University Course Timetabling System For Part-Time Students

Mohd Asyraf Ruslaan [a], Zalmiyah Zakaria [a,*], Mohd Zainuri Saringat [b], Shahreen Kasim [b]

[a]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia.
[b]Fakulti Sains Komputer Dan Teknologi Maklumat, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia
*zalmiyah@utm.my
* corresponding author

ARTICLE INFO

Article history
Received: September 2019
Revised: Oktober 2019
Accepted: November 2019

Keywords
Timetabling; Web System, Part-time Student

ABSTRACT

University timetabling system is a part of a timetabling problem that aims to produce course timetable that meets student needs such as the maximum number of subjects that can be offered, the maximum number of elective subjects that can be offered and the number of subject students can take. In every semester, the timetabling process in UTMSPACE is done manually where there are likely to be a small number of students who will have problems because the subject to be taken is not in the subjects offering list. Additionally, the number of subjects offered is also not optimal and this will result in a loss on UTMSPACE because each subject is offered at a cost. Therefore, in order to solve this problem, the Heuristic-based approach is used to overcome the problems mentioned and speed up the process to generate timetable. Heuristic engines have been developed using PHP language programming. This approach has been successfully tested and implemented using real-time scheduling data at UTMSPACE for Software Engineering course. The results show that Heuristics has successfully solved the problem of producing a timetable without affecting students who want to enroll the subject offered.

1. Introduction

In July 1993, the School of Professional and Continuing Education, University of Teknologi Malaysia, or better known as UTMSPACE, was established following the government's aspiration to increase the number of human capitals in Malaysia. UTMSPACE offers an alternative way for those who have missed the opportunity to study further in full-time mode.

There are three major programs offered by UTMSPACE in the School of Computing, which are Bachelor of Computer Science (Software Engineering), Bachelor of Computer Science (Computer Networks and Security) and Bachelor of Computer Science (Graphic Software & Multimedia) in three study centres, Kuala Lumpur, Penang and Johor Bahru. The course must be registered by each student according to their program.

The registration process will normally go through the systematic system where the system will automatically offer the student to take a course based on certain requirements. However, UTMSPACE still using the manual process to come out with the list of courses to be offered to the students. The way in which the course schedule for UTMSPACE students was created differed from a normal UTM student because of the class available only on the weekend, while the class for normal UTM students was available on the weekdays. The timetable slot is divided into 3 groups and each group has 6 hours of learning sessions.

The proposed system to overcome the manual process by developing a web-based system called University Courses Timetabling System for Part-time Students (UCTSPS) for UTMSPACE. The system will automate the process of filtering course taken by the student according to the specific required criteria and generate the list of course offered with a minimum of 3 courses can be taken by the student that has 3 or more course left and offer all of the subject left for student that has less than 3 courses. Next, to make sure that no student has no subject to register. Hence, the process of creating the timetable can be much easier

2. Problem Background

SPACE is an alternative program offered by UTM for a part-time student who wishes to continue their study. The study session only available on the weekends compared to the normal student session available on the weekdays. There is a big difference for the study session between both types of program. This difference leads to a problem in preparing the timetabling list for the UTMSPACE students. The problem tends to occur in the current timetabling list are:

- The current course timetabling system consuming a lot of time because of the manual process to filter the course taken by the student
- The current course timetable is not optimized and consumes a lot of resources.

3. Methodology

For UCTSPS, an agile methodology for software development is the most appropriate approach. Planning is incremental in agile processes and it is easier to change the process to reflect the changing requirements of stakeholders. Since the late 1990s, an agile methodology for software development has been explicitly known among software developers.

Agile focuses on the features of the system. Each feature is developed separately to focus on an important feature to deliver quickly to customers. In a short time box called iterations, the system is developed incrementally, usually lasting from one week to one month. The increment is deployed for use by the customers after one iteration.

The stakeholder feedback is collected after delivery. A stakeholder may comment on increment release demonstrations. This helps improve efficiency in responding more quickly to changing requirements, internal processes, and removing bad designs. As agile has an adaptive approach that can handle change, it reduces the cost of rework that includes both the analysis of requirements and the implementation of new functionality.

Other than that, as the validation process is also done incrementally, it manages to minimize the risk of overall project failure. The developer must then react quickly to the changes and proceed to the next iteration until the system is fully ready for customer release.

Basically, the agile process has many different types, but all involve specification, design, implementation, validation, and evolution. The scrum will be used for the University Courses Timetabling System for Part-time Student (UCTSPS) UTMSPACE as it divides the time into short work cadences, known as iteration. It will begin with the phase of planning, iterate, and end with the phase of release

A. Phase 1: Plan

The planning phase needs to be done to identify the background and case study of the University Courses Timetabling System for Part-time Student (UCTSPS). Then it will be accessed the need for the platform and importance to examine the relevance of developing UCTSPS. Finally, UCTSPS's goal and goals will be defined and clearly stated.

B. Iterate

There are other processes in iterate phase that will be divided into the features of the system. Each feature has its own phase of development that incrementally develops iterations in short time boxes. The development processes are analyzed, designed, developed and tested.

1) Analyze

In the analysis phase, the requirements are raised to describe the UTMSPACE behaviours of the University Courses Timetabling System for Part-time Student (UCTSPS). Functional, non-functional and domain requirements will be collected and elicited in this phase, then use case and any other details about UCTSPS will be modelled and documented.

2) Design

In the design phase, UCTSPS's software architecture, package decomposition, database design, and user interface will be modelled and documented. Unified Modeling Language (UML) will be used to model and design the diagram. The process of designing the UCTSPS components is done to describe, organize and structure. Once the developer understands, the phase will be developed in UCTSPS

3) Develop

UCTSPS will be developed using PHP with the Laravel Framework. In this phase, the database will be created, and the development environment will be prepared and configured to store all the records needed for UCTSPS. It will be released to the user after the development process has been completed as the increment release to be tested.

4) Test

In the testing phase, which is user and black-box testing, two methods will be used. In user testing, a sample data set will be tested by the users who are the AM or AAM and the results will be compared to the classification of the expert. This test will also measure the level of agreements and accuracy. While UCTSPS will be tested with different inputs for black-box testing, the actual output is compared with the expected output. In the next iteration, any correction of the test results can always be corrected.

C. Release

In the release phase, the UCTSPS is already a complete system. UCTSPS will be ready to be released to the user after some test correction and several iterations. Figure 1. Scrum Methodology for UCTSPS

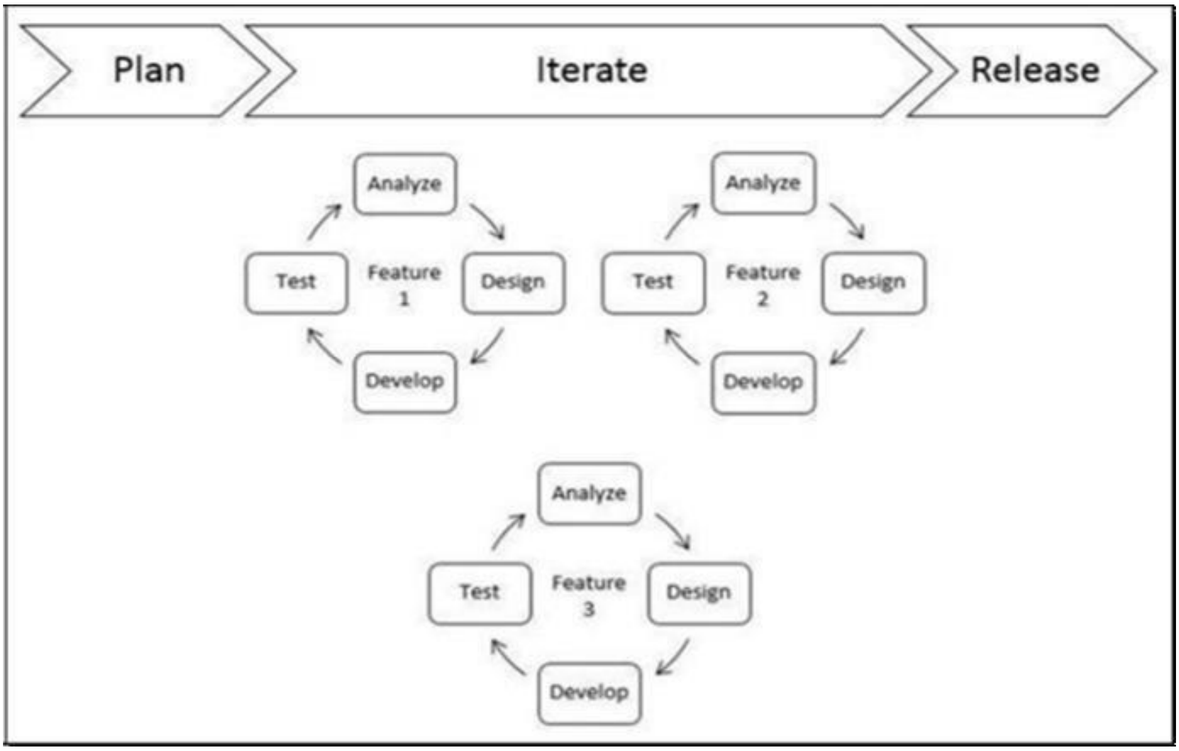


Figure 1. Scrum Methodology for UCTSPS

4. Result

A. Requirement Analysis and Design

Requirement analysis and design is the phase to determine requirements or needs to develop a new system. This phase is usually conducted after the requirement elicitation phase, where all requirements have already been gathered.

Through this phase, the functional and nonfunctional requirement are refined to discover the real needs of the system. UML diagrams are used to represent the components or elements of the proposed system in graphical form. The phase of requirement analysis and design gives a better understanding of system development because it represents the requirement of the whole system

1) Use case diagram

UCTSPS have 2 actors which are Academic Manager (AM) and Assistant of Academic Manager (AAM). There are 6 use cases for UCTSPS which are login, upload student data file,view list of the subject, view list of final year subject, generate a timetable and view subject offering list.

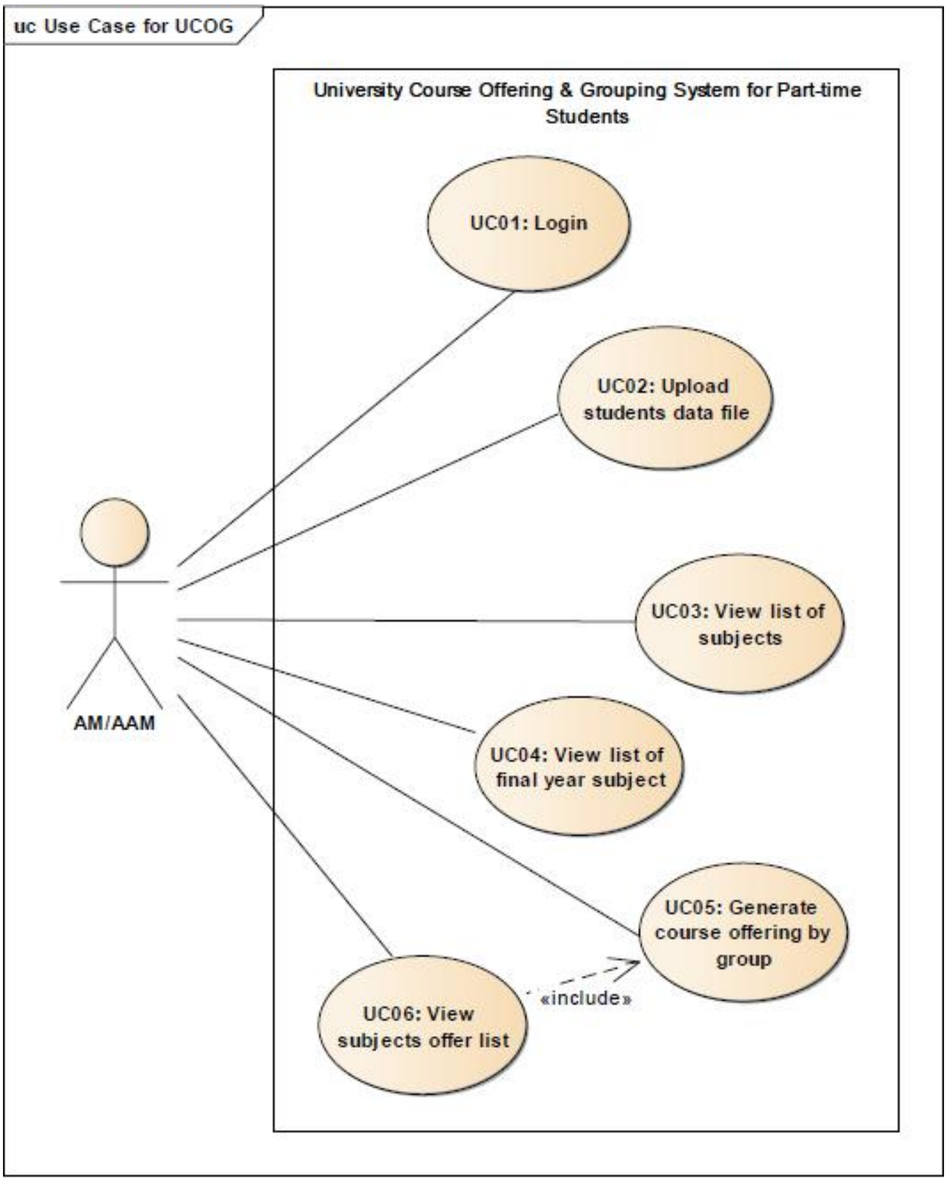

uc Use Case for UCOG
University Course Offering & Grouping System for Part-time Students
UC01: Login
UC02: Upload students data file
UC03: View list of subjects
UC04: View list of final year subject
UC05: Generate course offering by group
UC06: View subjects offer list
«include»
AM/AAM

Figure 2 : Use case diagram for UCTSPS

2) System Architecture

System Architecture is a set of rules and standards deployed in the technical framework of a computer system, customer requirements and specifications which design and integrate system components. In this project, a layered architecture is selected as UCTSPS system architecture. The layered architecture consists of the view layer, domain layer, and data access layer

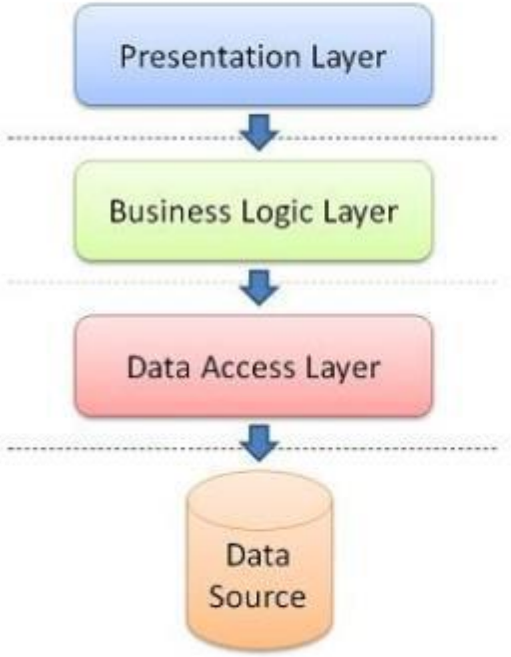


Figure 3 : Simple Three-tier laye architecture

In the view layer, only the system‘s interfaces are present to the users. Users interact with the interface like insert and view information. The data insert is then passed to the domain layer for processing and then goes to the data access layer. The domain layer is used to save the data and retrieve data from the data access layer. It acts as the bridge and coordinates the transfer of data between the view layer and data access layer. While the data access layer then builds the query and sends the data as a parameter to the database server.

By using the layered architecture style, the application is easier for maintenance. Since all layers‘ components are independent of each other, hence update or modification can be done without affecting the entire application. This ensures the modifiability and maintainability of UCTSPS.

3) Database Design

UCTSPS database consists of 5 tables which are students, matrices, subjects, temporary_subjects and users

TABLE I. DATABASE IN UCTSPS

Entity Name	**Description**
students	Contain information of the students
matrices	Contain information of the students that have been converted into a feasible matrix
subjects	Contain information on all subjects in the course
temporary_subjects	Contain information about the generated course offering
users	Contain information of the system users

4) The flow of generating timetable process

The process flow of generating timetable shows in Table 2 below.

TABLE 2 : The Flow Of Generating TimeTable For UCTSPS

	Input	Process	Output
1	Students Data in excel format	Upload excel file to data	Save uploaded excel data to the database
2	Students data in StudentsDB	Convert to feasible matrix	All data converted to ‘0’ or ‘1’ if data equals to ‘TC’ pr ‘Y’ or ‘0’, data will be saved in DB as ‘0’, else ‘1’
3	Feasible matrix	Calculate the total subject left	Get total subject left for each students row
4	Total subject left	Classify final year student	List of final year students
5	Final year student data	Remove constraints for elective groups	Insert needed subject to temporary_subjectsDB
6	Final year student	Check if a student with only 3 subjects left for compulsory subject offered	Insert needed subject to temporary_subjectsDB
7	Subjects in temporary_subjectsDB	Generate courses offering list	List of course offering
8	List of course offering	Generate timetable	Timetable

5) System Interface

This section will show the functionalities of the system in the graphical user interface form. These user interfaces act as an interaction medium between users and the system. A good user interface should easily be understandable and easy to use. Interface design should be simple and consistent because all these understandable and easy to use. Interface design should be simple and consistent because all these factors can give influence on the effectiveness of the system to interact with users.

This system is a web-based application. Thus, the interfaces are designed based on web interfaces. Figure 4 and Figure 5 are some of the interfaces that have been developed in this system. Figure 4 shows the interface for the user to import student‘s data. Figure 5 shows the interface for the user to view the generated subject offering list.

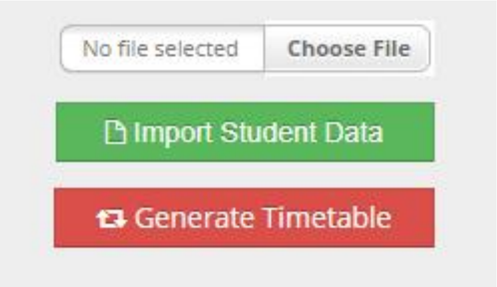
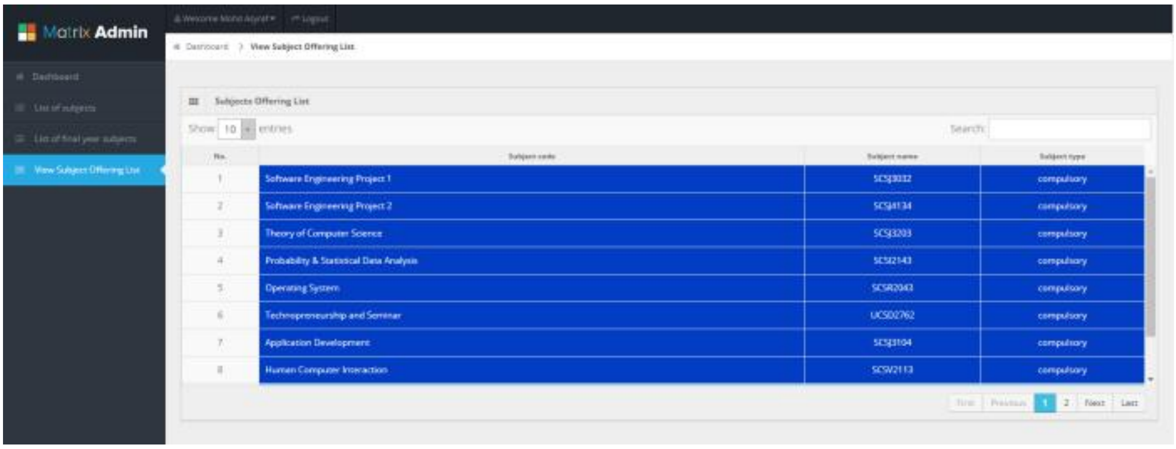


Figure 4. Interface for a user to import student‘s data


Matrix Admin
Welcome Mohd Asyraf
Logout
Dashboard
List of subjects
List of final year subjects
View Subject Offering List
Dashboard > View Subject Offering List
First
Previous
1
2
Next
Last

Subjects Offering List

Show 10 entries

Search:

No.	Subject code	Subject name	Subject type
1	Software Engineering Project 1	SCSJ3032	compulsory
2	Software Engineering Project 2	SCSJ4134	compulsory
3	Theory of Computer Science	SCSJ3203	compulsory
4	Probability & Statistical Data Analysis	SCSI2143	compulsory
5	Operating System	SCSR2043	compulsory
6	Technopreneurship and Seminar	UCSD2762	compulsory
7	Application Development	SCSJ3104	compulsory
8	Human Computer Interaction	SCSV2113	compulsory

Figure 5. Interface for a user to view generated subject offering list

5. Result Analysis

According to [5], software testing is any activities to evaluate the performance, capability or quality of software in order to determine whether it meets its required outputs and requirements. This activity is important for verification and validation of software purposes. The purpose of this activity is also for software quality assurance.

The main objective of software testing is to find errors when executing a program, system or software [6]. [7] stated that, as a result of software testing, it can increase customer confidence in the products. This is because, by conducting software testing, it can provide customers or stakeholders with details about the status or quality of a software product.

From the test result of the Black box testing and comparison between UCTSPS generated subject offering list and the manual subject offering list, the UCTSPS fulfil the objective of the project to produce a timetable with an optimized number of subjects to be offered.

6. Conclusion

At the end of the development of this project, this project managed to achieve its main goal and scope. The goal of this project is to develop a web-based University Courses Timetabling System for Part-time Student.

By using the system, users are able to generate a timetable with optimized number of subjects to be offer and ensure that every student that has more than three subjects left have at least three subjects to be registered while the student that have less than three subject left, all of the subject will be offered in the timetable.

However, there is always room for improvement. For future enhancements, the system‘s scope can be enhanced in term of function such as additional features to assign the generated subject to a timeslot by considering the subject that can be taken so that there would not be any subject clash for the student to register.

References

[1] Al-Maamari, A., & Omara, F. A. (2015). Task Scheduling Using PSO Algorithm in Cloud Computing Environments. International Journal of Grid Distribution Computing, 8(5), 245–256. http://doi.org/10.14257/ijgdc.2015.8.5.24

[2] Balaji, S., & Murugaiyan, M. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. International Journal of Information Technology and Business Management, 2(1), 26-30.

[3] Basics, S. (2011). Scrum Sprint. Retrieved 2 May 2018, from http://scrummethodology.com/scrum-sprint/

[4] Birds, F. o. (2015). Agile Software Development. Retrieved 10 April 2018, from http://www.flockofbirds.nl/summer-training

[5] W. C. Hetzel, The Complete Guide to Software Testing. Wellesley, Massachusetts: QED Information Sciences, 1988.

[6] ―Types of Software testing and definitions of testing terms — Software Testing Help,‖ 2017. [Online]. Available: http://www.softwaretestinghelp.com/types-ofsoftware-testing/. [Accessed: 09-Jan-2018].

[7] C. Kaner, ―Exploratory Testing,‖ Florida Institute of Technology, Florida, 2006

[8] Birds, F. o. (2015). Agile Software Development. Retrieved 10 April 2018, from http://www.flockofbirds.nl/summer-training/

[9] Burget, R., & Rudov, H. (2016). Teacher-oriented Fairness in Course Timetabling. Proceedings of the 11th International Conference on Practice and Theory of Automated Timetabling (PATAT-2016).

[10] Burke, E., Elliman, D., & Weare, R. (2014). A Genetic Algorithm Based University Timetabling System. Interface, 35–40. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2.3627&amp;rep=rep1&amp;type=pdf

[11] BusinessDictionary. (2013). System architecture. Retrieved 13 April 2018, from http://www.businessdictionary.com/definition/system-architecture.html

[12] Deshmukh, Y. (2015). Project Management Life Cycles - Evolution over the years. Retrieved 13 April 2018, from

http://www.simplilearn.com/project-management-life-cycles-evolution-article

[13] GreatPM. (2015). Agile an approach to Software Development. Retrieved 1 April 2018, from https://mukundtechie.wordpress.com/2013/08/24/agile-an-approach-to-software-development/

[14] Heights IT. (2015). Agile Development. Retrieved 10 March 2018, from http://www.heightsit.com/solutions/technology-solution