Class Scheduling System Using Genetic Algorithm

Ericson de Mesa Billedo
Master of Information Systems
University of the Philippines Open University
Los Baños, Laguna, Philippines
ericson.billedo@upou.edu.ph

Abstract

One of the most challenging tasks in an educational institution is the preparation of a timetable. Shinas VET provides vocational training to qualified students to further their education. There are two types of subjects offered, technical and academic. The number of subjects taken by a trainee per term ranges from five to eight. Although electronic spreadsheets can aid in the preparation, the bulk of the processing, analyzing the timetable still lies with the user which this project aims to reverse.

The Vocational Education and Training – Class Scheduling System (VET-CSS) aims to automate the process of creating timetables. By using Genetic Algorithm, a form of artificial intelligence, VET-CSS would be able to generate conflict-free timetables efficiently and on-demand. Moreover, the project will be a web-based application and consequently will provide users convenient access to the most current and updated timetable. This process will, in a way, also reduce the institution expenses as copies of the timetable will be in an electronic format.

The plan is to develop the VET-CSS in six phases, including this conceptualization phase. During the design phase, the client will be consulted as to what preferences they would like to see in the system. Once settled, the site development phase will start. This is where the actual system will be built which will take a little more than a month to complete. The last phase will be devoted to evaluation of the system and provides the client, as well as the developer, an opportunity to submit its last recommendation if there's any.

Keywords: timetabling, scheduling, genetic algorithm

I. THE PROBLEM DOMAIN

Shinas Vocational Education and Training is a small government training Centre that provides vocational training to qualified individuals. It also offers foundation courses required for students to pursue diploma programs in any VET and Colleges of Technology. The main programs it offers are courses in electrical, electronics, Mechatronics, refrigeration and air condition and metal welding and fabrication. Aside from these, students are also required to take academic courses in physics, chemistry, technical drawing, information technology, business and English language.

A. Statement of the Problem

Classes in the Centre starts at 8:00 am and ends usually at 1:50 pm, 2:30 pm would be the latest as this would also be the time for all instructors to leave the Centre. There are instances that classes are scheduled off the regular class time due to limited resources and/or limited timetabling capabilities that would otherwise be possible if enough possible combinations are exhausted or generated.

The Centre is currently following a variant of block scheduling. A class is usually held in a block of two (2) periods of fifty (50) minutes each or equivalent to an hour and forty minutes. Compared to the traditional scheduling (45-50 minutes per class), it is a little easier to build a timetable as the number of meetings is reduced by half. On the other hand, when there is a scheduled holiday, official school activities and the likes, two (2) or even four (4) periods will be missed on that day.

Most workshop classes are in a two (2) block schedule or two (2) hundred minutes. There are also some core academic subjects allotted with this kind of scheduling of which neither have a workshop nor laboratory activities. Some courses, due to its innate nature, could be delivered best on block scheduling and some others may not. Instructors from different departments have different views on this matter. There are those who would prefer block scheduling while others do not.

Only one person is assigned to prepare the timetable, although from time-to-time some others extend a helping hand. The preparation of the timetable is done in a spreadsheet application by mapping out a block of cells with different colors for different courses, layout in rows of time-period against columns of days with corresponding courses (subject classes). Three worksheets are prepared, one for student groups and from here another worksheet will be created for each individual instructor and lastly for room allocation. This is deemed necessary to facilitate detection of conflicting schedules. It would usually take the one in-charge two (2) working days (7:30 – 2:30) to prepare the timetable for 18 groups (sections) of students on the average. The in-charge visually checks the worksheet for conflicts. He admits that although experience helps in the preparation, it is not the most efficient way of doing this.

Designing a system that will tackle the above-mentioned situations will provide a venue to investigate a concept related to artificial intelligence which is genetic algorithm.

B. Background and Objectives of the Project

"The average attention span in 2015 is about 8.25 seconds from 12 seconds in 2000" [1]. In Finland "students' schedules are always different and changing; however, they typically have three to four 75 minute classes a day with several breaks in between" [2]. Having a straight three hours of lecture class may not at all guarantee teaching (and learning) effectiveness. The reverse may also hold true. There are subjects that require ample time to fully understand and that instructors would also be effective. This is the case for subjects that have workshop activities or laboratory exercises. For students to learn well, they would generally require ample time to perform their workshop activities.

Core subjects, or foundation courses on the other hand, are generally theoretical in nature. One of the attributes that made Finland's education system one of the world best is the class length. They usually have 75-minute class which somehow correlates to the average adult attention span theory. They also provide ample of "breaks" in between classes.

The Shinas Vocational Education and Training – Class Scheduling System or the VET-CSS aims to provide efficiency and flexibility in building up effective timetables regardless of principles followed and common constraints accommodated. It will explore scheduling possibilities enough to provide near best possible solution to the administration to maximize further the usage of resources the Centre have.

The following are the objectives of the VET-Class Scheduling System:

- Build a near best possible timetable automatically and efficiently,
- Provide flexibility to incorporate different scheduling types,
- Provide convenient way to access timetable using modern and commonly used UI and
- Reduce cost of reproducing copies of the timetable.

C. Significance and Scope of the Project

It is a high time to upgrade the old and manual process of preparing timetables as one of the thrust of Shinas VET is the continuous development and upgrading of its resources. The VET-CSS will be built on the current and mostly used front-end web framework which will provide an engaging graphical user interface that most users will be familiar with. Another goal of Shinas VET is to provide quality services to students and VET-CSS aims to contribute to this by providing easy and secure access to students and instructors to their recent and updated timetables. Since the system will be developed in-house, the source code will be available to all. The hope is to share knowledge with interested staff (and trainees) to further the staff development program of the Centre.

The proposed system will initially focus on hard constraints in creating a timetable and will gradually explore possibilities to accommodate other soft constraints.

There are seven vocational training Centers across the sultanate. Each may have their own unique cultures. Although this project may also be applicable to others Centers, it will solely be based and implemented in Shinas VET.

D. Existence and Seriousness of the Problem

Complexity in the preparation of timetable differs from one institution to another. The difference may be attributed to the number of course offered, number of groups of students, availability of instructors as well as other resources like rooms and laboratories. Typically, timetable for a small institution with wide resources would be easier to build compared to bigger institutions with very limited resources. Either way, preparing timetable manually is challenging and time-consuming as the burden of processing lies on the shoulders of the one preparing it. The timetable in this Centre is prepared using electronic spreadsheet. One of the drawbacks of using spreadsheets as a plain large table is updating it.

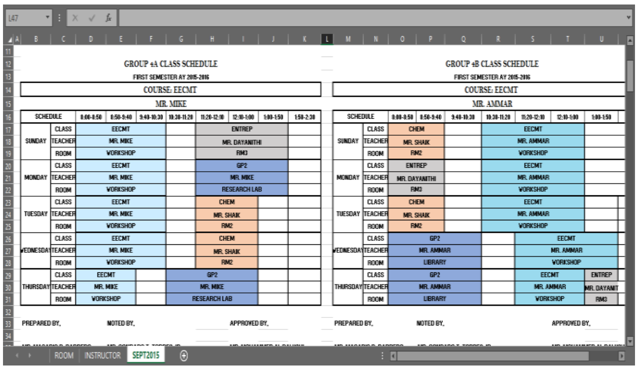

Fig. 1. Timetable Model in Spreadsheet

Whenever changes are made, one must manually reference the changes to other tables in the current worksheet as well as to the other worksheets to reflect the changes made and prevent conflicts which consequently will likely become complicated as more changes are introduced. Other concerns encountered are the following:

- Difficulty in producing a timetable,
- Other possible solutions may not be explored,
- Statistics on resources may require additional processing and may not be readily available,
- Historical information may not be readily available,
- Reproducing copies of the timetable for students and instructors has a cost,
- Notification of changes or updates may not be disseminated timely

II. Review of Existing Alternatives

“A timetable is essentially a schedule which must suit several constraints. Constraints are almost universally employed by people dealing with timetabling problems” [3].

The best solution to the timetabling problem may not be possible or it may be very costly and time consuming to produce. “The school timetabling problem is an NP-complete or NP-hard problem depending on the constraints associated with a specific problem” [4]. The required time to solve such problem “increases very quickly as the size of the problem grows” [5].

“This calls for the use of heuristic algorithms that do not guarantee an optimal solution, but are in many cases able to produce solutions that is "good enough" for practical purposes” [6].

“GAs are becoming popular as a technique for solving optimization problems, mainly because it does not involve sophisticated mathematics and it readily allows for hybridization with domain-dependent heuristics, which result in a more powerful search routine for specific problem” [7].

Genetic Algorithms is modeled on genetics and is part of the adaptive heuristic search algorithm. Being a form of a heuristic search, it is intended to produce near best result. As opposed to a brute force algorithm, GA performs an intelligent exploitation of a random search.

Although partially randomized, GAs are not plainly random because information gathered from previous “generation” is used to better the search.

GA may use some random techniques nevertheless its focus is in historical data and from that makes a better decision thereafter as it continues to search for near best optimal solution.

“Genetic algorithms (GAs) were invented by John Holland in the 1960s and were developed by Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s” [8].

The idea behind GA is found in Darwin’s theory of evolution in which the fittest survive. In this case, only the best ones remain and after some generations only one will survive. That is the fittest or the near best solution to the problem.

In genetic algorithms, individuals or chromosomes (possible solutions) are pooled in a population. The fitness of an individual is evaluated by some criteria against its genes. From this population, the least fit individual gets to be removed. A child (an individual) which comes from fit parent individuals are placed back to the pool. The process (generation) gets to be repeated until the near best solution is found (fittest individual) or the maximum number of generations was reached. The timetabling problem fits exactly into this process. Consequently, the timetable itself can be represented as a chromosome with its subject classes, student groups, instructors, rooms and other resources as its genes. In GA, it is possible to generate several timetables and pool them together. From the pool, the timetable with the most number of conflicts gets to be eliminated.

Crossover is a GA operator that takes two individuals and produces a child. These can be used to produce a child timetable, copying all the genes that are the same in a specific location for both parents to the child timetable.

1	0	0	1	1	1	0	0	1	Parent 1
0	1	0	1	1	1	0	1	0	Parent 2
Crossover									
0	0	0	1	1	1	0	0	1	Child

Fig. 2. - Crossover Operator

This version of the crossover operator ensures that the “good” genes from both parents are kept intact in the child timetable. We refer to a “good” gene in a timetable as a schedule slot that has no conflicts. Another important operator in GA is mutation. Mutation prevents the population from staying within local optima and not reaching the global optima or the near best timetable. Local optima happen after several generations and the children produced thereafter are the same as before.

1	0	0	1	1	1	0	0	1	Parent 1
0	1	0	1	1	1	0	1	0	Parent 2
Mutation									
0	0	0	1	1	1	0	0	1	Child

Fig. 3. Mutation Operator

With mutation, the child produced will have a higher probability to be different from generation to generation. Mutation makes it possible for the population to have a fresh and different offspring in every generation. The flowchart below describes the basic flow of operation in GA.

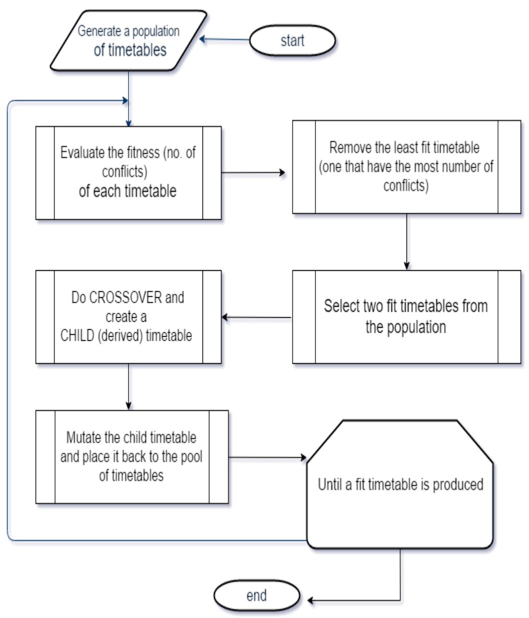

Fig. 4. Genetic Algorithm applied in timetabling

Simulated Annealing and Tabu Search are part of the meta-heuristic search methods in the field of AI that may also be used to solve timetabling problem. However, this project is set to use a purely genetic algorithm, but is open to explore other methods in the future.

III. Project Approaches

A. *Theoretical Framework*

Generating timetable for an educational institution is a challenging task because resources are scarce and that constraint need to be accommodated. Being so, when one finally came into solving the puzzle, efforts to further find a more suitable solution stops. The effort exerted in finding the solution is overwhelmingly huge to still search for the near optimal one.

A timetable can be derived in several possibilities which manual effort of doing it would likely be difficult. Using a brute force algorithm on the other hand is very costly in terms of computing resources, that is, it will take longer than usual and may not be able to produce the solution needed. Flipping a coin and getting a head has a 1 out of 2 probabilities. Getting combination of a head then tail after tossing a coin twice will 1/2 times 1/2 or one out of four. This is just a coin and timetabling possess wider number combinations. There have been a lot of studies conducted on the use of AI in solving timetabling problem. In this project, timetable resources like rooms, subjects, teachers, sections, subject classes as well as GA operators will be represented as OOP classes. Using classes allows components to be treated as objects. They will have attributes and methods to manipulate them as well as other objects efficiently.

Subject-class object will be composed of a teacher, subject, trainee group (section) and room objects. Suppose a trainee group "A" has a three-period Biology class (SubjectID 101) twice a week.

Subject ClassesID	SubjectID	Trainee GroupID	TeacherID
0	101	201	301
1	101	201	301
2	103	201	303
3	104	201	304

Fig. 5. Subject-class object

This will be represented by a unique two subject-class objects (subject-class 0 & 1) each of which is allotted a timeslot on a given day. All schedule objects, which belong to a specific timetable for a certain academic term in an academic year, will be grouped together into an associative array.

B. *Rationale for the framework*

There are several approaches in solving timetabling problems and that "it is observed that Genetic Algorithm is good solution technique for solving such problems".

The center of operation in GA is in its chromosomes or "individuals" with its underlying genes. The chromosome represented by a timetable object is composed of other related objects that forms its genes. GA is designed to manipulate genes then later checks the fitness of its chromosome in the population. The genes of the timetable chromosome are represented by the teacher, subject, trainee group (section), room, subject-class and schedule objects, each of which should have a provision to allow it to be modified and produce specific data when needed. Sample code in PHP representing a timetable object accessing its fitness value is provided below.

```php
for($x = 0; $x < 50; $x++){
    $timetable[$x] = createTimetables ($x);
    foreach($timetable[$x] as $key => $value){
        $timetableFitness[] = ($value->GetScheduleTimetableID()->GetTimetableFitness());
    }
}
```

Fig. 6. PHP code to access fitness value

The fitness value in this project represents the correctness of the timetable with the value zero representing no conflicts in schedules. With this representation accessing "key" value during the implementation of this project will be efficient if not easier compared to representing it with other type of data structure. The initial attempt to this project, including this paper and source code, can be found in this repository: https://github.com/ercson-R8/vet-css.git.

C. *Technologies*

The proposed system will be a web-based application. It should allow access from any mobile devices featuring up-to-date web browsers. Since it will be web-based, HTML will be used. To enhance its interface, the Bootstrap front-end framework will also be utilized as this will ensure that the user interface that will be used in developing the system will be modern and most users will be familiar with as the looks and feel of the interface will be similar to applications like Twitter, Facebook and Google products. This will also allow users to have the same experience when using their PC or their mobile devices such as mobile phones and tablets on virtually all kinds of operating systems.

Another technology that will be needed in this project is a relational database management system. MariaDB a community variant of MySQL with InnoDB as its storage engine will suffice the requirement for this project. MySQL, being the most popular database system, is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. MySQL is the de-facto standard database system for websites with HUGE volumes of both data and end-users (like Facebook, Twitter, and Wikipedia).

The server-side scripting language of choice will be PHP (PHP: Hypertext Preprocessor). The Genetic Algorithm will be implemented in this language as classes. Moreover, this structure will provide a relatively compact and effective way of manipulating its attributes.

Lastly, the system will be adopting the MVC (Model-Control-View) framework. This framework will compartmentalize the data from the application and from the front-end user display or view. It will allow programmers to direct its effort and focus on a single task rather than working on all of them at once. Coding and debugging will be easier when programmers can focus on a single piece of the application. In addition, template engine will also be utilized to maximize the MVC framework. Other plugins like JQuery might also be utilized to enhance the overall user experience.

IV. Project Plan

A. *Concept*

The VET-Class Scheduling System will be designed around the concept of a modern, effective yet simple to use web application. The thrust of this project is to make creation of timetable effective and updated reports are accessible to its users. The system will allow access to its timetable from virtually any type of devices, accommodating a number of web browsers. Tentative expansion plan for the project is to allow push notification. With push notification, users will be able to receive updates whenever there is a change in the timetable.

The VET-CSS will allow build-up of timetable resources such as teacher information as well as for the subjects, rooms and trainee groups. This build-up of resources will allow an effective way of creating subject-classes in which the timetable process will try to schedule following the constraints provided.

There are two types of constraints in this project, hard and soft. The hard constraints are rules that must be followed, such as a room can only have one subject class at any given period. Likewise, a trainee group may have many subject classes however it can only have one subject class at any given period. Soft constraints are preferences that if not followed may not necessarily cause conflicts in schedules. An example of this is a teacher requesting to have classes all the morning slots so he can then be "free" in the afternoon.

This project will require at least two developers, one on each end of the system side. Nonetheless with the help of frameworks, both front and back-end, this project would still be doable with just one developer. As such, careful analysis of the best and current practices must be considered.

1. Data Requirements

   The VET-CSS, as with most systems, will require data to be able to produce the desired result. Below is a data flow diagram that shows the overview of how data passes throughout the system.

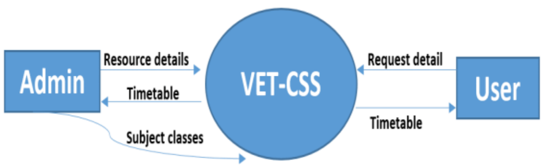

Fig. 7. Context - Data Flow Diagram

The DFD-0 above shows that data (resources) will only come from one source, the administration. The user (the students, instructors and staff) will be on the receiving end of the system. The DFD-1 below elaborate further the DFD-0 and presents where all the data will be stored.

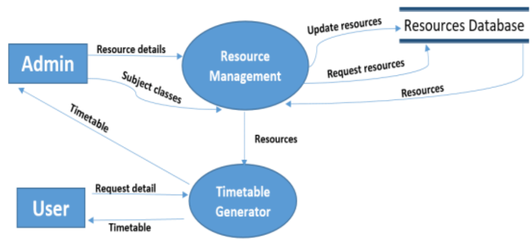

Fig. 8. Data Flow Diagram Level 1

The "Resource Management" process is further divided into two processes. DFD level 2 below shows which process handles the two types of data coming from the "Admin" entity.

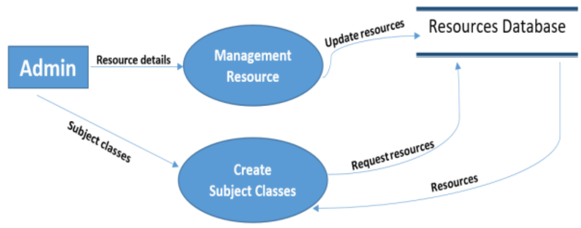

Fig. 9. Data Flow Diagram Level 2

One of the key feature of this system is its ability to provide historical data and as such it will require a database. Identified below are the sources of data in an EER-D model for this system to be able to generate timetables.

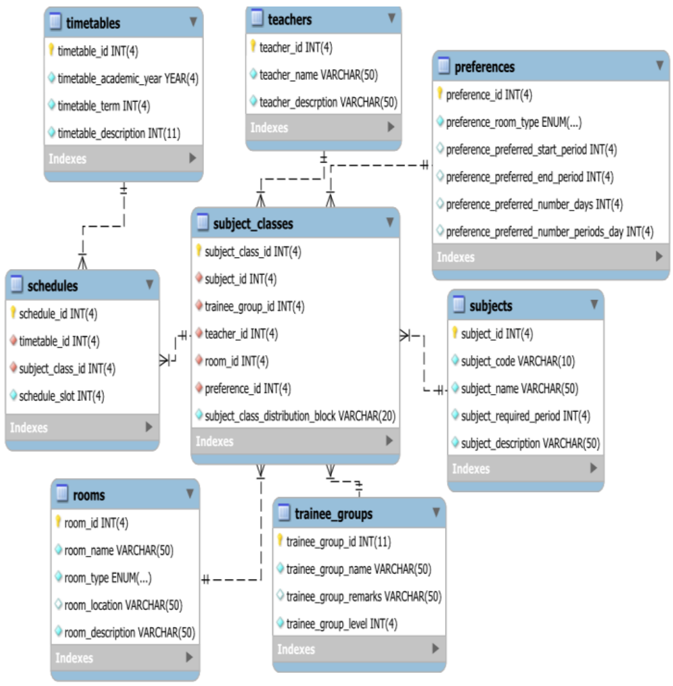

Fig. 10. Enhanced Entity Relationship Diagram

The EER-D above also shows the cardinality between entities. A teacher may be able to teach many subject classes but a subject class in this scenario will only have one teacher. Likewise, a room can be used by different classes but a class can use one room at a time.

2. Control Flow

The control flow diagram below shows how the system would be able to generate timetable.

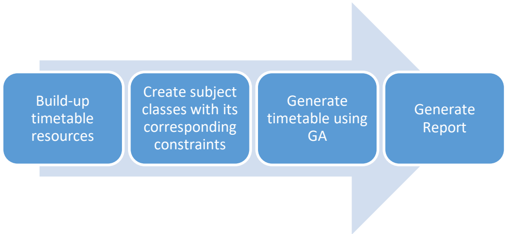

Fig. 11. Control Flow

3. System Interface

The UI is a web-based using HTML5 standards and Bootstrap enhanced pages which will provide a modern look and feel that is responsive to user's device. It will consist of a login page, a main page with common top "nav-bar", sidebar and footer, "manage resources" pages and modals for adding, edit, deleting and viewing resources. Below is a sample layout of how the system will look like.

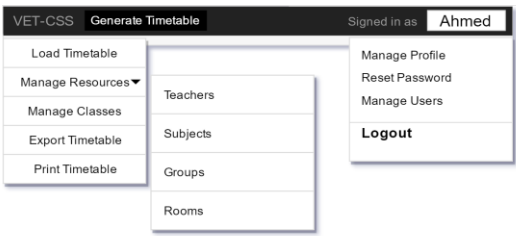

Fig. 12. Initial page layout

The timetable will be generated like the one presented below. The generated timetable will occupy the center of site main page.

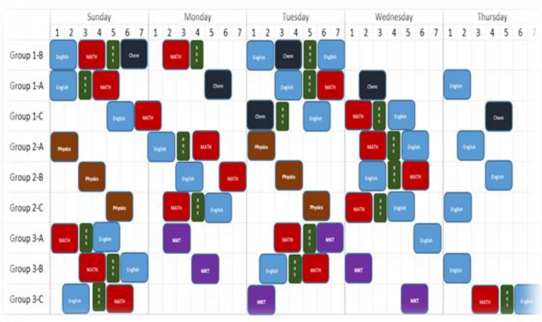

Fig. 13. Initial generated timetable layout

4. Software Requirements Specification

The VET-CSS aims to upgrade and the old, manual process of generating timetables in Shinas VET. It envisioned to provide ease of access to the current and up-to-date timetable to students and teachers alike and is accessible to most devices using modern browsers. This section describes the requirements and functions to be performed by the system.

TABLE I REQUIREMENT SPECIFICATIONS

ID	Requirement Description
REQ-01	The system shall authenticate users requesting access to the system
REQ-02	The system shall allow the system administrator to add users and designate appropriate system usage rights.
REQ-03	The system shall allow users tagged as "admin" to add resources and subject classes with details following standard a format.
REQ-04	The system shall allow all types of users to view current timetable.
REQ-05	The system shall allow all types of users to export current timetable.
REQ-06	The system shall allow all types of users to print the current timetable.
REQ-07	The system shall allow "admin" to set timetable to be the current one.
REQ-08	The system shall be able to generate conflict free timetable following constraints specified.

The table above provides a brief description of the initial client's requirement. The list may change during production of the system. The list will also be the basis for creating the "Use Case" for the system. The "Use Case" suite for the VET-CSS is presented below.

TABLE II USE CASE SUITE

UC-01	Login, *from REQ1*
UC-02	Add user, *from REQ-02*
UC-03	Manage Resources, *from REQ-03*
UC-04	Add Subject Class, *from REQ-03*
UC-05	View Timetable, *from REQ-04*
UC-06	Export Timetable, *from REQ-05*
UC-07	Print Timetable, from *REQ-06*
UC-08	Load Timetable, from *REQ-07*
UC-09	Generate Timetable, from *REQ-08*

The "use case" suite above will be the basis of the features the system will have. The initial features the system will be sporting may also evolve during the entire production phase of the system.

TABLE III FEATURE SUITE

User Account Management

F-000: User Login

F-001: Add User

F-002: Reset Password
F-003: Edit User Profile
F-004: Delete User

Resource Management

F-100: Add Teacher
F-101: Modify (edit/delete) Teacher
F-102: Add Subject
F-103: Modify (edit/delete) Subject
F-104: Add Group
F-105: Modify (edit/delete) Group
F-106: Add Room

Subject Class Management

F-200: Add Subject Class
F-201: Modify (edit/delete) Subject Class
F-202: Select Preferences

Timetable Management

F-300: Generate Timetable
F-301: Load Timetable
F-302: Export to CSV
F-302: Print Timetable
F-304: Filter Timetable (By teacher, group, room etc.)

5. Environmental Requirements

SYSTEM HARDWARE REQUIREMENTS

The VET-CSS is a web-based application and as such would require a web server. The client may be requested to produce a standard PC with an updated OS to act as a web server or the site may also be hosted online using a free web hosting site for testing purposes. A networking infrastructure (wire, wireless or a combination) is also required to access the system. Aside from these, a stable internet connection is also necessary since some plugins and application frameworks will be fetched from CDNs. In this way when an update is available to these plugins, the client won't need to do anything as they will automatically be using the latest version of the plugins. However, the plugins may also be stored on the web server side by side with the system itself to optimize the VET-CSS performance.

SYSTEM SOFTWARE REQUIREMENTS

Web-based application requires a web server, be it on the local network or on the internet. The system will be developed using Apache web server with PHP 5.6.x as its server-side scripting language and MariaDB SQL.

Since the front-end framework of the application will be using Bootstrap, users will be able to access the system using any modern browser on any device. However, the system will be developed and tested in the Chrome browser.

B. *Methods*

This project was one of three proposed projects for this course. The others being inventory system and faculty evaluation system. These systems have the potential help realize the Centre's goal, as define in SWOT analysis, in providing an excellent work environment and services for all employees as well as providing quality services for students.

In line with this, careful analysis of current and existing model as well as general procedures and preferences when preparing the timetable must be considered. The author is very pleased when the project was presented to the concerned department as they welcome the project and assured technical support whenever needed.

The next phase of the project will be on the actual production of the system. During the production phase, there will still be consultation with the client as to the overall theme and layout of the site as well as the timetable itself.

C. *Plan for user testing and project assessment*

VET-CSS main goal is to be able to generate enough number of possible timetables that are free of conflicts. It will take into consideration the hard constraints. The hard constraints are most important consideration that otherwise would create conflicts in schedules. Upgrading the system to take soft constraint will follow.

On the last two phases of this project, user testing will commence. The focus of this will be the evaluation and the final touch to the system. This will be done with the use of System Usability Scale or SUS. "SUS is not diagnostic - its use is in classifying the ease of use of the site, application or environment being tested" [9]. The survey will be hosted on an online site so users of the system will be able to access it anytime, anywhere. SUS scores have a range of 0 to 100. An SUS score above a 68 is considered above average and anything below 68 is below average. The result of this survey will lead the developer to further improve the system before the full handover to the client takes place.

D. *Schedule*

This project is scheduled in five phases. Except phase three, all the other phases of the development will last from five to nine days. Phase three will run a little more than a month. The total duration of the entire project is a little more than two months.

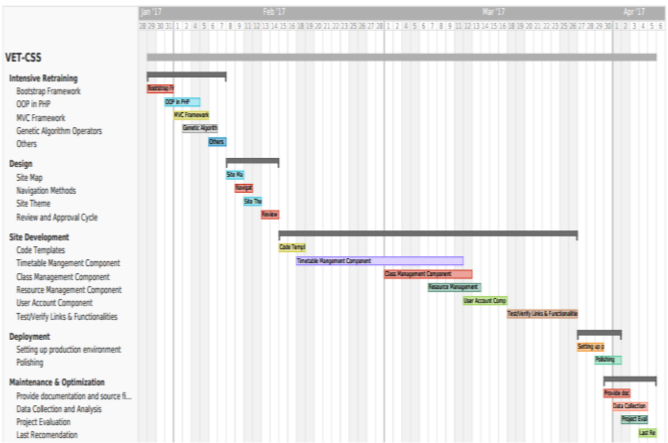

TABLE IV PROJECT GANTT CHART

Acknowledgment

I would like to acknowledge and thank all my previous and current professors here at OU. Their valuable insights, technical expertise and guidance helped me shape and complete this paper. I would also like to express my gratitude to my classmates in the past and in this course who have shared their opinions, knowledge, relevant work experiences in discussion boards and in group projects. To my family who is my source of inspiration and provided me much needed moral support, thank you.

Above all, I would like to thank our Lord for the continued blessings and the good health received throughout my stay here in OU making it possible for me to complete all of what is required in every semester.

References

[1] "Attention Span Statistics," Statistic Brain Research Institute, 2 July 2016. [Online]. Available: http://www.statisticbrain.com/attention-span-statistics/. [Accessed 15 November 2016].

[2] K. Day, "11 Ways Finland's Education System Shows Us that “Less is More”.," 15 April 2015. [Online]. Available: https://fillingmymap.com/2015/04/15/11-ways-finlands-education-system-shows-us-that-less-is-more/. [Accessed 15 November 2016].

[3] P. R. Edmund Burke, Practice and Theory of Automated Timetabling, Edinburgh, U.K.: Springer-Verlag Berlin Heidelberg, 1996.

[4] N. Pillay, "A survey of school timetabling research," Springer Science+Business Media , New York, 2013.

[5] O. Chohan, "University Scheduling using Genetic Algorithm," Department of Computer Engineering, Dalarna University, Borlänge, Sweden, 2009.

[6] T. H. V. O. M. Č. D. J. Zlatko Bratković, "University Course Timetabling with Genetic," University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia.

[7] K.-S. S. Enzhe Yu, "A Genetic Algorithm for a University Weekly Courses Timetabling Problem," Department of Industrial Engineering, Seoul National University, Seoul, South Korea, 2002.

[8] M. Mitchell, An Introduction to Genetic Algorithm, Cambrigde, Massachusetts: First MIT Press, 1998.

[9] "System Usability Scale (SUS)," U.S. Department of Health & Human Services, [Online]. Available: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html. [Accessed 16 November 20016].

[10] R. E. O. Roxas, Software Engineering, CP Garcia Avenue, Diliman, Quezon City: UP Open University, 1998.