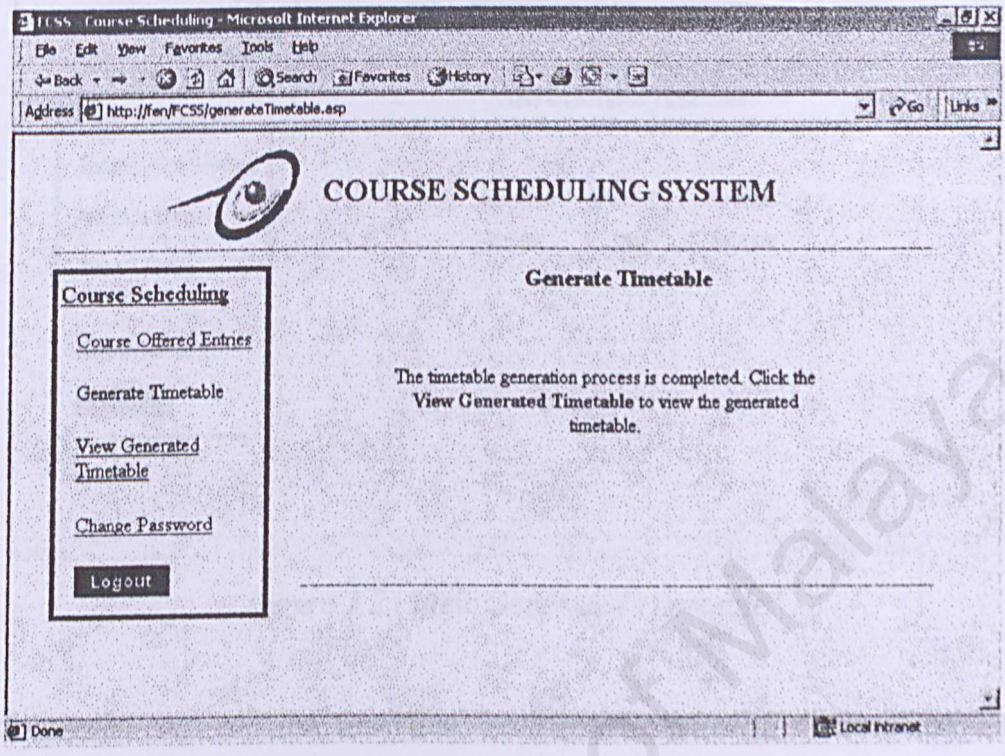


Figure 2.24 Timetable Generation Successfully

2.2.4 VIEW GENERATED TIMETABLE

This section allows you to view the generated timetable. If you are not satisfy with the generated timetable, you may regenerate the timetable.

1. From the menu list on the left hand side (Figure 2.17), choose View Generated Timetable and you will be directed to the sub-module (Figure2.25).

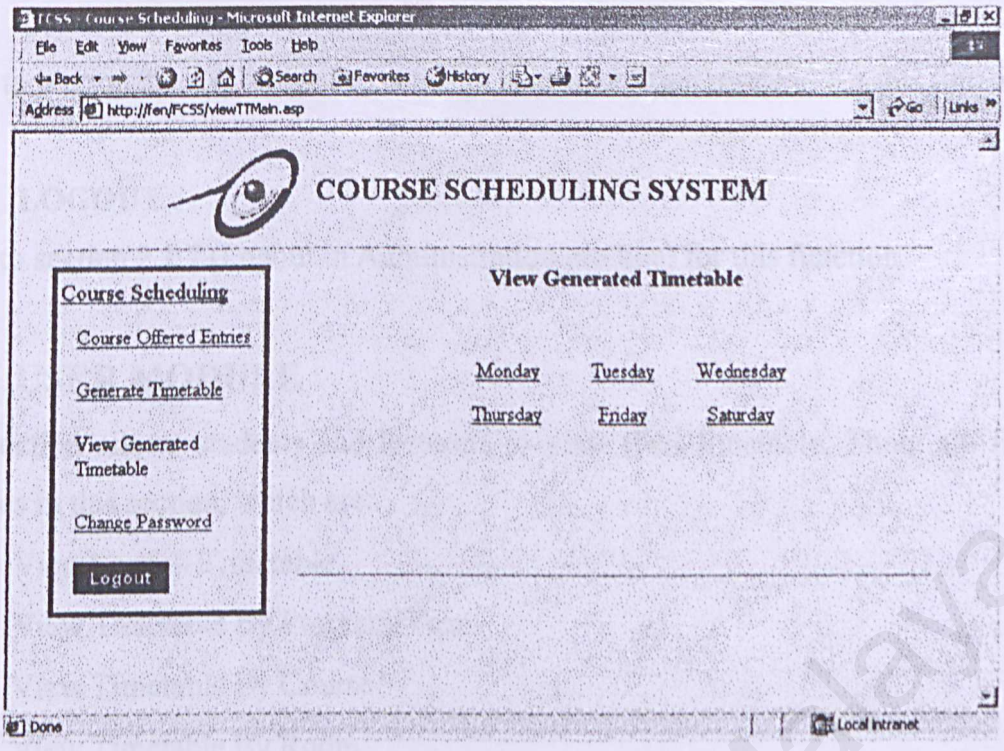


Figure 2.25 View Generated Timetable

2. Click on the day and the timetable for the corresponding day will be displayed. (Figure 2.26)

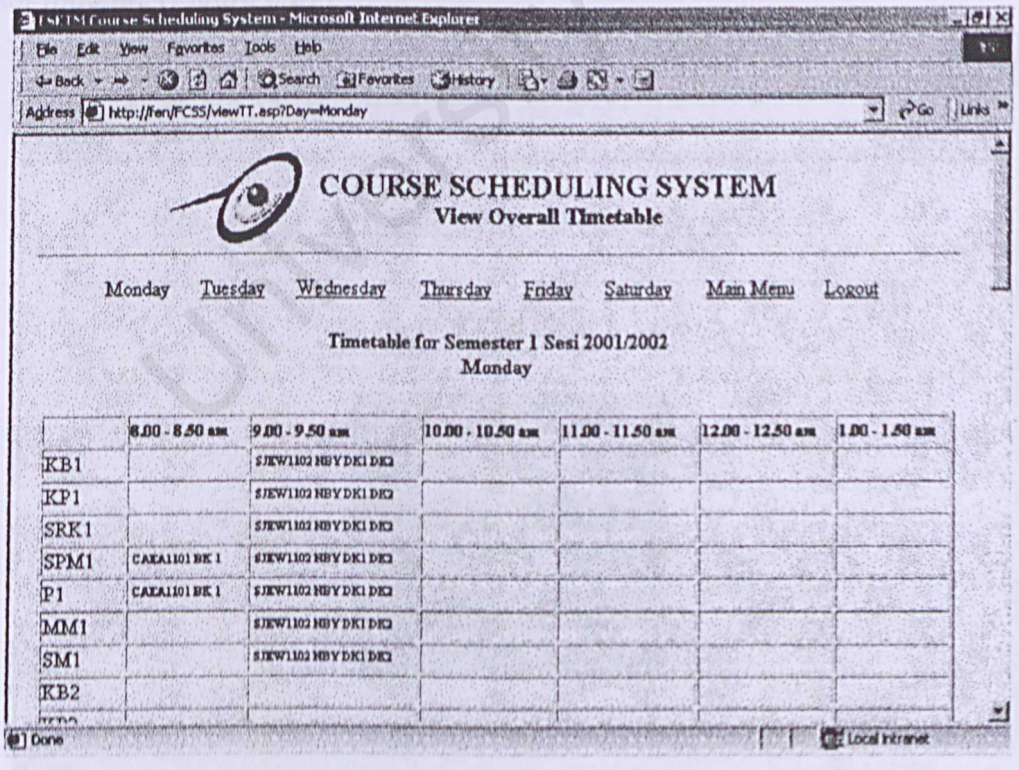


Figure 2.26 Timetable for Monday

2.2.5 CHANGE PASSWORD

Refer to section 2.1.6 (Change Password in Administration module) for this function.

2.2.6 LOGOUT

Refer to section 2.1.7 (Logout in Administration module) for this function.

2.3 USER MODULE

This section allows students and lecturers to view their timetable. There are five sub-modules in this section, which are:

- View Overall Timetable
- View Timetable By Program/Year
- View Timetable By Course
- View Timetable By Room
- View Timetable By Lecturer

1. From the FCSS Home Page (Figure 1.1), click on User and you will be directed to the User module. (Figure 2.27)

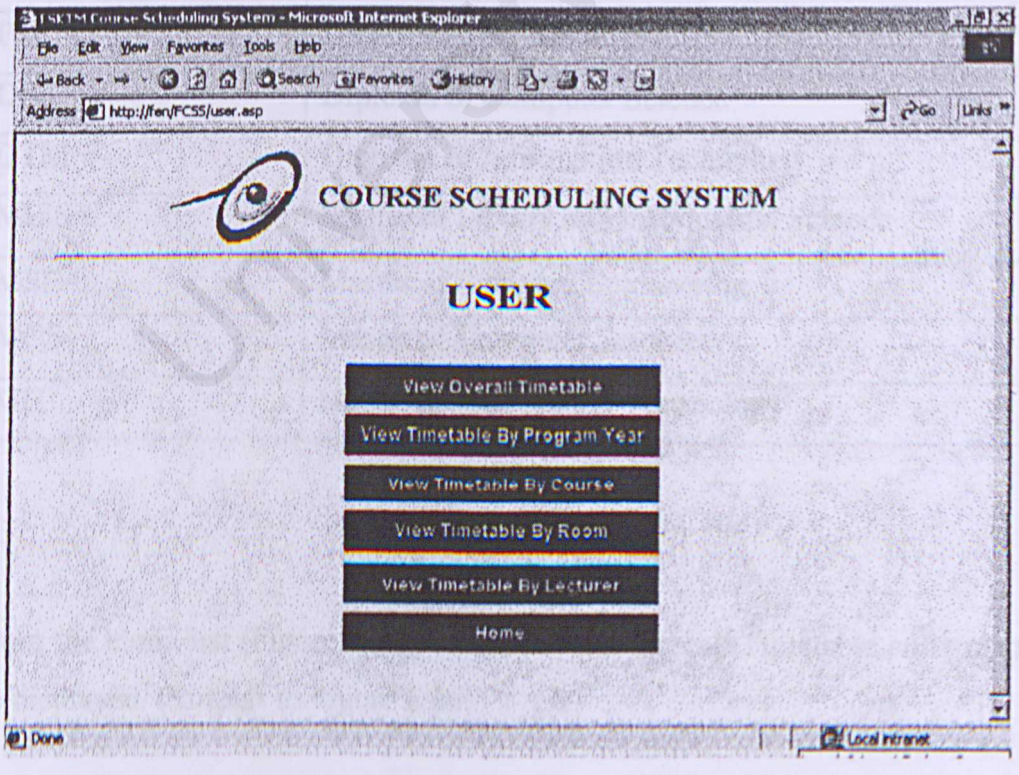


Figure 2.27 Main Page for User Module

2.3.1 VIEW OVERALL TIMETABLE

You can view overall FSKTM timetable sorting by all degree program/year. The code for each program must be known before you can view the timetable. Table 2.7 shows the description for each degree program code.

Degree Program/Year	**Description**
KB1, KB2, KB3	Artificial Intelligence First, Second, Third Year
KP1, KP2, KP3	Software Engineering First, Second, Third Year
SRK1, SRK2, SRK3	Computer Networking and Systems First, Second, Third Year
SPM1, SPM2, SPM3	Management Information Systems First, Second, Third Year
P1, P2, P3	Management (IT) First, Second, Third Year
MM1, MM2, MM3	Multimedia (IT) First, Second, Third Year
SM1, SM2, SM3	Information Science (IT) First, Second, Third Year
Minor IT	Minor IT for other faculties' students
PJJ	Long Distance Learning
DSK	Diploma of Computer Science
DTM	Diploma of Information Technology
MLIS	Master of Library and Information Science
MSE	Master of Software Engineering
MCS	Master of Computer Science
MIT	Master of Information Technology

Table 2.7 Degree Program Code

1. From the menu list (Figure 2.27), click on View Overall Timetable and timetable will be displayed. (Similar to Figure 2.26)

2.3.2 VIEW TIMETABLE BY PROGRAM / YEAR

You can view timetable for a particular degree program / year.

1. From the menu list (Figure 2.27), click on View Timetable By Program / Year. You will be directed to the following screen. (Figure 2.28)

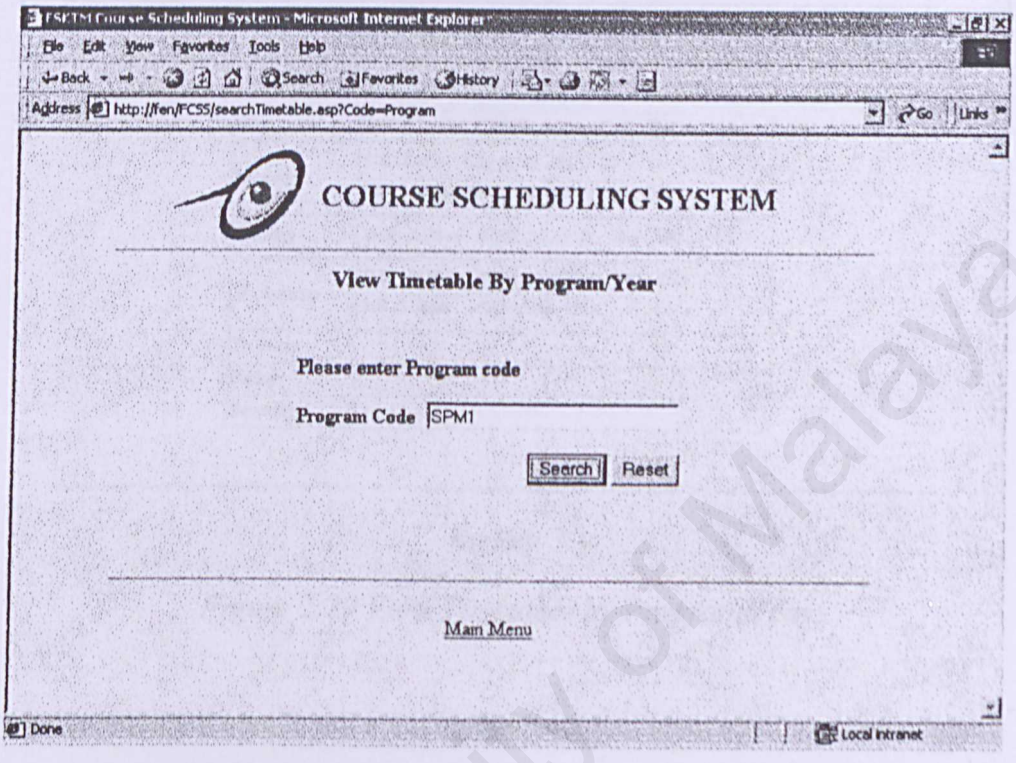


Figure 2.28 Search timetable for a particular Program/Year

2. Enter the program code and press Search. Timetable will be displayed for you in the following look. (Figure 2.29)

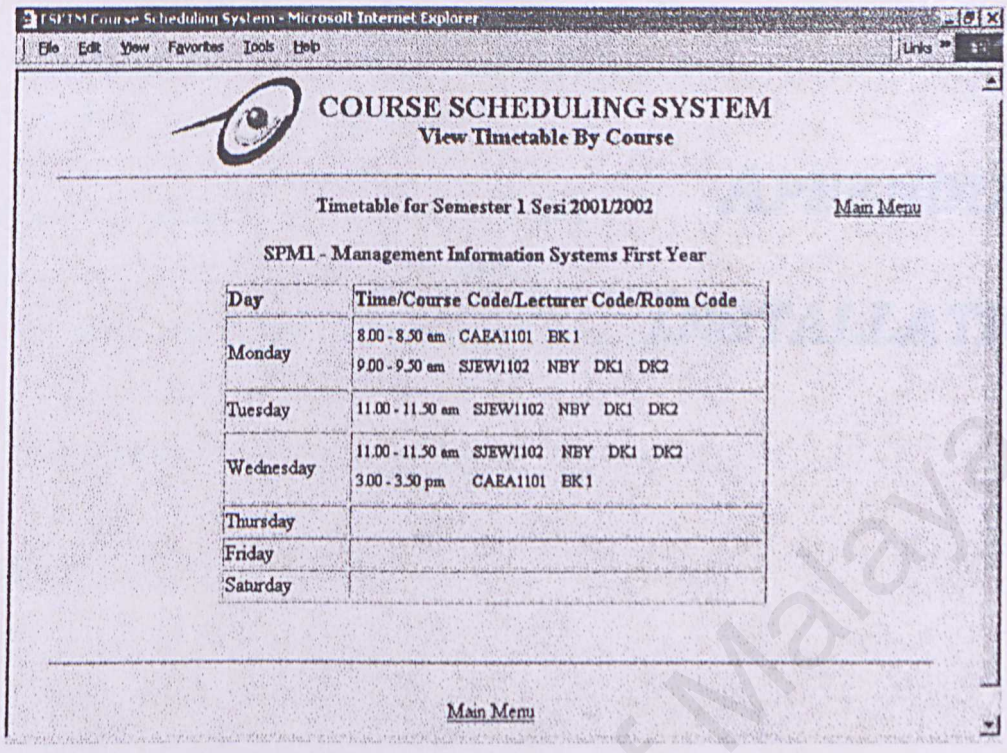


Figure 2.29 View Timetable by Program/Year

2.3.3 VIEW TIMETABLE BY COURSE

You can view timetable for a particular course. Refer to section 2.3.2 (View Timetable By Program/Year) for this function.

2.3.4 VIEW TIMETABLE BY ROOM

You can view timetable for a particular room. Refer to section 2.3.2 (View Timetable By Program/Year) for this function.

2.3.5 VIEW TIMETABLE BY LECTURER

You can view timetable for a particular lecturer. Refer to section 2.3.2 (View Timetable By Program/Year) for this function.

APPENDIX B

INSTALLATION

INSTALLATION

To set up ***FCSS (FSKTM Course Scheduling System)*** in the server, make sure your server meets the following requirements.

Hardware Requirements

The hardware required for running FCSS are:

- A server with at least Pentium 166 Mhz processor
- At least 64MB RAM
- Network Interface Card (NIC) and network connection

Software Requirements

The software required for running FCSS are:

- Microsoft Windows 2000
- Internet Information Server 3.0 or higher
- Microsoft SQL Server 2000

Follow the steps below to begin your setup:

1. Install Internet Information Services (IIS) from your window package and Microsoft SQL Server 2000.

2. Create a new folder in *C:\Inetpub\wwwroot*. Named the folder as *FCSS*. Your path for your new folder should be *C:\Inetpub\wwwroot\FCSS*.

3. From the CD, open folder *Database*. Copy the file named *Fcss* into *C:\Inetpub\wwwroot\FCSS*.

4. Open SQL Server Enterprise Manager. Create a new database named *fcss*.

5. Right click on the database, select All Tasks -> Restore Database (Figure 1). A dialog box as shown in Figure 2 will be prompted out.

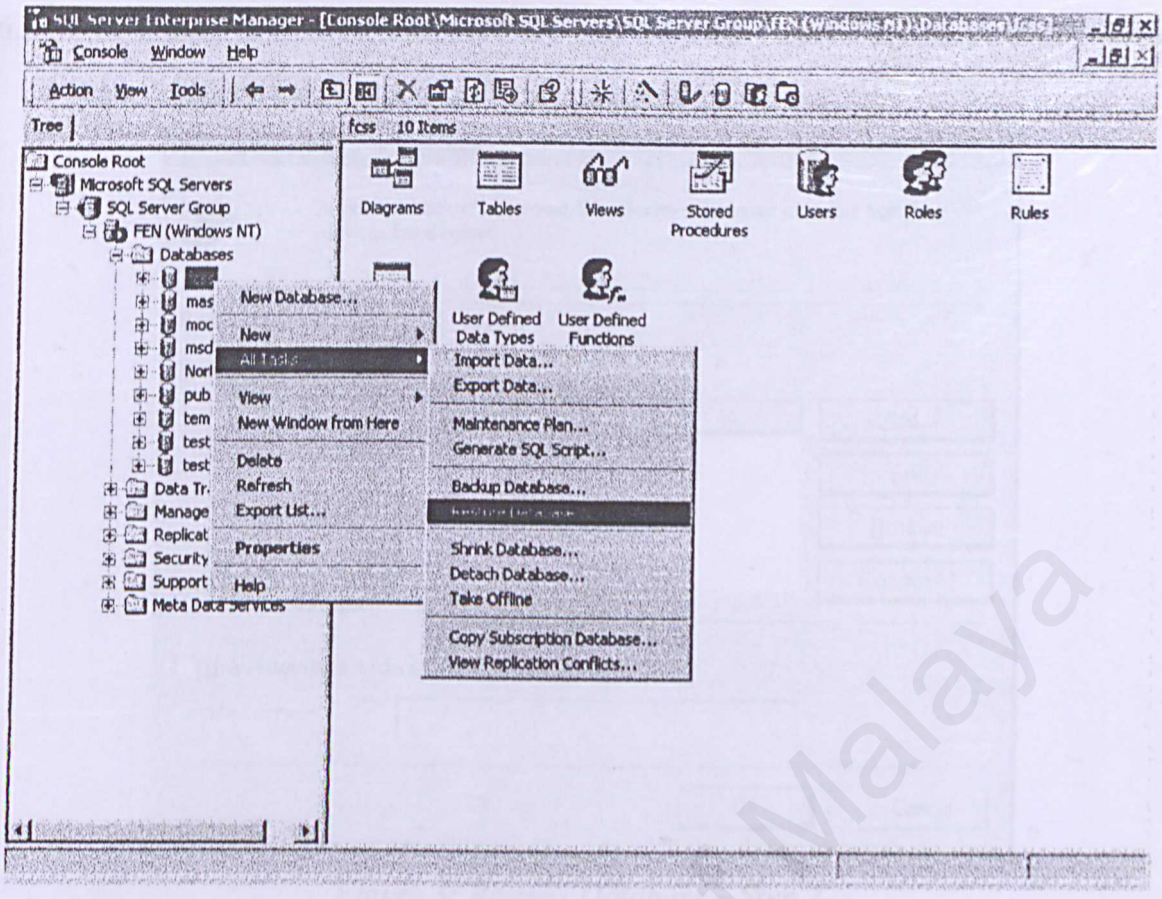


Figure 1: Restore Database

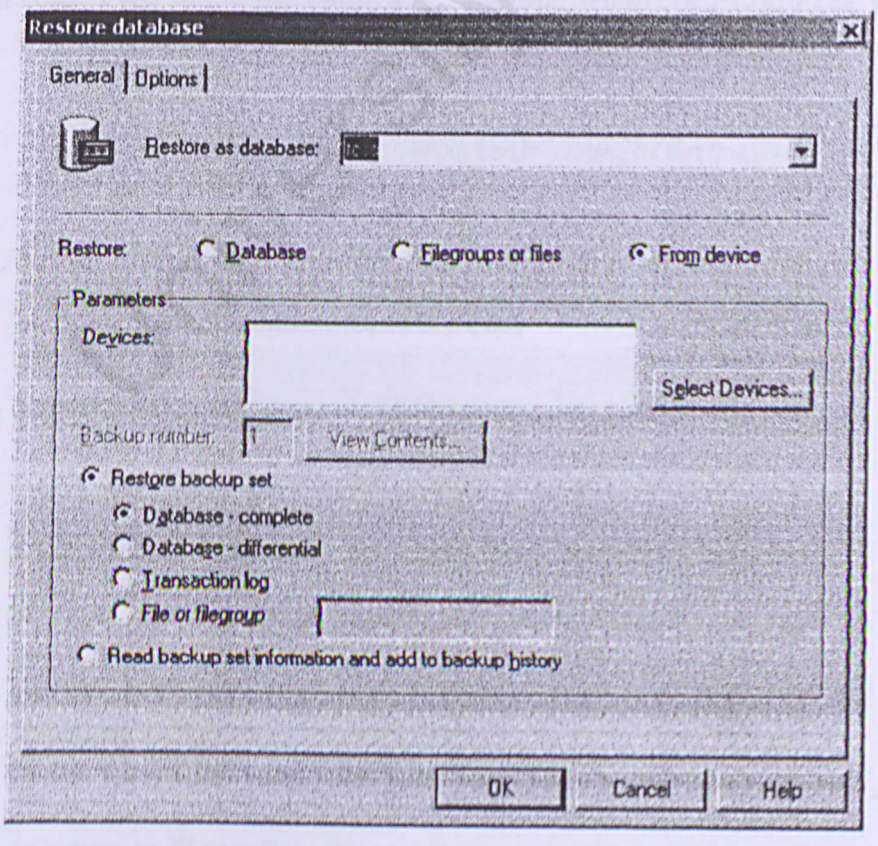


Figure 2: Restore Database – Step 1

6. In the dialog box, select From Device and click on Select Devices. A dialog box as shown in Figure 3 is prompted out.

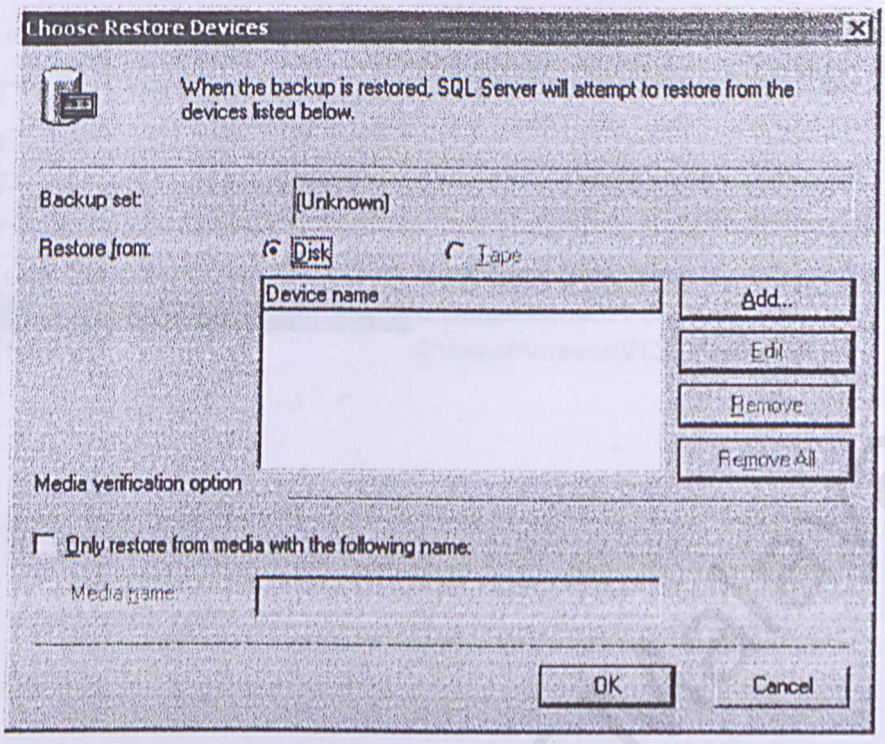


Figure 3: Restore Database – Step 2

7. Click on Add. Type in the file name as *C:\Inetpub\wwwroot\FCSS\Fcss*. (Figure 4)

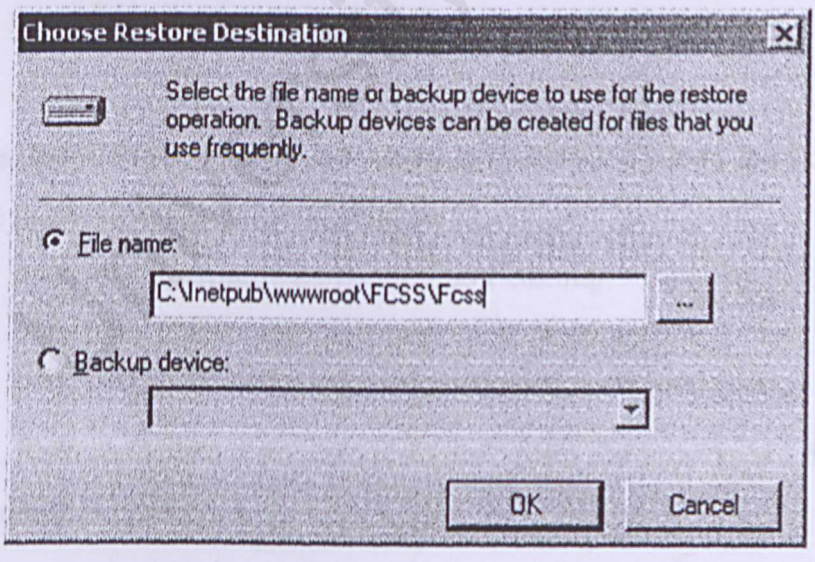


Figure 4: Restore Database – Step 3

8. Click Ok to return to Figure 3.

9. Click Ok to return to Figure 2.

10. From Figure 2, select the tab Options. You will get the screen as Figure 5.

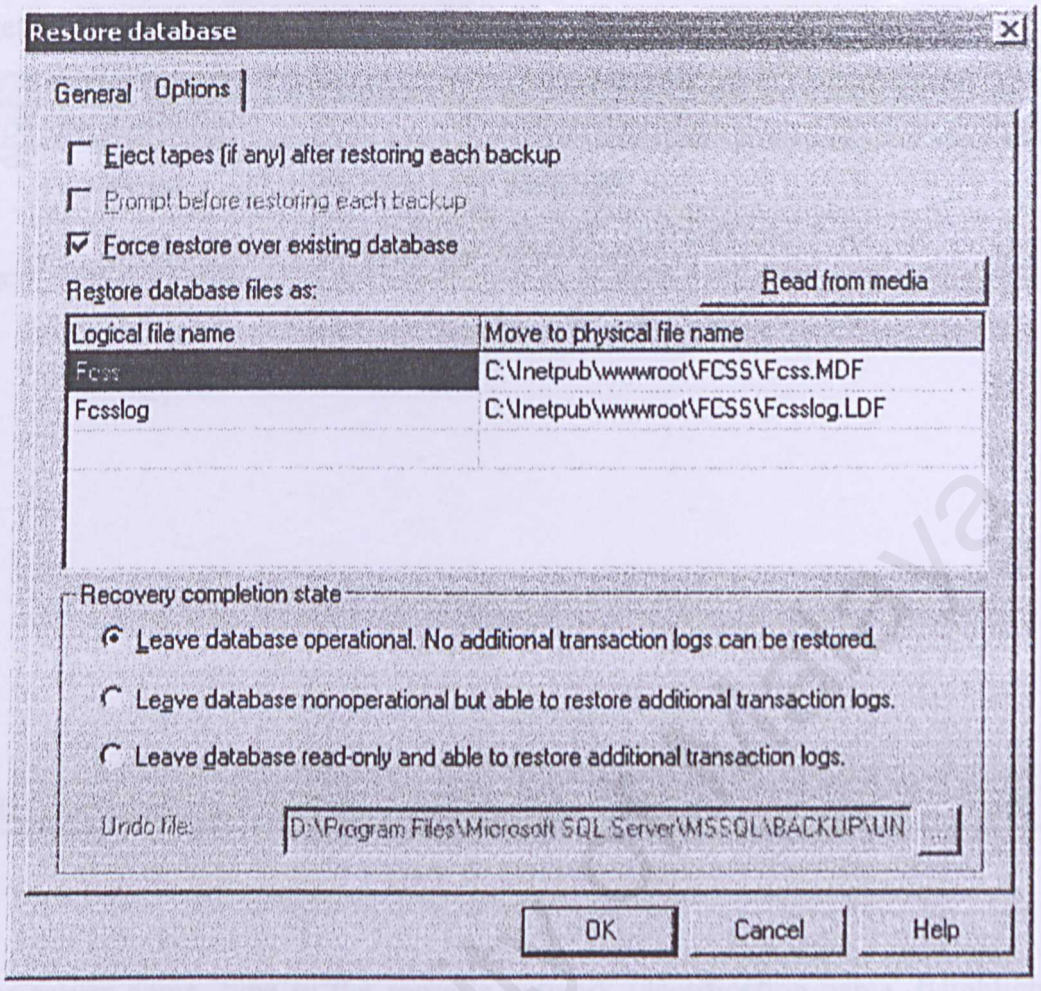


Figure 5: Restore Database – Step 4

11. Check the option Force restore over existing database. Change the Logical file name and Move to physical file name as: (Figure 5)

    Fcss – C:\Inetpub\wwwroot\FCSS\Fcss.mdf

    Fcsslog – C:\Inetpub\wwwroot\FCSS\Fcsslog.ldf

12. Click OK. Restore Database process is started. After the process is finished, click Ok and the process is competed.

13. From the CD, open folder *Program Files.* Copy all files into *C:\Inetpub\wwwroot\FCSS.*

14. From *C:\Inetpub\wwwroot\FCSS*, open file *DataStore.asp* using Notepad. You will get the screen as shown in Figure 6.

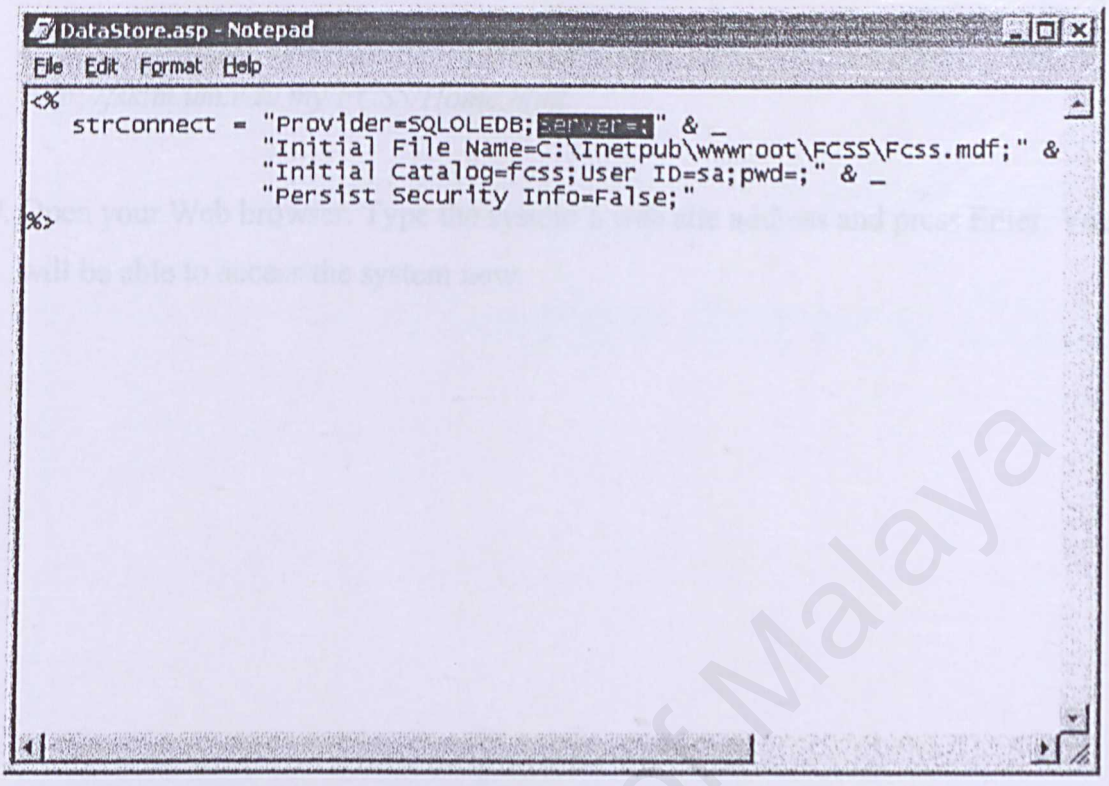


Figure 6: DataStore.asp

15. For server name, type in the server's network identification name. (Highlighted in Figure 6). If the server name is *fsktm.um.edu.my*, the file should be as follow:

```
<%
        strConnect = "Provider=SQLOLEDB;Server=fsktm.um.edu.my;" & _
                     "Initial File Name=C:\Inetpub\wwwroot\FCSS\Fcss.mdf;" & _
                     "Initial Catalog=fcss;User ID=sa;pwd=;" & _
                     "Persist Security Info=False;"
%>
```

16. Save the file and then close the file.

17. Open Internet Information Services administration tool and create a virtual directory named *FCSS* that point to the folder *C:\Inetpub\wwwroot\FCSS*. (Refer to Internet Information Services handout if you do not know how to create a virtual directory.)

18. The web address for the system is: ***http://server_name/FCSS/Home.html*** where server_name is the server's network identification name. If the server is *fsktm.um.edu.my* then the web address for the system should be *http://fsktm.um.edu.my/FCSS/Home.html*

19. Open your Web browser. Type the system's web site address and press Enter. You will be able to access the system now.

APPENDIX C

SAMPLE SOURCE CODE

SAMPLE SOURCE CODE

Here is a sample source code from FCSS program.

Display Timetable By Lecturer (viewByLecturer.asp)

```
<%
    Option Explicit
    Dim strConnect
%>

<!-- #include file="DataStore.asp" -->
<!-- #include file="MetaData.asp" -->

<%
    Dim D(6), T(12), j, k, s

    D(0) = "Monday"
    D(1) = "Tuesday"
    D(2) = "Wednesday"
    D(3) = "Thursday"
    D(4) = "Friday"
    D(5) = "Saturday"

    T(0) = "8.00 - 8.50 am"
    T(1) = "9.00 - 9.50 am"
    T(2) = "10.00 - 10.50 am"
    T(3) = "11.00 - 11.50 am"
    T(4) = "12.00 - 12.50 pm"
    T(5) = "1.00 - 1.50 pm"
    T(6) = "2.00 - 2.50 pm"
    T(7) = "3.00 - 3.50 pm"
    T(8) = "4.00 - 4.50 pm"
    T(9) = "5.00 - 5.50 pm"
    T(10) = "6.00 - 8.50 pm"
%>

<%
    Dim objRSSesi, semester, sesi

    Set objRSSesi = Server.CreateObject("ADODB.Recordset")
    objRSSesi.open "Semester_Sesi", strConnect, adOpenForwardOnly,
                   adLockReadOnly, adCmdTable

    if not objRSSesi.EOF then
        semester = objRSSesi("semester")
```

```
        sesi = objRSSesi("sesi")
    End IF

    objRSSesi.Close
    Set objRSSesi = Nothing
%>

<%
    Dim objRSLecturer, LecturerCode, LecturerName

    Set objRSLecturer = Server.CreateObject("ADODB.Recordset")
    objRSLecturer.open "Lecturer_Info", strConnect, adOpenForwardOnly,
                       adLockReadOnly, adCmdTable
    objRSLecturer.filter = "lecturer_code like '" & Request("searchCode") & "'"

    if not objRSLecturer.EOF then
        LecturerCode = objRSLecturer("lecturer_code") & " - "
        LecturerName = objRSLecturer("lecturer_name")
    End IF

    objRSLecturer.Close
    Set objRSLecturer = Nothing
%>

<%
    Dim objRS, strQuery, mesej, objRS2, strQuery2

    Set objRS = Server.CreateObject("ADODB.Recordset")

    strQuery = "Select Time_Slot.*, course_name, lecturer_name, room_name,
               timeValue, dayValue from Time_Slot, Course_Info, Lecturer_Info,
               Room_Info, Time_Value, Course_Offered where
               Time_Slot.course_code = Course_Info.course_code and
               Time_Slot.lecturer_code = Lecturer_Info.lecturer_code and
               Time_Slot.room_code = Room_Info.room_code and
               Time_Slot.course_time = Time_Value.course_time and
               Time_Slot.course_code = Course_Offered.course_code and facPaper = 'Y'
               and Time_Slot.lecturer_code = '" & Request("searchCode") & "' Order by
               dayValue, timeValue, Time_Slot.course_code, Time_Slot.room_code"

    objRS.Open strQuery, strConnect, adOpenForwardOnly, adLockReadOnly,
               adCmdText

    Set objRS2 = Server.CreateObject("ADODB.Recordset")

    strQuery2 = "Select Non_Fac.*, course_name from Non_Fac, Course_Info,
                Course_Offered where
                Non_Fac.course_code =  Course_Info.course_code and
```

```
            Non_Fac.course_code = Course_Offered.course_code and
            facPaper = 'N' and ( lecturer1 = '" & Request("searchCode") & "' or
            lecturer2 = '" & Request("searchCode") & "' )
            Order by Non_Fac.course_code"

    objRS2.Open strQuery2, strConnect, adOpenForwardOnly, adLockReadOnly,
            adCmdText
    If objRS.EOF and objRS2.EOF then
        mesej = "There are no timetable information for " & Request("searchCode")

ElseIf not objRS.EOF then

        Dim course_time(150), course_code(150), lecturer_code(150), room_code(150)
        Dim dayValue(150), course_name(150), lecturer_name(150), room_name(150)
        Dim timeValue(150), i

        i = 0

        Do While not objRS.EOF          'store information in array

            course_time(i) = objRS("course_time")
            course_code(i) = objRS("course_code")
            lecturer_code(i) = objRS("lecturer_code")
            room_code(i) = objRS("room_code")
            dayValue(i) = objRS("dayValue")
            course_name(i) = objRS("course_name")
            lecturer_name(i) = objRS("lecturer_name")
            room_name(i) = objRS("room_name")
            timeValue(i) = objRS("timeValue")

            i = i + 1

            objRS.MoveNext

        Loop

    End If

    If not objRS2.EOF then

        Dim nfCode(50), nfLecturer1(50), nfLecturer2(50), nfRoom1(50),nfRoom2(50)
        Dim day1(50), day2(50), day3(50), time1(50), time2(50), time3(50), nfName(50)
        Dim a

        a = 0

        Do While not objRS2.EOF         'store information in array
```

```
            nfCode(a) = objRS2("course_code")
            nfLecturer1(a) = objRS2("lecturer1")
            nfLecturer2(a) = objRS2("lecturer2")
            nfRoom1(a) = objRS2("room1")
            nfRoom2(a) = objRS2("room2")
            day1(a) = objRS2("day1")
            day2(a) = objRS2("day2")
            day3(a) = objRS2("day3")
            time1(a) = objRS2("time1")
            time2(a) = objRS2("time2")
            time3(a) = objRS2("time3")
            nfName(a) = objRS2("course_name")

            a = a + 1

            objRS2.MoveNext

        Loop
    End If

    objRS.close
    Set objRS = Nothing
    objRS2.close
    Set objRS2 = Nothing

%>

<html><!-- #BeginTemplate "/Templates/Template.dwt" -->
<head>
<!-- #BeginEditable "doctitle" -->
<title>FSKTM Course Scheduling System</title>
<!-- #EndEditable -->
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#FFFFFF" text="#000000" background="Light%20blue2.jpg">
<!-- #BeginEditable "Content" -->
<table width="757" border="0" cellpadding="0" cellspacing="0">
 <tr>
  <td width="22" height="11"></td>
  <td width="75"></td>
  <td width="17"></td>
  <td width="109" rowspan="3" valign="top"><img src="Logo%20FSKTM.gif"
width="105" height="70"></td>
  <td width="388"></td>
  <td width="29"></td>
  <td width="93"></td>
```

```
    <td width="24"></td>
   </tr>
   <tr>
    <td height="48"></td>
    <td></td>
    <td></td>
    <td valign="top">
     <div align="center"><font color="#0033CC" face="Times New Roman, Times, serif"
size="4"><b><font size="3" color="#990000"><font color="#0033CC" size="5">
      COURSE SCHEDULING SYSTEM<br>
      </font><font size="6" color="#0033CC"><b><font size="4">View Timetable
      By Course</font></b></font></font></b></font></div>
    </td>
    <td></td>
    <td></td>
    <td></td>
   </tr>
   <tr>
    <td height="11"></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
   </tr>
   <tr>
    <td height="21"></td>
    <td colspan="6" valign="top">
     <hr>
    </td>
    <td></td>
   </tr>
   <tr>
    <td height="69"></td>
    <td></td>
    <td colspan="4" valign="top">
     <div align="center"><font size="3" color="#0033CC"><b>Timetable for Semester
      <% = semester & " Sesi " & sesi & "<br>" %>
      <br><% = LecturerCode & LecturerName %>
      </b></font></div>
    </td>
    <td valign="top">
     <div align="center"><a href="user.asp">Main Menu</a></div>
    </td>
    <td></td>
   </tr>
   <tr>
```

```
<td height="274"></td>
<td colspan="6" valign="top">
 <% If mesej = "" then %>
 <table width="62%" border="1" align="center">
  <tr>
   <td width="23%"><b>Day</b></td>
   <td width="77%"><b>Time/Course Code/Room Code</b></td>
  </tr>
  <% for s=0 to 5 %>
  <tr>
   <td width="23%">
    <% = D(s) %>
   </td>
   <td width="77%">
    <table border = 0 cellpadding = 1>
    <% for k=0 to 10 %>
     <% for j=0 to i-1      'faculty courses list
          If dayValue(j) = D(s) then
             If timeValue(j) = T(k) then
       %>
      <tr>
       <td><font size=2>
        <% = timeValue(j) %>
        </font></td>
       <td> </td>
           <td><font size=2>
        <% = course_code(j) %>
        </font></td>
       <td> </td>
       <td><font size=2>
        <% = room_code(j) %>
        </font></td>
      </tr>
       <% End If %>
      <% End If %>
     <% Next %>

        <% for j=0 to a-1 %>
          <% If day1(j) = D(s) and time1(j) = T(k) then  %>
         <tr>
       <td><font size=2>
        <% = time1(j) %>
        </font></td>
       <td> </td>
           <td><font size=2>
        <% = nfCode(j) %>
        </font></td>
       <td> </td>
```

```
        <td><font size=2>
         <% = nfRoom1(j) %>
         </font></td>
       </tr>
            <% ElseIf day2(j) = D(s) and time2(j) = T(k) then %>
         <tr>
        <td><font size=2>
         <% = time2(j) %>
         </font></td>
        <td> </td>
            <td><font size=2>
         <% = nfCode(j) %>
         </font></td>
        <td> </td>
        <td><font size=2>
         <% = nfRoom1(j) %>
         </font></td>
       </tr>
            <% ElseIf day3(j) = D(s) and time3(j) = T(k) then %>
         <tr>
        <td><font size=2>
         <% = time3(j) %>
         </font></td>
        <td> </td>
            <td><font size=2>
         <% = nfCode(j) %>
         </font></td>
        <td> </td>
        <td><font size=2>
         <% = nfRoom1(j) %>
         </font></td>
       </tr>
            <% End If %>
          <% Next %>
     <% Next %>
      <tr>
       <td></td>
      </tr>
     </table>
    </td>
  </tr>
  <% Next %>
</table>
<% else %>
<table width="54%" border="0" align="center">
 <tr>
  <td><b>
   <% Response.Write mesej %>
```

```
          </b></td>
        </tr>
      </table>
      <p>
        <% End If %>
      </p>
      <br>
      <hr>
      <p align="center"><a href="user.asp">Main Menu</a></p>
    </td>
    <td></td>
  </tr>
  <tr>
    <td height="20"></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
  </tr>
</table>
<!-- #EndEditable -->
</body>
<!-- #EndTemplate --></html>
```