Automated Timetabling System

BY
CHONG ZHENG LUN

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF INFORMATION SYSTEM (HONS) BUSINESS INFORMATION SYSTEM
Faculty of Information and Communication Technology
(Perak Campus)

JAN 2017

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: _____
_____
_____

Academic Session: _____

I _____
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____ _____
(Author's signature) (Supervisor's signature)

Address:
_____
_____
_____
_____
Supervisor's name

Date: _____ **Date**: _____

Automated Timetabling System
BY
CHONG ZHENG LUN

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF INFORMATION SYSTEM (HONS) BUSINESS INFORMATION SYSTEM
Faculty of Information and Communication Technology
(Perak Campus)

JAN 2017

DECLARATION OF ORIGINALITY

I declare that this report entitled "**AUTOMATED TIMETABLING SYSTEM**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : _____

Date : _____

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Dr. Doris Wong and Mr. Kesavan who have provided me very helpful advice and guidance in accomplishing my project. This is the first project that I involve such long periods to develop starting from scratch to a completely workable system. A million thanks to you.

To all my beloved course mates and friends, thanks for your patience, unconditional support and love, and for standing by my side during hard times. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

This project is a web application development project for academic purpose. It will provide the readers with the basic concept of the system, design as well as the methodology. This project aims to develop an improvement of the existing system that automated generating of a timetable for UTAR students based on the subject they choose. Being automated system means reducing the human intervention to a minimum by electronic devices, as defined in Dictionary.com. The system is expected to reduce the effort of the students in planning for their timetable and minimize the error during the planning. This project development adapts the waterfall methodology as the process of developing the whole system. It will follow step by step to develop the system before the next step to start. Meanwhile, there are 50 of survey questionnaire will be distributed to students in UTAR from different faculties for data collection and analyzing purpose. This survey aims to study the problems facing during the planning of timetable and the factors of considerations. In order to be automatic, the system requires to randomly scheduling the timetable for students. Thus, there are some of the algorithms are studied for support the randomization such as tree algorithm, and randomized algorithm. The logics for randomization should be more in the looping process until the best match is returned. Moreover, the system is built on Java platform which J2EE will be the server side scripting, while Javascript, HTML5 and JQuery would be the client side scripting. The database support would be MySQL and the glassfish server will be used to host the web application. The system will be compiled using NetBeans IDE and runs on Linux operating system (OS). All in all, this project is expected to produce a web-based automated system that helps the student in planning for their timetable for next semester.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLE

LIST OF ABBREVIATIONS

UTAR	Universiti Tunku Abdul Rahman
Java EE	Java Platform, Enterprise Edition
JSP	Java Server Page
JSTL	Java Server Page Standard Tag Library
OS	Operating System
CRN	Course Reference Number
PDF	Portable Document Format
SIS	Student Information System
UML	Unified Modeling Language
HTML 5	Hypertext Markup Languages 5
CSS	Cascading Style Sheet
MVC Model	Model-View-Controller Model
CRN	Course Registration Number
ERD	Entity Relationship Diagram
AJAX	Asynchronous JavaScript And XML
DOM	Document Object Model
GUI	Graphical User Interface

Chapter 1: Introduction

1-1 Motivation and Problem Statement

In order to figure out what are the problems currently facing by UTAR students when using the system, a survey will be conducted to target 50 students in UTAR from different faculties. The survey questionnaires (Appendix A) will be distributed to the students and the data will be collected as the evidence to support the problem statements.

Based on the survey done, it believes that the processes of the registration have to be done in manually starting from planning to re-arrange the timetable which could be very time consume for a student to plan for his/her the desired timetable. Appendix B-1 shows the problems facing when student plan for their timetable, 70% of them always facing the problem of clashing of time with other classes. The minor problem such as don't know how to start to plan the timetable and too many options to choose also might affect them to plan their timetable.

Besides that, Appendix B-2 shows that most of the students might spend much of the time than the expected one during planning the timetable. However, only 24% of the UTAR students state that they always spend the time which they expected to plan their timetable. According to Appendix B-3 and B-4, there are some factors always considered by the students for choosing the lecture and tutorial class for that subject such as the time and venue for the class, lecturer or tutor who will be teaching the subject, the class for the next period and so forth.

Furthermore, in term of the human factor, it is easier for the students in UTAR to make mistakes during planning for the timetable due to the long list of course catalog in a page that without much of the whitespace. Sabina (2012) pointed out that "Reading from screen is tiring and takes 25% longer than reading a printed text". Michelle (2015) believes that whitespace increase content legibility and acts as the separator that can add up to a better user experience. Figure 1-2-F1 shows the course catalog available for registering subjects, 66% of the students (Appendix B-5) state that sometimes they might overlook the time schedule showed by the system and end up wrongly plan their timetable due to the design of readability. Furthermore, according to Appendix B-6, more than half of the students rate the design of readability of the class schedule displayed by the system about the scale of 3 to 4 whereas 1 is easiest to view while 5 is most difficult one.

220	T	1	33	Wed	05:00 PM - 06:30 PM	1.5	1-14	H008	
221	T	2	33	Mon	09:30 AM - 11:00 AM	1.5	1-14	H012	
222	T	3	33	Wed	03:30 PM - 05:00 PM	1.5	1-14	H012	
223	T	4	33	Mon	03:30 PM - 05:00 PM	1.5	1-14	H012	
224	T	5	33	Wed	02:00 PM - 03:30 PM	1.5	1-14	H010	
225	T	6	33	Wed	03:30 PM - 05:00 PM	1.5	1-14	H010	
226	T	7	33	Thu	09:30 AM - 11:00 AM	1.5	1-14	H016	
227	T	8	33	Thu	11:00 AM - 12:30 PM	1.5	1-14	H016	
228	T	9	33	Tue	08:00 AM - 09:30 AM	1.5	1-14	H010	
229	T	10	33	Tue	09:30 AM - 11:00 AM	1.5	1-14	H010	
230	T	15	27	Thu	08:00 AM - 09:30 AM	1.5	1-14	H012	
231	T	16	22	Wed	11:00 AM - 12:30 PM	1.5	1-14	H010	
232	T	17	20	Mon	03:30 PM - 05:00 PM	1.5	1-14	H010	
UBMM3013 - STRATEGIC MANAGEMENT [3.00]									
233	L	1	259	Wed	10:00 AM - 12:00 PM	2.0	1-14	IDK2	
234	L	2	260	Fri	02:30 PM - 04:30 PM	2.0	1-14	IDK2	FSC-22
235	T	7	33	Tue	03:30 PM - 05:00 PM	1.5	1-14	H010	
236	T	8	31	Mon	11:00 AM - 12:30 PM	1.5	1-14	H016	
237	T	9	33	Tue	11:00 AM - 12:30 PM	1.5	1-14	H012	
238	T	12	33	Mon	08:00 AM - 09:30 AM	1.5	1-14	H016	
UBTC2013 - CONSUMER BEHAVIOUR [3.00]									
239	L	1	68	Mon	08:00 AM - 10:00 AM	2.0	1-14	H215	
240	T	2	17	Fri	09:00 AM - 10:00 AM	1.0	1-14	H002	
241	T	3	19	Wed	10:00 AM - 11:00 AM	1.0	1-14	H001	

Figure 1-1-F1: List View of the Time Slot Available for Registration

Moreover, the system does not display the lecturer or tutor who will teach this particular subject on next trimester. The students are required to register the subject first before they can view it. After the students complete to register all subjects, they should wait for the system to update the timetable to the UTAR Portal for some minute and then can view who will teach this subject. Since the lecturer and tutor are considered as one of the factors for register the subject and more than half of the students (Appendix B-3) state that it is the factor for them to plan their timetable. Hence, it would be difficult for them to manage the timetable before and after the registration.

As conclusion, the typical problems are currently facing by UTAR students for registering for the subjects are as followed:

- The students in UTAR require spending much of the time to plan and re-arrange for their timetable for next trimester. (Appendix B-2)
- The students in UTAR might easily overlook for the correct time and plan the timetable wrongly due to the readability of the design. (Appendix B-5 and B-6)
- The students in UTAR are not able to view who will be lecturing the subjects before and during register the subjects which are the factor to consider for planning the timetable. (Appendix B-4)

1-2 Background Information

Subject registration/ timetabling/ unit registration system is a system that allows all students in university to register the subjects which they will take for the next semester. The students can choose the subjects available for register and the system will generate the timetable and display it to the students. Generally, the students can view, add, search and remove the subjects by using the system. Each university will have such system to allow the students to register the subject, however, the registration process of the system in each university is different from each other based on the rules and constraints set by the university. For example, the rules and constraints could be the credit hour constraint, courses, subject pre-requisite, time duration and so forth.

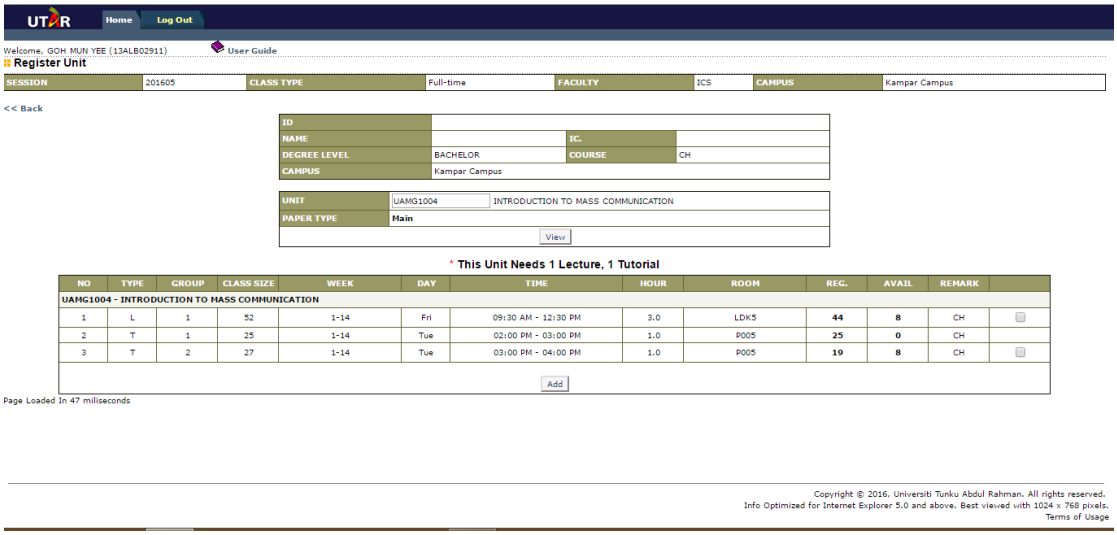


Figure 1-2-F1: Course Registration System in UTAR

In Universiti Tunku Abdul Rahman (UTAR), like other universities, the student can register the subjects for next semester via Course Registration System during the end of the semester, normally in week 12 to week 14 for a long semester and week 5 to week 6 for the short semester. Each student from different faculty should wait for the announcement to register their subject since the period for registering is different for each other. Before the registration, the students have to plan for their own timetable for next semester follow the course structure provided by the university. During the registration, the students required to login to the system and enter the subject code in order to register the subject. Besides, the students are given some time to modify and rearrange the timetable after the registration if they do not satisfy with the current one.

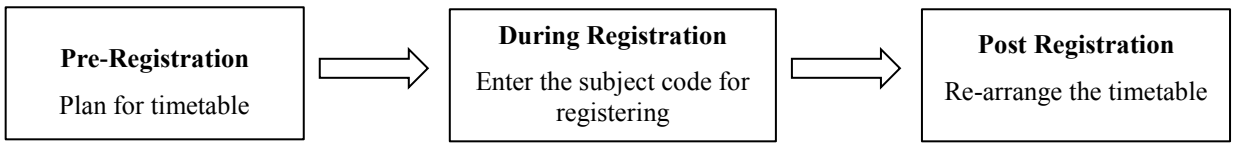


Figure 1-2-F2: Process of Registering Subjects in UTAR

Based on the handbook written as guideline for students (2010), there are some rules which are clearly stated and should be followed by the students that what should be done for biding the subjects, what should be followed in order to register the subjects for next semester and how should the students can register the subjects by using the system. The following are the highlighted rules should be followed by full-time undergraduate students:

- Full-time students who are not on probation period are allowed to register for maximum 21 credit hours for long trimester (14 weeks) and 12 credit hours for short trimester (7 weeks) and a minimum of one unit for both.
- Full-time students who are on probation period are allowed to register for maximum 15 credit hours for long trimester (14 weeks) and 9 credit hours for short trimester (7 weeks) and a minimum of one unit for both.
- A student who did not register for any unit in teaching trimester must apply for a leave of absence for that trimester.

The course registration system in UTAR is a web-based application that developed by using the Java Technology, which is Java Platform, Enterprise Edition (Java EE). It is a technology to allow the developer to build a web application through plenty of web components such as servlet, java server page (JSP), java server page standard tag library (JSTL), java libraries and so forth. The students can access the system through the browser on their own desktop computer or laptop.

1-3 Project Scope

Since the majority of the students in UTAR required an automated timetabling system (Appendix B-7), this project would be expected to produce an improvement of existing UTAR unit registration system to the system that provides several features. This project will involve some algorithms to automatic arrange the time schedule for students based on the subjects selected. It also expected to produce an improvement of the user interface to enhance user experience.

Thus, the deliverables are as following:

- Automatic generation and register of timetable during pre-registration based on the subject selected.
- Display the details about who will be teaching that subject on next semester when students click on the particular timeslot.
- Collapsing timeslot according to the subject and expand when the mouse clicks on it.
- Timetabling tool which is interactive and handy to allow student to plan timetable easily
- Highlighting the timeslot when the mouse hovers it to avoid overlook of the time.
- Email notification for the students about the timetable has been generated and sent.

1-4 Project Objectives

This project aims to deliver students an automated timetabling system. One of the features is an automated timetabling function during pre-registration. It would be able to automatically to generate the timetable for the student based on the subject selected by them which can reduce clashing issue or any other error. Besides, the system should be able to display the lecturer or tutor who will teach the subject on next semester which might ease of the students to manage and plan for their timetable. Lastly, it would also provide the improvement of the user interface that sectioning the time slot displayed according to the subject and highlight the slot when the mouse hovers it which can minimize the error in the human factor.

Generally, the system would be able to minimize the effort of the students in planning their timetable. It is trying to help the students to save a lot of the time for

other stuff instead of spending much of time just for planning the timetable. The project would be much more emphasize on the pre-registration process. Only minor system improvements on the registration and post-registration processes would be done if necessary, otherwise, it would be remained as same as the existing system. Besides, the student registration process would not be expected to be emphasized on this project as the students' profile and necessary data set such as faculty, course, subject, academic staff and so on are predefined in the database.

In short, the objectives of the project are as following:

- To develop a system to reduce the time students spend on planning and register timetable.
- To develop system to eliminate error on planning and registering timetable.
- To develop an interface to enhance the user experience and readability of the class schedule displayed to the students.
- To develop and study of the algorithm to generate the timetable automatically.

1-5 Proposed Approach/ Study

The proposed approaches and studies for this report are survey approach and literature review study. The survey questionnaire will be distributed to 50 UTAR students who using the existing. Besides, the literature review will be done to study and research how other universities implement the course registration system. Several systems from the research will be the benchmark of this proposed system.

1-6 Impact, Significance and Achievements

The primary beneficiary of this project would be the students in UTAR who suffering in spending too much time to plan the timetable. It aims to minimize the time of students in planning and registering their timetable. Besides, since minority of the universities implement the system that auto-generation of the timetable for students, this project is able to provide a basic idea for the one whom yet to implement it.

Moreover, the readers might be able to understand and study the fundamental of algorithms that implement in this system that support the auto-generation of timetable. This project worth for the readers to read as it can motivate or inspire them to figure out the other problems that might be also faced by UTAR students, not only

during the registration process.

Besides, it can also provide the opportunity for the readers who interested on this project to continue to further refine the algorithms or improve the system which can produce more advanced system to benefit the students in UTAR. By having this project, it could help them to reduce their efforts on developing the project start from scratch and emphasize more on other parts.

1-7 Report Organization

This report will be organized in 6 Chapters and the survey data and test result will be included in appendix. In Chapter 1, it will briefly describe the background of the system, problem statements, objectives, proposed study as well as significance and achievements. It provides readers the basic idea of the proposed system.

In Chapter 2, it will discuss about the literature reviews that have been done, 5 course registration from other universities will be the benchmark of this proposed system. In Chapter 3, it will discuss the system design in details which includes block diagram, database design, layout design, module and algorithms design.

In Chapter 4, the technology and methodology involved in this proposed system as well as the implementation plan will be discussed briefly. In Chapter 5, it discusses about the test plans. Several test plans will be conducted and the test result will be collated for analysis purpose. Last chapter will conclude the report and provides a brief summary to the readers about the whole proposed system.

Chapter 2: Literature Review and Data Collection

2-1 Literature Review

2-1-1 Concordia University's Course Registration System (n.d.)

The course registration system in Concordia University is a system that allows the student to browse, search, add, drop, swap, and verify the subject which offered by the university. The guidelines for course registration are available for at the official website of the university to guide the students step-by-step on registering process for the subjects. The students have to see the advisor as the prior requirement to register the subject. Once done for the advisory, the students are allowed to register the course through login to the system.

The students then have to plan for their own timetable before register the subjects. The students can browse the course catalog to access to a comprehensive listing of information about course offered. The subjects could be viewed by selecting the first letter of the subject of the course. For example, by selecting the letter "M", it will return the subject of Math and Computational Finance as well as Management. When students further to select the particular course subject, it will expand the details of the subject and click on "view class section" can view how many sections are available and when they are offered.

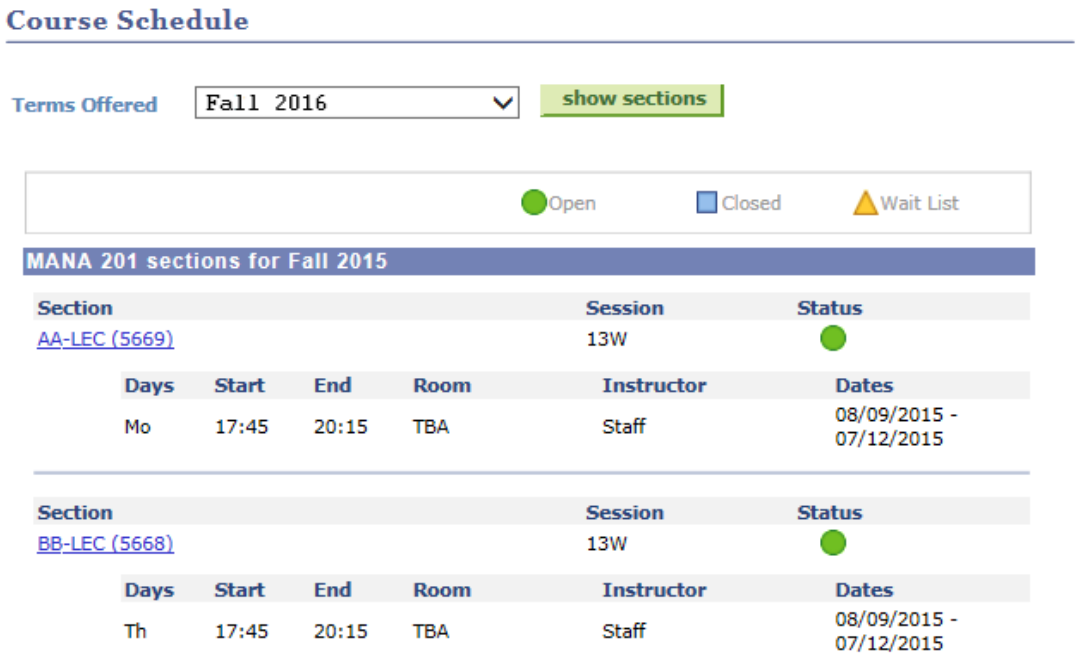


Figure 2-1-1-F1: Sections Available for Register

Besides, in order to assist the students in planning the timetable, the system provides a comprehensive search tool for the students to filter out the result. The

student not only can search by course number, term, subject number, course level and faculty, but also can search for class times, days of the week, location, instructor, and course keyword. After entering the search criteria, the system will return the classes that match the search criteria to the student.

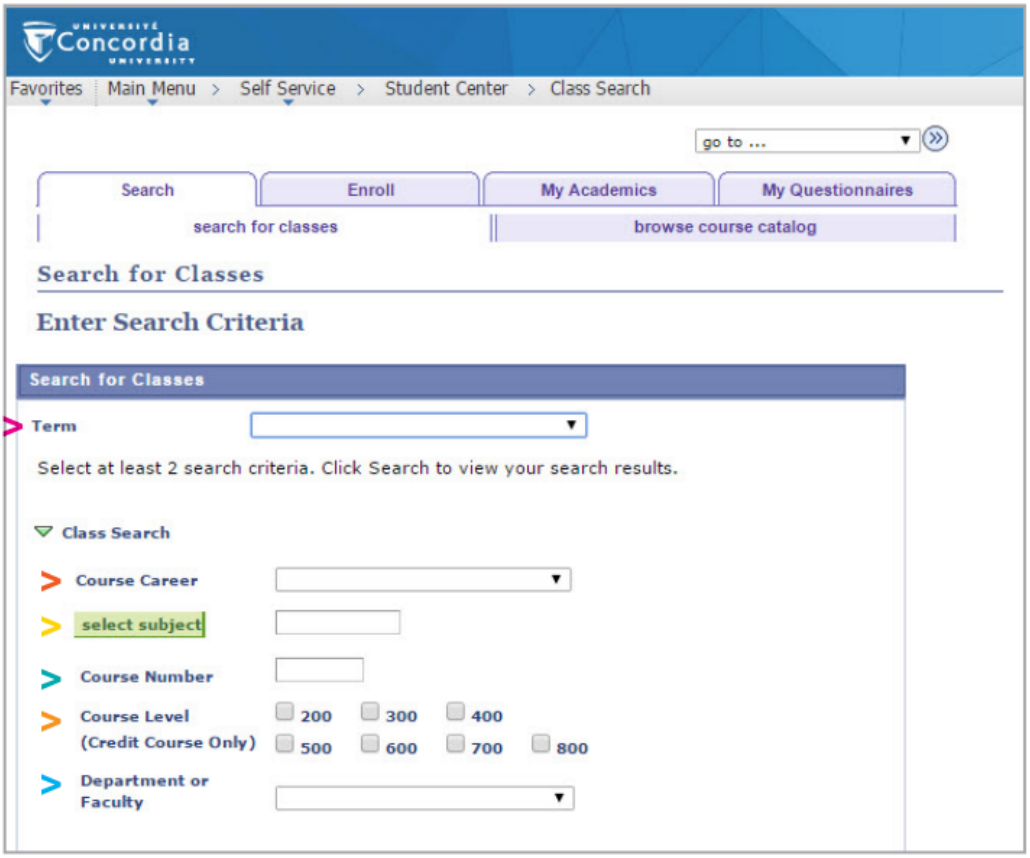


Figure 2-1-1-F2: Search Tool

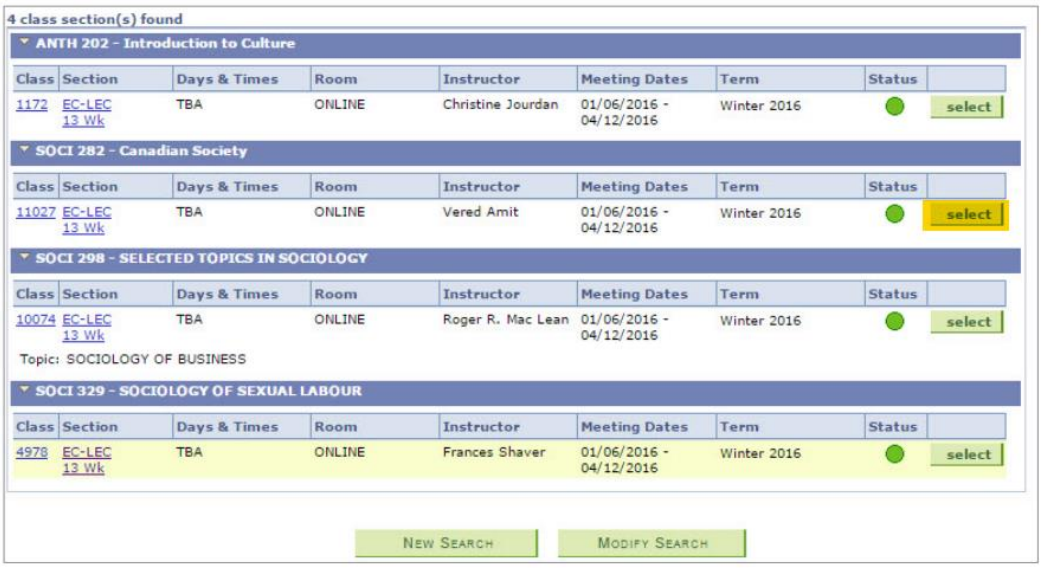


Figure 2-1-1-F3: Search Result

After the students plan their timetable, then can start to register for the subject they planned. The steps of class searching are required to repeat in order to select the subject to register. If the subject is satisfying for enrolling, click “select class” to add the subject to course cart. Moreover, once all seats in a course have been filled it may still be possible for the students to add their name to a wait list. After finish selecting all subject, clicking “Finish Enrolling” to confirm the registration. The students can view their weekly calendar to verify their courses. The resulting “My Class Schedule” page allows you to select the term for which you wish to verify your registrations.

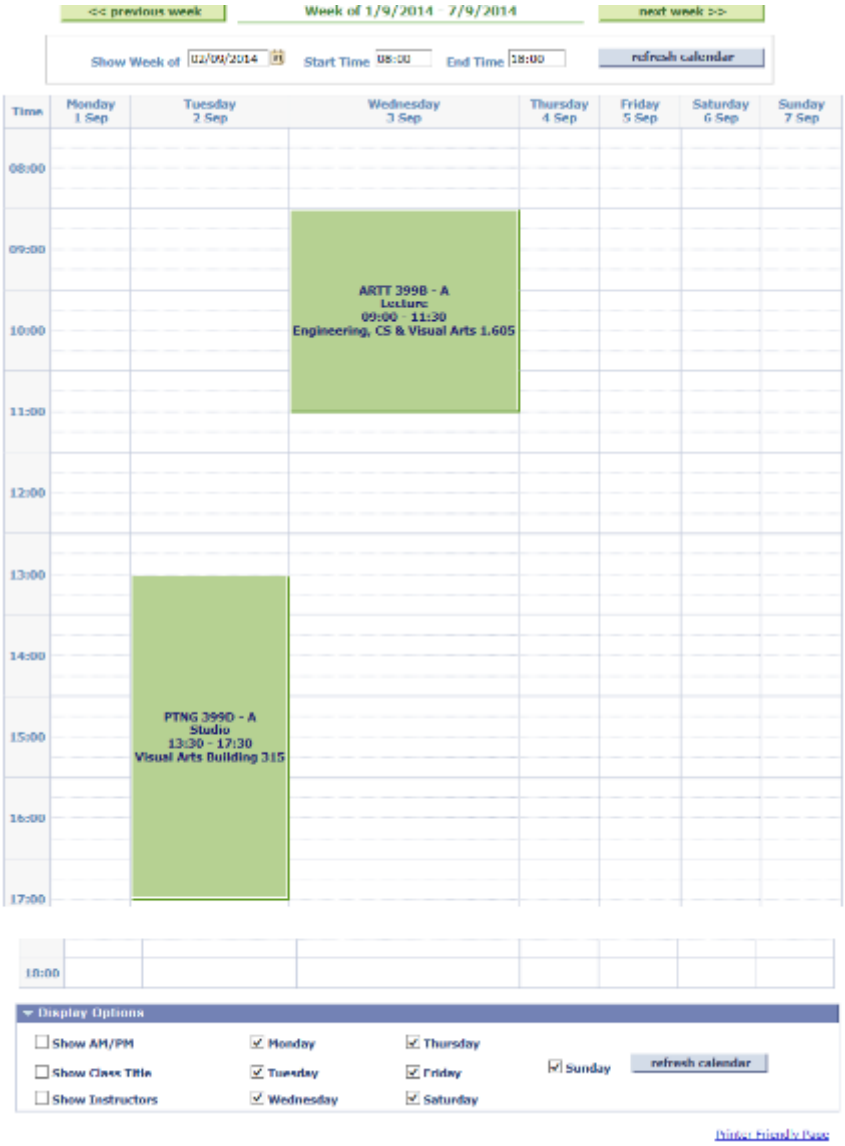


Figure 2-1-1-F4: The Weekly Calendar View

This system provides the better user experiences as the design allow the user to view the course subject by selecting the first letter rather than display all subjects course at once which might distract the user’s eyes. Besides, it offers a powerful

searching tool that allows the students search by more than one criterion to filter the result and return the best matches. The student can view the course details as well such as who are the instructors for the subject and course description. However, students also required to spend the time to plan for their timetable and go through many steps to view the section available for the register. The system might require providing the students for automatically generate the timetable randomly to reduce the step require to register the course subject.

2-1-2 Course Registration System of University of Melbourne (n.d.)

The course registration system of University of Melbourne allows the students to register the class they enroll in a particular subject. It is different between enrollment and class registration. Enrollment allows the students to enroll in subjects that they intend on studying. Once they have enrolled in the subjects, and then need to register for classes. In the other words, registration for the class is only allowed once the student has enrolled in the subjects. The purpose of class registration is to let staff know who to expect in their class. The students require planning for their own timetable before registering for classes. The period for registering the classes will be announced through the university website. The steps involve are as followed:

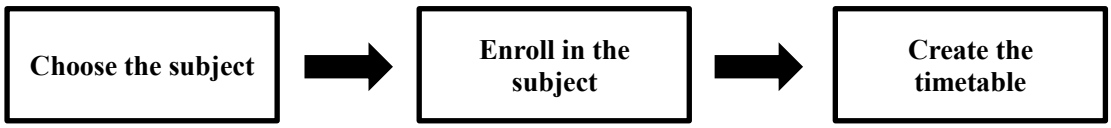


Figure 2-2-F1: Procedure to Create a Timetable

Before the class registration, the students can plan their preferred timetable using the online "University Timetable (SWS)" which will show all the proposed classes includes days, times, venue, weeks and so forth. After that, the students can register class by logging to their own account at the portal. When finish to select all the desired class, the options should be saved and the student will automatically be placed into single sitting classes which they required to attend at those time they registered.

During the students viewing the timetable at "University Timetable (SWS)", they are allowed to view the classes available for register according to the departments, subject itself and the rooms. Besides that, it can select by period(s) such as semester 1, next week; time period such as the morning (07:00 – 18.30),

full day period; and report type as well. Furthermore, the system provides the basic search function to allow the student filter the result by searching the departments, subjects, or rooms. The student either can search for the result or select the option from the drop-down list.

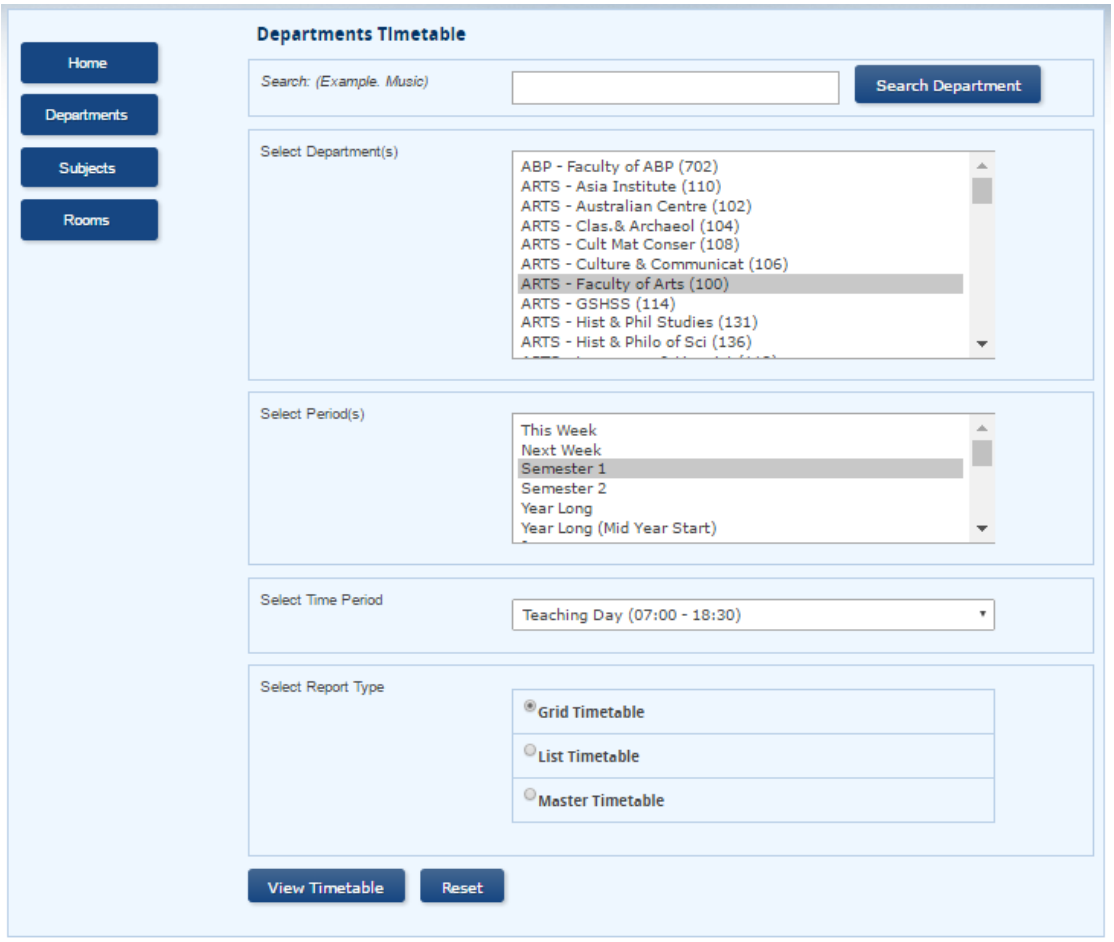
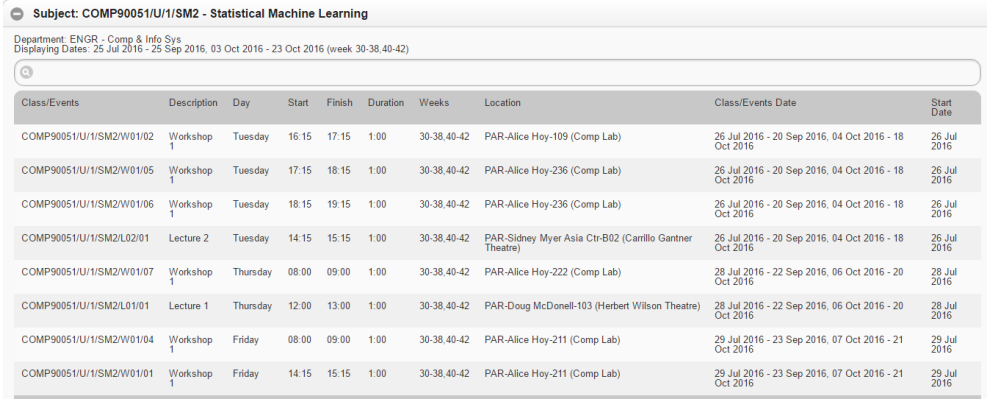


Figure 2-1-2-F2: View Timetable According to Departments

After selecting the criteria for the subject to view, the system can return it either in grid view, list view or master timetable.


Subject: COMP90051/U/1/SM2 - Statistical Machine Learning

Department: ENGR - Comp & Info Sys
Displaying Dates: 25 Jul 2016 - 25 Sep 2016, 03 Oct 2016 - 23 Oct 2016 (week 30-38,40-42)

Class/Events	Description	Day	Start	Finish	Duration	Weeks	Location	Class/Events Date	Start Date
COMP90051/U/1/SM2/W01/02	Workshop 1	Tuesday	16:15	17:15	1:00	30-38,40-42	PAR-Alice Hoy-109 (Comp Lab)	26 Jul 2016 - 20 Sep 2016, 04 Oct 2016 - 18 Oct 2016	26 Jul 2016
COMP90051/U/1/SM2/W01/05	Workshop 1	Tuesday	17:15	18:15	1:00	30-38,40-42	PAR-Alice Hoy-236 (Comp Lab)	26 Jul 2016 - 20 Sep 2016, 04 Oct 2016 - 18 Oct 2016	26 Jul 2016
COMP90051/U/1/SM2/W01/06	Workshop 1	Tuesday	18:15	19:15	1:00	30-38,40-42	PAR-Alice Hoy-236 (Comp Lab)	26 Jul 2016 - 20 Sep 2016, 04 Oct 2016 - 18 Oct 2016	26 Jul 2016
COMP90051/U/1/SM2/L02/01	Lecture 2	Tuesday	14:15	15:15	1:00	30-38,40-42	PAR-Sidney Myer Asia Ctr-B02 (Carrillo Gantner Theatre)	26 Jul 2016 - 20 Sep 2016, 04 Oct 2016 - 18 Oct 2016	26 Jul 2016
COMP90051/U/1/SM2/W01/07	Workshop 1	Thursday	08:00	09:00	1:00	30-38,40-42	PAR-Alice Hoy-222 (Comp Lab)	28 Jul 2016 - 22 Sep 2016, 06 Oct 2016 - 20 Oct 2016	28 Jul 2016
COMP90051/U/1/SM2/L01/01	Lecture 1	Thursday	12:00	13:00	1:00	30-38,40-42	PAR-Doug McDonell-103 (Herbert Wilson Theatre)	28 Jul 2016 - 22 Sep 2016, 06 Oct 2016 - 20 Oct 2016	28 Jul 2016
COMP90051/U/1/SM2/W01/04	Workshop 1	Friday	08:00	09:00	1:00	30-38,40-42	PAR-Alice Hoy-211 (Comp Lab)	29 Jul 2016 - 23 Sep 2016, 07 Oct 2016 - 21 Oct 2016	29 Jul 2016
COMP90051/U/1/SM2/W01/01	Workshop 1	Friday	14:15	15:15	1:00	30-38,40-42	PAR-Alice Hoy-211 (Comp Lab)	29 Jul 2016 - 23 Sep 2016, 07 Oct 2016 - 21 Oct 2016	29 Jul 2016

Figure 2-1-2-F3: List View of the Classes Available for Register

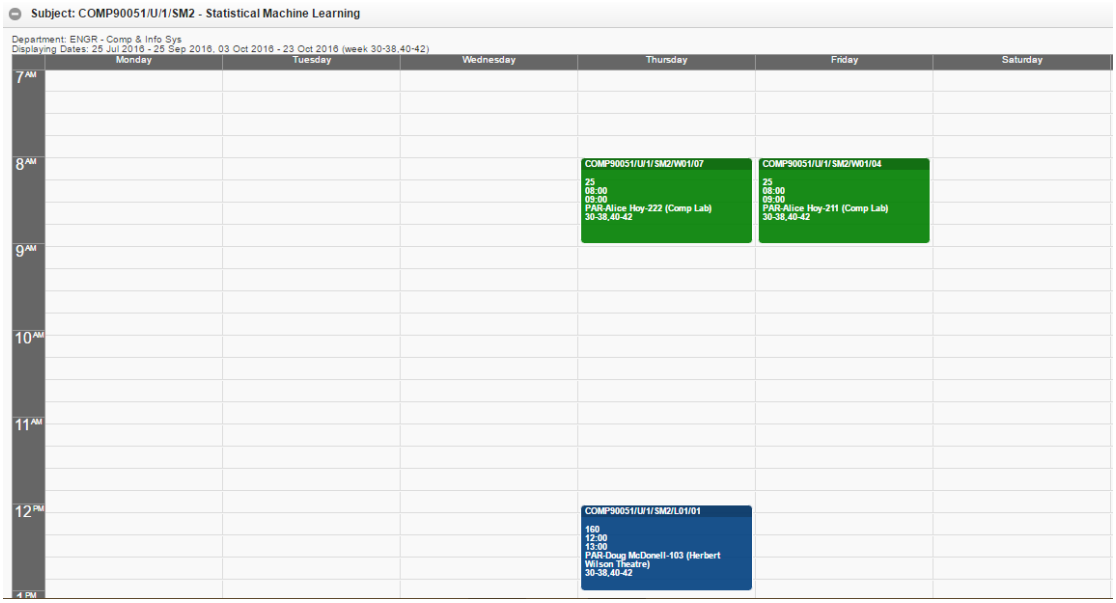


Figure 2-1-2-F4: Grid View of the Classes Available for Register

When the students click on the timeslot, it will prompt out a box to display the details of the class.

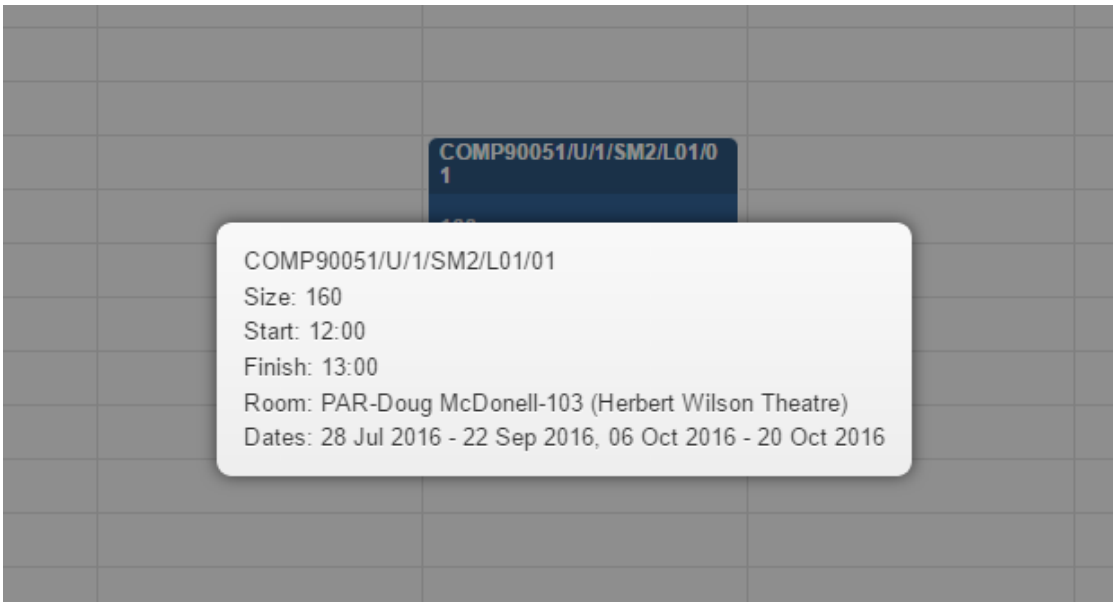


Figure 2-1-2-F5: Box to Display the Details of the Classes

The system allows the students to register any classes they desired (cannot repeat), however, the system will still allow the students to register the classes with timetable clash which is not recommended and consider as unacceptable for Special Consideration, fee refund requests, or poor academic performance. In order to fix the timetable issue encountered by the students during the planning and registering the timetable, the system provides a list of the recommendation against the common issue to solve the difficulties registering in one or all of the classes.

Furthermore, the system also provides "waiting list" feature when the classes are currently full. If a student who is registered in the class leaves, the first student on the waiting list may automatically move into that class. All in all, it is recommended that the system should provide a random picking of the classes and generate timetable which might reduce the time to plan the timetable as well as avoid clashing with other subjects.

2-1-3 Aurora System in University of Manitoba (n.d.)

Aurora System is a system that allows the students in University of Manitoba to manage the matter about the course registration. Students will be using the system for the services and functions of search course catalog, class schedule, apply for admission, register for classes, add/ drop classes, and so forth. Before the students plan for their timetable, it is better for them to understand the course details and the class schedules to reduce the chances of registration errors occurring. The students can access to the class description in Aurora system by searching the course catalog. The comprehensive course details are described clearly for the students.

After that, the student might start to plan for their own timetable by following the class schedules given. The students can search for the class schedule which shows a list of the courses and options are offered. The list should include the Course Reference Number (CRN), the course subject code and number, and the days of the week and the times on which the class meets. The system provides the search tools that allows students to filter the result that only display the best matches to reduce the information displayed. They can search by terms such as winter, falls, spring, and summer and so on, and then followed by subject, campus, instructor, start and end time, days and so forth.

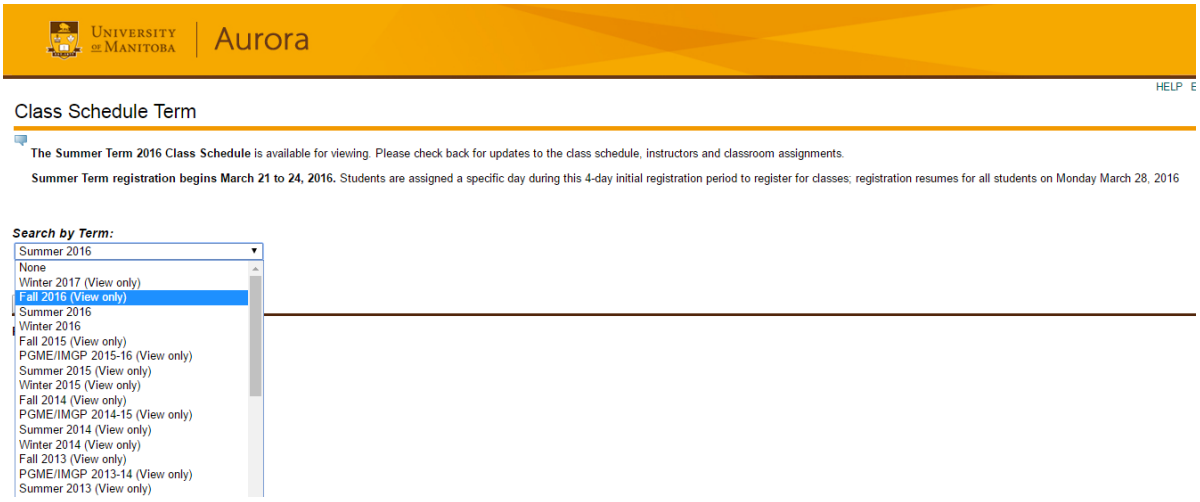


Figure 2-1-3-F1: Search Class Schedule by Term

After the student selects the term, the system will direct the student to the search tool to search the class based on the criteria entered.

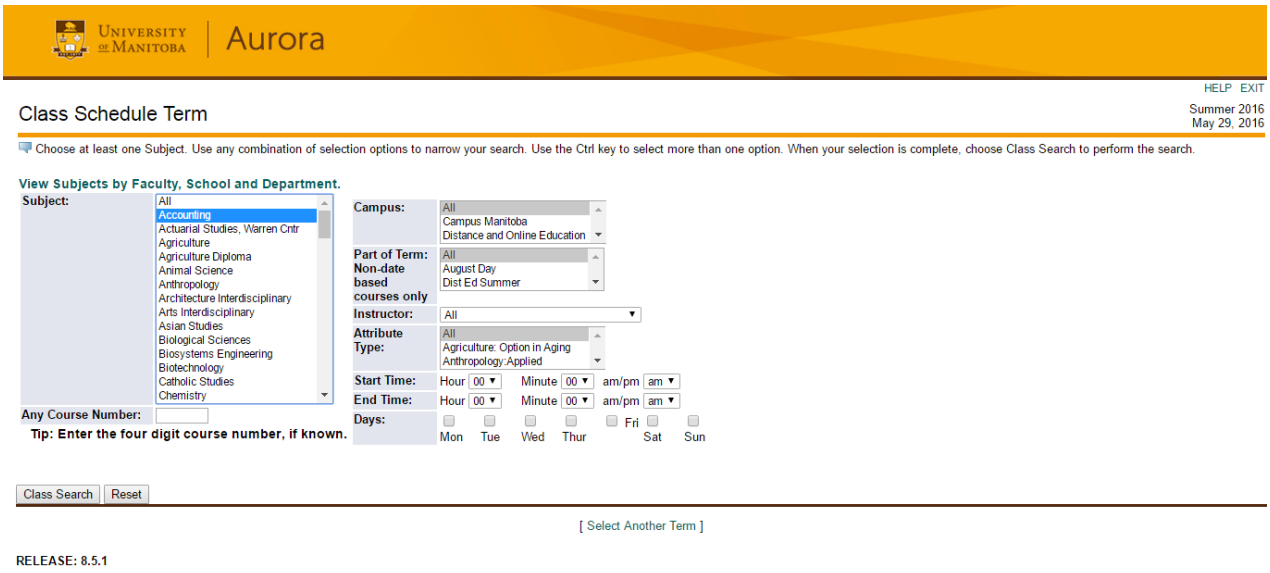


Figure 2-1-3-F2: Search by Faculty, School, and Department

Class Schedule Listing

CLICK THE TITLE OF THE COURSE TO VIEW DETAILED CLASS INFORMATION, INCLUDING **REGISTRATION RESTRICTIONS** AND SEAT AVAILABILITY.

Sections Found

Introductory Financial Accounting - 30102 - ACC 1100 - A01

This section will be assessed a lab fee.
This section will have a make-up class on Friday, May 27 (for May 23).

Associated Term: Summer 2016
Part of Term: May-June Evening
Registration Dates: Mar 21, 2016 to May 06, 2016
Levels: Graduate, Undergraduate
Attributes: Dry/Tutorial Lab

Main (Fort Garry & Bannatyne) Campus
Management, I.H. Asper School Schedule Type
3.000 Credits
View Catalog Entry

Scheduled Meeting Times

Type	Time	Days	Where	Date Range	Schedule Type	Instructors
Lecture	7:00 pm - 9:30 pm	MW	HELEN GLASS CENTRE 260	May 02, 2016 - Jun 15, 2016	Management, I.H. Asper School	Ryan A. Palmer (P)
Tutorial	4:00 pm - 6:30 pm	W	HELEN GLASS CENTRE 260	May 02, 2016 - Jun 15, 2016	Management, I.H. Asper School	Ryan A. Palmer (P)
Lecture	7:00 pm - 9:30 pm	F	HELEN GLASS CENTRE 260	May 27, 2016 - May 27, 2016	Management, I.H. Asper School	Ryan A. Palmer (P)
Final Exam	6:00 pm - 9:00 pm	F	DRAKE CENTRE 343	Jun 17, 2016 - Jun 17, 2016	Management, I.H. Asper School	Ryan A. Palmer (P)

Figure 2-1-3-F3: Class Schedule Listing

The students can plan for their timetable using the Timetable Planning Form provided by the system in Portable Document Format (pdf). The students have to plan the timetable carefully with free of time conflict as possible to avoid any error during the registration period. The student can further click on the title of the course to view the detailed class information, including registration restrictions and seat availability.

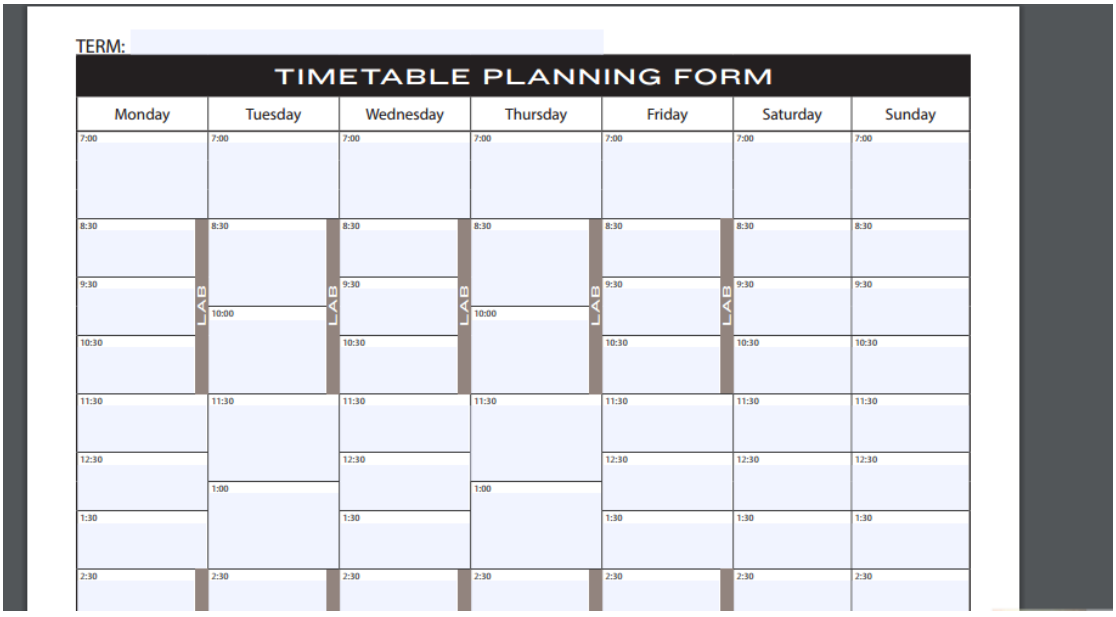


Figure 2-1-3-F4: Timetable Planning Form

During the registration time, student has to login to Aurora System and search for the class to add to the registration list. If the student knows the course,

section and CRN for the class, then choose for the "Add or Drop Classes", else choose the "Look Up Classes" option to search by course subject, course title, class type, part of term, attribute, day and time offered, or combination of these. After done the selection, then click "Register" to add the course immediately. The system will prompt out the status to indicate if any error occurred. For example, "Time Conflict with CRN" means that course times overlap; "Closed Section" means the course is full.

The system provides the timetable planning form the help the students in planning their own timetable. Furthermore, it also provides a powerful searching tool to filter the result to make it easier to view the option offered. There is not only provides guidance in text form but also has the video that provides the details guidelines for registration. However, the students have to spend much of time to plan for their timetable to meet their requirements as well as to avoid any clashing with other subjects. The system should require the feature that helps the students to generate their timetable which can save a lot of the time and also eliminate the possibility of time conflict.

2-1-4 Course Registration System in The Ohio State University (n.d.)

The Ohio State University offers Student Center which is a self-service page that provides students an entry point to the Student Information System (SIS). The students can navigate along to the page for enrollment, search for classes, financial aid, make payment and so forth by login into the account. In order to allow the students plan for their class schedule, the class search page enables the students to search for classes at any or all campuses of The Ohio State University which return the details of the classes such as days and times, room, instructor, and meeting dates. The students first have to select the institution and term before go for more advanced. After that, they can search for class schedules by course subject, course number, course career, and campus. There is an additional search criterion that allows the students to specify the search criteria to refine the search result.

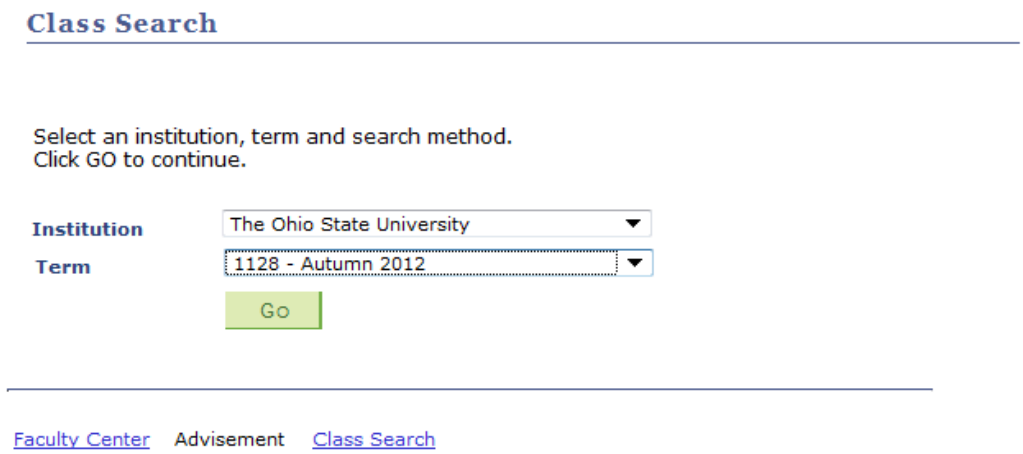


Figure 2-1-4-F1: Class Search Page

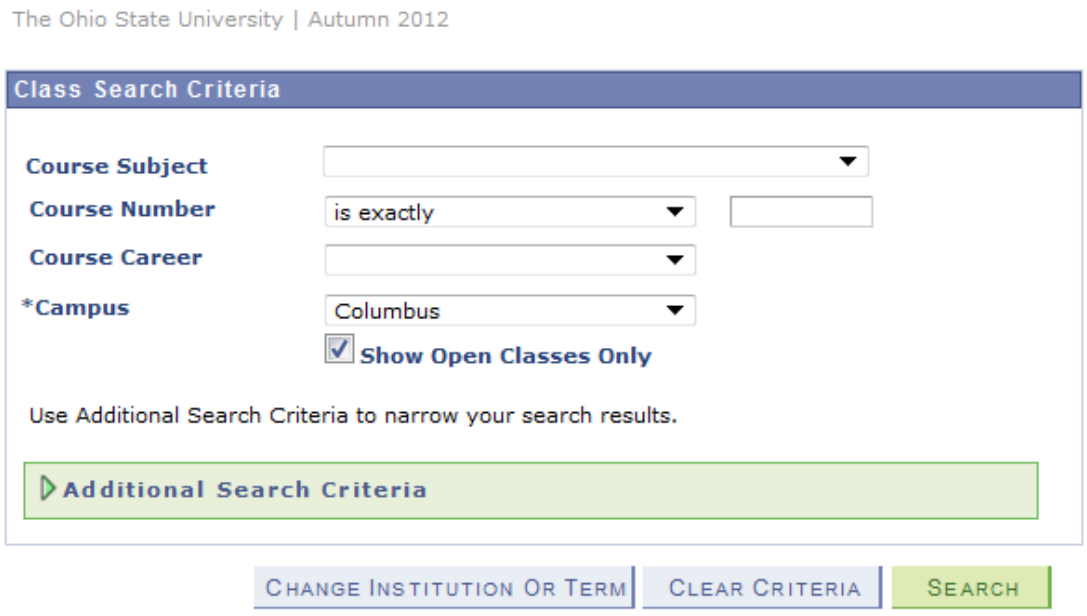


Figure 2-1-4-F2: Class Search Criteria

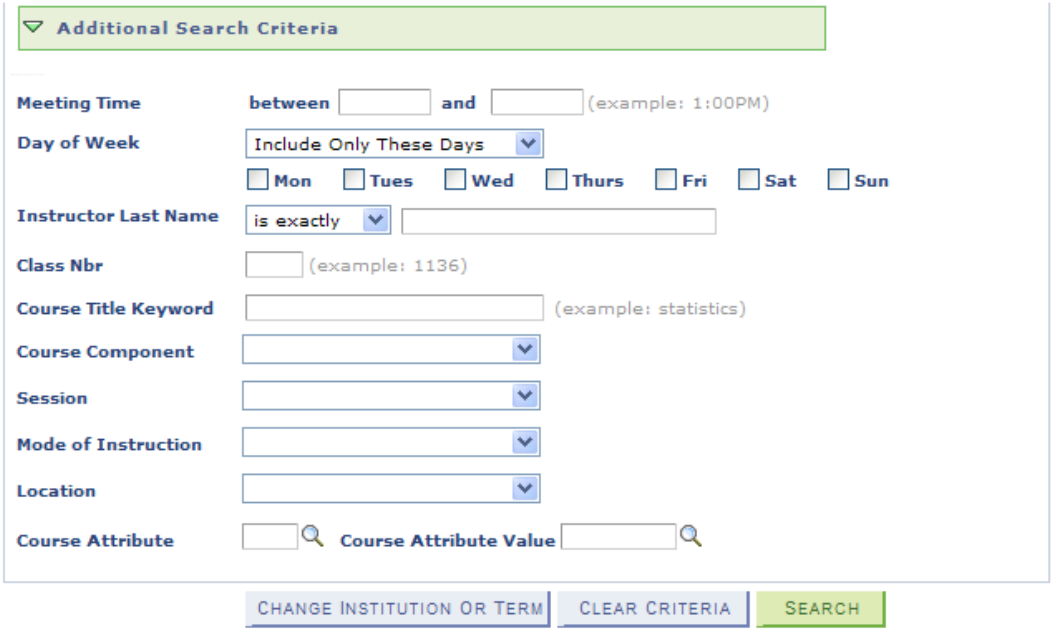


Figure 2-1-4-F3: Additional Search Criteria

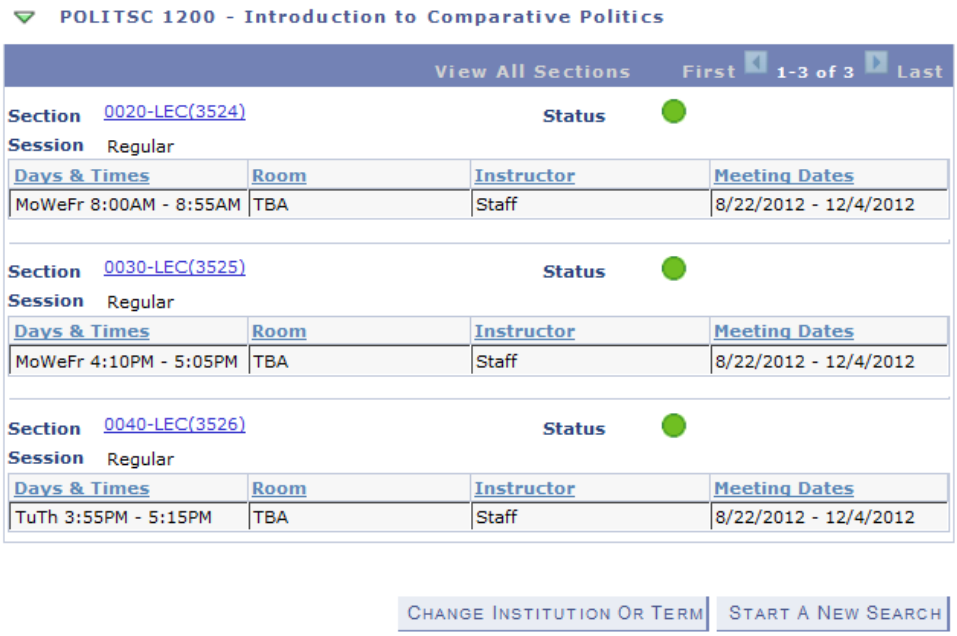


Figure 2-1-4-F4: Search Result

Besides the students plan for the schedule manually, the university also provides a system, "Schedule Planner" which is a web-based class scheduling system that allows the students to select the course from the schedule of classes and set a break time during the day for studying, extra-curricular activities, work schedules and

so on. It will automatically to generate the schedule based on the course selected which meet their criteria specified.

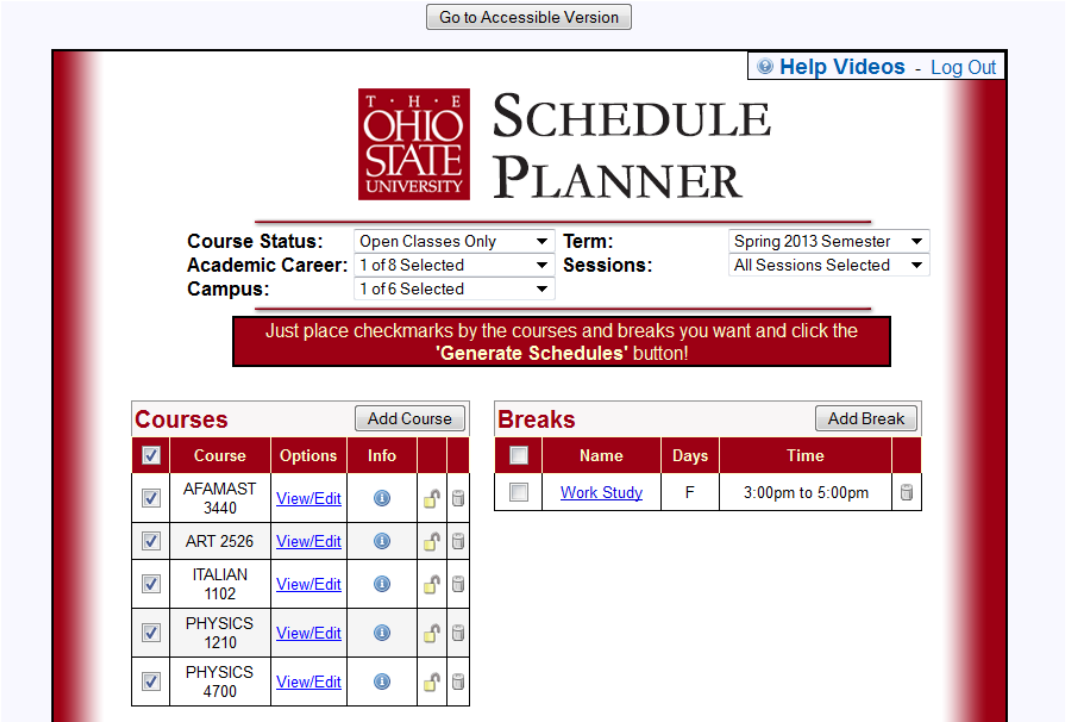


Figure 2-1-4-F5: Schedule Planner of The Ohio State University

The students should click "Add Course" to add the courses to take on next semester. Besides that, they can click "Add Break" to add the times during the day they not willing to attend the class. After done it, the student can click "Generate Schedules" to create the list which meets their criteria and click for the "View" to view the schedule. There might have more than one schedules are generated and the students are only allowed to choose one from them and enroll it. When the students find it is the best choice for enrolling, they require to "Send Schedule to Shopping Cart" to begin registration.

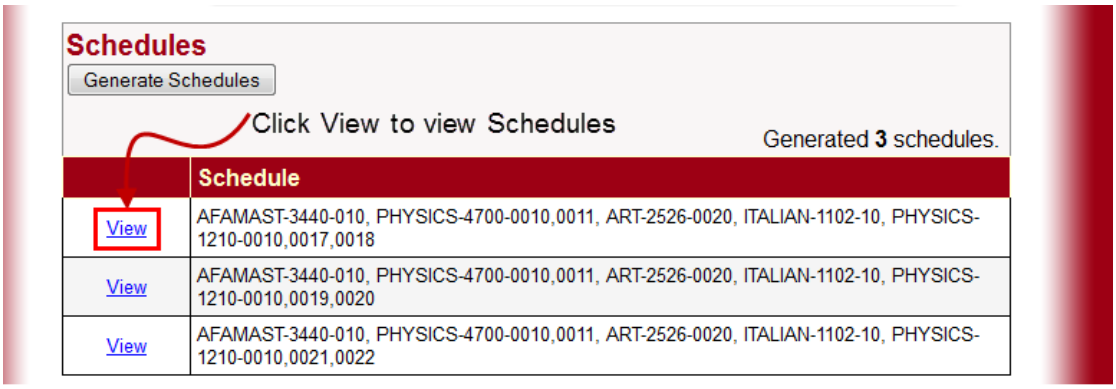


Figure 2-1-4-F6: Generate and View List of Schedule

View Schedule - Spring 2013 Semester

Print Page | Send Schedule to Shopping Cart | Close Schedule

*You are viewing a potential schedule and must send to your Shopping Cart to enroll.

More Info	Status	Subject	Course	Section	Class #	Open Seats	Day(s) & Time(s)	Date	Location (s)	Campus
	Not Enrolled	AFAMAST	3440	010	10585	25	WF - 11:10am - 12:30pm	1/9/2013 - 4/19/2013		Columbus
	Not Enrolled	PHYSICS	4700	0010	11775	20	M - 12:40pm - 1:35pm	1/7/2013 - 4/22/2013		Columbus
	Not Enrolled	PHYSICS	4700	0011	11776	20	TTh - 1:50pm - 3:55pm	1/8/2013 - 4/18/2013	Smith Lab 1081	Columbus
	Not Enrolled	ART	2526	0020	18153	15	MWF - 4:25pm - 6:10pm	1/7/2013 - 4/22/2013	Hopkins Hall 0266	Columbus
	Not Enrolled	ITALIAN	1102	10	3930	25	TWThF - 9:10am - 10:05am	1/8/2013 - 4/19/2013		Columbus
	Not Enrolled	PHYSICS	1210	0010	13544	150	TTh - 8:00am - 8:55am	1/8/2013 - 2/21/2013		Columbus
	Not Enrolled	PHYSICS	1210	0019	19560	25	F - 12:40pm - 2:45pm	1/11/2013 - 2/22/2013	Smith Lab 2024	Columbus
	Not Enrolled	PHYSICS	1210	0020	19562	25	M - 9:10am - 10:05am	1/7/2013 - 2/18/2013		Columbus

Figure 2-1-4-F7: View of the Schedule

Figure 2-1-4-F8: Timetable Generated by the Schedule Planner

During the registration, the students require following the steps of searching class to select the class to add to the cart. After finishing the selection, click "Finish Enrolling" to enroll the course. When an error occurred during the enrollment, the message will display to indicate what error is occurring, there is a "Fix Error" button to display more information and resolve the issue. After the issue is resolved, the student can make payment by clicking "Make a Payment" to pay the tuition fee.

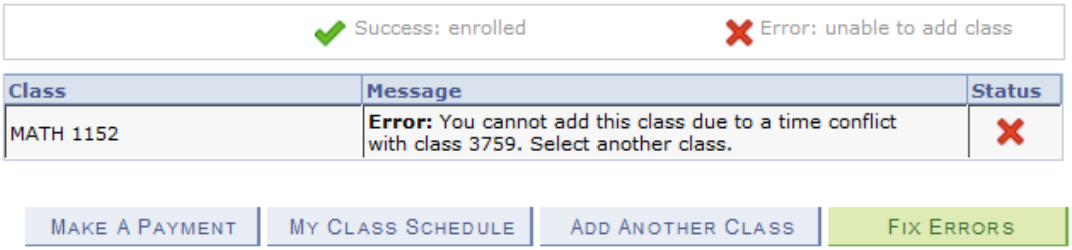


Figure 2-1-4-F9: Error Message Displayed

The system has a good navigation system for the student to navigate around the system easily. It also has the powerful search tool for the students to filter the result to meet the best matches for planning their timetable. Moreover, the university provides the Schedule Planner which can automatically generate a timetable to meet the students' requirement for their timetable. The error messages prompt out when the problem occurred which indicates the error that can help the students to identify the error easily for troubleshooting the issue. However, the system does not have the email notification to send the new timetable in image format to the students that notify the student about the timetable has successful to register. The system should provide this feature to allow the students to save as a copy in a mobile phone, desktop or other devices.

2-1-5 BannerWeb Online Registration System of Smith College (n.d.)

The course registration system, BannerWeb in Smith College allows the students to view, search, register, drop the subjects offered by the college. The students can view the details of the subjects such as department, subject code, course code, title, section, Course Registration Number (CRN), instructor, day, time, and location. The students either can search for the particular subject based on the search criteria or view comprehensive courses offered by the college in PDF file which available for download on the official website. Before the registration, the students require to review the schedule of the class and make an appointment with their adviser

to discuss the course program. Generally, they have to plan their schedule manually before register the subjects.

The system provides a search tool for the students to easily view for a particular subject based on the criteria they enter. Figure 2-5-F1 shows the search criteria available for search. The students have to select the term offered, department/ program, while the other options are optional to select or enter a value.

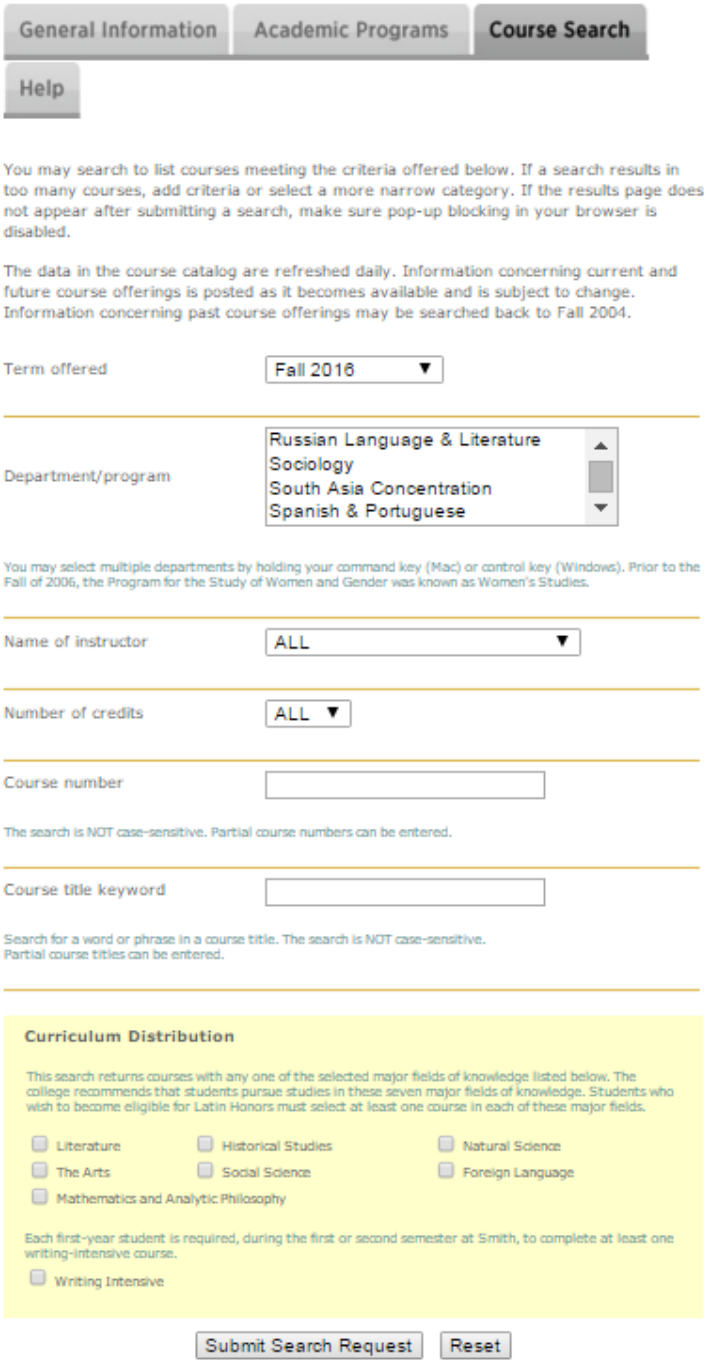
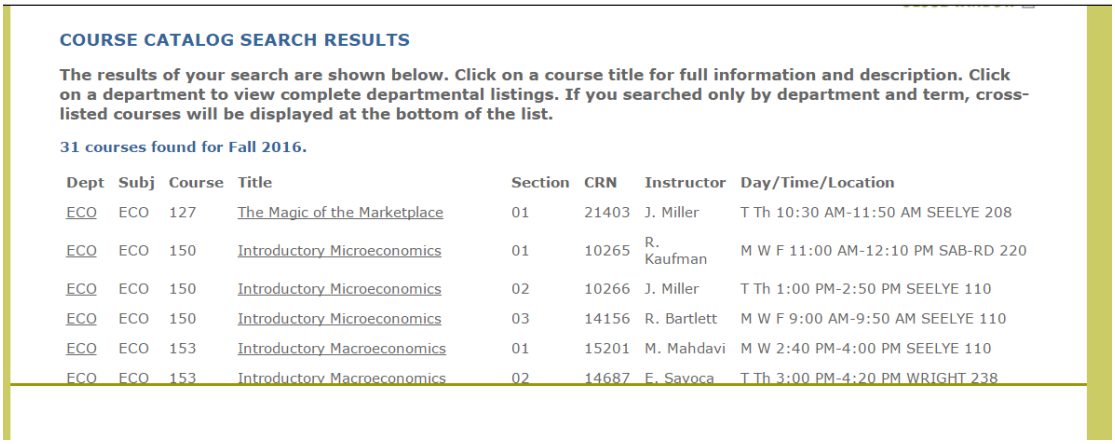


Figure 2-1-5-F1: Search Tool

After selecting the search criteria, the system will return the result back to the student in another browser. The students can further click on the underlined text to view more details about the subject; it will direct the students to a particular page and display the required information. Figure 2-5-F2 shows the search result.



COURSE CATALOG SEARCH RESULTS

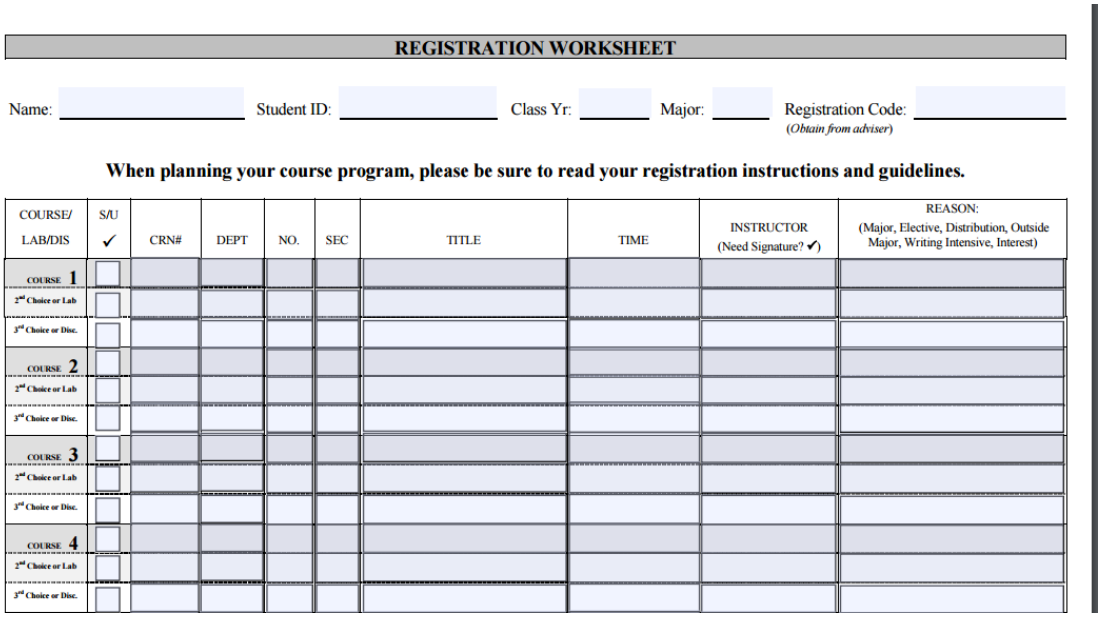
The results of your search are shown below. Click on a course title for full information and description. Click on a department to view complete departmental listings. If you searched only by department and term, cross-listed courses will be displayed at the bottom of the list.

31 courses found for Fall 2016.

Dept	Subj	Course	Title	Section	CRN	Instructor	Day/Time/Location
ECO	ECO	127	The Magic of the Marketplace	01	21403	J. Miller	T Th 10:30 AM-11:50 AM SEELYE 208
ECO	ECO	150	Introductory Microeconomics	01	10265	R. Kaufman	M W F 11:00 AM-12:10 PM SAB-RD 220
ECO	ECO	150	Introductory Microeconomics	02	10266	J. Miller	T Th 1:00 PM-2:50 PM SEELYE 110
ECO	ECO	150	Introductory Microeconomics	03	14156	R. Bartlett	M W F 9:00 AM-9:50 AM SEELYE 110
ECO	ECO	153	Introductory Macroeconomics	01	15201	M. Mahdavi	M W 2:40 PM-4:00 PM SEELYE 110
ECO	ECO	153	Introductory Macroeconomics	02	14687	E. Savoca	T Th 3:00 PM-4:20 PM WRIGHT 238

Figure 2-1-5-F2: Search Result

There is a registration worksheet and class schedule worksheet that contains pre-defined rows and column which available for download to assist the students in planning their timetable. They should discuss with their adviser and acquire the registration number before access to the system. It is important for the students to ensure that the CRN is correct before and during the registration as different timeslot associated with unique CRN which represents the particular subject for registration. The following figure shows the worksheets for planning the timetable.



REGISTRATION WORKSHEET

Name: _____ Student ID: _____ Class Yr: _____ Major: _____ Registration Code: _____
(Obtain from adviser)

When planning your course program, please be sure to read your registration instructions and guidelines.

COURSE/ LAB/DIS	S/U ✓	CRN#	DEPT	NO.	SEC	TITLE	TIME	INSTRUCTOR (Need Signature? ✓)	REASON: (Major, Elective, Distribution, Outside Major, Writing Intensive, Interest)
COURSE 1									
2nd Choice or Lab									
3rd Choice or Disc.									
COURSE 2									
2nd Choice or Lab									
3rd Choice or Disc.									
COURSE 3									
2nd Choice or Lab									
3rd Choice or Disc.									
COURSE 4									
2nd Choice or Lab									
3rd Choice or Disc.									

Figure 2-1-5-F3: Registration Worksheet

Class Schedule Worksheet

	Mon.	Tues.	Wed.	Thurs.	Fri.
8:00am	A 8 – 8:50am AB 8:30 – 9:50am	A 8 – 8:50am	A 8 – 8:50am AB 8:30 – 9:50 am	A 8 – 8:50am	A 8 – 8:50am AB 8:30 – 9:50 am
9:00am	B 9 – 9:50am BC 9 – 10:20 am	G 9 – 10:20am	B 9 – 9:50am BC 9 – 10:20am	G 9 – 10:20am	B 9 – 50am BC 9 – 10:20am
10:00am	C 10 – 10:50am	H 10:30 – 11:50am	C 10 – 10:50am	H 10:30 – 11:50am	C 10 – 10:50am
11:00am	D 11am – 12:10pm		D 11am – 12:10pm		D 11am – 12:10pm
	L	U	N	C	H
1:00pm	E 1:10 – 2:30pm	J 1 – 2:50pm N 1- 4pm	E 1:10 –2:30pm	L 1 – 2:50pm	E 1:10 – 2:30pm

Figure 2-1-5-F4: Class Schedule Worksheet

During the registration period, the students can login to BannerWeb and then select the term, enter the registration number acquire from the adviser. After succeeding, they just only require entering all the CRN numbers and clicking on "Submit Changes". After that, the student will be directed to the current schedule to ensure all the particular before register the course. If the section is full or closed, the "Action" can be changed to waitlist for further process. After all the process done, it's required the student to click on "Submit Changes" to register the course.

Current Schedule

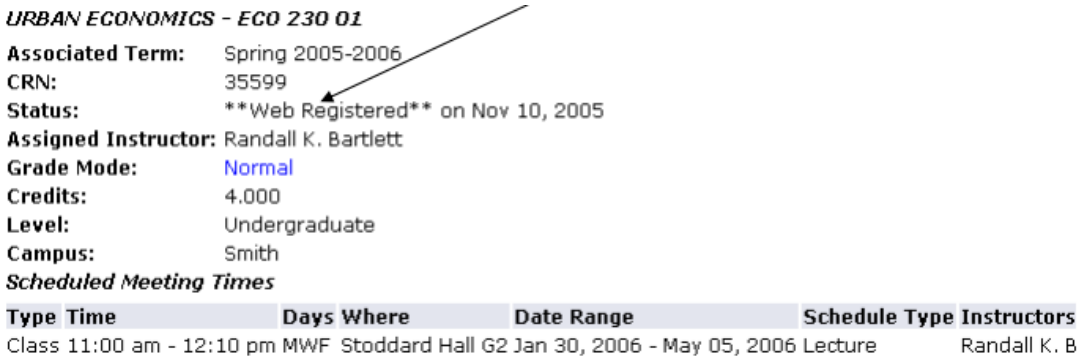
Status	Action	CRN	Subj	Crse	Sec	Level	Cred	Grade Mode	Title
Web Registered on Apr 08, 2013	None	10013	GEO	221	01	Undergraduate	4.000	Normal	MINERALOGY
Web Registered on Apr 08, 2013	None	11338	CHM	111	01	Undergraduate	5.000	Normal	CHEMISTRY I: GENERAL
Web Registered on Apr 08, 2013	None	18851	ESS	945	12	Undergraduate	1.000	Normal	PHYS COND: PILATES MAT TRAIN I
Web Registered on Apr 08, 2013	None	18059	ESS	107	02	Undergraduate	2.000	Normal	EMERGENCY CARE
Registered on Apr 24, 2013	None	10417	CHM	111	L01	Undergraduate	0.000	Normal	CHEMISTRY I: GENERAL

Submit Changes | Class Search | Reset

Figure 2-1-5-F5: Current Schedule

There are error messages will prompt to the students if any of it occurs such as class restriction, course not offered, duplicate, limited course and so on. The possible explanations are available in the user manual. Besides, the students are allowed to

drop the subject during the add/drop period. However, they require the permission of the instructor, and adviser. They should go through the same process as register for a subject to changes the schedule. Once the students have submitted all the registration changes, they can check their schedule by clicking "View Student Schedule" to finalize the registration.



URBAN ECONOMICS - ECO 230 01

Associated Term:	Spring 2005-2006
CRN:	35599
Status:	**Web Registered** on Nov 10, 2005
Assigned Instructor:	Randall K. Bartlett
Grade Mode:	Normal
Credits:	4.000
Level:	Undergraduate
Campus:	Smith

Scheduled Meeting Times

Type	Time	Days	Where	Date Range	Schedule Type	Instructors
Class	11:00 am - 12:10 pm	MWF	Stoddard Hall G2	Jan 30, 2006 - May 05, 2006	Lecture	Randall K. B

Figure 2-1-5-F6: Student Schedule

The course registration system at Smith College does provide the worksheet which helps the students in manage and plan for their schedule more effective. However, it does not provide the feature that can automatically generate the schedule for the students which can save their time in planning their schedule. Besides, the registration process requires the students to enter the CRN which might increase the possibility of the students to enter the error CRN and lead to unnecessary error. The system should provide more user-friendly interface such as radio button to select the course instead of entering the code manually.

2-1-6 Comparison between Benchmark Review

Table 2-1-6-T1: Comparison between Timetabling System

Timetabling System Function/ Features	Automated Timetabling System	Concordia University's Course Registration	Course Registration System of University of	Aurora System in University of Manitoba	Course Registration System in The Ohio State University	BannerWeb Online Registration System of Smith
Auto-registration	✓				✓	
Search Tool	✓	✓	✓	✓	✓	✓
Email Notification	✓					
Course Registration	✓	✓	✓	✓	✓	✓
Error Message	✓	✓	✓		✓	✓
Course Details Viewing	✓	✓	✓	✓	✓	✓
Timetabling Tool	✓					
Hybrid Mode	✓					
Image Downloading	✓					

2-2 Data Collection

In order to build a system which is meet the requirements by UTAR students, it is necessary to conduct the survey to understand clearly what are the problems currently faced by UTAR students when using the existing course registration system. There are set of questionnaires are distributed to UTAR students from different faculties. The data collection will be discussed in this section.

Based on the research done, Appendix B-1 shows about 70% of the students always clashing their timetable when planning it. Besides, it also indicates that too many classes on a day/ period and venue too far for next class would second and third problems faced by majority of the UTAR students.

Moreover, the result shows in Appendix B-2 indicates that most of the students will spend more of the time than the expectation to plan their timetable. Only 24% of the students always plan their timetable with expected time. Next, more than half of the students say that the lecturer or tutor who teaches the class is the factor for them in planning their timetable. The other factors such as break time between classes, the time class start of the day would be second and third factor as shows in Appendix B-3.

In addition, according to Appendix B-5, about 66% of UTAR students will wrongly plan the timetable due to overlook the schedule displays on screen. This is because the readability is difficult for them to read easy and without overlook the schedule. As result shows in Appendix B-6, only 24% of the students think the schedule display by existing system is easy to read.

Lastly, from Appendix B-7, 64% of the students will choose to use the proposed system to help them in planning the timetable for next trimester. 36% of the students would choose not to do so as they prefer more to plan their timetable by themselves. Thus, based on all analyzed data from the survey, the information is sufficient to build a system which could help UTAR students to plan their timetable in easier way.

Chapter 3: System Design

This chapter will discuss about the entire system design which include system use case, activity diagram, class diagram, database design, wireframe diagram and so forth. There will be 3 main modules in the proposed system and the sub-modules will be developed under the main module. The table below shows the description of the module proposed in the system:

Table 3-1: Main modules and sub-modules of proposed system

Module	Sub-modules	Description	Function s
Login and Logout	None	Security verification of the user before accessing to the system	- Login verification - Logout
Course Preview	None	Allow the student to view the course before registering.	- Search course - View lecturer/ tutor
Course Registration	Manually registration	Allow the student to register the course for next trimester.	- Add course - Update course - Drop course
	Auto-registration	Allow the system help user to register course	- Load timetable - Add course
UTAR Timetabling Tool	Automated Timetabling Tool	Allow the student generate the timetable by using the system	- Generate timetable automatically - Export Image - Send Mail - Save/Load
	IN-HAND Timetabling Tool	Allow the user to generate the timetable manually in handy and interactive way	- Generate timetable manually - Export Image - Send Mail - Save/Load
	Hybrid Mode	Allow the user to change between the module to generate timetable	- Let Me Do! - Let System Do!

3-1 System Use Case

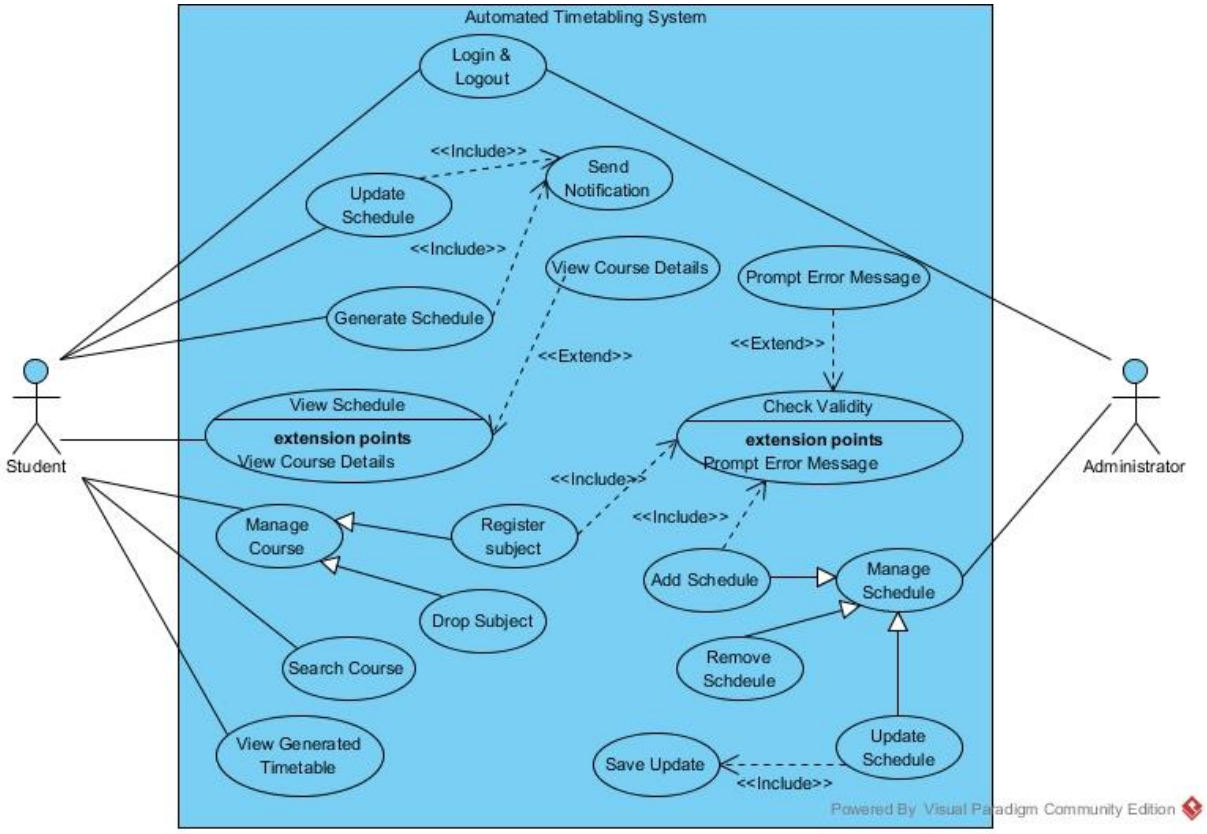


Figure 3-1-F1: Use Case Diagram of Automated Timetabling System

3-2 Activity Diagram

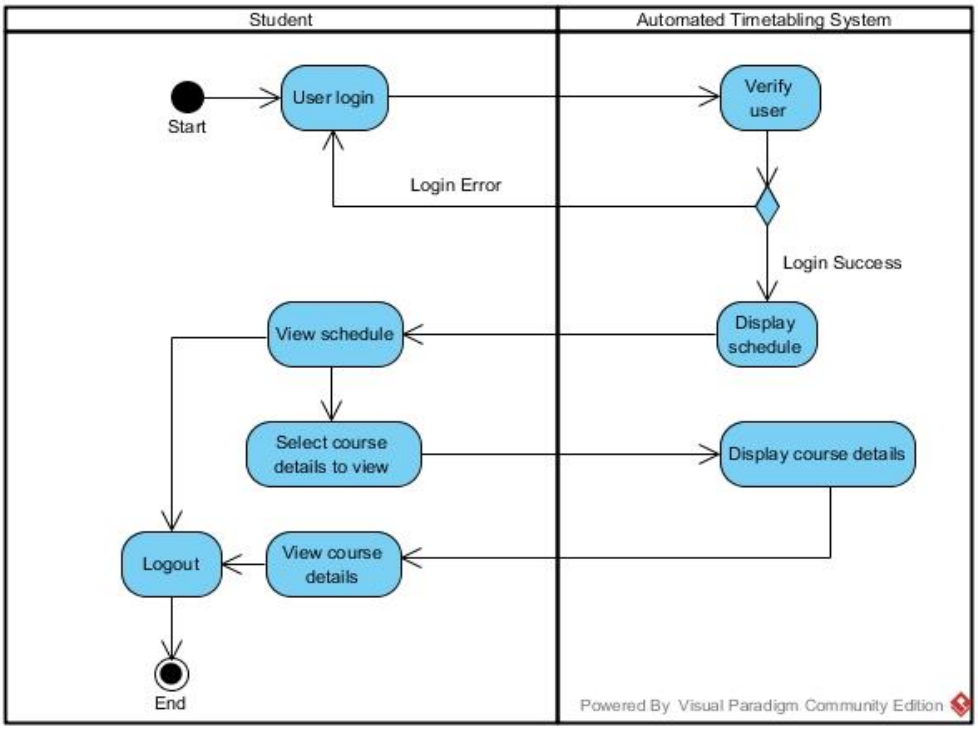


Figure 3-2-F1: Student - View Schedule

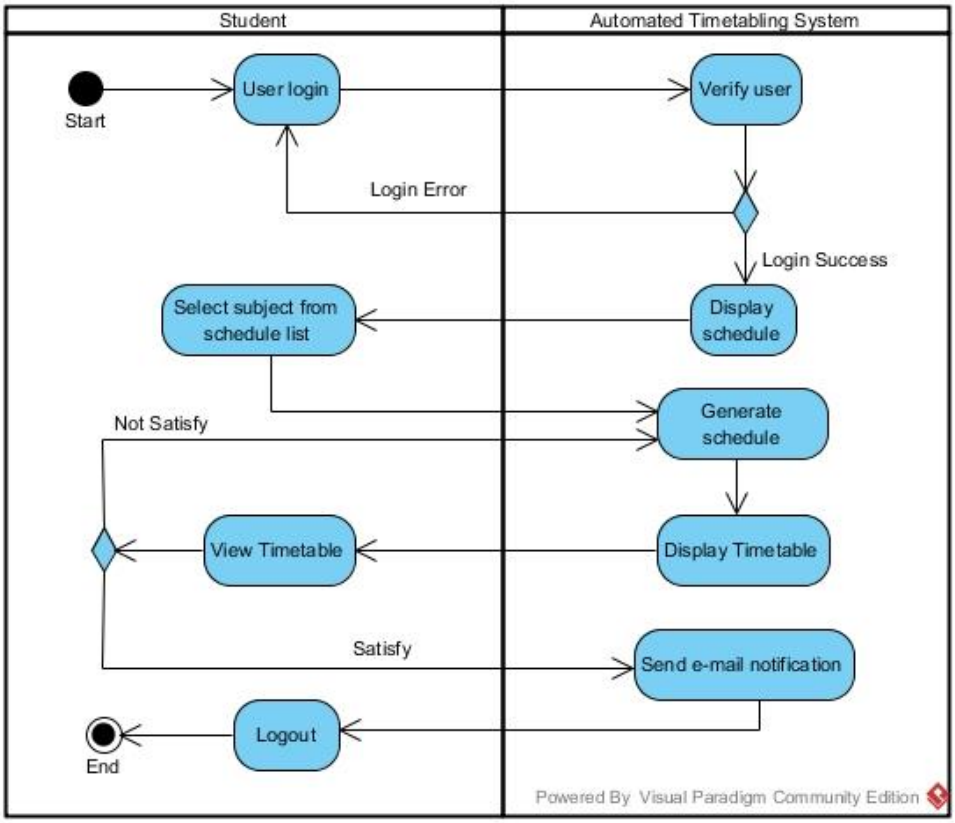


Figure 3-2-F2: Student - Generate Schedule

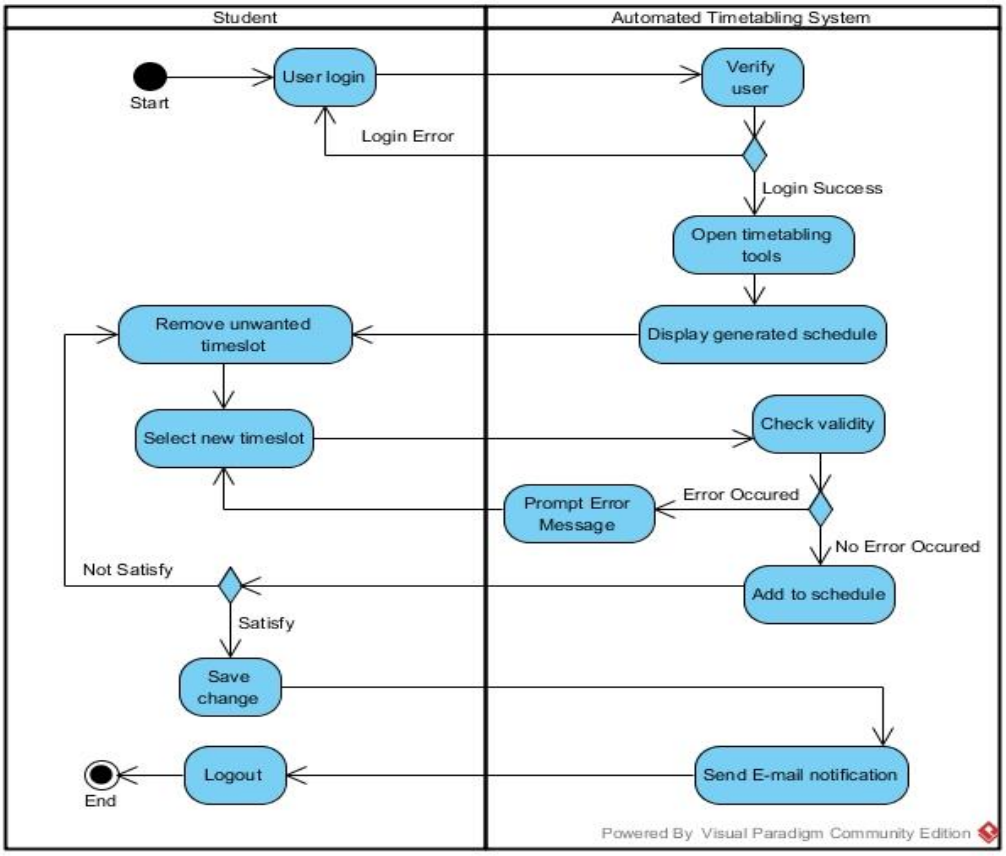


Figure 3-2-F3: Student - Update Timetable

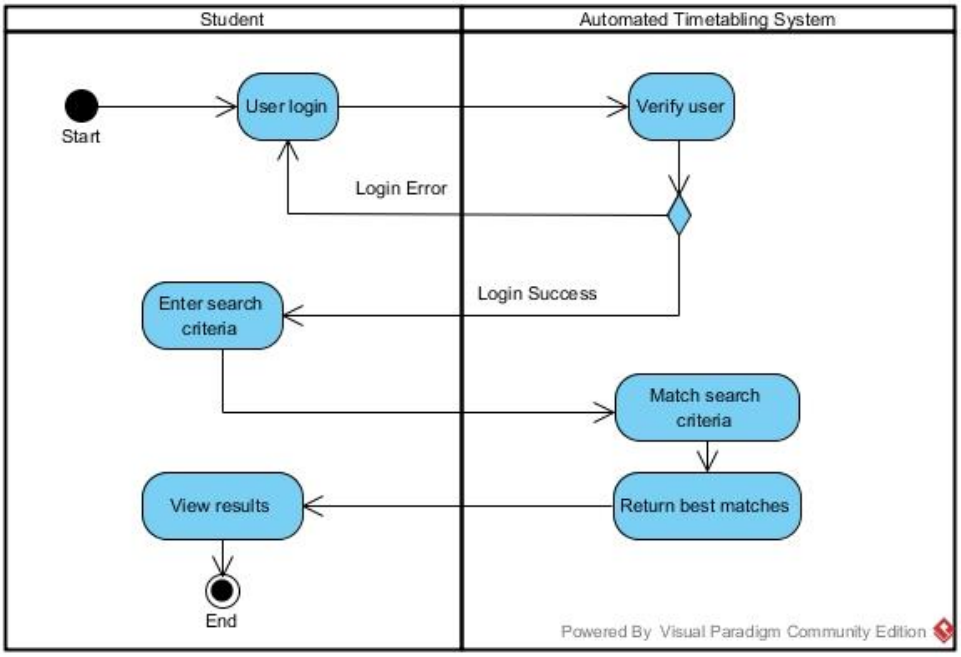


Figure 3-2-F4: Student - Search Course

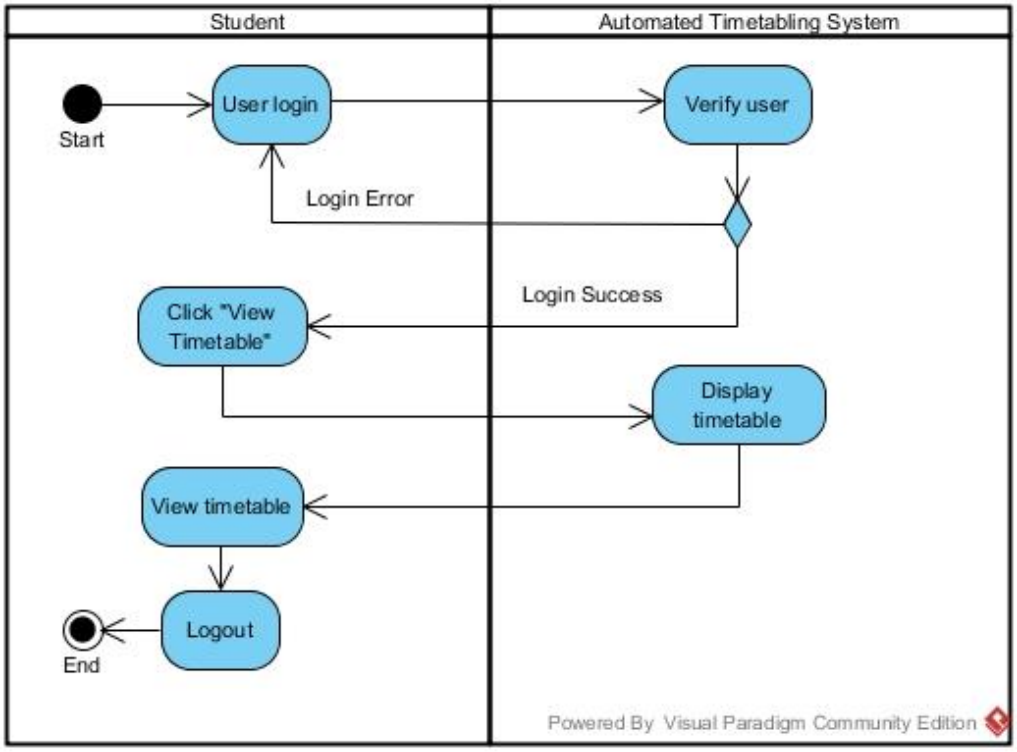


Figure 3-2-F5: Student - View Generated Timetable

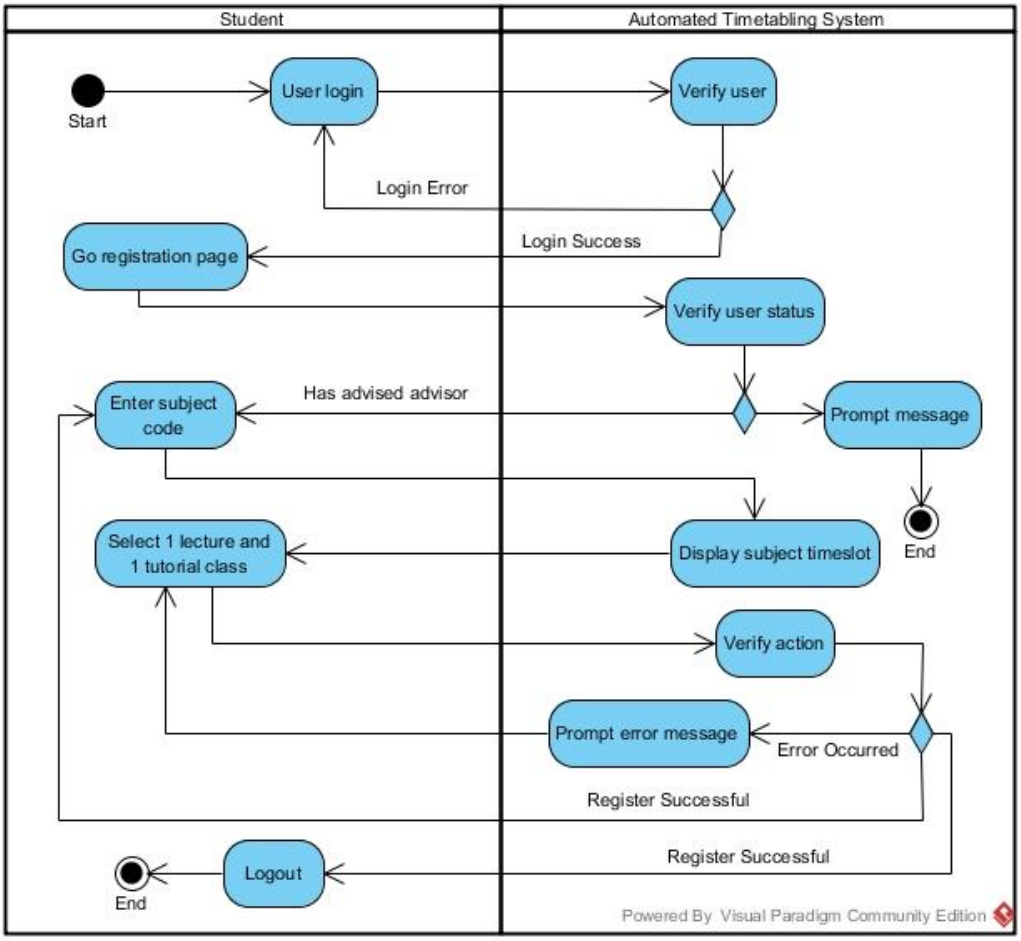


Figure 3-2-F6: Student - Register Subject

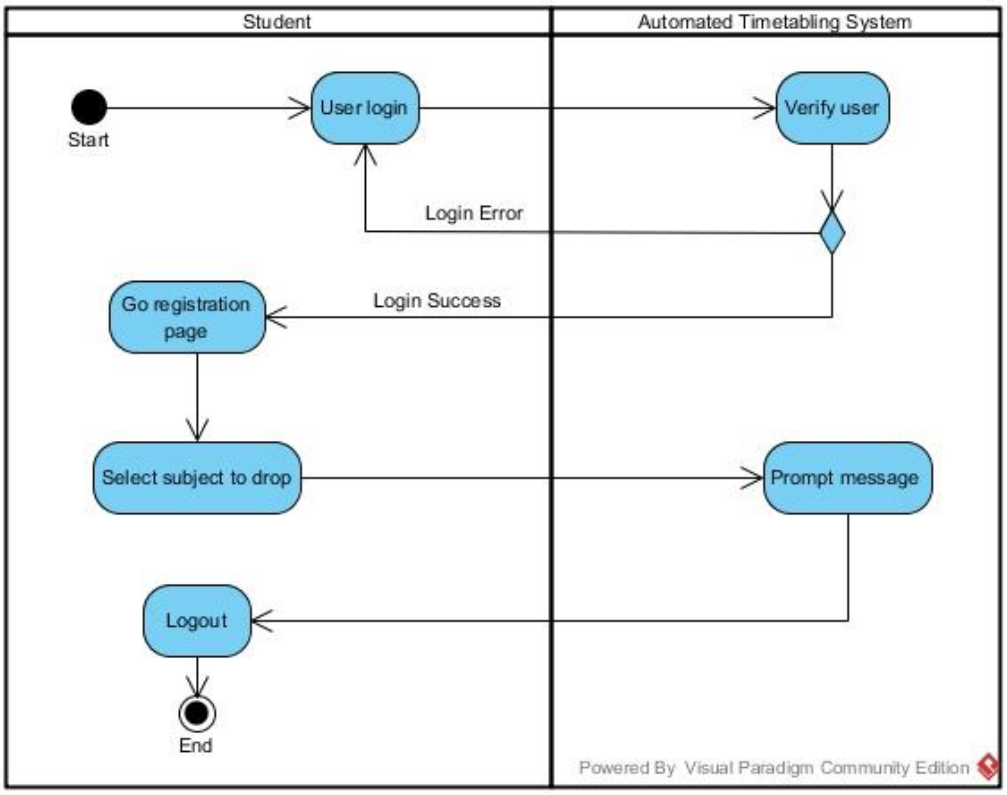


Figure 3-2-F7: Student – Drop Subject

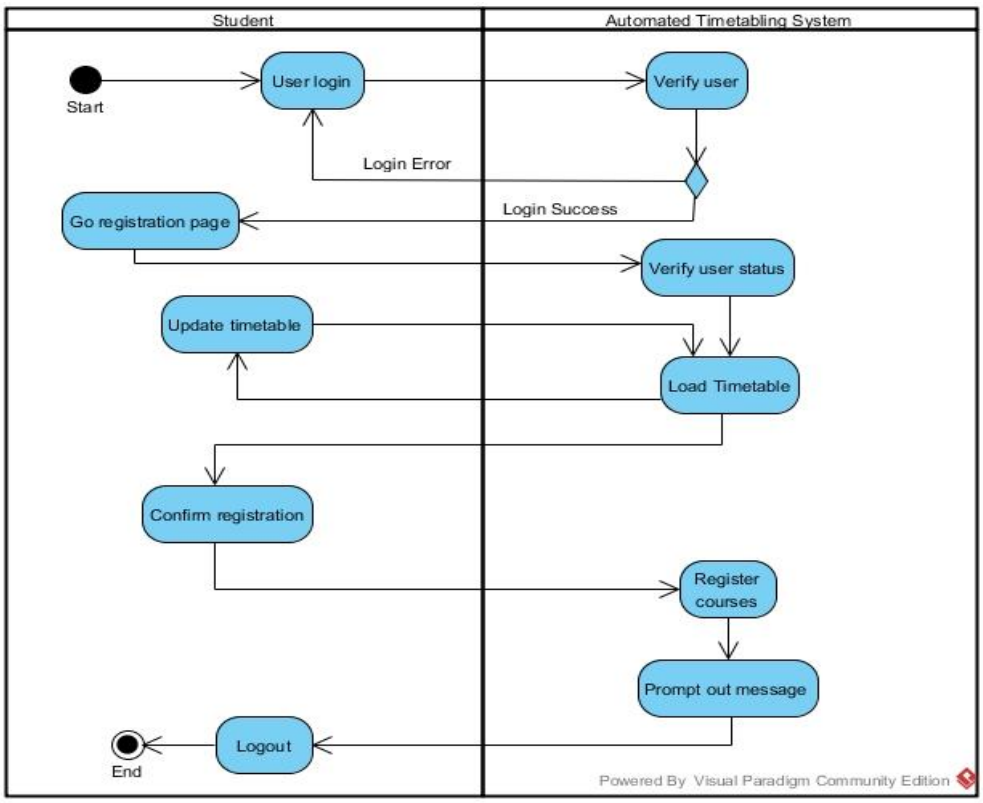


Figure 3-2-F7: Student – Auto Registration

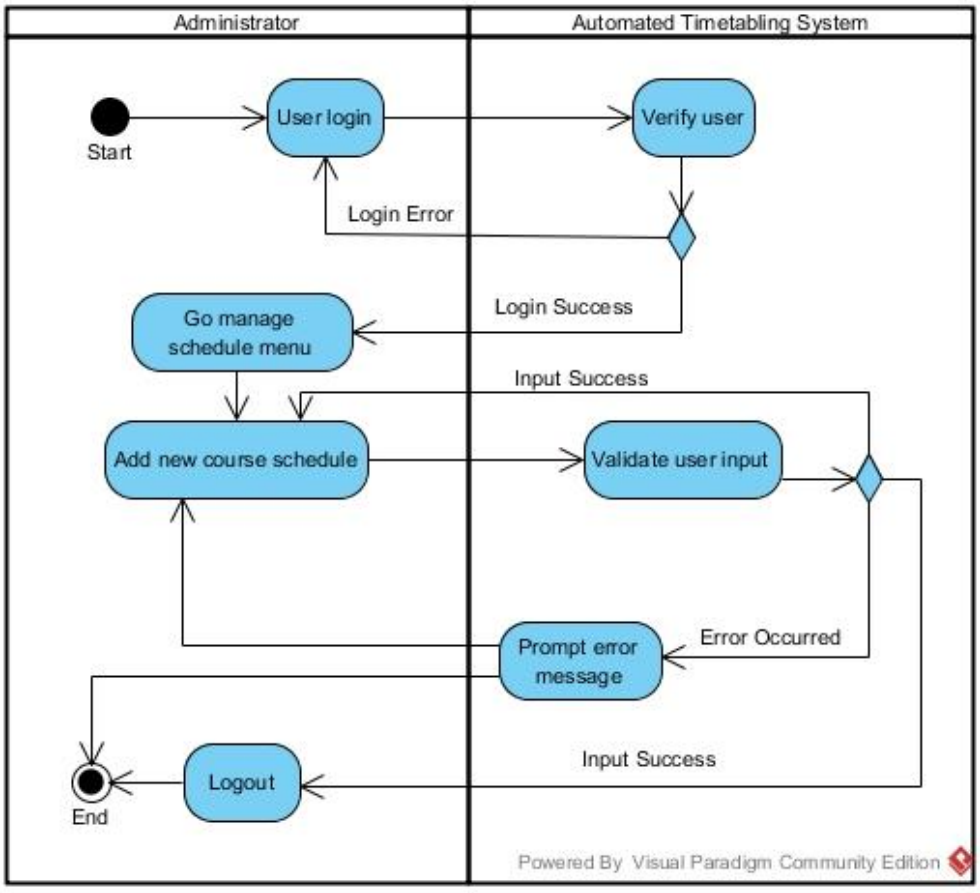


Figure 3-2-F9: Administrator – Add Schedule

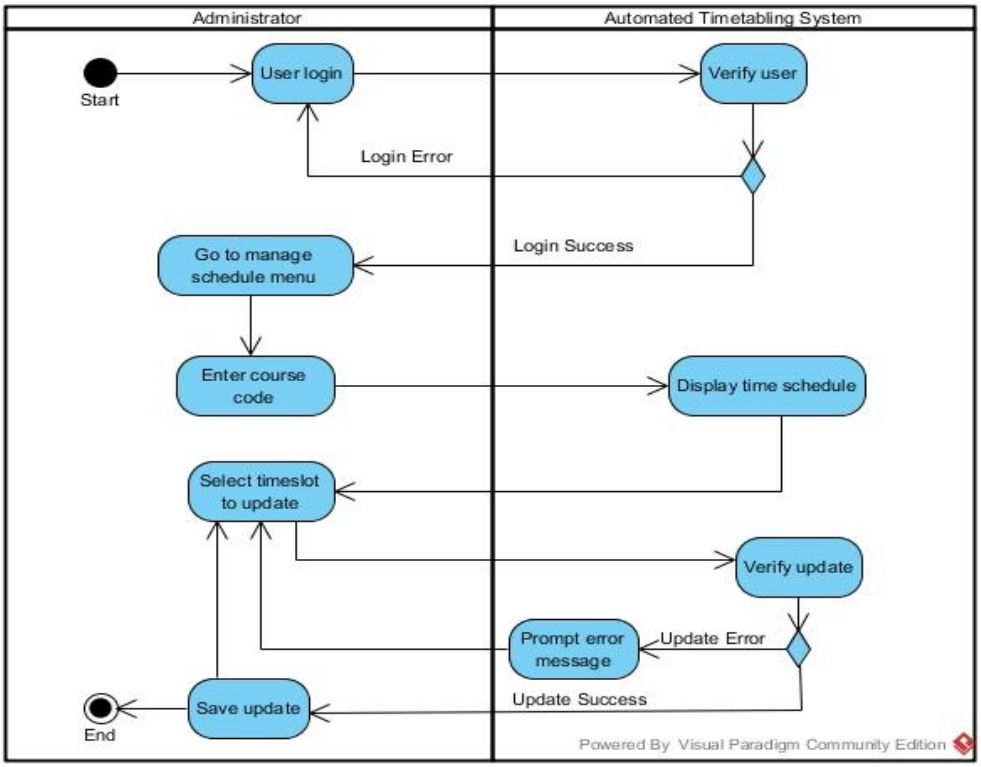


Figure 3-2-F10: Administrator – Update Schedule

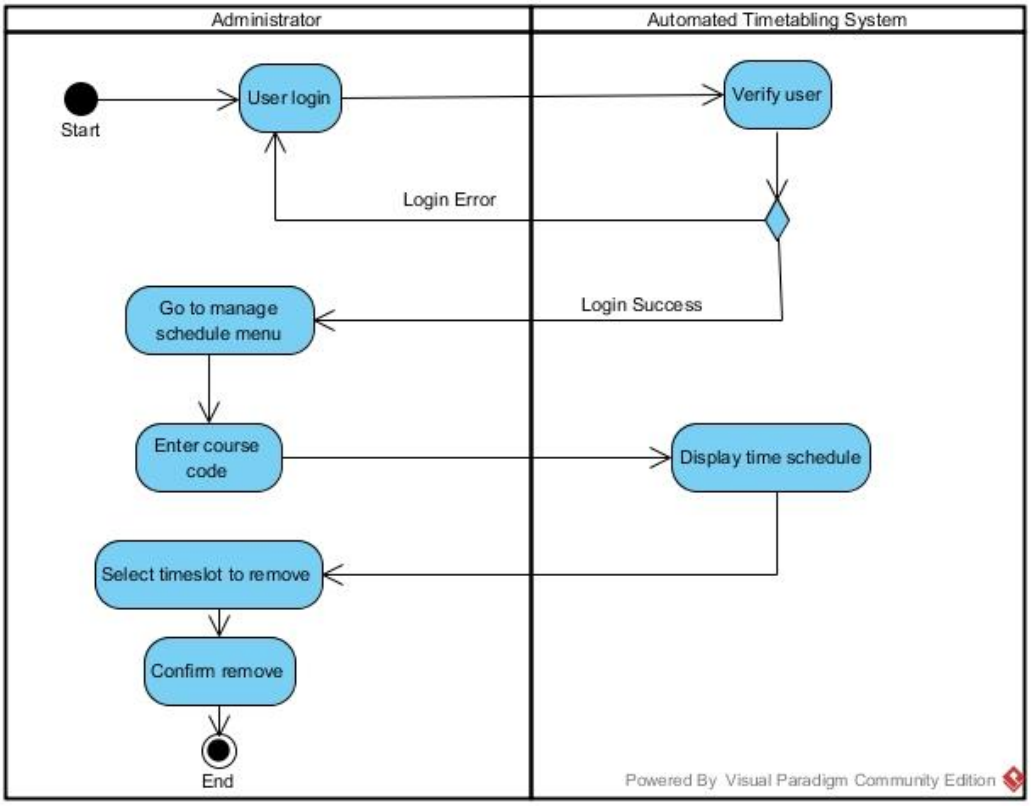


Figure 3-2-F11: Administrator – Remove Schedule

3-3 Design Class Diagram

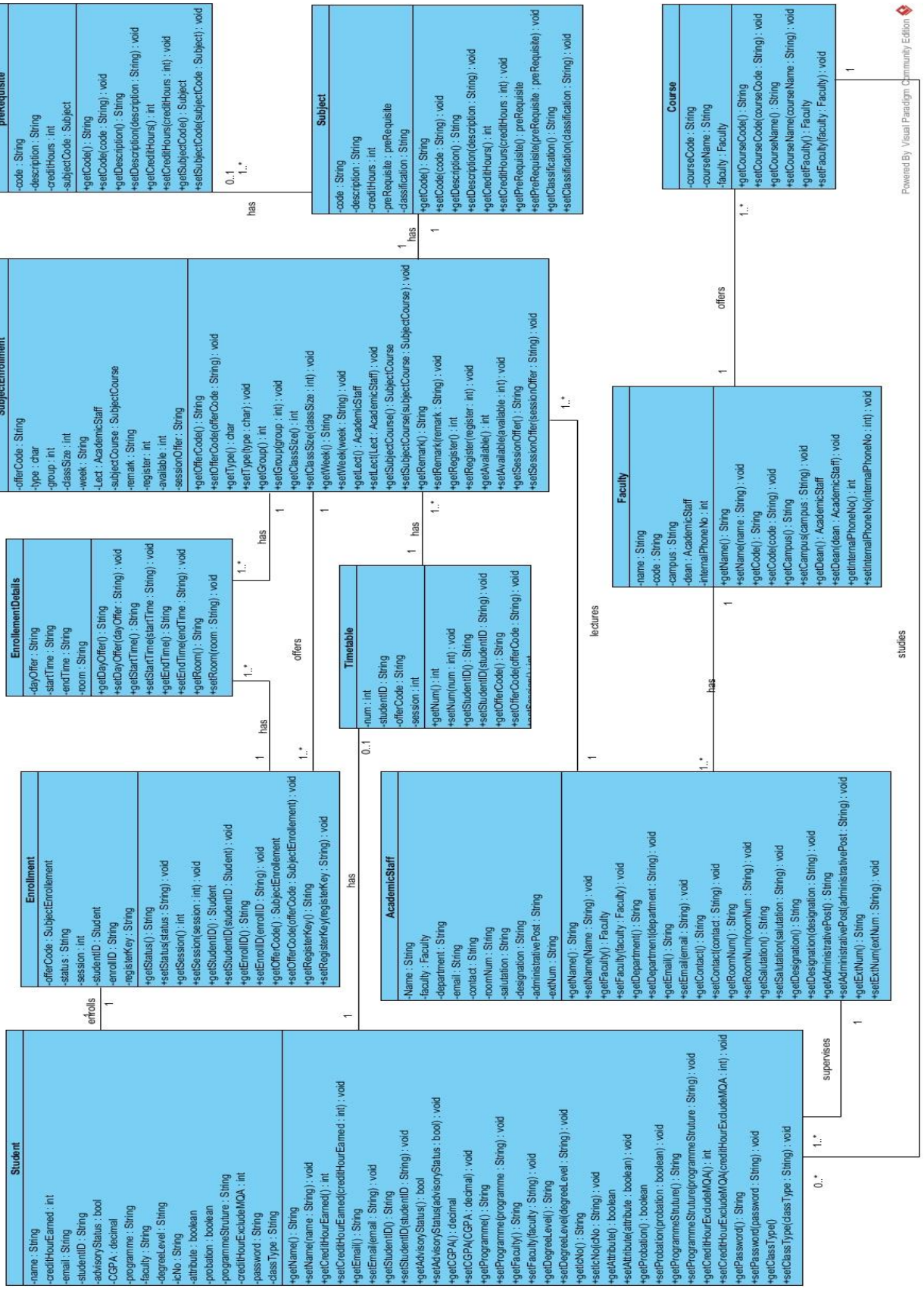


Figure 3-3-F1: Design Class Diagram of Automated Timetabling System

3-4 Wireframe Diagram

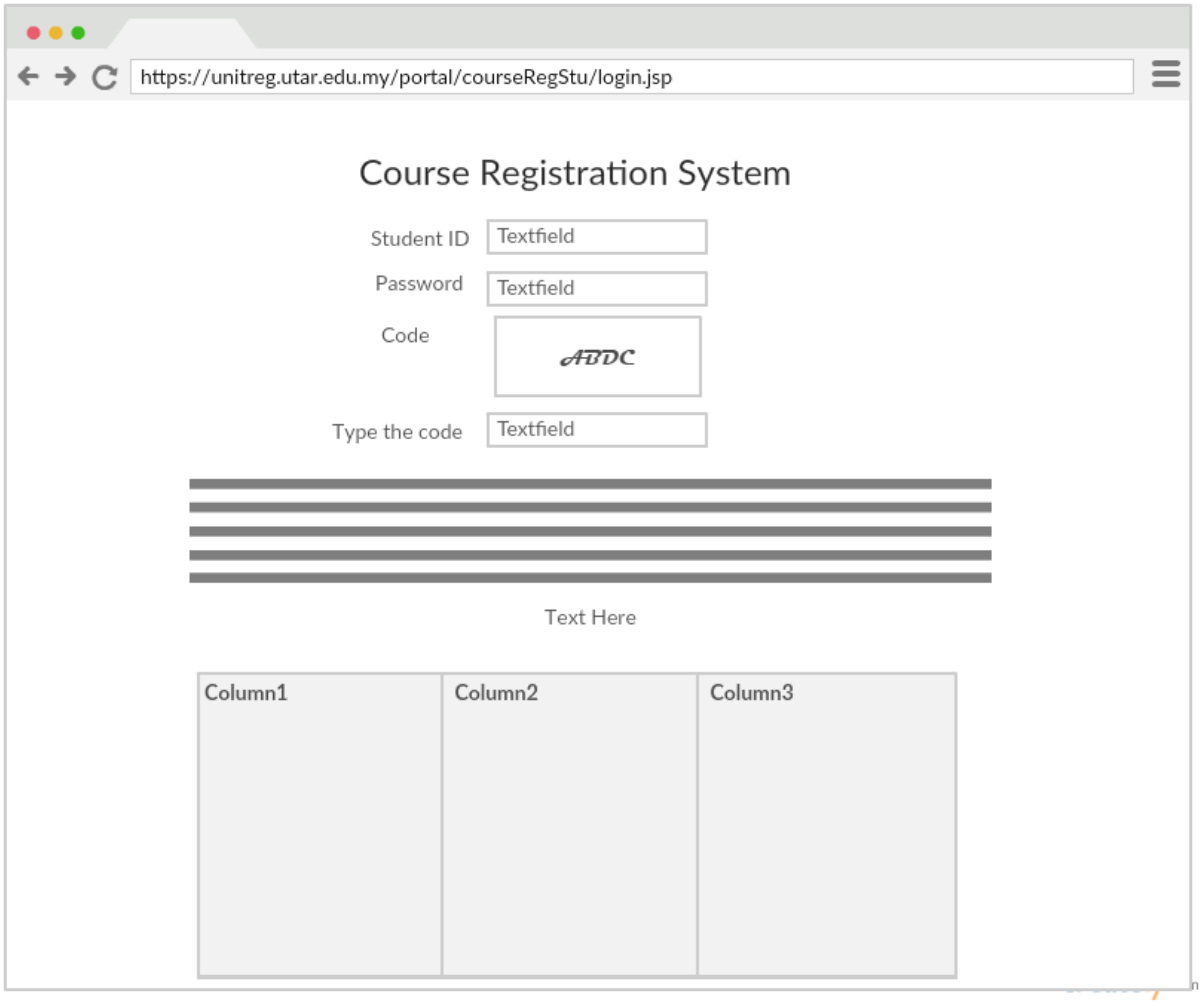


Figure 3-4-F1: User Login Interface

The user requires to login with the student ID, password and captcha code in the text fields. There is a table to shows the course registration period for each faculty.

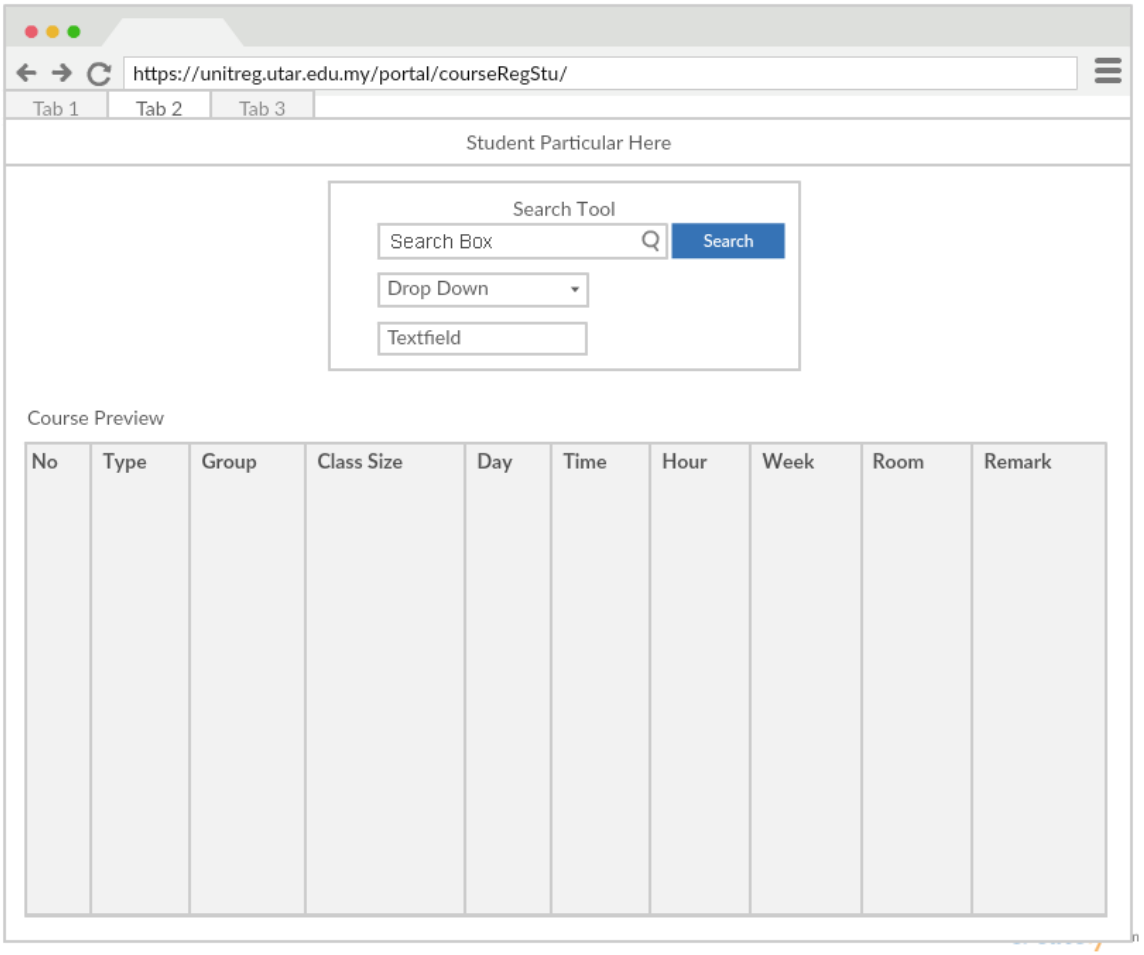


Figure 3-4-F2: Course Preview Interface

There is a list of schedule of the course will be displayed on the screen with details. Each timeslot is categorized according to the subject itself. When the user clicks to expand the particular subject, the details of the subject will display on the screen. Moreover, the user can use the search tool to search for certain subject with some criteria.

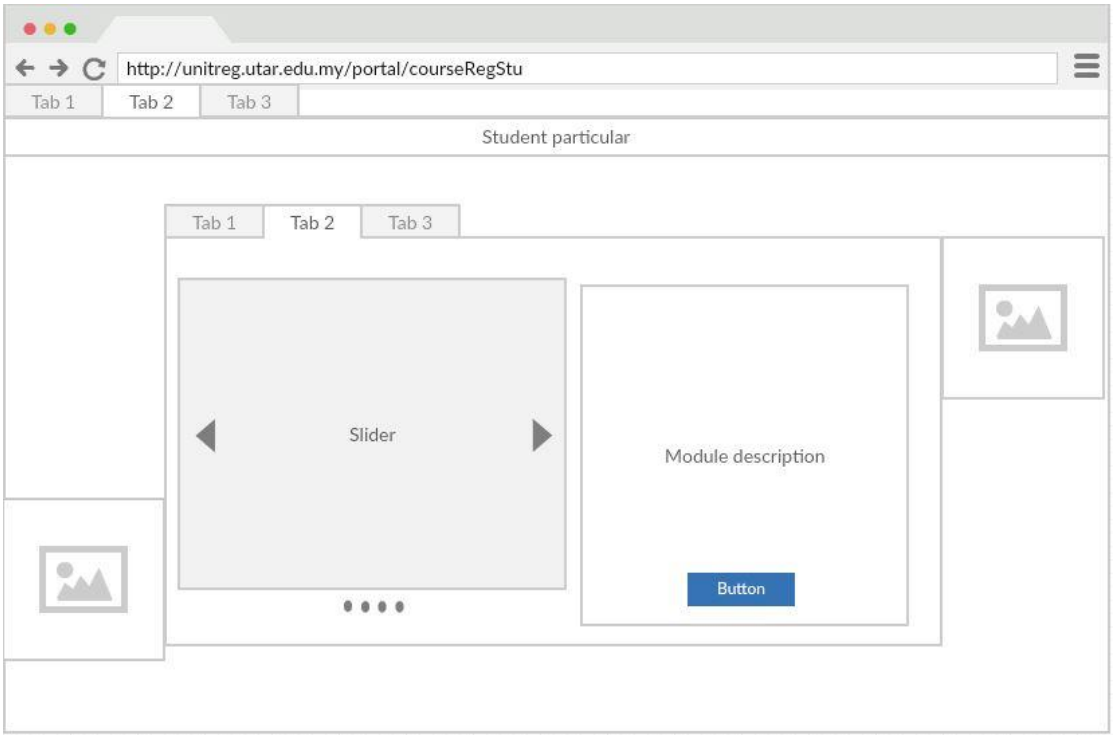


Figure 3-4-F3: Module Selection Interface

When the user first direct to Automated Timetabling System, it will display the module selection interface which allows the user preview the module. There are three tabs which are “Automated Timetabling System”, “IN-HAND Timetabling Tools” and “Hybrid Mode”. Each tab contains the slides and description of each module. There is also a button to direct the user to the selected module.

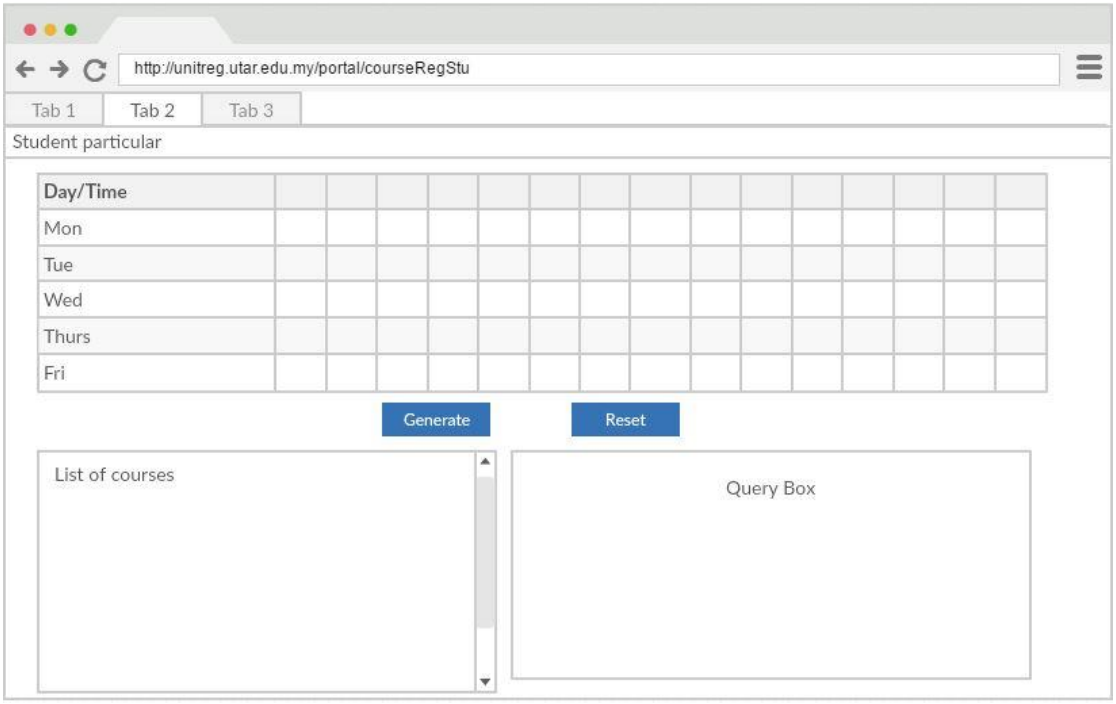


Figure 3-4-F4: Automatically Generating of Timetable Interface

The user just requires to select the subjects from the list and clicks on "Generate Schedule", it will take few seconds to generate and prompt out a message once generated. Besides, the user can select the day and time from query box which they would not like to have class. However, it might lead to the problem of could not generate of timetable due to no time is available for querying.

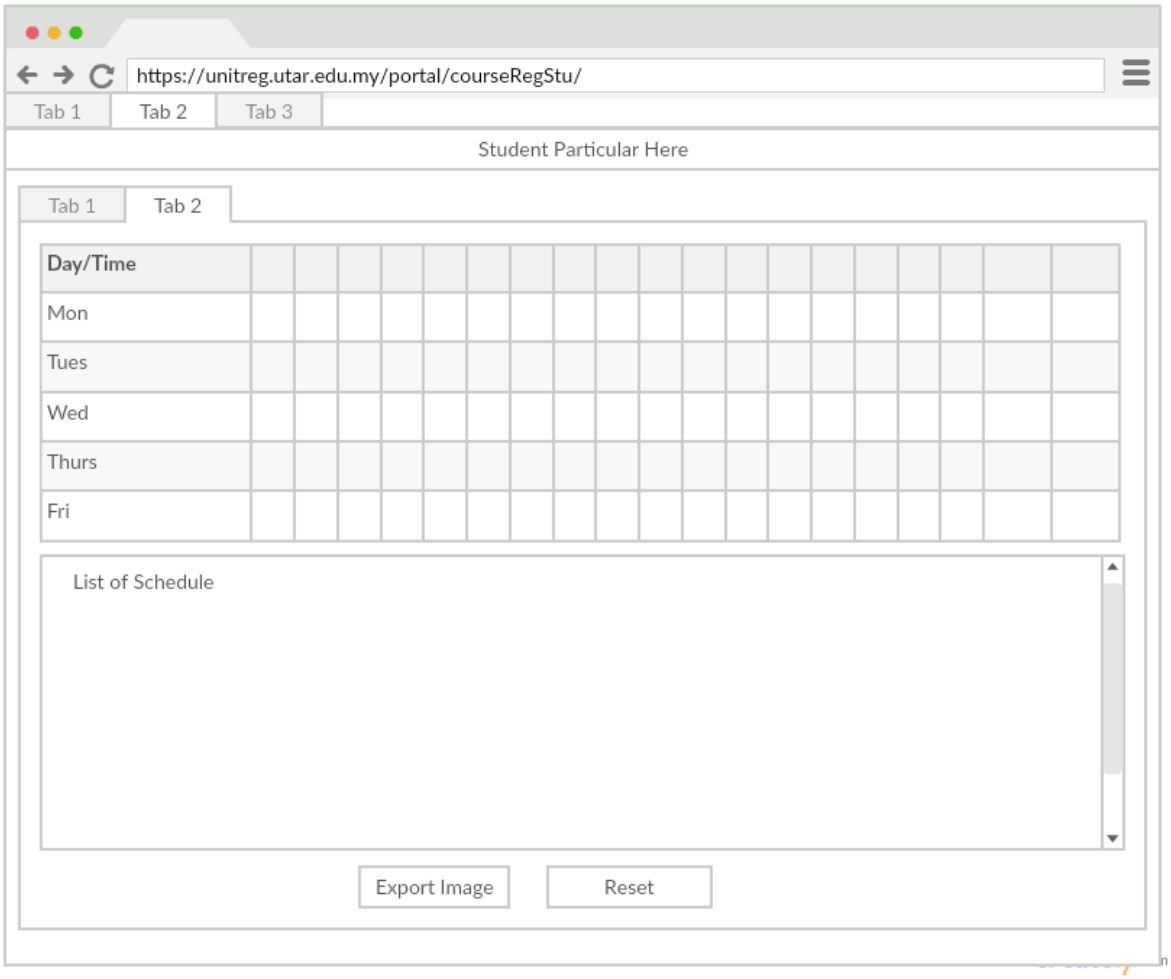


Figure 3-4-F5: IN-HAND Timetabling Tool

The user can go to on the second module to manually to plan for his/ her timetable based on own preferences. With the help of the timetabling tool, the user can easily to view the timetable while he/ she planning for it. After finish planning, the user can export the image as a record for later registration.

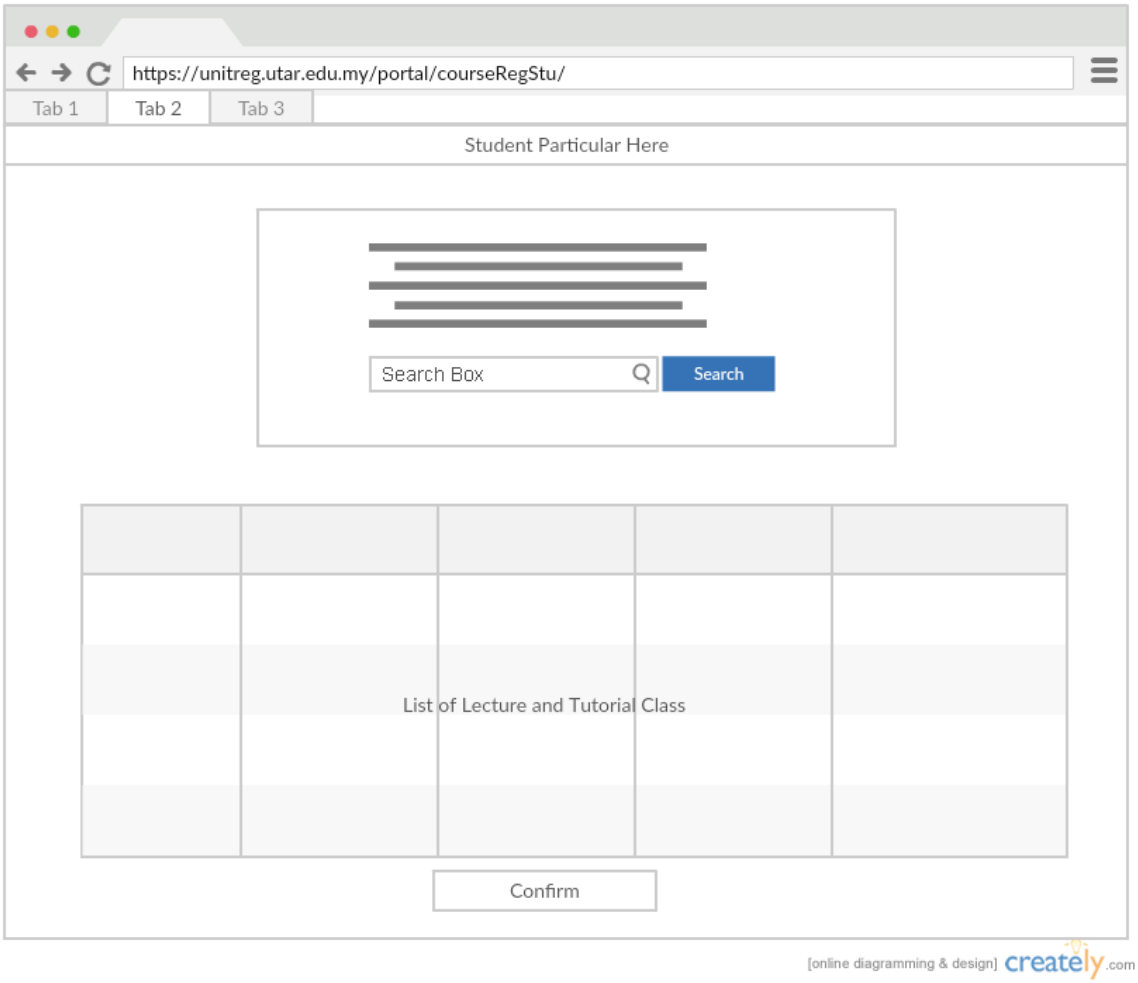


Figure 3-4-F6: Course Registration Interface

During the registration period, the user can click the third tab on top of the page; the registration is opening for the user. The user should enter the subject code on the search toolbar to register the subject. It will return the subject's lecture, tutorial, and practical class. Once finish selecting, the user requires to click "Confirm" to register the subject.

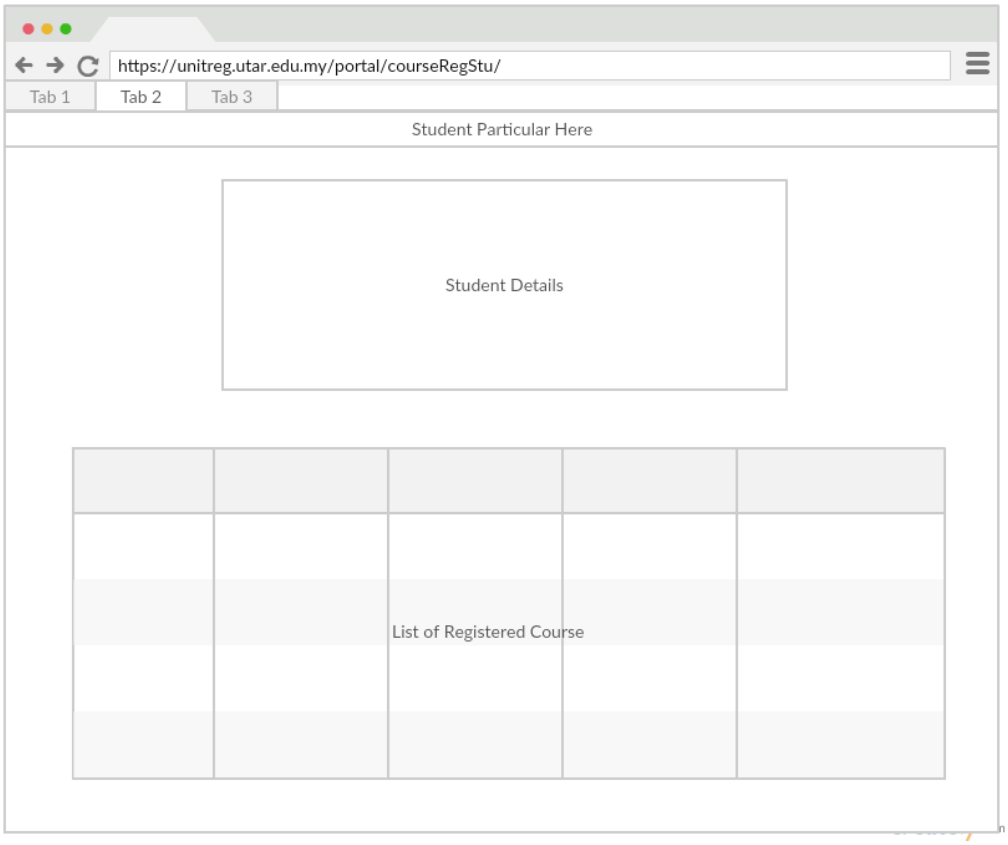


Figure 3-4-F7: View of Registered Course

After register for all courses, the user can view the class he/ she register on the table list. The user can further to remove or update the class from the list.

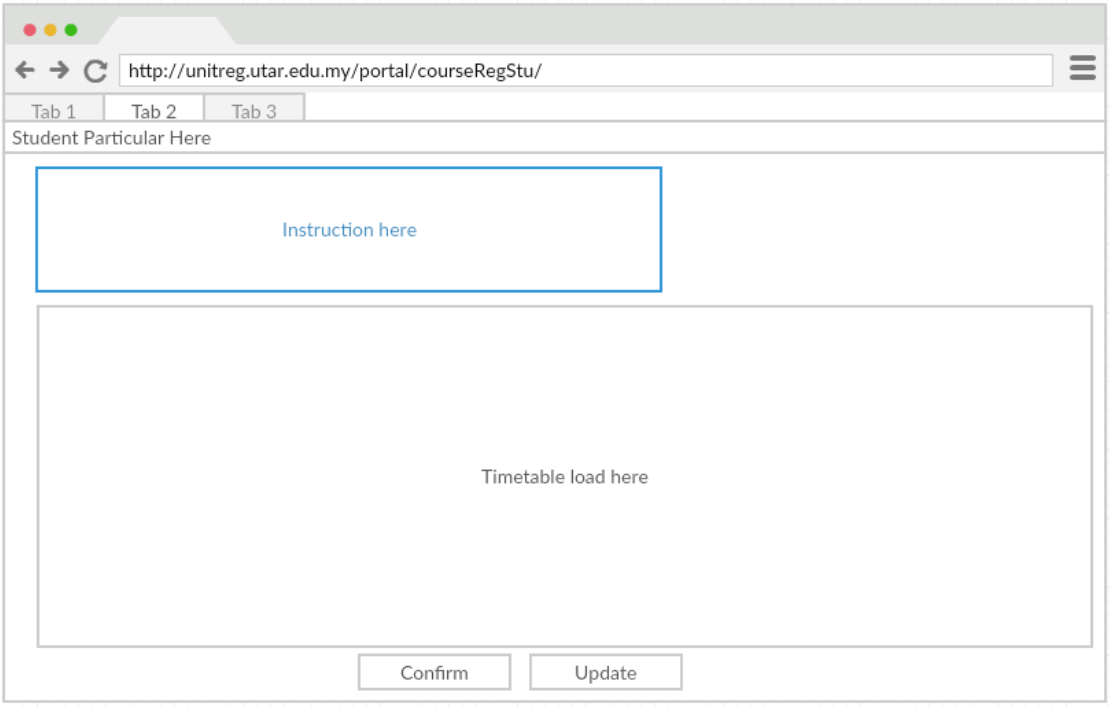


Figure 3-4-F8: Auto-registration Interface

When the user access to page, it will load the timetable which have saved by using timetabling tool before. After that, user can confirm the timetable and register those courses by using the system automatically. Furthermore, the user can further update the timetable by clicking “Update” button which to direct the user to IN-HAND Timetabling Tool Interface.

3-5 Database Design and Implementation

3-5-1 Entity Relationship Diagram (ERD)

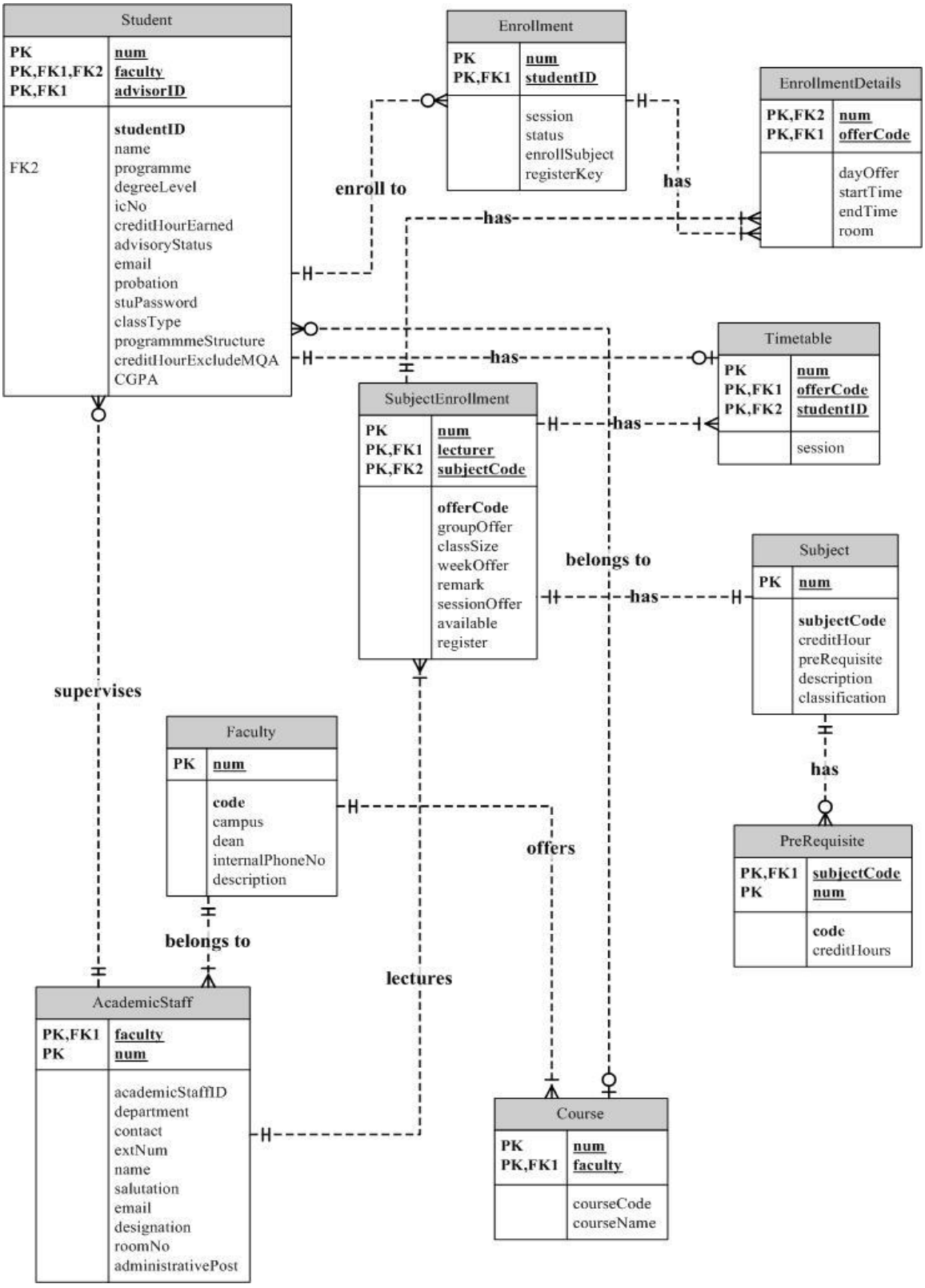


Figure 3-5-1-F1: Entity Relationship Diagram of System

3-5-2 Data Dictionary

Table 3-5-2-T1: Student Entity

Name	Type	Primary Key	Foreign Key	Unique Key	Entity	Description
num	int	Yes	No	Yes	-	Auto increment
studentID	varchar	No	No	Yes	-	Uniquely identify each student
name	varchar	No	No	No	-	Student's name
programme	varchar	No	Yes	No	Course	Student's course
degreeLevel	varchar	No	No	No	-	Student's degree level
icNo	varcahr	No	No	No	-	Student identity card number
advisoryStatus	varchar	No	No	No	-	Student's advisory status
email	varchar	No	No	No	-	Student's email
probabtion	varchar	No	No	No	-	Student's probation status
stuPassword	varchar	No	No	No	-	Student's login password
classType	varchar	No	No	No	-	Student's class type
programmeStruture	varchar	No	No	No	-	Programme structure
creditHourExclude MQA	int	No	No	No	-	Credit hour earned exclude MQA
CGPA	decimal	No	No	No	-	Student's CGPA
creditHourEarned	int	No	No	No	-	Total credit hour earned
faculty	varchar	No	Yes	Yes	Faculty	Student's faculty
advisorID	varchar	No	Yes	Yes	Academic Staff	Student's advisor

Table 3-5-2-T2: Enrollment Entity

Name	Type	Primary Key	Foreign Key	Unique Key	Entity	Description
num	int	Yes	No	Yes	-	Auto increment
studentID	varchar	No	Yes	Yes	Student	Student's ID
session	varchar	No	No	No	-	Enrollment's session
enrollSubject	varchar	No	Yes	Yes	Enrollment Details	EnrollementDetails's offer code
status	varchar	No	No	No	-	Payment status
registerKey	varchar	No	No	No	-	Register time stamp and offer code

Table 3-5-2-T3: EnrollmentDetails Entity

Name	Type	Primary Key	Foreign Key	Unique Key	Entity	Description
num	int	Yes	No	Yes	-	Auto increment
offerCode	varchar	No	Yes	Yes	SubjectEnrollment	Subject's offer code
dayOffer	varchar	No	No	No	-	Day offer the subject
startTime	varchar	No	No	No	-	Start time
endTime	varchar	No	No	No	-	End Time
room	varchar	No	No	No	-	Venue/ room

Table 3-5-2-T4: SubjectEnrollment Entity

Name	Type	Primary Key	Foreign Key	Unique Key	Entity	Description
num	int	Yes	No	Yes	-	Auto increment
offerCode	varchar	No	No	Yes	-	Uniquely identify each record
lecturer	varchar	No	No	Yes	-	Academic staff's ID
subjectCode	varchar	No	No	Yes	Subject	Subject's code
classSize	int	No	No	No	-	Class size
groupOffer	varchar	No	No	No	-	Group offer
weekOffer	varchar	No	No	No	-	Week offer
remark	varchar	No	No	No	-	Remark
available	int	No	No	No	-	Number of seat available to register
register	int	No	No	No	-	Number of seat is registered

Table 3-5-2-T5: Subject Entity

Name	Type	Primary Key	Foreign Key	Unique Key	Entity	Description
num	int	Yes	No	Yes	-	Auto increment
subjectCode	varchar	No	No	Yes	-	Uniquely identify each subject
creditHour	int	No	No	No	-	Subject's credit hour
preRequisite	varchar	No	Yes	Yes	preRequisite	Subject's pre-requisite
description	varchar	No	No	No	-	Subject description
classification	varchar	No	No	No	-	Subject's classification

Table 3-5-2-T6: Faculty Entity

Name	Type	Primary Key	Foreign Key	Unique Key	Entity	Description
num	int	Yes	No	Yes	-	Auto increment
code	varchar	No	No	Yes	-	Uniquely identify each faculty
campus	varchar	No	No	No	-	Faculty's campus
dean	varchar	No	Yes	Yes	AcademicStaff	Faculty's dean
internalPhone No	varchar	No	No	No	-	Internal phone number
description	varchar	No	No	No	-	Faculty description

Table 3-5-2-T7: AcademicStaff Entity

Name	Type	Primary Key	Foreign Key	Unique Key	Entity	Description
num	int	Yes	No	Yes	-	Auto increment
academic StaffID	varcahr	No	No	Yes	-	Uniquely identify each staff
department	varchar	No	No	No	-	Staff's department
extNum	varchar	No	No	No	-	Extension number
name	varchar	No	No	No	-	Staff name
contact	varchar	No	No	No	-	Contact number
salutation	varchar	No	No	No	-	Staff's salutation
email	varchar	No	No	No	-	Staff's email
designation	varchar	No	No	No	-	Staff's designation
roomNo	varchar	No	No	No	-	Room number
administrative Post	varchar	No	No	No	-	Administrative post
faculty	varchar	No	Yes	Yes	-	Staff's faculty

Table 3-5-2-T8: Course Entity

Name	Type	Primary Key	Foreign Key	Unique Key	Entity	Description
num	int	Yes	No	Yes	-	Auto increment
courseCode	varchar	No	No	Yes	-	Uniquely identify each course
courseName	varchar	No	No	No	-	Course name
faculty	varchar	No	Yes	Yes	-	Faculty offer

Table 3-5-2-T9: PreRequisite Entity

Name	Type	Primary Key	Foreign Key	Unique Key	Entity	Description
num	int	Yes	No	Yes	-	Auto increment
code	varchar	No	No	Yes	-	Uniquely identify each record
subjectCode	varchar	No	Yes	Yes	Subject	Subject's code
creditHours	int	No	No	No	-	Credit hour constraint

Table 3-5-2-T10: Timetable Entity

Name	Type	Primary Key	Foreign Key	Unique Key	Entity	Description
num	int	Yes	No	Yes	-	Auto increment
offerCode	varchar	No	Yes	No	SubjectEnrollment	Subject's offer code
studentID	varchar	No	Yes	Yes	Student	Student's ID
session	int	No	No	No	-	Current academic session

3-6 System Hierarchy Diagram

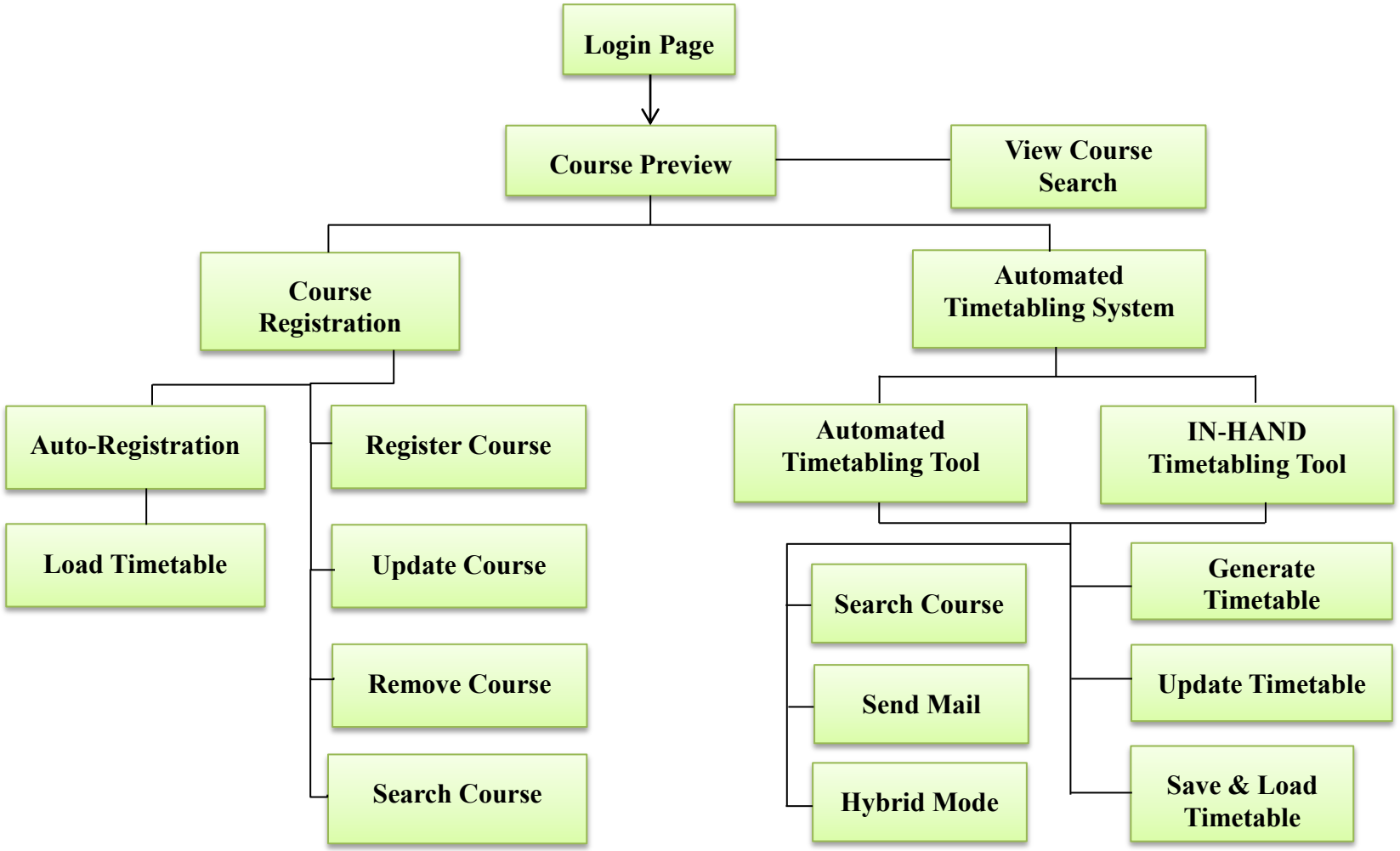


Figure 3-6-F1: System Hierarchy Diagram of System

3-7 Modules and Algorithms Design

This section discusses about how the modules and algorithms would be designed in this proposed system. The following algorithms and techniques will be designed and discussed while implementing the modules:

- Randomization, Swapping and Permutation
- Slot Matching Technique
- 3-phases Looping
- Dynamic Timetable Generation

Besides, only significant modules would be discussed in this section, the remaining modules would not be discussed. The following modules would be focused:

- Automated Timetabling Tool
- IN-HAND Timetabling Tool
- Hybrid Feature

3-7-1 Algorithms and Techniques

Randomization, Swapping and Permutation

- Randomization : randomly select a course from the given list
- Swapping : change the position between the given elements
- Permutation : generate all possible arrangements of the given list

Theses algorithms are important part while generating the timetable automatically. Since there is a lot of combination to generate a timetable, these algorithms will help to generate a totally random list of course for the students. It is better to generate a timetable randomly instead of generate all combination of it. The following pseudo code will show how the algorithms generate a random list:

```
//randomly swap the element
FUNCTION {
        LIST 1: Store random number
        LIST 2: Temporary store object

        LOOP {
                GENERATE A RANDOM NUMBER X
                IF X NOT CONTAINS IN LIST 1{
                        ADD RANDOM NUMBER X TO LIST 1
                        ADD THE GIVEN LIST'S ELEMENT TO LIST 2
                }ELSE {
                        REDUCE NUMBER OF LOOP
                }
                RETURN LIST 2;
```

```
        }
}

//generate all possible of arrangement
FUNCTION {

        IF LIST SIZE IS 0{
                CREATE NEW LIST X
                ADD EMPTY LIST Y TO LIST X
                RETURN LIST X
        }

        LIST 1: STORE LIST OF COMBINATION TO RETURN
        STRING X: EXTRACT AND REMOVE OF FIRST ELEMENT OF GIEVEN LIST

        LIST 2: STORE RECURSIVE RESULT OF LIST

        LOOP {
                LOOP UNTIL INT A {
                        LIST 3: TEMPORARY STORE RECURSIVE RESULT OF LIST
                        ADD STRING X TO LIST 3 AT POSITION INT A
                        ADD LIST 3 TO LIST 1
                }

        }
        RETURN LIST 1
}
```

Slot Matching Technique

Time clashing should be avoided to ensure an error free timetable could be generated to students. It is important to check the timeslot availability to add a new course before the course is selected to add to timetable. The following are the condition should be checked before the course is available to add:

1. Slot clashing with different subject
2. Slot clashing with same subject itself (e.g. tutorial clashing with lecture class)
3. Slot clashing with constraint/ query set

The following pseudo code shows how the time clashing could be check:

```
//global variable
RESERVE LIST: STORE STRING OF SLOTS WHICH ARE OCCUPIED
GIVEN LIST: STRORE STRING OF SLOTS WHICH READY TO ADD TO TIMETABLE

//add day time to reserve slot
FUNCTION A {
        INT X: GET THE NUMBER OF LOOP FROM FUNCTION B

        LOOP UNTIL INT X {
                STRING Y: FORMAT OF "DAY + START_TIME"
                ADD STRING Y TO RESERVE LIST
                ADD EVERY 30 MINS TO CURRENT START_TIME
```

```
        }
}

//get number of loop
FUNCTION B {
        RETURN [(START_TIME) – (END_TIME)]/60
}

//to match the given list with reserve list
FUNCTION {
        FLAG X: FLASE IF CONTAINS IN SLOT RESERVE

        LOOP {
                IF ELEMENT IN GIVEN LIST MATCH WITH RESERVE LIST{
                        SET FLAG TO FALSE
                        BREAK LOOP
                }
        }
        RETURN FLAG
}
```

3-phases Looping

This algorithm is a looping which contains of 3-phases to completely get a course to add to timetable. The loop also adapts randomization and tree selection to select the elements to be added. The following are the description of 3 phases:

- Phase 1 : random course selection looping, since class in each course is not mutually exclusive to each other
- Phase 2 : random group selection looping, since class in each group is not mutually exclusive to each other
- Phase 3 : random class selection looping

*Not mutually exclusive: the sequence of selection is important and dependent

Consider the following case, 3 courses which 2 lecture classes and 3 tutorial classes for course A, 2 lecture classes and 2 practical classes for course B as well as 2 lecture classes and 3 tutorial classes for course C. The tree diagram would be showed in Figure 3-7-1-F1 and Table 3-7-1-T1 demonstrates the steps what will happen during looping.

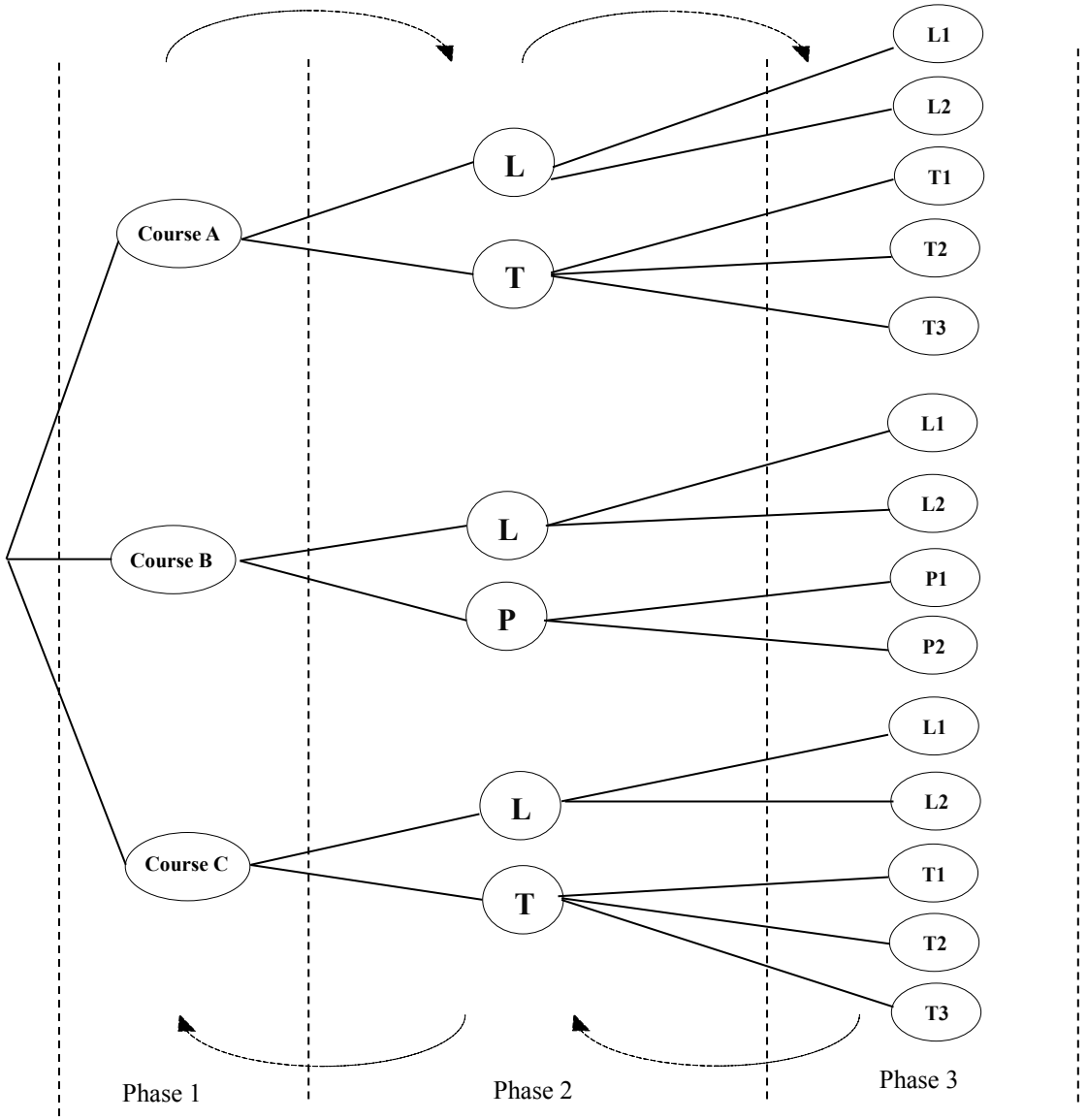


Figure 3-7-1-F1: 3-phases looping for course, group and class

The assumption for the case is no class will clash with each other. In real case, there is possible that the class will clash with each other frequently. Thus, necessary approach will be taken and discussed details in module design. The approaches taken will be as followed:

- slot availability checking
- while loop implementation
- time checker to break while loop

Table 3-7-1-T1: Steps involve in 3-phases Looping

Step	Phase 1		Phase 2		Phase 3	
	Action	**Result**	**Action**	**Result**	**Action**	**Result**
1	Randomly select from course A, B, C	Course A				
2			Randomly select from lecture and tutorial	Tutorial class		
3					Randomly select tutorial class	T2
4			Select lecture class	Lecture class		
5					Randomly select lecture class	L1
6	Return to Phase 1 with {T2, L1}					
7	Randomly select from course B and C	{A – T2, L1}, Course C				
8			Randomly select from lecture and tutorial	Lecture class		
9					Randomly select lecture class	L2
10			Select tutorial class	Tutorial class		
11					Randomly select tutorial class	T3
12	Return to Phase 1 with {L2, T3}					
13	Select course B	{A – T2, L1}, {C – L2, T3}, Course B				
14			Randomly select from lecture and practical	Lecture class		
15					Randomly select lecture class	L2
16			Select practical class	Practical class		
17					Randomly select practical class	P2
18	Return to Phase 1{L2, P2}					
19	Break loop	{A – T2, L1}, {C – L2, T3}, {B- L2, P2}				

Dynamic Timetable Generation

Timetable generation happens in client side instead of server side. Javascript and Document Object Model (DOM) build a timetable which allows student to:

1. Add classes to table cell
2. Remove classes from table cell
3. Clear all classes from table cell

The following HTML code demonstrates how the timetable should be created at first:

```
<table>
<tr>
<th rowspan="2">Day/Time</th>
<th colspan="2>
        <div>07:00</div>
</th>
<th colspan="2">
        <div>08:00</div>
</th>

/ … /

<th colspan="2">
        <div>08:00</div>
</th>
</tr>
<tr id="Mon">
<th>Mon</th>
                <td id="Mon_07.00AM"></td>
                <td id="Mon_07.30AM"></td>
                <td id="Mon_08.00AM"></td>
                <td id="Mon_08.30AM"></td>
                / … /
</tr>
<tr id="Tue">
<th>Tue</th>
                <td id="Tue_07.00AM"></td>
                <td id="Tue_07.30AM"></td>
                <td id="Tue_08.00AM"></td>
                <td id="Tue_08.30AM"></td>
                / … /
</tr>

/ … /
                <td id="Fri _07.00PM"></td>
                <td id="Fri _07.30PM"></td>
</tr>
</table>
```

1. Adding classes to table cell

The following pseudo code in Javascript shows how the timetable could be generated when the classes added to table cell:

```
FUNCTION {
        / … /
        VAR X: GET COLSPAN VALUE
        VAR Y: GET TABLE CELL OBJECT BY ID FROM TIMETABLE

        //set necessary attribute to var Y
        SET ATTRIBUTE TO VAR Y: COLSPAN
        SET STYLE TO VAR Y: BACKGROUND COLOR
        SET INNER HTML TO VAR Y: COURSE TITLE + TYPE + GROUP + VENUE + WEEK
        SET CLASS TO VAR Y: NEW CLASS
        / … /

        TRY {
                LOOP UNTIL VAR X {
                        IF VAR Y HAS NEXT ELEMENT SIBLING {
                                REMOVE NEXT ELEMENT SIBLING OF VAR Y
                        }
                }

                / … /
        } CATCH {
                //handle error when generating timetable
                ALERT: CANNOT GENERATE TIMETABLE
        }

        / … /
}
```

2. Remove classes from table cell

The following pseudo code shows how the table cell could be recovered when the student removes the course from the timetable:

```
//global variables
LIST 1: STORE LIST OF CLASSES ADDED TO TIMETABLE

FUNCTION {
        / … /
        VAR X: SELECTED CLASSES FRO LIST 1 TO BE REMOVED
        VAR Y: SELECTED TABLE CELL OBJECT BY ID FROM TIMETABLE
        VAR Z: LENGTH OF COLSPAN OF VAR X

        REMOVE ALL THE ATTRIBUTES FROM VAR Y

        LOOP UNTIL VAR Z {
                VAR A: ID OF THE ELEMENT BEFORE NEW CELL TO BE ADDED

                ASSIGN OF ID TO NEW CELL
                INSERT NEW CELL TO TIMETABLE AFTER ELEMENT ID OF VAR A
```

```
        }

        / ... /
}
```

3. Clear all classes from timetable

The technique of clear all content in timetable is basically replaces and rewrites the existing HTML code with the new one. The following pseudo code shows how the classes from the timetable could be completely removed just by one-click:

```
FUNCTION {

        VAR X: DIV OBEJCT CONTAINS OF TIMETBALE
        VAR Y: INITIAL HTML CODE OF TIMETABLE

        INITALIZE ALL GLOBAL VARIABLES
        SET INNER HTML OF VAR Y TO VAR X
        RESET OF HEIGHT AND WIDTH OF NEW TIMETABLE

}
```

3-7-2 Modules Design

Automated Timetabling Tool

Automated Timetabling Tool is one the core module of the system which proposes to automatically help the student to generate the timetable based on the course selected during pre-registration. The server will respond to the request from the client and generate the JSON String through the algorithm and pass back to the client to display the timetable and Figure 3-7-1-F1 shows how the data passing between client, server and database.

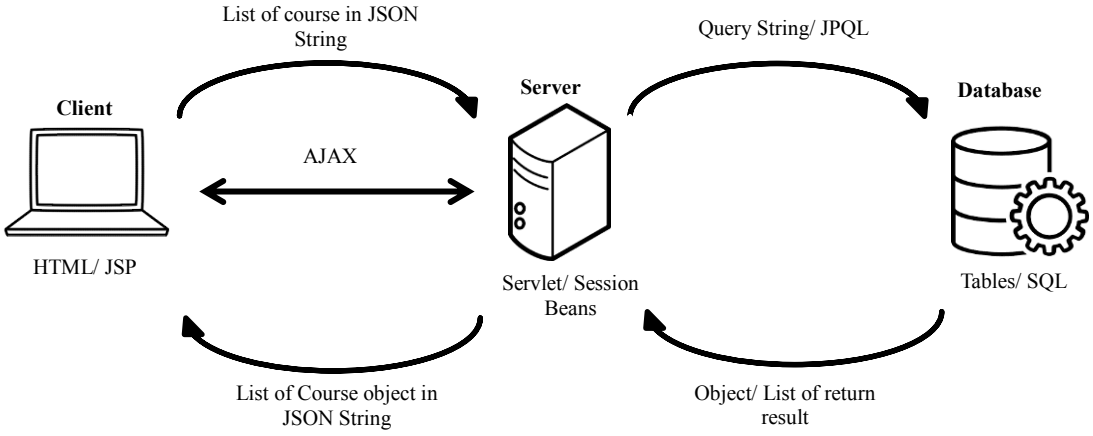


Figure 3-7-1-F2: Basic architecture of Automated Timetabling Tool

Client Side

Student will select course(s) from the list given and the query from the query box. Javascript will validate and format the student input to JSON String and pass to servlet through AJAX. The pseudo code shows how the JSON String passes data to servlet through AJAX:

```
//global variables
LIST 1: STORE LIST OF COURSE CODE
FUNCTION {
        //validate should select at least 1 course from checklist
        $("input[name=course]:checked").each(function () {
                IF THIS OBJECT IS CHECKED{
                        SET FLAG TO TRUE
                        ADD COURSE CODE STRING TO ARRAY
                }ELSE {
                        SET FLAG TO FALSE
                        RETURN ERROR MESSAGE
                }
        }

        //get query set
        $("input[name=day]:checked").each(function () {
                ADD QUERY SET TO ARRAY
        }
}

//function to send data through AJAX
FUNCTION{
        / … /

        VAR CODE: COURSE CODE STRING DATA TO JSON STRING
        VAR QUERY: QUERY SET DATA TO JSON STRING

        / … /

        xhttp.open(SET SERVLET PATH);
        xhttp.setRequestHeader("Content-Type", "application/x-www-form-
        urlencoded");
        xhttp.send(CODE + QUERY);

        / … /
        ON READY STATE CHANGE
        if (xhttp.readyState === 4 && xhttp.status === 200) {
                GET RESPONSE TEXT
        }
}
```

Server Side

The data will be received from client side which contains list of course code in JSON String format. The following pseudo code shows how the JSON String will be parsed

to List object. The following pseudo code shows how the module would be designed:

```
/ … /
import javax.json.JsonReader;
import javax.json.Json;
import javax.json.JsonArray;
/ … /

//global variables
LIST 3: LIST OF COURSE

FUNCTION {
        / … /
        STRING X: COURSE CODE JSON STRING
        STRING Y: QUERY SET JSON STRING
        LIST 1: COURSE CODE ARRAY LIST
        LIST 2: QUERY SET ARRAY LIST

        JsonReader reader = null;
        reader = Json.createReader(new StringReader(STRING X));
         JsonArray JSONArray = reader.readArray();

                LOOP {
                        ADD OBJECT TO LIST
                }

        / … /

        CALL FUNCTION OF 3-PHASES LOOPING

        IF CAN GENERATE TIMETABLE {
                PARSE LIST OBJECT TO JSON STRING
                RETURN JSON STRING
        } ELSE {
                RETURN EMPTY JSON STRING
        }
}

//phase 1
FUNCTION {
        BOOLEAN FLAG_1: DETERMINE IF GENERATE SUCCESSFULLY

        LOOP UNTIL 5 SECONDS OR FLAG_1 IS TRUE {

                CALL FUNCTION TO SWAP LIST 1
                CALL FUNCTION TO PERMUTE LIST 1

                LOOP UNTIL LENGTH OF LIST 1 {

                        LOOP UNTIL LENGTH OF ELEMENT IN LIST 1 {
                                CALL FUNCTION OF PAHSE 2
                        }

                        IF CAN ADD COURSE {
```

```
                                        ADD COURSE TO LIST 3
                                } ELSE {
                                        INITIALIZE VARIABLES
                                        BREAK LOOP
                                }

                                IF ALL CLASSES ADDED {
                                        RETURN TO MAIN FUNCTION
                                } ELSE {
                                        INITIALIZE ALL VARIABLES
                                        BREAK LOOP
                                }
                        }
                }
        }

        //phase 2
        FUNCTION {
                LIST 4: LIST OF GROUPS
                LONG X: COUNT OF LECTURE GROUP
                LONG Y: COUNT OF TUTORIAL GROUP
                LONG Z: COUNT OF PRACTICAL GROUP

                IF LONG X, Y, Z IS NOT EQUAL TO 0 {
                        ADD X, Y, Z TO LIST 4
                }

                CALL FUNCTION TO SWAP LIST 4
                CALL FUNCTION TO PERMUTE LIST 4

                LOOP UNTIL LENGTH OF LIST 4 {

                        LOOP UNTIL LENGTH OF ELEMENT IN LIST 4 {

                                CALL FUNCTION OF PHASE 3

                                IF CAN GET SLOT {
                                        CALL REGISTER FUNCTION TO REGISTER COURSE
                                        RETURN TRUE TO PHASE 1
                                } ELSE {
                                        INITIALIZE VARIABLES
                                        RETURN FALSE TO PHASE 1
                                }
                        }
                }

        }

        //phase 3
        FUNCTION {
                LIST 5: LIST OF DATA FROM DATABASE
                LIST 6: TO BE RETURNED TO PHASE 2
                CALL FUNCTION TO SWAP LIST 5
```

```
            LOOP UNTIL LENGTH OF LIST 5 {

                        IF NOT CLASHING WITH QUERY SET &
                        IF NOT CLASHIG WITH SAME SUBJECT &
                        IF NOT CLASHING WITH DIFFERENT SUBJECT {
                                    ADD SUBJECT TO LIST 6
                                    RETURN LIST 6 TO PHASE 2
                        } ELSE {
                                    RETURN EMPTY LIST TO PHASE 2
                        }
            }
}

/ … /

//swapping element
FUNCTION {
            *REFER TO PSEUDO CODE AT RANDOMIZATION, SWAPPING AND PERMUTATION
}

//permutation of element
FUNCTION {
            *REFER TO PSEUDO CODE AT RANDOMIZATION, SWAPPING AND PERMUTATION
}

//slot availability checking
FUNCTION {
            * REFER TO PSEUDO CODE OF SLOT MATCHING TECHNIQUE
}
```

From the pseudo code above, it is necessary to avoid the looping fall in infinite loop when there is a course always clashing with other course and lead to impossible to generate a complete timetable. The approach to break the possible infinite is to set a timer which is all looping must be done within 5 seconds or else break the looping. The pseudo code shows how to break the looping with a timer in Phase 1:

```
/ … /
INT X:  SECOND BETWEEN START TIME AND END TIME
/ … /

LONG Y: GET CURRENT TIME AS START TIME

/ … /

//main function
FUNCTION {
            /… /
            WHILE (X < 5000) {

                        / … /
                        *3 PHASE LOOPING FUNCTIONS
                        / … /
```

```
            LONG Z: GET CURRENT TIME AS END TIME
            INT X = LONG Z – LONG Y
            }
            / … /

}
/ … /
```

After the data generate a timetable, it will pass the JSON String back to the JSP page through AJAX and generate a timetable by using Javascript and DOM (refer pseudo code at Dynamic Timetable Generation).

IN-HAND Timetabling Tool

This module aims to provide both interactive and handy way for students to build and manage their own timetable manually. The word IN-HAND extracts the word of IN-teractive and HAND-y and it means "in your control" which allows the students control over the tool to build their timetable. This module is mostly build in client side by using Javascript and DOM. The only part which involves server side is loading the data from database through AJAX. Figure 3-7-1-F2 illustrates how the module could be run:

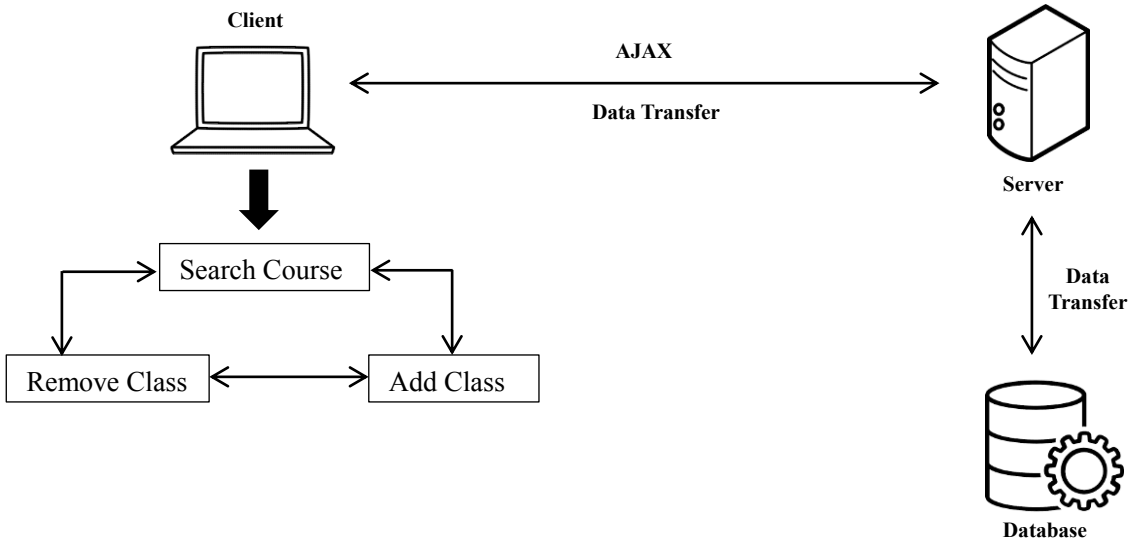


Figure 3-7-1-F3: Basic architecture of IN-HAND Timetabling Tool

Besides, the pseudo code below shows part of module code on how to create an interactive timetable; the HTML should be loaded at first by AJAX:

```
<table>
<thead>
```

```
/ … /
</thead>
<tbody>
	<tr class="rows">
		<td rowspan="2">1</td>
		<td rowspan="2">L</td>
		<td rowspan="2">1</td>
		<td rowspan="2">70</td>
		<td rowspan="2">1-14</td>
		<td>Thu</td>
		<td>03.00pm - 05.00pm</td>
		<td>2.0</td>
		<td>LDK1</td>
		<td rowspan="2">UCCD3184 -  seat joined</td>
	</tr>
	<tr>
		<td>Wed</td>
		<td>02.00pm - 03.00pm</td>
		<td>1.0</td>
		<td>LDK3</td>
	</tr>
	<tr class="rows">
	<td rowspan="3">2</td>
	<td rowspan="3">L</td>
	<td rowspan="3">2</td>
	/ … /
</tbody>
</table>
```

It should take noted that each class should have class “rows” in each <tr> tag, <tr class=“rows”> to indicate the class has been clicked by user when call the function to add classes to timetable. The pseudo code below show how can the student generate a timetable by using Javascript function:

```
//global variables
LIST 1: COLOR PALETTE CONTAINS OF 10 COLORS
LIST 2: COLOR USED
LIST 3: SELECTED COURSE CODE
LIST 5: SLOT USED

FUNCTION {

	VAR X: GET CLICKED ROWS OBJECT WITH CLASS NAME “ROWS”
	VAR CODE: COURSE CODE FROM VAR X
	VAR TYPE: TYPE FROM VAR X
	/ … /

	//get color to each course
	IF LIST 2 IS NOT EMPTY {
		VAR Y: PICK COLOR WHICH USED BY COURSE IN LIST 3
	} ELSE {
		VAR Y: -1
```

```
        }

        LOOP WHEN VAR Y IS -1{
            VAR COLOR: RANDOMLY PICK COLOR FROM LIST 1

            IF COLOR CONTAINS IN LIST 2 {
                VAR Y: -1
            } ELSE {
                ADD COLOR TO LIST 2
                ADD COURSE CODE TO LIST 3
            }
        }

        VAR ROWSPAN: GET ROWSPAN OF VAR X
        IF ROWSPAN IS NULL {
            LIST 4: SLOT REQUIRE TO REGISTER CLASS

            GET NECESSARY DATA FROM VAR X
            ADD SLOT REQUIRE TO LIST 4
            COMPARE LIST 4 AND LIST 5

            IF CLASHING {
                RETURN ERROR MESSAGE
            } ELSE {
                ADD CLASS TO LIST 3 AND LIST 5
            }

        } ELSE {        //if class require more than slot
                        GET NECESSARY DATA FROM VAR X
                        ADD SLOT REQUIRE TO LIST 4
                        COMPARE LIST 4 AND LIST 5

            LOOP UNTIL ROWSPAN {
                IF CLASHING {
                    RETURN ERROR MESSAGE
                } ELSE {
                    ADD CLASS TO LIST 3 AND LIST 5
                }
                 / … /
            }
        }

        / … /

        CALL FUNCTION TO ADD CLASSES TO TIMETABLE

}

//function to add classes to timetable
FUNCTION {
    * REFER TO PSEUDO CODE OF DYNAMIC TIMETABLE GENERATION
}
```

```
//function to check slot availability
FUNTION {
        * REFER TO PSEUDO CODE OF SLOT MATCHING TECHNIQUE
}

//function to remove classes from timetable {
        * REFER TO PSEUDO CODE OF DYNAMIC TIMETABLE GENERATION
}

//function to clear all classes from timetable {
        * REFER TO PSEUDO CODE OF DYNAMIC TIMETABLE GENERATION
}

/ … /
```

The basic module design has been discussed above which the core in this module. The remaining function such as image download, form submission, class hovering and so forth effect would not be discussed details as some function is implementing the external libraries to complete the function.

Hybrid Feature

This module provides the hybrid feature which integrates the module of Automated Timetabling Tool and IN-HAND Timetabling Tool together. It allows the user to direct from one to another module to plan and generate the timetable. Furthermore, this module provides 2 mode which are "Let Me Do!" and "Let System Do!". Figure 3-7-1-F4 shows the architecture of the module and Table 3-7-2-T1describe about the details of the module:

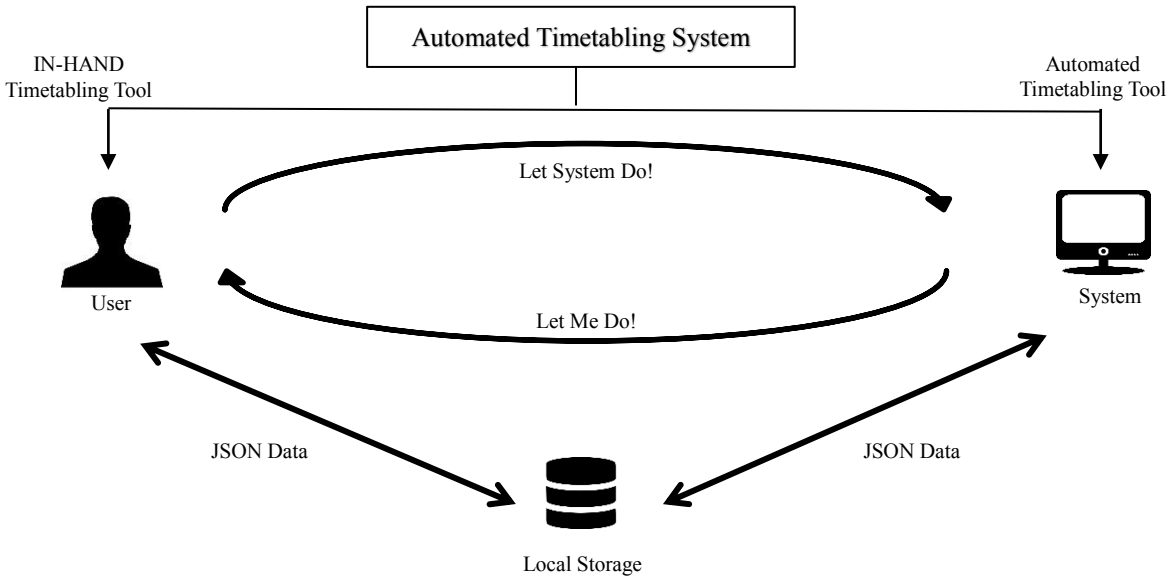


Figure 3-7-1-F4: Architecture of Hybrid Feature

Table 3-7-2-T1: Modes in Hybrid Mode

Mode	Description	Assumption/ Purpose
Let Me Do!	- From Automated Timetabling Tool to IN-HAND Timetabling Tool. - From system to user - The user further modify the timetable which generated by Automated Timetabling Tool.	- User does not satisfy the timetable generated by system. - User want to further customize the timetable. - User can further generate a better timetable than system.
Let System Do!	- From IN-HAND Timetabling Tool to Automated Timetabling Tool. - From user to system - The user let system helps to continue to generate the timetable	- User is difficult to plan the timetable due to time always clashing with other classes. - User want to customize some classes before let the system continue to generate timetable.

In order to share the data among 2 modules, it is necessary to store the data which allows both modules can read the same data to generate the timetable. The data will be stored at local storage in JSON string format and it will allow the modules can read same data easily. The data includes course data and mode indicators. The data will be stored in the storage shows in Table 3-7-2-T2:

Table 3-7-2-T2: Data Store in Local Storage in Hybrid Mode

Attribute Stored	**Description**	**Example Value**
Course Data		
courseCode	Store the course code	UCCD1004/ UCCD2223…etc
description	Store the course description	Programming Concept and Practices…etc
type	Store type of the class	L/ T/ P
group	Store the group of the class	1/ 2 / 3…etc
startTime	Store the start time of the class	12.00PM/ 01.00PM…etc
endTime	Store the end time of the class	12.00PM/ 01.00PM…etc
dayOffer	Store the day offer of the class	Mon/ Tue/ Wed/ Thu/ Fri

venue	Store the venue of the class	N001/ LDK3…etc
week	Store the week of the class	1-14/ 1,2,4,5/ 1-7…etc
courseID	Store the offer ID of the course	IB0515012…etc
cellID	Store the cell ID of the timetable for generating the timetable	*Format: dayOffer_startTime Tue_08.00AM/ Wed_12.00PM…etc
Mode Indicators		
isHybrid	Determine the hybrid is turn on or off	True/ False
letMeDo	Determine "Let Me Do!" is activated/ deactivated and direct user to proper module	True/ False
letSystemDo	Determine "Let System Do!" is activated/ deactivated and direct user to proper module	True/ False

Mode 1 - Let Me Do!

The mode "Let Me Do!" is the mode which allows the user to change from Automated Timetabling Tool to IN-HAND Timetabling Tool. The pseudo code below shows in Automated Timetabling Tool on how the data stored and transfer to the other module:

```
/…/
LIST 1: STORE LIST OF COURSE DATA OBJECTS WHICH READY TO TRANSFER

FUCNTION letMeDo(){

        IF LIST 1 IS NOT EMPTY{

                CLEAR LOCAL STORAGE BEFORE STORE
                SET ITEM TO LOCAL STORAGE: JSON.stringify(LIST 1)
                SET ITEM TO LOCAL STORAGE: isHybrid – TRUE
                REDIRECT TO IN-HAND Timetabling Tool
        } ELSE {
                RETURN ERROR MESSAGE
        }
}
/…/
```

The pseudo code below shows in IN-HAND Timetabling Tool on bow the data is received and integrates to the module:

```
/…/
//global variables
```

```
VAR Y: DETERMINE IF IS HYBRID MODE WITHIN THE MODULE
LIST 2: STORE LIST OF COURSE DATA OBJECTS WHICH READY TO TRANSFER
FUNCTION callLetMeDo {

        IF IS HYBRID MODE {
                LIST 1: JSON OBJECT OF COURSE DATA FROM LOCAL STORAGE
                SET Y TO TRUE
                CALL DATA PASSING FUNCTION
        }
}

/…/
FUCNTION dataPassing {

TRY {
        IF LIST X IS EMPTY {
                RETURN ERROR MESSAGE
        } ELSE {

                LOOP UNTIL LENGTH OF VAR X {
                        SET COLOR DATA
                        SET COURSE DATA OBJECT
                        ADD COURSE DATA OBJECT TO LIST 2
                        SET NECESSARY GLOBAL DATA
                        /…/
                        CALL FUNCTION OF GENERATE TIMETABLE

                        }
                }
        } CATCH {
                CLEAR ALL DATA
                INITIATE GLOBAL DATA
                RETURN ERROR MESSAGE
        }
}
```

Function 2 – Let System Do!

The mode "Let System Do!" is the mode which allows the user changes from IN-HAND Timetabling Tool to Automated Timetabling Tool. The pseudo code below shows in Automated Timetabling Tool on how the data stored and transfer to other module:

```
/…/
LIST 1: STORE LIST OF COURSE DATA OBJECTS WHICH READY TO TRANSFER

FUNCTION letSystemDo(){

        IF LIST 1 IS NOT EMPTY{

                CLEAR LOCAL STORAGE BEFORE STORE
                SET ITEM TO LOCAL STORAGE: JSON.stringify(LIST 1)
```

```
                SET ITEM TO LOCAL STORAGE: isHybrid – TRUE
                REDIRECT TO IN-HAND Timetabling Tool
        } ELSE {
                RETURN ERROR MESSAGE
        }
}
/…/
```

The pseudo code below shows in Automated Timetabling Tool on bow the data be received and integrate to the module:

```
/…/

//global variables
VAR Y: DETERMINE IF IS HYBRID MODE WITHIN THE MODULE
LIST 2: STORE LIST OF COURSE DATA OBJECTS WHICH READY TO TRANSFER
LIST 3: STORE COURSE ID TO GENERATE TIMETABLE BY SYSTEM

FUNCTION callLetSystemDo {

        IF IS HYBRID MODE {
                LIST 1: JSON OBJECT OF COURSE DATA FROM LOCAL STORAGE
                SET Y TO TRUE
                CALL DATA PASSING FUNCTION

                CALL GENERATE TIMETABLE FUNCTION

                LOOP UNTIL LENGTH OF LIST 1 {
                        STORE COURSE ID TO LIST 3 WITHOUT REPEATING DATA
                }

                LOOP {
                        CHECK THE CHECK BOX ACCORDING TO LIST 1
                        DISABLED CHECK BOX ACCORDING LIST 1
                }

                /…/
        }

        /…/
}
```

The following pseudo code below in back-end server shows how the system will be continue to help the user to further generate the timetable without clashing the classes the user had selected previously, the code is merging in to the module of Automated Timetabling Tool:

```
/…/
//global variable
LIST 1: LIST OF COURSE ID SELECTED BY USER PREVIOUSLY
```

```
LIST 4: SLOT REQUIRED
LIST 5: QUERY SET BY THE USER
LIST 6: CLASS TO BE ADD IN HYBRID MODE
FUNCTION {
        /…/
        BOOLEAN X: DETERMINE IF IS HYBRID MODE

        IF LIST 1 IS NOT EMPTY {
                LIST 2: GET JSON STRING FROM REQUEST OBJECT
                PARSE LIST 2 IN OBJECT ARRAY TO LIST 1
                BOOLEAN X: CALL HYBRID MODE FUNCTION AND RETURN VALUE
        }

        /…/

        IF BOOLEAN X IS TRUE {
                CONTINUE TO LOOP REST OF THE CLASS
                CALL 3-PHASES LOOPING FUNCTION
                /…./
        } ELSE {
                RETURN ERROR MESSAGE
        }

}

/…/
FUNCTION OF HYBRID MODE {
        LIST 3: STORE LIST OF COURSES

        LOOP UNTIL LENGTH OF LIST 1 {
                GET COURSES ACCORDING TO LIST 1 FROM DATABASE
                ADD COURSES TO LIST 3
                /…/
                SET SLOT REQUIRED BY LIST 3 TO LIST 4

                IF THERE IS QUERY SET {
                        CHECK IF LIST 4 CLASHING WITH LIST 5
                }

                IF NOT CLASHING BETWEEN LIST 4 AND LIST 5 {
                        ADD OBJECT OF LIST 1 TO LIST 6
                } ELSE {
                        CLEAR LIST 6
                        BREAK LOOP
                }

        }

        IF LIST 6 IS NOT EMPTY {

                LOOP UNTIL LENGTH LIST 6 {
                        CALL REGISTER SLOT FUNCTION
                }
```

```
                RETURN TRUE TO MAIN FUNCTION
        } ELSE {
                RETURN FLASE TO MAIN FUNCTION
        }

}
```

The pseudo codes above have roughly discussed on how the 2 module basically work. One should be concerned is when the user access to hybrid mode when there is nothing in local storage, this means that currently the hybrid mode has yet to be turned on. Thus, it should direct the user to proper module and prompt the appropriate instruction to instruct the user what to do. The following pseudo code shows in the module selection interface, the function will check the local storage and direct the user to proper module with proper instruction:

```
FUNCTION {
        //MODE SELECTION
        //1 - automatedTimetabling
        //2 - IN-HAND Timetabling
        //3 - Let Me Do!
        //4 - Let System Do!
        IF MODE IS 1 {

                SET isHybrid TO FALSE IN LOCAL STORAGE
                DIRECT TO AUTOMATED TIMETABLING TOOL

        } ELSE IF MODE IS 2 {

                SET isHybrid TO FALSE IN LOCAL STORAGE
                DIRECT TO IN-HAND TIMETABLING TOOL

        } ELSE IF MODE IS 3 {

          IF COURSES IN LOCAL STORAGE IS NOT EMPTY {
                SET isHybrid TO TRUE IN LOCAL STORAGE
                DIRECT TO IN-HAND TIMETABLING TOOL
          } ELSE {
                SET isHybrid TO FALSE IN LOCAL STORAGE
                SET letMeDo TO TRUE IN LOCAL STORAGE
                DIRECT TO AUTOMATED TIMETABLING TOOL
                 }

        } ELSE IF MODE IS 4{

          IF COURSES IN LOCAL STORAGE IS NOT EMPTY {
                SET isHybrid TO TRUE IN LOCAL STORAGE
                DIRECT TO AUTOMATED TIMETABLING TOOL
          } ELSE {
                SET isHybrid TO FALSE IN LOCAL STORAGE
                SET letSystemDo TO TRUE IN LOCAL STORAGE
                DIRECT TO IN-HAND TIMETABLING TOOL
```

```
            }
        }
}
```

Every time the module is first loaded, it will load to callLetMeDo() function in IN-HAND Timetabling Tool and callSystemDo() function in Automated Timetabling Tool. The purpose is to check whether the user access to Hybrid Mode with empty timetable, the pseudo code is as followed:

```
FUNCTION callLetMeDo/ callLetSystemDo {

        IF IS HYBRID {
                /../
        } ELSE {
                CHECK MODE FROM LOCAL STORAGE
                IF letMeDo{
                        PROMPT MESSAGE AT THE HEADING WITH BLINK TEXT
                "Let Me Do: You should generate timetable first!"
                } ELSE {
                        PROMPT MESSAGE AT THE HEADING WITH BLINK TEXT
                "Let System Do: You should generate timetable first!"
                }
                /…/
        }
}
/…/
```

In summary, Figure 3-7-1-F5 shows the paths direct to each module of the whole system in hybrid mode according to the mode given:

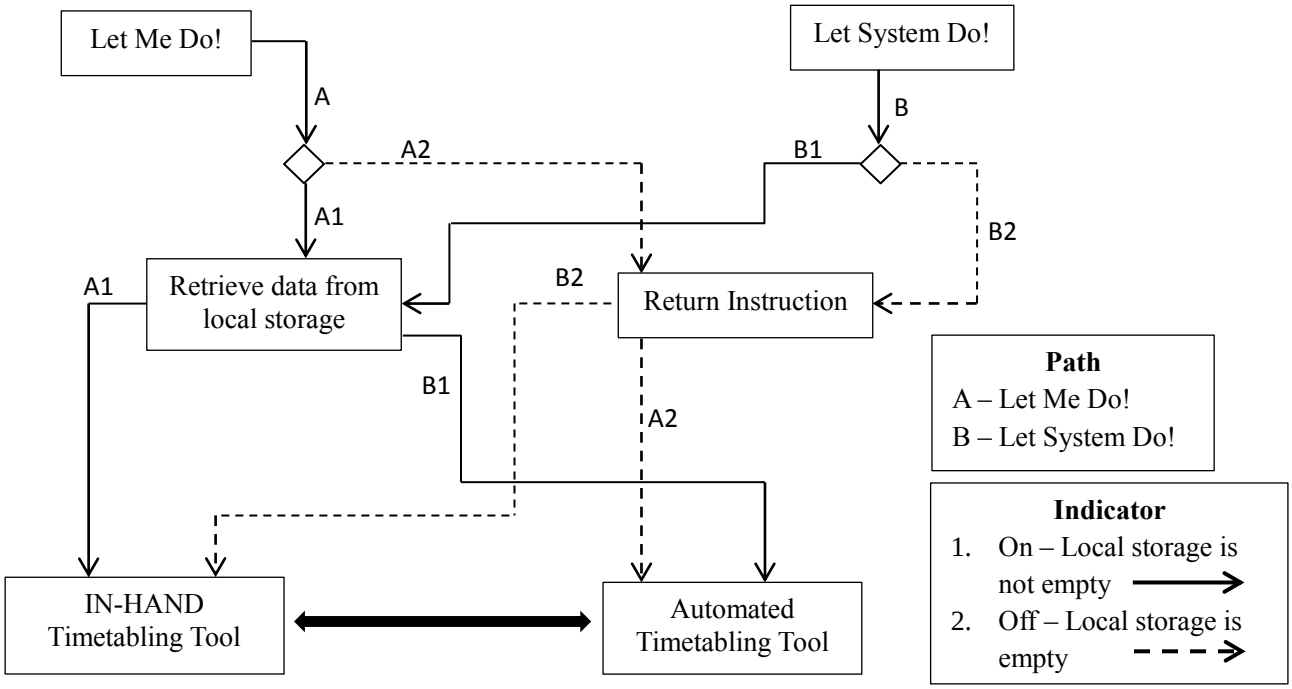


Figure 3-7-1-F5: Path of System in Hybrid Mode

Chapter 4: Method/ Technologies Involved

4-1 Design Specification

The methodology implements for this project would be waterfall methodology which the process sequentially moves from a step to another. The processes that involved to this project would be planning, analyzing and evaluating, designing, coding, implementing, testing and maintaining. It is an iterative process which the step is possible to backward whenever the problems do not solve or there are new requirements or problems appeal to the new system (Tutor, 2011).

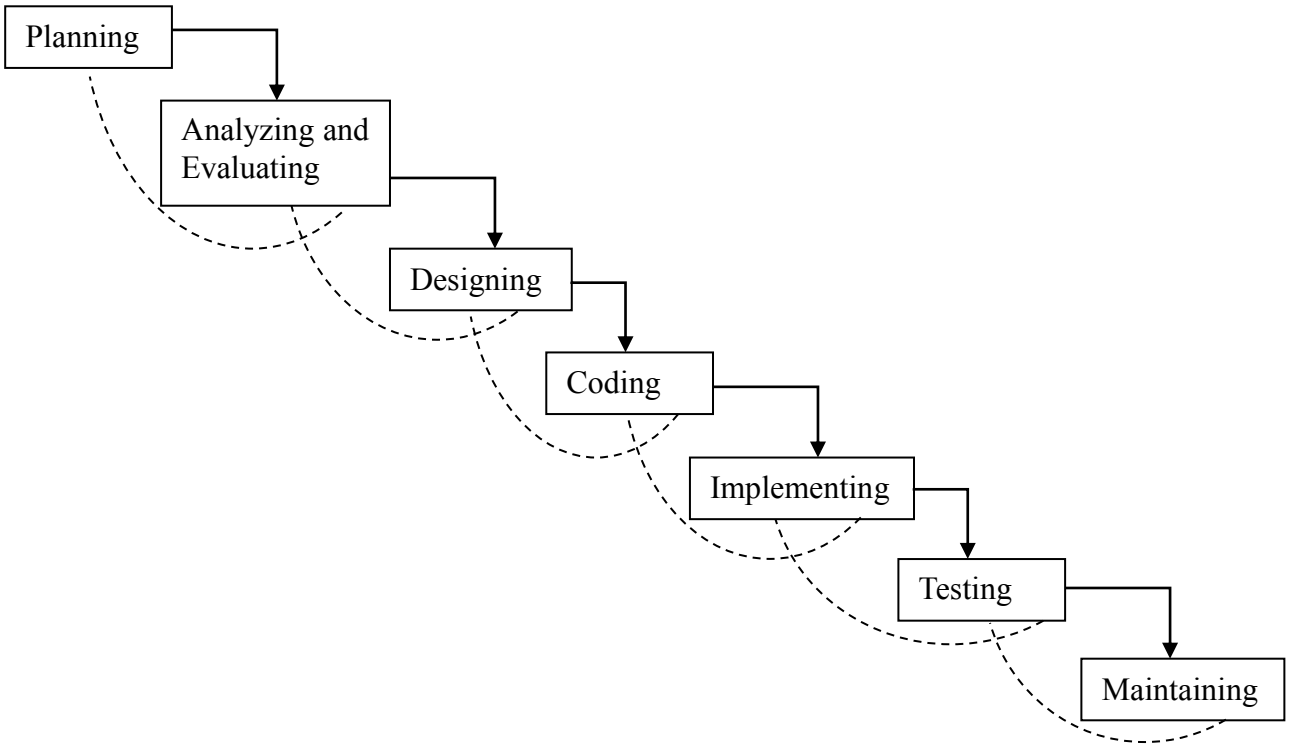


Figure 4-1-F1: Process of Developing Automated Timetabling System

Planning

In the planning phase, the project would be initiated by determine the problems of the topic and generate the rough idea of proposed system which to overcome the problems. After that, the proposal for the proposed system will start to develop to determine the problem statements and main objectives. Besides, several benchmark system and the similar project will be reviewed to understand the similar project or research done by others about the proposed system.

Analyzing and Evaluating

In analyzing and evaluating phase, the progress will start on analyzing the system requirements and specifications need to be developed. 50 of the survey questionnaire will be distributed to collect the data and requirements from students in UTAR. Besides, the event flow and the use case scenario of the system would be represented using Unified Modeling Language (UML). Furthermore, the algorithms relevant to the proposed system would be evaluated through some websites to understand how to apply those algorithms for improving the proposed system.

Designing

After the analysis and evaluation, the work will start on designing the Graphical User Interface (GUI) by creating form and images. Furthermore, the development of a database on how to manage the data should be included in this phase. The sample data would be collected from each faculty in UTAR.

Coding

After designing, the work will start on coding the web application and apply the algorithms which review in analyzing and evaluating phase. The project will be compiled by using NetBeans IDE and the Glassfish will be the server to host the web application. The technologies involve in the project would be Java technologies, J2EE as the server side programming and web programming such as Hypertext Markup Language 5 (HTML5), Cascading Style Sheet (CSS), JQuery and JavaScript as the client-side programming. Furthermore, the database used by this system would be MySQL database which has the huge capacity enough to handle a large number of transaction and data. The system will adapt MVC model which the Model (M), View (V), and Controller (C) responsible for each task to achieve load balancing and optimize the functionality of the system.

Implementing, Testing and Maintaining

Once the system complete to code, it will be implemented the system and test the system. A series of testing will be conducted to verify the reliable of the system such as beta testing. During the testing, a group of targeted student will be arranged to test the system performance and Table 3-1-T1 shows the system performances definition of the project that should be achieved. The last phase would be maintenance which to refine the system in order to satisfy and add-value to the end user.

Table 4-1-T1: System Performances Definition

System Performances	Definition
Accuracy	The system should produce the schedule with zero error and no time clashing with other subjects
Speed	The system should respond to user less than 10 seconds automatically
Readability	The system should display information that is easy for read and without tiring the eyes of users
Ease of use	The system should have well design user interface to minimize the user effort to navigate the system
Effectiveness	The system should produce the schedule that the users are required
Efficiency	The system should produce the schedule with minimal user effort required

4-2 Implementation Issues and Challenges

The most challenging part of implementing this system is the design of the database since the data is required to support for automatic generation of timetable. It is required a good design of the database in order to retrieve the necessary data for programming the logic. Besides, the design of user interface should be the challenge as the more function of the system, the more complex of the system design. It requires to well designing of navigation panel to enhance user experience instead making them suffer from using the system.

It believes that the more challenging to implement the system, it could provide the effective and efficient functionality for the user. The system provides a hybrid function in which the user not only can generate the timetable automatically but also can further update the timetable to meet the user requirement. However, it should require the complex logic to implement this function.

4-3 Timeline of Implementation

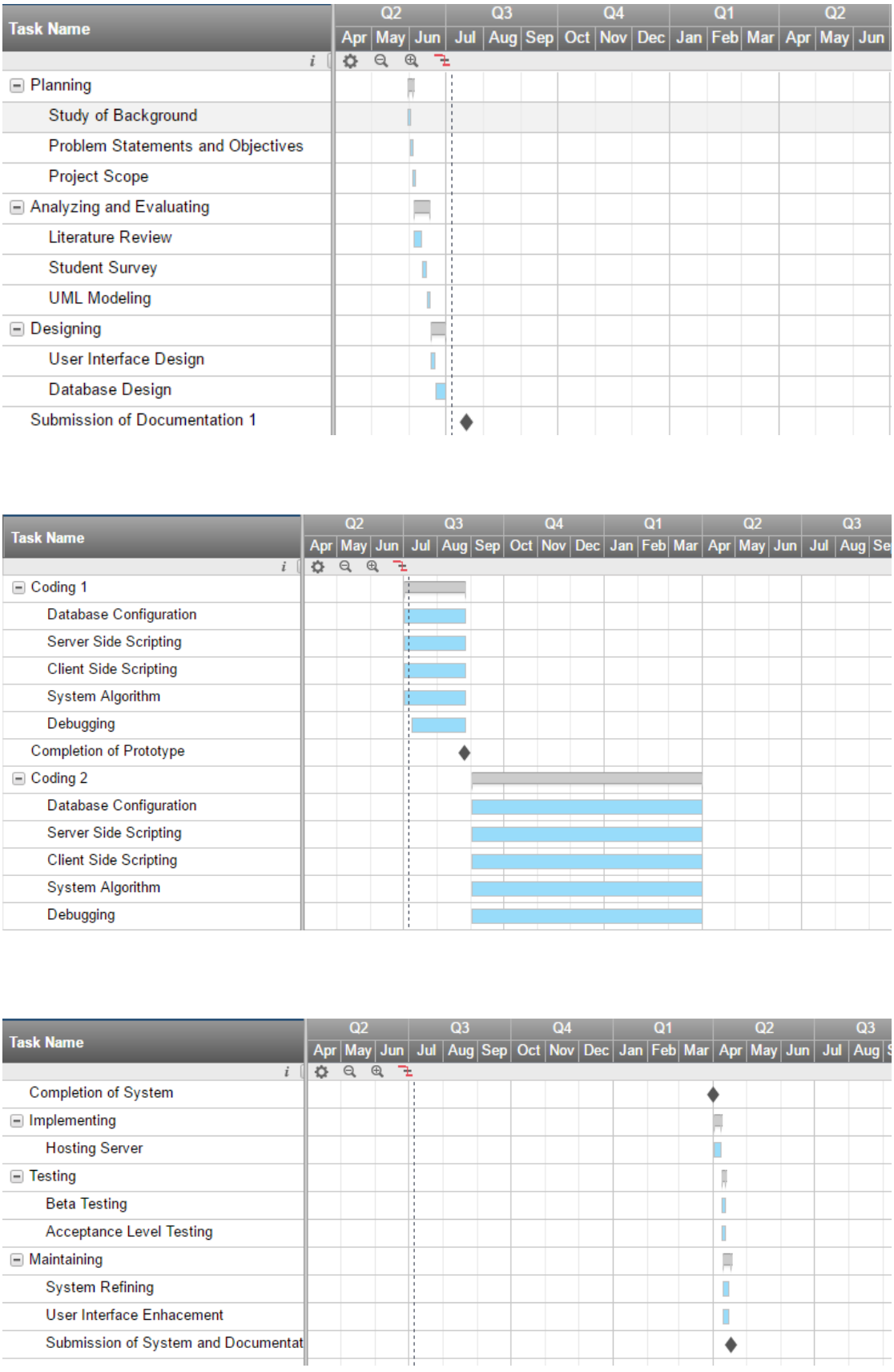


Figure 4-3-F1: Gantt Chart of Implementing the System

Chapter 5: System Testing and Screen Shot

This section will discuss on designing the test plan for the proposed system. There are several test plans will be conducted to verify and evaluate system functionality, performance, and user acceptance level. The test cases also will be prepared for each test plan. The following table shows the details of the test plan which will be conducted:

Table 5-T1: Proposed Test Plan

Test Plan	Description	Performed by
GUI Testing	Testing of buttons, check box, drop down list, text fields, icon and link.	System Developer
System Performance Testing	Testing of output accuracy and quality as well as time to generate timetable to ensure zero error on result produced.	System Developer
Functional Testing	Testing of system function on input verification, error and exception handling as well as output display.	System Developer

5-1 GUI Testing

During GUI Testing, the elements on the webpage such as button, check box, drop down list, text fields will be the target to test. The following figures will shows the user interface of each page and the list of target to be tested.

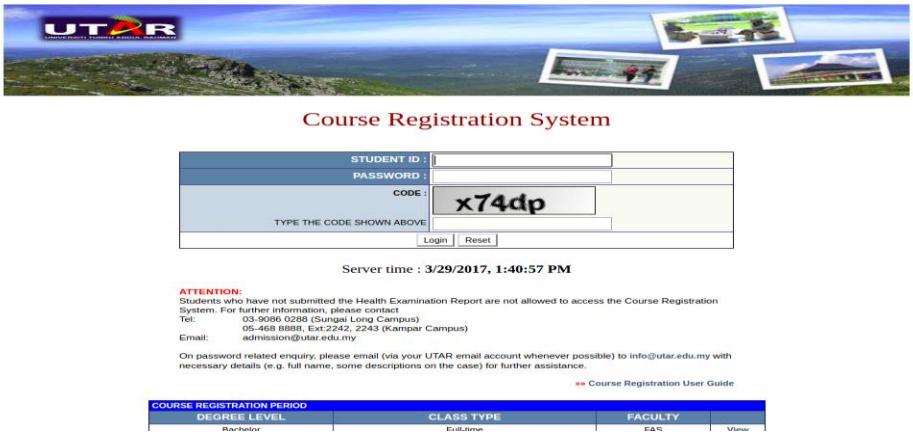


Figure 5-1-F1: Login Interface

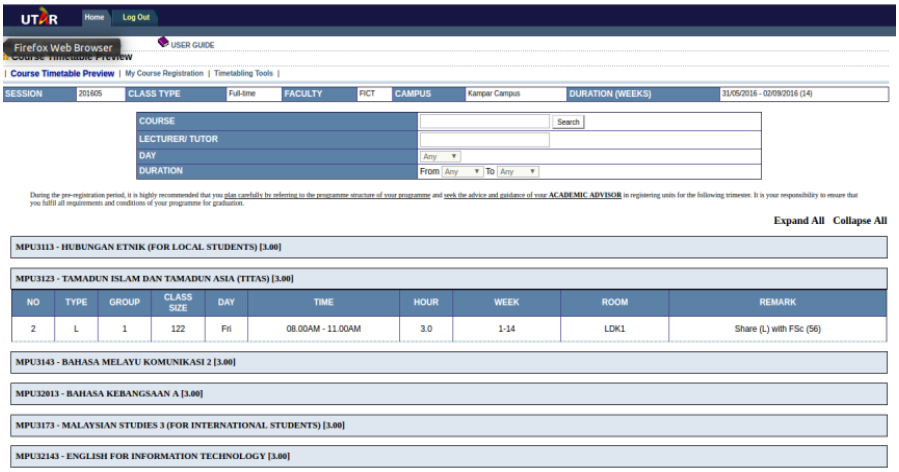


Figure 5-1-F2: Course Preview Interface

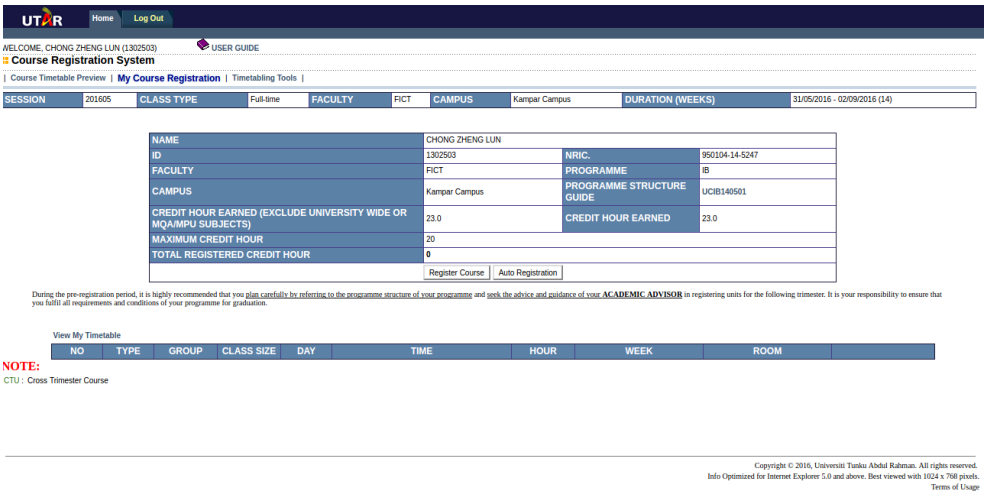


Figure 5-1-F3: Course Registration Interface

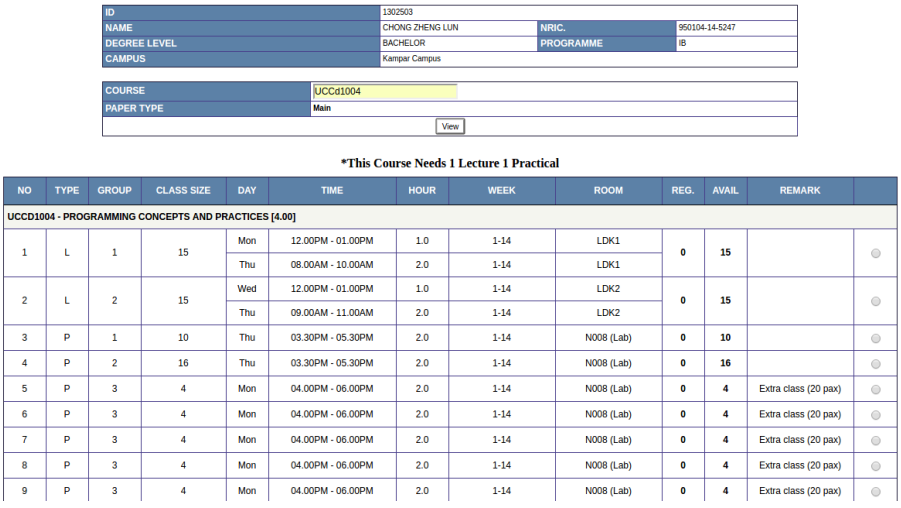


Figure 5-1-F4: Course Search and Selection Interface

Figure 5-1-F5: Course Update Interface

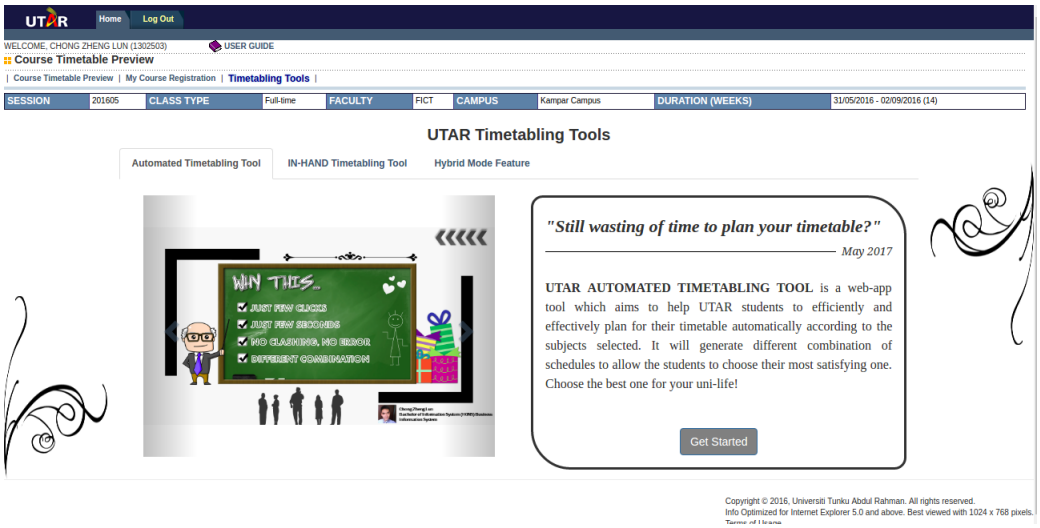


Figure 5-1-F6: UTAR Timetabling Tools Interface

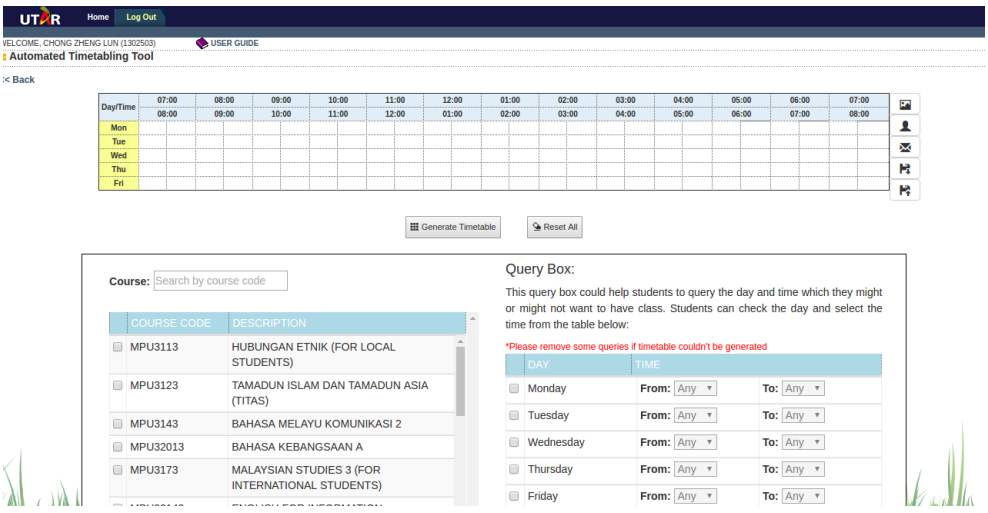


Figure 5-1-F7: Automated Timetabling Tool Interface

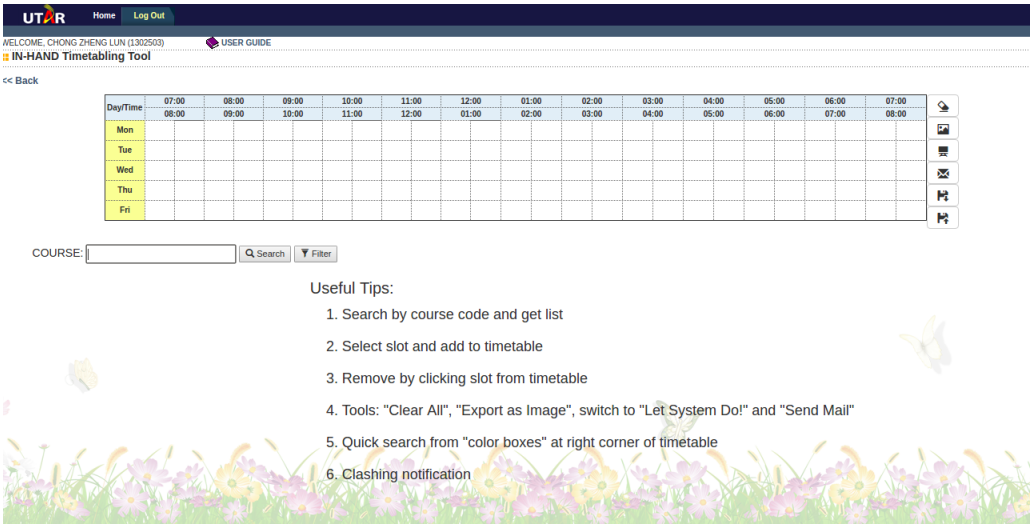


Figure 5-1-F8: IN-HAND Timetabling Tool Interface

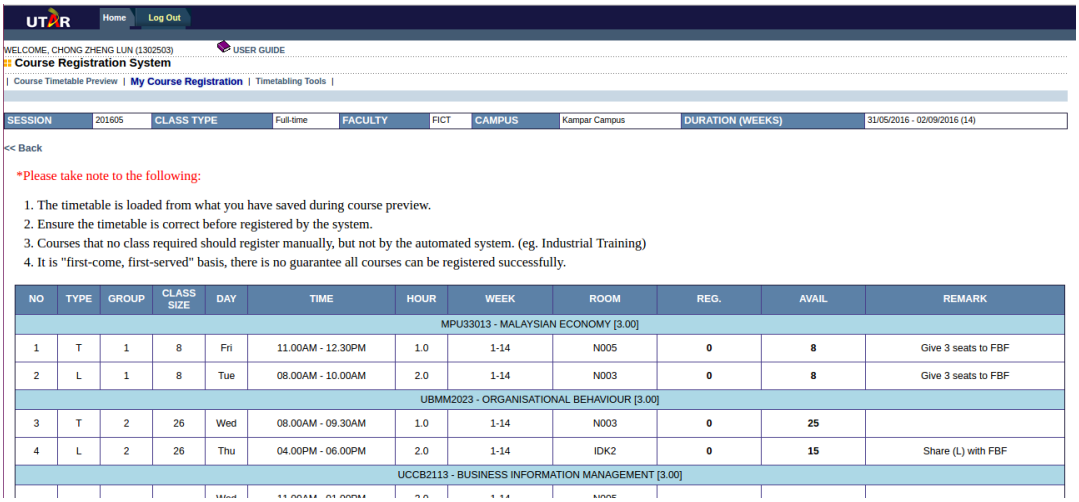


Figure 5-1-F9: Auto-registration Interface

The table below shows the target to be tested in each user interface and the desirable action to be performed:

Table 5-1-T1: List of Target to be Tested in GUI Testing

User Interface	Object	Target to be Tested	Desirable Action	Workable?
Login Interface	Text Field	User ID	Enter alphabet, numeric character	Yes
	Text Field	Password	Enter masking value	Yes
	Text Field	Code	Enter alphabet, numeric character	Yes

	Button	Login	Verify user login process	Yes
	Button	Reset	Clear all text field	Yes
Course Preview Interface	Link	Logout	Verify user logout process	Yes
	Link	My Course Registration	Direct user to course registration interface	Yes
	Link	Timetabling Tool	Direct user to timetabling tool interface	Yes
	Text Field	Course	Enter alphabet, numeric character	Yes
	Text Field	Lecture/ Tutor	Enter alphabet, numeric character	Yes
	Drop Down List	Day	Display day in drop down list	Yes
	Drop Down List	From Time	Display start time in drop down list	Yes
	Drop Down List	To Time	Display end time in drop down list	Yes
	Link	Expand All	Display all collapse slot	Yes
	Link	Collapse All	Hide all display slot	Yes
	Table header	-	Expand/ Collapse particular slot	Yes
	Button	Search	Direct user to search page	Yes
Course Registration Interface	Link	Logout	Verify user logout process	Yes
	Link	Course Timetable Preview	Direct user to course preview interface	Yes
	Link	Timetabling Tool	Direct user to timetabling	Yes

			tool interface	
	Button	Register Course	Direct user to course selection interface	Yes
	Button	Auto Registration	Direct user to auto registration interface	Yes
	Check Box	Delete	Select course to be removed	Yes
	Icon	-	Direct user to course update interface	Yes
	Button	Confirm Delete	Confirm user delete action and process	Yes
Auto Registration Interface	Link	Logout	Verify user logout process	Yes
	Link	Course Timetable Preview	Direct user to course preview interface	Yes
	Link	Timetabling Tool	Direct user to timetabling tool interface	Yes
	Link	Back	Direct user to previous page	Yes
	Button	Confirm	Auto register course by system	Yes
	Button	Update	Direct user to IN-HAND Timetabling Tool	Yes
	Link	Click here (no record found)	Direct user to Timetabling Tool Interface	Yes
Course Search and Selection Interface	Link	Logout	Verify user logout process	Yes
	Link	Course Timetable Preview	Direct user to course preview interface	Yes
	Link	Timetabling	Direct user to	Yes

		Tool	timetabling tool interface	
	Link	Back	Direct user to previous page	Yes
	Text Field	Course	Enter alphabet, numeric character	Yes
	Button	View	Display search course	Yes
	Radio Button	-	Select slot to be register	Yes
	Button	Register	Confirm registration process	Yes
Course Update Interface	Link	Logout	Verify user logout process	Yes
	Link	Course Timetable Preview	Direct user to course preview interface	Yes
	Link	Timetabling Tool	Direct user to timetabling tool interface	Yes
	Link	Back	Direct user to previous page	Yes
	Radio Button	-	Select slot to be register	Yes
	Button	Submit	Confirm update process	Yes
UTAR Timetabling Tools Interface	Link	Logout	Verify user logout process	Yes
	Link	My Course Registration	Direct user to course registration interface	Yes
	Link	Course Timetable Preview	Direct user to course preview interface	Yes
	Link	Automated Timetabling Tool	Slide to module description	Yes
	Button	Get Started	Direct user to Automated Timetabling	Yes

			Tool Interface	
	Link	IN-HAND Timetabling Tool	Slide to module description	Yes
	Button	Get Started	Direct user to IN-HAND Timetabling Tool Interface	Yes
	Link	Hybrid Mode	Slide to module description	Yes
	Link	Let Me Do!	Direct user to particular module	Yes
	Link	Let System Do!	Direct user to particular module	Yes
IN-HAND Timetabling Tool	Link	Logout	Verify user logout process	Yes
	Link	Back	Direct user to previous page	Yes
	Button	Clear All	Clear all content in timetable	Yes
	Button	Export Image	Download image of timetable	Yes
	Button	Let System Do!	Direct user to particular module	Yes
	Button	Send Mail	Send email notification to user mail box	Yes
	Button	Load	Load user generated timetable	Yes
	Button	Save	Save user generated timetable	Yes
	Text Field	Course	Enter alphabet, numeric character	Yes
	Button	Search	Display search	Yes

			course	
	Button	Filter	Display container of filter search	Yes
	Text Field	Course (filter)	Enter alphabet, numeric character	Yes
	Text Field	Lecturer (filter)	Enter alphabet, numeric character	Yes
	Drop Down List	Day (filter)	Display day in drop down list	Yes
	Drop Down List	From Time (filter)	Display start time in drop down list	Yes
	Drop Down List	To Time (filter)	Display end time in drop down list	Yes
	Button	Submit (filter)	Display search course	Yes
	Button	Clear (filter)	Reset all text fields and drop down lists	Yes
	Icon	x	Close container of filter	Yes
Automated Timetabling Tool	Link	Logout	Verify user logout process	Yes
	Link	Back	Direct user to previous page	Yes
	Button	Export Image	Download image of timetable	Yes
	Button	Let Me Do!	Direct user to particular module	Yes
	Button	Send Mail	Send email notification to user mail box	Yes
	Button	Load	Load user generated timetable	Yes
	Button	Save	Save user generated	Yes

		timetable	
Button	Generate Timetable	Verify user generate timetable process	Yes
Button	Reset All	Clear all content in timetable and query box	Yes
Text Field	Course	Display most related course	Yes
Check Box	-	Select course to be register	Yes
Check Box	(Query Box)	Select day to be query	Yes
Drop Down List	From Time (Query Box)	Display start time in drop down list	Yes
Drop Down List	To Time (Query Box)	Display end time in drop down list	Yes

All in all, all elements and objects in each user interface work well to perform the desire action when user clicks on it. There is no error or any elements or objects cannot function well found.

5-2 System Performance Testing

During system performance testing, series of input will be tested to evaluate the correct expected output is produced and the time required for each action to be done. There will also have some assumption made for certain situation during testing; the most important to assume that the network connection is stable all the time when performing the test. The test only emphasizes on the time and result produced for particular modules especially the “automated” timetabling and registration module to evaluate the system performs in acceptance level. Appendix C shows the records of series of the tests have been conducted for each module and Table 5-2-T1 shows the summary result tested by different inputs and the expected output is produced.

Table 5-2-T1: Summary of System Performance Testing

Test No	Test Name	Test Pass?	Remark
1	Page Loading Time on Course Preview	Pass	26 course to load
2	Page Loading Time on Course Preview	Pass	26 course to load
3	Page Loading Time on Course Preview	Pass	1 course to load
4	Page Loading Time on Timetabling Tool Interface	Pass	
5	Auto-generation of timetable – No query, No clashing	Pass	
6	Auto-generation of timetable – query, no clashing	Pass	
7	Auto-generation of timetable – query, partially clashing	Pass	
8	Auto-generation of timetable – query, partially clashing	Pass	
9	Auto-generation of timetable – more courses	Pass	
10	Auto-generation of timetable – more courses, partially clashing	Pass	
11	Auto-generation of timetable – more courses, clashing	Fail	Bug – long looping time
12	Auto-generation of timetable – more courses, clashing	Pass	Bug fixed for Test 11
13	Auto-generation of timetable – more courses, query, clashing	Pass	
14	Save Timetable	Pass	
15	Load Timetable	Pass	
16	Auto-generation – Load timetable	Pass	
17	Auto-registration of Timetable – fail to register	Pass	
18	Auto-registration of Timetable – successful to register	Pass	

19	Auto-registration of Timetable – fail to register	Pass
20	Page loading Time of course registration interface	Pass

According to the test performed, most of the tests meet the system performance benchmark stated in Table 4-1-T1 which the system should respond to the user less than 10 second. Besides, the output result has no error and the schedule produced is accurate as well. In Test 11, there is bug found where the looping time is too long when the user select the course which always clashing to each other. The bug is fixed and the test is performed again in Test 12. In short, all performance tests finally are passed.

5-3 Functional Testing

5-3-1 Error Handling and Respond Message

During functional testing, it will test the function of the proposed system which to ensure the system verifies input, produces desirable output and handles unexpected error. The following figure shows how the system responds to error.

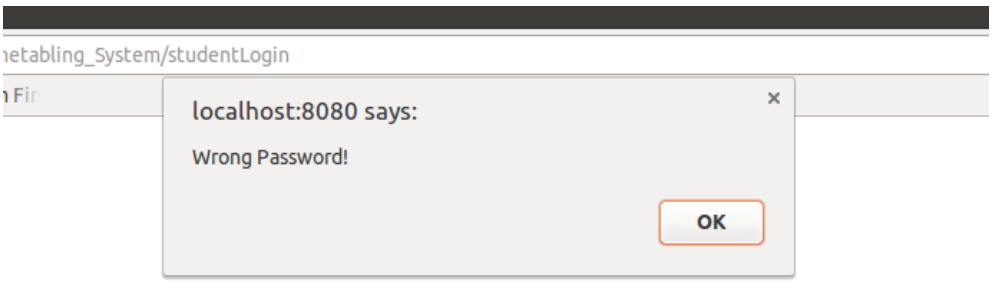


Figure 5-3-1-F1: User password error

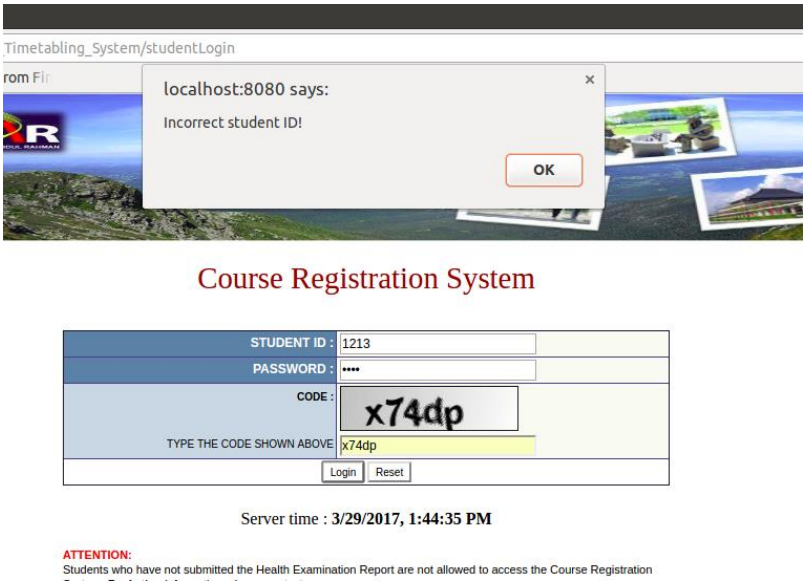


Figure 5-3-1-F2: User ID error

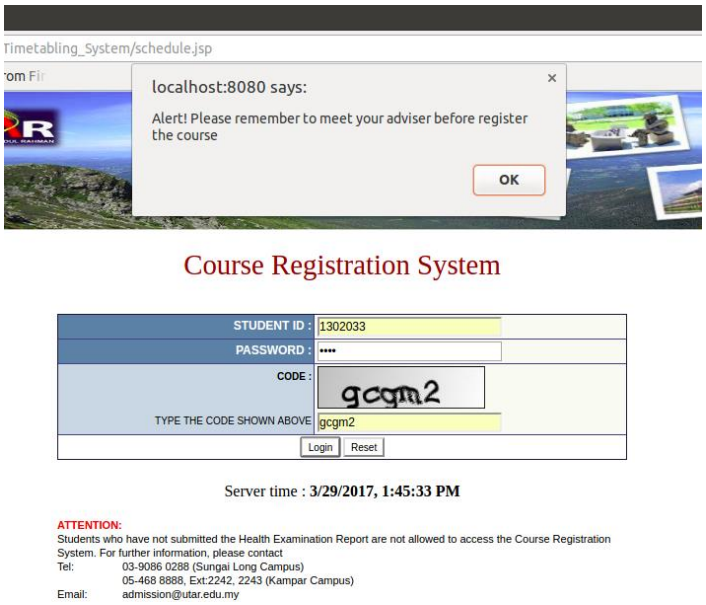


Figure 5-3-1-F3: Advisory status notification

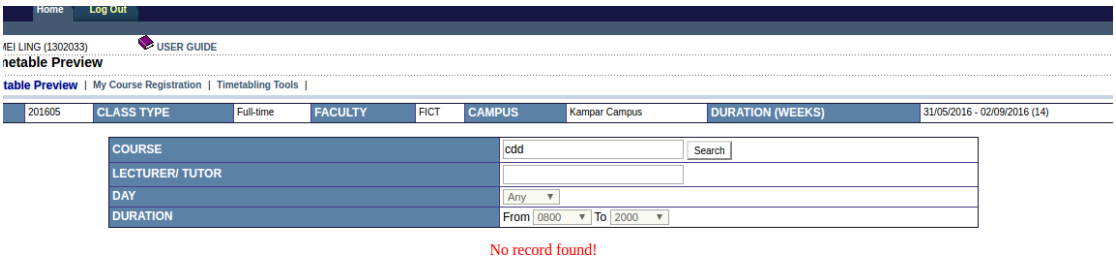


Figure 5-3-1-F4: No search record found during course preview

Figure 5-3-1-F5: No Search Record Found during course registration

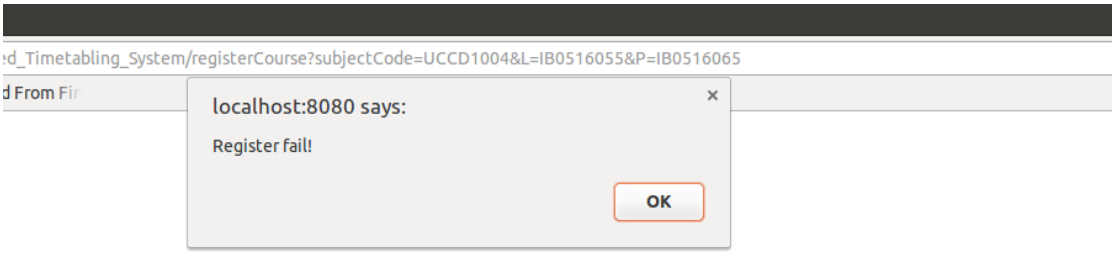


Figure 5-3-1-F6: Course registration fails

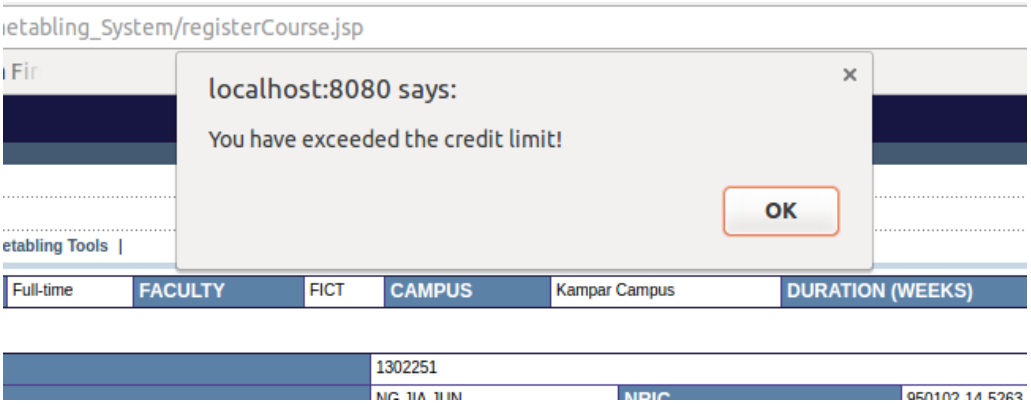


Figure 5-3-1-F7: Exceed credit hour error message

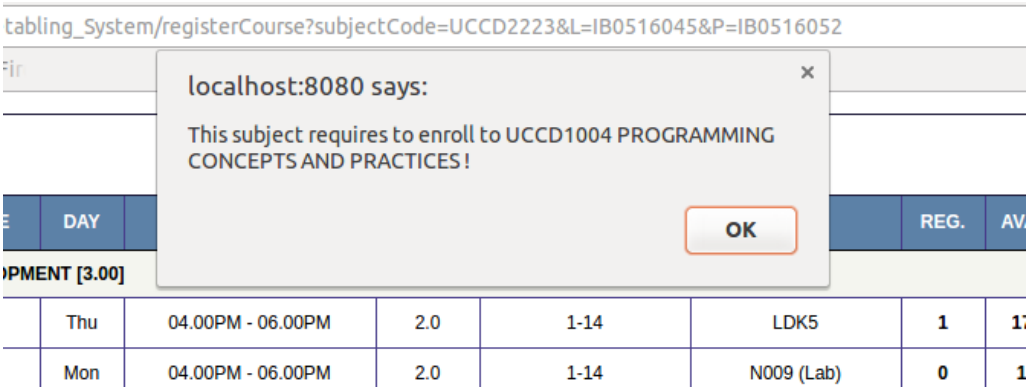


Figure 5-3-1-F8: Pre-requisite error handling

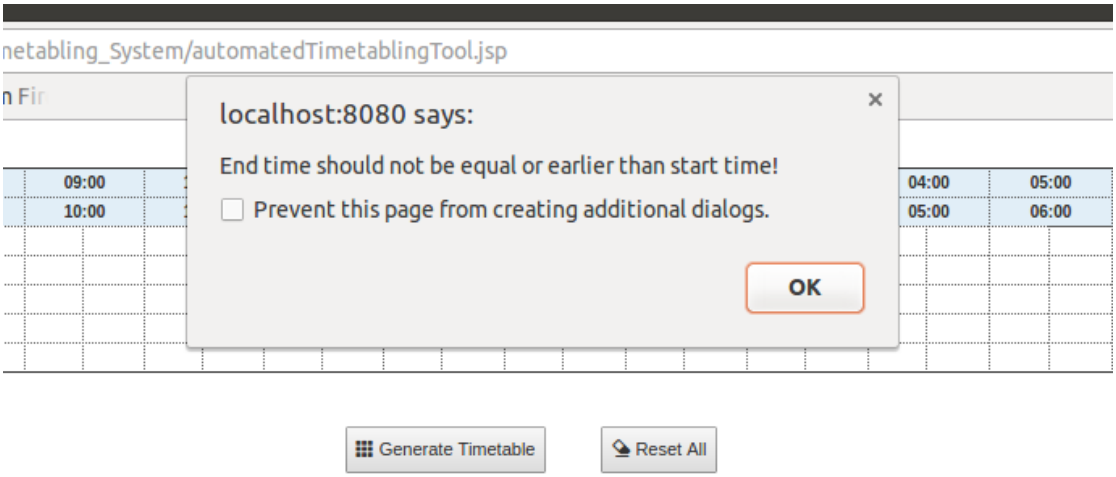


Figure 5-3-1-F9: Empty course selection before generating timetable

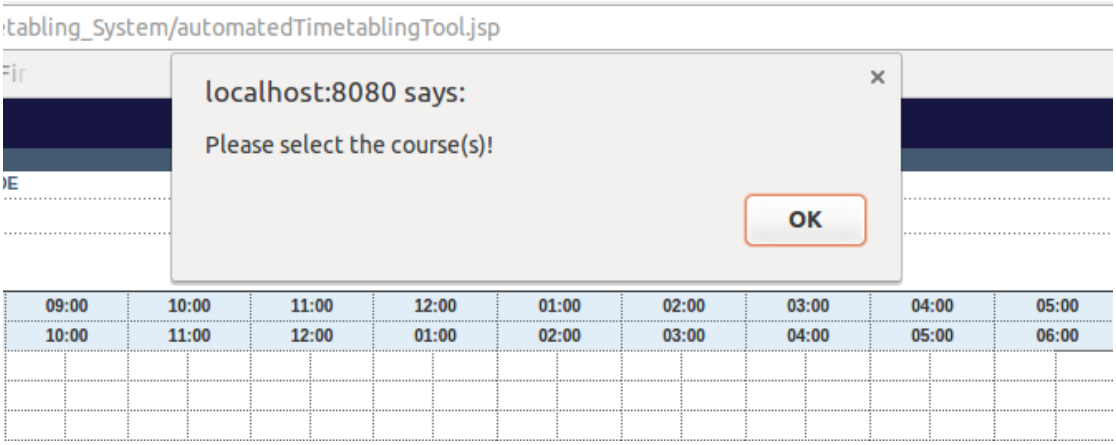


Figure 5-3-1-F10: Start time later than end time error

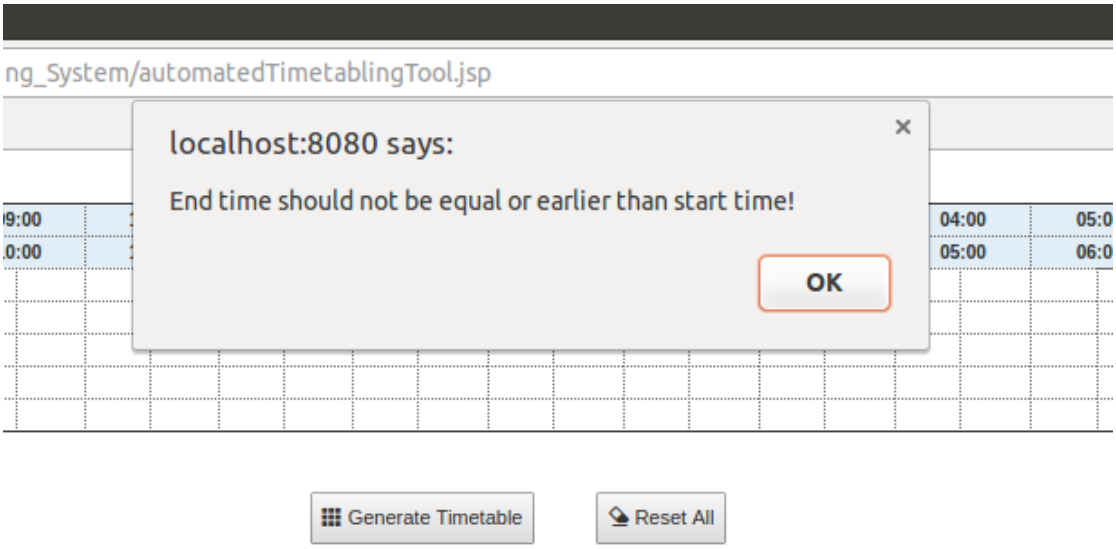


Figure 5-3-1-F11: End time earlier than start time error

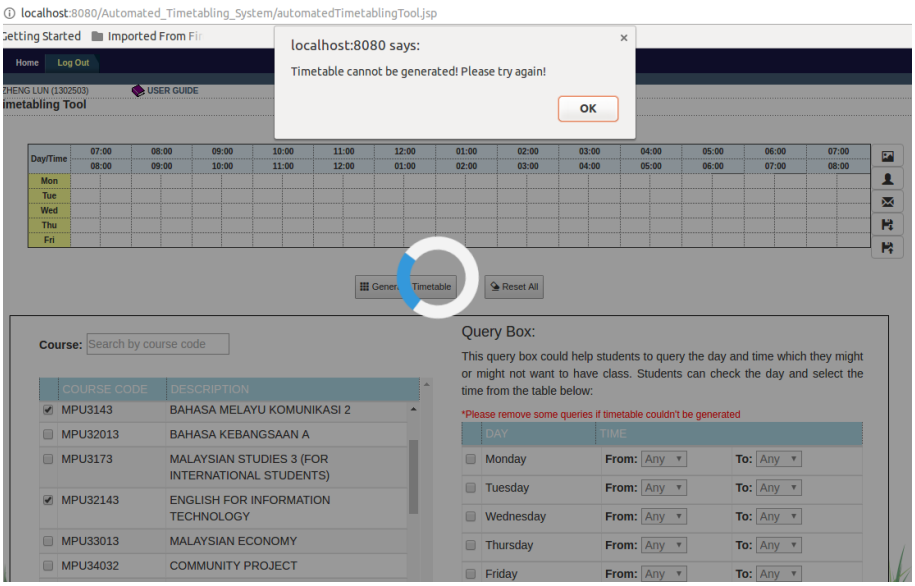


Figure 5-3-1-F12: Timetable generation fails

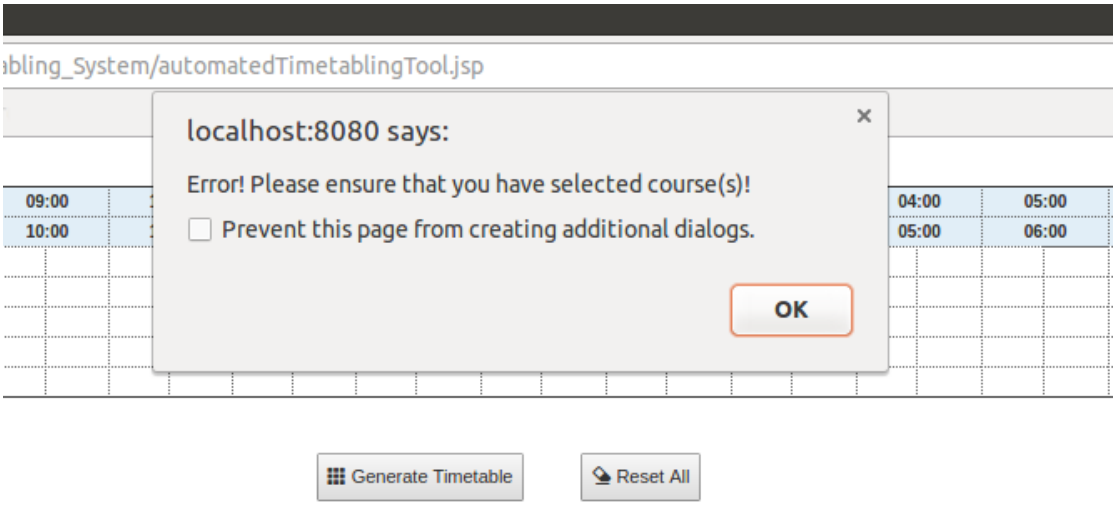


Figure 5-3-1-F13: Let Me/ System Do proceed error

Figure 5-3-1-F14: No search record found in IN-HAND Timetabling Tool

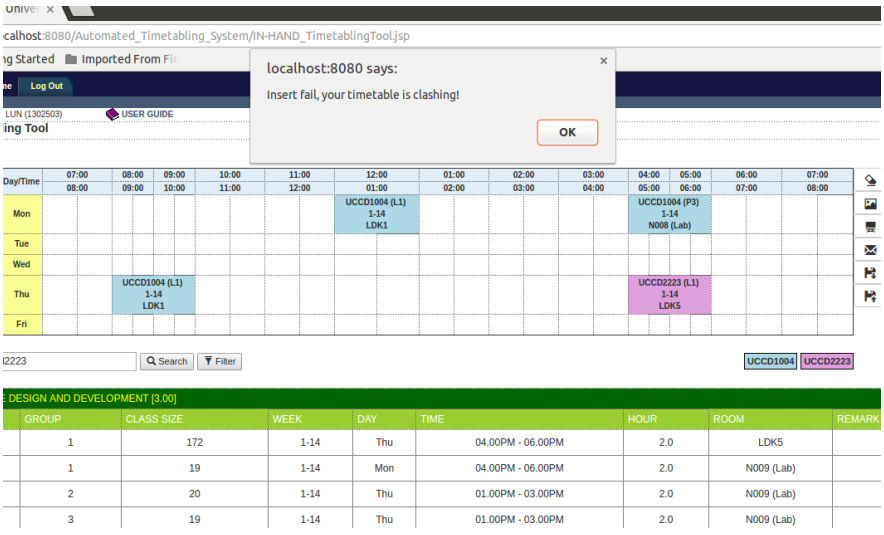


Figure 5-3-1-F15: Slot clashing

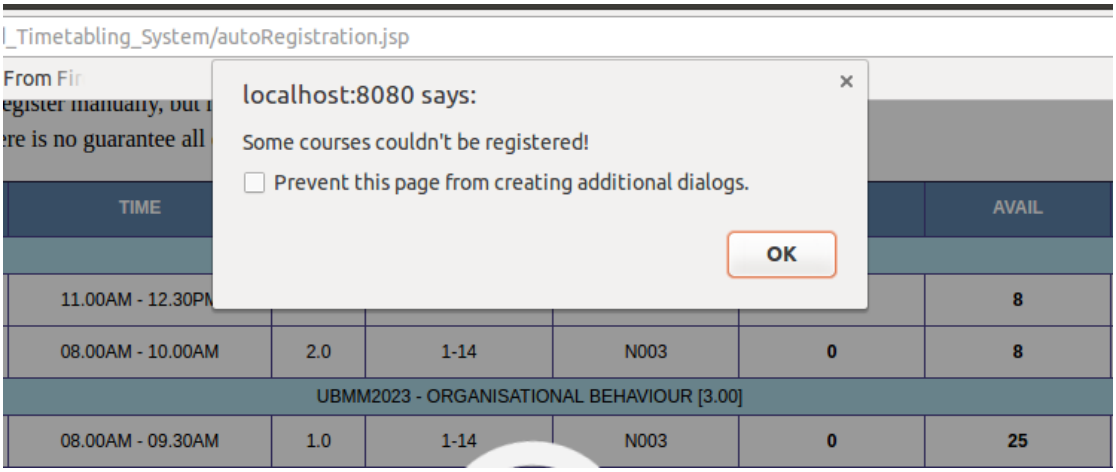


Figure 5-3-1-F16: Auto-registration error

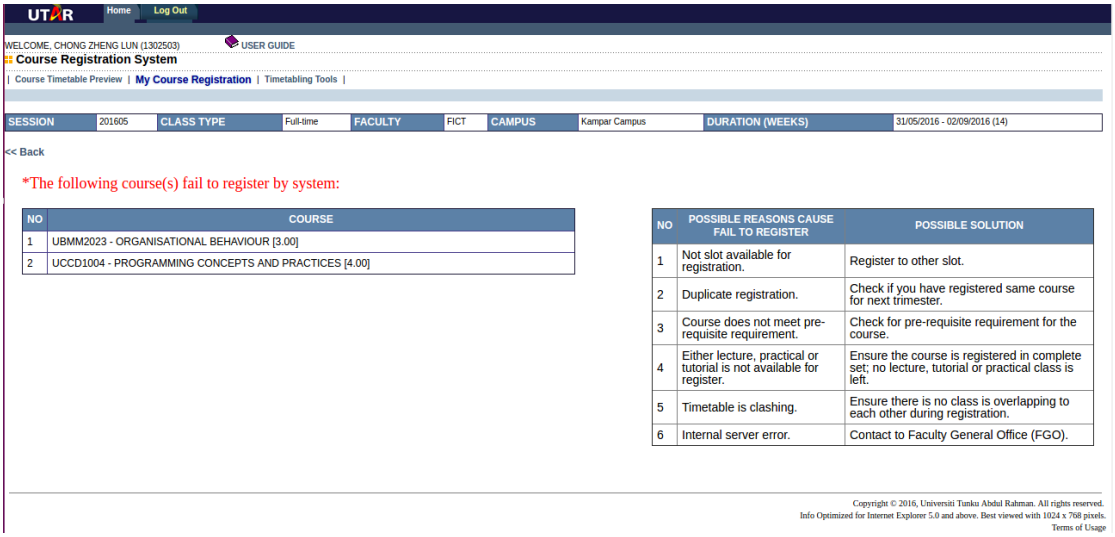


Figure 5-3-1-F17: Auto-registration display

Table 5-3-T1 shows the possible error and the action cause to error as well as whether the error could be handled by the system in each module. Some inputs will be randomly enter to test the possible error:

Table 5-3-1-T1: Error handled by the system in each module

Error	Possible action to cause error	Error caught by system?
User Login & Logout		
Password Error	Invalid password	Yes
User ID Error	Invalid user ID	Yes

Advisory Status	False value of advisory status in database	Yes
Captcha Error	Value enter does not match captcha value	Yes
Course Preview Module		
Search Record Not Found Error	Invalid course code	Yes
	Invalid lecturer's or tutor's name	Yes
	No record match to the day query	Yes
	No record match to start time query	Yes
	No record match to end time query	Yes
	End time is earlier than start time	Yes
	Start is later than end time	Yes
Course Registration Module		
Search Record Not Found Error	Invalid course code	Yes
Auto-register/ register course fail	Slot is full	Yes
	Does not meet pre-requisite requirement	Yes
	Exceed credit hour	Yes
	Does not register the course according to requirement	Yes
	Time is clashing	Yes
Update course fail	Slot is full	Yes
	Time is clashing	Yes
Automated Timetabling Tool		
Refuse to generate timetable	No course is selected	Yes
Query error	Start time later than end time	Yes
	End time earlier than start time	Yes
Timetable generation fail	Class clashing between the courses	Yes
	Generating time exceed the boundary	Yes

	Time clashing with the query set	Yes
Load error	No saved record in database	Yes
Save error	Blank timetable to save	Yes
Send to mail error	Cannot connect to the service	Yes
Refuse to proceed to Let Me Do!	Blank timetable is generated	Yes
IN-HAND Timetabling Tool		
Search Record Not Found Error	Invalid course code	Yes
	Invalid lecturer's or tutor's name	Yes
	No record match to the day query	Yes
	No record match to start time query	Yes
	No record match to end time query	Yes
	End time is earlier than start time	Yes
	Start is later than end time	Yes
Insert fail	Timetable is clashing	Yes
Load error	No saved record in database	Yes
Save error	Blank timetable to save	Yes
Send to mail error	Cannot connect to the service	Yes
Refuse to proceed to Let System Do!	Blank timetable is generated	Yes
Hybrid Mode		
Let Me Do fail	Hybrid mode is deactivated	Yes
	No record found in session storage	Yes
Let System Do fail	Hybrid mode is deactivated	Yes
	No record found in session storage	Yes

5-3-2 System Functionality and System Respond

The following figures show the basic system's functions and some notification after the action is performed:

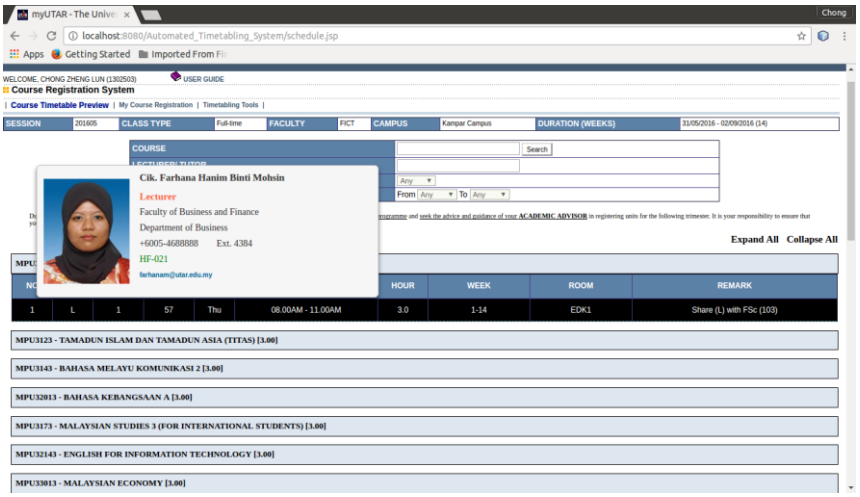
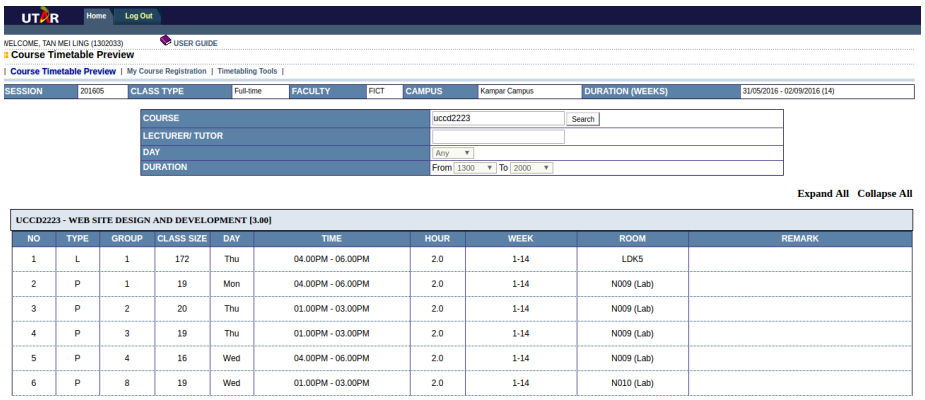


Figure 5-3-2-F1: Show lecturer and tutorial in course preview



NO	TYPE	GROUP	CLASS SIZE	DAY	TIME	HOUR	WEEK	ROOM	REMARK
1	L	1	172	Thu	04.00PM - 06.00PM	2.0	1-14	LDK5	
2	P	1	19	Mon	04.00PM - 06.00PM	2.0	1-14	N009 (Lab)	
3	P	2	20	Thu	01.00PM - 03.00PM	2.0	1-14	N009 (Lab)	
4	P	3	19	Thu	01.00PM - 03.00PM	2.0	1-14	N009 (Lab)	
5	P	4	16	Wed	04.00PM - 06.00PM	2.0	1-14	N009 (Lab)	
6	P	8	19	Wed	01.00PM - 03.00PM	2.0	1-14	N010 (Lab)	

Figure 5-3-2-F2: Search result return

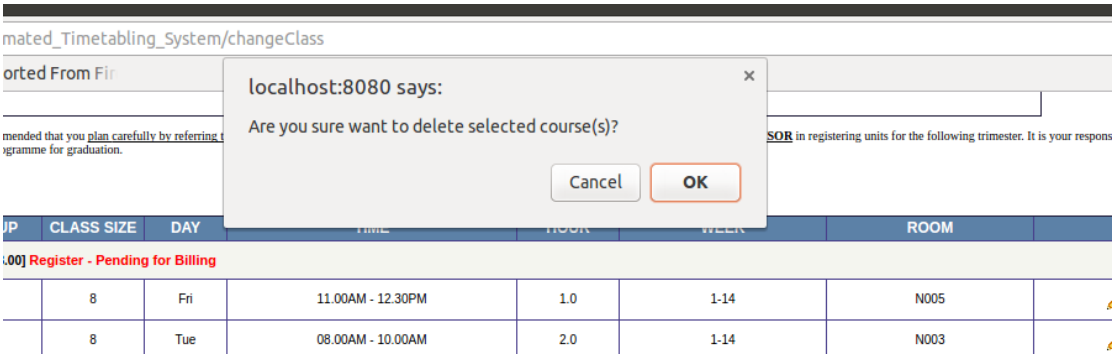


Figure 5-3-2-F3: Delete course confirmation

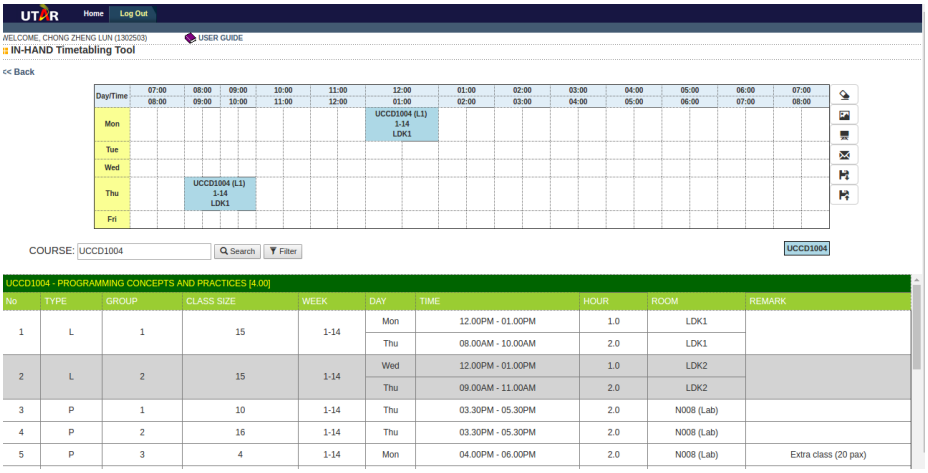


Figure 5-3-2-F4: IN-HAND add class to timetable

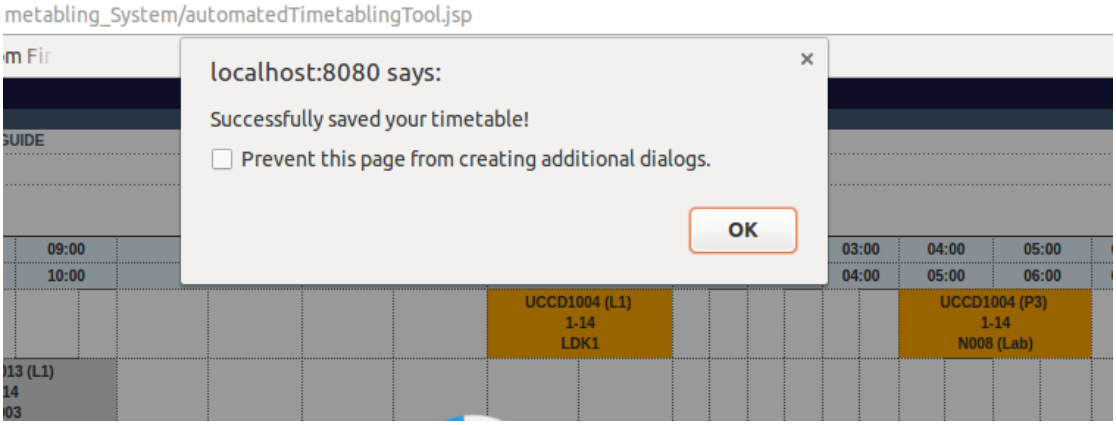


Figure 5-3-2-F5: Save Timetable

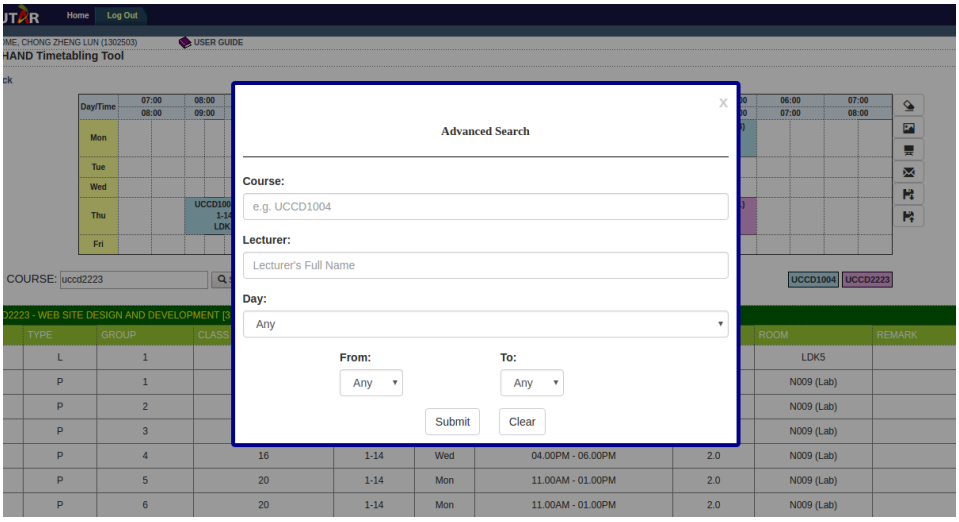


Figure 5-3-2-F6: Filter in IN-HAND Timetabling Tool

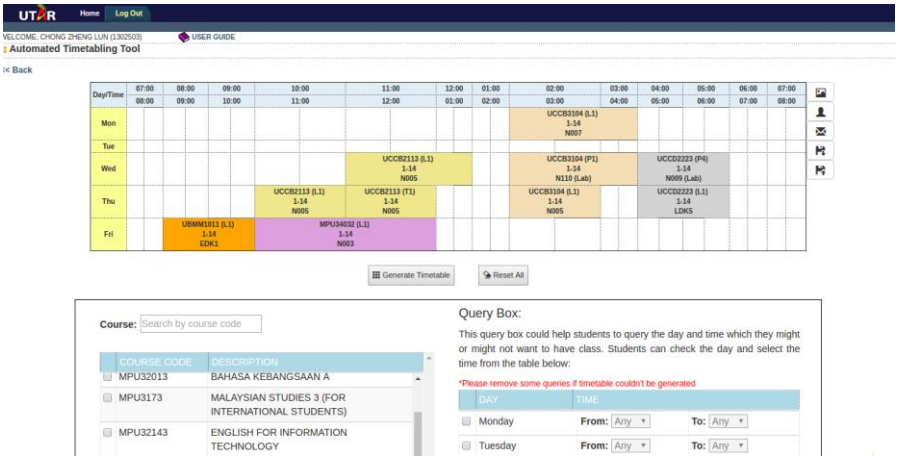


Figure 5-3-2-F7: Successfully generate timetable

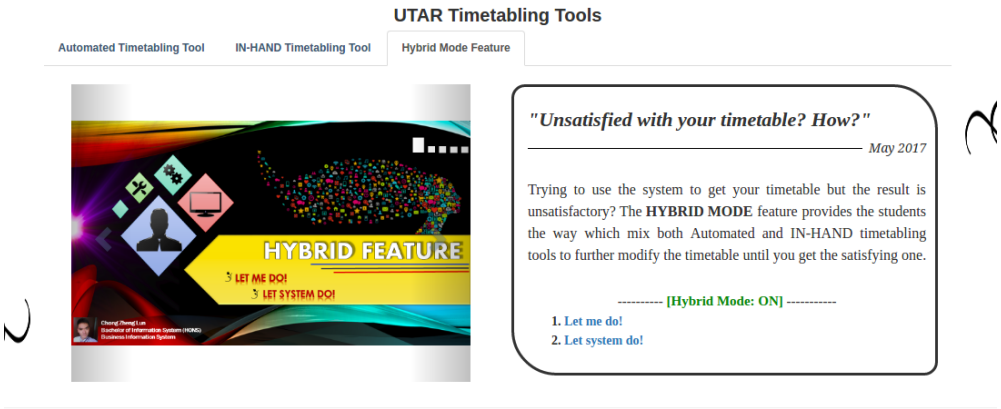


Figure 5-3-2-F8: Hybrid Mode off/ on indicator

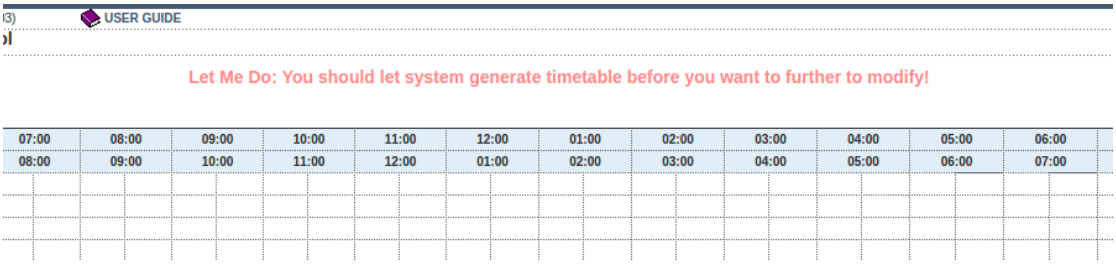


Figure 5-3-2-F9: Let Me Do when hybrid mode is off

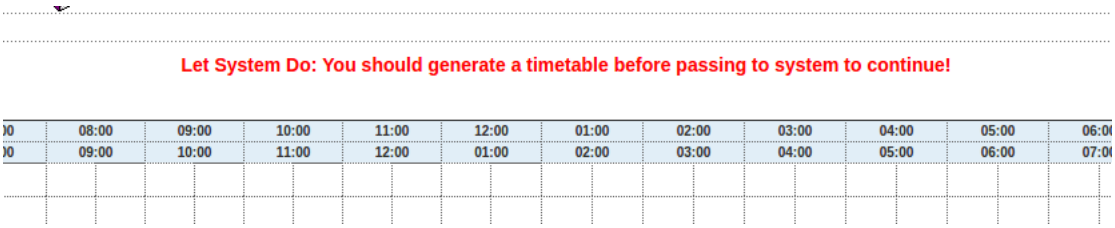


Figure 5-3-2-F10: Let System Do when hybrid mode is off

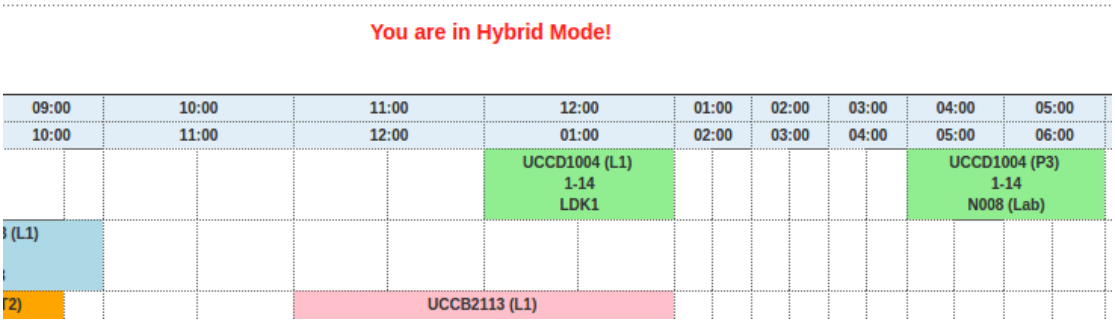


Figure 5-3-2-F11: Hybrid mode is activated and message display

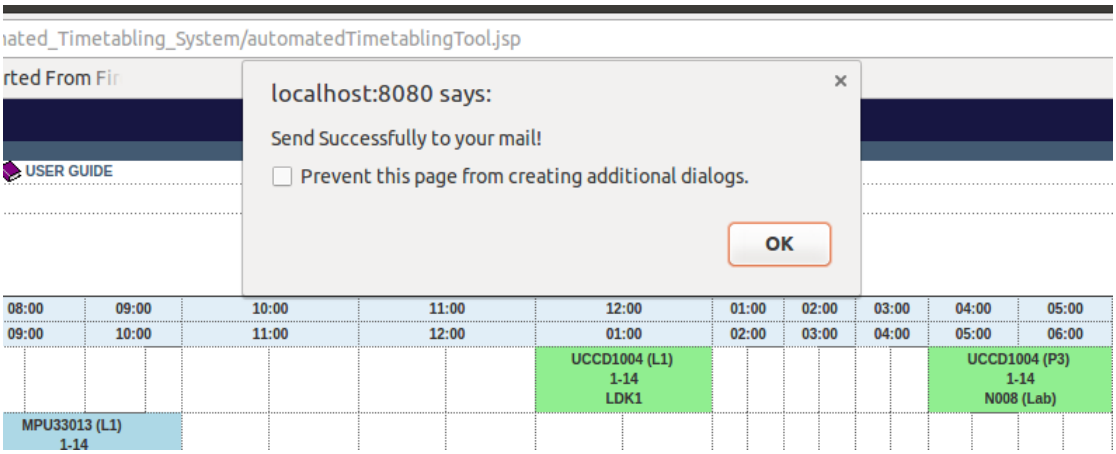


Figure 5-3-2-F12: Successfully send mail notification

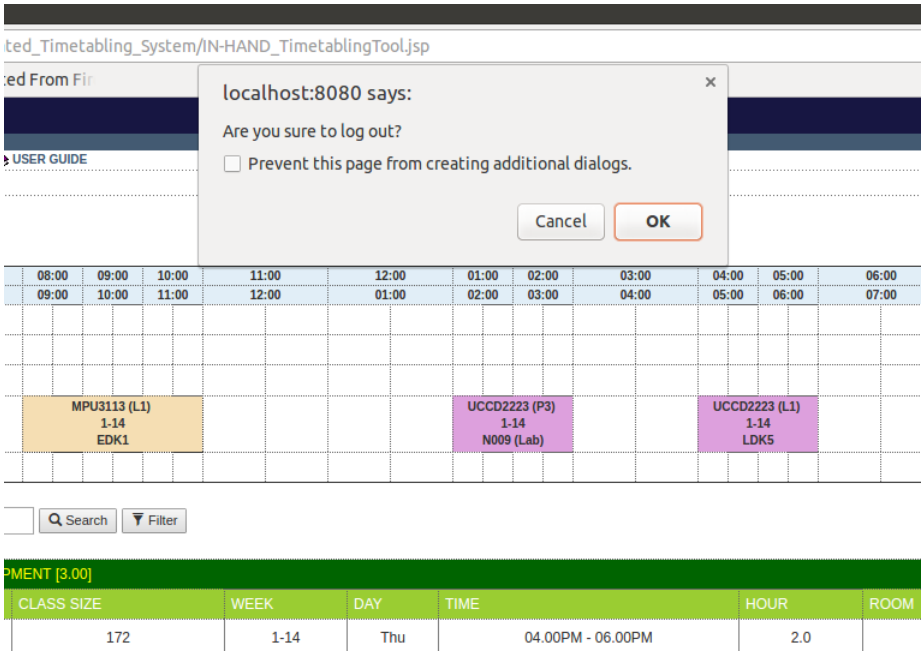


Figure 5-3-2-F13: Logout confirmation

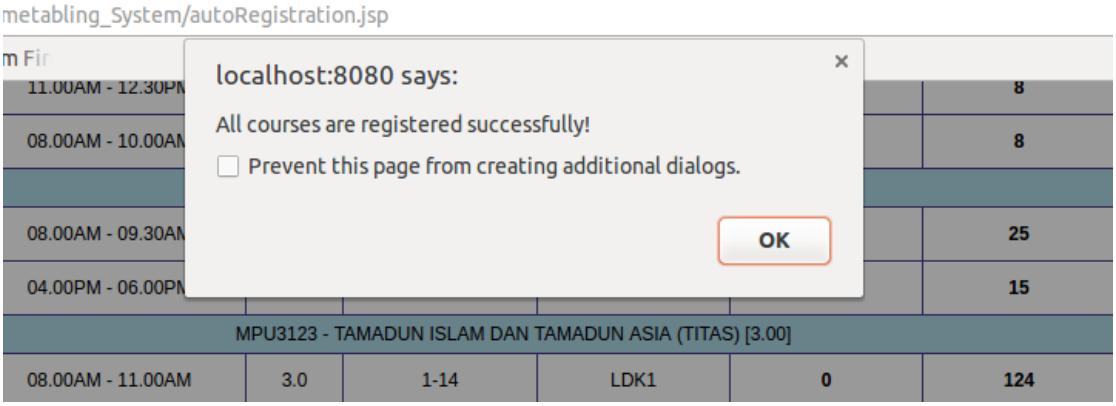


Figure 5-3-2-F14: Auto-registration successful

Table 5-2-2-T2 shows the functionality of each module and the test result to indicate the function of the element in each module is partially or fully workable to produce desirable output or result.

Table 5-3-2-T2: Functionality of each module

Module	Functionality	Fully/ Partially Workable?
Login & Logout	Login	Fully Workable
	Reset	Fully Workable
	Logout	Partially Workable

Course Preview	Display collapsed course	Fully Workable
	Toggle course	Fully Workable
	Collapse all	Fully Workable
	Expand all	Fully Workable
	Search by course code	Fully Workable
	Search by lecture/ tutor	Fully Workable
	Search by day	Fully Workable
	Search by start time	Fully Workable
	Search by end time	Fully Workable
	Search by different combination	Fully Workable
	Highlight rows	Fully Workable
	Display lecture/ tutor for each slot	Fully Workable
Course Registration	Search by course	Fully Workable
	Register course	Fully Workable
	Update course	Fully Workable
	Remove course	Fully Workable
	Auto Registration	Fully Workable
IN-HAND Timetabling Tool	Clear all	Fully Workable
	Export image	Fully Workable
	Let System Do!	Fully Workable
	Send to mail	Fully Workable
	Save	Fully Workable
	Load	Fully Workable
	Search by course code	Fully Workable
	Search by lecture/ tutor	Fully Workable
	Search by day	Fully Workable
	Search by start time	Fully Workable
	Search by end time	Fully Workable
	Search by different combination	Fully Workable
	Add slot	Fully Workable
	Remove slot	Fully Workable

	Highlight slot	Fully Workable
	Express search	Fully Workable
	Highlight rows	Fully Workable
	Display tooltip	Fully Workable
	Colour assignment	Fully Workable
Automated Timetabling System	Export image	Fully Workable
	Let Me Do!	Fully Workable
	Send to mail	Fully Workable
	Save	Fully Workable
	Load	Fully Workable
	Search by course code	Fully Workable
	Query box	Fully Workable
	Timetable generation	Fully Workable
	Reset all	Fully Workable
	Display tooltip	Fully Workable
	Colour assignment	Fully Workable
Hybrid Mode	Redirect user	Fully Workable
	Message Display	Fully Workable
	Session storage checking	Fully Workable
	Remain data when clear all	Fully Workable
	Remain colour assignment when clear all	Fully Workable
	Activate hybrid mode indicator	Fully Workable

In short, all functions in each proposed module are function well and produce desirable output to the user. However, for the logout function, it consider as work partially because after user logout the account, the page will still display when user click to back button. The data still keep in the browser even the user logout, but when the user restarts the browser, the user still requires to login in order to access to the system.

Chapter 6: Conclusion

Automated Timetabling System is a web-based system proposes for UTAR students to help them in planning and registering for their timetable. As mentioned in Chapter 1, most of the students usually utilize the time more than what they expected just for planning the timetable with the existing system. This is because the time schedule always clashing with other classes during the planning and other factors such as considering of a lecturer who lecturing the class, break time and so forth. Thus, this system mainly serves for the purpose of reducing the time and error during the students plan for their timetable. Besides, it also aims to improve the user interface to increase the readability level which can enhance the user experience.

In Chapter 2, some of the course registration systems from other universities have been studied as the benchmark. The course registration system in The Ohio State University provides the schedule planner to help students plan their timetable which the function is similar to the proposed system. However, the proposed system provides the hybrid feature which the students not only can auto-generate the timetable but also can further manually to modify the timetable by using the timetabling tool. Moreover, after completing the planning, the schedule will send to the students as notification for them to keep as a record.

While in Chapter 4, the entire system designs are discussed and the significant modules and algorithms are discussed in details. The diagrams such as use case diagram, class diagram, wireframe diagram and so forth demonstrate the design of the system and how it is working. Furthermore, the database design shows what data to be stored in database. For the part of module design, it shows how the automation done and architecture of the module.

As mentioned in Chapter 4, the methodology also has discussed which the waterfall pattern will be using for building the whole system. In order to collect the user requirement and problems they facing, 50 of survey questionnaire will be distributed for data collection and result will be analyze in Chapter 2. Besides, the proposed system will be built by using the Java platform as server side scripting which is J2EE, and for client side scripting, there will be using HTML5, JQuery, JavaScript, JSP and so forth. For the database, it will be using MySQL to manage the collected data.

Chapter 5 shows the system testing with the screen shots of the respond in

respond to different scenarios. 3 testing are conducted which include User Interface Testing, Functionality Testing and System Performance Testing are conducted for each module to ensure the system is working well. The result is discussed and further modification is done to debug and enhance the system based on the result found to improve the quality of the system.

It is a challenge for building such system to support the automated generation and registration of the timetable at the same time to satisfy the user's requirement. The database should be well designed and the algorithms should be well studied in order to support the auto-function. The design of user interface such as navigation panel, labeling, and form also should be well designed to enhance the user experience. The design is expected to minimize the user effort to use the system and increase the productivity.

The important and novelties of the system should be highlighted is the part of automated generation and registering of the timetable. This is because it could significantly help lot of the UTAR students in reducing their time and effort during planning and registering of their timetable. Unlike the existing one, it provides the convenient way the student to plan and register the timetable which just by search-and-click. Based on the benchmarking done, there is no other university provide such features for the students to plan and register their timetable.

However, the system could be improved in future by enhancing the automated algorithms to be smart to generate the timetable which could match the preference of the student based on pass record generated. Besides, for the auto-registration, it could provide the "wish list" feature for those who unable to register all courses successfully, the courses which could not register by the system could put in the wish list and wait for registering if there is slot available for it when someone drop such course. All in all, there are still many improvement could be done to enhance the system; the developers should put more effort to develop the system which could further benefit the students not only for UTAR, but other universities in future.

REFERENCES

Michella, L 2015. *Why White Space is so Important in Web Design.* Available from: <http://www.seguetech.com/blog/2013/03/20/why-whitespace-important-web design> [21 May 2016]

Sabina, I 2012. *8 Guidelines For Better Readability On The Web*. Available from: <http://blog.usabilla.com/8-guidelines-for-better-readability-on-the-web/> [21 May 2016]

Concordia University n.d. *Course Registration*. Available from: < https://www.concordia.ca/students/registration.html#step3a> [24 May 2016]

Concordia University n.d. *How to Register for a Course*. Available from: < https://www.concordia.ca/content/dam/concordia/docs/your-sis/SIS-Register for-Course.pdf> [24 May 2016]

Smith College n.d. *Course Rgistration*. Available from: < http://www.smith.edu/registrar/registration.php> [23 June 2016]

Smith College n.d. *Web Registration-Student Access*. Available from: < http://www.smith.edu/registrar/documents/10BannerWebregistration.pdf> [23 June 2016]

The Ohio State University n.d. *Schedule Planner: Plan Your Schedule*. Available from: <https://it.osu.edu/assist/sis/WebHelp/studentcenter/sc_schedule_planner.htm> [22 May 2016]

The Ohio State University n.d. *Search for a Class.* Available from: < https://it.osu.edu/assist/node/1635> [22 May 2016]

The University of Melbourne n.d. *Class registration.* Available from: < https://students.unimelb.edu.au/admin/class-registration> [24 May 2016]

The University of Melbourne n.d. *Timetable*. Available from: < https://sws.unimelb.edu.au/2016/> [24 May 2016]

Tutor 2011. *Waterfall Model*. Available from: < https://www.sdlc.ws/waterfallmodel/> [29 May 2016]

University of Manitoba n.d. *Aurora Student: What is Aurora Student*. Available from: <http://intranet.umanitoba.ca/student/records/2332.html> [20 May 2016]

University of Manitoba n.d. *Class Schedule*. Available from: < https://aurora.umanitoba.ca/banprod/bwckschd.p_disp_dyn_sched> [20 May 2016]

Zeeshan, AA 2015. *Programming in Java using MVC architecture.* Available from: < http://www.codeproject.com/Articles/879896/Programming-in-Java-using-theMVC-architecture> [29 May 2016]

APPENDIX A: SURVEY QUESTIONNAIRE

UNIVERSITI TUNKU ABDUL RAHMAN
Faculty of Information and Communication Technology
Jalan Universiti, Bandar Barat, 31900, Kampar, Perak

BACHELOR OF INFORMATION SYSTEM (HONS) BUSINESS INFORMATION SYSTEM

UCCD3583 PROJECT 1
SURVEY QUESTIONAIRE

Faculty: _____

1. What are the problems you normally face during planning your timetable? (Can choose more than one)
   - ☐ Clashing of time with other classes
   - ☐ Don't know how to start to plan the timetable
   - ☐ Too many options to choose for lecture/ tutorial class
   - ☐ Too many classes on a day/ period (No break time between classes)
   - ☐ The venue is too far for the next class

   Others: _____

2. Do you spend much of the time than what you expected on planning or re-arrange your timetable in UTAR? If yes, how much? (eg. Expected in 10 minutes, but 5% more than what I expected that is 12 minutes I spent.)
   - ☐ No
   - ☐ 5% - 10%
   - ☐ 11% - 20%
   - ☐ 21% - 30%
   - ☐ More than 30%

3. The lecturer/ tutor who going to teach the subject are the factor for you to plan your timetable.
   - ☐ Yes
   - ☐ No

4. What is the other factor will you considered on planning your timetable?
   - ☐ Break time between the classes
   - ☐ Venue between 2 classes
   - ☐ Time the class start of the day (eg. The class start at 10.00a.m. on Tuesday)
   - ☐ Time the class end of the day (eg. The class end at 4.00p.m. on Monday)
   - ☐ The days to take classes

5. Have you been wrongly to plan your timetable due to the overlook the schedule? (design issue: looking for the wrong time for the other classes)

   ☐ Always ☐ Sometimes ☐ Never

6.

220	T	1	33	Wed	05:00 PM - 06:30 PM	1.5	1-14	H008	
221	T	2	33	Mon	09:30 AM - 11:00 AM	1.5	1-14	H012	
222	T	3	33	Wed	03:30 PM - 05:00 PM	1.5	1-14	H012	
223	T	4	33	Mon	03:30 PM - 05:00 PM	1.5	1-14	H012	
224	T	5	33	Wed	02:00 PM - 03:30 PM	1.5	1-14	H010	
225	T	6	33	Wed	03:30 PM - 05:00 PM	1.5	1-14	H010	
226	T	7	33	Thu	09:30 AM - 11:00 AM	1.5	1-14	H016	
227	T	8	33	Thu	11:00 AM - 12:30 PM	1.5	1-14	H016	
228	T	9	33	Tue	08:00 AM - 09:30 AM	1.5	1-14	H010	
229	T	10	33	Tue	09:30 AM - 11:00 AM	1.5	1-14	H010	
230	T	15	27	Thu	08:00 AM - 09:30 AM	1.5	1-14	H012	
231	T	16	22	Wed	11:00 AM - 12:30 PM	1.5	1-14	H010	
232	T	17	20	Mon	03:30 PM - 05:00 PM	1.5	1-14	H010	
UBMM3013 - STRATEGIC MANAGEMENT [3.00]									
233	L	1	259	Wed	10:00 AM - 12:00 PM	2.0	1-14	IDK2	
234	L	2	260	Fri	02:30 PM - 04:30 PM	2.0	1-14	IDK2	FSC-22
235	T	7	33	Tue	03:30 PM - 05:00 PM	1.5	1-14	H010	
236	T	8	31	Mon	11:00 AM - 12:30 PM	1.5	1-14	H016	
237	T	9	33	Tue	11:00 AM - 12:30 PM	1.5	1-14	H012	
238	T	12	33	Mon	08:00 AM - 09:30 AM	1.5	1-14	H016	
UBTC2013 - CONSUMER BEHAVIOUR [3.00]									
239	L	1	68	Mon	08:00 AM - 10:00 AM	2.0	1-14	H215	
240	T	2	17	Fri	09:00 AM - 10:00 AM	1.0	1-14	H002	
241	T	3	19	Wed	10:00 AM - 11:00 AM	1.0	1-14	H001	

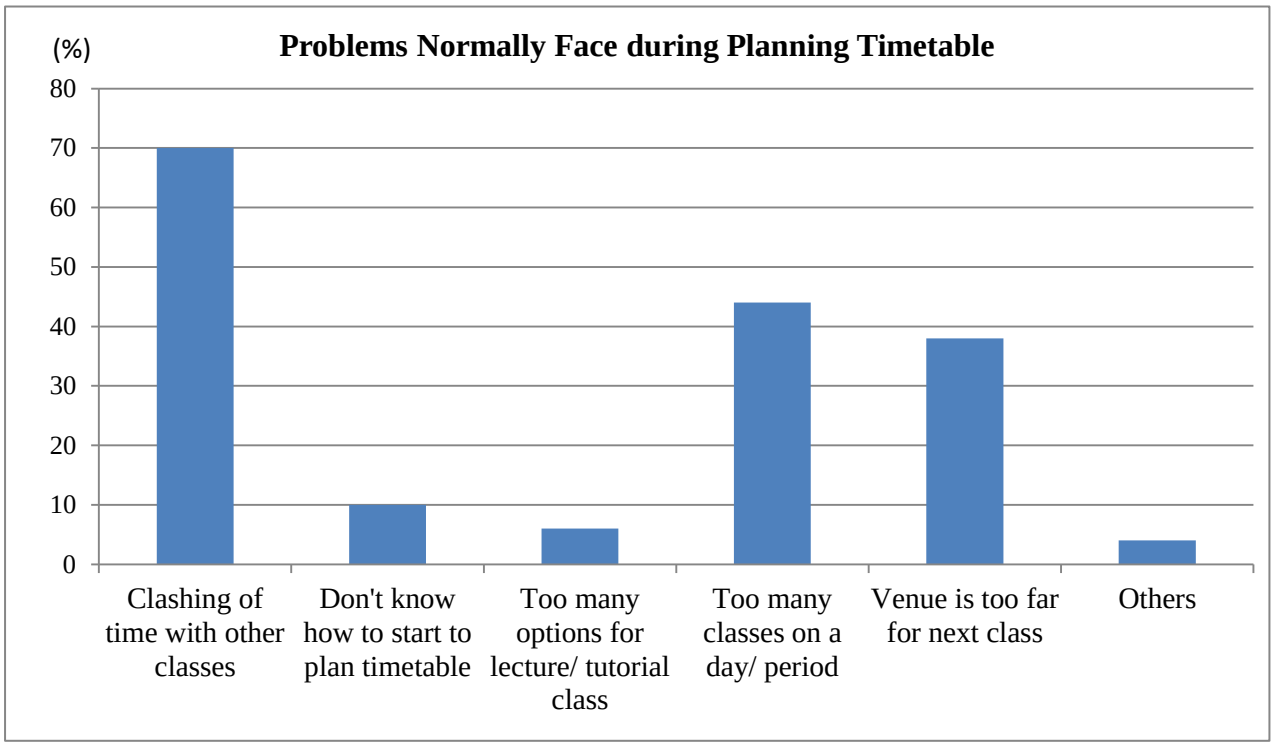
Please rate the design of the readability of the class schedule display on the UTAR course registration system (eg. View the time available for each tutorial/ lecture class). (1 = easiest to view, 5 = difficult to view)

1	2	3	4	5

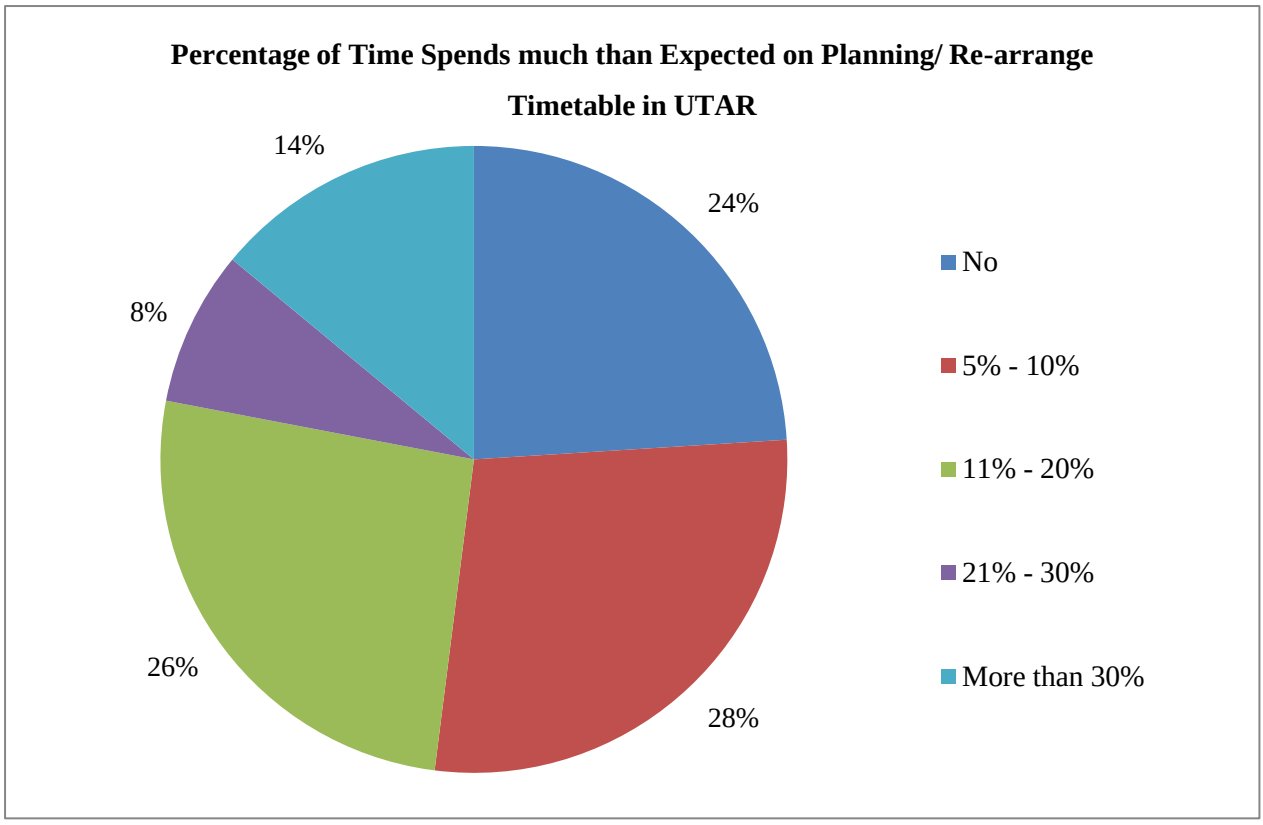
7. If there UTAR implements a system to automatically help you to plan the timetable based on the subject you selected, would you like to use it?
   - ☐ Yes
   - ☐ No

- Thank You-

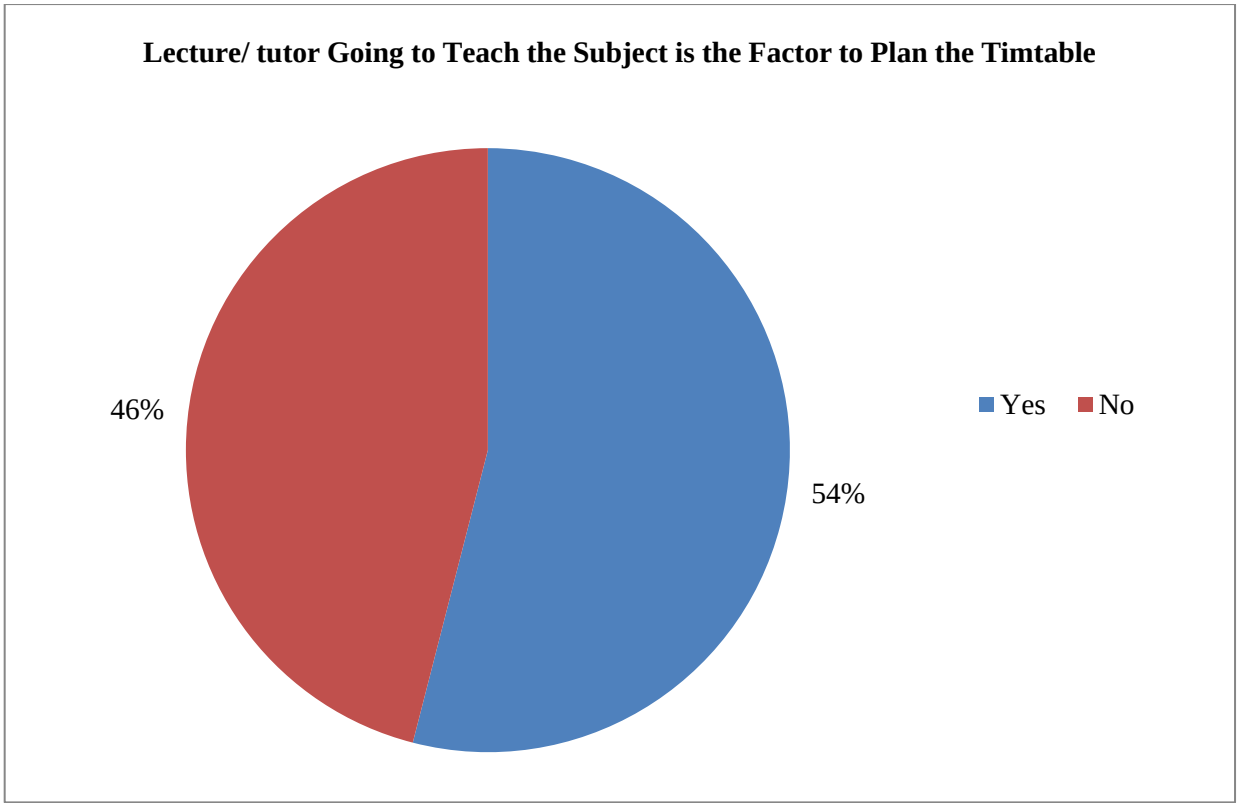
APPENDIX B: FINDING OF DATA COLLECTION



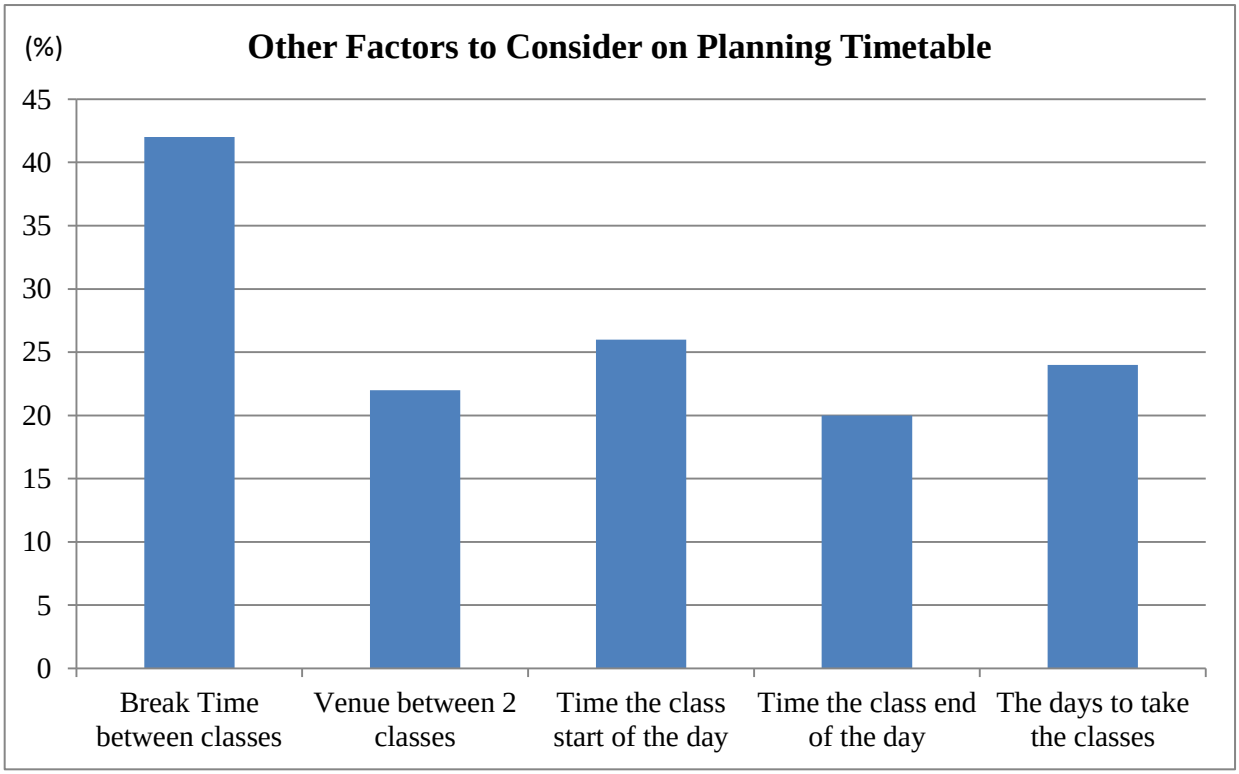
B-1: Problems normally face during planning timetable



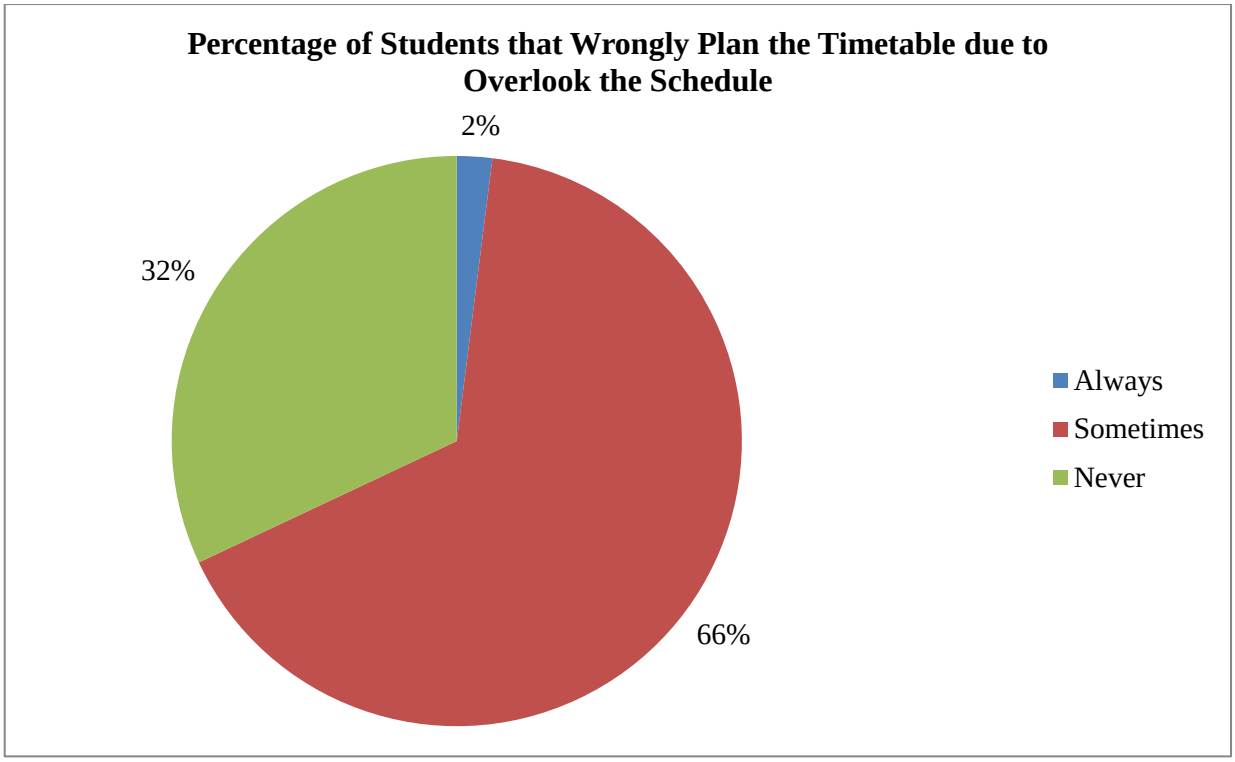
B-2: Percentage of time spends much than expected on planning/ re-arrange timetable in UTAR



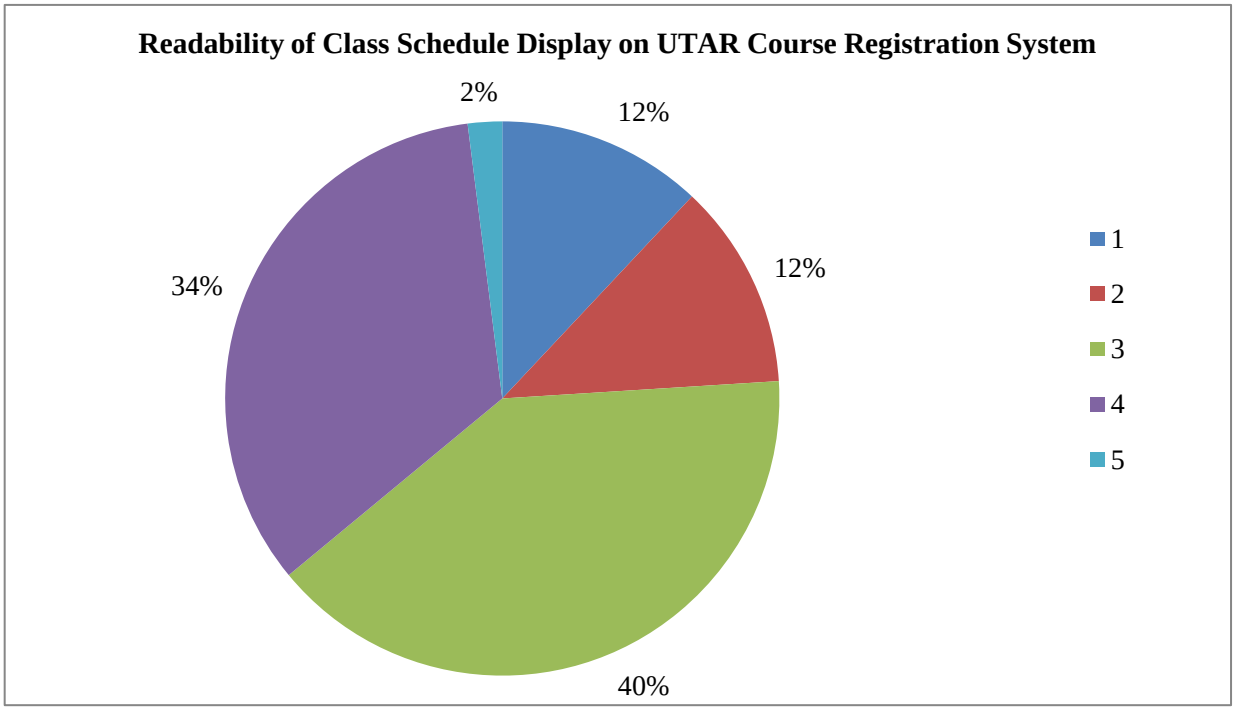
B-3: Lecture/ tutor going to teach the subject is the factor to plan the timetable



B-4: Other factors to consider on planning timetable

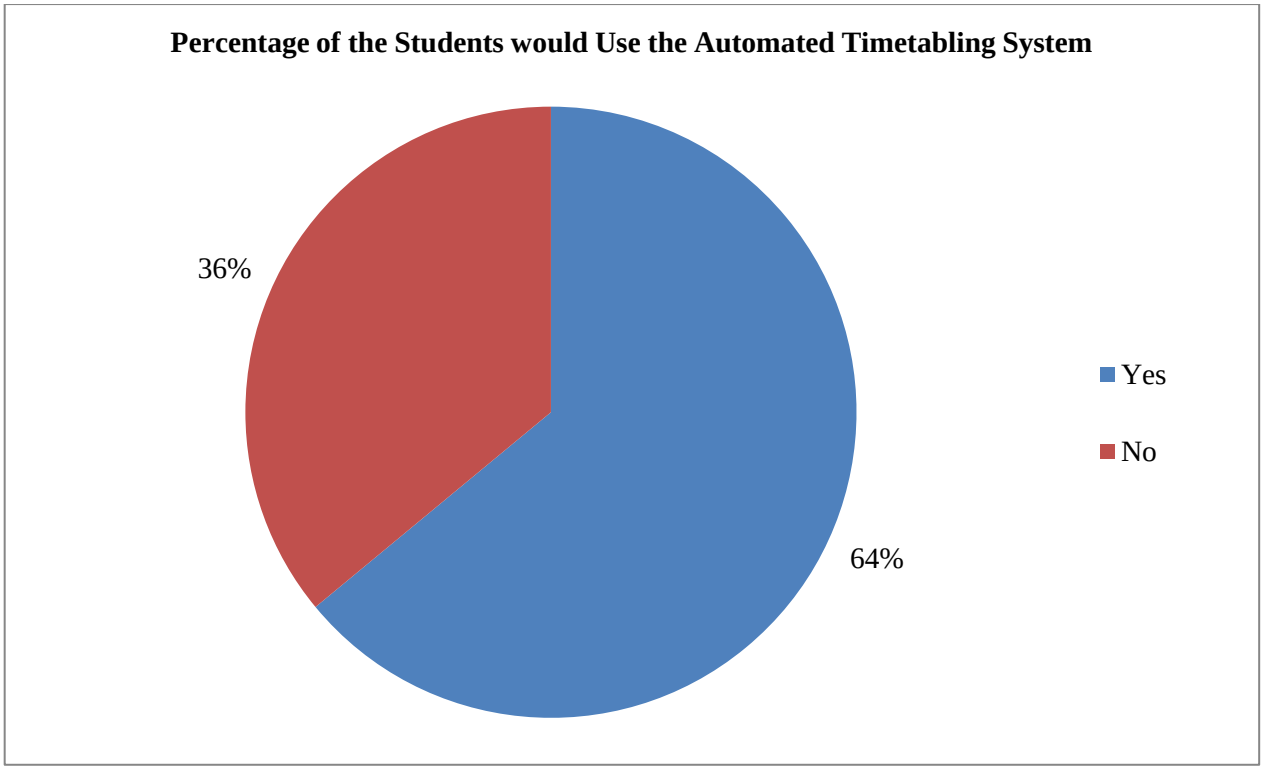


B-5: Percentage of students that wrongly plan the timetable due to overlook the schedule



* 1 = Easiest to view, 5 = Most difficult to view

B-6: Readability of class schedule display on UTAR course registration system



B-7: Percentage of the students would use the automated timetabling system

APPENDIX C: System Performance Evaluation Form

Test No.		**Date**			
Test Name					
Module					
Test Description:					
Assumption:					
Input:			**Remark**		
Output:			**Remark**		
Comment:					
Test Pass?	**Pass**		**Fail**		

Automated Timetabling System

By: Chong Zheng Lun

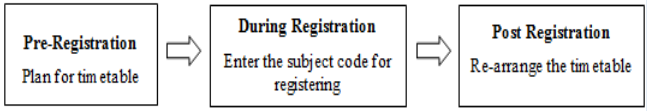
Supervised by: Dr. Doris Wong Hooi Ten & Mr Kesavan a/l Krishnan

UNIVERSITI TUNKU ABDUL RAHMAN

System Overview

A system allows the students in university to register the units that they would take for the next semester. Generally, the students can view, add, search and remove the subjects by using the system. The system also support automatically generation of timetable based on the units selected.

It has more interactive user interface and timetabling tools to assist the process of planning of timetable. The processes of registering the units for next semester are as following:



Problem Statements

- Students require spending much of the time to plan and re-arrange the timetable for next trimester.
- Students might easily overlook for the correct time and plan the timetable wrongly due to the readability of the design.
- Students are not able to view who will be lecturing the subjects before and during register the subjects.

Objectives

- To develop a system to reduce the time students spend on planning and register timetable.
- To develop system to eliminate error on planning and registering timetable.
- To develop an interface to enhance the user experience and readability of the class schedule displayed to the students.
- To develop and study of the algorithm to generate the timetable automatically.

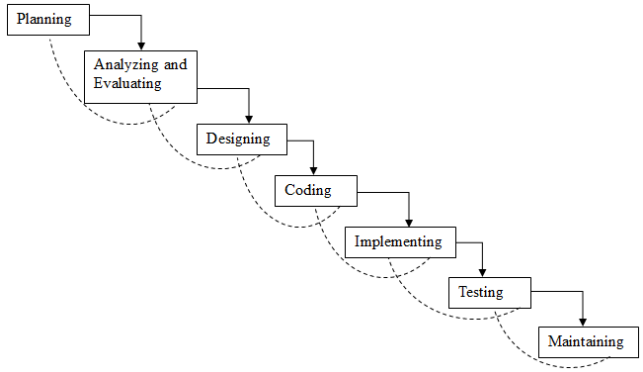
Literature Review

Timetabling System / Function/ Features	Automated Timetabling System	Concordia University's Course Registration	Course Registration System of University of	Aurora System in University of Manitoba	Course Registration System in The Ohio State University	BannerWeb Online Registration System of Smith
Auto-registration	✓				✓	
Search Tool	✓	✓	✓	✓	✓	✓
Email Notification	✓					
Course Registration	✓	✓	✓	✓	✓	✓
Error Message	✓	✓	✓		✓	✓
Course Details Viewing	✓	✓	✓	✓	✓	✓
Timetabling Tool	✓					
Hybrid Mode	✓					
Image Downloading	✓					

Special Features

- ✓ More interactive user interface
- ✓ Auto-generation and register of timetable based on selected units
- ✓ Timetabling tools assist the planning process
- ✓ Hybrid mode of generation of timetable

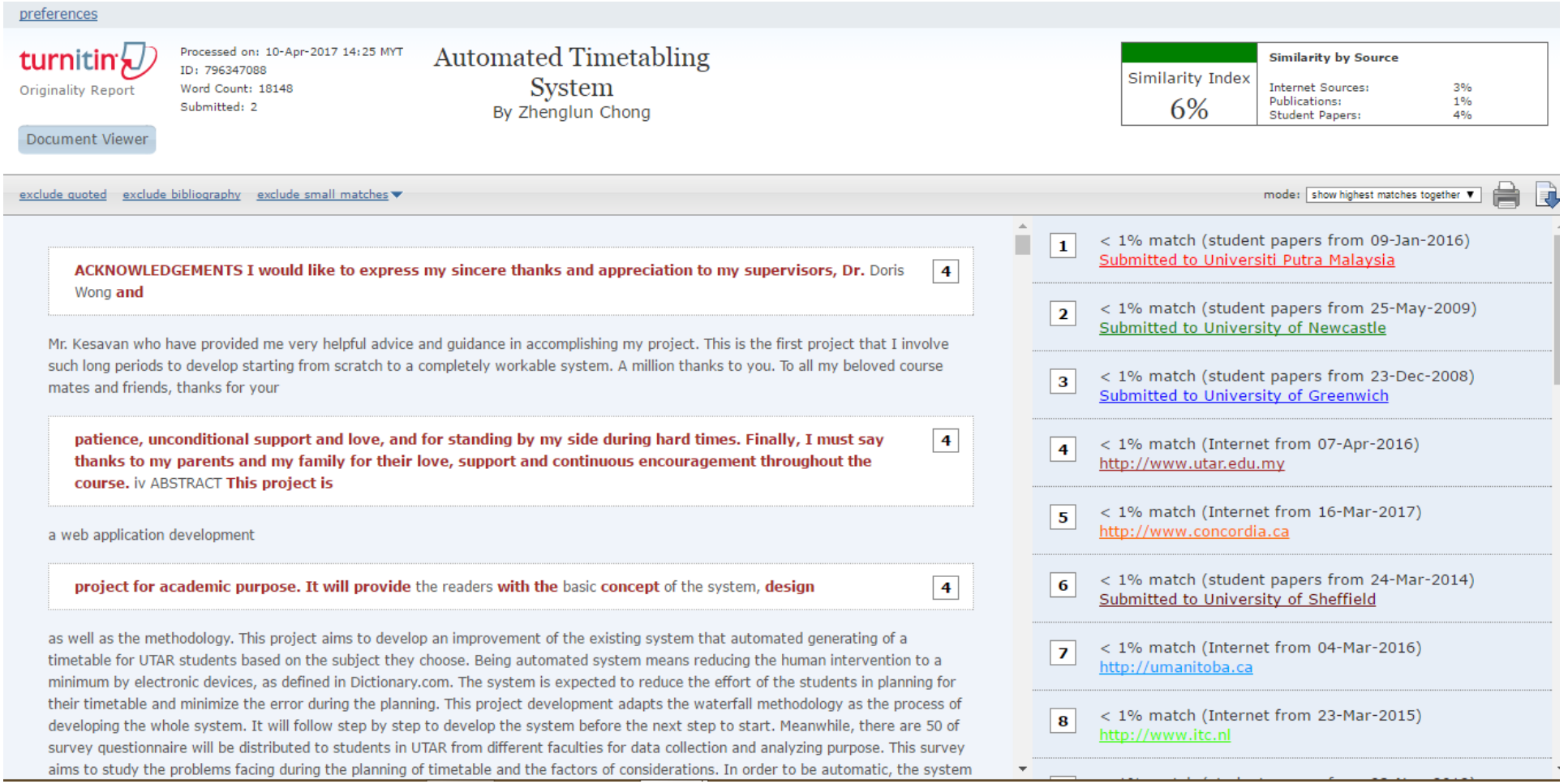
Design Methodology



Conclusion & Contributions

The primary beneficiary of this project would be the students who suffering in spending too much time to plan the timetable. It aims to minimize the time of students in planning their timetable. Besides, this project could help readers to reduce their efforts on developing the project start from scratch and emphasize more on other parts.

This project provides opportunity for the readers who interested on it to continue and further refine the algorithms or improve the system. As conclusion, this system mainly serves for the purpose of reducing the time and error during the students plan for their timetable and improving the user interface to enhance the user experiences.


preferences
turnitin
Originality Report
Processed on: 10-Apr-2017 14:25 MYT
ID: 796347088
Word Count: 18148
Submitted: 2
Automated Timetabling System
By Zhenglun Chong
Similarity Index
6%
Similarity by Source
Internet Sources: 3%
Publications: 1%
Student Papers: 4%
Document Viewer
exclude quoted
exclude bibliography
exclude small matches
mode: show highest matches together
ACKNOWLEDGEMENTS I would like to express my sincere thanks and appreciation to my supervisors, Dr. Doris Wong and
Mr. Kesavan who have provided me very helpful advice and guidance in accomplishing my project. This is the first project that I involve such long periods to develop starting from scratch to a completely workable system. A million thanks to you. To all my beloved course mates and friends, thanks for your
patience, unconditional support and love, and for standing by my side during hard times. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course. iv ABSTRACT This project is
a web application development
project for academic purpose. It will provide the readers with the basic concept of the system, design
as well as the methodology. This project aims to develop an improvement of the existing system that automated generating of a timetable for UTAR students based on the subject they choose. Being automated system means reducing the human intervention to a minimum by electronic devices, as defined in Dictionary.com. The system is expected to reduce the effort of the students in planning for their timetable and minimize the error during the planning. This project development adapts the waterfall methodology as the process of developing the whole system. It will follow step by step to develop the system before the next step to start. Meanwhile, there are 50 of survey questionnaire will be distributed to students in UTAR from different faculties for data collection and analyzing purpose. This survey aims to study the problems facing during the planning of timetable and the factors of considerations. In order to be automatic, the system
1 < 1% match (student papers from 09-Jan-2016) Submitted to Universiti Putra Malaysia
2 < 1% match (student papers from 25-May-2009) Submitted to University of Newcastle
3 < 1% match (student papers from 23-Dec-2008) Submitted to University of Greenwich
4 < 1% match (Internet from 07-Apr-2016) http://www.utar.edu.my
5 < 1% match (Internet from 16-Mar-2017) http://www.concordia.ca
6 < 1% match (student papers from 24-Mar-2014) Submitted to University of Sheffield
7 < 1% match (Internet from 04-Mar-2016) http://umanitoba.ca
8 < 1% match (Internet from 23-Mar-2015) http://www.itc.nl