UNIVERSITY CLASS TIMETABLE TOOL

BY

ANG WEN JIE

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2022

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ____UNIVERSITY CLASS TIMETABLE TOOL____

Academic Session: ____202205___

I _____ANG WEN JIE_____

(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:

__NO 15, JALAN KERETAPI_

34650 KUALA SEPETANG

_PERAK_____

Ku Chin Soon

Supervisor's name

Date: _____06/09/2022_____

Date: 07/09/2022

FACULTY/INSTITUTE* OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: ____06/09/2022_____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that _____***Ang Wen Jie***_____ (ID No: __***19ACB00369***__) has completed this final year project/ dissertation/ thesis* entitled "___*University Class Timetable Tool*_____" under the supervision of _____Ts Dr Ku Chin Soon_____ (Supervisor) from the Department of ___Computer Science____, Faculty/Institute* of __Information and Communication Technology__ , and __Ts Wong Chee Siang_ (Co-Supervisor)* from the Department of ___Computer Science_____, Faculty/Institute* of ____ Information and Communication Technology __.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

DECLARATION OF ORIGINALITY

I declare that this report entitled "**UNIVERSITY CLASS TIMETABLE TOOL**" 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 : _____Ang Wen Jie_____

Date : _____06/09/2022_____

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Ts Dr Ku Chin Soon, who has given me this bright opportunity to engage in a timetable scheduling project. It is my first step to establish a career in scheduling field. A million thanks to you.

Finally, I must say thanks to my parents, my family, and friends for their love, support, and continuous encouragement throughout the course.

ABSTRACT

This project is a timetable scheduling project for purpose of generating the timetable automatically. It will provide the students with the concepts, structures, and design of an automated timetabling tool using Genetic Algorithm. University class timetabling is an optimization problem where sets of the class events need to be scheduled into the timeslots available and at the same time, consider the existence of the constraints. Due to its NP level of difficulty, every added constraint into the timetable scheduling will be resulting in the exponential increment of its complexity. To solve this problem, many algorithms or optimizers had been introduced to simplify the problem complexity, as well as find out the optimal solution for the scheduling. Example of the algorithms are genetic algorithm, particle swarm optimization, simulated annealing, and some other heuristic approaches. As there are different kind of constraints in different institutes or universities, different methods need to be defined to obey the constraints. During the pandemic of Covid-19, the timetable structure has changed greatly, significantly the changes of the study mode which is from physical study mode to online or hybrid mode. In this project, the newly constraints related to the study modes in the timetable scheduling will be defined, genetic algorithm will be studied and used to solve the timetabling problem. Then, create a university class timetabling tool with the newly added constraint. In the end of this project, a university class timetable tool will be developed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1: Project Inspiration

University class timetable tool is referred to as a scheduling tool to aid the university course timetabling. Timetable scheduling is a necessity to ensure the smooth and effective operations of an institute [1]. For academic bodies, timetable scheduling is concerned to be an essential requirement or a must to do. It has been the common problem that worries nearly all universities in the world for decades and its problem arises every semester of the year. Due to the changes of constraints in every academic session, either by manually, or using timetabling tools, the timetabling process will require a lot of rescheduling and will be consuming plenty of time to create a non-conflict timetable.

The timetable scheduling problem is a Non-Polynomial (NP-hard) problem since decades ago, and this problem does not have a particular solution. Or in another word, there is no specific way or algorithm to handle each, and every timetable scheduling process. Plus, the computation works required to find the optimal solution will be increasing exponentially as the constraints amount increases [2]. To look for the optimal solution in the timetabling process, there were plenty of optimization algorithms introduced by researchers from the past few decades. Examples of algorithms introduced are the Genetic algorithm, Particle Swarm Optimization (PSO), Backtracking algorithm, Constraints Satisfaction Programming (CSP) approach, Linear integer model, Tabu search algorithm, etc.

Although there were many algorithms introduced to solve the problem of timetable scheduling, most of these algorithms do not work individually, but interacting or working with each other, combining ideas from different fields to get a wider angle of views in developing an optimal solution. This action of engaging one algorithm to another is called Hyper-heuristics, which objective is to find the solutions using many different ways provided [3].

Always, the timetabling problems are working around a set of rules/constraints which are categorized into the hard constraints and the soft constraints. Hard constraints

are always referring to the set of rules that must be fulfilled to produce the correct and feasible timetable. One example of the hard constraints is “a lecturer can only teach one class at a time”. Whilst the soft constraints define a set of rules which may determine the effectiveness or optimization of the timetable [4]. One example of soft constraints is “the latest class in a day should not be later than 6:00 pm”. In a university, the parameters of the constraints that are mostly to take into considerations include classes, lecturers, courses, classrooms, and students. The relationships between these parameters and the rules applied on these relationships will make up the constraints to be followed in scheduling the timetable. Based on different scenarios and requirements of the timetable’s users, the constraints may be changed from time to time. The outcome of the timetabling process should be the timetable that meets all the requirements in hard constraints and is optimized to follow the soft constraints.

Constraints are important in varying the difficulty and complexity of the timetabling problem. Because of the different complexity of the constraints, the effectiveness of the timetabling algorithm and the time of the deliverable of the algorithms may vary tremendously. Besides the complexity of the constraints, the differences between the constraints in different universities are the causes of there has no optimal scheduling algorithm exists to suit the different requirements of different universities. In this circumstance, it is known that timetabling problem relies greatly on the constraints and its uncertain manipulations.

During the pandemic of Covid-19 in the December of 2019, and the first MCO on the 18March 2020 in Malaysia, the common structure of the teaching and learning process had been changed greatly. Most of the university has changed their study mode from physical to online mode. As mentioned above, the constraints will be changed due to different scenarios, thus the transformation of physical teaching and learning mode to online teaching and learning mode had varied the commonly defined constraints to meet the new requirements in the new situation. Hence, this new constraint will be studied well and included in the timetable scheduling process to develop a better timetabling tool that can fit nicer to the current global situation. In this situation, the timetable scheduling structure needs to be replanned and redesigned. For example, the hybrid study mode may lessen the usage of the classrooms.

1.2 Problem Statements

The first problem of timetabling is that **the existing solutions are not able to generate online mode class timetable**. During the pandemic of Covid-19, most of the universities and institutes have changed the teaching and learning mode from the physical mode to online mode. This situation has created the new constraints to be considered when generating the timetables. For example, the constraint which stated that what online platform should be used to conduct the online classes. This has increased the difficulties and works of the timetable scheduler to consider the extra constraint. In this project, one of the problem statements is to identify the hard and soft constraints that existed result from the new teaching and learning mode wisely, and apply the scheduling algorithm to them, obeying the regulations, so that a feasible and optimal timetable will be produced.

After that, the next problem of the university timetabling algorithm is **there is no alternatives to deal with the different modes of courses in the timetable**. For example, physical mode, online mode. This means that for a course, there may have multiple mode of study. Among them are physical, online, or hybrid where hybrid means online, and physical mode are being conducted together or concurrently. For example, a lab class may be divided into two groups of students where one group of students may attend that class physically, while other attend the class through online platform.

1.3 Objectives

In this section, the objectives of this paper will be discussed. The first objective of this project is **to investigate the hard constraints and the soft constraints for the university class timetabling problems**. As discussed in chapter 1.1, constraints are the set of regulations that the timetable scheduling tools should workaround and obey to develop a feasible and optimal timetable. Thus, before we can design a timetabling tool, the first thing to do is to understand and identify the constraints that existed in a university. In this project, one parameter to be included in the constraints' identification will be the study modes which are physical mode and online mode.

After defining the constraints, the second objective will be **to design a university class timetabling tool that will help the users to generate the timetable with hybrid class automatically using the genetic algorithm**. Typically, the physical university will be using days to arrange the timeslots for the timetable creations before each academic semester. This process will require lots of time and human resources and cannot get the optimal output most of the time. Therefore, this project aims to develop a software timetable scheduling tool that can use the power of technology and the help of mathematics algorithms to automate the process of timetabling, The current existing similar systems will be studied, and the functional modules of similar tools existed will be implemented and enhanced in this project. The current existing timetabling system such as aSc Timetables [5], FET [6], etc. will be reviewed in this project, their functionalities will be studied and innovated to suit the current constraints. Besides, the genetic algorithm used to schedule the timetable will be integrated with the newly identified constraints to fulfil the new requirements of the scenario.

To sum up, the objectives of this project are to investigate the hard and soft constraints for the university class timetabling problems, then to design a university class timetabling tool that will help the users to generate the timetable with hybrid class automatically using the genetic algorithm.

1.4 Project Scope

This project is to develop a university class timetable tool that can help in the timetable scheduling of the university. In this project, the university involved is University Tunku Abdul Rahman (UTAR) and the faculty involved to help in the validations and verifications of the timetabling tool is Faculty of Information and Communication Technology (FICT). The target users of this timetabling tool are the university management team who are responsible to schedule the timetable before every beginning of each trimester. After that, it will only consider when students follow the program structure assigned to them. Then, the timetabling tool will be developed as a standalone application system.

The focus of the functionalities of the timetabling tool is its abilities to perform automated timetable scheduling based on the input of the attributes that make up the

timetable which are courses, lecturers, classrooms, and students. This timetabling tool will be allowing the users to key in the variables listed above into the database, then based on these variables, generate a non-conflict timetable to be used for future purposes. In this circumstance, the Genetic Algorithm will be used to generate the timetable.

Besides, one special feature of this tool is the inclusiveness of the mode of study. Among the modes of study are online teaching and learning mode (OTL) and physical mode. Users will be having more options to choose whether the classes should be conducted in OTL mode or physical mode. This is due to the emergence of the needs of dual-mode of study since the outbreak of Covid-19 pandemics.

The hardware to be used for the development of timetabling tool in this project involve a laptop with moderate and enough processing power. For the software, Eclipse IDE for Java Developer will be used for the coding stuffs. After that, the programming language used include Java language, with xml as the markup language for the webpage appearances. For the storage technology, MySQL will be used to store the data required to generate the timetables, as well as the data of the users' accounts.

1.5: Project impact and contribution

In this project, the main contribution is to develop a timetable scheduling tool that can work more effectively and efficiently to help reduce the workloads of the academic staff. For example, by developing the data retrieval functionality from database in this project, the workloads of the staff in inserting the real-life data will be greatly reduced.

Besides, other than the commonly considered constraints, a new field of constraints was added into the consideration in this project. The study mode is the newly added constraint that can help the staff to generate the timetable that fits the current practical matter. With this constraint added, the normal structure of the constraints will be varied. For example, an online study platform reduces the competes of the resources such as the classrooms. This will be one potential value that may help in the future

timetable scheduling process as the hybrid study mode is becoming more and more popular around the world.

Furthermore, parallel processing will be adapted in this project to allow the faster timetable generation. The parallel processing will be implemented to check the different constraints in parallel.

Finally, this project will introduce new functions that implements and checks the new constraints. This can lead to a better timetable allocation. The better allocation of the timetable by the timetable scheduling tool developed in this project will reduce the crashes of the resources. For example, crashes of the usage of the classrooms, crashes of students' subjects assigned in the same timeslot, and crashes of two subjects taught by the same lecturer at a single timeslot. After a deeper constraints analysis and take them into the considerations of timetable scheduling, the crashes of the resources will be prevented.

1.6 Chapter summary

In this proposal, the distribution of the contents will be as follow. The introduction which consists of project inspiration, problem statements, project objectives, project scope, and project impact and contribution will be discussed in chapter 1.

Whilst chapter 2 will be discussing the different algorithm approaches proposed by other researchers in previous papers. After reviewing the past research, their strengths, limitations, and proposed solutions will be discussed. The review of some similar systems will be studied in this chapter too.

In chapter 3, the system methodology will be discussed. The process of project development, project verification plans, and system functionalities will be discussed in this chapter.

Furthermore, the system architecture design, system interface design, data storage design, and hardware, software requirement will be introduced in chapter 4.

In chapter 5, the test plan introduced in chapter 3 will be tested, the expected results together with the actual results of the testing will be investigated and analysed.

After that, the discussion on the system performance in achieving the objectives, system limitations, and future enhancement/improvement will be discussed in chapter 6.

Lastly, conclusion will be made.

CHAPTER 2

Literature Reviews

2.1 Overview of the Past Research

The timetable scheduling is the process of scheduling something to the set of resources. The university class timetabling is the process of assigning the attributes such as subjects, lecturers and student groups into the limited number of resources such as timeslots and classrooms [7]. Timetable scheduling is often being considered as a common scheduling problem. And the aim of the timetable scheduling is always to minimize the resources used.

The challenge of timetabling is that it needs to allocate the limited locational and the temporal resources to a series of events where this process needs to be done under a very huge number of limitations [8]. After that, to produce a feasible and optimal timetable, the timetable scheduling process needs to follow a set of constraints which are the rules to control the correctness and the optimal level of the solutions.

Furthermore, as the timetable scheduling problem is an NP problem, one more constraint added into the consideration of the timetable schedule will be resulting in the exponential increase of the difficulties and computation resources [9]. This problem have challenged the pass researchers for decades and until now, there is no any universal algorithm that suit for all the given constraints in timetable scheduling.

Before the computational algorithm had been introduced, the timetable scheduling was being done manually by the management team and this process may require too many days of working with the timetable, and usually, the output of the manual timetabling is not efficient and do not make use of the resources wisely [10]. Hence, to solve the difficulties of timetable scheduling, several approaches had been introduced by many researchers for decades. For example, several algorithms such as genetic algorithm [2], [4] [11], [12], [13], Constraints Satisfaction Programming (CSP) approach [13], Linear integer model[14], Tabu search algorithm [15], [16], the Particle Swarm Optimization (PSO) [17], [18], the Simulated Annealing (SA) [1], Ant Colony Optimization (ACO) [19], etc. Among these algorithms, the genetic algorithm is the most popular and used by many researchers to perform a heuristic approach, which is

an approach to combine one algorithm with one or more other algorithms to get higher performance on the result.

In this chapter, several algorithms introduced before will be discussed, the hard and soft constraints in those papers will be evaluated, and some similar products will be reviewed to get a deeper understanding of the timetabling tool development.

2.2 Timetable scheduling techniques

2.2.1: Genetic algorithm

Originally, a genetic algorithm is an algorithm used in biological fields. However, its powerful function allows it to be used also in many other fields such as the mechanics of natural genetics, scheduling, and solving optimization problems. Also, genetic algorithm is widely used in solving the complexity of timetable scheduling. For example, [12], [2], [4], and [13] were implementing genetic algorithm to solve the timetabling problems.

The genetic algorithm is having some differences compared to other algorithms proposed. For example, a genetic algorithm does not work on the parameters, but it works on the coding of the parameters. Besides, the genetic algorithm will explore the search space, and this will be resulting in a higher probability of finding the more enhanced performance of the solution [20].

The genetic algorithm is made up of six phases which initialization, fitting, selection, crossover, and mutation. Before the first phase of the genetic algorithm, we might need to encode the gene. In gene encoding, the gene will be represented as a parameter (timeslot) in the teaching and learning session, and each of them is a string made up of two parts. For example, a subject called X and a room called Y will be making up a string called XY. The result of this step is the genes created are put into a set, then combined to produce a chromosome [2].

Then, the initialization phase starts. In this phase, the gene encoded and produced in the previous step is joined up to produce the chromosomes. Then, multiple

chromosomes generate a population randomly where the population. The illustrations of the gene, chromosomes, and population are shown in the figure 2.2.1.1 below [21].

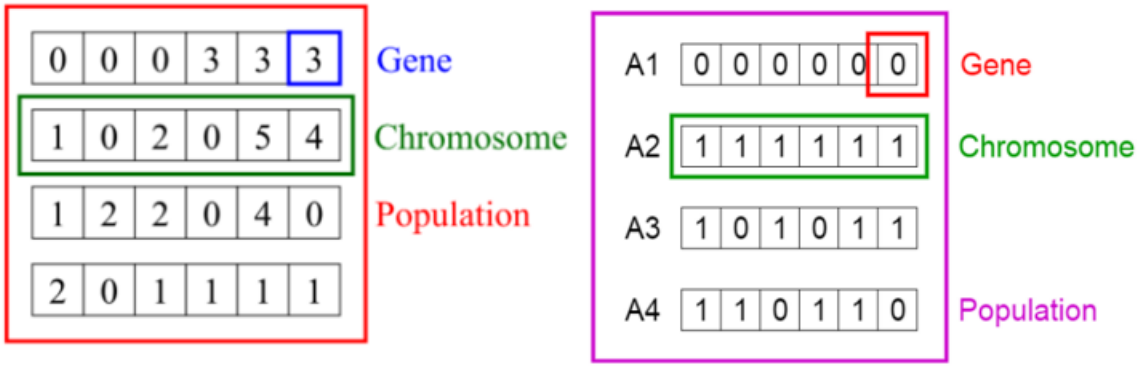


Figure 2.2.1.1 Illustration of genes, chromosomes, and population

After that, in the fitting phase, a fitness function which is the problem the genetic algorithm wants to solve is produced. This fitness function is used to test how well will the potential solution fit the circumstances. The fitness test of each chromosome will be evaluated in the fitness score and based on the score. Once the scores of the chromosomes are evaluated, the tool can select the chromosomes which have the good fitting result in the selection phase [22]. The main purpose of this selection phase is to find more chances to get the best solution instead of the only functionable solution.

Followed by the crossover phase. Three types of crossovers can be implemented in this phase. Among them are modified uniform crossover (MUX), uniform crossover (UX), and single-point crossover (SX) [23]. The random point will be selected to match up a pair of parent chromosomes to produce the offspring. In the production of the offspring, the genes in the parent chromosomes will be exchanged until the point of the crossover is reached.

The last phase will be the mutation where this phase is conducted to avoid the production of offspring that are similar to the parent chromosomes. By randomly changing the value of the characteristics in the offspring, mutation can be done. The figure 2.2.1.2 below shows the flowchart of the phases in genetic algorithm [24].

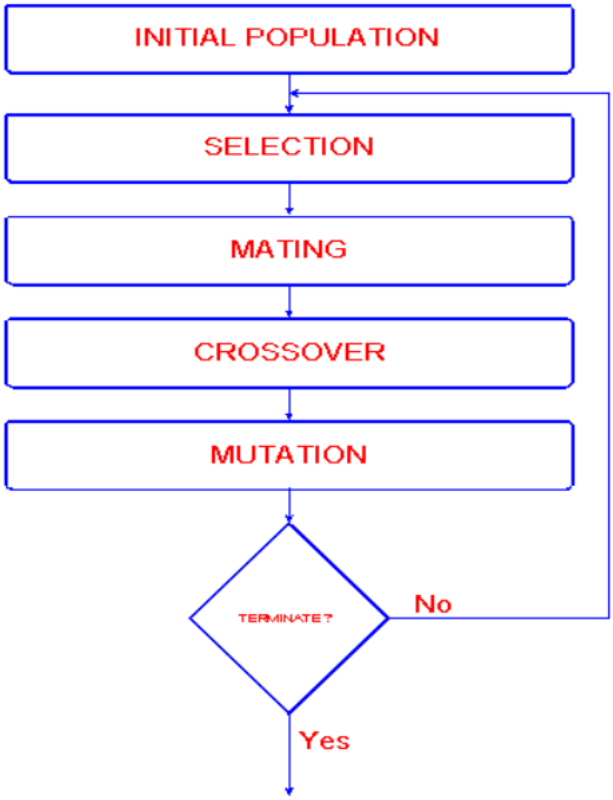


Figure 2.2.1.2 Overall flowchart of the genetic algorithm

Genetic algorithm works in parallel [25]. Comparing to other algorithms, genetic algorithm is useful in calculating the fitness of the individuals in an independent manner. Thus, it can be said that genetic algorithm is able to evaluate a large group of individuals hence can approach the optimal solutions in a more effective way.

Genetic algorithm can handle multiple parameters [26]. Compared to most of traditional algorithms which can only handle single parameter, genetic algorithm can work with multiple parameters. Most real-world problems need to be described in the terms of multiple constraints and objectives, in this circumstance, the genetic algorithm is able to resolve the problems by creating the solutions that suit for each constraint instead of exploiting one and improve another. Each constraint will be given consideration to be optimized, then based on users preferable and fitness value, the global optimal solution is likely to be produced.

Genetic algorithm looks for solutions in multiple directions in a given time [26]. By working with the fitness functions, the genetic algorithm can evaluate the

optimization values of the solutions produced. For example, its parallel feature allows it to evaluate many different parameters at a time and define their fitness values. Thus, enabling it to achieve the optimal solution easier compared to the traditional algorithms which are mostly solve the constraints in a serial manner.

Genetic algorithm requires the special definition for the problems [25]. The real-world problems to be solved using genetic algorithm needs to be described in the way that can bear with the random alternations during the scheduling process. To define the problems, often, real-valued list of numbers, binary, or integer are used.

After that, fitness function selection is a bottleneck of genetic algorithm [26]. To get a good and functionable fitness function, the selections of the methods in the phases (e.g., crossover and mutation phases) need to be taken care. After that, the wrong results might be happened if the fitness function is not well defined.

The next weakness of the genetic algorithm is that when small population are taken care, the local optimum might be obtained. To explain this, when an individual with the better fitness value is selected for reproduction, it may massively deprive the chances of other individual in the population to get proceed [26]. As a result, the local optimum result will be got but the global optimum might never be achieved.

Lastly, the last weakness of genetic algorithm will be its high time and computational complexity.

2.2.2 Backtracking Algorithm

Constraints Satisfaction Programming problem is the problem of assigning the values from the domains to the variables and at the same time following the constraints defined. A good solution for this is to assign the values to the variables in a consistent way. This problem is like the timetable scheduling problem as timetabling requires the assignment of the values (subject, students, etc.) to the variables (resources – timeslots, classrooms, etc.) following a set of regulations (hard and soft constraints). From [22], there are two approaches to solve the constraints satisfaction problems which are search

algorithm and consistency technique. One of the most common techniques used as the search algorithm is backtracking.

A backtracking algorithm is one algorithm that solves a problem iteratively and tends to build the solution incrementally, one at a time [27]. To make the solution is free from violation of the constraints, the solutions that are failed to follow the regulations will be discarded.

In the timetable scheduling process or other problem domains, one variable is filled up with a value from the value's domain. Then, this assignment of values into the variables will be checked. If there is any violation of the constraints found in this variable and the previous one, the value assigned to this variable will be removed and this variable is now free to accept other values and the checking will be running again [22]. The backtracking algorithm is very intuitive to code. Its coding complexity is much simpler than other algorithms such as genetic algorithm and particle swarm optimization.

After that, by using the recursion function code, the backtracking algorithm can work with all the possible solutions for the given problem [28]. Thus, it does not eliminate the chances of any individual to be included in finding the global optimal solution.The backtracking algorithm has a very high time complexity, and the complexity will exponentially increase in solving the NP problem. As mentioned above, timetable scheduling is a hard NP problem. Hence, the time complexity in solving the timetabling problem will be high.

Besides, redundant work, will be its drawback. Even if the fault individual was recognized previously, the information was not memorized [29]. Subsequently, repeated work on that particular fault individual will be done, hence consuming more resources and time.

The next weakness of this algorithm are late error detection and thrashing [30]. Late error detection means that the backtracking algorithm could not detect the errors or conflicts before it occurs. Whilst the thrashing means that the failure occurred will repeat due to the same reason. This is due to the incompetency of backtracking algorithm in detecting the real reason for such conflict [29].

2.2.3 Particle Swarm Optimization (PSO)

Particle Swarm Optimization is an optimization algorithm developed by Kennedy and Eberhart in the year 1995. This optimization problem was built based on a simplified social system simulation [31]. From the historical point of view, Particle Swarm Optimization is developed through the observation of the social behaviours of groups of animals [32]. The algorithm was designed and introduced by the developer to simulate the birds' behaviours.

Particle Swarm Optimization works by using the concept of social interaction and self-learning. Like the genetic algorithm and other evolutionary computation techniques, the Particle Swarm Optimization is initially having a population that consists of random solutions, and then by undergoing reproduction, the optimal solutions of the problem will be obtained. From the first until the last iteration, the particles in the swarm can remember their best positions. After that, the best positions of each of them are shared by the other particles. Then, the particles will be able to make use of the positions of others to determine their next position in the iteration process. As a result, they will be able to reach the optimal result or near the optimal result.

The Particle Swarm Optimization algorithm is starting in an initialization phase where then the position of the particles and the velocity of the particles will be generated in a random manner [33]. Then, the desired optimization fitness function of the variables is evaluated for each of the particles. After that, unlike genetic algorithm, the fitness evaluated is used to calculate the individual historical optimal solution and then followed by the swarm historical optimal solution calculation. Finally, by updating the position and velocity of the particle using the updating equation, the result of this algorithm can be obtained. After looping the steps from the initialization until updating the position and velocity, the optimal solution can be obtained when the users are satisfied [31]. The figure 2.2.3.1 below is showing the flowchart of the Particle Swarm Optimization [10].

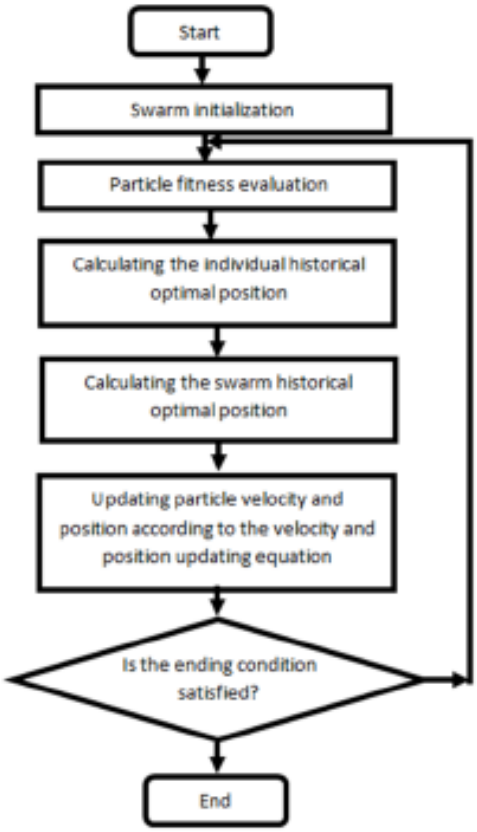


Figure 2.2.3.1 Flowchart of the Particle Swarm Optimization

As Particle Swarm Optimization is concerning with the scheduling problem, hence it is suitable to be used in the timetabling problem. The Particle Swarm Optimization is often combined with another scheduling algorithm such as with a genetic algorithm [18], with a simulated annealing algorithm [34], etc. The calculation of the Particle Swarm Optimization is easy to understand, and its coding is easy to be implemented compared to other algorithms. After that, this algorithm is derivative free. Or in other word, this optimization is free from the problem of unavailable derivative information in certain code function.

Besides, the PSO is an optimization method that consider very few algorithm parameters comparing to others. Or in other word, the parameters of the problems will not be greatly affected the solutions calculated as in other algorithms. Due to its less dependent of the initial points, it is said to be robust comparing to other algorithms.

Lastly, the PSO algorithm can generate the result in a speedier manner compared to other algorithms. As there is no mutation and crossover in the PSO

algorithm, the PSO algorithm only consider the scheduling by the particle's speed. Only the optimal particle can proceed and transfer the information to another particles, hence the latency to deal with the non-optimal particle is reduced, and the speed of the calculation is faster. The weakness of particle swarm optimization is that it only considers the best particles, and this deprives chances to consider other particles which may also produce a correct and optimal result later. In other word, the PSO algorithm is too depending on the global optimum particles, where the local optimum will be trapped. After that, the velocity and position of the previous best particle will increase the particle's position, which may cause the solution to move away from the optimal result [35].

The last drawback of particle swarm optimization algorithm is that it cannot handle the scattering and optimization problems [17]. Besides, it cannot also handle the non-coordinate system problem such as the rules of movement of the particles in the energy field as well as the solution to the energy field.

2.2.4 Comparison among the timetable scheduling techniques

	Advantages	**Disadvantages**	**Complexity**	**Iteration**	**Additional techniques**	**Accuracy**	**Redundant work**	**Difficulty**	**Speed**
Genetic Algorithm	1. works in parallel 2. able to handle multiple parameters 3. looks for the solutions in multiple directions in given time	1. require special definition for the problems 2. fitness function needs to be handled well 3. local optimal might be obtained and might affect the achievement of the global optimal	$O(n^2)$	Yes	Crossover and Mutation	High	No	High	Moderate
Backtracking Algorithm	1. very intuitive to code 2. work with all possible solutions for the given problem	1. has very high time complexity 2. redundant work 3. late error detection 4. thrashing	O(n!)	Yes	Depth-first search method	High	Yes	Low	Slow

Particle Swarm Optimization Algorithm	1. easy concept to understand and code 2. consider only a few parameters 3. generate the result fast	1. only consider the best particles 2. cannot handle the scattering and optimization problems	O(n)	Yes	-	High	No	Moderate	Fast

Table 2.2.4.1Comparison among the Timetable Scheduling Techniques

According to the table above, it is known that the backtracking algorithm has the highest complexity where its processing time is the highest. This is due to its nature of comparing every individual during the scheduling process. Unlike, backtracking algorithm, the genetic algorithm and particle swarm optimization algorithm only concern about the optimal individuals. Besides, the backtracking algorithm does also redundant work during the comparing processes. In terms of concept and coding difficulties, genetic algorithm is the most difficult to understand among these algorithms. Whilst backtracking algorithm is easy to be understood and its coding is quite intuitive.

After that, the particle swarm optimization algorithm is the speediest algorithm among them because it only consider the most optimal solution and process only that solution during the scheduling process. There is no additional /special techniques for particle swarm algorithm. Whilst the genetic algorithm did use the techniques of crossover and mutation, where this is quite differing from others. Lastly, the backtracking algorithm uses the technique of depth-first search to search for the solutions.

In terms of similarity, all these algorithms can produce the accurate result, although in some circumstances may not be the optimal one. Lastly, all of these three algorithms did process in a recursive manner.

In conclusion, the genetic algorithm is the one which has the highest degree of toleration to generate an optimal solution. The backtracking algorithm is the easiest algorithm to implement , however, its speed and resources consumption is questionable. For the particle swarm optimization algorithm, it is the fastest among the three algorithms, however, as compared to genetic algorithm, it is not so tolerated to generate the optimal solution although in most cases it can successfully produce the optimal one.

2.3 Constraints used for the Related Works

2.3.1 Hard Constraints

To schedule a timetable, hard constraints are the rules that must be followed to create a feasible timetable. From the previous research papers, there are many different hard constraints proposed to create a timetable scheduling system. Among them constraints, there are a few hard constraints being the common visitors in the papers. Table below shows the hard constraints and the papers which proposed them.

Hard constraints	Past research papers
Each student can only take one class at a time	[1], [12], [9], [10], [13], [36], [37], [16], [38], [39], [40], [41], [42], [19], [14], [15],
One lecturer can only have one class at a single time	[1], [4], [12], [8], [9], [10], [36], [37], [16], [38], [39], [40], [41], [14], [15], [43], [44]
A classroom should have only one class assigned at a time	[1], [4], [8], [9], [10], [13], [16], [38], [19], [14], [15]
The capacity of the classroom should be large enough to hold the students' size in each class	[1], [12], [8], [10], [16], [38], [39], [41], [19], [44]
Specific room is required to be considered	[1], [8], [9], [39], [41], [19], [44]
The main subjects' classes should not be conducted at the same time	[9], [10], [36]
Each lecturer's working hours are limited during a week	[8], [10]
The available periods per day is maximum at 8 hours	[10], [40]
No subject has more than one lecture a day	[1], [4]

Each class should be taught by one lecturer at one time	[8]
Students should not have consecutive classes more than 4 hours	[38]
Specific timeslot must be reserved for the faculty meeting/ special events for the lecturers	[38]
Minimum number of students are required to open the class for a subject	[10]

Table 2.3.1.1Hard Constraints Used in the Related Works

From the table above, it is showed that there are a few hard constraints always used when generating the timetable. Among them are "each student can only take one class at a time", "each lecturer can only take one class at a time" and "a classroom should have only one class assigned at a time". The above constraints showed that the resources or parameters for a class should not be assigned to two or more classes/events at the same time. Students, lecturers, and classrooms here are the resources to be considered to form a class, they cannot exist together at the same time in the different classes.

After that, the constraints related to classrooms are considered also. For example, the constraint "the room capacity should be large enough to hold the students in each class" stated that a classroom size needs to have enough seats for the student to sit. Besides, "specific room is required to be considered" defined that for the special classes such as practical classes which require the uses of accessories or classroom resources should be assigned to the lab which can fulfil the classes' needs.

Lastly, there are some other hard constraints which may not be so popular among the past research, but also good enough to be considered to produce a good timetable. Among them are "the main subjects' classes should not be conducted at the same time" which this is ensuring the students can always attend to the important classes without clashes, "each lecturer's working hours are limited during a week"

which tends to reduce the workloads of the lecturers, “the available periods per day is maximum at 8 hours” which tends to control the time to conduct the classes in a specific amount of time, and etc.

2.2.2 Soft constraint

Soft constraints	Past research papers
Lecturer inclines to choose their preferable timeslots and rooms	[2], [4], [8], [9], [38], [44], [43],
The gap between lecturers’ timetable should be minimized/avoided	[36], [39], [40], [41], [14], [15]
Classes should not be held after 6pm	[38], [39], [19]
Lecture and tutorial/practical for a course should not be held on the same day	[1], [38]
The gap between students’ timetable should be minimized	[39], [40]
A classroom should be selected near the lecturer’s office	[8]
The number of classrooms used should be minimized	[8]
The students need to have consecutive class events in the same room/building	[2], [44]
Largest number of the students has the highest priority	[38]

General course should be opened to all faculty students	[1]
Friday session stops by 1pm to 3 pm	[1]
The daily teaching sessions of the lecturer should not exceed a specified number of periods	[2]
Lecturer should be only allowed to teach the subject within their expertise area	[38]

Table 2.3.1.2 Soft Constraints Used in the Related Works

From the table 2.3.1.2 above, we can see that there are a few soft constraints which are quite popular among the related works. Among them are "Lecturer inclines to choose their preferable timeslots and rooms", "The gap between lecturers' timetable should be minimized/avoided", and "Classes should not be held after 6pm". The first soft constraint stated that it is for the lecturers to choose their preferable timeslots and rooms based on their needs and other reasons. After that, the second soft constraints stated that there should be some empty time for the lecturer to rest before they can continue to conduct the next class after finishing one. The third soft constraint stated that the class after 6pm should be greatly reduced.

After that, there are some soft constraints being considered by the related works also. For example, "lecture and tutorial/practical for a course should not be held on the same day", "the gap between students' timetable should be minimized", "students should have consecutive classes in the same building/room", etc.

2.4 Timetable Scheduling Systems

2.4.1: aScTimeTables [5]

aSc Timetables is a Timetable Scheduling application that was founded in 1993 in Slovakia. It is used to generate the primary and secondary school timetables. By using this software, the timetable can be generated automatically with some users'

manual adjustments. This software came out with an algorithm that can help the users to verify their schedule to check for the existence of any conflicts.

aSc Timetables will firstly allow the users to input the data required such as the subjects. Classes, classrooms, teachers, students, etc (see figures 2.4.1.1, figures 2.4.1.2, figures 2.4.1.3, figures 2.4.1.4). As the data input requires a lot of time especially for a huge amount of data, the software provided users convenient to input and simply import the data. However, if the users like to input the data in a format such as a CSV file, the software does not support it. After import or input all the data inside, the users can either generate the timetable automatically or manually following their preferences (see figure 2.4.1.5).

Strength of aScTimetables:

1. This software is mobile responsive to aid in the users' mobilities.

2. It can analyze the distance between buildings to shorten the movements of the timetable's users from one building to another.

3. Spread the lessons equally throughout the week.

4. Allow manually adjust of the timatable

Weaknesses of aScTimetables:

1. It does not support multiuser use

2. Students and lecturers cannot view the scheduled timetable themselves

3. It is not accessible from remote places

4. It does not support CSV file, only accepts data files in XML filetype

The figures below show some interfaces in the aSc Timetables.

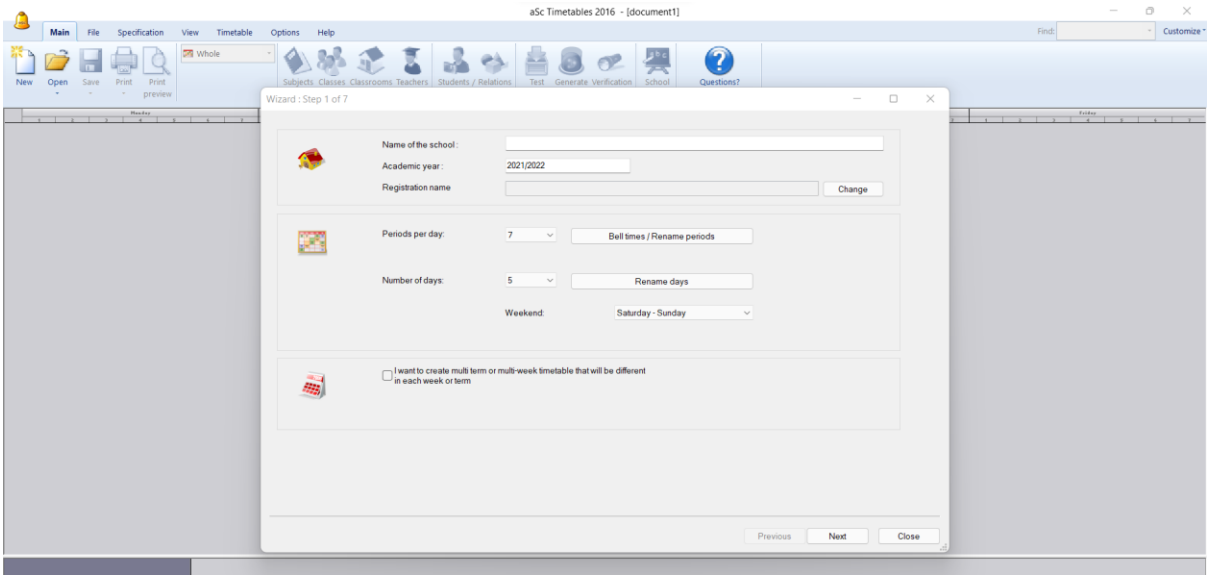


Figure 2.4.1.1 Main page of aSc Timetables

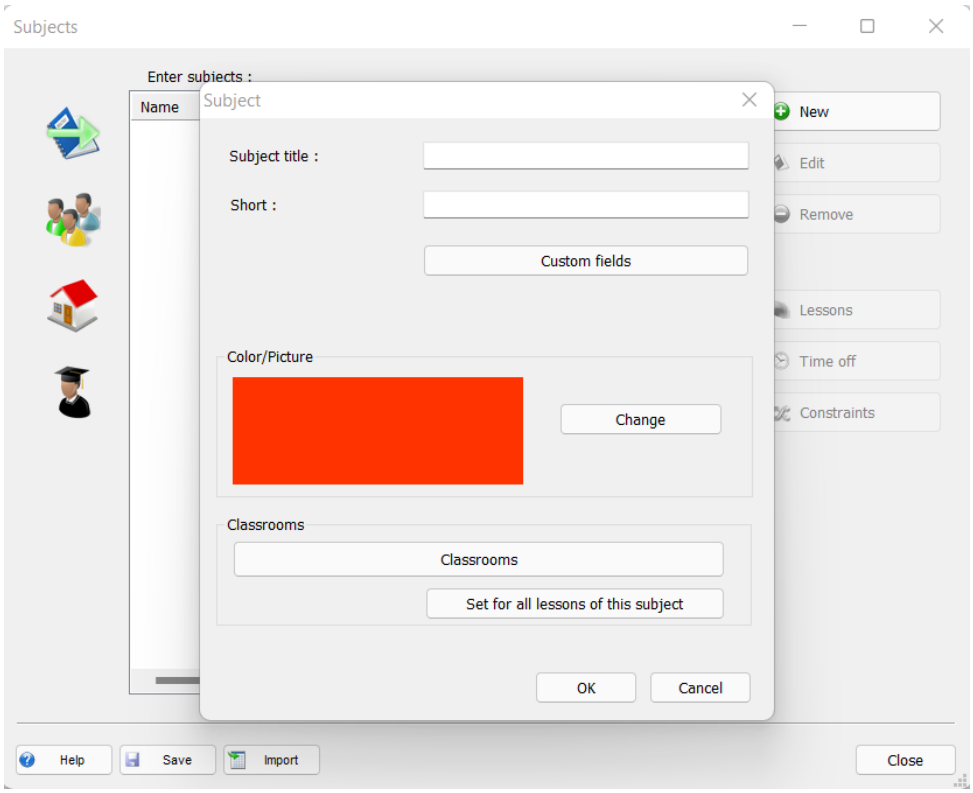


Figure 2.4.1.2 The subject data input interface of aSc Timetables

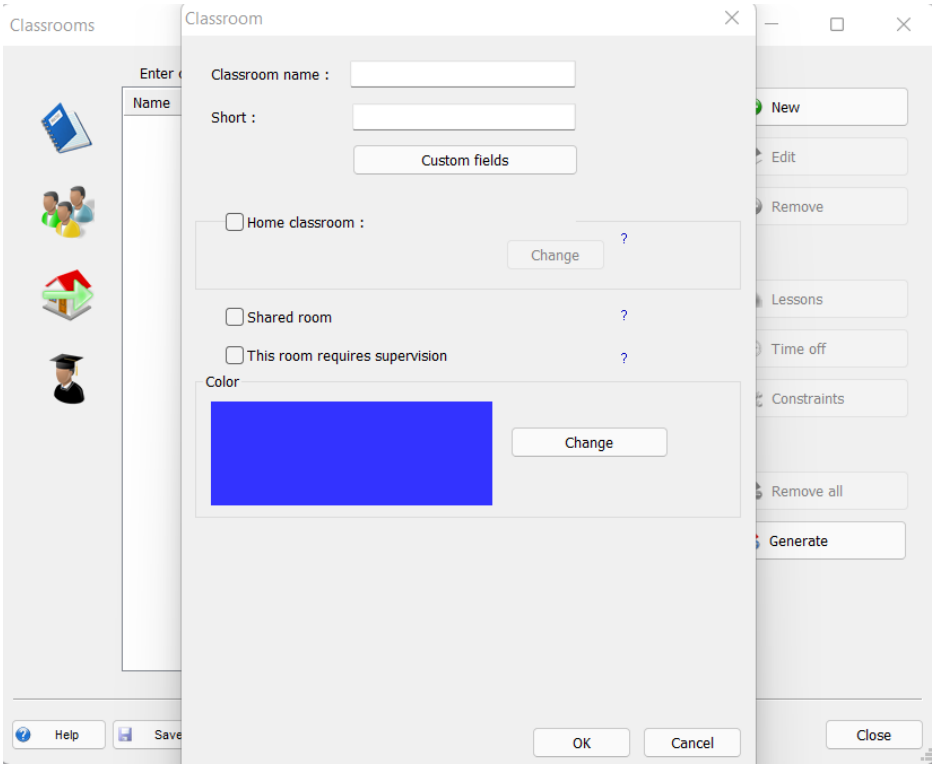


Figure 2.4.1.3 The classroom data input interface of aSc Timetables

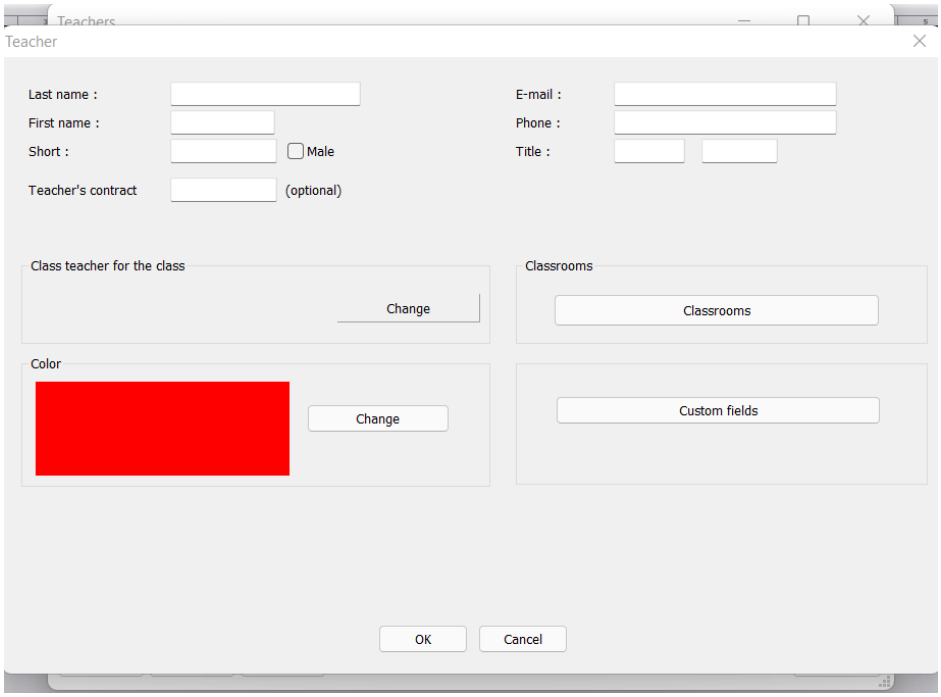


Figure 2.4.1.4 The teacher data input interface of aSc Timetables

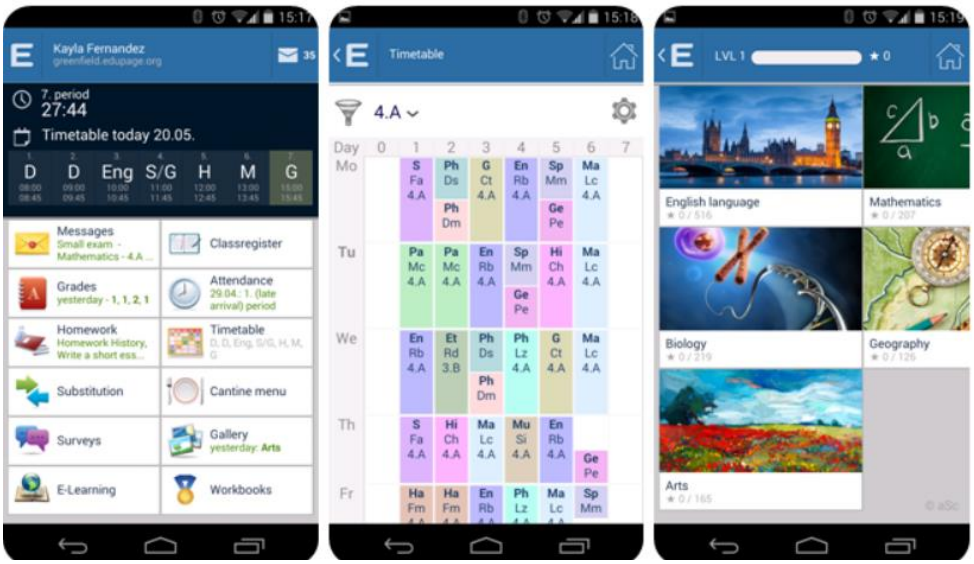


Figure 2.4.1.5 The created timetable using aSc Timetables in mobile view

2.4.2 FET [6]

FET is another timetabling software, but it functions for the high school or the university management teams. FET was founded in the year 2002. It provides many different features for the users in creating their timetables.

FET supports multiple languages for a large variety of users. After that, FET consists of a large variety of constraints which are grouped into different classes such as student's time constraints, teacher's time constraints, activity's time constraints, space constraints, etc. Besides, it supports the CSV files import and export format. This feature is very convenient for the users who are using Microsoft Excel as their software platform in doing the scheduling activity.

To use the FET, the users are required to input the data such as basic information of the subjects' names, classrooms, lecturers' names, etc (see figures 2.4.2.1, figures 2.4.2.2). After the data are inputted, the users can verify the activities that are required by each subject. For example, a subject can be made up of lecture class, tutorial class, practical class, and discussion class. Users can also specify some specific time they want (see figure 2.4.2.3). After all the above activities are done, the timetable can be generated if there is no error found [11].

Strengths of FET.:

1. It supports multiple languages

2. It allows manual, semi-automatic and fully automatic timetable generation

3. It is platform independent

4. It allows import and export of data in CSV format

Weaknesses of FET.:

1. It does not support multiuser uses

2. Students and lecturers cannot view the scheduled timetable themselves

The figures below show some interfaces of FET software.

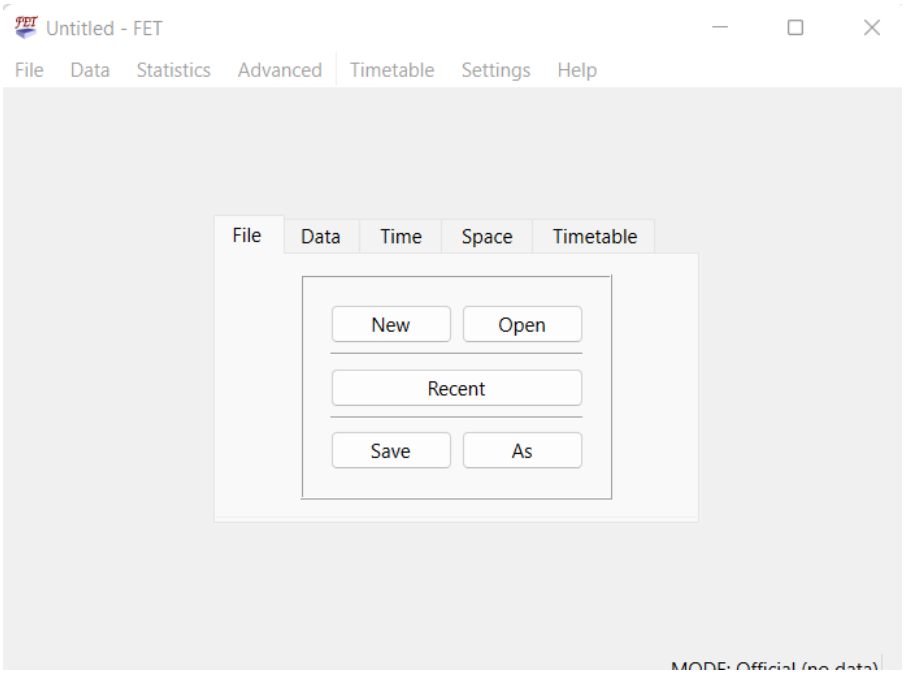


Figure 2.4.2.1 The file input interface pf FET

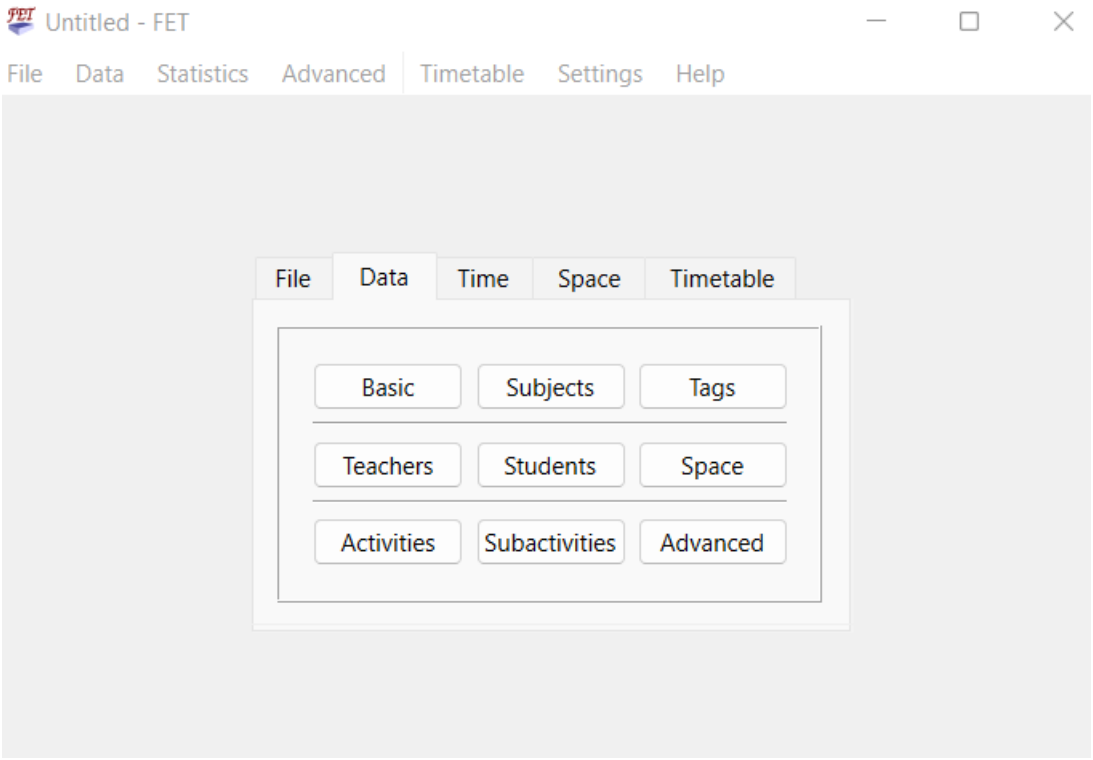


Figure 2.4.2.2 The resources data input interface of FET

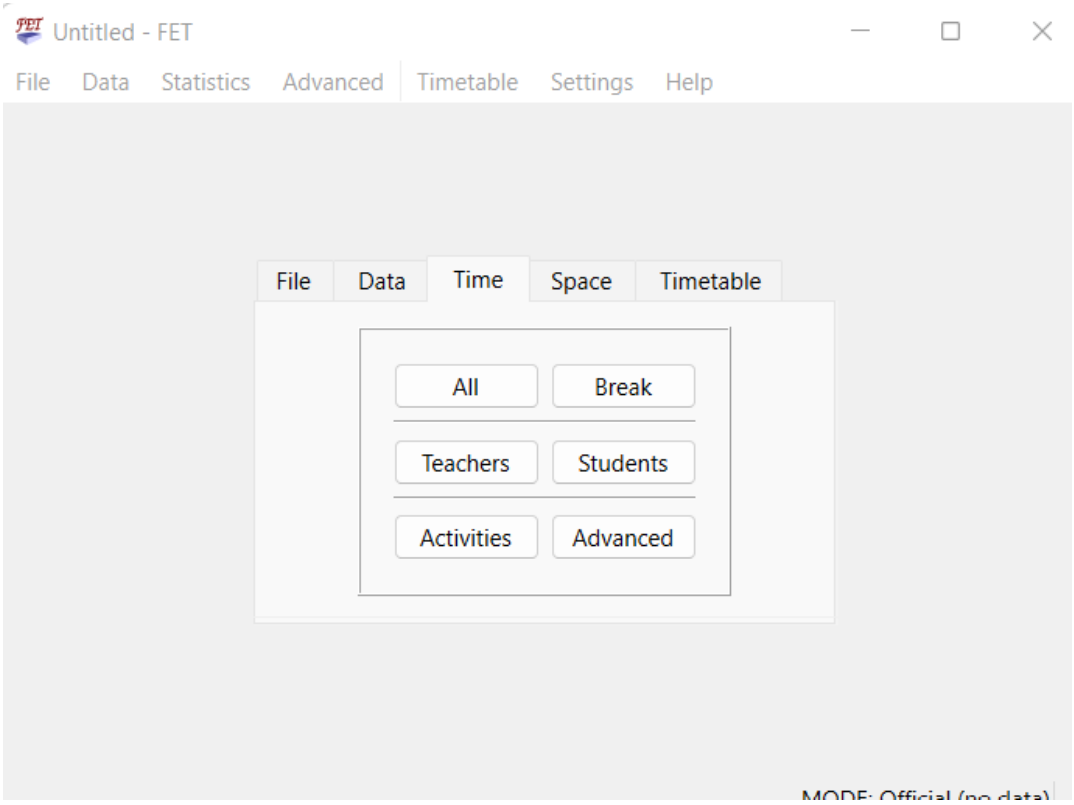


Figure 2.4.2.3 The time data input interface of FET

2.4.3 iMagic [45]

The next software is the iMagic software which allows the timetable creation for schools, colleges and the universities. By using this software, one can key in the details of the timetables, and the software will be creating a workable timetable to the user.

To generate the timetable, the user needs to first select the study days and the timeslots as the range for the timetable. After that, the users need to input the data such as the tutors name, subjects name, rooms, and classes name (see figure 2.4.3.1, figure 2,4,3,2). The users can then manage the subjects taught by the tutors. This is to ensure that the tutors are in charge of their respective subjects. The users can also modify or delete the inputted data.

After that, the users can either choose to input the timetable slots manually by selecting the respective tutor-subject pair and the room, or the user can select the auto-fill option to fill up the timeslot automatically (see figure 2.4.3.3). Then, after selecting the data to be used in the timetable form the data pool, the timetable will be generated automatically, and the user can view the timetable and print it out (see figure 2.4.3.4).

Strengths of iMagic:

1. It has very user-friendly interface

2. It allows the output of CSV file exportation or web page

3. It has a search and replacement of the inputted data

Weaknesses of iMagic:

1. It does not support the multiuser use

2. It does not support other languages

3. It does not allow the accessibility from remote places

4.Lecturers and students cannot view the timetable by themselves

The figures below show the snapshots of several interfaces of iMagic software:

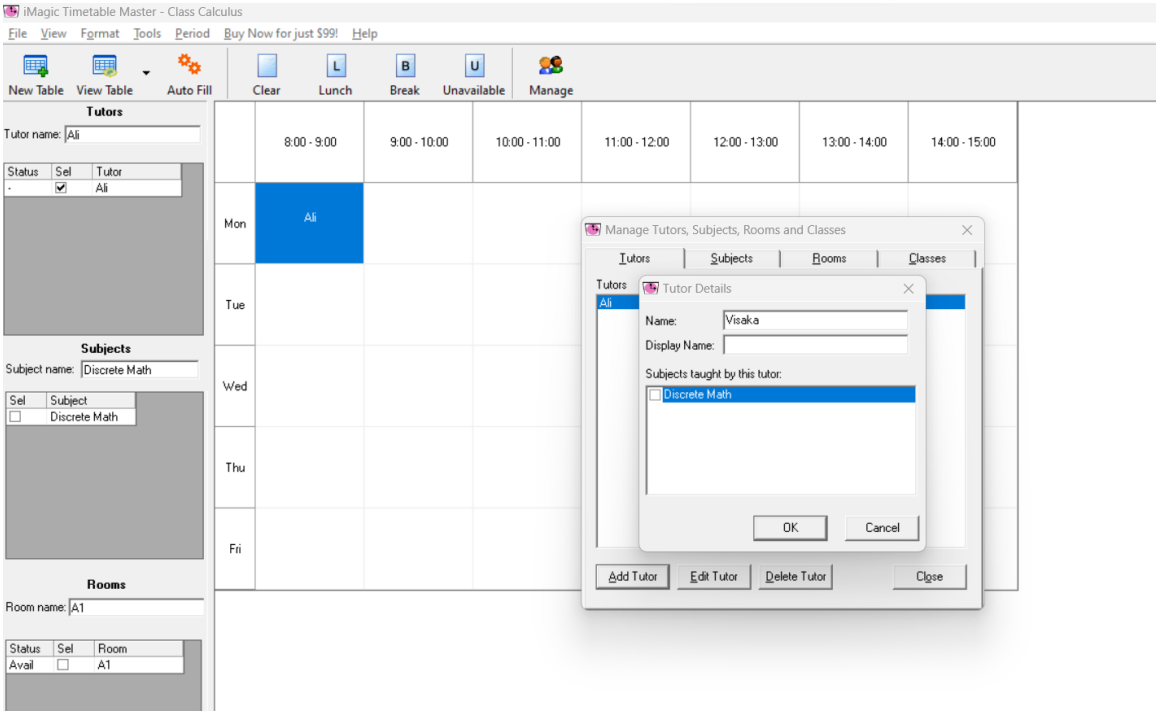


Figure 2.4.3.1 The tutor's data input interface of iMagic software

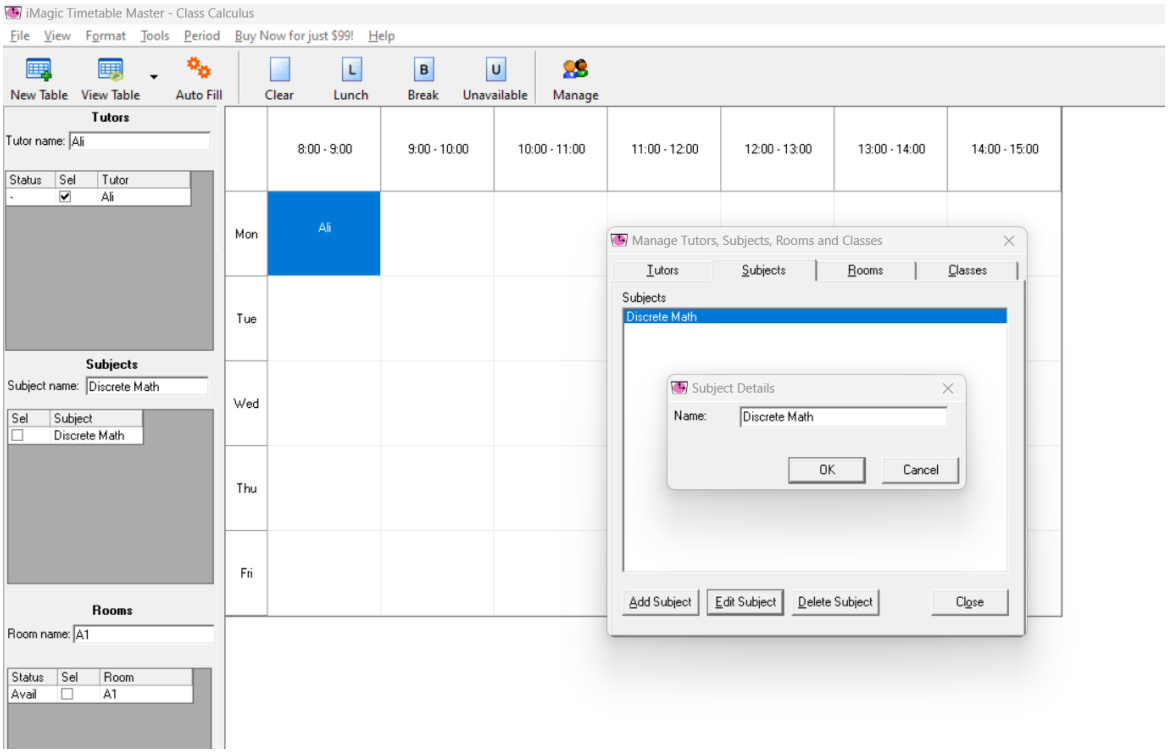


Figure 2.4.3.2 The subject's data input interface of iMagic software

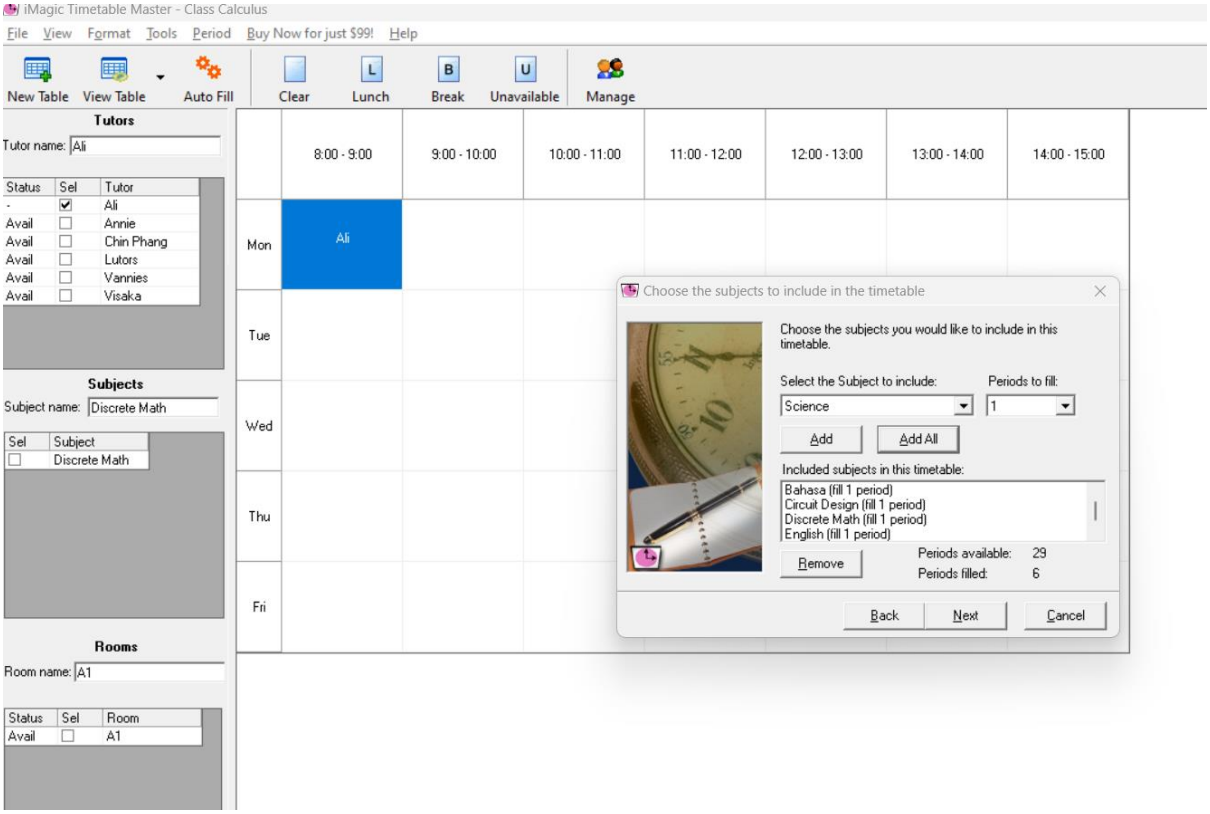


Figure 2.4.3.3 The configuration of the timetable when generate the timetable automatically in iMagic software

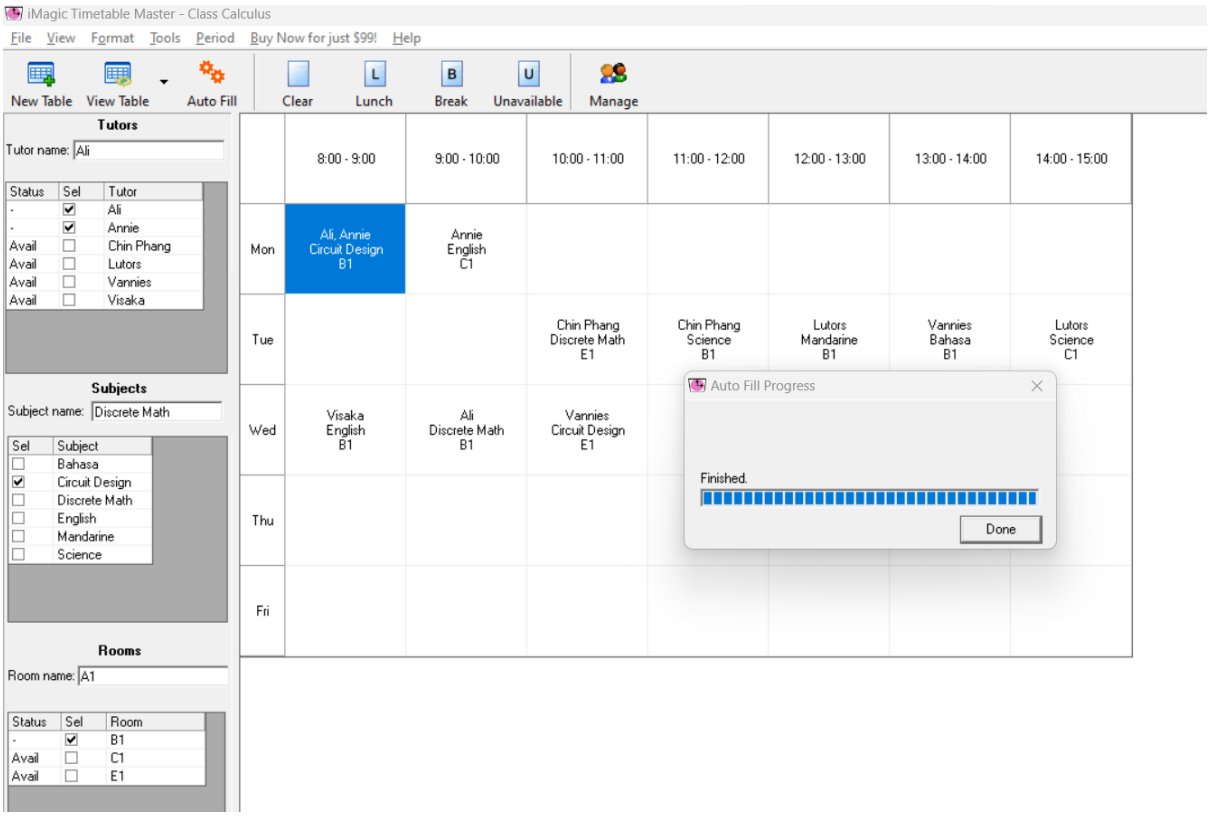


Figure 2.4.3.4 The view of generated timetable in iMagic software

2.4.4 Mimosa [46]

Mimosa is a timetabling software developed in 1986. The objective of this software is to provide an automated timetabling system for all kinds of institutes, companies, and organizations.

Using this software, the users can initialize the terms amount, weekly teaching days, and slots of classes per day before going into the main part of the software (see figure 2.4.4.1). Then, the users can edit the resources and the events required in the timetable. The software was not only developed for the academic institutions uses, but there are also several special resources and events to be chose (see figure 2.4.4.2). The users are free to add, modify, or delete the resources and events in their file.

After that, Mimosa allows the input of data using the .MXT text file. After all the data are being entered, Mimosa can generate the timetable automatically, and the users can choose to adjust the timetable manually. After the timetable is finalized, the users can export it to the web or print it out.

Strength of Mimosa:

1. It support multiple languages.

2. It supports businesses, com panies, and other institutes structure

3. It allows export of timetable to .mxt format, .csv format or .vcs format

Weaknesses of Mimosa:

1. It only supports the .mxt file data where this filetype is seldom used by public

2. Lecturers and students cannot view the timetable by themselves

The figures below show the snapshots of several interfaces of iMagic software:

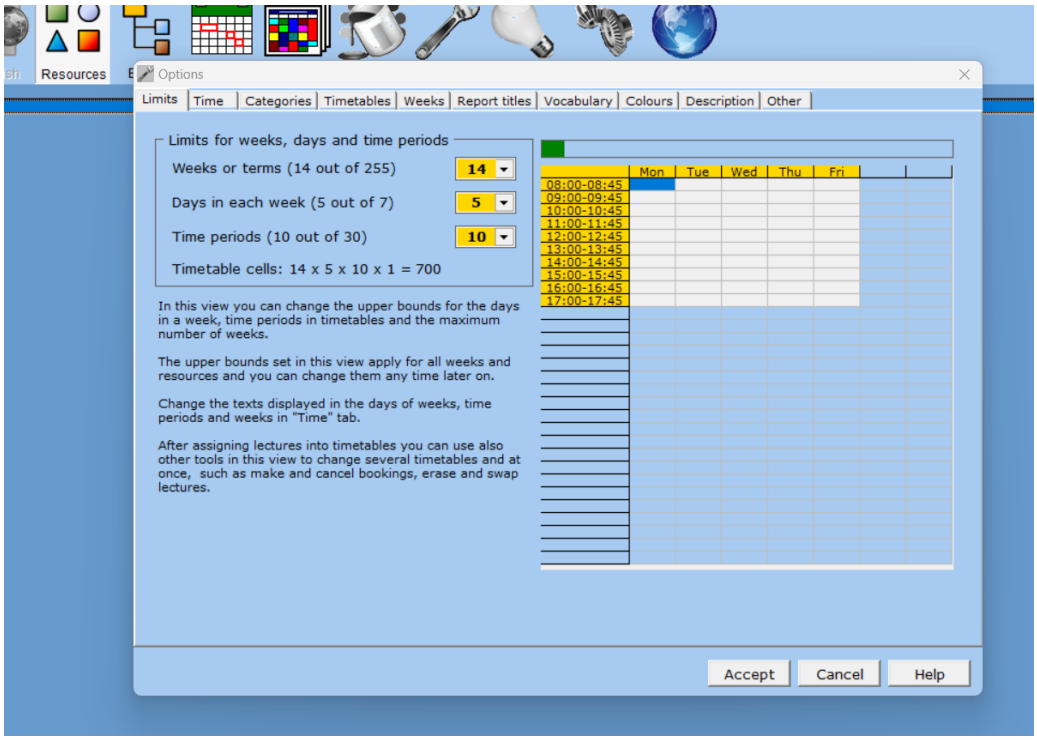


Figure 2.4.4.1 input of terms, days of a week, and timeslots per day in Mimosa software

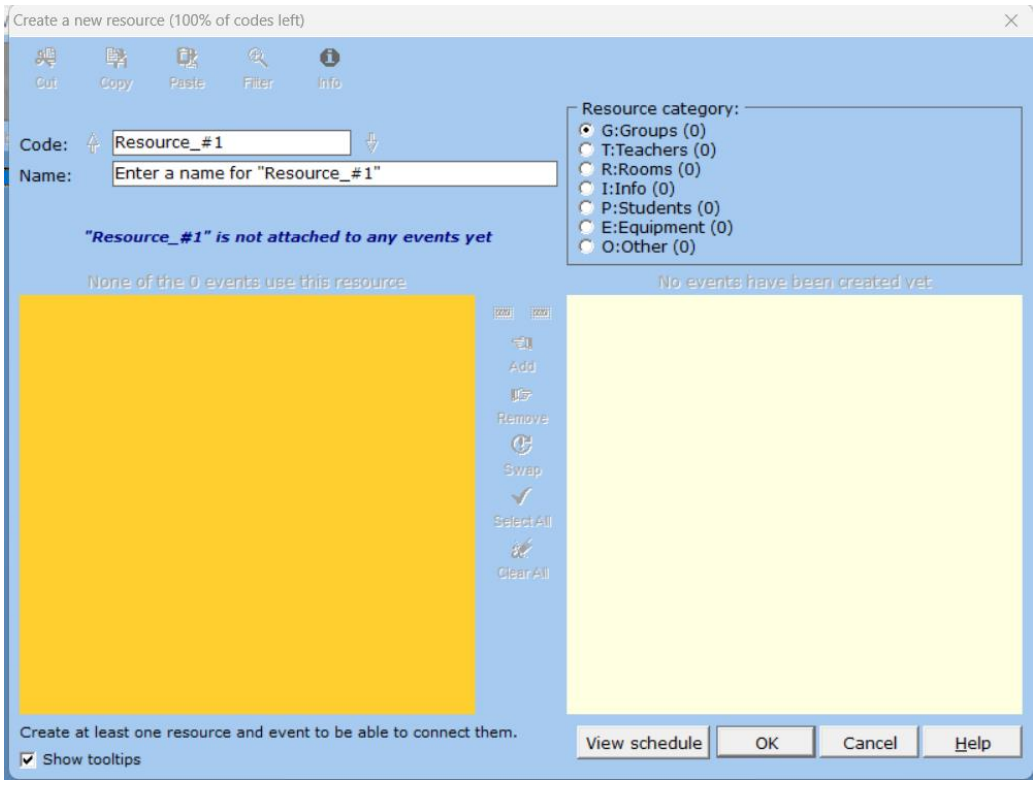


Figure 2.4.4.2 The resources and events input interface of Mimosa software

2.4.5 Comparisons among the Timetable Scheduling Systems

Table 2.4.5.1: Comparison among the timetable scheduling system:

Features\Software	aScTimetables	FET.	iMagic	Mimosa
Generate timetable	Yes	Yes	Yes	Yes
CRUD of data	Yes	Yes	Yes	Yes
Multiple languages support	No	Yes	No	Yes
Data Import	Yes	Yes	Yes	Yes
Calculation of rooms' distances	Yes	No	No	No
Target users	Educational institution	Educational institution	Educational institution	Educational institution, businesses and companies
Mobility	Yes	No	Yes	No

Figure 2.4.5.1 Comparisons among the Timetable Scheduling System

Based on the Table 2.4.5.1, it can be known that a useful timetabling system must have the functionalities of generating the timetable automatically. Although the algorithm behind the systems might differ, but the point is to generate a non-conflict timetable. After that, for the data management, the CRUD function of the data is necessary for the users to have a more dynamic and false-tolerate operations when using the software. Data import from external file such as .csv file is important to reduce the works of the users. That is all the necessary functionalities to be existed in the timetable scheduling system.

From the systems reviewed, there are also some extra functionalities that can ease the users operation. For example, multi-language support allows the users from different language backgrounds to use the software. The mobility of the software is also convenient for the users who demand for it.

Lastly, extra scheduling function such as ability to calculate the distances between rooms did also allow the aScTimeTables to produce a more optimal timetable for users. It can be said that the deep study and implementation of the users' needs in the timetable scheduling will be the advantage for a system to produce timetables that match the users' needs. It this project, the users' needs in terms of hard and soft constraints will be studied and implemented in the timetable scheduling algorithm, to act as extra functionalities in the proposed system.

2.4 Critical Remarks

The timetabling problem has been troublesome for the organizations that need to generate the timetables for their daily operation, especially for universities. Many algorithms had been introduced year by year and the researchers tend to find an algorithm that will work for any kind of scenario. However, it is not possible to have an algorithm that can solve any problem. Furthermore, as mentioned above, the timetable scheduling complexity is different based on different constraints proposed. Hence, there is no a specific solution to solve the timetabling problem. Moreover, by using one and only one algorithm like a genetic algorithm[4], [12], Particle Swarm Optimization algorithm, Tabu Search algorithm [16], etc. the previous researchers found that it was hard to get the most optimal timetable scheduling as every algorithm has its shortcomings.

As there will be one new constraint added inside this project, hence the complexity of the timetabling problem will be unknown. Besides, different algorithms will have different performances given different constraints and regulations. In these circumstances, to solve the problem of the newly added constraints and inconsistency of the algorithm selection, genetic algorithm is always the best choice to handle these uncertainties. In fact, we can see that from previous research, many researchers have chosen genetic algorithm as their first choice to solve the problem. Even the heuristic approaches of algorithms are always using genetic algorithm as the base algorithm to work with other algorithms. This is due to its strength of high accuracy and no redundant works. To ensure the obtain of more optimal results, the later researchers started to hybridize different algorithms to work together and cover up each other's

shortcomings to get the optimal output. For example, the hybridization between Genetic Algorithm with Particle Swarm Optimizer [25], hybridization between the Genetic Algorithm with the Fuzzy algorithm [29], Hybridization of Particle Swarm Optimizer with the Annealing algorithm [26], etc.

In this project, to implement a timetabling tool with the uncertainty of the newly added constraints, the genetic algorithm which has the advantages of easily understandable, population of points searching, multi-objective optimization support, stochastic, etc. will be chosen to be studied and be implemented in this project.

Chapter 3

SYSTEM METHODOLOGY/APPROACH

3.1 Project Development

In this project, the agile system development model will be chosen. This is due to its characteristic of high flexibility and ease of changes of system requirements. For the timetabling system, there are a lot of constraints to be considered while generating the timetable. Hence, it is suitable to use agile model in this project. For example, the study mode of the university may be changed from time to time corresponding to current situation of the Covid-19 pandemic. Agile model is suitable to deal with these changes. There are 6 phases in this model which include planning, analysis, design, implementation, testing, and maintenance where these phases are interleaved within each other.

During **planning phase** of this project, the idea and draft to develop the system were introduced based on the current situation and problems. For example, the lack of functionalities of the existing software to generate timetable in hybrid mode is one of the problems discovered. Besides, the project scope of this project is being created. The objectives of the project are then being proposed to solve the problems found. The project scope which defines the dataset to be tested, the target users of the system, the platform to develop the system, etc are defined. Lastly, to have the ability to generate the timetable, certain algorithms and software were gathered to be studied.

After that, during the **analysis phase**, the collected algorithms and software functionalities were studied and analysed. The capabilities of the algorithms in terms of strengths and weaknesses were analysed. For example, their time complexity, structure, accuracy, speed, etc. Then, the existing software functionalities to solve the existing problem were also studied. Besides, the requirement of the system is being reviewed to study how to propose a better and newer product to the users. For example, the characteristics of the timetable in the university will be studied by observing the documentation of the timetable structure exists on the university official website.

In the **design phase**, after analysing the algorithms and selected the most suitable one (genetic algorithm), the structure of the system is being designed. For example, the new constraints to be adopted in this system were thought of. Furthermore, the design of the phases of the selected algorithm was conducted. For instance, the design of the chromosome structure in genetic algorithm, the design of the selection phase method, cross over method and etc. Diagrams such as UML diagrams and the algorithm structure diagram will be included in this section too. After that, the system architecture, system database, and hardware and software to be used are designed in this phase.

Followed by the **implementation phase** where the design proposed above will be implemented in the specific platforms. The system will be implemented as a standalone application using java language code in eclipse IDE. While developing the system, the new requirements or constraints will be reviewed, and any changes will be accepted to be implemented in the system.

Then, in **testing phase**, the developed system will be tested to ensure that there is no error in scheduling the timetable and no violations of the constraints in the timetable produced. Different constraints have different ways to be tested. For example, each timeslot in lecturers' timetable should not exist more than one class fulfil the constraint of "Each lecturer can only teach one class at a time".

The last phase will be the **maintenance phase** where the errors or bugs found in the developed system will be found and updated from time to time.

3.1.1 Gantt Chart Showing the Time Flow of the System Development

ID	Title	Start Time	End Time
20	Test the Algorithm Speed using Different Scena...	06/20/2022	06/22/2022
1	Code the System (added Hybrid features)	06/23/2022	07/01/2022
2	Test the System	07/04/2022	07/06/2022
3	Code the System (added constraints selector)	07/08/2022	07/11/2022
4	Test the System	07/12/2022	07/14/2022
5	Code the System (added parallel programming...	07/18/2022	07/20/2022
6	Test the System	07/21/2022	07/22/2022
7	Code the System (work on User Interface)	07/25/2022	08/05/2022
8	Test the System	08/06/2022	08/09/2022
9	Code the System (continue on User Interface)	08/11/2022	08/19/2022
10	Test the System	08/21/2022	08/24/2022
12	Modify the Report	08/25/2022	09/04/2022
16	Finalize the FYP2 Report	09/04/2022	09/06/2022
17	FYP2 Report Submission	09/09/2022	09/09/2022
19	Presentation	09/12/2022	09/12/2022

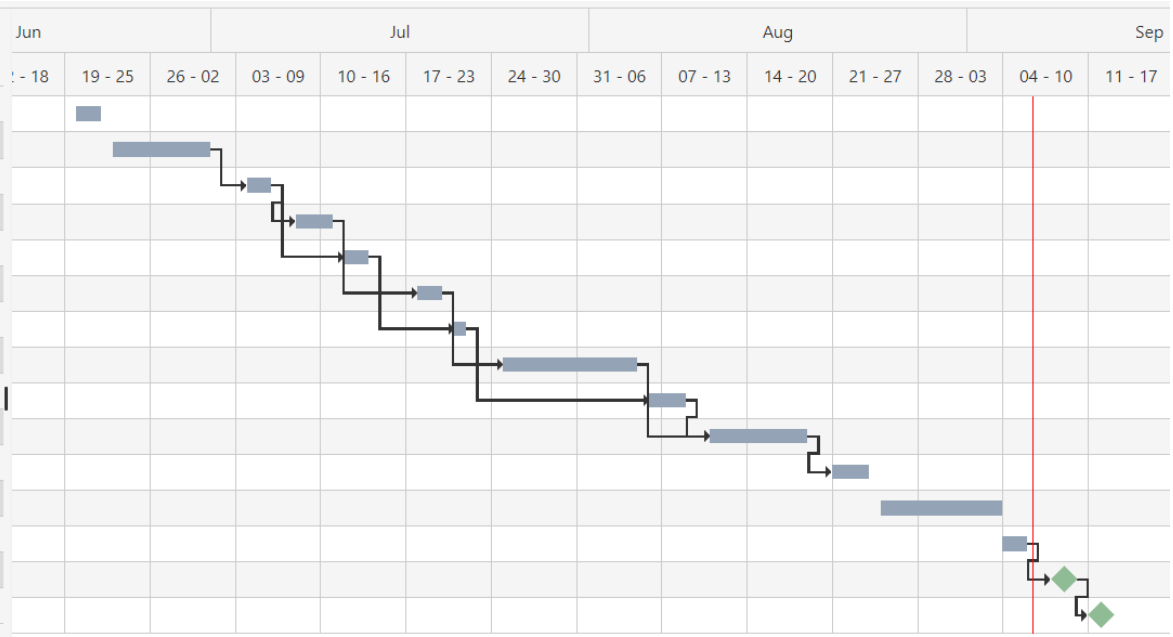
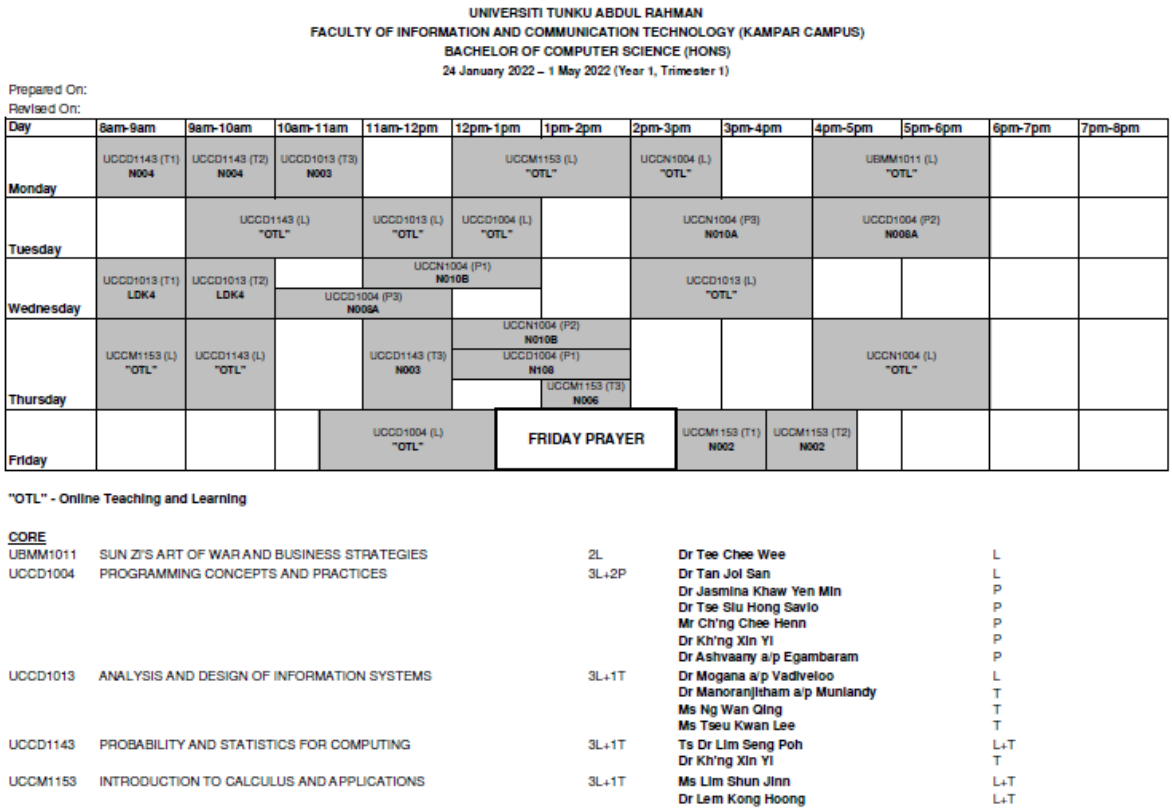


Figure 3.1.1 Gantt Chart of the System Development

3.2 Data Information Gathering Required for Timetable Tool

The information gathering in this project is referred to the activity of getting the timetable structure in Faculty of Information and Communication Technology in UTAR. The timetables of different programmes in FICT in the current trimester were got from the UTAR official website. Then, the timetable structure is being studied. There are various information can be obtained from the timetable such as the courses to be taken by students from different trimesters, the lecturers' names and their correspond courses to teach, the rooms names and types to conduct the class events, special timeslots of the timetable in each week, etc. The figure 3.2.1, figure 3.2.2 and figure 3.3.3 below shows some timetables obtained.



UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)
BACHELOR OF COMPUTER SCIENCE (HONS)
24 January 2022 – 1 May 2022 (Year 1, Trimester 1)

Prepared On:
Revised On:

Day	8am-9am	9am-10am	10am-11am	11am-12pm	12pm-1pm	1pm-2pm	2pm-3pm	3pm-4pm	4pm-5pm	5pm-6pm	6pm-7pm	7pm-8pm
Monday	UCCD1143 (T1) N004	UCCD1143 (T2) N004	UCCD1013 (T3) N003		UCCM1153 (L) "OTL"		UCCN1004 (L) "OTL"		UBMM1011 (L) "OTL"			
Tuesday		UCCD1143 (L) "OTL"		UCCD1013 (L) "OTL"	UCCD1004 (L) "OTL"		UCCN1004 (P3) N010A		UCCD1004 (P2) N008A			
Wednesday	UCCD1013 (T1) LDK4	UCCD1013 (T2) LDK4	UCCN1004 (P1) N010B / UCCD1004 (P3) N008A				UCCD1013 (L) "OTL"					
Thursday	UCCM1153 (L) "OTL"	UCCD1143 (L) "OTL"		UCCD1143 (T3) N003	UCCN1004 (P2) N010B / UCCD1004 (P1) N108	UCCM1153 (T3) N006			UCCN1004 (L) "OTL"			
Friday				UCCD1004 (L) "OTL"	FRIDAY PRAYER		UCCM1153 (T1) N002	UCCM1153 (T2) N002				

"OTL" - Online Teaching and Learning

CORE

UBMM1011	SUN ZI'S ART OF WAR AND BUSINESS STRATEGIES	2L	Dr Tee Chee Wee	L
UCCD1004	PROGRAMMING CONCEPTS AND PRACTICES	3L+2P	Dr Tan Joi San	L
			Dr Jasmina Khaw Yen Min	P
			Dr Tse Siu Hong Savio	P
			Mr Ch'ng Chee Henn	P
			Dr Kh'ng Xin Yi	P
			Dr Ashvaany a/p Egambaram	P
UCCD1013	ANALYSIS AND DESIGN OF INFORMATION SYSTEMS	3L+1T	Dr Mogana a/p Vadiveloo	L
			Dr Manoranjitham a/p Muniandy	T
			Ms Ng Wan Qing	T
			Ms Tseu Kwan Lee	T
UCCD1143	PROBABILITY AND STATISTICS FOR COMPUTING	3L+1T	Ts Dr Lim Seng Poh	L+T
			Dr Kh'ng Xin Yi	T
UCCM1153	INTRODUCTION TO CALCULUS AND APPLICATIONS	3L+1T	Ms Lim Shun Jinn	L+T
			Dr Lem Kong Hoong	L+T

Figure 3.2.1 Timetable of Y1T1 Students from Computer Science Programme

UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)
BACHELOR OF INFORMATION TECHNOLOGY (HONS) COMPUTER ENGINEERING
24 January 2022 – 1 May 2022 (Year 2, Trimester 1)

Prepared On:
Revised On:

Day	8am-9am	9am-10am	10am-11am	11am-12pm	12pm-1pm	1pm-2pm	2pm-3pm	3pm-4pm	4pm-5pm	5pm-6pm	6pm-7pm	7pm-8pm
Monday			UCCD2103 (L) "OTL"		UCCE1053 (L) "OTL"			UCCE1074 (P) Even N110A UCCE2023 Odd N110B				
Tuesday		UCCE1074 (L) "OTL"					UCCE2023 (L) "OTL"					
Wednesday	UCCE1053 (P) Even N110A						UCCE1074 (L) "OTL"	UCCE1053 (L) "OTL"				
Thursday	UCCD2103 (L) "OTL"				UCCE2073 (L) "OTL"			UCCE2073 (T) N006	UCCD2103 (T) N006			
Friday	UCCE2023 (L) "OTL"			UCCE2073 (L) "OTL"	FRIDAY PRAYER							

"OTL" - Online Teaching and Learning

CORE

UCCD2103	OPERATING SYSTEMS	3L+1T	Ts Wong Chee Siang	L+T
			Prof Dr Leung Kar Hang	T
			Ts Dr Tan Hung Khoon	T
UCCE1053	PROCESSOR AND INTERFACING	3L+3P	Mr Teoh Shen Khang	L+P
UCCE1074	ELECTRONIC DESIGN AND TESTING	3L+3P	Mr Leong Chun Farn	L+P
UCCE2023	DIGITAL SYSTEMS DESIGN	3L+3P	Mr Lee Heng Yew	L+P
UCCE2073	INTRODUCTION TO DIGITAL SIGNAL PROCESSING	3L+1T	Mr Lee Heng Yew	L+T

Figure 3.2.2 Timetable of Y2T1 Students from Computer Engineering Programme

UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)
BACHELOR OF INFORMATION SYSTEMS (HONS) INFORMATION SYSTEMS ENGINEERING
24 January 2022 – 1 May 2022 (Year 3, Trimester 1)

Prepared On:
Revised On:

Day	8am-9am	9am-10am	10am-11am	11am-12pm	12pm-1pm	1pm-2pm	2pm-3pm	3pm-4pm	4pm-5pm	5pm-6pm	6pm-7pm	7pm-8pm	8pm-9pm
Monday	UCCD3223 (P1) N108 UCCD3223 (P2) N108		UCCD3243 (P) N104 UCCD3113 (T) N006		UCCD2043 (L) "OTL" UCCM1153 (L) "OTL" UCCB1223 (L) "OTL"	UCCD1113 (L) "OTL" UCCB2333 (L) "OTL'	UCCD3013 (T1) N006 UCCA3053 (T) N005		UAMG1043 (L) "OTL" UBMH1013 (L) "OTL" UBMM2013 (T) LDK1 UBMM2023 (T) LDK2				
Tuesday	UBMM1013 (L) "OTL" UBTM1013 (L) "OTL"			UCCD3113 (L) "OTL" UCCD3233 (P) N104		UCCB1223 (P) N110A UCCD1113 (T) N003	UCCM1153 (T) N003	UBMM1013 (T) LDK1 UBTM1013 (T) LDK2 UAMG1043 (T) LDK4	UALE1083 (L) "OTL"		UBMM2023 (L) "OTL"		
Wednesday	UCCA3053 (L) "OTL"		UCCD3113 (L) "OTL" UCCD3243 (L) "OTL"	UCCD3233 (L) "OTL"		UCCD3223 (L) "OTL"			UCCD3053 (L) "OTL"				
Thursday	UCCM1153 (L) "OTL" UCCB2333 (L) "OTL"	UCCD1113 (L) "OTL" UCCD3013 (L) "OTL"		UCCD3053 (L) "OTL"			UBMM2013 (L) "OTL" UBMMH1013 (T) LDK2 UALE1083 (T) N001			UCCD2043 (T) N003			
Friday	UCCD3223 (P3) N104 UCCD3053 (T1) N003	UCCD3013 (T2) N001	UCCA3053 (L) "OTL" UCCB2333 (T) N003	UCCD3013 (L) "OTL"	FRIDAY PRAYER		UCCD2043 (L) "OTL"						

"OTL" - Online Teaching and Learning

CORE

UCCA3583	PROJECT I			
UCCD3053	INFORMATION TECHNOLOGY PROFESSIONAL ETHICS	3L+1T	Mr Tey Chee Chieh	L+T
UCCD3223	MOBILE APPLICATIONS DEVELOPMENT	2L+2P	Mr Tan Chiang Kang	L+P
			Ts Saw Seow Hui	P

UCIA-05 (Choose One)

UALE1083	BASIC PROFESSIONAL WRITING	2L+1T	Ms Thian Pik Hang	L+T
			Ms Christina Ong Sook Beng	T

Figure 3.2.3 Timetable of Y3T1 Students from Information Systems Engineering Programme

After that, the information such as the program structures of students were studied. The information in the course registration of UTAR which consist of the lecture hour, tutorial hour, practical hour, the class event size and class event number were studied too.

3.3 Verification Plan

After implementing the system, the system testing will be conducted to ensure the performance and correctness of the system in producing the timetable based on users' data inputted. To verify the correctness of the system functions, verification testing will be conducted.

3.3.1 Test Plan for Hard Constraints

To ensure the timetable produced by this system works as desired and follow the hard constraint, test cases are introduced as in table 3.3.1.1 below.

Test Case	Test Case Description	Test Data	Expected Result
1.	Each student can only take one class at a time	Students' timetables	From the students' timetable produced, it is made sure that in each timeslot, there are no more than two classes. However, if there found more than two class events in one timeslot, if the class events are the same where students can choose one of them to attend, it is considered no violation.
2.	Each lecturer can only take one class at a time	Lecturers' timetables	From the lecturers' timetable produced, it is made sure that in each timeslot, there are no more than two classes.

3.	Each classroom can only have one class assigned at a time	Classrooms' timetables	From the classrooms' timetable produced, it is made sure that in each timeslot, there are no more than two classes.
4.	Each class should be taught by one lecturer at one time	Classrooms' timetables	From the classrooms' timetable produced, it is made sure that the classes in each timeslot contain only one lecturer's information.
5.	The capacity of the classroom should be large enough to hold the students' size in each class	Classrooms' timetables	From the classrooms' timetable produced, it is made sure that for classroom with capacity less than the class size cannot have class filled in. (e.g., LESSON has class size of 30, classroom with less than 30 of room capacity cannot have any LESSON class)
6.	The main subjects' classes should not be conducted at the same time	Students' timetables	From the students' timetable produced, it is made sure that in each timeslot, there are no more than two LECTURE classes. If there found more than two TUTORIAL/ PRACTICAL class events in one timeslot, if the class events are the same where students can choose one of them to attend, it is considered no violation.
7.	Specific room is required to be considered (tutorial classroom/lab)	Classrooms' timetable	From the classrooms' timetable produced, it is made sure that the TUTORIAL/PRACTICAL classes should be located inside the classroom with classroom types of TUTORIAL/ PRACTICAL respectively.

8.	The available periods per day is maximum at 10	All timetables	This constraint is initially defined. From the timetable produces, there should not have any timetables having more than 10 periods, assuming that 1 period is 1 hour.
9.	Each lecturer's working hours are limited during a week	Lecturers' timetables	From the lecturers' timetables, for the lecturers teaching the same courses, the number of timeslots taken by each lecturer should be the same/the differences are only one
10.	Friday session stops by 12pm to 2pm	All timetables	From all timetables, it is observed that for timeslot of 1200-1400 on Friday, the slot should be empty
11.	The number of students in Online Teaching and Learning (OTL) lecture classes can be unlimited	Classrooms' timetables	In the classrooms' timetables produced, for the OTL classes, there is no need to specify the number of students to attend the classes
12.	Classes which require physical attempt cannot be in OTL mode	Classrooms' timetables	For the classes with specific requirement to be attended physically, physical classroom should be assigned to it

Table 3.3.1.1 Test Plans for Hard Constraints

3.3.2 Test Plan for Soft Constraints

To test for soft constraints, the test plan is based on the condition where the soft constraints must be fulfilled completely. The table below shows the soft constraints test plans and their results.

Test Case	Test Case Description	Test Data	Expected Result
1.	Classes should not be held after 6pm	All timetables	From all the timetables produced, there should not be classes held after 6pm.
2.	General course should be opened to all faculty students	Students' timetables	From the students' timetables produced, any general course will not be abandoned from any student group
3.	No subject has more than one lecture a day	Courses' timetables	From the course' timetable produced, it is made sure that in each day, there are no course that has more than one lecture class.
4.	The daily teaching sessions of the lecturer should not exceed a specified	Lecturers' timetables	From the lecturers' timetable produced, it is made sure that the lecturers have no more than 4 class events to handle

	number of periods		

Table 3.3.2.1 Test Plans for Soft Constraints

3.3.3 Test Plan for Hybrid Classes Generation

The table below is the test plan for the Hybrid Classes Generation which used to indicate whether the hybrid timetable generation is workable by making sure there are no errors such as the repetition of the same class events in both physical and OTL mode, different distribution of the classes for the modes based on user's database data, etc.

Test Case	**Test Case Description**	**Test Data**	**Expected Result**
1.	The lecture class should either be conducted full in OTL mode or full in physical mode	Courses' timetables	From the courses' timetables produced, the lecture classes are either in full OTL mode or full physical mode.
2.	The tutorial and practical classes study mode can be dynamically decided by the users.	Courses' timetables	From the courses' timetables produced, the tutorial or practical classes can be all in OTL mode or physical mode, or half in OTL mode or physical mode, or in any other ratio of study mode distribution.

Table 3.3.3.1 Test Plans for Hybrid Classes Generation

3.3.4 Test Plan for Timetable Generation

In this section, the test plan is to do the experiment related to the speed of the timetable generation using genetic algorithm. Different number of class events are used to conduct the test plan, with the manipulated variable of the number of constraints selected. Here, the constraints type includes only the constraints that are selectable by the user. The resources mentioned later is the sum of the timeslots of all classrooms available.

Test Case	Test Case Description
1.	Generate the timetable with small class events amount (~10% of the resources) and 2 constraints involved
2.	Generate the timetable with large student group amount (~40% of the resources) and 2 constraints involved
3.	Generate the timetable with small student group amount (~10% of the resources) and all constraints involved
4.	Generate the timetable with large student group amount (~40% of the resources) and all constraints involved
5.	Generate the timetable with class events more than the resources available

Table 3.3.4.1 Test Plan for Timetable Generation

3.4 System Functionalities of the Timetable Scheduling System

3.4.1 Users' Actions and Steps in Using the Timetable Tool

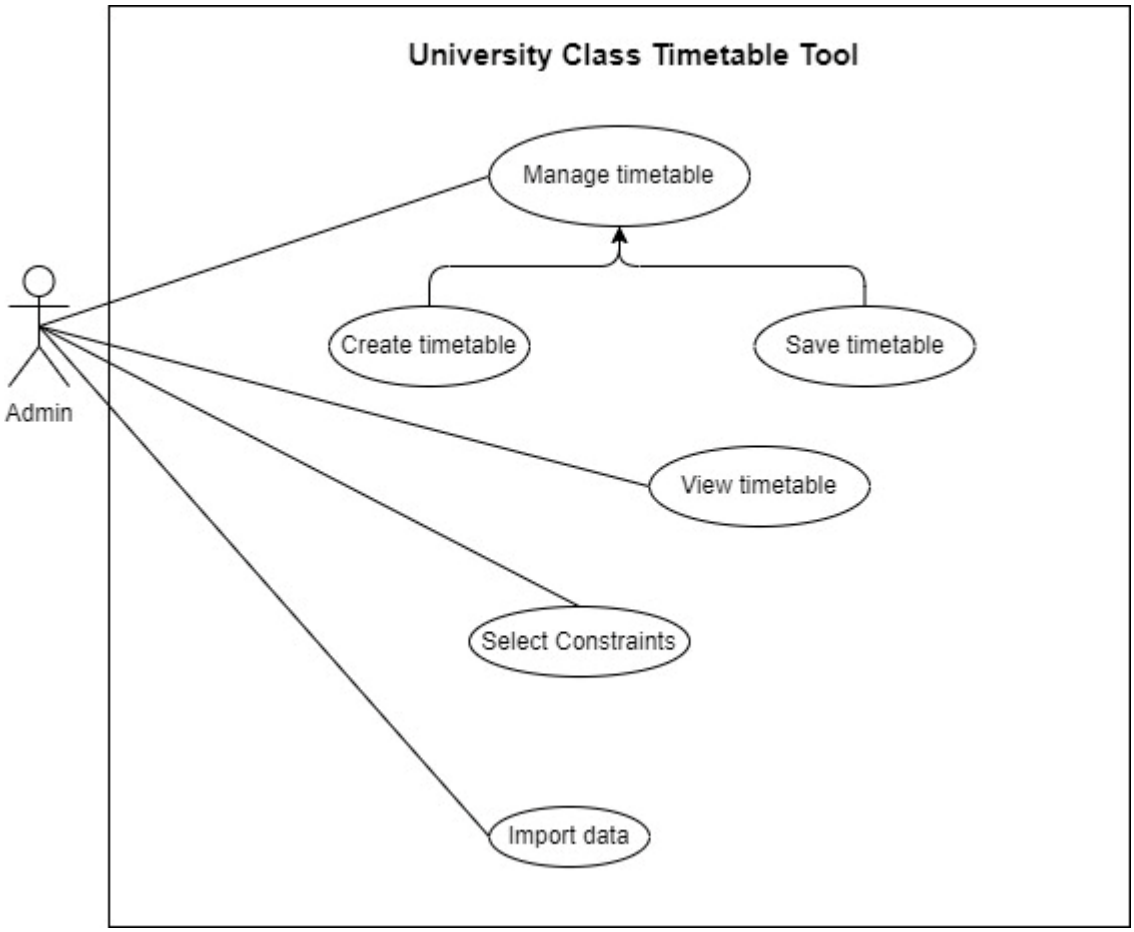


Figure 3.4.1.1 The system functionalities of the class timetable scheduling system

The figure 3.4.1.1 above is used to model the functionalities and interactions between the system users with the system's functions. The user (administrator) can manage the timetable by creating and saving. Besides, the admin can also import the data required for the timetable scheduling. After that, the admin can select the constraints required to generate the timetable too.

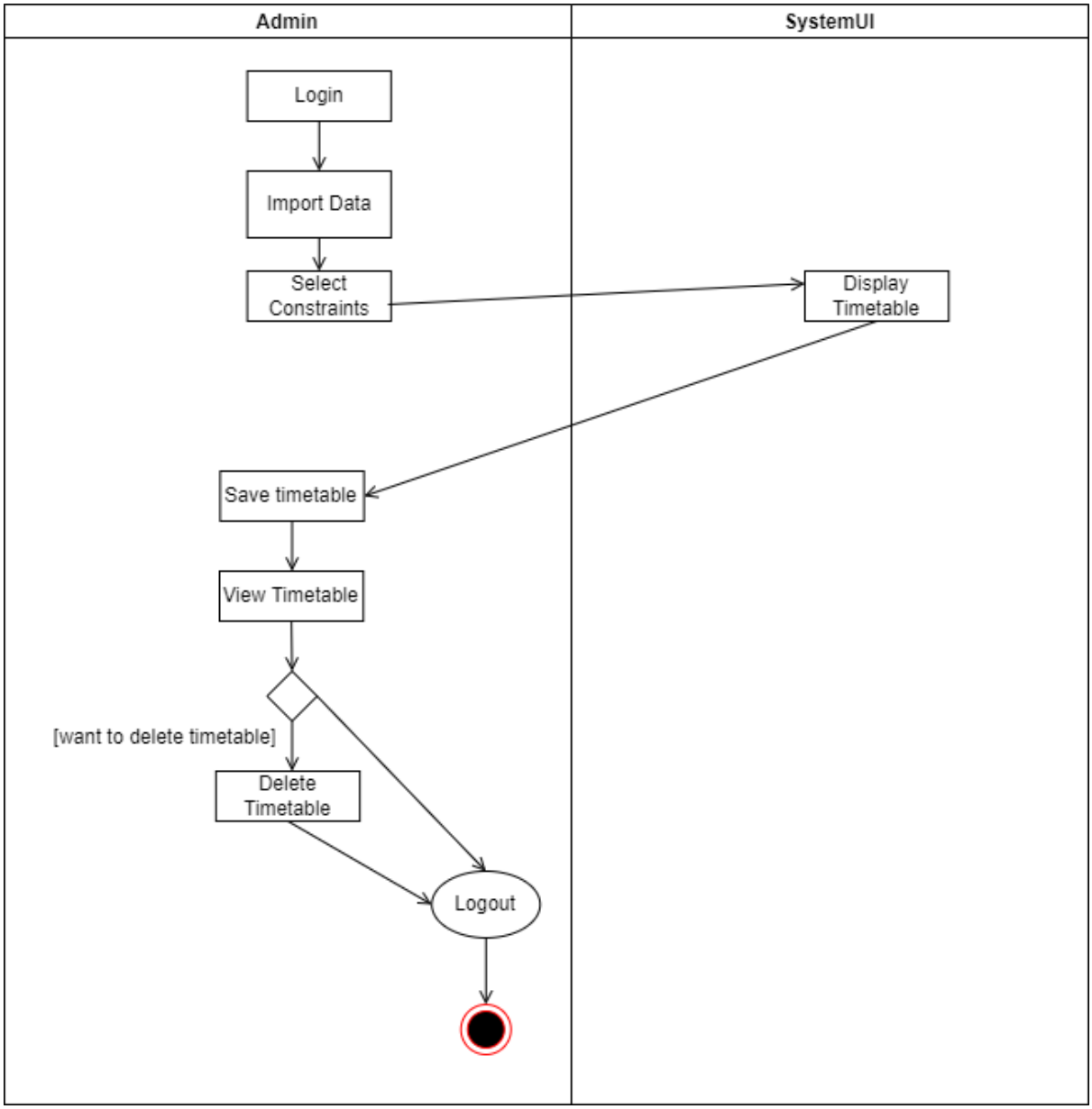


Figure 3.4.1.2 Design of Steps in Managing the Timetable Scheduling

Figure 3.4.1.2 above shows the process of managing the timetable by an administrator. After the admin logins to the system, he /she can import the data required by the timetable scheduling process. Then, the systemUI will display the timetable created. After that, the admin can save and view the final version of the timetable. After that, if the admin wants to delete the timetable, he/she can delete the final version of the timetable then log out. If no, the admin can log out of the system directly.

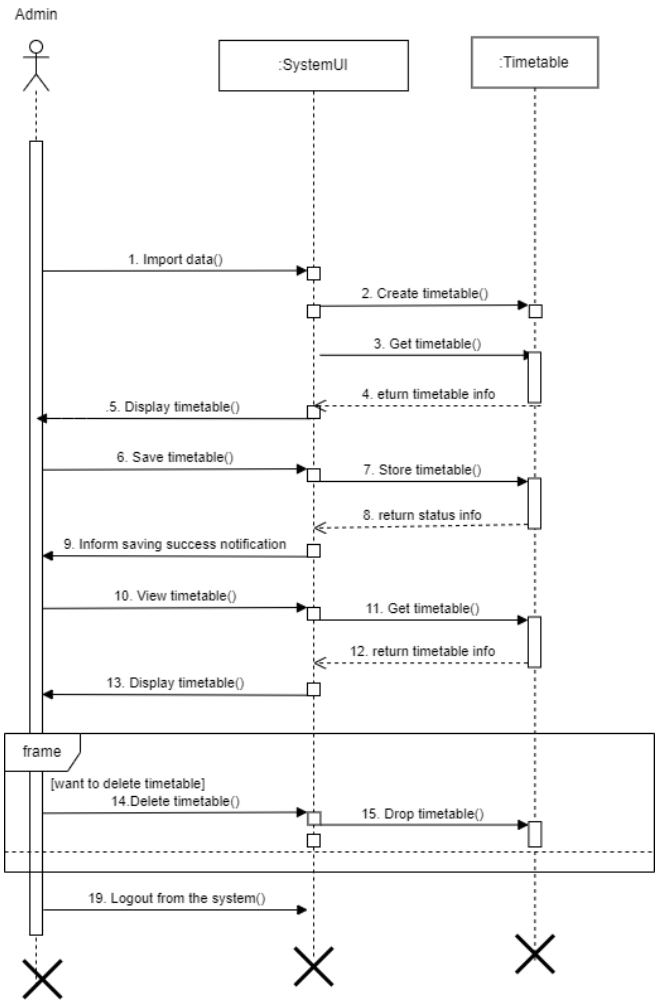


Figure 3.4.1.3 Design of Steps in Timing Manner in Managing the Timetable Scheduling

Figure 3.4.1.3 above shows the Design of steps in timing manner for the admin in managing the timetable scheduling. After the system user (admin) logins to the system. The user can import the data required by the timetable scheduling process and the systemUI will process the data user managed. Then, the system will generate the timetable and the systemUI gets the timetable from the object 'Timetable'. The 'Timetable' object will then return the timetable info to the systemUI and the systemUI will display the created timetable. Then, the admin can save the timetable and the systemUI will now display status of the timetable saved. The admin can view the timetable which is the final version of the data. After that, if the admin wants to delete the timetable or want to release the timetable, he/she can delete the final version of the timetable or release the final version of the timetable. If no, the admin can log out of the system.

3.4.2 The Constraints used for the Class Timetable Scheduling System

3.4.2.1 Hard Constraints

I. Each student can only take one class at a time
II. Each lecturer can only take one class at a time
III. Each classroom can only have one class assigned at a time
IV. Each class should be taught by one lecturer at one time
V. The capacity of the classroom should be large enough to hold the students' size in each class
VI. The main subjects' classes should not be conducted at the same time
VII. Specific room is required to be considered (lab)
VIII. The available periods per day is maximum at 10
IX. Each lecturer's working hours are limited during a week
X. Friday session stops by 12pm to 2 pm
XI. The number of students in Online Teaching and Learning (OTL) lecture classes can be unlimited
XII. Classes which require physical attempt cannot be in OTL mode

3.4.2.2 Soft Constraints

I. Classes should not be held after 6pm
II. General course should be opened to all faculty students
III. No subject has more than one lecture a day
IV. The daily teaching sessions of the lecturer should not exceed a specified number of periods

3.4.3 The Design of Genetic Algorithm for the Class Timetable Scheduling System

In this section, the processes of the timetable scheduling will be explained and described using the genetic algorithm in the steps of initialization, fitting, selection, crossover and lastly mutation. Figure 3.4.3.1 below shows the flow of the timetable scheduling using genetic algorithm.

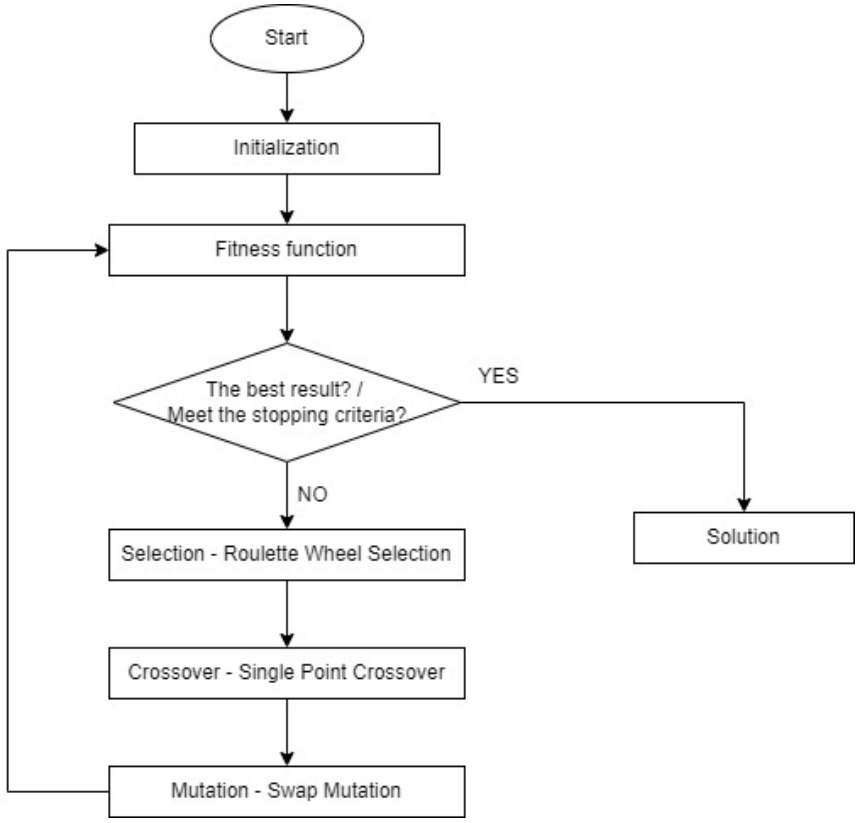


Figure 3.4.3.1 Flow of the Timetable Scheduling Using Genetic Algorithm

The infeasible timetable will be produced at the very first time through the initialization. After that, evaluation of the timetable events (chromosomes) produced will be conducted using the fitness function. If the timetable produced is not the best result or the timetable scheduling process does not meet the stopping criteria, the timetable produced will be going to the selection phase where the Roulette Wheel Selection will be performed. After the parent chromosomes are being selected in selection phase, the single point crossover will be performed to cross over the selected parent chromosomes. Finally, swap mutation performed to mutate the child chromosomes produced in crossover phase. Then, evaluation of the chromosomes will be performed again. After undergoing the loops of fitting, selection, crossover, and mutation, if the feasible timetable will be produced at the end or the scheduling processes have reached the stopping criteria, the result of the timetable will be produced.

In this project, the timetabling tool will be functioning follow the design of the algorithm stated above. When the users insert the data into the timetabling tool, the data such as classroom data, student data, lecturer data and the course data will be used to form the class events, and the class events will be used to form the chromosomes. After

that, the consequent phases of the algorithm will be carried out. If the chromosomes produced in each generation is not fit enough, then the timetable generating activities will not be terminated. The selection of the good performance timetable will be done by referring to their fitness value using the Roulette Wheel Selection. After that, for the timetables chose from the selection phase, the single point crossover will be performed on them where a crossover point will separate the chromosomes into two parts, then those two parts will be interchanged to produce to children chromosome. Lastly, to ensure the genetic diversity of the system, swap mutation will be carried out to swap the random timeslots inside the timetable. Lastly, after going through these timetable generations, if it is found that the timetable produced has achieved the optimal fitness value, then the looping will be escaped and the optimal timetable will be produced.

3.4.3.1 Initialization

The initialization phase is the phase of initial random population generation. In this phase, sets of events data will be used to form the timetable events. Then, the events will be used to fill the timeslots available in the timetable to form a population. The timeslots available in the timetable are the slots that the event can fit in. In this system, the days to conduct the events per week, the slots to conduct the event per day, and the classroom to hold the events will be the resources to build up the timetable. Then, for each of the classrooms, there are same number of timeslots for the events to fit inside.

The event data to be considered in this timetable scheduling system are the classrooms, courses, lecturers, and students (group). The variables for the event data are showed below:

Classroom: classroom name, classroom type, classroom capacity

Course: course code, course name, number of lectures, number of tutorials, number of practical

Lecturer: lecturer name, a list of courses that the lecturer is assigned to teach

Student (group): student name. student amount, a list of courses that the student group need to attend

In this system, the event will be storing the information of:

- the type of event (L: lecture, T: tutorial, P: practical, O: OTL)
- the number of people to attend the event
- the lecturer for that course in the event.
- the course of the event
- the student to attend the event
- Event group id

For example:

Event1: <L, 120, DrTan, UCCD1004, CSY1S1, 1, 1>

Event2: <P, 20, DrLee, UCCD1024, CSY1S2, 7, 7>

Event3: <P, 20, DrLee, UCCD1024, CSY1S2, 8, 7>

Event4: <P, 20, DrLee, UCCD1024, CSY1S2, 9, 7>

Event5: <T, 30, DrChan, UCCM1153, CSY1S1, 10, 8>

Event5: <O, 30, DrChan, MPU3113, CSY1S1, 10, 11>

For each event, a unique id will be generated for them. The unique id assigned will be used to fill up the chromosomes in each timeslot of the chromosome. The structure of the chromosome will be showed in table 3.4.3.1.1.

As each of the course may have different event, which are lecture, tutorial, or practical for the student to attend, hence different events will be created for each of these special situations. For example, each practical class event can only hold up to 20 students as following the practical lab size, and each tutorial class event can only hold up to 30 students as following the tutorial class size. For the lecture class size, it is always assumed that the class can hold all the students who attend it. Under these circumstances, the student group will need to form several events for one tutorial class or one practical class. For these events generated under this circumstance, an event group id will be assigned to them so to group them together.

For example, the events below are different events with different event id but with the same event group id.

Event2: <P, 20, DrLee, UCCD1024, CSY1S2, 7, 7>

Event3: <P, 20, DrLee, UCCD1024, CSY1S2, 8, 7>

Event4: <P, 20, DrLee, UCCD1024, CSY1S2, 9, 7>

After all the events have been generated, they will then be randomly filled in the available timeslots in each of the classrooms available, forming classrooms' timetables. After filling in all the events in all the classrooms, the set of classrooms together will form a timetable, which is chromosome in this algorithm. The structure of the chromosomes was merely made up from the sets of timeslots generated, which contains of the id of the class events, and the chromosome length will be the classroom size * timeslots per day * days per week. Then, the genetic algorithm will then work on the timeslots and the events placed inside the timeslots.

In addition, for the special structure of the timetable such as "special timeslots in some day are blocked", special condition will be set while placing the events into that particular timeslot. Special constraint function will also be used to ensure the timetable produced is obeying that constraint while generating the timetable using the Genetic Algorithm.

The chromosome structure will be showed as table 3.4.3.1.1 below:

Chromosome A	0	1	0	3	0	2	4	5
Chromosome B	2	1	0	0	0	0	8	15
Chromosome C	17	0	4	15	5	0	3	0
Chromosome D	25	0	0	6	0	0	0	0

Table 3.4.3.1.1 The Chromosome Structure Designed in the System

The table above just showed a small part of the chromosomes. From the table above, each cell represents a gene that all genes in the row form a chromosome. The 0 represents that there is no any class event assigned to that slot. Whilst the cell with

number represents the event id where that class event is being assigned to the slot. The assignation of the class events into the slots will be made in a randomly manner.

Each of the chromosome represent one set of timetable which consists of every classroom timetables. Then, the timetables together will be forming a population. The timetables as chromosomes will be produced randomly until reaches the maximum population size, which is set to 100 in this system. The later works of the genetic algorithm will be working on this structure of chromosomes and makes changes on the timeslots where the class events are assigned to.

3.4.3.2 Fitting

After that, the fitness value of the population's chromosomes will be calculated. The fitness value of the chromosome represents the feasibility of the timetable session being produced. It helps the algorithm to make the better determination towards the better solution. The more the violations towards the hard constraint, the lower the fitness value it has. In this project, for each constraint violated, the fitness score will be decreased by 1. On the other hand, the maximum fitness score to get will be 0. When the timetable has a greater number of clashes towards the constraints, for example, one lecturer teaches two class at the same time, the fitness value will be lower. Most of the constraints' violations will be monitored using functions, meanwhile there are some constraints will be directly followed by the timetable produced based on the structure of the data inputted or chromosomes.

3.4.3.3 Selection

The Roulette Wheel Selection will be performed in this system. Roulette Wheel Selection is a method which provide the selection method based on the probability of the selecting members. The probability of a timetable sessions to be selected is proportional to its fitness value. From the fitness value gained from the fitting phase, the timetable sessions produced are given the fitness values which can act as the probabilities of each of them to be selected in Roulette Wheel Selection. This method ensures that even the timetable sessions with the lower priority have the chances to be selected to participate in the future processes to produce optimal solution. On the other

hand, keeping the higher selection probability of the high fitness value timetable sessions can ensure the best solution to be sustained in the scheduling process. In each iteration, the parent chromosomes will be chosen and with the help of cross over phase, the new population will be produced. The example below shows the Roulette Wheel Selection method used to select the members for next crossover phase.

Let the chromosomes have different fitness value:

Chromosomes	Fitness value
A	-4
B	-2
C	-1
D	-3

Table 3.4.3.3.1 The Example Chromosomes with Their Fitness Values

From the table above, the total fitness will be 10.

The normalized fitness value will be showed in the table below:

Chromosomes	Fitness value	Normalized Fitness value	Percentage
A	-4	-0.4	10%
B	-2	-0.2	30%
C	-1	-0.1	40%
D	-3	-0.3	20%

Table 3.4.3.3.2 The Example Chromosomes with Their Normalized Fitness Value and the Percentage of Them to be Selected

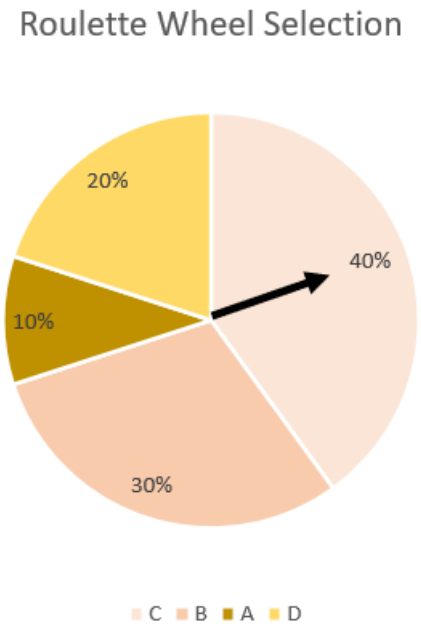


Figure 3.4.3.3.2 Pie Chart on Percentage of Chromosomes to be Selected

From the figure 3.4.3.3.2 above, we can see that Chromosome C with the highest probability to be chosen stands a greater opportunity to be the parent chromosomes in the next phase. The selection opportunities for the chromosomes are C>B>D>A.

3.4.3.4 Crossover

In the crossover phase, the single point crossover will be applied. This kind of crossover ensures that the chromosomes or the timetable sessions are broken at one point, and then crossing occurs. Instead of cross overing the timetable sessions wholly, each row of the timetable session is cross overed separately. In this project, from the point selected onwards, the parameters in each timetable sessions will be exchanged between the pair of parent chromosomes, and produce the offspring. The crossover pairs will be selected randomly from the whole population. The figure 3.4.2.4.1 below shows the procedures of single point crossover in the timetabling system.

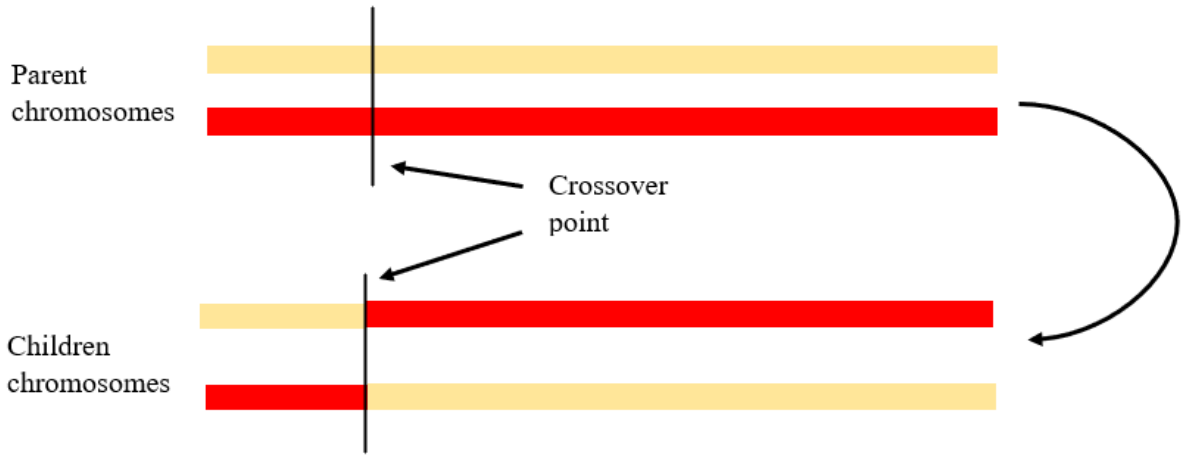


Figure 3.4.3.4.1 Single Point Crossover Model

Then, the single point crossover will be showing the result as in the table 3.4.3.4.1 below. From the table, the front and back parts of the chromosome A and chromosome B will be interchanged at the cross over point, and then generate two child chromosomes A and B. The interchanged parts are represented in different colours.

Parent chromosome A	0	1	0	3	0	2	4	5
Parent chromosome B	2	1	0	0	0	0	8	15
Child chromosome A	0	1	0	0	0	0	8	15
Chile chromosome B	2	1	0	3	0	2	4	5

Table 3.4.3.4.1 Example of Single Point Crossover in the System

3.4.3.5 Mutation

In this phase, swap mutation is used to mutate the timetable sessions in the unit of timeslots. Using swap mutation, the timeslots will be gone through where each timeslot will have a probability of being mutated (5% in this system). The mutation rate to be used will be defined from time to time to find the most suitable one. If the current timeslot is chosen to be mutated, other timeslot will be found randomly, and the swapping between the randomly found timeslot with the current timeslot will be

performed. The values at those positions will be interchanged. The figure 3.4.2.5.1 below shows the swap mutation operation in this system.

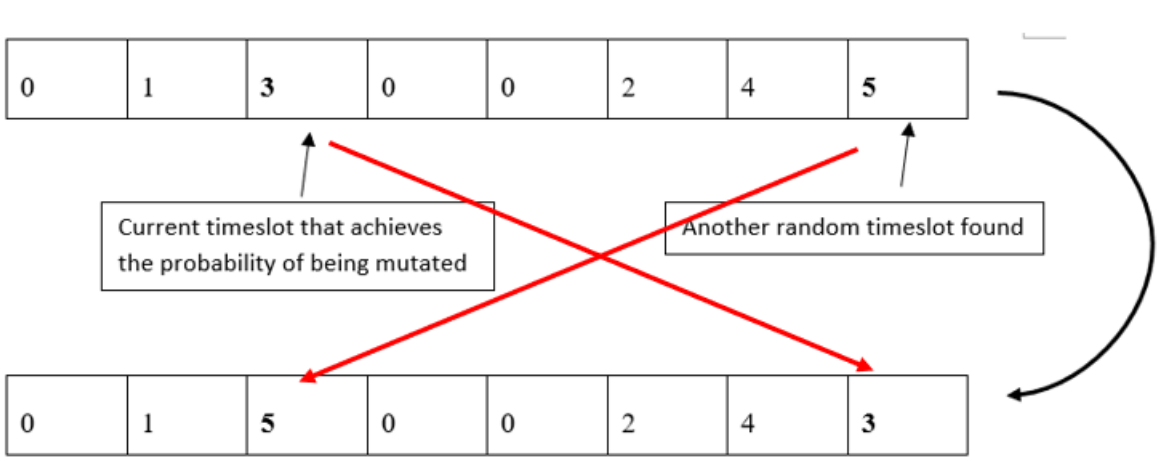


Figure 3.4.3.5.1 Swap Mutation Operation in this System

From the figure 3.4.3.5.1 above, at the timeslot where the eventid is 3, that particular timeslot has achieved the probability of being mutated, then random timeslot in the chromosome will be selected and swap with the timeslot with eventId 3. As a result, swap mutation occurred.

3.5 Hardware and Software requirements

3.5.1 Hardware

The hardware used in developing the system is a computer. A laptop computer is used to code the timetabling tool using the Java programming language for the scheduling part, as well as the xml to design the graphical user interface of the system.

Description	**Specifications**
Model	HP Pavilion Laptop 15-eg
Processor	11th Gen Intel(R) Core(TM) i5
Operating System	Window 11 64-bit

Graphic	NVIDIA® GeForce® MX250 graphics
Memory	8GB RAM
Storage	512GB SSD

Table 3.5.1.1 Hardware Used in this Project

3.5.2 Software

The main software used to develop the system includes Eclipse IDE for Java Developer 2021-12. Other software required are listed below:

Description	**Tools**
Operating system	Window 11 Home Single Language
Integrated Development Environment	Eclipse IDE for Java Developer 2021-12
Support GUI	Java Swing
Support programming language	Java
Local Server for Database	XAMPP
Database	MySQL

Table 3.5.2.1 Software Used in this Project

Chapter 4

System Design

4.1 System Architecture Design

In this project, the system architecture to be used is the JDBC Architecture in two tier model. In this architecture, the java application (timetabling tool) will be communicating directly to the MySQL database. In between them, JDBC driver acts as a middleman to enable the communications[47].

When the application sends the data query to the database, the reply of the queries will be sent back from the database to the application. The figure 4.1.1 below shows the system architecture of the system which consist of application and database as the major actor, with the JDBC and MySQL driver act as the middleman to direct the communications.

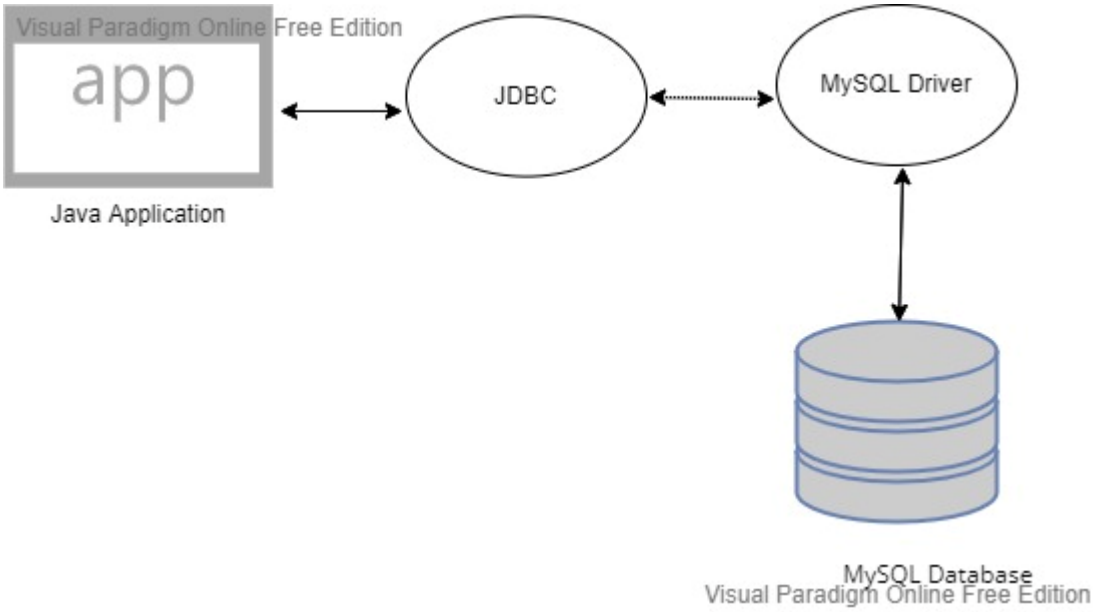


Figure 4.1.1 The System Architecture of the System

4.2 Visualization of Timetable Generation Steps

From the current stage of implementation, the data from the Computer Science Programme Student Year 1 Trimester 1, Year 1 Trimester 2, and Year 1 Trimester 3 are being used to test the effectiveness of the proposed solutions. The csv file of the

classrooms' timetables, student groups' timetables, and lecturers' timetables were produced. The result obtained in the figure 4.2 shows the process of the scheduling and result in console.

```
#GENERATIONS: 1220 BEST FITNESS: -2
#GENERATIONS: 1221 BEST FITNESS: -2
#GENERATIONS: 1222 BEST FITNESS: -2
#GENERATIONS: 1223 BEST FITNESS: -2
#GENERATIONS: 1224 BEST FITNESS: -2
#GENERATIONS: 1225 BEST FITNESS: 0
============ Room: N001 Capacity: 40 ============
[    -    ][    -    ][ 136 - CSY1S3 UCCD1013 LESSON MA ][    -    ][    -    ]
[    -    ][    -    ][    -    ][    -    ][    -    ]
[    -    ][    -    ][    -    ][    -    ][    -    ]
[ 43 - CSY1S1 UCCM1153 LESSON LKH ][    -    ][ 37 - CSY1S1 UCCD1143 LESSON KXY ][    -    ][    -    ]
[    -    ][    -    ][    -    ][ 187 - CSY1S3 UCCM1363 LESSON SPC ][    -    ]
[    -    ][    -    ][    -    ][ 188 - CSY1S3 UCCM1363 LESSON LFW ][    -    ]
[    -    ][    -    ][    -    ][    -    ][    -    ]
[    -    ][    -    ][    -    ][ 92 - CSY1S2 UCCD1013 LESSON MO ][    -    ]
[    -    ][    -    ][    -    ][    -    ][    -    ]
[    -    ][    -    ][    -    ][ 178 - CSY1S3 UCCM1353 LESSON LSJ ][    -    ]
============ Room: N002 Capacity: 50 ============
[    -    ][    -    ][    -    ][    -    ][    -    ]
[    -    ][    -    ][    -    ][    -    ][    -    ]
[    -    ][    -    ][    -    ][    -    ][    -    ]
[    -    ][    -    ][    -    ][    -    ][    -    ]
[    -    ][    -    ][    -    ][    -    ][    -    ]
[    -    ][ 98 - CSY1S2 UCCD1143 LESSON LSP ][    -    ][    -    ][ 100 - CSY1S2 UCCD1143 LESSON LSP ]
[    -    ][    -    ][ 29 - CSY1S1 UCCD1013 LESSON MA ][    -    ][    -    ]
[    -    ][ 101 - CSY1S2 UCCD1143 LESSON KXY ][    -    ][    -    ][    -    ]
[ 180 - CSY1S3 UCCM1353 LESSON LSJ ][    -    ][    -    ][    -    ][    -    ]
[    -    ][    -    ][    -    ][    -    ]
============ Room: N003 Capacity: 50 ============
[    -    ][    -    ][    -    ][    -    ][    -    ]
[    -    ][    -    ][ 42 - CSY1S1 UCCM1153 LESSON LSJ ][    -    ][    -    ]
[    -    ][    -    ][    -    ][    -    ][    -    ]
[    -    ][ 109 - CSY1S2 UCCM1153 LESSON LKH ][    -    ][    -    ][    -    ]
[ 108 - CSY1S2 UCCM1153 LESSON LSJ ][    -    ][    -    ][    -    ][ 105 - CSY1S2 UCCM1153 LESSON LKH ]
```

Figure 4.2.1 Process of the Scheduling and Result in Console

From the figure 4.2.1 above, it shows the fitness values in each generation and when the fitness value of a certain generation comes to the value 0, the generations stop and the sample results will be printed in the console.

4.3 Interface Design

In this section, the interface design of the system will be showed with elaborations.

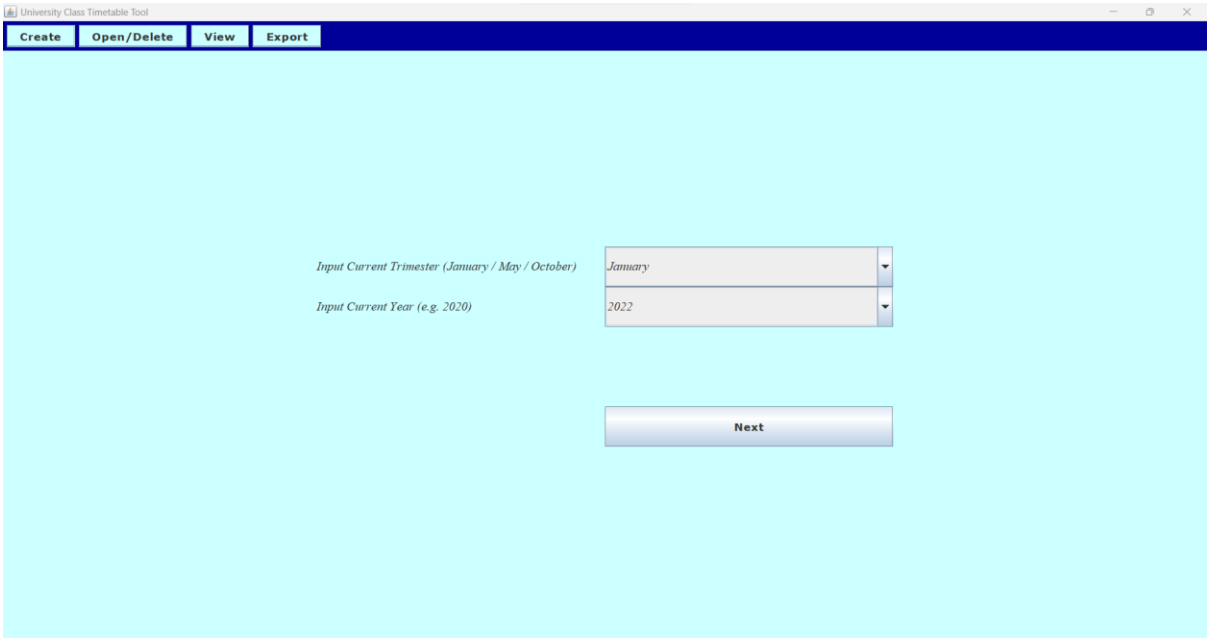


Figure 4.3.1 Main Interface of the System

The figure above shows the main interface of the system. Upon user opens the system, they will see the interface above. The user is allowed to select the trimester and year he may like to create the timetable. After selecting, the user can proceed by clicking the next button. Besides that, the user can also select the open/delete button from the navigation bar above. The user is not allowed to click on view and export button in the navigation bar as there is no timetable generated or opened in the system.

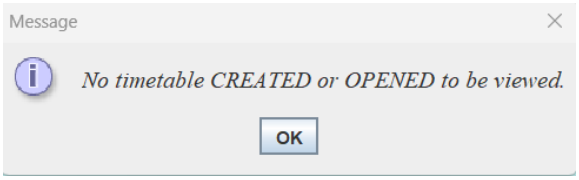


Figure 4.3.2 Alert when Open View Interface

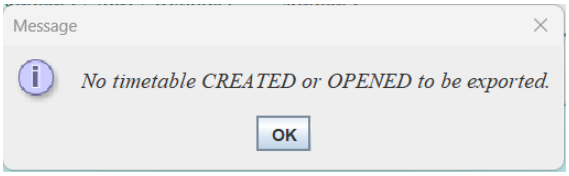


Figure 4.3.3 Alert when Open Export Interface

Alert messages above will be showed if the user clicks on view or export button before generating new timetable or opening the existing timetable.

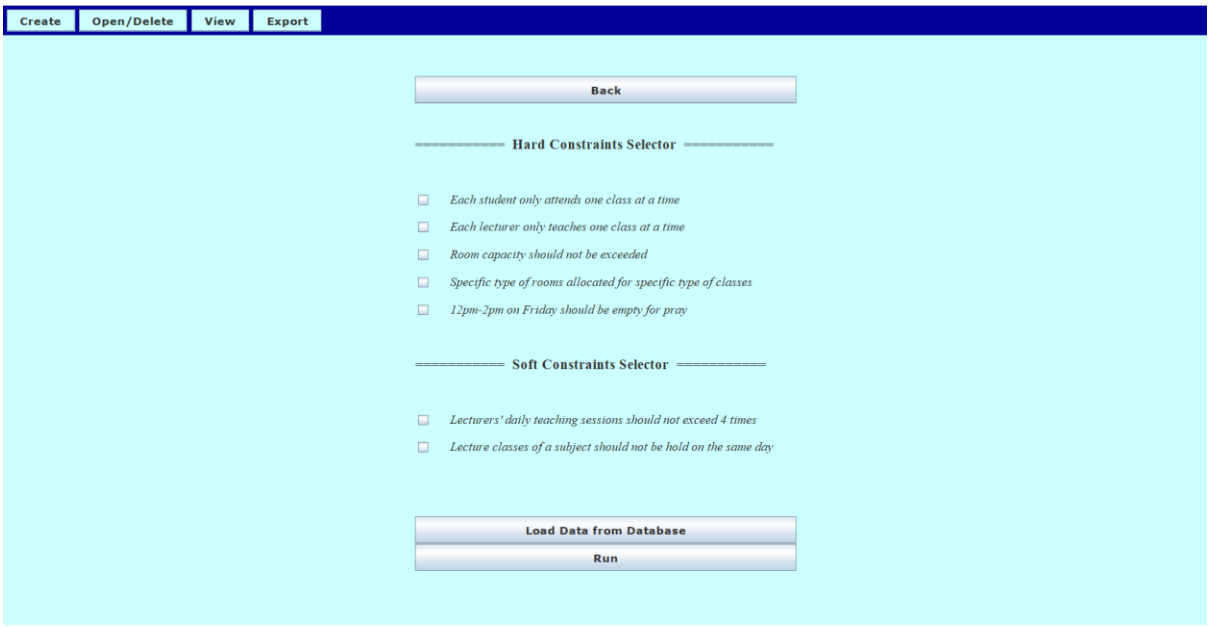


Figure 4.3.4 Constraints Selection Page

After the user clicks next on the main panel, he will be directed to the constrain selection page. In this page, the user is allowed to select the constraints he wants to fulfil in generating the timetable. There are 5 main hard constraints and 2 soft constraints. This is due to the reason that some constraints are automatically fulfilled by the timetable structure. After selecting the constraint he wants, the user can load the data from the database. Then, the user can click the run button to generate the timetable.

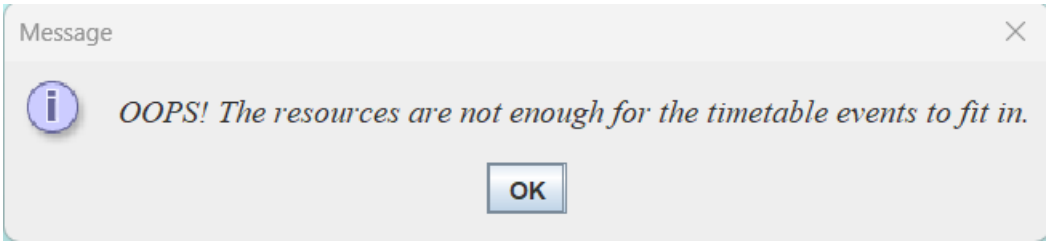


Figure 4.3.5 Alert when Events Number Exceed Resources

If the events number exceed the resources provided, the above message will pop up and then the run button will become unclickable for user to proceed.

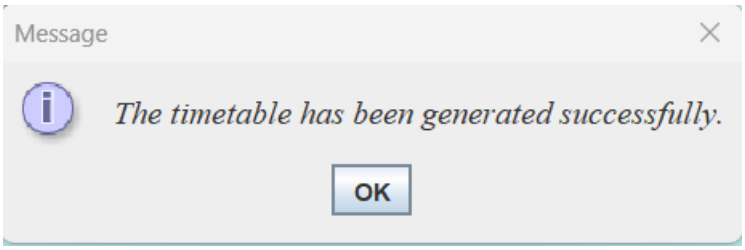


Figure 4.3.6 Message when Timetable Generated Successfully

After the timetable is created successfully, the alert message will notify the user.

Figure 4.3.7 View Page of the System

Now, the user can view the generated timetable. The user can see the scroll panes consist of different timetables, which are classroom timetable, student timetable, lecturer timetable, and course timetable. At the bottom left area, the user can choose to save the current timetable or modify the current timetable.

Figure 4.3.8 View Page of the System (2)

For different classes in the class timetable, the user can click on certain class to view only that class. The figure above shows the 'LDK1' class only after user clicks the LDK1 button.

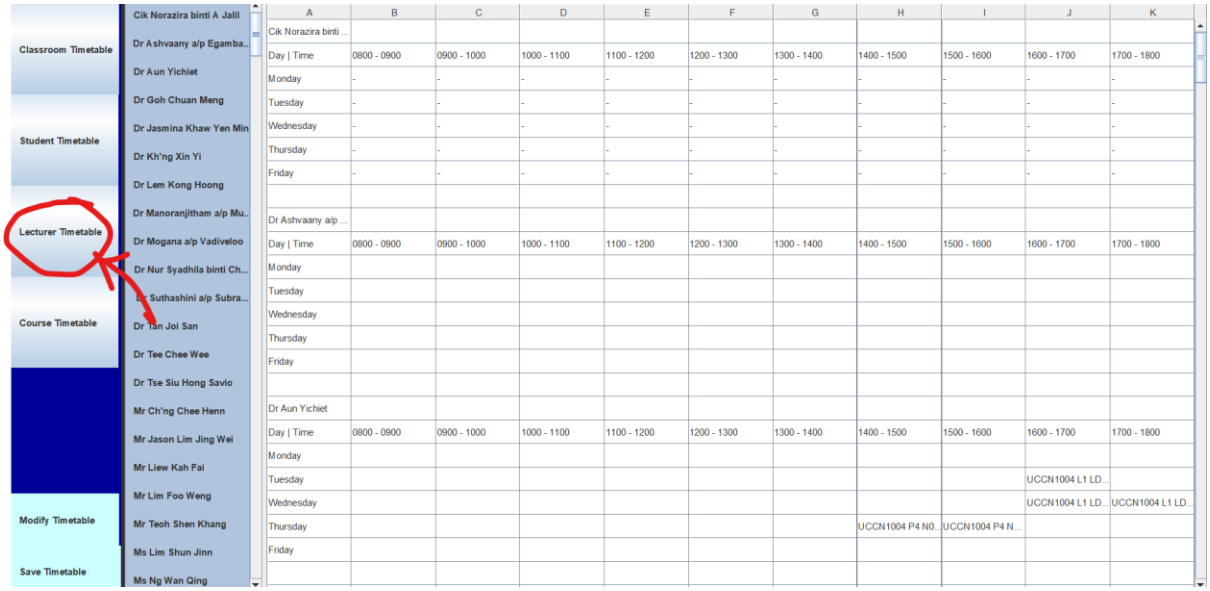


Figure 4.3.9 View Page of the System (3)

If the user wants to see the lecturer timetable, he can click on the lecturer timetable button. Same as the classroom timetable, the user can select to see the timetable of only one lecturer by clicking the lecturer's name button. The same logic goes for the student timetable and the course timetable.

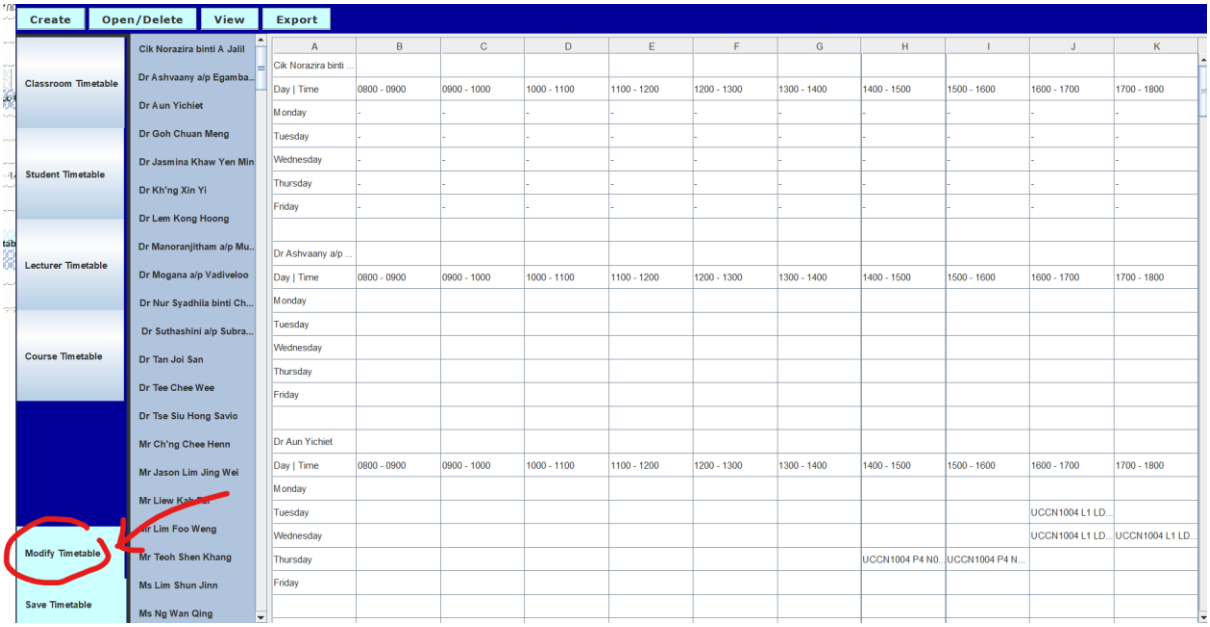


Figure 4.3.10 Clicking on Modify Timetable Button

If the user wants to modify the timetable, he can click the modify timetable button at the bottom left part. Then, the user will be directed to the modification panel.

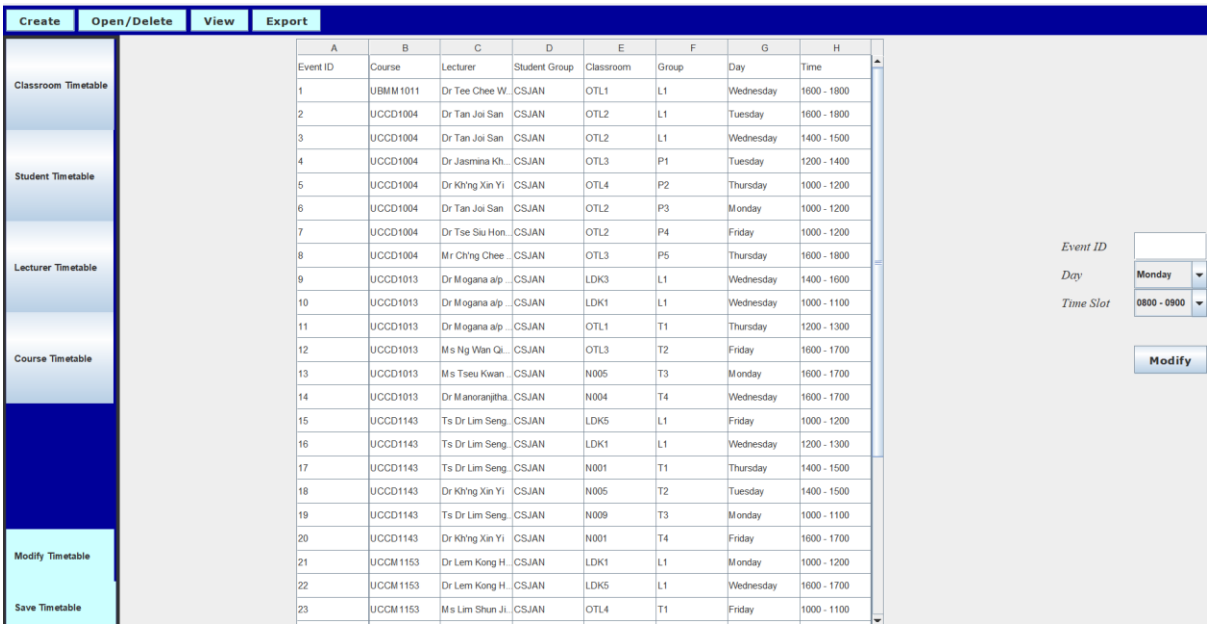


Figure 4.3.11 Modify Page of the System

The figure above shows the interface of the modification panel. The class events are listed for the user to select. After the user select certain class event, the correspond class event ID will be showed in the Event ID text field at the right side of the interface. Then, the user can select the day and timeslot he is interested to change. The timetable

generator will then check if the modification violates the constraints the user selected in the previous step. Let say the user wants to modify the first class event from Wednesday 1600-1800 to Monday 0800-1000, he first clicks the class event on the list, then click on the modify button at the right side of the interface.

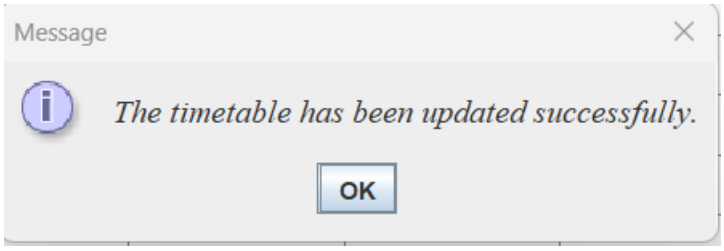


Figure 4.3.12 Message shows Timetable Update Successfully

The alert message will be popped up. Saying that the timetable has been update successfully.

Figure 4.3.13 Message shows Timetable Update Failed

If the modification of the timetable violates the hard constraints selected by the user, the message above will pop up saying that the modification of the timetable is unsuccessful.

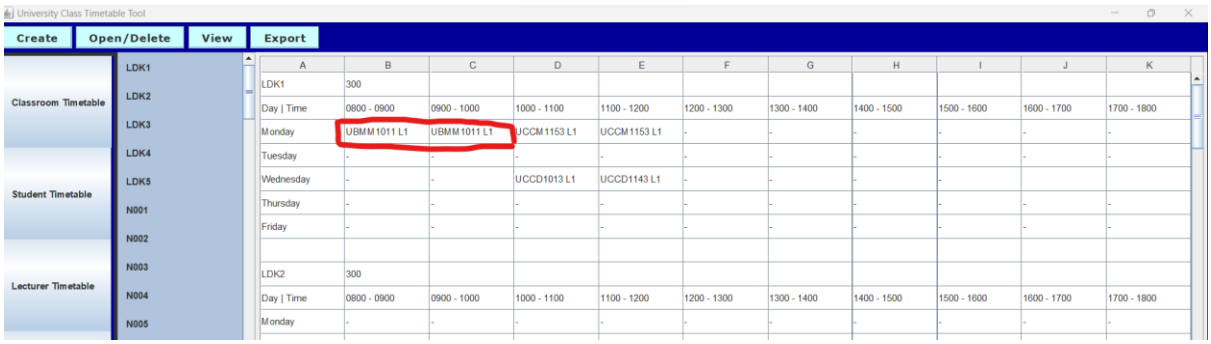


Figure 4.3.14 Data Changed in View Page after Timetable Modification

The first class event is now changed to Monday 0800-1000 class in LDK1.

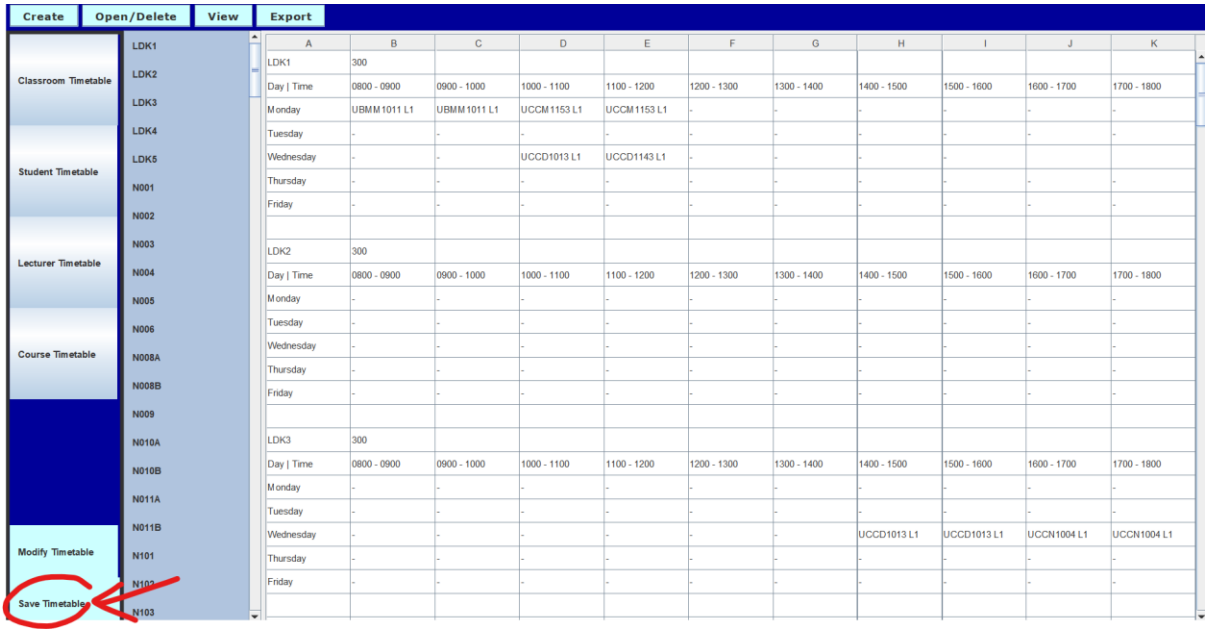


Figure 4.3.15 Clicking of Save Timetable Button

After that, the user can choose to save the timetable to the database by clicking on the save button at the bottom right of the interface. A message will be popped up asking the user to key in the timetable name he wants to save.

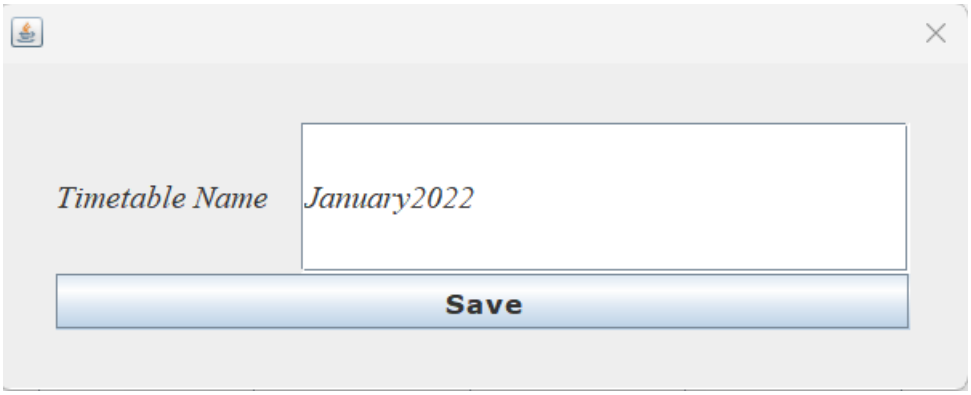


Figure 4.3.16 Key in of Timetable Name to be Saved before Saving

After clicking the save button, the timetable will be saved with the name keyed in by the user. Then, to open the existing timetable, the user can click on the Open/Delete button in the navigation bar.

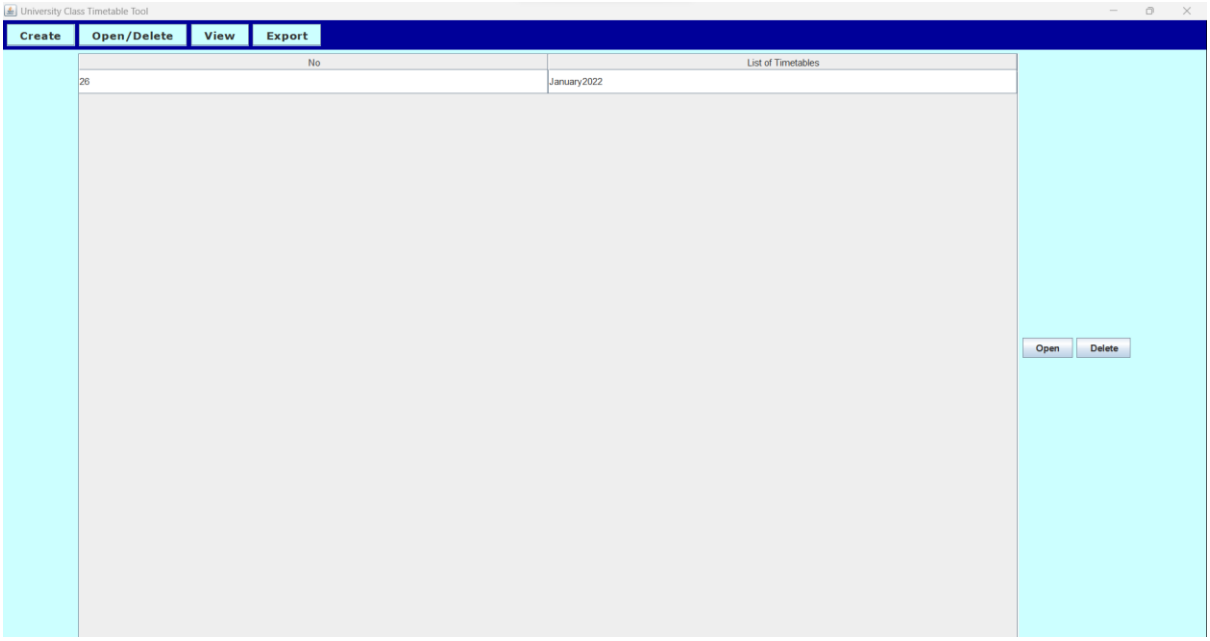


Figure 4.3.17 Retrieving of Saved Timetable in Open Page

This interface shows the existing timetable data stored in the database. The user can choose the timetable to open or delete. If the user chooses to open the file, he selects the data he wanted to open, then click on the open button at the right site of the interface. The user will then be directed to the interface where he can view the timetable as the same with the interface after he created the new timetable in the previous step.

Figure 4.3.18 View Page after Opening the Saved Timetable

The user will be directed to this interface again after he opens the timetable.

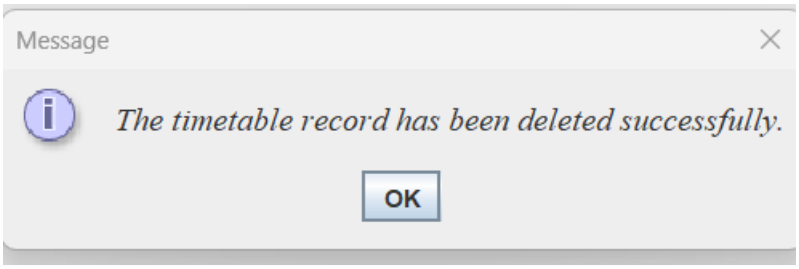


Figure 4.3.19 Message when Deleting the Saved Timetable Successfully

If the user wants to delete the timetable data, he clicks on the delete button in the open/delete interface. Then, the alert message will pop up.

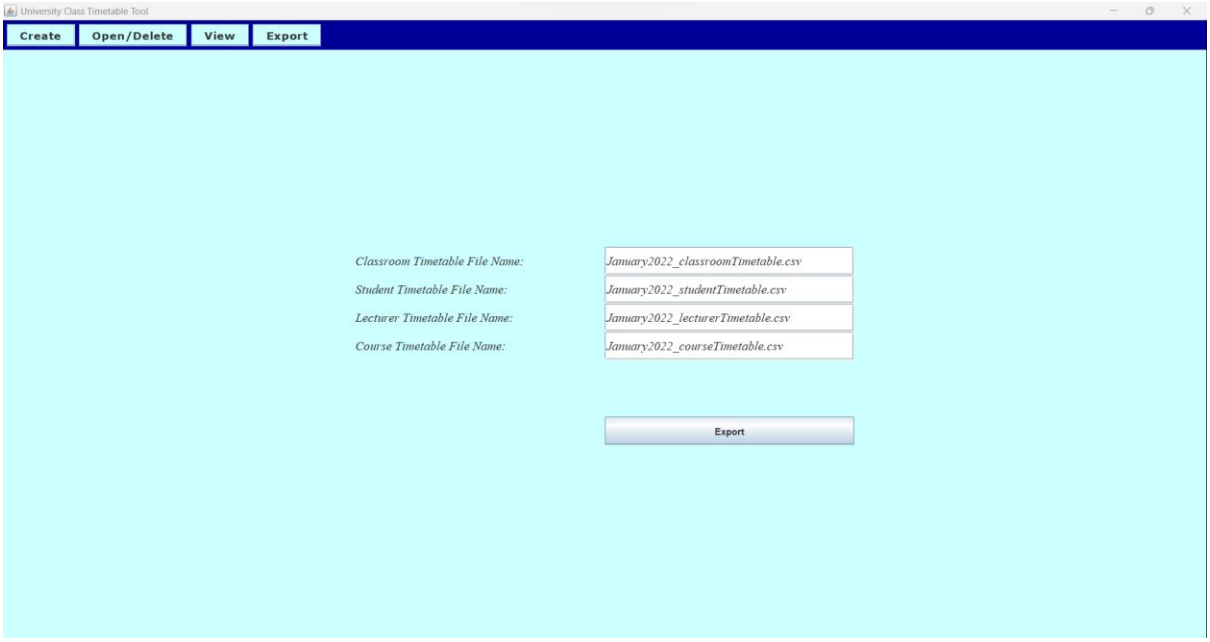


Figure 4.3.20 Export Pane of System

Finally, the user can export the timetable data as csv.

4.4 Preliminary Results Produced in CSV File Format

4.4.1 Classrooms' Timetable

N001	40				
Timeslots|Days	Monday	Tuesday	Wednesday	Thursday	Friday
1	-	-	[CSY1S3 UCCD1013 LESSON MA]	-	-
2	-	-	-	-	-
3	-	-	-	-	-
4	[CSY1S1 UCCM1153 LESSON LKH]	-	[CSY1S1 UCCD1143 LESSON KXY]	-	-
5	-	-	-	[CSY1S3 UCCM1363 LESSON SPC]	-
6	-	-	-	[CSY1S3 UCCM1363 LESSON LFW]	-
7	-	-	-	-	-
8	-	-	-	[CSY1S2 UCCD1013 LESSON MO]	-
9	-	-	-	-	-
10	-	-	-	[CSY1S3 UCCM1353 LESSON LSJ]	-
N002	50				
Timeslots|Days	Monday	Tuesday	Wednesday	Thursday	Friday
1	-	-	-	-	-
2	-	-	-	-	-
3	-	-	-	-	-
4	-	-	-	-	-
5	-	-	-	-	-
6	-	[CSY1S2 UCCD1143 LESSON LSP]	-	-	[CSY1S2 UCCD1143 LESSON LSP]
7	-	-	[CSY1S1 UCCD1013 LESSON MA]	-	-
8	-	[CSY1S2 UCCD1143 LESSON KXY]	-	-	-
9	[CSY1S3 UCCM1353 LESSON LSJ]	-	-	-	-
10	-	-	-	-	-
N003	50				
Timeslots|Days	Monday	Tuesday	Wednesday	Thursday	Friday
1	-	-	-	-	-
2	-	-	[CSY1S1 UCCM1153 LESSON LSJ]	-	-

Figure 4.4.1.1 Screen Capture of Classrooms' Timetable in CSV file

The figure 4.4.1.1 above shows the classrooms' timetables with the allocated timeslots. The timeslots designed so far are in the 5 days per week and 10 timeslots per day manner. The "-" in the cells means that the timeslot is empty. While the cell with content shows the information of student groups, course code, class event type and assigned lecturer.

4.4.2 Students' Timetable

CSY1S1	150				
Timeslots|Days	Monday	Tuesday	Wednesday	Thursday	Friday
1	UCCD1004 LECTURE TJS LDK4 |||	UCCD1004 LAB AAPE N010A ||| UCC	UCCN1004 LAB AY N009 |||	UCCD1013 LECTURE MO LDK5 |||	UCCM1153 LECTURE LSJ LDK1 |||
2	UCCN1004 LECTURE AY LDK5 |||	UCCD1013 LESSON MA N006 |||	UCCM1153 LESSON LSJ N003 |||	UCCD1004 LAB TSHS N010A |||	UCCN1004 LAB TSK N008B ||| UCCN1004 LAB TLY N103 |||
3	UCCN1004 LAB OCY N008A |||	UCCN1004 LAB TLY N011B ||| UCCN	UCCD1013 LESSON NWQ N004 ||| U	UCCD1143 LECTURE LSP LDK1 |||	UCCN1004 LAB TLY N010A |||
4	UCCM1153 LESSON LKH N001 |||	UCCN1004 LECTURE AY LDK1 |||	UCCD1143 LESSON KXY N001 ||| UC	UCCD1143 LECTURE LSP LDK2 |||	UCCM1153 LESSON LSJ N004 |||
5	UCCN1004 LAB TSK N102 ||| UCCN	UCCD1004 LAB TSHS N010A |||	UCCN1004 LECTURE AY LDK3 |||	UCCD1004 LECTURE TJS LDK3 |||	UCCN1004 LAB AY N008B |||
6	UBMM1011 LECTURE TCW LDK1 |||	UCCD1004 LAB KXY N101 ||| UCCD	UCCD1143 LESSON KXY N003 ||| UC	UBMM1011 LECTURE TCW LDK4 |||	UCCD1004 LAB KXY N011A |||
7	UCCD1013 LESSON MO N006 |||	UCCM1153 LESSON LKH N005 |||	UCCD1013 LESSON MA N002 |||	UCCM1153 LECTURE LSJ LDK1 |||	UCCD1004 LAB TSHS N011B |||
8	UCCD1004 LECTURE TJS LDK4 |||	UCCN1004 LAB AY N101 |||	UCCD1004 LAB TJS N011A |||	UCCD1004 LAB JKYM N103 |||	UCCN1004 LAB OCY N010A ||| UCCN1004 LAB TSK N010B ||
9	UCCD1143 LECTURE LSP LDK1 |||	UCCM1153 LESSON LKH N003 |||	UCCD1013 LECTURE MO LDK5 |||	UCCD1004 LAB AAPE N008A ||| UCC	UCCM1153 LECTURE LSJ LDK2 |||
10	UCCD1143 LESSON KXY N006 |||	UCCD1013 LECTURE MO LDK5 |||	UCCD1004 LAB JKYM N102 |||	UCCD1004 LAB CCH N102 |||	UCCN1004 LAB OCY N011B ||| UCCN1004 LAB TSK N009 |||
CSY1S2	150				
Timeslots|Days	Monday	Tuesday	Wednesday	Thursday	Friday
1	UCCM1153 LESSON LSJ N005 |||	UCCD1004 LAB TSHS N008A |||	UCCM1153 LECTURE LSJ LDK5 |||	UCCD1013 LESSON MA N005 |||	UCCD1143 LECTURE LSP LDK2 |||
2	UCCN1004 LAB TLY N010A ||| UCCN	UCCN1004 LAB TLY N008B ||| UCCN	UCCN1004 LAB AY N011A ||| UCCN	UCCN1004 LAB OCY N011B |||	UCCD1004 LAB TJS N011B |||
3	UCCD1004 LAB CCH N010B ||| UCC	UCCD1013 LECTURE MO LDK5 |||	UCCD1004 LAB JKYM N009 |||	UCCD1004 LAB KXY N008A ||| UCCD	UCCD1004 LAB AAPE N008B |||
4	UCCN1004 LAB TLY N102 |||	UCCM1153 LESSON LKH N003 |||	UCCD1004 LECTURE TJS LDK5 |||	UCCD1004 LAB JKYM N008A ||| UCC	UCCD1004 LAB CCH N008B ||| UCCD1004 LAB TSHS N103 ||
5	UCCM1153 LESSON LSJ N003 ||| UC	UCCD1143 LESSON KXY N004 |||	UCCN1004 LAB OCY N011B ||| UCC	UCCD1004 LAB JKYM N011A |||	UCCM1153 LESSON LKH N003 |||
6	UCCD1013 LECTURE MO LDK3 |||	UCCD1143 LESSON LSP N002 |||	UCCD1004 LAB KXY N010B |||	UCCN1004 LAB AY N101 ||| UCCN1	UCCD1143 LESSON LSP N002 |||
7	UCCN1004 LECTURE AY LDK3 |||	UCCD1013 LECTURE MO LDK3 |||	UCCD1004 LECTURE TJS LDK2 |||	UCCN1004 LAB TSK N101 |||	UCCD1143 LECTURE LSP LDK2 |||
8	UCCN1004 LAB TSK N101 |||	UCCD1143 LESSON KXY N002 |||	UCCN1004 LECTURE AY LDK5 |||	UCCD1013 LESSON MO N001 ||| UC	UCCD1143 LECTURE LSP LDK4 |||
9	UBMM1011 LECTURE TCW LDK2 |||	UCCM1153 LECTURE LSJ LDK4 |||	UCCN1004 LAB AY N010B |||	UCCD1013 LESSON NWQ N006 |||	UCCD1004 LECTURE TJS LDK3 |||
10	UBMM1011 LECTURE TCW LDK5 |||	UCCD1004 LAB TJS N011A ||| UCCD	UCCM1153 LECTURE LSJ LDK3 |||	UCCN1004 LECTURE AY LDK3 |||	UCCD1143 LESSON KXY N004 |||
CSY1S3	150				
Timeslots|Days	Monday	Tuesday	Wednesday	Thursday	Friday
1	UCCD1203 LAB SAPS N011B |||	UCCM1353 LECTURE LSJ LDK4 |||	UCCD1013 LESSON MA N001 ||| UC	UCCD1024 LECTURE GCM LDK3 |||	UCCD1024 LAB LSC N008B ||| UCCD1024 LAB GCM N103 |||
2	UCCD1024 LECTURE GCM LDK3 |||	UCCD1203 LECTURE NBAI LDK3 |||	UCCD1024 LAB JLIW N008B ||| UCC	UCCD1203 LAB NSBCL N011A ||| UC	UCCD1024 LAB GCM N101 ||| UCCD1024 LAB LSC N102 |||

Figure 4.4.2.1 Screen Capture of Students' Timetable in CSV file

The figure 4.4.2.1 above shows the students' timetables with the allocated timeslots. In each timeslot, there may have more than one class events. For the multiple class events in single timeslots, the students can choose any one of them to attend as

they are the same event with the same event group id. The symbol "|||" is to separate the multiple class events in each timeslot. Each class event contains the information of the course id, class event type, lecturer name, and the classroom name.

4.4.3 Lecturers' Timetable

TCW					
Timeslots|Days	Monday	Tuesday	Wednesday	Thursday	Friday
1					
2					
3					
4					
5					
6	UBMM1011 LECTURE CSY1S1 LDK1 |||			UBMM1011 LECTURE CSY1S1 LDK4 |||	
7					
8					
9	UBMM1011 LECTURE CSY1S2 LDK2 |||				
10	UBMM1011 LECTURE CSY1S2 LDK5 |||				
TJS					
Timeslots|Days	Monday	Tuesday	Wednesday	Thursday	Friday
1	UCCD1004 LECTURE CSY1S1 LDK4 ||	UCCD1004 LAB CSY1S1 N101 |||			
2					UCCD1004 LAB CSY1S2 N011B |||
3					
4			UCCD1004 LECTURE CSY1S2 LDK5 |||		
5				UCCD1004 LECTURE CSY1S1 LDK3 |||	
6					
7			UCCD1004 LECTURE CSY1S2 LDK2 |||		
8	UCCD1004 LECTURE CSY1S1 LDK4 |||		UCCD1004 LAB CSY1S1 N011A |||		
9					UCCD1004 LECTURE CSY1S2 LDK3 |||
10		UCCD1004 LAB CSY1S2 N011A |||			
JKYM					
Timeslots|Days	Monday	Tuesday	Wednesday	Thursday	Friday
1		UCCD1004 LAB CSY1S1 N103 |||			
2					

Figure 4.4.3.1 Screen Capture of Lecturers' Timetable in CSV file

The figure 4.4.3.1 above shows the lecturers' timetables with the allocated timeslots. In each timeslot, there can only have one class event. The symbol "|||" is to separate the multiple class events in each timeslot. However, since there cannot be having more than one class event in each timeslot of the lecturer, then the "|||" is to test if each timeslot contain only one class event. Each class event contains the information of the course id, class event type, student group name, and the classroom name.

4.4.4 Courses' Timetable

UBMM1011					
Timeslots|Days	Monday	Tuesday	Wednesday	Thursday	Friday
1					
2					
3					
4					
5					
6	TCW LECTURE CSY1S1 LDK1 |||			TCW LECTURE CSY1S1 LDK4 |||	
7					
8					
9	TCW LECTURE CSY1S2 LDK2 |||				
10	TCW LECTURE CSY1S2 LDK5 |||				
UCCD1004					
Timeslots|Days	Monday	Tuesday	Wednesday	Thursday	Friday
1	TJS LECTURE CSY1S1 LDK4 |||	TSHS LAB CSY1S2 N008A ||| AAPE LAB CSY1S1 N010A ||| TJS LAB CSY1S1 N101 ||| JKYM LAB CSY1S1 N103 |||			
2				TSHS LAB CSY1S1 N010A |||	TJS LAB CSY1S2 N011B |||
3	CCH LAB CSY1S2 N010B ||| AAPE LAB CSY1S2 N103 |||		JKYM LAB CSY1S2 N009 |||	KXY LAB CSY1S2 N008A ||| JKYM LAI	AAPE LAB CSY1S2 N008B |||
4			TJS LECTURE CSY1S2 LDK5 |||	JKYM LAB CSY1S2 N008A ||| TSHS L/	CCH LAB CSY1S2 N008B ||| TSHS LAB CSY1S2 N103 |||
5		TSHS LAB CSY1S1 N010A |||		JKYM LAB CSY1S2 N011A ||| TJS LECTURE CSY1S1 LDK3 |||	
6		KXY LAB CSY1S1 N101 ||| TSHS LAB (	KXY LAB CSY1S2 N010B |||		KXY LAB CSY1S1 N011A |||
7			TJS LECTURE CSY1S2 LDK2 |||		TSHS LAB CSY1S1 N011B |||
8	TJS LECTURE CSY1S1 LDK4 |||		TJS LAB CSY1S1 N011A |||	JKYM LAB CSY1S1 N103 |||	
9				AAPE LAB CSY1S1 N008A ||| JKYM L	TJS LECTURE CSY1S2 LDK3 |||
10		TJS LAB CSY1S2 N011A ||| TSHS LAB	JKYM LAB CSY1S1 N102 |||	CCH LAB CSY1S1 N102 |||	
UCCD1013					
Timeslots|Days	Monday	Tuesday	Wednesday	Thursday	Friday
1			MA LESSON CSY1S3 N001 ||| TKL LE	MA LESSON CSY1S2 N005 ||| MO LECTURE CSY1S1 LDK5 |||	

Figure 4.4.4.1 Screen Capture of Courses' Timetable in CSV file

The figure 4.4.4.1 above shows the courses' timetables with the allocated timeslots. In each timeslot, there can have more than one class event. The symbol "|||" is to separate the multiple class events in each timeslot. Each class event contains the information of the lecturer name, class event type, student group name, and the classroom name.

4.5 Data Storage Design

In this system, the data will be stored in Relational Database Management System (RDBMS) which is MySQL database. The purpose of using MySQL database is because it is an open-source RDBMS where it can be connected to the java application with Java Database Connectivity (JDBC) through the java code. The data required in this system are the information required by the timetable such as the classrooms' information, courses' information, lecturers' information, and students' information. In current stage, Xampp will be used as the middleman to host the application and the database. The data storge design in the MySQL database will be showed using the class diagram below:

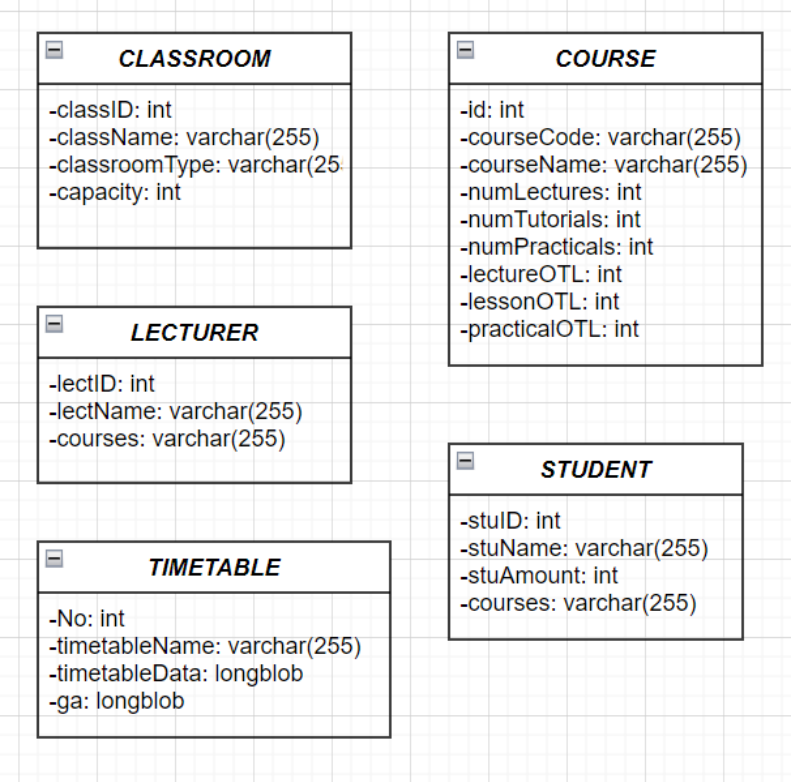


Figure 4.5.1 The Design of the Database Structure of the Timetabling Tool

For the use of the timetabling tool, the data inside the database will be read and based on the data read, the timetabling tool will then generate the timetable.

Classroom:

Columns	**Data Types**	**Descriptions**
classID	Int	ID to identify the classroom
className	Varvhar(255)	The name of the classroom
classroomType	Varchar(255)	The type of the classroom
capacity	int	The capacity of the classroom

Table 4.5.1 Data Structure of Entity Classroom

Course:

Columns	**Data Types**	**Descriptions**

id	Int	ID to identify the course
courseCode	Varvhar(255)	The code of the course
courseName	Varchar(255)	The description of the course
numLectures	int	The number of the lectures of the course
numTutorials	int	The number of the tutorials of the course
numPracticals	int	The number of practicals of the course
lectureOTL	int	The number indicates the percentage of the lecture class to be held in OTL mode
lessonOTL	int	The number indicates the percentage of the tutorial class to be held in OTL mode
practicalOTL	int	The number indicates the percentage of the practical class to be held in OTL mode

Table 4.5.2 Data Structure of Entity Course

Lecturer:

Columns	**Data Types**	**Descriptions**
lectID	Int	ID to identify the lecturer
lectName	Varvhar(255)	The name of the lecturer
courses	Varchar(255)	The lists of courses taught by lecturer

Table 4.5.3 Data Structure of Entity Lecturer

Student:

Columns	**Data Types**	**Descriptions**
stuID	Int	ID to identify the student group

stuName	Varvhar(255)	The name of the student group
stuAmount	int	The number of students in the student group
courses	Varchar(255)	The lists of courses attended by the student group

Table 4.5.4 Data Structure of Entity Student

Timetable:

Columns	**Data Types**	**Descriptions**
No	Int	No to identify the timetable
timetableName	Varvhar(255)	The name of the timetable
timetableData	Longblob	The timetable data which represents the java class
ga	longblob	The ga data that represents the genetic algorithm setup of the timetable generation

Table 4.5.5 Data Structure of Entity Timetable

4.6 Feasibility of Proposed Method

During the implementation of the system and the test case, it is found that for the big size of data, the Genetic Algorithm will be taking a very long time to find the optimal and feasible solution. If the hardware capability is not high enough, time consuming will be one of the implementation problems.

Besides, the mutation rate of the algorithm needs to be handled well as if it is too high, the fitness improvement in each generation will be very slow. On the other hand, if the mutation rate is low, the generations produced might be stuck at the local optimal instead of going to reach the fitness value of zero. Hence, one can only know if the certation mutation rate is suitable after going through the timetable generation processes.

Chapter 5

System Testing

In this section, the system testing will be done to test the system performance and correctness. In chapter 3, there are a set of test s defined. After finishing the implementation of the system, the test cases were gone through, and the result sets were obtained. Section 5.1 will define the system testing setup; section 5.2 to 5.5 will be the test plans to be conducted.

5.1 System Testing Setup and Procedure

5.1.1 System Testing Setup

The system testing will be done in one shot for the test plans in section 5.2, 5.3, and 5.4. The input to the test case in section 5.2, 5.3, and 5.4 are a set of data which consist of the classroom data, course data, lecturer data, and student group data. The input data was stored in the MySQL database. The terms below are the particulars that will be used in the test plans.

Input Data: Figure 5.1.1, 5.1.2, 5.1.3 and 5.1.4 In the test plans in section 5.2, 5.3, and 5.4, the inputs of the data are all fixed and same. This input data set will be called as "**Dataset1**" throughout the section 5.1, 5.2 and 5.3.

Description: To describe the purpose of current testing.

Expected result: The expected results define the preferable output that should be produced by the timetable generator in generating the timetable.

Actual result: All the actual result will be observed from the csv files produced to analyse whether the timetable produced is violating any constraints defined

Status: to define whether the test plan is pass or failed

Explanation: To elaborate the test plans conducted and the results.

The figures below show the data stored in the database:

	Name	Capacity	Type
Edit Copy Delete	LDK1	300	L
Edit Copy Delete	LDK2	300	L
Edit Copy Delete	LDK3	300	L
Edit Copy Delete	LDK4	300	L
Edit Copy Delete	LDK5	300	L
Edit Copy Delete	N001	40	T
Edit Copy Delete	N002	40	T
Edit Copy Delete	N003	40	T
Edit Copy Delete	N004	40	T
Edit Copy Delete	N005	40	T
Edit Copy Delete	N006	40	T
Edit Copy Delete	N008A	25	P
Edit Copy Delete	N008B	25	P
Edit Copy Delete	N009	25	T
Edit Copy Delete	N010A	25	P
Edit Copy Delete	N010B	25	P
Edit Copy Delete	N011A	25	P
Edit Copy Delete	N011B	25	P
Edit Copy Delete	N101	25	P
Edit Copy Delete	N102	25	P
Edit Copy Delete	N103	25	P
Edit Copy Delete	N104	25	P

Figure 5.1.1.1 The Test Plan Classroom Data

	lecName	courses	teachLecture
Edit Copy Delete	Cik Nor Fatiha Binti Subri	UCCD2303	0
Edit Copy Delete	Cik Norazira binti A Jalil	UCCD1203	1
Edit Copy Delete	Dr Ashvaany a/p Egambaram	UCCD1004	0
Edit Copy Delete	Dr Aun Yichiet	UCCN1004	1
Edit Copy Delete	Dr Chai Meei Tyng	UCCD2044	1
Edit Copy Delete	Dr Goh Chuan Meng	UCCD1024	1
Edit Copy Delete	Dr Jasmina Khaw Yen Min	UCCD1004	0
Edit Copy Delete	Dr Kh'ng Xin Yi	UCCD1004 UCCD1143	0
Edit Copy Delete	Dr Lem Kong Hoong	UCCM1153	1
Edit Copy Delete	Dr Manoranjitham a/p Muniandy	UCCD1013	0
Edit Copy Delete	Dr Mogana a/p Vadiveloo	UCCD1013	1
Edit Copy Delete	Dr Ng Hui Fuang	UCCD2044	0
Edit Copy Delete	Dr Nur Syadhila binti Che Lah	UCCD1203	0
Edit Copy Delete	Dr Ramesh Kumar Ayyasamy	UCCD2303	0
Edit Copy Delete	Dr Robithoh Annur	UCCN2243	0
Edit Copy Delete	Dr Sayed Ahmad Zikri Bin Sayed Aluwee	UCCD2044	0
Edit Copy Delete	Dr Suthashini a/p Subramaniam	UCCD1203 UCCD2303	1
Edit Copy Delete	Dr Tan Joi San	UCCD1004	1
Edit Copy Delete	Dr Tee Chee Wee	UBMM1011	1
Edit Copy Delete	Dr Tse Siu Hong Savio	UCCD1004	0
Edit Copy Delete	Mr Ch'ng Chee Henn	UCCD1004	0

Figure 5.1.1.2 The Test Plan Course Data

	courseCode	numLectures	numTutorials	numPracticals	lectureOTL	lessonOTL	practicalOTL
Edit Copy Delete	MPU3113	3	0	0	100	0	0
Edit Copy Delete	MPU34072	2	0	0	100	0	0
Edit Copy Delete	UBMM1011	2	0	0	100	0	0
Edit Copy Delete	UCCD1004	3	0	2	100	0	100
Edit Copy Delete	UCCD1013	3	1	0	0	50	0
Edit Copy Delete	UCCD1024	3	0	2	0	0	100
Edit Copy Delete	UCCD1143	3	1	0	0	0	0
Edit Copy Delete	UCCD1203	2	0	2	0	0	0
Edit Copy Delete	UCCD2003	3	1	0	100	100	0
Edit Copy Delete	UCCD2044	1	1	0	100	100	0
Edit Copy Delete	UCCD2103	3	1	0	100	100	0
Edit Copy Delete	UCCD2303	2	0	2	100	0	100
Edit Copy Delete	UCCM1153	3	1	0	100	100	0
Edit Copy Delete	UCCM1353	3	1	0	100	0	0
Edit Copy Delete	UCCM1363	3	1	0	0	0	0
Edit Copy Delete	UCCN1004	3	0	2	0	0	100
Edit Copy Delete	UCCN2243	2	0	2	100	0	100

Figure 5.1.1.3 The Test Plan Lecturer Data

	stuName	stuAmount	courses
Edit Copy Delete	2022JAN	200	UBMM1011 UCCD1004 UCCD1013 UCCD1143 UCCM1153 UCCN1...
Edit Copy Delete	2022MAY	220	MPU3113 UCCD1024 UCCD1203 UCCM1353 UCCM1363
Edit Copy Delete	2022OCT	220	UCCD2003 UCCD2044 UCCD2103 UCCD2303 UCCN2243

Figure 5.1.1.4 The Test Plan Student Group Data

5.1.2 System Testing Procedure

After all the data stored into the database, the data will be retrieved from the database to the system when running the system. After selecting the constraints in the system, the timetable can then be generated. The evaluation of the test plans will be done by observing the timetable produced.

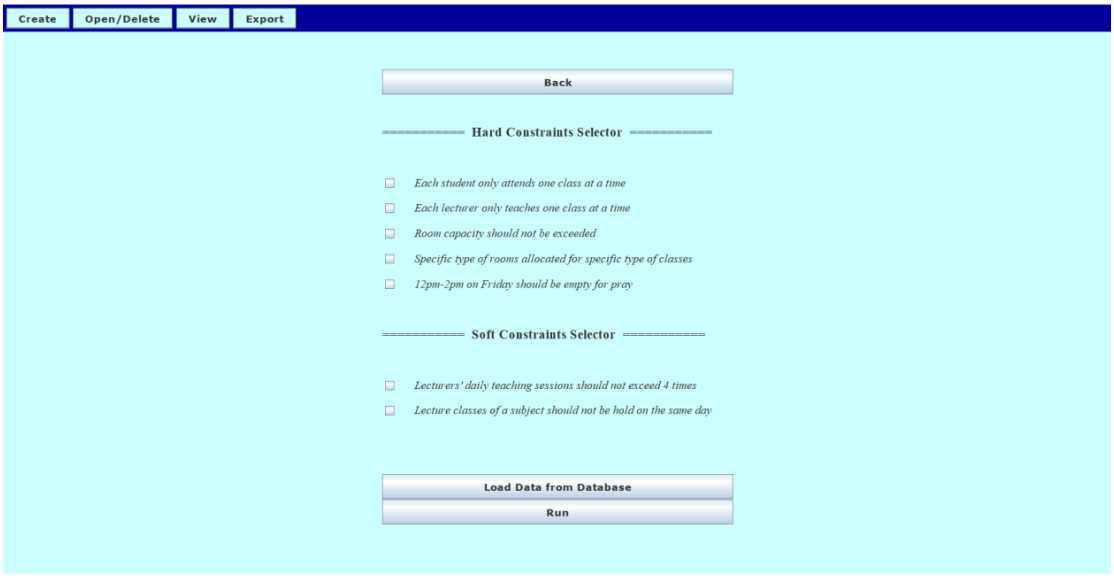


Figure 5.1.2.1 The Constraints Selection Interface of the System

First of all, tick all the constraints in the constraints selection interface. Then, click the load data button. After that, click the run button. After the timetable generated, export the timetables csv files in the export panel.

5.2 System Testing Results for the Hard Constraints Defined

In this section, after generating the timetable, the timetable was tested to ensure that it follows the constraints. The test plans' results will be conducted by observing the csv timetable files produced based on different scenarios.

5.2.1 Each student can only take one class at a time

This test case is to ensure that in each timeslot, one student can only take one class event.

Input data	Description	Expected result	Actual result	Status
Dataset1	This test case is to ensure that in each timeslot, one student can only take one class event.	From the observation on the student timetable csv file produced, the student should	The student groups have one class event to attend in a single timeslot.	Pass

		only have one class event to attend in a single timeslot.		

Table 5.2.1.1 Test Case of "Each student can only take one class at a time"

Explanation:

Cell	Value	Formula
B3	UCCN1004 LAB Practical 25 Mr Teoh Shen Khang N011B |||	
C3	UCCN1004 LAB Practical 25 Mr Teoh Shen Khang N011B |||	
D3	UCCN1004 LECTURE Lecture 80 Dr Aun Yichiet LDK5 |||	
F3	UCCM1153 LESSON Tutorial 35 Dr Lem Kong Hoong N002 |||	
H3	UCCD1143 LESSON Tutorial 35 Ts Dr Lim Seng Poh N003 |||	
J3	UCCD1143 LECTURE Lecture 80 Ts Dr Lim Seng Poh LDK1 |||	
B4	UCCM1153 LESSON Tutorial 5 Ms Lim Shun Jinn N005 |||	
D4	UCCD1143 LECTURE Lecture 80 Ts Dr Lim Seng Poh LDK5 |||	
E4	UCCD1143 LECTURE Lecture 80 Ts Dr Lim Seng Poh LDK5 |||	
F4	UCCN1004 LAB Practical 25 Ms Tan Lyk Yin N010B ||| UCCN1004 LAB Practical 25 Ts Dr Ooi Chek Yee N104 |||	
G4	UCCN1004 LAB Practical 25 Ms Tan Lyk Yin N010B ||| UCCN1004 LAB Practical 25 Ts Dr Ooi Chek Yee N104 |||	
H4	UCCD1013 LESSON Tutorial 5 Ms Tseu Kwan Lee N006 |||	
J4	UCCD1013 LECTURE Lecture 80 Dr Mogana a/p Vadiveloo LDK3 |||	
K4	UCCD1013 LECTURE Lecture 80 Dr Mogana a/p Vadiveloo LDK3 |||	
B5	UCCD1143 LESSON Tutorial 35 Dr Kh'ng Xin Yi N003 |||	
D5	UBMM1011 OTL Lecture 80 Dr Tee Chee Wee OTL14 |||	
E5	UBMM1011 OTL Lecture 80 Dr Tee Chee Wee OTL14 |||	
F5	UCCM1153 OTL Lecture 80 Dr Lem Kong Hoong OTL7 |||	
H5	UCCM1153 OTL Tutorial 40 Ms Lim Shun Jinn OTL16 |||	
J5	UCCD1004 OTL Practical 10 Dr Tan Joi San OTL3 ||| UCCD1004 OTL Practical 35 Dr Kh'ng Xin Yi OTL14 |||	
K5	UCCD1004 OTL Practical 10 Dr Tan Joi San OTL3 ||| UCCD1004 OTL Practical 35 Dr Kh'ng Xin Yi OTL14 |||	
D6	UCCD1004 OTL Lecture 80 Dr Tan Joi San OTL2 |||	
E6	UCCD1004 OTL Lecture 80 Dr Tan Joi San OTL2 |||	
F6	UCCM1153 OTL Lecture 80 Dr Lem Kong Hoong OTL1 |||	
G6	UCCM1153 OTL Lecture 80 Dr Lem Kong Hoong OTL1 |||	
H6	UCCD1013 LESSON Tutorial 35 Ms Ng Wan Qing N002 ||| UCCD1013 OTL Tutorial 40 Dr Mogana a/p Vadiveloo OTL4 |||	
J6	UCCD1004 OTL Lecture 80 Dr Tan Joi San OTL5 |||	
B7	UCCD1004 OTL Practical 35 Dr Jasmina Khaw Yen Min OTL1 |||	

Figure 5.2.1.1 Proof of "Each student can only take one class at a time"

From the figure 5.2.1.1 above, each cell represents the class events of student that the students may join. The "|||" symbol is to separate different class events if there is more than one event in a single timeslot. For tutorial and practical classes, since there can have many different student groups to fit all students inside, there can have multiple same tutorial/practical class events at the same time, and each student choose one from them to attend. However, when observing the lecture event, one and only lecture class event can be existed at one timeslot. Means the student can only attend one class at one time. The test case success.

5.2.2 Each lecturer can only take one class at a time

This test case is to ensure that in each timeslot, each lecturer can only take one class at a time.

Input data	Description	Expected result	Actual result	Status
Dataset1	Each lecturer can only take one class at a time	From the observation on the lecturer timetable csv file produced, the lecturer should only have one class event in a single timeslot.	The lecturers have only one class event to attend in a single timeslot.	Pass

Table 5.2.2.1 Test Case of "Each lecturer can only take one class at a time"

Explanation:

Cell	Value	Formula
F12	UCCD1203 LECTURE Lecture 120 2022MAY LDK1 |||	
G12	UCCD1203 LECTURE Lecture 120 2022MAY LDK1 |||	
D24	UCCN1004 LECTURE Lecture 80 2022JAN LDK5 |||	
F28	UCCN1004 LAB Practical 5 2022JAN N101 |||	
G28	UCCN1004 LAB Practical 5 2022JAN N101 |||	
H28	UCCN1004 LECTURE Lecture 80 2022JAN LDK5 |||	
I28	UCCN1004 LECTURE Lecture 80 2022JAN LDK5 |||	
H32	UCCD2044 OTL Lecture 100 2022OCT OTL4 |||	
I32	UCCD2044 OTL Lecture 100 2022OCT OTL4 |||	
D39	UCCD1024 LECTURE Lecture 120 2022MAY LDK4 |||	
E39	UCCD1024 LECTURE Lecture 120 2022MAY LDK4 |||	
B42	UCCD1024 LECTURE Lecture 120 2022MAY LDK3 |||	
H42	UCCD1024 LAB Practical 25 2022MAY N101 |||	
I42	UCCD1024 LAB Practical 25 2022MAY N101 |||	
B49	UCCD1004 OTL Practical 35 2022JAN OTL1 |||	
C49	UCCD1004 OTL Practical 35 2022JAN OTL1 |||	
B54	UCCD1143 LESSON Tutorial 35 2022JAN N003 |||	
J54	UCCD1004 OTL Practical 35 2022JAN OTL14 |||	

Figure 5.2.2. 1 Proof of "Each lecturer can only take one class at a time"

From the figure 5.2.2.1 above, each cell represents the class events of lecturers that the lecturers need to teach. The "|||" symbol is to separate different class events if there is more than one event in a single timeslot. For all the cells observed, there is no

cell that has more than one class event, means that each lecturer only teaches one class event at a single time. The test case success.

5.2.3 Each classroom can only have one class assigned at a time

This test case is to ensure that for all classrooms, only one class event can be held in in one timeslot.

Input data	**Description**	**Expected result**	**Actual result**	**Status**
Dataset1	Each classroom can only have one class assigned at a time	From the classrooms' timetable produced, it is made sure that in each timeslot, there are no more than two classes.	There are no more than two classes in single timeslot of the classrooms' timetables	Pass

Table 5.2.3.1 Test Case of "Each classroom can only have one class assigned at a time"

Explanation:

Cell	Value	Formula
J3	[2022JAN L1 80 UCCD1143 LECTURE Lecture Ts Dr Lim Seng Poh]	
F5	[2022MAY L1 120 UCCD1203 LECTURE Lecture Cik Norazira binti A Jalil]	
G5	[2022MAY L1 120 UCCD1203 LECTURE Lecture Cik Norazira binti A Jalil]	
D7	[2022MAY L1 120 UCCM1363 LECTURE Lecture Mr Lim Foo Weng]	
E7	[2022MAY L1 120 UCCM1363 LECTURE Lecture Mr Lim Foo Weng]	
J18	[2022JAN L1 80 UCCD1013 LECTURE Lecture Dr Mogana a/p Vadiveloo]	
K18	[2022JAN L1 80 UCCD1013 LECTURE Lecture Dr Mogana a/p Vadiveloo]	
B21	[2022MAY L1 120 UCCD1024 LECTURE Lecture Dr Goh Chuan Meng]	
D25	[2022MAY L1 120 UCCD1024 LECTURE Lecture Dr Goh Chuan Meng]	
E25	[2022MAY L1 120 UCCD1024 LECTURE Lecture Dr Goh Chuan Meng]	
D31	[2022JAN L1 80 UCCN1004 LECTURE Lecture Dr Aun Yichiet]	
D32	[2022JAN L1 80 UCCD1143 LECTURE Lecture Ts Dr Lim Seng Poh]	
E32	[2022JAN L1 80 UCCD1143 LECTURE Lecture Ts Dr Lim Seng Poh]	
J33	[2022MAY L1 120 UCCM1363 LECTURE Lecture Mr Lim Foo Weng]	
D35	[2022JAN L1 80 UCCD1013 LECTURE Lecture Dr Mogana a/p Vadiveloo]	
H35	[2022JAN L1 80 UCCN1004 LECTURE Lecture Dr Aun Yichiet]	
I35	[2022JAN L1 80 UCCN1004 LECTURE Lecture Dr Aun Yichiet]	
F38	[2022MAY T4 15 UCCM1363 LESSON Tutorial Mr Lim Foo Weng]	
D40	[2022MAY T1 35 UCCM1363 LESSON Tutorial Ms Song Poh Choo]	
F45	[2022JAN T2 35 UCCM1153 LESSON Tutorial Dr Lem Kong Hoong]	

Figure 5.2.3. 1 Proof of "Each classroom can only have one class assigned at a time"

From the figure 5.2.3.1 above, each cell represents the class events of the classrooms in one single timeslot. The [] bracket is used to enclose each class event. For all the cells observed, there is no cell that has more than one class event, means that each classroom is used for only one class event at a single time. The test case success.

5.2.4 Each class should be taught by one lecturer at one time

This test case is to ensure that for all class event, only one lecture is in charged for that class event.

Input data	**Description**	**Expected result**	**Actual result**	**Status**
Dataset1	Each class should be taught by one lecturer at one time	From the classrooms' timetable produced, it is made sure that the classes in each timeslot	Only one lecturer's information in each timeslot from the classrooms'	Pass

		contain only one lecturer's information.	timetable produced	

Table 5.2.4.1 Test Case of "Each class should be taught by one lecturer at one time"

Explanation:

Cell	Value	Formula
J3	[2022JAN L1 80 UCCD1143 LECTURE Lecture Ts Dr Lim Seng Poh]	
F5	[2022MAY L1 120 UCCD1203 LECTURE Lecture Cik Norazira binti A Jalil]	
G5	[2022MAY L1 120 UCCD1203 LECTURE Lecture Cik Norazira binti A Jalil]	
D7	[2022MAY L1 120 UCCM1363 LECTURE Lecture Mr Lim Foo Weng]	
E7	[2022MAY L1 120 UCCM1363 LECTURE Lecture Mr Lim Foo Weng]	
J18	[2022JAN L1 80 UCCD1013 LECTURE Lecture Dr Mogana a/p Vadiveloo]	
K18	[2022JAN L1 80 UCCD1013 LECTURE Lecture Dr Mogana a/p Vadiveloo]	
B21	[2022MAY L1 120 UCCD1024 LECTURE Lecture Dr Goh Chuan Meng]	
D25	[2022MAY L1 120 UCCD1024 LECTURE Lecture Dr Goh Chuan Meng]	
E25	[2022MAY L1 120 UCCD1024 LECTURE Lecture Dr Goh Chuan Meng]	
D31	[2022JAN L1 80 UCCN1004 LECTURE Lecture Dr Aun Yichiet]	
D32	[2022JAN L1 80 UCCD1143 LECTURE Lecture Ts Dr Lim Seng Poh]	
E32	[2022JAN L1 80 UCCD1143 LECTURE Lecture Ts Dr Lim Seng Poh]	
J33	[2022MAY L1 120 UCCM1363 LECTURE Lecture Mr Lim Foo Weng]	
D35	[2022JAN L1 80 UCCD1013 LECTURE Lecture Dr Mogana a/p Vadiveloo]	
H35	[2022JAN L1 80 UCCN1004 LECTURE Lecture Dr Aun Yichiet]	
I35	[2022JAN L1 80 UCCN1004 LECTURE Lecture Dr Aun Yichiet]	
F38	[2022MAY T4 15 UCCM1363 LESSON Tutorial Mr Lim Foo Weng]	
D40	[2022MAY T1 35 UCCM1363 LESSON Tutorial Ms Song Poh Choo]	
F45	[2022JAN T2 35 UCCM1153 LESSON Tutorial Dr Lem Kong Hoong]	

Figure 5.2.4. 1 Proof of "Each class should be taught by one lecturer at one time"

By default, due to the structure of data input for timetable generation, one event is only allocated with one and only one lecturer. Since it was proved that the constraint 3 "Each classroom can only have one class assigned at a time", we can now observe each class event in the classrooms' timetables, to check if one class event has only one lecturer. From the figure above, we can see that all class events have only one lecturer assigned. The test case success.

5.2.5 The capacity of the classroom should be large enough to hold the students' size in each class

This test case is to ensure that the students' amount assigned to each class event in particular classroom does not exceed the capacity of the classroom.

Input data	Description	Expected result	Actual result	Status
Dataset1	The capacity of the classroom should be large enough to hold the students' size in each class	From the courses' timetable produced, it is made sure that for classroom with capacity less than the class size cannot have class filled in. (e.g., LESSON has class size of 30, classroom with less than 30 of room capacity cannot have any LESSON class)	The classrooms do not hold the class events that have students size exceeding its capacity	Pass

Table 5.2.5.1 Test Case of "The capacity of the classroom should be large enough to hold the students' size in each class"

Explanation:

For the lecture class, the room capacity is assumed to be large enough to fit all students who take the course. Whereas the tutorial and practical classes can only handle 35 and 25 students respectively. Same as lecture class, the OTL class which required online platform to conduct the class events have no limit on the students' amount to attend the class. Although it is set that only 50 and 35 maximum number of students can attend one OTL tutorial and OTL practical classes respectively to ensure the smooth teaching process.

K53	Ts Dr Chan Lee Kwun LAB Practical 20 2022MAY N008A |||
F54	Cik Norazira binti A Jalil LECTURE Lecture 120 2022MAY LDK1 |||
G54	Cik Norazira binti A Jalil LECTURE Lecture 120 2022MAY LDK1 |||
F56	Ms Yap Seok Gee LAB Practical 25 2022MAY N010A ||| Pn Siti Hajar Binti Abrain LAB Practical 25 2022MAY N102 |||
G56	Ms Yap Seok Gee LAB Practical 25 2022MAY N010A ||| Pn Siti Hajar Binti Abrain LAB Practical 25 2022MAY N102 |||
J56	Dr Nur Syadhila binti Che Lah LAB Practical 25 2022MAY N011B |||
K56	Dr Nur Syadhila binti Che Lah LAB Practical 25 2022MAY N011B |||
D59	Ts Dr Ku Chin Soon OTL Tutorial 50 2022OCT OTL6 |||
J61	Ts Dr Ku Chin Soon OTL Lecture 100 2022OCT OTL13 |||
D63	Mr Tou Jing Yi OTL Tutorial 50 2022OCT OTL8 |||
H63	Ts Dr Ku Chin Soon OTL Lecture 100 2022OCT OTL8 |||
I63	Ts Dr Ku Chin Soon OTL Lecture 100 2022OCT OTL8 |||

Cell	Value
D3	Ms Yuvashini a/p Salvamani OTL Lecture 120 2022MAY OTL14 |||
E3	Ms Yuvashini a/p Salvamani OTL Lecture 120 2022MAY OTL14 |||
B6	Ms Yuvashini a/p Salvamani OTL Lecture 120 2022MAY OTL3 |||
D19	Dr Tee Chee Wee OTL Lecture 80 2022JAN OTL14 |||
E19	Dr Tee Chee Wee OTL Lecture 80 2022JAN OTL14 |||
J26	Dr Tan Joi San OTL Practical 10 2022JAN OTL3 ||| Dr Kh'ng Xin Yi OTL Practical 35 2022JAN OTL14 |||
K26	Dr Tan Joi San OTL Practical 10 2022JAN OTL3 ||| Dr Kh'ng Xin Yi OTL Practical 35 2022JAN OTL14 |||
D27	Dr Tan Joi San OTL Lecture 80 2022JAN OTL2 |||
E27	Dr Tan Joi San OTL Lecture 80 2022JAN OTL2 |||
J27	Dr Tan Joi San OTL Lecture 80 2022JAN OTL5 |||
B28	Dr Jasmina Khaw Yen Min OTL Practical 35 2022JAN OTL1 |||
C28	Dr Jasmina Khaw Yen Min OTL Practical 35 2022JAN OTL1 |||
H32	Ms Tseu Kwan Lee LESSON Tutorial 5 2022JAN N006 |||
J32	Dr Mogana a/p Vadiveloo LECTURE Lecture 80 2022JAN LDK3 |||
K32	Dr Mogana a/p Vadiveloo LECTURE Lecture 80 2022JAN LDK3 |||
H34	Ms Ng Wan Qing LESSON Tutorial 35 2022JAN N002 ||| Dr Mogana a/p Vadiveloo OTL Tutorial 40 2022JAN OTL4 |||
D35	Dr Mogana a/p Vadiveloo LECTURE Lecture 80 2022JAN LDK5 |||
H38	Mr Jason Lim Jing Wei LAB Practical 25 2022MAY N011A |||
I38	Mr Jason Lim Jing Wei LAB Practical 25 2022MAY N011A |||
J38	Ts Lai Siew Cheng LAB Practical 20 2022MAY N008B |||
K38	Ts Lai Siew Cheng LAB Practical 20 2022MAY N008B |||
D39	Dr Goh Chuan Meng LECTURE Lecture 120 2022MAY LDK4 |||
E39	Dr Goh Chuan Meng LECTURE Lecture 120 2022MAY LDK4 |||
F39	Ts Lai Siew Cheng LAB Practical 25 2022MAY N102 |||
G39	Ts Lai Siew Cheng LAB Practical 25 2022MAY N102 |||

1Figure 5.2.5. 1 Proof of "The capacity of the classroom should be large enough to hold the students' size in each class"

The figure 5.2.5.1 above shows the courses' timetable which in the file each cell represents one timeslot. From the figure observed, the fourth data in each cell is the number of students in each timeslot of a class event. It is observed that the maximum number of students in the Lecture class is very high (120) which is full students based on the dataset. For the tutorial classes in physical mode, the maximum student is 35. For the practical classes in physical mode, the maximum student is 25. For the tutorial classes in OTL mode, the maximum student is 50. Lastly, for the practical classes in

physical mode, the maximum student is 35. From the observation above, the test case success.

5.2.6 The main subjects' classes should not be conducted at the same time

This test case is to ensure that for all students group, the subjects that they are required to take should not be held at the same time.

Input data	Description	Expected result	Actual result	Status
Dataset1	The main subjects' classes should not be conducted at the same time	From the students' timetable produced, it is made sure that in each timeslot, there are no more than two LECTURE classes. If there found more than two TUTORIAL/ PRACTICAL class events in one timeslot, if the class events are the same where students can choose one of them to attend,	There are no more than two LECTURE classes in a single timeslot in the students' timetables. For the TUTORIAL/ PRACTICAL classes, there can be found that more than two classes in one timeslot, but the students can choose anyone to attend.	Pass

		it is considered no violation.		

Table 5.2.6.1 Test Case of "The main subjects' classes should not be conducted at the same time"

Explanation:

Cell	Value
F4	UCCN1004 LAB Practical 25 Ms Tan Lyk Yin N010B ||| UCCN1004 LAB Practical 25 Ts Dr Ooi Chek Yee N104 |||
G4	UCCN1004 LAB Practical 25 Ms Tan Lyk Yin N010B ||| UCCN1004 LAB Practical 25 Ts Dr Ooi Chek Yee N104 |||
J5	UCCD1004 OTL Practical 10 Dr Tan Joi San OTL3 ||| UCCD1004 OTL Practical 35 Dr Kh'ng Xin Yi OTL14 |||
K5	UCCD1004 OTL Practical 10 Dr Tan Joi San OTL3 ||| UCCD1004 OTL Practical 35 Dr Kh'ng Xin Yi OTL14 |||
H6	UCCD1013 LESSON Tutorial 35 Ms Ng Wan Qing N002 ||| UCCD1013 OTL Tutorial 40 Dr Mogana a/p Vadiveloo OTL4 |||
D13	UCCM1353 LESSON Tutorial 35 Ms Lim Shun Jinn N003 ||| UCCM1353 LESSON Tutorial 15 Mr Liew Kah Fai N005 |||
F13	UCCM1363 LESSON Tutorial 35 Mr Lim Foo Weng N002 ||| UCCM1363 LESSON Tutorial 35 Ms Song Poh Choo N005 |||
F14	UCCD1203 LAB Practical 25 Ms Yap Seok Gee N010A ||| UCCD1203 LAB Practical 25 Pn Siti Hajar Binti Abrain N102 |||
G14	UCCD1203 LAB Practical 25 Ms Yap Seok Gee N010A ||| UCCD1203 LAB Practical 25 Pn Siti Hajar Binti Abrain N102 |||
B17	UCCN2243 OTL Practical 35 Ms Tan Lyk Yin OTL4 ||| UCCN2243 OTL Practical 35 Ts Dr Gan Ming Lee OTL7 |||
C17	UCCN2243 OTL Practical 35 Ms Tan Lyk Yin OTL4 ||| UCCN2243 OTL Practical 35 Ts Dr Gan Ming Lee OTL7 |||

Figure 5.2.6. 1 Proof of "The main subjects' classes should not be conducted at the same time"

From the students' timetables csv file produced, ctrl + f is used to find the cell which has multiple class events. This means that it is to find the timeslots which the students might have more than one option of class events to attend. From the observation on the timetable, it is known that in all the timeslots that have multiple class, the different class events are not colliding with each other. The different classes are all the same class event but separated to fit the students amount. The students can choose any of the class events to attend. For example, if there are multiple lecture classes existed in one timeslot for a student group, the violation will occur as only one lecture class for the student to attend, students cannot go both lecture classes at the same time. The concept of the above figure is like the scenario where P1 and P2 of a subject are held together and taught by different lecturers. The test case success.

5.2.7 Specific room is required to be considered (tutorial classroom/lab)

This test case is to ensure that different types of class events (Lecture, Practical, Tutorial) should be held in the specific classrooms. In this project, the events that held

in the OTL mode is not considered in this test case since the online platform used to conduct different types of the class events are the same.

Input data	Description	Expected result	Actual result	Status
Dataset1	Specific room is required to be considered (tutorial classroom/lab)	From the students' timetable produced, it is made sure that the TUTORIAL/PRACTICAL classes should be located inside the classroom with classroom types of TUTORIAL/ PRACTICAL respectively.	LECTURE classroom only consists of LECTURE class, TUTORIAL classroom only consist of TUTORIAL class, PRACTICAL classroom only consist of PRACTICAL class.	Pass

Table 5.2.7.1 Test Case of "Specific room is required to be considered (tutorial classroom/lab) "

Explanation:

Cell	Value
B3	UCCN1004 LAB Practical 25 Mr Teoh Shen Khang N011B |||
C3	UCCN1004 LAB Practical 25 Mr Teoh Shen Khang N011B |||
D3	UCCN1004 LECTURE Lecture 80 Dr Aun Yichiet LDK5 |||
F3	UCCM1153 LESSON Tutorial 35 Dr Lem Kong Hoong N002 |||
H3	UCCD1143 LESSON Tutorial 35 Ts Dr Lim Seng Poh N003 |||
J3	UCCD1143 LECTURE Lecture 80 Ts Dr Lim Seng Poh LDK1 |||
B4	UCCM1153 LESSON Tutorial 5 Ms Lim Shun Jinn N005 |||
D4	UCCD1143 LECTURE Lecture 80 Ts Dr Lim Seng Poh LDK5 |||
E4	UCCD1143 LECTURE Lecture 80 Ts Dr Lim Seng Poh LDK5 |||
F4	UCCN1004 LAB Practical 25 Ms Tan Lyk Yin N010B ||| UCCN1004 LAB Practical 25 Ts Dr Ooi Chek Yee N104 |||
G4	UCCN1004 LAB Practical 25 Ms Tan Lyk Yin N010B ||| UCCN1004 LAB Practical 25 Ts Dr Ooi Chek Yee N104 |||
H4	UCCD1013 LESSON Tutorial 5 Ms Tseu Kwan Lee N006 |||
J4	UCCD1013 LECTURE Lecture 80 Dr Mogana a/p Vadiveloo LDK3 |||
K4	UCCD1013 LECTURE Lecture 80 Dr Mogana a/p Vadiveloo LDK3 |||
B5	UCCD1143 LESSON Tutorial 35 Dr Kh'ng Xin Yi N003 |||
D5	UBMM1011 OTL Lecture 80 Dr Tee Chee Wee OTL14 |||
E5	UBMM1011 OTL Lecture 80 Dr Tee Chee Wee OTL14 |||
F5	UCCM1153 OTL Lecture 80 Dr Lem Kong Hoong OTL7 |||
H5	UCCM1153 OTL Tutorial 40 Ms Lim Shun Jinn OTL16 |||

Figure 5.2.7.1 Proof of "Specific room is required to be considered (tutorial classroom/lab)"

From the figure above, the lecture class events are all held in the classrooms start with the name ‘LDK’ which in the input data stands for the lecture hall name. The tutorial classes are held from the classroom ‘N001’ to classroom ‘N006’ where these classrooms are the tutorial classrooms. Whereas the practical classes are held from ‘N008A’ to ‘N104’ which are the practical classrooms. Lastly, for the OTL class events, there is no such restriction on it. The test case success.

5.2.8 The available periods per day is maximum at 10

This test case is to ensure that in all the study days, there is no timeslots in a day that exceed 10.

Input data	Description	Expected result	Actual result	Status
Dataset1	The available periods per day is maximum at 10	This constraint is initially defined. From the timetable produces, there should not have any	There are no more than 10 slots exist in any timetable.	Pass

		timetables having more than 10 periods, assuming that 1 period is 1 hour.		

Table 5.2.8.1 Test Case of "The available periods per day is maximum at 10 "

Explanation:

Cik Nor Fatiha Binti Subri										
Days|Timesl	8AM - 9AM	9AM - 10AM	10AM - 11AI	11AM - 12PI	12PM - 1PM	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM
Monday										
Tuesday										
Wednesday										
Thursday										
Friday										
Cik Norazira binti A Jalil										
Days|Timesl	8AM - 9AM	9AM - 10AM	10AM - 11AI	11AM - 12PI	12PM - 1PM	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM
Monday										
Tuesday										
Wednesday					UCCD1203 LI	UCCD1203 LECTURE Lecture 120 2022MAY LDK1 |||				
Thursday										
Friday										
Dr Ashvaany a/p Egambaram										
Days|Timesl	8AM - 9AM	9AM - 10AM	10AM - 11AI	11AM - 12PI	12PM - 1PM	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM
Monday										
Tuesday										
Wednesday										
Thursday										
Friday										
Dr Aun Yichiet										
Days|Timesl	8AM - 9AM	9AM - 10AM	10AM - 11AI	11AM - 12PI	12PM - 1PM	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM
Monday			UCCN1004 LECTURE Lecture 80 2022JAN LDK5 |||							
Tuesday										
Wednesday										

Figure 5.2.8. 1 Proof of "The available periods per day is maximum at 10"

From the figure 5.2.8.1 above, for every lecturer, the timeslots are from '8AM-9AM' until '5PM-6PM'. A total of 10 timeslots in each day for all lecturers. The test case success.

5.2.9 Each lecturer's working hours are limited during a week

This test case is to ensure that for the lecturers teaching the same course, their differences in the number of class events handled should not be more than 1, after excluding the lecture class events that are mostly handled by only one specific lecturer.

Input data	Description	Expected result	Actual result	Status
Dataset1	Each lecturer's working hours are limited during a week	From the courses' timetables, for the lecturers teaching the same courses, the number of timeslots taken by each lecturer should be the same/the differences are only one	After subtracting the lecture class, the tutorial or practical classes are distributed evenly for the lecturers in charged.	Pass

Table 5.2.9.1 Test Case of "Each lecturer's working hours are limited during a week "

Explanation:

UCCD1203											
Days|Time	8AM - 9AM	9AM - 10A	10AM - 11	11AM - 12	12PM - 1P	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM	
Monday											
Tuesday	Cik Norazi	Cik Norazira binti A Jalil LECTURE Lecture 120 2022MAY LDK5 |||									
Wednesda	Dr Suthash	Dr Suthashini a/p Subramaniam I	Ts Dr Chan	Ts Dr Chan Lee Kwun LAB Practical 20 2022MAY N101 |||							
Thursday			Pn Siti Haja	Pn Siti Hajar Binti Abrain LAB Practical 25 2022MAY N010B ||| Dr Nur Syadhila binti Che Lah LAB Practical 25 2022MAY N011B ||						
Friday											
UCCD2003											
Days|Time	8AM - 9AM	9AM - 10A	10AM - 11	11AM - 12	12PM - 1P	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM	
Monday	Ts Dr Ku Chin Soon OTL Tutorial 50 2022OC	Ts Dr Ku Chin Soon OTL Lecture 100 2022OCT OTL11 |||									
Tuesday											
Wednesday									Mr Tou Jing Yi OTL Tutorial 50 2022OCT OTL4 |||		
Thursday											
Friday	Ts Dr Ku Ch	Ts Dr Ku Chin Soon OTL Lecture 100 2022OCT OTL4 |||									
UCCD2044											
Days|Time	8AM - 9AM	9AM - 10A	10AM - 11	11AM - 12	12PM - 1P	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM	
Monday									Dr Chai Me	Dr Chai Meei Tyng OTL Lecture 100 2022OCT OTL14 |||	
Tuesday	Dr Sayed Ahmad Zikri Bin Sayed Aluwee OTL Tutorial 50 2022OCT OTL14 |||								Dr Ng Hui Fuang OTL Tutorial 50 2022OCT OTL13 |||		
Wednesday											
Thursday											
Friday											
UCCD2103											
Days|Time	8AM - 9AM	9AM - 10A	10AM - 11	11AM - 12	12PM - 1P	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM	
Monday						Ts Wong C	Ts Wong Chee Siang OTL Lecture 100 2022OCT OTL10 |||				
Tuesday			Ts Dr Tan Hung Khoon OTL Tutorial 50 2022OCT OTL16 |||								
Wednesday			Ts Wong Chee Siang OTL Lecture 100 2022OCT OTL1 |||								
Thursday	Ts Wong Chee Siang OTL Tutorial 50 2022OCT OTL5 |||										

Figure 5.2.9. 1 Proof of "Each lecturer's working hours are limited during a week"

For every lecturers taking the same subject, after excluding the lecture class events, it appears that the difference of the workloads in terms of number of events handled are similar with the biggest difference of one. The test case success.

5.2.10 Friday session stops by 12pm to 2pm

This test case is to ensure all the class events should not be held between 12pm to 2pm on Friday for the purpose of prayers of Muslims staffs and students.

Input data	Description	Expected result	Actual result	Status
Dataset1	Friday session stops by 12pm to 2pm	From all timetables, it is observed that for timeslot of 1200-1400 on Friday, the slot should be empty	No events in all timetables Friday 1200-1400 slot	Pass

Table 5.2.10.1 Test Case of "Friday session stops by 12pm to 2pm "

Explanation:

LDK1	300										
Days|Time	8AM - 9AI	9AM - 10A	10AM - 11	11AM - 12	12PM - 1P	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM	
Monday	-	-	-	-	-	-	-	-	-	-	
Tuesday	[2022JAN	-	-	-	-	-	-	-	-	-	
Wednesda	-	-	-	-	[2022JAN	[2022JAN	-	-	-	-	
Thursday	-	-	-	-	-	-	-	-	-	-	
Friday	-	-	-	-	-	-	-	-	-	-	
LDK2	300										
Days|Time	8AM - 9AI	9AM - 10A	10AM - 11	11AM - 12	12PM - 1P	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM	
Monday	-	-	-	-	-	-	-	-	-	-	
Tuesday	-	-	-	-	-	-	-	-	-	-	
Wednesda	-	-	-	-	-	-	-	-	-	-	
Thursday	-	-	-	-	-	-	-	-	-	-	
Friday	-	-	-	-	-	-	[2022MAY	-	-	-	
LDK3	300										
Days|Time	8AM - 9AI	9AM - 10A	10AM - 11	11AM - 12	12PM - 1P	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM	
Monday	[2022JAN	-	-	-	-	-	-	-	-	-	
Tuesday	-	-	-	-	-	-	-	-	-	-	
Wednesda	-	-	-	-	-	-	-	-	-	-	
Thursday	-	-	[2022JAN	[2022JAN	-	-	-	-	[2022JAN	-	
Friday	-	-	-	-	-	-	-	-	-	-	
LDK4	300										
Days|Time	8AM - 9AI	9AM - 10A	10AM - 11	11AM - 12	12PM - 1P	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM	
Monday	[2022MAY	[2022MAY	-	-	-	-	-	-	-	-	
Tuesday	-	-	-	-	-	-	-	-	-	-	
Wednesda	-	-	-	-	-	-	-	-	-	-	
Thursday	-	-	-	-	[2022JAN	[2022JAN	-	-	-	-	
Friday	-	-	-	-	-	-	-	-	-	-	

Figure 5.2.10. 1 Proof of "Friday session stops by 12pm to 2-pm"

From the timetables observed, there is no classroom has class events from 12pm-2pm on Friday. The test case success.

5.2.11 The number of students in Online Teaching and Learning (OTL) lecture classes can be unlimited

This test case is to test that whether the OTL lecture classes is limiting the number of students to attend.

Input data	Description	Expected result	Actual result	Status
Dataset1	The number of students in Online Teaching and Learning (OTL) lecture classes can be unlimited	In the classrooms' timetables produced, for the OTL classes, there is no need to specify the number of students to attend the classes	The practical or tutorial classes for in the OTL mode did not get distributed to fit the students inside, just one class for all.	Pass

Table 5.2.11.1 Test Case of "The number of students in Online Teaching and Learning (OTL) lecture classes can be unlimited "

Explanation:

Cell	Value	Formu
J165	[2022JAN L1 80 UCCD1004 OTL Lecture Dr Tan Joi San]	
H168	[2022JAN L1 80 UCCM1153 OTL Lecture Dr Lem Kong Hoong]	
B182	[2022OCT L1 100 UCCD2003 OTL Lecture Ts Dr Ku Chin Soon]	
C182	[2022OCT L1 100 UCCD2003 OTL Lecture Ts Dr Ku Chin Soon]	
D187	[2022OCT L1 100 UCCD2103 OTL Lecture Ts Wong Chee Siang]	
H192	[2022JAN L1 80 UCCD1004 OTL Lecture Dr Tan Joi San]	
I192	[2022JAN L1 80 UCCD1004 OTL Lecture Dr Tan Joi San]	
D201	[2022JAN L1 80 UCCM1153 OTL Lecture Dr Lem Kong Hoong]	
E201	[2022JAN L1 80 UCCM1153 OTL Lecture Dr Lem Kong Hoong]	
F214	[2022OCT L1 100 UCCN2243 OTL Lecture Ts Dr Gan Ming Lee]	
G214	[2022OCT L1 100 UCCN2243 OTL Lecture Ts Dr Gan Ming Lee]	
D224	[2022JAN L1 80 UBMM1011 OTL Lecture Dr Tee Chee Wee]	
E224	[2022JAN L1 80 UBMM1011 OTL Lecture Dr Tee Chee Wee]	
D248	[2022MAY L1 120 UCCM1353 OTL Lecture Mr Liew Kah Fai]	
E248	[2022MAY L1 120 UCCM1353 OTL Lecture Mr Liew Kah Fai]	
H248	[2022OCT L1 100 UCCD2103 OTL Lecture Ts Wong Chee Siang]	
I248	[2022OCT L1 100 UCCD2103 OTL Lecture Ts Wong Chee Siang]	
F255	[2022OCT L1 100 UCCD2003 OTL Lecture Ts Dr Ku Chin Soon]	
B265	[2022MAY L1 120 MPU3113 OTL Lecture Ms Yuvashini a/p Salvamani]	
C265	[2022MAY L1 120 MPU3113 OTL Lecture Ms Yuvashini a/p Salvamani]	
J273	[2022OCT L1 100 UCCD2303 OTL Lecture Dr Suthashini a/p Subramaniam]	
K273	[2022OCT L1 100 UCCD2303 OTL Lecture Dr Suthashini a/p Subramaniam]	
J276	[2022OCT L1 100 UCCD2044 OTL Lecture Dr Chai Meei Tyng]	
K276	[2022OCT L1 100 UCCD2044 OTL Lecture Dr Chai Meei Tyng]	
H279	[2022MAY L1 120 UCCM1353 OTL Lecture Mr Liew Kah Fai]	
D287	[2022MAY L1 120 MPU3113 OTL Lecture Ms Yuvashini a/p Salvamani]	

Figure 5.2.11. 1 Proof of "The number of students in Online Teaching and Learning (OTL) lecture classes can be unlimited"

From the figure 5.2.11.1 above, we can see that for all the OTL lecture class events, the student number per class events are the maximum number of the students. The test case success.

5.2.12 Classes which require physical attempt cannot be in OTL mode

This test case is to ensure that if the users do not set the OTL mode for a specific class event, then the event should not be held in the OTL mode.

Input data	**Description**	**Expected result**	**Actual result**	**Status**
Dataset1	Classes which require physical attempt cannot be in OTL mode	For the classes with specific requirement to be attended physically, physical classroom should be assigned to it	The result is based on the system users input to the database. For the course that require physical mode, the OTL mode class will not be assigned to it.	Pass

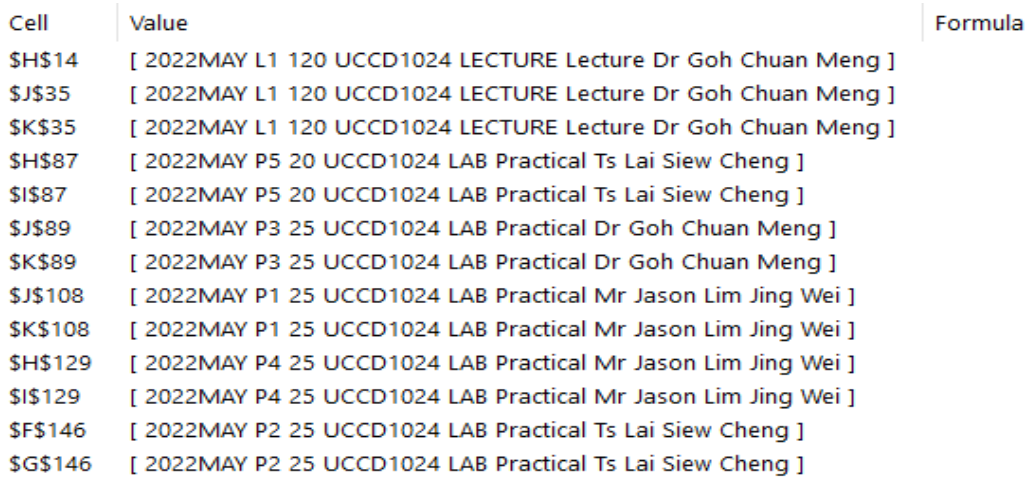
Table 5.2.12.1 Test Case of "Classes which require physical attempt cannot be in OTL mode "

Explanation:

	courseCode	numLectures	numTutorials	numPracticals	lectureOTL	lessonOTL	practicalOTL
Edit Copy Delete	UCCD1024	3	0	2	0	0	0

Figure 5.2.12. 1 Proof of "Classes which require physical attempt cannot be in OTL mode" [1]

From the figure 5.2.12.1 above, we can see that the lectureOTL and practicalOTL columns for the course 'UCCD1024' is 0. This means that there is no class events from UCCD1024 should be held in OTL mode.



Cell	Value	Formula
H14	[2022MAY L1 120 UCCD1024 LECTURE Lecture Dr Goh Chuan Meng]	
J35	[2022MAY L1 120 UCCD1024 LECTURE Lecture Dr Goh Chuan Meng]	
K35	[2022MAY L1 120 UCCD1024 LECTURE Lecture Dr Goh Chuan Meng]	
H87	[2022MAY P5 20 UCCD1024 LAB Practical Ts Lai Siew Cheng]	
I87	[2022MAY P5 20 UCCD1024 LAB Practical Ts Lai Siew Cheng]	
J89	[2022MAY P3 25 UCCD1024 LAB Practical Dr Goh Chuan Meng]	
K89	[2022MAY P3 25 UCCD1024 LAB Practical Dr Goh Chuan Meng]	
J108	[2022MAY P1 25 UCCD1024 LAB Practical Mr Jason Lim Jing Wei]	
K108	[2022MAY P1 25 UCCD1024 LAB Practical Mr Jason Lim Jing Wei]	
H129	[2022MAY P4 25 UCCD1024 LAB Practical Mr Jason Lim Jing Wei]	
I129	[2022MAY P4 25 UCCD1024 LAB Practical Mr Jason Lim Jing Wei]	
F146	[2022MAY P2 25 UCCD1024 LAB Practical Ts Lai Siew Cheng]	
G146	[2022MAY P2 25 UCCD1024 LAB Practical Ts Lai Siew Cheng]	

Figure 5.2.12. 2 Proof of "Classes which require physical attempt cannot be in OTL mode" [2]

From the snapshot from the classroom timetable csv file, it is observed that the lecture and practical classes of the course 'UCCD1024' are all held in physical LECTURE and LAB. The test case success.

5.3 System Testing Results for the Soft Constraints Defined

In this section, to test for soft constraints, the test plan is based on the condition where the soft constraints must be fulfilled completely. After generating the timetable, the timetable was tested to ensure that it follows the constraints. The test plans' results will be conducted by observing the csv timetable files produced based on different scenarios.

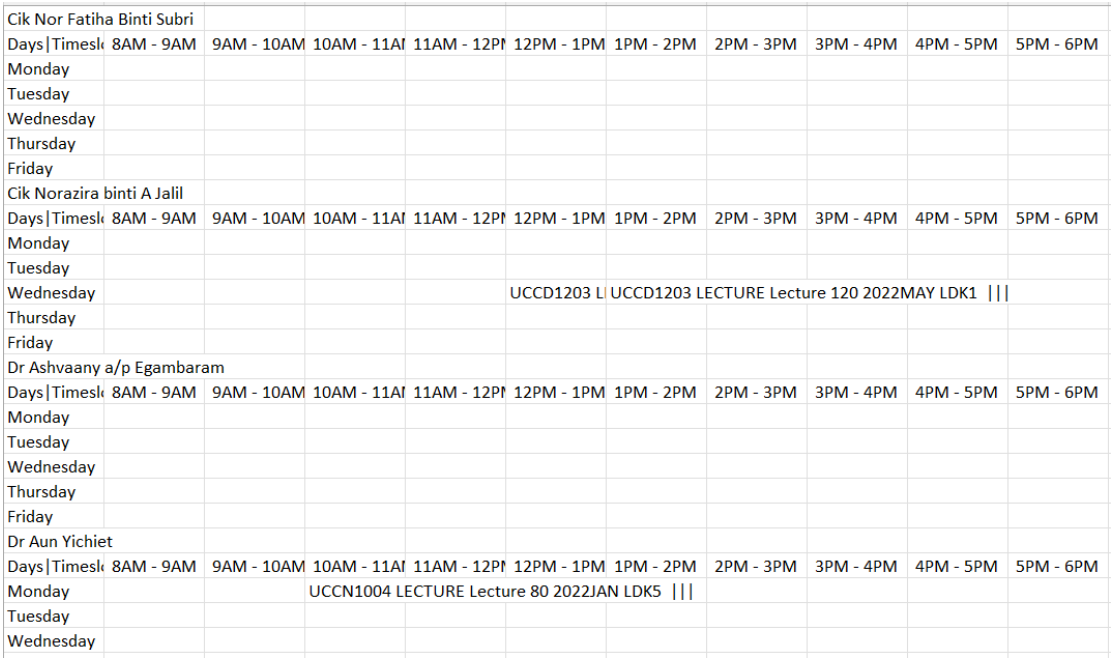
5.3.1 Classes should not be held after 6pm

This test case is to test if a class event is held after 6pm.

Input data	Description	Expected result	Actual result	Status
Dataset1	Classes should not be held after 6pm	From all the timetables produced, there should not be classes held after 6pm.	From the original structure of the timetable, no slots after 6pm are considered as a valid timeslot.	Pass

Table 5.3.1.1 Test Case of "Classes should not be held after 6pm "

Explanation:



Cik Nor Fatiha Binti Subri										
Days|Timesl	8AM - 9AM	9AM - 10AM	10AM - 11AI	11AM - 12PI	12PM - 1PM	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM
Monday										
Tuesday										
Wednesday										
Thursday										
Friday										
Cik Norazira binti A Jalil										
Days|Timesl	8AM - 9AM	9AM - 10AM	10AM - 11AI	11AM - 12PI	12PM - 1PM	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM
Monday										
Tuesday										
Wednesday					UCCD1203 LI	UCCD1203 LECTURE Lecture 120 2022MAY LDK1 |||				
Thursday										
Friday										
Dr Ashvaany a/p Egambaram										
Days|Timesl	8AM - 9AM	9AM - 10AM	10AM - 11AI	11AM - 12PI	12PM - 1PM	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM
Monday										
Tuesday										
Wednesday										
Thursday										
Friday										
Dr Aun Yichiet										
Days|Timesl	8AM - 9AM	9AM - 10AM	10AM - 11AI	11AM - 12PI	12PM - 1PM	1PM - 2PM	2PM - 3PM	3PM - 4PM	4PM - 5PM	5PM - 6PM
Monday			UCCN1004 LECTURE Lecture 80 2022JAN LDK5 |||							
Tuesday										
Wednesday										

Figure 5.3.1. 1 Proof of "Classes should not be held after 6pm"

From the figure 5.3.1.1 above, for every lecturer, the timeslots are from '8AM-9AM' until '5PM-6PM'. The last timeslot to hold an event is from '5PM-6PM'. Hence, no events can be hold after 6pm. The test case success.

5.3.2 General course should be opened to all faculty students

The test case is to test if all students can bid the general course without any restriction.

Input data	Description	Expected result	Actual result	Status
Dataset1	General course should be opened to all faculty students	From the students' timetables produced, any general course will not be abandoned from any student group	As the system user insert the student group's data that match with the general course to be taken, the timetable generator will follow it	Pass

Table 5.3.2.1 Test Case of "General course should be opened to all faculty students"

Explanation:

This test case is not observable as this is based on the input of the user's data into the database. To ensure the general course opened to all students, the user needs to key in the student groups' data which includes that course to be bid by the students. The test case success.

5.3.3 No subject has more than one lecture a day

The test case is to ensure that for a course, there should not have more than one lecture class event held in a single day.

Input data	Description	Expected result	Actual result	Status
Dataset1	No subject has more than one lecture a day	From the course' timetable produced, it is made sure that in each day, there are no course that has more than one lecture class.	There are no courses have more than one lecture class per day	Pass

Table 5.3.3.1 Test Case of "No subject has more than one lecture a day "

Explanation:

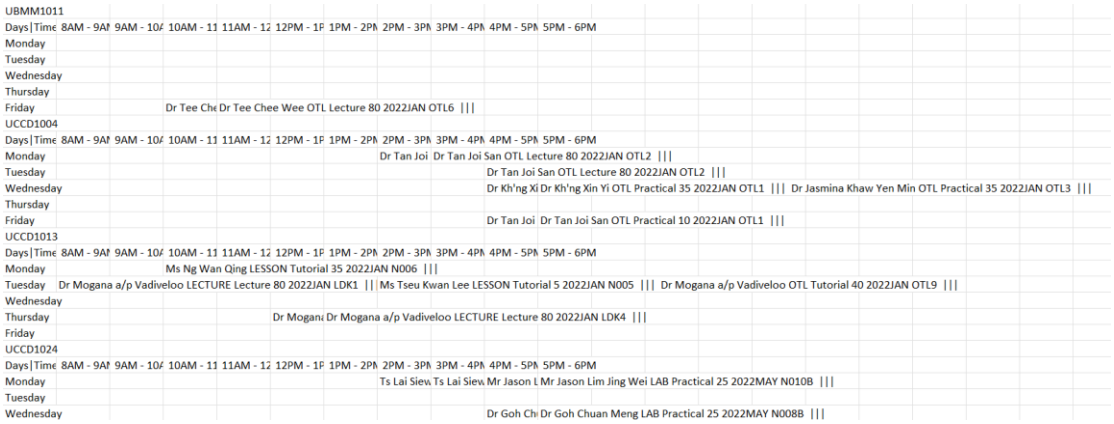


Figure 5.3.3. 1 Proof of "No subject has more than one lecture a day"

From the observation on the courses' timetable csv file, it is observed that all subjects have no more than one lecture class events in a single day. The test case success.

5.3.4 The daily teaching sessions of the lecturer should not exceed a specified number of periods

This test case is to ensure that for each lecturer, they should not have more than 4 class events to handle per day.

Input data	Description	Expected result	Actual result	Status
Dataset1	The daily teaching sessions of the lecturer should not exceed a specified number of periods	From the lecturers' timetable produced, it is made sure that the lecturers have no more than 4 class events to handle	For every lecturer, there is no existence that one lecturer needs to teach more than 4 classes each day	Pass

Table 5.3.4.1 Test Case of "The daily teaching sessions of the lecturer should not exceed a specified number of periods "

Explanation:

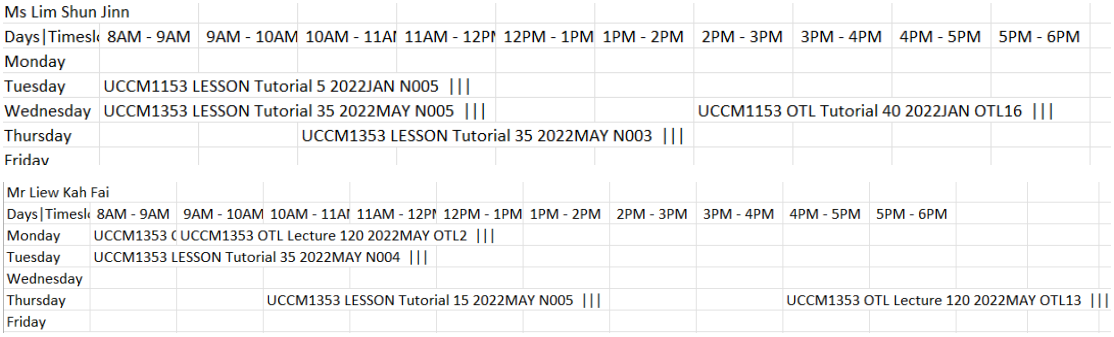


Figure 5.3.4. 1 Proof of "The daily teaching sessions of the lecturer should not exceed a specified number of periods"

For the two lecturers in the figure 5.3.4.1 above, they have 4 class events to handle per week. However, the class event gap between the class events are high. This test case is abstract to be observed, however, by observing all the lecturers' timetbales, it is found that there is no lecturer has more than 4 classes per day.

5.4 System Testing Results for the Hybrid Class Generation

In this section, the test plan is for the Hybrid Classes Generation which is used to indicate whether the hybrid timetable generation is workable by making sure there are no errors such as the repetition of the same class events in both physical and OTL mode, different distribution of the classes for the modes based on user's database data, etc.

5.4.1 The lecture class should either be conducted full in OTL mode or full in physical mode

The test case is to make sure that the lecture class events are either in full OTL mode or in full physical mode.

Input data	Description	Expected result	Actual result	Status
Dataset1	The lecture class should either be conducted full in OTL mode or full in physical mode	From the courses' timetables produced, the lecture classes are either in full OTL mode or full physical mode.	The result is based on the user's input to the database. If the user set 100% input to the lecture OTL column, the lecture class is full in OTL mode and vice versa.	Pass

Table 5.4.1.1 Test Case of "The lecture class should either be conducted full in OTL mode or full in physical mode"

Explanation:

This test case can be ensured by the input of the users. The user need to either input either 0 or 100 in the particular course's lectureOTL column in the database. After

that, the timetable tool will automatically assign the lecture class events either to full OTL mode or full physical mode.

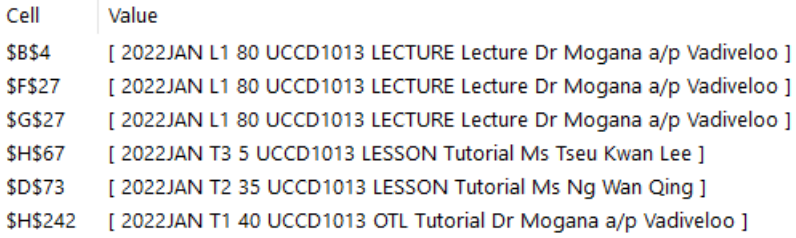
	courseCode	numLectures	numTutorials	numPracticals	lectureOTL	lessonOTL	practicalOTL
Edit Copy Delete	UCCD1004	3	0	2	100	0	100
Edit Copy Delete	UCCD1013	3	1	0	0	50	0

Figure 5.4.1. 1 Proof of "The lecture class should either be conducted full in OTL mode or full in physical mode" [1]

From the figure 5.4.1.1 above, the lectureOTL column of UCCD1004 is 100 while the lectureOTL column of UCCD1013 is 0.

Cell	Value	For...
J159	[2022JAN P2 35 UCCD1004 OTL Practical Dr Kh'ng Xin Yi]	
K159	[2022JAN P2 35 UCCD1004 OTL Practical Dr Kh'ng Xin Yi]	
J161	[2022JAN P3 10 UCCD1004 OTL Practical Dr Tan Joi San]	
K161	[2022JAN P3 10 UCCD1004 OTL Practical Dr Tan Joi San]	
J165	[2022JAN L1 80 UCCD1004 OTL Lecture Dr Tan Joi San]	
J173	[2022JAN P1 35 UCCD1004 OTL Practical Dr Jasmina Khaw Yen Min]	
K173	[2022JAN P1 35 UCCD1004 OTL Practical Dr Jasmina Khaw Yen Min]	
H192	[2022JAN L1 80 UCCD1004 OTL Lecture Dr Tan Joi San]	
I192	[2022JAN L1 80 UCCD1004 OTL Lecture Dr Tan Joi San]	

Figure 5.4.1. 2 Proof of "The lecture class should either be conducted full in OTL mode or full in physical mode" [2]



Cell	Value
B4	[2022JAN L1 80 UCCD1013 LECTURE Lecture Dr Mogana a/p Vadiveloo]
F27	[2022JAN L1 80 UCCD1013 LECTURE Lecture Dr Mogana a/p Vadiveloo]
G27	[2022JAN L1 80 UCCD1013 LECTURE Lecture Dr Mogana a/p Vadiveloo]
H67	[2022JAN T3 5 UCCD1013 LESSON Tutorial Ms Tseu Kwan Lee]
D73	[2022JAN T2 35 UCCD1013 LESSON Tutorial Ms Ng Wan Qing]
H242	[2022JAN T1 40 UCCD1013 OTL Tutorial Dr Mogana a/p Vadiveloo]

Figure 5.4.1. 3 Proof of "The lecture class should either be conducted full in OTL mode or full in physical mode" [3]

From the capture of the classroom's timetable, the lecture classes of UCCD1004 are all in OTL mode whereas the lecture classes of UCCD1013 are all in physical mode. The big capital LECTURE stands for the physical lecture class, whereas OTL stands for online class. The test case success.

5.4.2 The tutorial and practical classes study mode can be dynamically decided by the users.

This test case is to test that if the user can dynamically allocate the ratio of OTL mode to physical mode of a course for their tutorial or practical class events.

Input data	Description	Expected result	Actual result	Status
Dataset1	The tutorial and practical classes study mode can be dynamically decided by the users.	From the courses' timetables produced, the tutorial or practical classes can be all in OTL mode or physical mode, or half in OTL mode or physical mode, or in any other ratio of study mode distribution.	The result is based on the user's input to the database. If the user set 100% input to the tutorial or practical OTL column, the tutorial or practical class is full in OTL mode and vice versa. Other ratio of study mode distribution are also acceptable by keying other percentage values. The distribution of the study modes of the course in the courses' timetables are same as the setting of the user's input to the database.	Pass

Table 5.4.2.1 Test Case of "The tutorial and practical classes study mode can be dynamically decided by the users. "

This test case is again based on the users' input to the data course in their lessonOTL or practicalOTL columns. The examples showed below:

Cell	Value
B4	[2022JAN L1 80 UCCD1013 LECTURE Lecture Dr Mogana a/p Vadiveloo]
F27	[2022JAN L1 80 UCCD1013 LECTURE Lecture Dr Mogana a/p Vadiveloo]
G27	[2022JAN L1 80 UCCD1013 LECTURE Lecture Dr Mogana a/p Vadiveloo]
H67	[2022JAN T3 5 UCCD1013 LESSON Tutorial Ms Tseu Kwan Lee]
D73	[2022JAN T2 35 UCCD1013 LESSON Tutorial Ms Ng Wan Qing]
H242	[2022JAN T1 40 UCCD1013 OTL Tutorial Dr Mogana a/p Vadiveloo]

Figure 5.4.2. 1 Proof of "The tutorial and practical classes study mode can be dynamically decided by the users"

For UCCD1013 which has total of 3 lessons and total students of 80, after assigning 50% of OTL tutorial classes, 40 students are allocated for the tutorial class in OTL mode, and 40 students are allocated to physical mode. The test case success.

5.5 System Testing Results for Timetable Generation

In this section, the test plan is to do the experiment related to the speed of the timetable generation using genetic algorithm. Different number of class events are used to conduct the test plan, with the manipulated variable of the number of constraints selected. Here, the constraints type includes only the constraints that are selectable by the user. The resources mentioned later is the sum of the timeslots of all classrooms available. The terms include in the test plan are:

Class event size: resources size: This term defines the ratio of class event size to the size of the resources

Constraints: This term defines the constraints to be selected for the timetable generation

Min time: The minimum amount of time to generate the timetable for current scenario

Max time: The maximum amount of time to generate the timetable for current scenario

Average time: The average amount of time to generate the timetable for current scenario

GA population generated: The average number of population generated by the Genetic algorithm in current scenario.

Explanation: to explain the scenario and the result obtained.

5.5.1 Generate the timetable with small class events amount (10% of resources) and 2 constraints involved

Class event size : resources size	Constraints	Min time	Max time	Average time	GA population generated
100 : 1050	1. Each lecturer can only teach one class at a time 2. Specific room is required to be considered	0.27 second	0.3 second	0.285 second	1

Table 5.5.1.1 Test Case of "Generate the timetable with small class events amount (10% of resources) and 2 constraints involved"

Explanation:

While running this test case, 2 times of timetable generation were done. The maximum time used to generate the timetable is 0.2 second. While the minimum time used to generate the timetable is also 02 second. Hence, the total of 0.2 second average time required to generate the timetable under this test case. The figure 5.5.1.1 below shows the results obtained from the console.

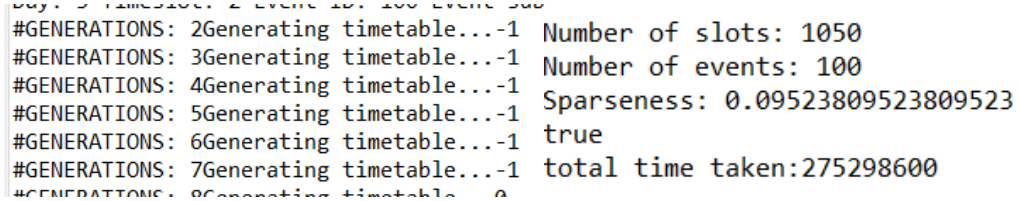


Figure 5.5.1. 1 Proof of "Generate the timetable with large class events amount (100 class events, 1050 resources) and 2 constraints (each lecturer can only teach one class at a time, Specific room is required to be considered) involved"

5.5.2 Generate the timetable with large class events amount (40% of resources) and 2 constraints involved

From the above test case, the input data to the timetable generation are the same with the previous test cases.

Class event size : resources size	**Constraints**	**Min time**	**Max time**	**Average time**	**GA population generated**
100 : 252	1. Each lecturer can only teach one class at a time 2. Specific room is required to be considered	2 seconds	2.1 seconds	2.05 seconds	29

Table 5.5.2.1 Test Case of "Generate the timetable with large class events amount (40% of resources) and 2 constraints involved"

Explanation:

While running this test case, 2 times of timetable generation were done. The maximum time used to generate the timetable is 2 second. While the minimum time used to generate the timetable is also 2.1 second. Hence, the total of 2.05 second average time required to generate the timetable under this test case. The figure 5.5.2.1 below shows the results obtained from the console.

```
#GENERATIONS: 21Generating timetable...-3
#GENERATIONS: 22Generating timetable...-3
#GENERATIONS: 23Generating timetable...-2
#GENERATIONS: 24Generating timetable...-2
#GENERATIONS: 25Generating timetable...-1
#GENERATIONS: 26Generating timetable...-1
#GENERATIONS: 27Generating timetable...-1
#GENERATIONS: 28Generating timetable...-1
#GENERATIONS: 29Generating timetable...0
Number of slots: 252
Number of events: 100
Sparseness: 0.3968253968253968
true
total time taken:2014641500
```

Figure 5.5.2. 1 Proof of "Generate the timetable with small class events amount (100 class events, 252 resources) and 2 constraints (each lecturer can only teach one class at a time, Friday session stops by 12pm to 2pm) involved"

5.5.3 Generate the timetable with small class events amount (10% of resources) and 4 constraints involved

From the above test case, the input data to the timetable generation are the same with the previous test cases.

Class event size : resources size	**Constraints**	**Min time**	**Max time**	**Average time**	**GA population generated**
100 : 1050	1. Each lecturer can only teach one class at a time 2. Specific room is required to be considered 3. room capacity should not be exceeded 4. Friday session stops by 12pm to 2pm	1.9 seconds	2.1 seconds	2 seconds	23

Table 5.5.3.1 Test Case of "Generate the timetable with small class events amount (10% of resources) and 4 constraints involved"

Explanation:

While running this test case, 2 times of timetable generation were done. The maximum time used to generate the timetable is 2.1 second. While the minimum time used to generate the timetable is also 1.9 second. Hence, the total of 2 second average time required to generate the timetable under this test case.

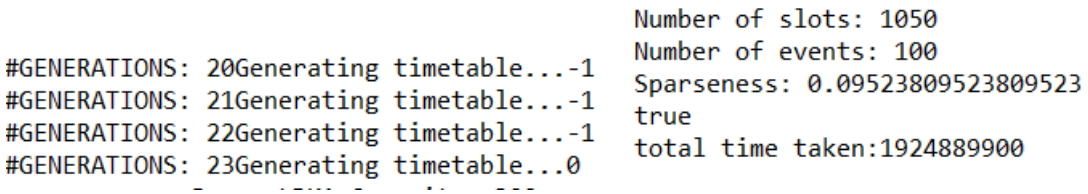


Figure 5.5.3. 1 Proof of "Generate the timetable with small class events amount (100 class events, 1050 resources) and 4 constraints (each lecturer can only teach one class at a time, Specific room is required to be considered, room capacity should not b

5.5.4 Generate the timetable with small class events amount (40% or resources) and 4 constraints involved

From the above test case, the input data to the timetable generation are the same with the previous test cases.

Class event size : resources size	**Constraints**	**Min time**	**Max time**	**Average time**	**GA population generated**
100 : 252	1. Each lecturer can only teach one class at a time 2. Specific room is required to be considered 3. room capacity should not be exceeded 4. Friday session stops by 12pm to 2pm	3.2 seconds	4.5 seconds	3.85 seconds	52

Table 5.5.4.1 Test Case of "Generate the timetable with small class events amount (40% or resources) and 4 constraints involved "

Explanation:

While running this test case, 2 times of timetable generation were done. The maximum time used to generate the timetable is 3.2 second. While the minimum time used to generate the timetable is also 4.5 second. Hence, the total of 3.85 second average time required to generate the timetable under this test case.

```
#GENERATIONS: 45Generating timetable...-1
#GENERATIONS: 46Generating timetable...-1
#GENERATIONS: 47Generating timetable...-1
#GENERATIONS: 48Generating timetable...-1
#GENERATIONS: 49Generating timetable...-1
#GENERATIONS: 50Generating timetable...-1
#GENERATIONS: 51Generating timetable...-1
#GENERATIONS: 52Generating timetable...0
```

```
Number of slots: 252
Number of events: 100
Sparseness: 0.3968253968253968
true
total time taken:3197208500
```

Figure 5.5.4. 1 Proof of "Generate the timetable with small class events amount (100 class events, 252 resources) and 4 constraints (each lecturer can only teach one class at a time, Specific room is required to be considered, room capacity should not be

5.5.5 Generate the timetable with same student amounts and resources amount and 4 constraints involved

The test case is to evaluate the performance of timetable generator when the number of class events is equal to the number of resources.

Class event size : resources size	**Constraints**	**Min time**	**Max time**	**Average time**	**GA population generated**
100 : 100	1. Each lecturer can only teach one class at a time 2. Specific room is required to be considered 3. room capacity should not be exceeded 4. Friday session stops by 12pm to 2pm	> 1 day and the generator could not generate the timetable	> 1 day and the generator could not generate the timetable	> 1 day and the generator could not generate the timetable	-

Table 5.5.5.1 Test Case od "Generate the timetable with same student amounts and resources amount and 4 constraints involved"

Explanation:

Since the problem size is as big as the resources, it can be predicted that the generation of the result will be taking very long time. In this test case, the result of the

timetable failed to be obtained even after one day. This is one limitation of the genetic algorithm used as the algorithm does not study the real problem happened but randomly swap the class events between the resources. This will lead to inconsistent improvement in the timetable generation, or even the class events that are violating the constraints are not handled by the algorithm, and hence resulting in the situation where the time taken to generate the timetable will be incredibly long with no improvement at certain point. The test case failed. No result to be showed.

5.4.6 Generate the timetable with class events more than the resources available

This test case is to test the situation when the number of class events is larger than the number of resources available. When the user's data inside the database will lead to the creation of class events that is more than the number of resources, an alert message will pop out to alert the user and the user is not allowed to continue the timetable generation. The test case success.

Class event size : resources size	**Constraints**	**Min time**	**Max time**	**Average time**	**GA population generated**
120 : 100	*constraints here is not used 1. Each lecturer can only teach one class at a time 2. Specific room is required to be considered 3. room capacity should not be exceeded 4. Friday session stops by 12pm to 2pm	-	-	-	-

Table 5.4.6.1 Test Case of "Generate the timetable with class events more than the resources available"

Explanation:

The user is not allowed to proceed to the timetable generation after he loaded the data. The test case success.

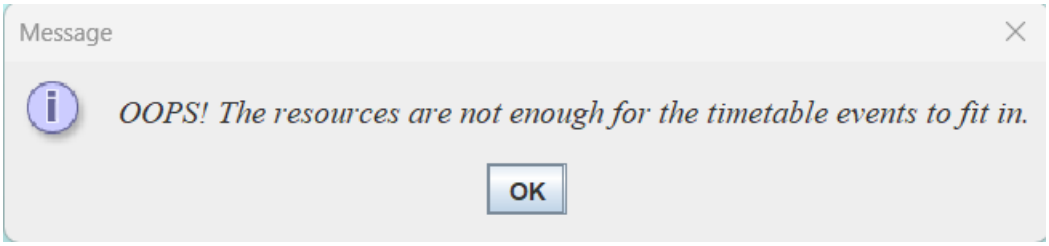


Figure 5.8.6. 1 Proof of "Generate the timetable with class events more than the resources available"

Chapter 6

Discussion

6.1 Discussion on the System Performance in Achieving the Objective

In this section, the system performance will be elaborated and evaluated based on the test plans and their results in Chapter 5.

Based on the test plan in Chapter 5, the University Class Timetable Tool is considered as a functionable tool that can help the user to generate the timetable based on the data in the database provided by the user. The hard constraints involved in this project can be fulfilled by the timetable generator where non crashes of the class events in the timetable generated. For all the hard constraints created and considered, by observing the timetables produced which are classroom timetable, student timetable, lecturer timetable, and course timetable, all the hard constraints proposed were being fulfilled by the timetable generated.

Besides, for the soft constraints, although the soft constraints are not the rules that must be fulfilled in generating the timetable, in this project, soft constraints were tested as the rules that must be fulfilled during the implementation of the system. And the test plans of all the soft constraints show also the positive results.

After that, the hybrid classes consideration which added some new conditions to the timetable generation was successfully integrated into the system too.

For the performance of the algorithm to generate the timetable in term of speed, when the problem size (class events), the time required to generate the feasible timetable would also be increased. Besides, the number of hard constraints involved will also affect the speed of timetable generation where the more the hard constraints involved, the more time required to generate the timetable obeying the hard constraints.

To sum up, the system implemented did achieve all the objectives defined which are **to investigate the hard constraints and the soft constraints for the university class timetabling problems** and **to design a university class timetabling tool that will help the users to generate the timetable with hybrid class automatically using the genetic algorithm**. The hard constraints and soft constraints did successfully be

designed and implemented, besides the timetabling tool which can generate the timetable automatically using genetic algorithm with the involvement of hybrid mode also be implemented successfully.

6.4 System Novelties

In this project, there are a few novelties included during the designing and implementation of the system.

First, the system did consider the situation where the university may undergo the hybrid teaching and learning mode due to the pandemic of Covid-19. The include of the hybrid mode classes was considered as the database structure is designed to handle the hybrid mode, new constraints related to the hybrid mode were introduced, and the timetable generation algorithm was modified and adapted to handle the hybrid mode class events allocation. For example, the system allows the user to distribute the class events of a course into different percentage of OTL mode to percentage of physical mode ratio. The class events will be in hybrid mode where the student and lecturer may attend one from both kinds.

Besides, since the genetic algorithm is used, due to its nature of calculating the fitness score based on different constraints violations checking, the parallel programming was used to handle those different kinds of functions that handle different constraint violations.

After that, the users are provided with the function of selecting the desired constraints themselves. The user may choose the constraints they want to generate a timetable. This may help them to have more control on the timetable generated.

Finally, after generating the timetable, the users are allowed to modify the timetable generated by selecting the class event they may want to change and allocate the new time and day to it. The timetable generator will then help the user to check if there is available slot for the changes, following the constraints selected by the users.

6.3 System Limitations

One limitation found in this system is that the timetable generation will become longer and longer when the user increases the class events number. This is due to the nature of the genetic algorithm where the algorithm scheduling is based on the random swapping of the genes instead of study the real problems occurred during the generation. In this system, when the user adds a lot of class events into the system, the generation of the timetable will be taking unpredictable long time to finish.

6.4 Future Enhancement/Improvement

Since the limitation of the system is the nature of the genetic algorithm that may face the problem of long result generation, the future enhancement can be focused on the algorithm used. From past papers, among the algorithms discussed, the genetic algorithm is often used by many researchers to perform a heuristic approach. This is because all algorithms have their own weaknesses. The heuristic approach is an approach combines one algorithm with one or more other algorithms to cater with their own weaknesses, hence get higher performance on the result.

Besides the algorithm itself, the system can be improved in terms of the timetable generation. The system can be added with module that allows the user to manually modify the timetable generated instead of relying fully on the timetable generator. This may improve the experience and provide more room for the user when he wants to have more control on the timetable generation.

Lastly, some user-friendly features such as the guideline to use the system, the linkage of the system with the course registration system, the acceptance of the timetable data in forms of csv or text file, and more user-attractive user interface can be implemented in the system.

Chapter 7

Conclusion

To sum up, university class timetable tool is an automated tool that can help the staffs to generate the timetable in each trimester. Due to its NP-hard characteristic, the automated timetabling tool may help in speeding up the timetable generations, as well as increase the correctness of the timetable produced. There are a lot of algorithms introduced in previous work such as genetic algorithm [2], [4] [11], [12], [13], Constraints Satisfaction Programming (CSP) approach [13], Linear integer model[14], Tabu search algorithm [15], [16], Particle Swarm Optimization (PSO) [17], [18], Simulated Annealing (SA) [1], Ant Colony Optimization (ACO) [19], etc.

In this project, due to the changes of study mode during pandemic of Covid-19, there are new constraints to be considered in the timetable scheduling. The problem here is the lack of timetabling systems in the market that provide the feature of allocating the timetable into different study mode. Besides, there are lack of alternatives which can deal with this situation well. Hence, there is a need to design an automated timetable scheduling tool that can take these changes into the considerations, and produce the optimal timetable based on the newly added constraints.

There are several new constraints defined in this project to suit the current situation. For example, the allocation of students into different study mode with the same class events. The algorithm used in the system development is genetic algorithm that can greatly help in the automate the timetable scheduling by generating the optimal timetable. The genetic algorithm is suitable to solve the large size problem by selecting and interchanging the elements, as well as creating a sequence of new population generations.

Lastly, by designing the structure and methods to use in each phase in genetic algorithm, the effectiveness and efficiency of the algorithm will be changed. For example, Roulette Wheel selection, single point crossover, and swap mutation. The mutation rate and crossover rate were also being modified and trained to find the most suitable set to generate the timetable in a shorter time with the high correctness ensured.

REFERENCES

[1] A. M. Hambali, Y. A. Olasupo, and M. Dalhatu, "Automated university lecture timetable using Heuristic Approach," *Niger. J. Technol.*, vol. 39, no. 1, pp. 1–14, 2020, doi: 10.4314/njt.v39i1.1.

[2] P. Pongcharoen, W. Promtet, P. Yenradee, and C. Hicks, "Stochastic Optimisation Timetabling Tool for university course scheduling," *Int. J. Prod. Econ.*, vol. 112, no. 2, pp. 903–918, 2008, doi: 10.1016/j.ijpe.2007.07.009.

[3] R. A. Oude Vrielink, E. A. Jansen, E. W. Hans, and J. van Hillegersberg, "Practices in timetabling in higher education institutions: a systematic review," *Ann. Oper. Res.*, vol. 275, no. 1, pp. 145–160, 2019, doi: 10.1007/s10479-017-2688-8.

[4] S. Ghaemi, M. T. Vakili, and A. Aghagolzadeh, "Using a genetic algorithm optimizer tool to solve university timetable scheduling problem," *2007 9th Int. Symp. Signal Process. its Appl. ISSPA 2007, Proc.*, no. 1000, pp. 2–5, 2007, doi: 10.1109/ISSPA.2007.4555397.

[5] "aSc TimeTables - School Scheduling." .

[6] L. Lalescu, "{FET} - Free Timetabling Software." 2012, [Online]. Available: http://lalescu.ro/liviu/fet/.

[7] H. Rudová, T. Müller, and K. Murray, "Complex university course timetabling," *J. Sched.*, vol. 14, no. 2, pp. 187–207, 2011, doi: 10.1007/s10951-010-0171-3.

[8] V. Sahargahi and M. F. Drakhshi, "Comparing the Methods of Creating Educational Timetable," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 16, no. 12, pp. 26–36, 2016, [Online]. Available: http://paper.ijcsns.org/07_book/201612/20161204.pdf.

[9] T. Thepphakorn and P. Pongcharoen, "Performance improvement strategies on Cuckoo Search algorithms for solving the university course timetabling problem," *Expert Syst. Appl.*, vol. 161, p. 113732, 2020, doi:

10.1016/j.eswa.2020.113732.

[10] H. Alghamdi, T. Alsubait, H. Alhakami, and A. Baz, “A Review of Optimization Algorithms for University Timetable Scheduling,” *Eng. Technol. Appl. Sci. Res.*, vol. 10, no. 6, pp. 6410–6417, 2020, doi: 10.48084/etasr.3832.

[11] Z. Mansor, J. Arbain, M. Ashri, and A. Hassan, “Implementation of Fet Application in Generating a University Course and Examination Timetabling,” *Second Int. Conf. Onformation Technol. Bus. Appl.*, no. May 2014, p. 7, 2013.

[12] O. El Mahdi, R. N. Ainon, and R. Zainuddin, “Using a Genetic Algorithm optimizer tool to generate good quality timetables,” *Proc. IEEE Int. Conf. Electron. Circuits, Syst.*, vol. 3, pp. 1300–1303, 2003, doi: 10.1109/ICECS.2003.1301753.

[13] V. Sapru, K. Reddy, and B. Sivaselvan, “Time table scheduling using Genetic Algorithms employing guided mutation,” *2010 IEEE Int. Conf. Comput. Intell. Comput. Res. ICCIC 2010*, pp. 335–339, 2010, doi: 10.1109/ICCIC.2010.5705788.

[14] S. Daskalaki, T. Birbas, and E. Housos, “An integer programming formulation for a case study in university timetabling,” *Eur. J. Oper. Res.*, vol. 153, no. 1, pp. 117–135, 2004, doi: 10.1016/S0377-2217(03)00103-6.

[15] D. Costa, “A tabu search algorithm for computing an operational timetable,” *Eur. J. Oper. Res.*, vol. 76, no. 1, pp. 98–110, 1994, doi: 10.1016/0377-2217(94)90009-4.

[16] T. Islam, Z. Shahriar, M. A. Perves, and M. Hasan, “University Timetable Generator Using Tabu Search,” no. December 2016, 2018, doi: 10.4236/jcc.2016.416003.

[17] D. Palupi Rini, S. Mariyam Shamsuddin, and S. Sophiyati Yuhaniz, “Particle Swarm Optimization: Technique, System and Challenges,” *Int. J. Comput. Appl.*, vol. 14, no. 1, pp. 19–27, 2011, doi: 10.5120/1810-2331.

[18] Y. T. Kao and E. Zahara, “A hybrid genetic algorithm and particle swarm

optimization for multimodal functions," *Appl. Soft Comput. J.*, vol. 8, no. 2, pp. 849–857, 2008, doi: 10.1016/j.asoc.2007.07.002.

[19] K. Socha, M. Sampels, and M. Manfrin, "Ant algorithms for the university course timetabling problem with regard to the state-of-the-art," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2611, pp. 334–345, 2003, doi: 10.1007/3-540-36605-9_31.

[20] Y. Zhang, Q. Hu, Z. Meng, and A. Ralescu, "Fuzzy dynamic timetable scheduling for public transit," *Fuzzy Sets Syst.*, vol. 395, pp. 235–253, 2020, doi: 10.1016/j.fss.2019.01.010.

[21] S. Shetty, "Overview of Genetic Algorithm in Artificial Intelligence with Examples," *Greatlearning*. 2020, [Online]. Available: https://www.mygreatlearning.com/blog/introduction-to-genetic-algorithm/.

[22] Z. Lixi and L. SimKim, "Constructing university timetable using constraint satisfaction programming approach," *Proc. - Int. Conf. Comput. Intell. Model. Control Autom. CIMCA 2005 Int. Conf. Intell. Agents, Web Technol. Internet*, vol. 2, pp. 55–60, 2005, doi: 10.1109/cimca.2005.1631445.

[23] P. Pedamkar, "What is {Genetic} {Algorithm}? {\textbar} {Phases} and {Applications} of {Genetic} {Algorithm}." [Online]. Available: https://www.educba.com/what-is-genetic-algorithm/.

[24] A. Schatten, "Genetic Algorithms Short Tutorial." 2002, [Online]. Available: http://www.cs.ucdavis.edu/~vemuri/classes/ecs271/Genetic Algorithms Short Tutorial.htm.

[25] S. Rajan, "Genetic Algorithms and its use-cases in Machine Learning." 2021, [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/genetic-algorithms-and-its-use-cases-in-machine-learning/.

[26] A. K. Herath, "Genetic Algorithm For University Course Timetabling Problem," 2017.

[27] “Backtracking Algorithms - GeeksforGeeks.” [Online]. Available: https://www.geeksforgeeks.org/backtracking-algorithms/.

[28] S. Writer, “What are the advantages and disadvantages of Ergonomics?,” *Healthline*. pp. 1–4, 2016, [Online]. Available: https://www.reference.com/science/advantages-disadvantages-ergonomics-296fb450a073b13d.

[29] Roman Barták, “Constraint Guide - Systematic Search.” 2010, [Online]. Available: http://kti.mff.cuni.cz/~bartak/constraints/backtrack.html.

[30] V. Kumar, “Algorithms for constraint-satisfaction problems: a survey.,” *{AI} Mag.*, vol. 13, no. 1, pp. 32–44, 1992.

[31] R. C. Eberhart and Yuhui Shi, “Kernels on structured objects through nested histograms,” *Part. Swarm Optim. Dev. Appl. Resour. Russell*, 2001, doi: 10.7551/mitpress/7503.003.0046.

[32] B. Seixas Gomes de Almeida and V. Coppo Leite, “Particle Swarm Optimization: A Powerful Technique for Solving Engineering Problems,” *Swarm Intelligence - Recent Advances, New Perspectives and Applications*. 2019, doi: 10.5772/intechopen.89633.

[33] M. A. A. Aziz, M. N. Taib, and N. M. Hussin, “The effects of event selection based on soft constraint violation (ESSCV) in a modified PSO algorithm to solve class scheduling problems,” *ICCAIE 2010 - 2010 Int. Conf. Comput. Appl. Ind. Electron.*, no. Iccaie, pp. 584–587, 2010, doi: 10.1109/ICCAIE.2010.5735148.

[34] G. Hongwei, D. Wenli, and Q. Feng, “A hybrid algorithm based on particle swarm optimization and simulated annealing for job shop scheduling,” *Proc. - Third Int. Conf. Nat. Comput. ICNC 2007*, vol. 3, no. Icnc, pp. 715–719, 2007, doi: 10.1109/ICNC.2007.44.

[35] S. P. Lim, “Differential Evolution and Particle Swarm,” pp. 41–46, 2013.

[36] A. Chaudhuri and K. De, “Fuzzy genetic heuristic for university course

timetable problem," *Int. J. Adv. Soft Comput. its Appl.*, vol. 2, no. 1, pp. 100–123, 2010.

[37] H. Komaki, S. Shimazaki, K. Sakakibara, and T. Matsumoto, "Interactive optimization techniques based on a column generation model for timetabling problems of university makeup courses," *2015 IEEE 8th Int. Work. Comput. Intell. Appl. IWCIA 2015 - Proc.*, pp. 127–130, 2016, doi: 10.1109/IWCIA.2015.7449475.

[38] N. I. Ilham, E. H. M. Saat, N. H. A. Rahman, F. Yasmin, A. Rahman, and N. Kasuan, "Auto-Generate Scheduling System Based on Expert System," no. November, pp. 24–26, 2017.

[39] W. Erben and J. Keppler, "A genetic algorithm solving a weekly course-timetabling problem," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1153, pp. 198–211, 1996, doi: 10.1007/3-540-61794-9_60.

[40] J. P. Caldeira and a. C. Rosa, "School Timetabling using Genetic Search," *Proc. Int. Conf. Int. Conf. Pract. Theory Autom. Timetabling. PATAT 97*, pp. 115–122, 1997, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.1112.

[41] E. K. Burke and S. Petrovic, "Recent research directions in automated timetabling," *Eur. J. Oper. Res.*, vol. 140, no. 2, pp. 266–280, 2002, doi: 10.1016/S0377-2217(02)00069-3.

[42] "Pr rin t n ot pe er re v iew Pr ep rin t n ot pe er ed," no. 1, p. 16, 2021, [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3842490.

[43] Ihab Sbeity, Mohamed Dbouk, and Habib Kobeissi, "Combining the Analytical Hierarchy Process and the Genetic Algorithm To Solve the Timetable Problem," *Int. J. Softw. Eng. Appl.*, vol. 5, no. 4, pp. 39–50, 2014.

[44] R. Alvarez-Valdes, E. Crespo, and J. M. Tamarit, "Design and implementation

of a course scheduling system using Tabu Search," *Eur. J. Oper. Res.*, vol. 137, no. 3, pp. 512–523, 2002, doi: 10.1016/S0377-2217(01)00091-1.

[45] "iMagic Timetable Master - timetable software." .

[46] "» Download Mimosa Free." .

[47] "Introduction to JDBC (Java Database Connectivity) - GeeksforGeeks." .

APPENDIX

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: 2, 3	**Study week no.: 2**
Student Name & ID: Ang Wen Jie, 1900369	
Supervisor: Ts Dr Ku Chin Soon	
Project Title: University Class Timetable Tool	

1. WORK DONE

The testing of the system from FYP 1 is done.

2. WORK TO BE DONE

After testing the previous work, define the works to be done such as designing the UI, involving hybrid features, etc.

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

The progress seems progressing on track and goes smoothly.

Supervisor's signature

Student's signature

Trimester, Year: 2, 3	**Study week no.: 4**
Student Name & ID: Ang Wen Jie, 1900369	
Supervisor: Ts Dr Ku Chin Soon	
Project Title: University Class Timetable Tool	

1. WORK DONE

Adding the Hybrid feature to the system and test the features added. Add the constraints selection feature and test it.

2. WORK TO BE DONE

Add parallel programming to handle the constraints validation.

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

The progress seems progressing on track and goes smoothly.

_____ _____

Supervisor's signature Student's signature

Trimester, Year: 2, 3	**Study week no.: 6**
Student Name & ID: Ang Wen Jie, 1900369	
Supervisor: Ts Dr Ku Chin Soon	
Project Title: University Class Timetable Tool	

1. WORK DONE

Implementing the parallel programming feature and test it. Code the UI.

2. WORK TO BE DONE

Test and design more on UI.

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

The progress seems progressing on track and goes smoothly.

_____ _____

Supervisor's signature Student's signature

Trimester, Year: 2, 3	**Study week no.: 8**
Student Name & ID: Ang Wen Jie, 1900369	
Supervisor: Ts Dr Ku Chin Soon	
Project Title: University Class Timetable Tool	

1. WORK DONE

Keep testing on UI and refine the UI coding.

2. WORK TO BE DONE

Continue on UI and refine it.

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

The progress seems progressing on track and goes smoothly.

_____ _____

Supervisor's signature Student's signature

Trimester, Year: 2, 3	**Study week no.: 10**
Student Name & ID: Ang Wen Jie, 1900369	
Supervisor: Ts Dr Ku Chin Soon	
Project Title: University Class Timetable Tool	

1. WORK DONE

Working on UI and test the UI.

2. WORK TO BE DONE

Test the completed system and modify the report.

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

The progress seems progressing on track and goes smoothly.

_____ _____

Supervisor's signature Student's signature

Trimester, Year: 2, 3	**Study week no.: 12**
Student Name & ID: Ang Wen Jie, 1900369	
Supervisor: Ts Dr Ku Chin Soon	
Project Title: University Class Timetable Tool	

1. WORK DONE

Modify and finalize the report.

2. WORK TO BE DONE

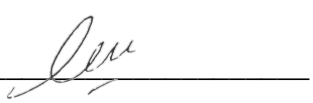
Submit the report and prepare for presentation.

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

The progress seems progressing on track and goes smoothly.



Supervisor's signature

Student's signature

POSTER

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

University Class Timetable Tool

Project Developer: Ang Wen Jie

Project Supervisor: Ts Dr Ku Chin Soon

Introduction

This University Class Timetable Tool aims to use the Genetic Algorithm to resolve the problem of timetable scheduling in UTAR with the newly added constraints related to the study modes during Covid-19 pandemic.

Objectives

- to investigate the hard and soft constraints for the university class timetabling problems
- to design a university class timetabling tool that will help the users to generate the timetable with hybrid class automatically using the genetic algorithm

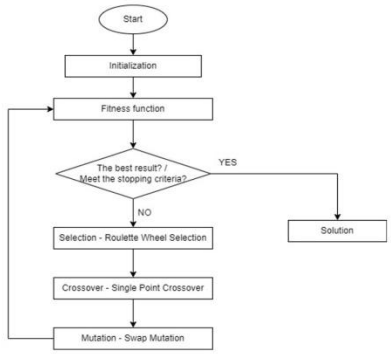
Methodology

- Collect the data required for the timetable generations in the university by getting the documents from the university's official website
- Study the current university timetable structure
- Define the new hard and soft constraints
- Design the chromosome structure

Discussion

There are many methods used in different phases in Genetic Algorithm, in this project the methods used in the phases include:
Selection phase - Roulette Wheel Selection
Crossover phase - Single Point Crossover
Mutation phase - Swap Mutation

Flow Chart of Genetic Algorithm Implemented



Result

A different set of timetables will be produced. Among them are:

- Classroom timetable
- Student timetable
- Lecturer timetable
- Course timetable

The timetables produced need to be verify that they follow the constraints set

Conclusion

With the new constraints considered and implemented, the timetable generated will be more adaptive to meet the current situations of the teaching and learning process using the Genetic Algorithm.

PLAGIARISM CHECK RESULT

Document Viewer

Turnitin Originality Report

Processed on: 06-Sep-2022 14:49 +08
ID: 1893445096
Word Count: 17619
Submitted: 2

1900369_FYP2 By Wen Jie ANG

Similarity Index	Similarity by Source	
2%	Internet Sources:	2%
	Publications:	1%
	Student Papers:	0%

exclude quoted | exclude bibliography | excluding matches < 8 words | mode: quickview (classic) report | Change mode | print | download

<1% match (publications)
Ali Aghagolzadeh, "Using a genetic algorithm optimizer tool to solve University timetable scheduling problem", 2007 9th International Symposium on Signal Processing and Its Applications, 02/2007

<1% match (Internet from 29-Jun-2021)
https://www.guru99.com/test-case.html

<1% match (Internet from 27-Jul-2021)
http://eprints.utar.edu.my

<1% match (Internet from 12-Jun-2022)
http://eprints.utar.edu.my

<1% match (Internet from 27-Jul-2021)
http://eprints.utar.edu.my

<1% match (Internet from 15-Apr-2019)
https://www.ijcit.com/archives/volume5/issue6/Paper050606.pdf

<1% match (publications)
R Raghavjee, N Pillay, "Using genetic algorithms to solve the South African school timetabling problem", 2010 Second World Congress on Nature and Biologically Inspired Computing (NaBIC), 2010

<1% match (Internet from 09-Jan-2022)
https://www.coursehero.com/file/102121630/ASSIGNMENT-2-FRONT-SHEET-14pdf/

<1% match (Internet from 24-Jul-2022)
https://www.coursehero.com/file/149858115/Lab-Activity-1docx/

<1% match (Internet from 06-Oct-2021)
https://dokumen.site/download/vhdl-for-engineers-kenneth-l-shortpdf-a5b39f0b435755

<1% match ()
Tan , Hean Giap, "Web based agent for promotion of mercantile - company management / Tan Hean Giap"

<1% match (student papers from 02-May-2010)
Submitted to Asian Institute of Technology on 2010-05-02

<1% match (Internet from 23-Feb-2021)
https://dokumen.pub/machine-learning-with-health-care-perspective-machine-learning-and-healthcare-1nbsped-3030408493-9783030408497.html

<1% match (publications)
Hai Shen, Yunlong Zhu, Li Jin, Wenping Zou, "Two-phase heuristic for Capacitated Vehicle Routing Problem", 2010 Second World Congress on Nature and Biologically Inspired Computing (NaBIC), 2010

<1% match (student papers from 17-Apr-2011)
Submitted to INTI International University on 2011-04-17

<1% match (publications)
Khare, Anula, and Saroj Rangnekar, "Particle swarm optimization: A review", Applied Soft Computing, 2012.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

PLAGIARISM CHECK RESULT

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Ang Wen Jie
ID Number(s)	19ACB00369
Programme / Course	Bachelor of Computer Science (Honours)
Title of Final Year Project	University Class Timetable Tool

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 2 % **Similarity by source** Internet Sources: 2 % Publications: 1 % Student Papers: 0 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required, and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Ku Chin Soon

Date: 07/09/2022

Signature of Co-Supervisor

Name: _____

Date: _____

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

FYP 2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB00369
Student Name	Ang Wen Jie
Supervisor Name	Ts Dr Ku Chin Soon

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
.	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

_____ _____

(Signature of Student)
Date: 06/09/2022