TIMETABLE SCHEDULING SYSTEM IMPLEMENTING GENETIC ALGORITHM

by O. RAMAREA L. RADIKGOMO O. SIMULA
N. MOSELE

Contents

List of Figures

List of Tables

Listings

Acknowledgements

In contribution to making this project a success, the group would like to thank the following for the help, guidance and effort they contributed:

- MR RESEGO TOM RABALONE (Project Supervisor): the supervisor contributed a lot acting as a guide towards the complesion of this project.
- MR kEITUMETSE GOBOTSAMANG (LECTURER): in addition, we would like to thank lecturer MR Keitumetse for availing his time to also assist the team to complete this project.
- Lastly, the group also appreciates each member for their coporation in making sure that the project and system are coordinated together through their team working efforts.
- Unrecognisably, the group would like to thank the school for availing resources needed all the time in labs, such as softwares and print media

To. . . . . . RESEGO TOM RABALONE (SUPERVISOR)
. . . . . . ADVANCED RESEARCH PROJECT (AR PROJECT)

Chapter 1

INTRODUCTION

Over the years, information technology has been a working success from the day it came into play. This rapid growth or evolve has brought so many learning outcomes into play, some into action and some into a review such as artificial intelligence, robotics and engineering [?]. These all together are made possible with the implementation of genetic algorithms, but not limited to them. Genetic Algorithms are mainly known to imitate the course of natural selection and can be utilized as a skill for unraveling difficult optimization complications which have very outsized search spaces [ABRAMSON and ABELA(1992)]. Besides Artificial Intelligence and robotics GAs are applied in different concepts such as gaming concepts and scheduling concepts [Bambrick(1997.)]. In the initial stages of the project, gaming concepts such as picture puzzle and sodoku (number game) were considered with the use of genetic algorithms. In exchange of ideas, timetable scheduling seemed more prominent than the rest of the other concepts. In addition, GAs as a concept is a very broad topic and hence in this study GA implementation will be limited and focus on timetable scheduling in tertiary institutes.

The main question could be why timetable scheduling for tertiary institutions? This is so because most tertiary institutions or even academia learning institutions that make use of a timetable schedule make it manually. Making of a timetable manually has its disadvantages such having claches in the timetable, which is the major threat in making a schedule manually. Clashes in a timetable result in ineffectiveness in class attendance and causes inconveniences. The use of genetic algorithms in this project should be able to curb such problems not to rise again. The desired outcome of the final timetable should be similar to the timetable representation figure 2.3.

In order to make this project a success, a series of events had to take place such as which programming language to use and other integrated softwares used in the timetable making such as SQL server 2012. In addition, the schedule proforma was made using a Gantt chart. The document is only limited to research made and implementations that were put in place to make a timetable schedule.

Chapter 2

NEGOTIATED LEARNING AGREEMENT

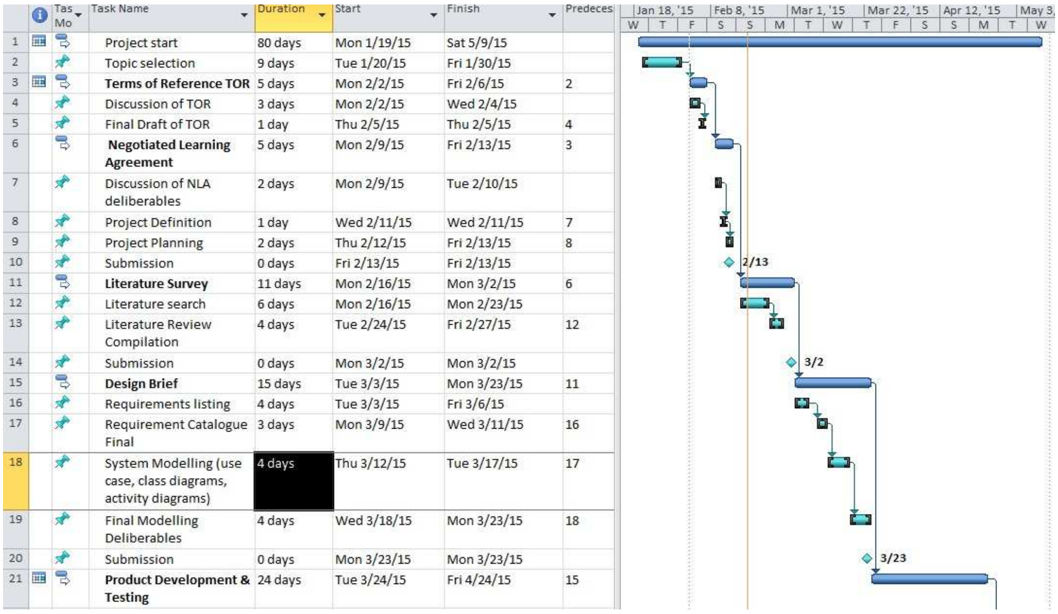
2.1 APPLIED RESEARCH PROJECT (AR Proj) BASIC TERMS OF REFERENCE

2.2 LEARNING OUTCOMES

The learning outcomes as defined indicate what the group is planning to have achieved at the end of the undertaken project, which is genetic algorithm implementation in a timetable scheduling system. The learning outcomes are derived from the outcomes and deliverables stated in the above terms of reference table.

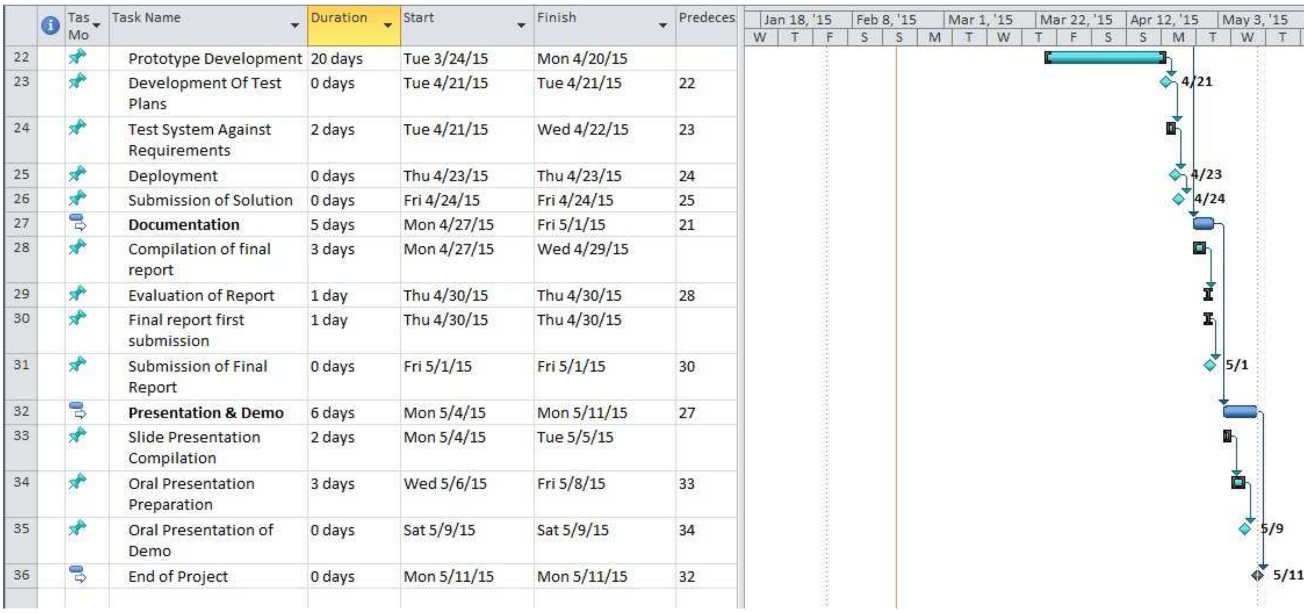
Having a functional system is not just the aim of this project but the group is aimed at learning about genetic algorithms, i.e. in general terms to their applications. It is highly empirical to know about the concept to be applied in order to model a better system and for future referencing about genetic algorithms. With the application of genetic algorithms, there are factors that may come into play as any other concept, so understanding the time factor of genetic algorithms when implemented in real time situations is also to be reviewed. Genetic algorithms are also said to work hand in hand with other algorithms, hence the concept of mutation comes into play. Mutation is a broad topic, so understanding it when applied on a very specific topic is also under the group learning scope. In any project undertaken, there are violations that may occur. Learning about these violations is also important and understanding these violations will also help make the project a success.

Name of Student1: LESEDI RADIKGOMO	ID Number1: CIS12-059
Name of Student2: NALEDI MOSELE	ID Number2: CIS12-050
Name of Student3: OMPHILE SIMULA	ID Number3: CIS12-077
Name of Student4: OTENG RAMAREA	ID Number4: CIS12-204
Supervisor: RESEGO RABALONE	Client/Sponsor: Tertiary Schools
Title	Timetable scheduling system
Context	The system is mainly targeted at tertiary schools as their client. Tertiary schools (universities) are reputable for offering high learning education with classes scheduled for students and lecturers. Nonetheless, university course timetabling is a widely spread and studied topic but the use of automated timetable scheduling system is not commonly known or even used among tertiary schools (commonly known as universities). It is of great value to automate the creation of the timetable in tertiary schools as it will result in the system being effective and efficiently precise.
Aims and Objectives	AIM • To produce an automated timetable scheduling system for tertiary schools that requires less input from the user through the implementation of genetic algorithms. OBJECTIVES • To find out how a timetable schedule system can be implemented using genetic algorithms. • To find out the traditional/current methods of timetabling system. • To find out how other individuals were able to tackle the issue of tertiary/college timetable scheduling using genetic algorithms. • To find out how genetic algorithms can be integrated with other algorithms and systems to enhance a systems functionalities. • To know and understand about the genetic algorithm (have clear knowledge and implementation of genetic algorithms in different projects). • To find out and understand about the time factor of genetic algorithms when applied in real time situations.
Skills Required	For the project undertaken to be successful, certain fundamental skills should be exhibited and put in place. Such required skills are mainly of great importance and include programming using visual basic language, research and analysis, team-work abilities, communication skills, genetic algorithm implementation and systems analysis and understanding. Knowl-



	Task Name	Duration	Start	Finish	Predeces
1	Project start	80 days	Mon 1/19/15	Sat 5/9/15	
2	Topic selection	9 days	Tue 1/20/15	Fri 1/30/15	
3	**Terms of Reference TOR**	5 days	Mon 2/2/15	Fri 2/6/15	2
4	Discussion of TOR	3 days	Mon 2/2/15	Wed 2/4/15	
5	Final Draft of TOR	1 day	Thu 2/5/15	Thu 2/5/15	4
6	**Negotiated Learning Agreement**	5 days	Mon 2/9/15	Fri 2/13/15	3
7	Discussion of NLA deliberables	2 days	Mon 2/9/15	Tue 2/10/15	
8	Project Definition	1 day	Wed 2/11/15	Wed 2/11/15	7
9	Project Planning	2 days	Thu 2/12/15	Fri 2/13/15	8
10	Submission	0 days	Fri 2/13/15	Fri 2/13/15	
11	**Literature Survey**	11 days	Mon 2/16/15	Mon 3/2/15	6
12	Literature search	6 days	Mon 2/16/15	Mon 2/23/15	
13	Literature Review Compilation	4 days	Tue 2/24/15	Fri 2/27/15	12
14	Submission	0 days	Mon 3/2/15	Mon 3/2/15	
15	**Design Brief**	15 days	Tue 3/3/15	Mon 3/23/15	11
16	Requirements listing	4 days	Tue 3/3/15	Fri 3/6/15	
17	Requirement Catalogue Final	3 days	Mon 3/9/15	Wed 3/11/15	16
18	System Modelling (use case, class diagrams, activity diagrams)	4 days	Thu 3/12/15	Tue 3/17/15	17
19	Final Modelling Deliberables	4 days	Wed 3/18/15	Mon 3/23/15	18
20	Submission	0 days	Mon 3/23/15	Mon 3/23/15	
21	**Product Development & Testing**	24 days	Tue 3/24/15	Fri 4/24/15	15

FIGURE 2.1: gantt chart pahse 1



	Task Name	Duration	Start	Finish	Predeces
22	Prototype Development	20 days	Tue 3/24/15	Mon 4/20/15	
23	Development Of Test Plans	0 days	Tue 4/21/15	Tue 4/21/15	22
24	Test System Against Requirements	2 days	Tue 4/21/15	Wed 4/22/15	23
25	Deployment	0 days	Thu 4/23/15	Thu 4/23/15	24
26	Submission of Solution	0 days	Fri 4/24/15	Fri 4/24/15	25
27	**Documentation**	5 days	Mon 4/27/15	Fri 5/1/15	21
28	Compilation of final report	3 days	Mon 4/27/15	Wed 4/29/15	
29	Evaluation of Report	1 day	Thu 4/30/15	Thu 4/30/15	28
30	Final report first submission	1 day	Thu 4/30/15	Thu 4/30/15	
31	Submission of Final Report	0 days	Fri 5/1/15	Fri 5/1/15	30
32	**Presentation & Demo**	6 days	Mon 5/4/15	Mon 5/11/15	27
33	Slide Presentation Compilation	2 days	Mon 5/4/15	Tue 5/5/15	
34	Oral Presentation Preparation	3 days	Wed 5/6/15	Fri 5/8/15	33
35	Oral Presentation of Demo	0 days	Sat 5/9/15	Sat 5/9/15	34
36	End of Project	0 days	Mon 5/11/15	Mon 5/11/15	32

FIGURE 2.2: gantt chart pahse 2

2.3 PROJECT SCHEDULE

The events of this project follow each other in a predefined order (as depicted by the Gantt chart) and hence the scheduling of those events are easier to time slot given a specific time limit. These events were scheduled using a Microsoft Project editor tool called a Gantt chart.

2.4 Project Overview

2.5 LITERATURE REVIEW

The focus of this report is mainly on tertiary institutes because almost all tertiary institutes face the problem of timetable scheduling [?]. In any case, the current method of manual timetable scheduling is considered very inadequate as it is time consuming and provides very high chances violations on the timetable. Timetable scheduling is described as the sharing out of resources for factors under predefined constrictions so that it maximizes the likelihood of allocation or reduces the violation of restrictions set [Shengxiang and Sadaf(2009)]. In the new approach, genetic algorithms seems to be a way forward to solving this problem. GAs are a dominant force in the overall purpose as optimization tools which model principles of evolution [Davis(2007)], in another aspect they can be said to be adaptive systems that inspired by the nature of evolution. [?], also mentioned that they are most often capable of finding globally optimal solutions even in the most complex of search spaces, thus in this case GAs are used to automate the scheduling of classes, and in contrast they are known to keep several distinct outcomes in the form of a population [?]. [?] Continues to say that, the distinct outcomes known as parents are selected from the total population and mated together to form a new offspring called a child. [?] Substantiated on this by mentioning that GAs work on a populated strategy and by combining together to form new optimal solutions. The new offspring generated is further mutated, adopting the biological concept, in order to bring about diversity into the total population [?] and [?].

For GAs to be implemented, they have to have a special representation for their chromosomes [Golub()]. Chromosomes are genetic materials said to be made up of genes (objects). [Golub()] Mentioned that there are various ways to represent chromosomes, which is very important, as they can be represented as array of numbers, array of bits (binary bit), hash maps, a number, a string of characters, a matrix and or other data structure representation chosen by a user.

Figure 2.3 displays the use of hash maps in chromosome representation.

Figure 2.4 displays an example of how a chromosome will be represented in binary for the mutation process to take precedence.

Figure 2.3 and 2.4 are just a few examples of choromosome representation as explained in the passage.

In order to proceed with chromosome representation for mutation to take place, chromosomes must be encoded and also satisfy restrictions called constraints and precision given (Jose and Philip, 1994). In the timetable scheduling problem, the genes that make the chromosome are mostly what is define to be essential to be able to slot the classes

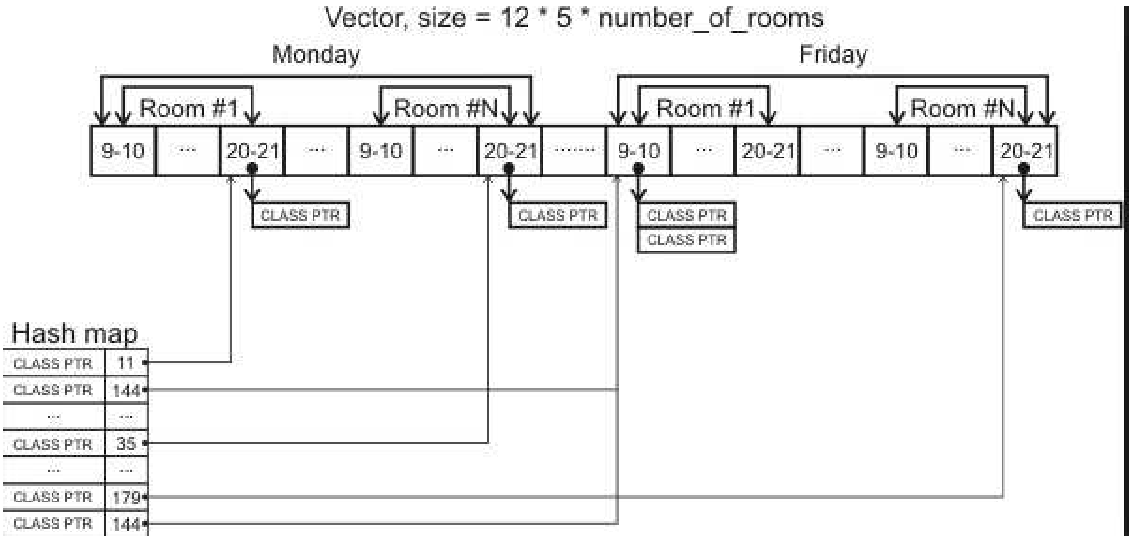


FIGURE 2.3: example of use of hash maps

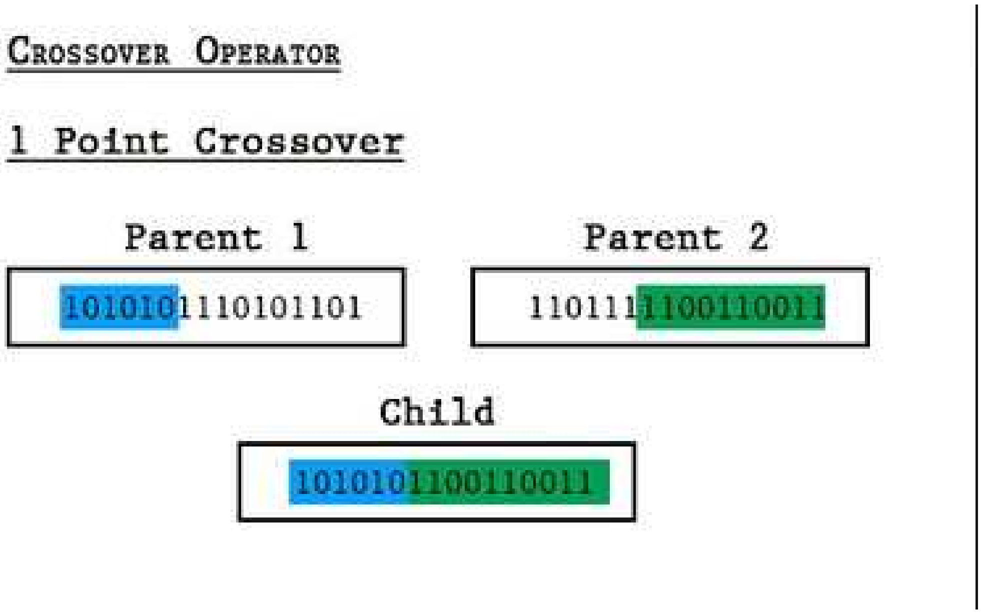


FIGURE 2.4: chromosome representation in binary in the mutation process

into timeslot that are made available, which are basically courses, classes, lecturers and resources available [?], [?] and [?].

One other aspect to be looked into when dealing with genetic algorithms after chromosome representation is making an initial population. An initial population in GA, basically means creating a significant sum of indiscriminate or random entities based on hard constraints set. The initial population is what evolution concept of genetic algorithm is based upon in order to take place, and so is are the processes involved in this such as fitness evaluation, mutation, crossover, selection and repairing. Population choice depends entirely on the users needs and a small population is not very ideal to create. A very small population can be said to get really small with the evolution process and may destroy the total population as invalid chromosomes may be attained. In contrast, a larger population will be ideal as it will give better results but it requires time and resources and hence it will be very slow. Population set as initial population in essence has to be checked or rather evaluated for mutation to take place. The evaluation of a population is done using a function known as the Fitness Function. [Larget(2012)] States that the fitness of a solution is the estimation of how good the solution is mostly using soft constraints set. As the number of constraints that the solution satisfies increases so does its quality thus ranking the solution at a more eligible spot to be selected [?]. The probability of the solution being selected is proportional to its fitness either greater or less than that of competitors. To strengthen the next generation, it is vital that selection of the current population's individuals takes place. The number of offspring produced for the next generation from the selection process is based on the selection pressure which will be determined by the fitness scores attained during evaluation [?].

After the selection process is carried out, an exchange or recombination of segments between chromosomes referred to as crossover [Lan Fang(1992)]. This is an important operation which involves the creation of a new chromosome dubbed the offspring from a pair of chromosomes from the current population which is built from a sequence of genes that portray the best characteristics from both selected chromosomes that follows a definable crossover probability [?]. The effect of crossover can be beneficial or detrimental thus to ensure the preservation and use of good chromosomes, not all chromosomes will be utilized during the crossover operation as a result of that probability [?].

In the last stages of the mutation process, a timetable should be generated and represented in a user friendly manner such as; figure 2.5.

In conclusion, the use of genetic algorithms in timetable implementation involves several processes to perfect the final timetable. Chromosome representation may be different but the implementation follows the same processes. At the end of all the processes, the chosen timetable's fitness should converge to give a value that equals one (1) or closer to closer one (1) but less than one (1).

From the gathered data about the use genetic algorithms in implementing a timetable schedule, the project being conducted implements one other unique way of chromosome representation described in the passage, being binary representation. Since data is captured into the system for different entities such the lecturer, module and facility, genes (unique data entries) are derived from the three entities to make up a chromosome (lecturer ID, module name, maximum number of students and room number). Implemented binary chromosome representation will be as follows below;

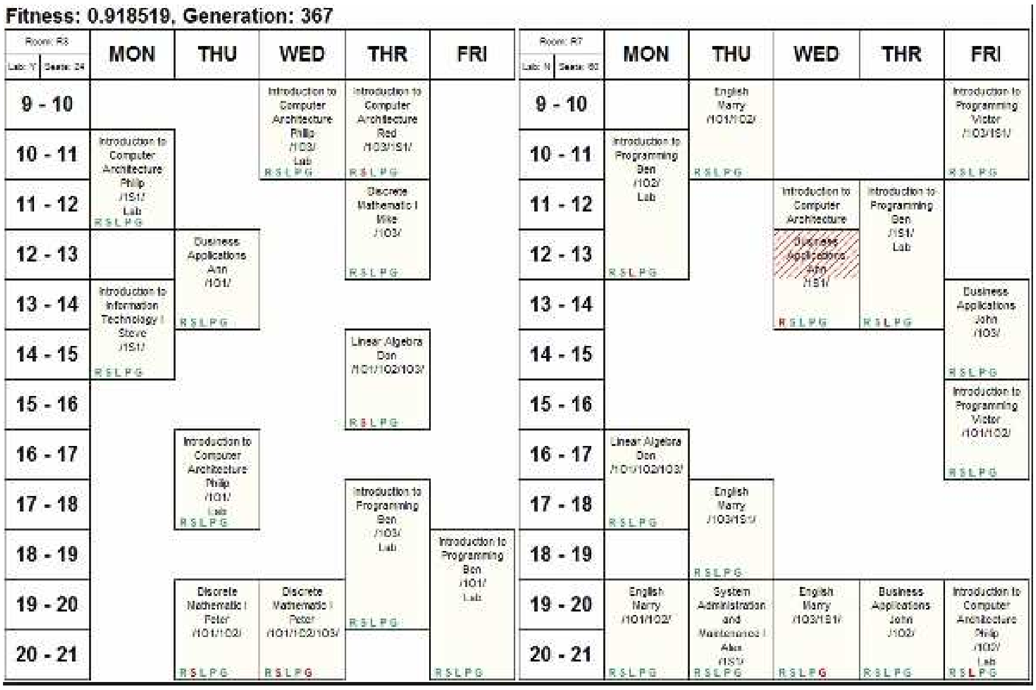


FIGURE 2.5: Final timetable

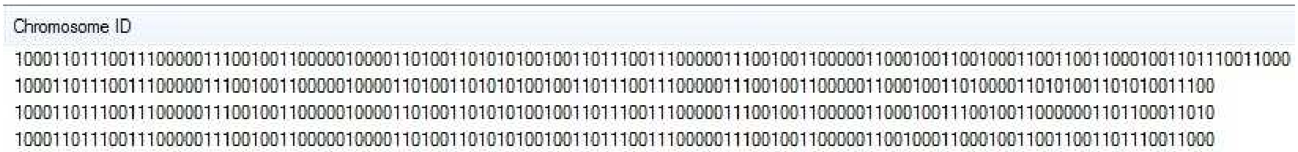


FIGURE 2.6: chromosome representation

Figure 2.6 shows an example of how chromosomes will be represented in binary. The data entries are stored in a database which in essence will also work as a controller. A controller class is said to an object that mediate or act as an intermediary between entities and boundaries [?]. In this case, a database acts as intermediary between the user and the system itself as a central point of holding data captured from input by the user. After chromosomes have been represented, an initial population has to be generated. In this unlike having an algorithm to create an initial population, the initial population will just generated randomly. From the initial population, the determination of chromosome fitness was derived from a function used by [?] who used hash maps to generate a timetable, the total score of candidate divided by the maximum score. The desired crossover method will be to generate a simple algorithm that will make mutation simple and possible in a small limited amount of time.

Chapter 3

DESIGN BRIEF

3.1 METHODOLOGY

3.2 FEATURE DRIVEN DEVELOPMENT

FDD is one of the agile software development methodologies that emerged over the years as an alternative to traditional waterfall development; it?s a very highly adaptive software development process. The use of FDD as a methodology may be in question, why FDD as a methodology? There are several reasons such as that it:

- Is highly and short iterative, it is mostly used when there is just limited amount of time to develop the required software.
- Emphases quality at all steps.
- Delivers frequent tangible working results at all steps.
- Provides accurate and meaningful progress and status information with the minimum of overhead and disruption for the developers.

FDD is meat to decompose the entire problem domain into tiny problems (the timetable scheduling) and these can be solved in a small period of time. FDD also splits the project into iterations so that the distance in the time between analysis and test is reduced. FDD mainly starts with the creation of a domain object model. In this case we define the problem that led to the need development of the automated timetable scheduling system. Defining the domain object involves even defining the modeling activities, then using the information from the modeling activity and from any other requirement activities that have taken place on the planning stages, features list was created. Each feature listed is termed as a process or phase. The development of each feature was allocated time limit as indicated in the Gantt Chart. In the last stages of the development, the features were developed concurrently as the chosen methodology allows.

3.3 REQUIREMENT ANALYSIS AND DESIGN

3.3.1 REQUIREMENTS LISTING

1. The system should be able to identify genes that make up a particular chromosome.
2. The system should be able to combine the genes that make up a particular chromosome.
3. A user should be able to capture data inputs that will necessitate the system to identify genes that will make up the chromosome.
4. The system should be able to generate the initial population of timeslots with varying timeslots.
5. The system should be able to use chromosomes generated to populate the timeslots.
6. The system should be able to determine the fitness of various candidates in the initial population.
7. The system should be able to carry out the selection process by identifying candidates with a value of 0.5 to 1.
8. The system should be able to determine candidates with the best value of 1 and identify it as the final timetable.
9. The clashes identified to converge to the value of 0 in the final timetable.
10. Allocated timeslots should show users the module name, lecturer name, room number and time.
11. The timetable should display free timeslots.

3.3.2 NON-FUNCTIONAL REQUIREMENT

- The system should be user friendly i.e. easy to use.
- Variability (fixing function).
- Non-volatile information captured should not be lost when advancing to the next stage or with every improvement made.
- Modularity (improvement of the initial population).

ht!

REQUIREMENT	The system should be able to identify genes that make up a particular chromosome
REQUIREMENT ID	RQ1
REQUIREMENT DESCRIPTION	From the captured data (Lecturer, facility and module details), there are specific fields that will be identified as genes (lecturer name, module name, room number and maximum number of students in a module), the system will retrieve this as genes.
NON-FUNCTIONAL REQUIREMENTS	1. The data that composes genes should be able to be traced where it will be retrieved, thus a clear path should be specified for this to take place. 2. Information should be readily available on the system. 3. The captured or available on the system should not be lost after it has used or captured.

TABLE 3.1: REQUIREMENT CATALOG 1

- Responsiveness - feedback to the user should be in real time.
- Traceability.

3.3.3 REQUIREMENT CATALOG

3.4 USE CASE DIAGRAM

Figure 3.1 depicts a use case diagram for the system being modeled. It illustrates how the users interact with the system. Briefly, it depicts use cases from the point the user inputs data which the system captures and uses to retrieve genes to the creation of the chromosomes. After making up of the chromosomes, it is demonstrated that an initial population is generated. In the use case diagram, it is also depicted that fitness of each candidate in the initial population is determined using a fitness function, which is already stated as: (total points of a candidate/maximum number of points). Other processes depicted in the use case diagram are the selection process and mutation. These processes necessitate for the creation of the final population.

REQUIREMENT	The system should be able to combine the genes that make up a particular chromosome.
REQUIREMENT ID	RQ2
REQUIREMENT DESCRIPTION	The identified genes will be combined together to form one entity called a chromosome. The chromosomes vary due to different entities with different captured data.
NON-FUNCTIONAL REQUIREMENTS	1. The identified path for gene retrieval should be maintained. 2. The genes identified should be part of data captured on the system. 3. The combining of genes should follow an algorithm defined to combine them.

Table 3.2: REQUIREMENT CATALOG 2

REQUIREMENT	A user should be able to input data that will necessitate the system to identify genes that will make up a chromosome.
REQUIREMENT ID	RQ3
REQUIREMENT DESCRIPTION	In order for a user to enter data to be captured, there are fields specified to that particular user to enter the required data in different form interfaces. For instance in the module form, there is module ID, module name and maximum number of students. All processes in the system depends on the data entry such as the fore-mentioned.
NON-FUNCTIONAL REQUIREMENTS	1. The system should be user friendly, all fields and instruction should not be technical but simplified in clear plain text.

Table 3.3: REQUIREMENT CATALOG 3

REQUIREMENT	The system should be able to generate the initial population of timeslots with varying timeslots.
REQUIREMENT ID	RQ4
REQUIREMENT DESCRIPTION	Timeslots are classes given time periods on the timetable. The initial population will be composed of various timetables with varying timeslots for each lecture session. The initial population necessitate for fitness evaluation and precision. The initial population will only focus on one constraint to give each lecture a time slot and its fitness being evaluated by: capacity of the room weighed against the maximum number of students a module can accommodate.
NON-FUNCTIONAL REQUIREMENTS	1. On constraints it should be stated that; the maximum number of students that a module can accommodate should not exceed the capacity a room can accommodate. 2. Timeslots depend on the natural time and days of the week as other factors. 3. The data used to populate time slots should not be lost after populating the allocated timeslots. 4. The data used to populate the timeslots should also be traceable in the system, the pre-defined path should be maintained.

Table 3.4: REQUIREMENT CATALOG 4

3.5 USE CASE SCENARIOS

IDENTIFIER	UC7
INITIATOR	USER
GOAL	To edit any information that was previously captured and stored by the system.
PRE-CONDITION	Data needs to be captured before it can be edited.
POST-CONDITION	After the selected information is edited the system then saves and updates the relevant records.
ASSUMPTIONS	The expected initiator for this use case is the user of the system.
MAIN SUCCESS SCENARIO	• the user searches for what they would like to edit. • After locating what they want, the user can then edit the records, for instance, by

REQUIREMENT	The system should be able to use chromosomes generated to populate the timeslots.
REQUIREMENT ID	RQ5
REQUIREMENT DESCRIPTION	Prior to initial population creation, timeslots will have to be assigned to chromosomes generated. Assigning of these timeslots to chromosomes only happens if the maximum number of students a module can accommodate does not exceed the capacity a room can accommodate.
NON-FUNCTIONAL REQUIREMENTS	1. Timeslots make use of the week day and time as a variable. 2. Pre-captured data on the system should not be lost but utilized whenever needed. 3. Chromosomes used should be different across the timeslots.

Table 3.5: REQUIREMENT CATALOG 5

REQUIREMENT	The system should be able to determine the fitness of various candidates in the initial population.
REQUIREMENT ID	RQ6
REQUIREMENT DESCRIPTION	As the initial population is comprised of varying timetables as candidates, a fitness function is set in order to calculate each candidates fitness. The fitness function is set as the points of a candidate divided by the maximum points a candidate should obtain.
NON-FUNCTIONAL REQUIREMENTS	1. The initial population has to compose of varying timetables as candidates 2. Data for the created timetables should not be lost after the initial population should not be lost after the initial population is created. 3. The data used to create the initial population should be able to be traced to the source. 4. The fitness function defined should be available or accessible for the determination of the fitness of candidates to take place.

Table 3.6: REQUIREMENT CATALOG 6

REQUIREMENT	The system should be able to carry out the selection process by identifying candidates with a fitness value of 0.5 to 1.
REQUIREMENT ID	RQ7
REQUIREMENT DESCRIPTION	After determining and assigning each candidate a fitness score, a selection of candidate with a fitness score of 0.5 to 1 will done for the next stage which is crossover. This will also necessitate the mutation process to take place.
NON-FUNCTIONAL REQUIREMENTS	1. The initial population should be available for the selection process to take place. 2. A pre-defined constraint for selection should be present (selection of fitness score of candidates of 0.5 to 1).

Table 3.7: REQUIREMENT CATALOG 7

REQUIREMENT	The system should be able to determine candidates with the best fitness value of 1 and identify it as the final timetable.
REQUIREMENT ID	RQ8
REQUIREMENT DESCRIPTION	In the selectin process, the process of crossover is also initiated. After the crossover is done, there should be a candidate with a fitness score of 1 or even closer, such candidate will have to be meeting all constraints to have such a value. The candidate with the specified fitness value of one or closer will be selected as the final timetable. Even though the candidate will selected as the final timetable, it will still be checked against all constraints through the fixing function.
NON-FUNCTIONAL REQUIREMENTS	1. Defining the choosing of the final timetable should follow constraints set as the fitness value is to be 1 or closer to 1. 2. The fitness scores of the selected candidates should be made available for the process of crossover to occur.

Table 3.8: REQUIREMENT CATALOG 8

REQUIREMENT	The clashes identified should converge to the value of 0 in the final timetable.
REQUIREMENT ID	RQ9
REQUIREMENT DESCRIPTION	After the fixing function has been performed, there should be no clashes identified on the timetable hence the value of clashes should be 0. This will mean that all the constraints have been met.
NON-FUNCTIONAL REQUIREMENTS	1. All constraints must be met.

Table 3.9: REQUIREMENT CATALOG 9

REQUIREMENT	Allocated timeslots should show users the module name, lecturer name, room number and time.
REQUIREMENT ID	RQ10
REQUIREMENT DESCRIPTION	This gives the timetable more detailed information as to where a certain module is been attended (the room number), the lecturer who is taking that module and the time the lecture is schedule for.
NON-FUNCTIONAL REQUIREMENTS	1. Each day of the week should be represented on the timetable having a schedule for lecturers, modules and rooms. 2. The data selected for the final timetable should not be lost but kept as a record or final output for users to view or use. 3. The data selected for the final timetable should be made user friendly, that is, it should be easy to read and understand to the users.

Table 3.10: REQUIREMENT CATALOG 10

REQUIREMENT	The timetable should display free timeslots.
REQUIREMENT ID	RQ11
REQUIREMENT DESCRIPTION	After the slotting of all modules across week days, there should be free timeslots left on the timetable. This should be identified by not having an entry on that a particular time. Free timeslots should be able to help timetable users to identified unoccupied rooms.
NON-FUNCTIONAL REQUIREMENTS	1. No entries of lecture session at certain time periods on the timetable should be identified.

Table 3.11: REQUIREMENT CATALOG 11

TABLE MANAGEMENT SYSTEM1.jpg

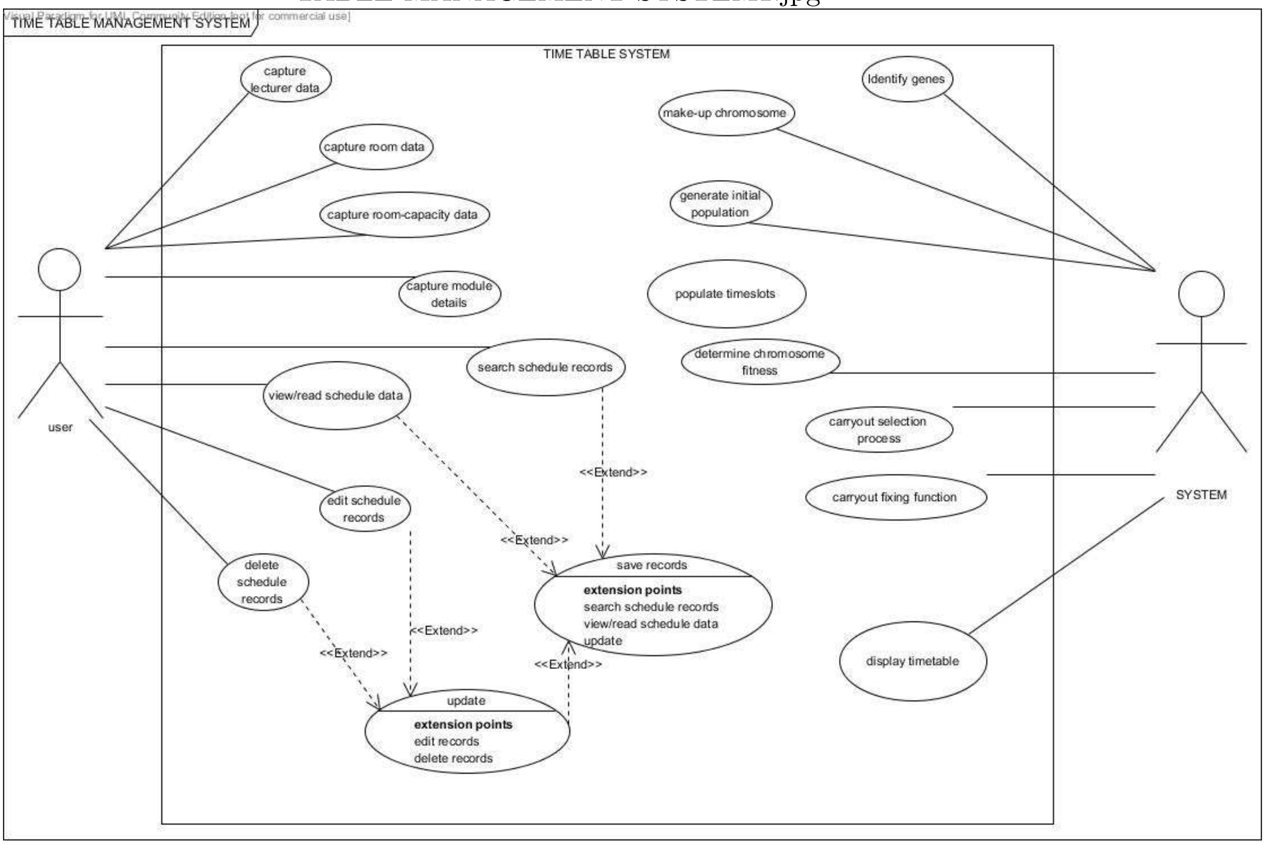


Figure 3.1: Use case diagram

IDENTIFIER	UC1
INITIATOR	USER
GOAL	To store data about the lecturer in a repository held in a server.
PRE-CONDITION	None
POST-CONDITION	After such data has been captured, it will be retrieved as a gene that can subsequently be used to constitute a chromosome. After data is stored, it can be reviewed, edited, deleted, updated. Reports can be printed out(which in this case will be in the form of the final timetable).
ASSUMPTIONS	The expected initiator in this case is the any authorized user of the system (the user inputs various types of records).
MAIN SUCCESS SCENARIO	1. The user selects the section in which relevant data is to be input (the system will in this case capture data about the lecturer). 2. Capturing this data allows the update of any information that the system has stored.

Table 3.12: Use case scenario 1

IDENTIFIER	UC2
INITIATOR	USER
GOAL	To store data about the room/facility in a repository held in a server.
PRE-CONDITION	None
POST-CONDITION	After such data has been captured, it will be retrieved as a gene that can subsequently be used to constitute a chromosome. After data is stored, it can be reviewed, edited, deleted, updated. Reports can be printed out(which in this case will be in the form of the final timetable).
ASSUMPTIONS	The expected initiator in this case is the any authorized user of the system (the user inputs various types of records).
MAIN SUCCESS SCENARIO	1. The user will select the section in which relevant data is to be input.(the system will in this case capture data about the room/facility). 2. Capturing this data allows the update of any information that the system has captured.

TABLE 3.13: Use case scenario 2

IDENTIFIER	UC18
INITIATOR	System
GOAL	To display the final timetable after, the fitness evaluation, selection process, and the fixing function has been carried out on those candidates that were selected, and the clash value converges to 0, and the fitness is close or equal to the value of 1.
PRE-CONDITION	The fixing function has to have performed, where the clash value converges to 0, and the fitness is close or equal to the value of 1.
POST-CONDITION	none
ASSUMPTIONS	The expected initiator for this use case is the system itself.
MAIN SUCCESS SCENARIO	1. The candidates with a fitness value of 0.5 or more have to be first selected. 2. The fixing function can now be carried out on those candidates that were selected, and the clash value converges to 0, and the fitness is close or equal to the value of 1. 3. After this, the final timetable can now be

IDENTIFIER	UC3
INITIATOR	USER
GOAL	To store data about the room/facility in a repository held in a server.
PRE-CONDITION	None
POST-CONDITION	After such data has been captured, it will be retrieved as a gene that can subsequently be used to constitute a chromosome. After data is stored, it can be reviewed, edited, deleted, updated. Reports can be printed out(which in this case will be in the form of the final timetable).
ASSUMPTIONS	The expected initiator in this case is the any authorized user of the system (the user inputs various types of records).
MAIN SUCCESS SCENARIO	1. The user selects the section in which they want to input relevant data (the system will in this case capture data about the room/facility CAPACITY). 2. Capturing this data allows the update of information that the system has stored.

TABLE 3.14: Use case scenario 3

IDENTIFIER	UC4
INITIATOR	USER
GOAL	To store data about the module data in a repository held in a server.
PRE-CONDITION	None
POST-CONDITION	After such data has been captured, it will be retrieved as a gene that can subsequently be used to constitute a chromosome. After data is stored, it can be reviewed, edited, deleted, updated. Reports can be printed out(which in this case will be in the form of the final timetable).
ASSUMPTIONS	The expected initiator in this case is the any authorized user of the system (the user inputs various types of records).
MAIN SUCCESS SCENARIO	1. The user selects the section in they want to input data (the system will in this case capture data about the module). 2. Capturing this data allows the update of information that the system has stored/captured.

TABLE 3.15: Use case scenario 4

IDENTIFIER	UC5
INITIATOR	USER
GOAL	To search for already existing information pertaining schedule records.
PRE-CONDITION	For this use case to be successfully executed, there has to be information that already exists, or that has been captured
POST-CONDITION	After the user has searched their information of choice, the records can be viewed, edited, deleted, added and/or updated.
ASSUMPTIONS	The expected initiator for this use case is the user.
MAIN SUCCESS SCENARIO	1. The user accesses saved information, and searches anything that they desire.

TABLE 3.16: Use case scenario 5

h!

IDENTIFIER	UC6
INITIATOR	USER
GOAL	To view or read any information that was previously captured and stored by the system.
PRE-CONDITION	Before anything can be viewed, a user has to have provided data that will be captured by the system.
POST-CONDITION	After the user has viewed their information of choice, the records can be edited, deleted, added and/or updated.
ASSUMPTIONS	The expected initiator for this use case is the user.
MAIN SUCCESS SCENARIO	1. the user searches for what they would like to view. 2. After locating what they desire to view, they can then view or read any information of their choice.

TABLE 3.17: Use case scenario 6

h!

IDENTIFIER	UC8
INITIATOR	USER
GOAL	To delete any information that was previously captured and stored by the system.
PRE-CONDITION	Data need to be captured before they can be deleted.
POST-CONDITION	After the selected information is deleted the system then saves and updates the relevant records.
ASSUMPTIONS	The expected initiator for this use case is the user of the system.
MAIN SUCCESS SCENARIO	• the user searches for what they would like to delete. • After locating what they want, the user can then deleted the records of their choice.

Table 3.18: Use case scenario 8

h!

IDENTIFIER	UC9
INITIATOR	USER
GOAL	To save any information that was previously captured, edited, stored or deleted within the system.
PRE-CONDITION	Data need to be captured, edited, updated or deleted before they can be saved.
POST-CONDITION	After the information is saves the system then updates the relevant records.
ASSUMPTIONS	The expected initiator for this use case is the user of the system.
MAIN SUCCESS SCENARIO	• The user searches for what they would like to edit, deletes or captures data the system will use. • After editing, deleting or capturing this data, the system then saves this data.

Table 3.19: Use case scenario 9

h!

IDENTIFIER	UC10
INITIATOR	System
GOAL	To update any information that was previously, edited, deleted by the user, or captured and stored by the system.
PRE-CONDITION	Data needs to be captured, deleted or edited before they can be updated by the system.
POST-CONDITION	After the selected information is edited the system then saves and updates the relevant records.
ASSUMPTIONS	The expected initiator for this use case is the user of the system.
MAIN SUCCESS SCENARIO	• the user searches for what they would like to edit. • After locating what they want, the user can then edit, delete or capture new data. • The user will then save this, and the system will automatically update.

Table 3.20: Use case scenario 10

h!

IDENTIFIER	UC11
INITIATOR	System
GOAL	To make up a chromosome using the genes that have been identified by the system.
PRE-CONDITION	Genes should be identified before this use case can be executed.
POST-CONDITION	After the genes have been identified and the chromosomes made-up, timeslots can now be populated.
ASSUMPTIONS	The expected initiator for this use case is the system itself.
MAIN SUCCESS SCENARIO	• The user inputs information that will be captured by the system. • Through the information that will be stored, the system will then identify those particular genes that will constitute a chromosome.

Table 3.21: Use case scenario 11

h!

IDENTIFIER	UC12
INITIATOR	System
GOAL	To detect or recognize those particular genes that will make up a chromosome.
PRE-CONDITION	Data needs to be captured before genes can be identified.
POST-CONDITION	After the genes are identified, they can be used to make up particular chromosomes.
ASSUMPTIONS	The expected initiator for this use case is the system itself.
MAIN SUCCESS SCENARIO	• The user inputs information that will be captured by the system. • Through the information that will be stored, the system will then identify those particular genes that will constitute a chromosome. • The chromosome will then be formed, using the identified genes.

TABLE 3.22: Use case scenario 12

h!

IDENTIFIER	UC13
INITIATOR	System
GOAL	To populate timeslots on potential timetables.
PRE-CONDITION	The chromosomes should be made up.
POST-CONDITION	After the timeslots have been populated, an initial population is then generated.
ASSUMPTIONS	The expected initiator for this use case is the system itself.
MAIN SUCCESS SCENARIO	• Chromosomes should have been constituted using the relevant genes. • Timeslots should be populated. • An initial population is now ready to be generated.

TABLE 3.23: Use case scenario 13

h!

IDENTIFIER	UC14
INITIATOR	System
GOAL	To generate an initial population from the chromosomes that were populated into the timeslots. This population will form a group of timetables.
PRE-CONDITION	A set of chromosome should have been constituted, and timeslots populated using these chromomes, in order for this use case to be executed successfully.
POST-CONDITION	After the genes have been identified the chromosomes made-up and populated the timeslots, an initial population can then be generated. After this, the fitness of each candidate will be evaluated.
ASSUMPTIONS	The expected initiator for this use case is the system itself.
MAIN SUCCESS SCENARIO	• The user inputs information that will be captured by the system. • Through the information that will be stored, the system will then identify those particular genes that will constitute a chromosome. • The chromosome will then be formed, using the identified genes, and these chromosomes will later be used to populate these timeslots. • An initial population will be generated from a set of those chromosomes that were made up.

Table 3.24: Use case scenario 14

3.6 ENTITY RELATION DIAGRAM

Figure 3.2 depicts an entity relation diagram (ERD). ERD typically shows all classes that are relevant and used in the modeling of the system.

h!

IDENTIFIER	UC15
INITIATOR	System
GOAL	To determine the fitness the candidates that were populated into the timeslots.
PRE-CONDITION	An initial population should be readily generated.
POST-CONDITION	After the fitness of these candidates has been evaluated, the selection process is then carried out.
ASSUMPTIONS	The expected initiator for this use case is the system itself.
MAIN SUCCESS SCENARIO	• After an initial population has been generated, the fitness of the candidates that were populated is now ready for evaluation. • The fitness of these candidates will be evaluated by the function: candidatepoints/-maximumpoints. • Only candidates with a fitness value of 0.5 or more will qualify. • The fitness level will then assist in carrying out the selection process.

TABLE 3.25: Use case scenario 15

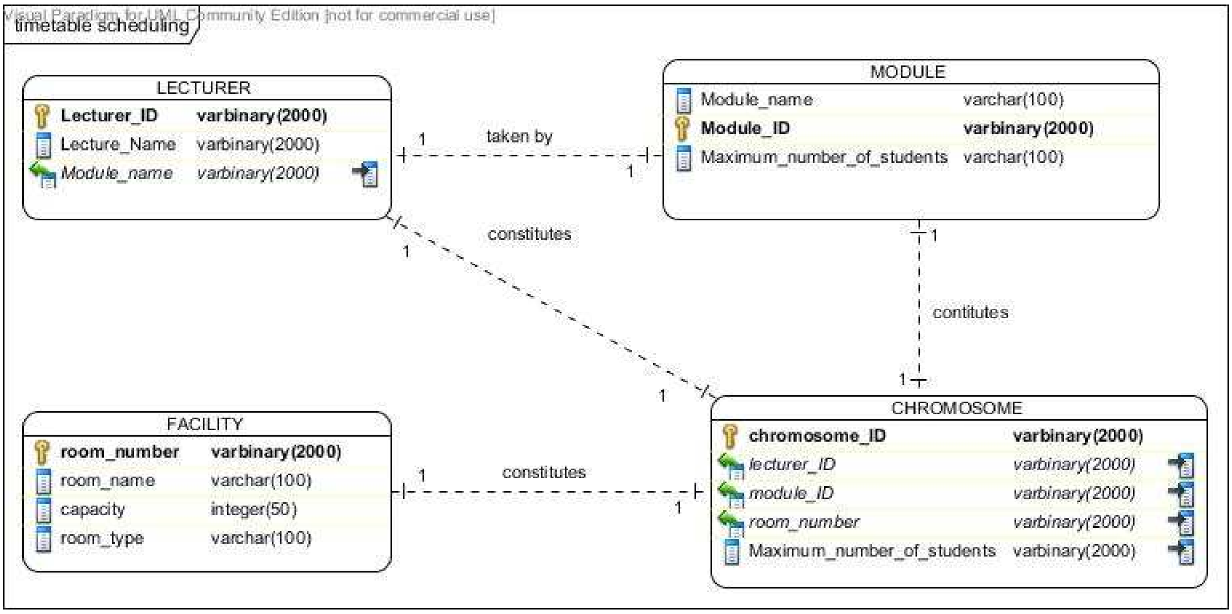


FIGURE 3.2: Entity Relation Diagram

h!

IDENTIFIER	UC16
INITIATOR	System
GOAL	To carry out the selection process of the candidates previously evaluated for their fitness.
PRE-CONDITION	In order for this selection to take place, the fitness evaluation of these candidates needs to have been carried out.
POST-CONDITION	After the selection process has been carried out, the fixing function is now ready to be carried out.
ASSUMPTIONS	The expected initiator for this use case is the system itself.
MAIN SUCCESS SCENARIO	• The selection will be carried out on candidates whose fitness has previously been carried out. • Candidates with a fitness value of 0.5 or more are the only ones who will qualify • In other words, those who will qualify will then be selected.

TABLE 3.26: Use case scenario 16

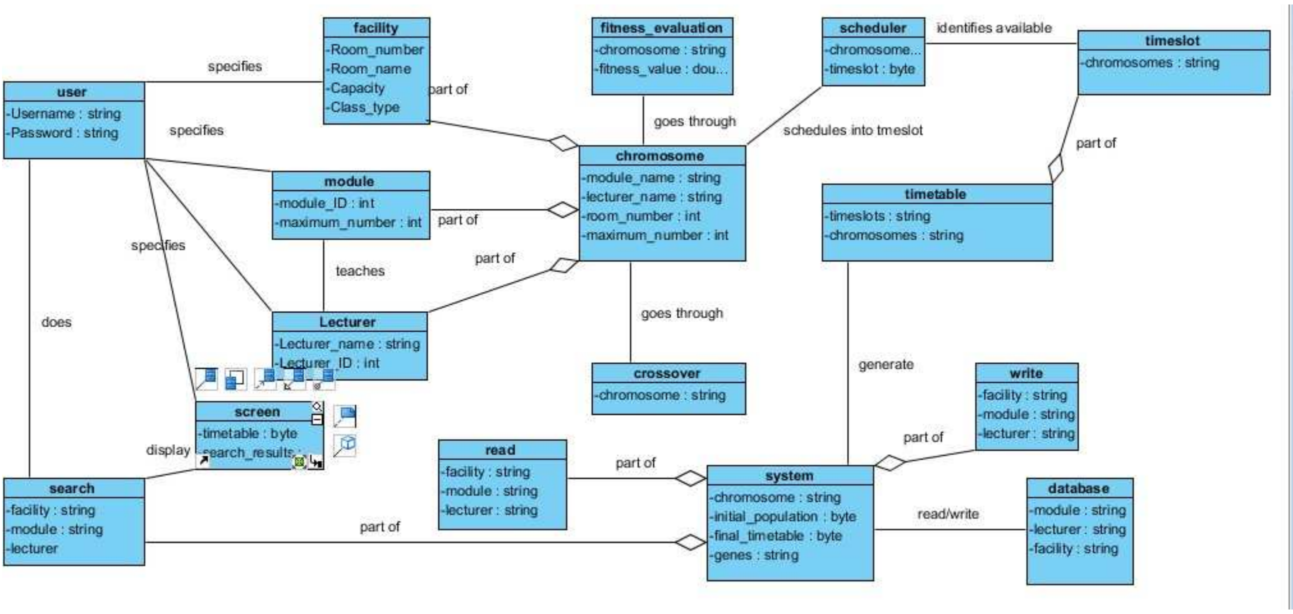


FIGURE 3.3: Activity diagram

IDENTIFIER	UC17
INITIATOR	System
GOAL	To carry out the fixing function on those candidates that were selected, and the clash value converges to 0, and the fitness is close or equal to the value of 1.
PRE-CONDITION	The selection of candidates whose fitness value is 0.5 or more has to have taken place.
POST-CONDITION	After the fixing function has been carried out on these selected candidates, the timetable can now be displayed.
ASSUMPTIONS	The expected initiator for this use case is the system itself.
MAIN SUCCESS SCENARIO	• The candidates with a fitness value of 0.5 or more have to be first selected. • The fixing function can now be carried out on those candidates that were selected, and the clash value converges to 0, and the fitness is close or equal to the value of 1. • After this, the final timetable can now be displayed.

Table 3.27: Use case scenario 17

3.7 ACTIVITY DIAGRAM

An activity diagram expands and elaborates more on the use case diagram illustrating the flow of events as shown in figure 3.3. The flow of events in the activity diagram (figure 3.3) flow from the back end of the design of the system, after all necessary data has been captured into the system. It is shown that captured data is saved to the database and at the same time genes that make up a chromosome are converted to binary. The flow of events flow one after the other to the display of the final timetable.

3.8 CLASS DIAGRAM

3.9 CRC CARDS

Figure 3.5 shows an actor CRC card represented with its attributes and responsibilities.

Figure 3.6 shows a system CRC card with its attributes and responsibilities, responsibilities show an association between classes.

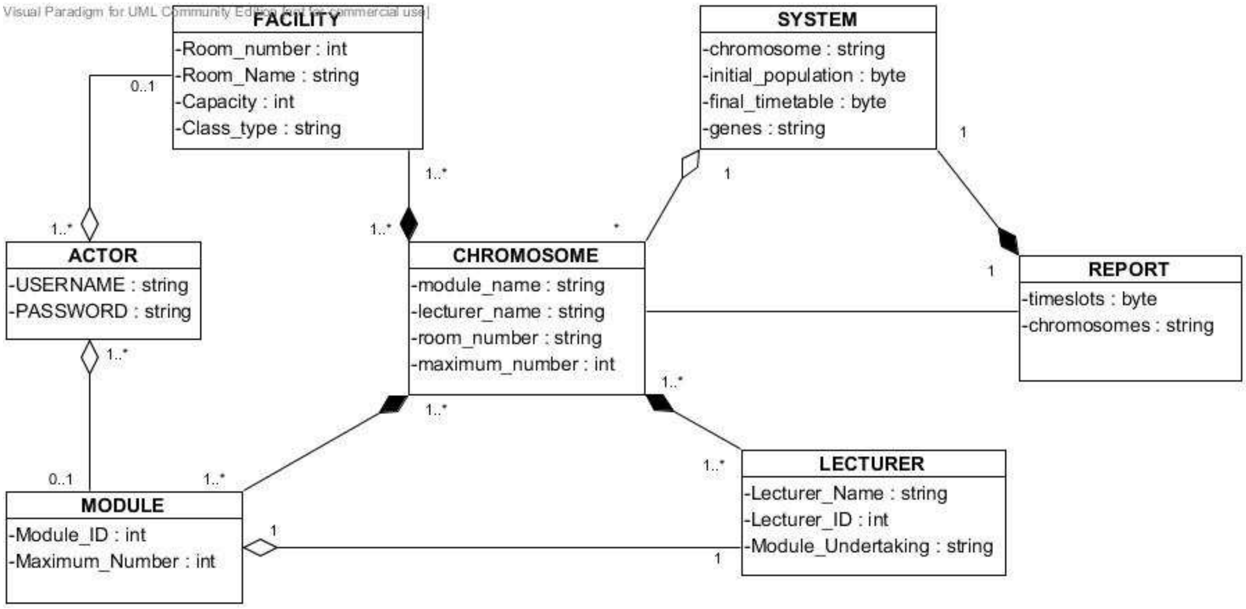


Figure 3.4: Class diagram

Actor

Description:

Attributes:

Name	Description
username	the username that a user inputs to log into the system
password	the password used for authentication to the system

Responsibilities:

Name	Collaborator
input data into the system	module
	facility
	lecturer

Figure 3.5: ACTOR CRC CARD

System	
Description:	
Attributes:	
Name	Description
chromosome	this is a combination of genes that make up one unit.
initial population	this is a list or group of timetables with varying timeslots.
final timetable	this is the outcome of the whole process to be displayed at the end of the genetic algorithm.
genes	this are different elements from different entities to make up a chromosome
Responsibilities:	
Name	Collaborator
generate an initial population	chromosome
diaplay the final timetable	report
capture and store the entered data	module
	facility
	lecturer

Figure 3.6: SYSTEM CRC CARD

Chromosome	
Description: This is a combination of genes (attributes from other classes) to make up one unit.	
Attributes:	
Name	Description
Lecturer Name	This is name of a lecturer that undertakes the specified module
Module Name	The name of the module taken by a lecturer and to be sloted to different timeslots
Room Number	The room number that the module will be taken in
maximum number	identifies the maximum number of students that are enrolled for a module
Responsibilities:	
Name	Collaborator
make up chromosome	module
	facility
	lecturer
populate timeslots	system

Figure 3.7: CHROMOSOME CRC CARD

Facility

Description:

Attributes:

Name	Description
room number	this identifies the number that is given to a particular room
room name	this states the name that has been given to a particular room according to the room number
capacity	this states the maximum number that the room can accomodate
class type	identifies the type of room whether it is a classroom or a laboratory

Responsibilities:

Name	Collaborator
make up a chromosome	chromosome

FIGURE 3.8: FACILITY CRC CARD

Figure 3.7 a chromosome CRC card represented with its attributes and responsibilities represented to make the chromosome class.

Figure 3.8 depicts facility CRC card that represent the facility class and its attributes, including its affiliated classes.

Figure 3.9 depicts lecturer CRC card that represent the lecturer class and its attributes, including its affiliated classes (responsibilities).

Figure 3.10 depicts module CRC card that represent the module class and its attributes, including its affiliated classes (responsibilities).

Figure 3.11 depicts report CRC card that represent the report class and its attributes, including its affiliated classes (responsibilities).

Figure 3.12 depicts database CRC card that represent the database class and its attributes, including its affiliated classes (responsibilities).

Figure 3.13 depicts read CRC card that represent the read class and its attributes, including its affiliated classes (responsibilities)

Figure 3.14 depicts search CRC card that represent the search class and its attributes, including its affiliated classes (responsibilities)

Lecturer	
Description:	
Attributes:	
Name	Description
Lecturer name	The name of the lecturer registered to take a module
Lecturer ID	This constitudes of numeric values that serve as a unique identification for a registered lecturer (short-hand representation of the lecturer name).
Module Undertaking	This states the name of the module that a particular lecturer is registered to take.
Responsibilities:	
Name	Collaborator
Indicate the module that a registered lecturer is undertaking	module
make up a chromosome	chromosome

FIGURE 3.9: LECTURER CRC CARD

Module	
Description:	
Attributes:	
Name	Description
Module Name	The name of the module undertaken or registered
Module ID	This constitudes of numeric values that serve as a unique identification for a registered module (short-hand representation of the module).
Maximum Number	This specifies the maximum number of students that can be registered for that module.
Responsibilities:	
Name	Collaborator
Identifies the lecturer that undertakes the module	Lecturer
make up a chromosome	chromosome

FIGURE 3.10: MODULE CRC CARD

Report	
Description:	
Attributes:	
Name	Description
timeslots	
Responsibilities:	
Name	Collaborator
display the final timetable	system
populate timeslots	chromosome

FIGURE 3.11: REPORT CRC CARD

Database	
Description: This is a storage unit for all the information needed to produce the	
Attributes:	
Name	
Facility details	this is the capacity
Module details	this is the informatio
Lecturer details	this is thelecturer na
Responsibilities:	
Name	
Store provided data(genes)	user, module, facilit

FIGURE 3.12: Database CRC CARD

Read

Description:

Attributes:

Name	Description
database	stores the provided genes in the system

Responsibilities:

Name	Collaborator
Read stored data from databasse	system

Double click: edit Attribute Description; Right click: add Attribute; remove Attr...

FIGURE 3.13: read CRC CARD

Search

Description: this is a process of finding certain information from a specified location

Attributes:

Name	Description
database	a storage unit for all details concerning timetable scheduling

Responsibilities:

Name	Collaborator
Produce search results	user, system

FIGURE 3.14: search CRC CARD

Crossover	
Description: This is where chromosomes are mated in order to produce a new chromosome(offspring) in the next generation	
Attributes:	
Name	Description
Chromosome	it is a combination of certain genes in oder to make a chromosome
Responsibilities:	
Name	Collaborator
produce offspring	chromosomes

FIGURE 3.15: crossover CRC CARD

Timeslot	
Description: The allocated time segment for a particular chromosome in a timetable	
Attributes:	
Name	Description
Chromosome	a chromosome presented after selection
Day	this is any day of the week from monday to friday
Time	this is the time of day from 8am to 4pm
Responsibilities:	
Name	Collaborator
specify timeslot	timetable

FIGURE 3.16: timeslot CRC CARD

Figure 3.15 depicts crossover CRC card that represent the crossover class and its attributes, including its affiliated classes (responsibilities)

Figure 3.16 depicts timeslot CRC card that represent the timslot class and its attributes, including its affiliated classes (responsibilities)

Figure 3.17 depicts fitness evaluation CRC card that represent the fitness evaluation class and its attributes, including its affiliated classes (responsibilities)

Figure 3.18 depicts screen CRC card that represent the screen class and its attributes, including its affiliated classes (responsibilities).

Figure 3.19 depicts scheduler CRC card that represent the scheduler class and its attributes, including its affiliated classes (responsibilities).

Fitness Evaluation	
Description: This measures the chromosomes ability to meet certain constraints	
Attributes:	
Name	Description
chromosomes	this is a combination of the provided genes that will undergo the
Fitnes value	the value/score given to chromosomes for meating constraints p
Responsibilities:	
Name	Collaborator
checks if a chromosomes satisfies constraints/conditions for making up a timetable	chromosome
assigns a fitness value based on the fitness evaluation carried out	chromosome

Figure 3.17: Fitness Evaluation CRC CARD

Screen	
Description: This gives the neded output	
Attributes:	
Name	Descript
final timetable	this is the final timetable without clashes
search results	this is the output provided after querying
Responsibilities:	
Name	Collabora
Interact with user by providing output	user

Figure 3.18: screen CRC CARD

Scheduler	
Description: This schedules the initial population to certain timeslots deemed fit	
Attributes:	
Name	Description
module(number of apperrance in the timetable)	this is the number of appearances of the module in the timetable
facility(available rooms)	thi is the number of available room to slot more classes on
timeslots(available time)	this is the time available to slot a class on
lecturer(number of appearance in the timetable)	this is the number of appearances of the lecturer on the timetable
Responsibilities:	
Name	Collaborator
Build up a timetable	timeslots

Figure 3.19: scheduler CRC CARD

Write	
Description: this is a processes of writting to a certain location	
Attributes:	
Name	Description
facility details	this includes the room sizes and number of room available
module details	this includes the module name and ID
lecturer details	these details includes the lecturer name and ID
Responsibilities:	
Name	Collaborator
update the database	system
adding new record	system

FIGURE 3.20: write CRC CARD

Figure 3.20 depicts write CRC card that represent the write class and its attributes, including its affiliated classes (responsibilities).

Chapter 4

IMPLEMENTATION

4.1 SYSTEM LOGIC

The application is launched. After it has been launched there two choices presented to the user, for the administrator and just an ordinary timetable user. When a user chooses the administrator option, they will be required to provide administration credentials. This was done as a security measure to guard against users editing or even entering wrong set of information not relevant to the class scheduling (such as recording a non-existent module, lecturer or facility or even deleting an existing module, lecturer or even facility). Such information may lead to misleading other timetable users.

Administrator is privileged to carry the most important task at the application start up, that is entering the required information to necessitate for mutation and selection processes to take place to the creation of the final timetable. There are other additional functions an administrator is privileged to carry out such as to delete and edit information needed for all processes. Login details for the administrator are kept in a log file. If the details are invalid, access will be denied and if they are valid that is when the user will be able to gain access to the system. Once the administrator has accessed the system they are able to add the lecturer, module and facility details that will be captured by the system and stored in the database. The system will then use the captured details to retrieve genes from three of the created tables in the database to a be viewed and chromosomes will be simultaneously created in the process and displayed in a table. Prior to saving data to the database, the data entities that make up a chromosome are converted to binary and stored in a binary format.

The chromosomes generated and displayed in a table will then be used to populate various timeslots in different timetables that compose the initial population. From the initial population, each candidate will be tested for fitness using a fitness function (total points of a candidate/maximum points). From the fitness determination of candidates follows the selection process that necessitate for the mutation process to the final timetable.

The other user advocated in this system is a just an ordinary timetable user who does not any other privilege to make changes to the system or records used in the whole system.

4.2 CODE LISTING

```
Public Sub save_details()
        If txt_lecturerID.Text <> "" And txt_lecturerNAME.Text <> "" And cmb_module.Text <> "" Then

            Dim val As String = Nothing
            Dim val1 As String = Nothing
            Dim result As New System.Text.StringBuilder
            For Each character As Byte In System.Text.ASCIIEncoding.ASCII.GetBytes(txt_lecturerID.Text)
                result.Append(System.Convert.ToString(character, 2).PadLeft(8, "0"))
                result.Append("")

            Next
            val = result.ToString.Substring(0, result.ToString.Length - 1)
            For Each character As Byte In System.Text.ASCIIEncoding.ASCII.GetBytes(cmb_module.Text)
                result.Append(System.Convert.ToString(character, 2).PadLeft(8, "0"))
                result.Append("")

            Next

            val1 = result.ToString.Substring(0, result.ToString.Length - 1)
            Try
                Dim insrtStr As String = "INSERT INTO lecturer (lecturer_ID, lecturer_name, module_unde
             & val & "','" & txt_lecturerNAME.Text & "','" & val1 & "')"

                obj_conc.Open()
                obj_com = New SqlCommand(insrtStr, obj_conc)

                obj_com.ExecuteNonQuery()

                obj_conc.Close()

                MsgBox("records successfully recorded")

            Catch ex As Exception
                Throw ex

            End Try
        Else
            MsgBox("there is nothing to convert to binary and save")
        End If
    End Sub
```

LISTING 4.1: LECTURER CLASS

The above method is used to covert the lecturer ID to binary and add lecturer details to the database. Variables were declared to be used in the method after adding the controls such as textboxes and combo-boxes in the design view. In addition, message boxes also show for any errors that may have occurred or success in execution of events.

This method is universal across all classes that save data to the database, being module and facility.

```
Private Sub lecturer_register_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        If obj_conc.State = ConnectionState.Closed Then
            obj_conc.ConnectionString = ("Data Source=bss\sqlexpress;Initial Catalog=timetable;I
            'obj_conc.Open()
        Else
            MsgBox("System Ready", MsgBoxStyle.Information)
        End If
```

Listing 4.2: DATABASE CONNECTION

The above method is used to connect to a database, this in essence acts as controller class as it acts as an intermediary between the boundary and entity classes.

```
   Public Sub load_chromosomes()



        'Dim chromo(3) As String

        With obj_com

            .CommandText = "Select room_no, lecturer_ID, module_undertaking, maximum From facili
            .Connection = obj_conc
        End With
        With obj_adapter
            .SelectCommand = obj_com
            .Fill(tbl_chromo)

        End With

        chromo_view.Items.Clear() 'Avoids saving duplicate data through the text boxes
        For Me.i = 0 To tbl_chromo.Rows.Count - 1

            With chromo_view
                .Items.Add(tbl_chromo.Rows(i)("room_no"))
                With .Items(.Items.Count - 1).SubItems
                    .Add(tbl_chromo.Rows(i)("module_undertaking"))
                    .Add(tbl_chromo.Rows(i)("maximum"))
                    .Add(tbl_chromo.Rows(i)("lecturer_ID"))
                    '.Add(tbl_chromo.Rows(i)(""))
                    ' .Add(tbl_chromo.Rows(i)(st))
                    st = tbl_chromo.Rows(i)("room_no") & tbl_chromo.Rows(i)("module_undertaking"
                    chromo_view.Items(i).SubItems.Add(st)
                    '.Add(tbl_chromo.Rows(i)(st))
                End With
            End With
            next
            end sub
             obj_conc.Close()
```

Listing 4.3: Load Chromosomes

The above method is used to retrieve genes from relevant entities to a listview (table). In the same method, chromosomes are also generated and populated in the chromosome ID column for the initial population creation.

```
Imports System.IO
Imports System.Data.SqlClient
Imports Microsoft.VisualBasic
Imports System.Collections.Generic

Public Class chromosome_view
    Dim obj_conc As New SqlClient.SqlConnection 'create a variable that will be used to connect to the
    Dim obj_com As New SqlClient.SqlCommand
    Dim obj_adapter As New SqlClient.SqlDataAdapter
    Dim dataset As New DataSet
    Dim comstring As String
    Dim path_ As String
    Dim respond As MsgBoxResult
    Dim mod_ID As Long
    Dim lec_ID As Long
    Dim roomNo As Long
    Dim NoOfStudents As Long
    Dim binChromo As Integer
    Dim tempchromo As Integer
    Dim chromo_genes As String
    Dim index As String
    Dim st As String = ""
    Dim tbl_chromo As New DataTable
    Dim i As Integer
    Dim j As Integer
    Public Shared weekDayNames As New List(Of String)
    Public Shared lectureTimings As New List(Of String)()
```

Listing 4.4: Declarations

The declarations shown above are used across all classes to declare variables and namespaces used in the development of the product.

Chapter 5

CONCLUSION

5.1 FINDINGS

Genetic algorithms are mainly known to imitate the course of natural selection and can be utilized as a skill for unraveling difficult optimization complications which have very outsized search spaces.

Genetic algorithms can be and have been applied in different concepts such as gaming, scheduling and artificial intelligence. For GAs to be implemented there has to be chromosomes which are made up various genes (different data entries from different entities). Afterwards, these chromosome have to be represented either by arrays of bits, numbers, hash maps or binary. In this case, chromosomes were represented using binary bits. The following processes lead to the generation of the final population: gene retrieval, chromosome formation, population of timeslots, generation of initial population, fitness determination of candidates in the initial population, selection process, mutation and repairing.

5.2 EVALUATION

5.3 SETBACKS

1. Learning the concept of genetic algorithms and its complexity was challenging in the implementation process.
2. Having to come up with a new way chromosome representation form all the methods learnt, using Binary.
3. Implementing a criteria of awarding candidates in the initial population points.

4. Time spent on finding a way to initiate the crossover process, that is which chromosomes to use to attain the process of mutation.

5. The group adopted the V-model methodology at the start of the project which required modeling and building the system before hand. This was a setback in a way that it consumed more time in learning about the implementation of GAs in timetable scheduling which led to change of methodologies to Feature Driven Methodology (FDD) which showed progress afterwards.

6. Time limit was a factor, as implementing GAs for schedule making requires a lot of time to implement.

5.4 RECOMMENDATIONS

- In future the group intends to select a proper methodology in the early phases of the project.

- In future, the group suggests that GAs should be implemented with other algorithms to ease the complexity.

- Proper planning should be done, as in this case with GA, because most pahses solely depend on the previous phases that proceeds each phase.

- Proper resources should be provided to students. These may include computing resources such as full resourced labs with better network performance for research purposes.

- The group intends to properly manage the allocated for the project completion efficiently and effectively even in if there will still be lack or shortage of skills and resources to make the project a success.

Bibliography

[ABRAMSON and ABELA(1992)] D. ABRAMSON and J ABELA. A parallel genetic algorithm for solving the school timetabling problem, 1992.

[Bambrick(1997.)] L Bambrick, 1997.

[Davis(2007)] L Davis. Handbook of genetic algorithms, April 2007.

[Golub()] M Golub. An implementation of binary and floating point chromosome representation in genetic algorithms.

[Lan Fang(1992)] H. Lan Fang. Investigating genetic algorithms for scheduling, 1992.

[Larget(2012)] H. Larget. Genetic algorithms used in timetable management, 2012.

[Shengxiang and Sadaf(2009)] Y. Shengxiang and N J. Sadaf, 2009.

Appendices

Figure 5.1 depicts the first class diagram to be made.

5.4.1 minutes

Meeting DATE: 26/01/2014 PRESENT: - Lesdi Radikgomo - Oteng Ramarea - Omphile Simula - Naledi Mosele AGENDA: - Project brief discussion Project summary

1. Project brief layout.
2. Background information discussion about the current system.
3. Objectives stating.
4. Desired outcome discussion.
5. Methodology to be used.

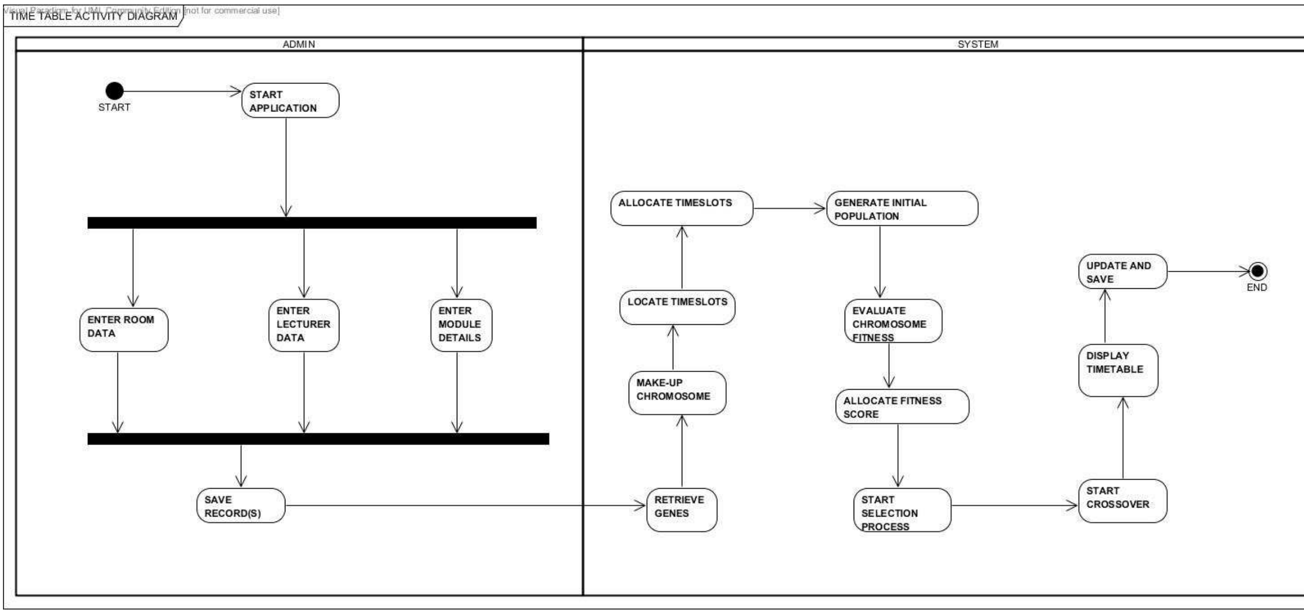


FIGURE 5.1: first class diagram

- Assigning of group member roles or duties.

Team leader: Omphile Simula Editor: Lesedi Radikgomo Secretary: Naledi Mosele

Meeting DATE: 10/02/2015 PRESENT:

1. Lesdi Radikgomo
2. Oteng Ramarea
3. Omphile Simula
4. Naledi Mosele

AGENDA:

1. Discussion on system design
2. Discussing about use case diagram
3. Discussing about class diagrams
4. Use case scenarios

   Meeting DATE: 12/02/2015 PRESENT:

   (a) Lesdi Radikgomo
   (b) Oteng Ramarea
   (c) Omphile Simula
   (d) Naledi Mosele

   AGENDA: Requirements analysis discussion

   (a) Requirements analysis must have both the user requirements and system requirements.
   (b) A requirements catalogue must be done. It comprises of the requirement, requirement ID, functional and non-functional requirements.

   Meeting DATE: 16/02/2015 PRESENT:

   (a) Lesdi Radikgomo
   (b) Oteng Ramarea
   (c) Omphile Simula
   (d) Naledi Mosele

   AGENDA: Meeting with the supervisor

5. The team met with the supervisor to get feedback on the project brief that was handed to the supervisor.

   Meeting DATE: 17/03/2015 PRESENT:

   (a) Lesdi Radikgomo
   (b) Oteng Ramarea
   (c) Omphile Simula
   (d) Naledi Mosele

   AGENDA:

   (a) Meeting with the supervisor

       [I]

       i. Get feedback on the project brief. The team was told to go and look for a better methodology. The Rapid Application Development methodology was used.
       ii. Get feedback on the requirements analysis. The team was told to go over the non-functional requirements again. (Non-functional requirements are defined as the constraints on the services or the functions offered by the system.)

   (b) Have a group meeting after seeing the supervisor

6. Discussed briefly about what the supervisor said and allocated each other tasks.