SOFTWARE REPORT

AUTOMATIC TIMETABLE GENERATOR USING GENETIC ALGORITHM

SOFTWARE ENGINEERING (CSE3001)

Submitted To

Prof. Swathi J.N.
Associate Professor
SCOPE

Submitted By

Achyut Tripathi
Regno: 17BCE0954
Mob no: 9651807367

Priyanshu Mittal
Regno: 17BCE0027
Mob no: 7014776110

Table of Content

Table of Content 2
Introduction 3
Theoretical background 3
Motivation 6
Aim of the proposed work 6
Objective of the proposed work 7
Report Organization 7
Literature Survey 7
Survey of the existing models 7
Summary Gaps identified in the Survey 8
Proposed System Requirements Analysis and Design 8
Introduction 8
Requirement Analysis 9
Functional Requirements 9
Product Perspective 9
Product Features 9
User Characteristics 13
Domain Requirements 14
User Requirements and Product Specific System Requirements 14
Non Functional Requirements 16
System Requirements 18
H/W Requirements 18
S/W Requirements 18
Design of the proposed system 18
Introduction 18
High level Design 18
Process Model 18
Module Decomposition 20
Activity Network 20
Detailed Design (ER Diagram/UML Diagram/Mathematical Modeling) 21
ER Diagram 21

Use-Case Diagram 22
Sequence Diagram 23
Collaboration Diagram 23
Login 23
Choosing course 24
Teacher Courses 24
Class Diagram 25
State Diagram 26
Component Diagram 27
Data flow Diagram 27
Implementation and Testing 29
Conclusion 30
Limitations 35
Future Work 36
References 36

Introduction

Theoretical background

Time Table Scheduling is an NP-hard problem and hence polynomial time verifiable using genetic algorithms. It a typical scheduling problem that appears to be a tedious job in every academic institute once or twice a year. In earlier days, time table scheduling was done manually with a single person or some group involved in task of scheduling it manually, which takes a lot of effort and time.

Planning timetables is one of the most complex and error-prone applications. Timetabling is the task of creating a timetable while satisfying some constraints. There are basically two types of constraints, soft constraints and hard constraints. Soft constraints are those if we violate them in scheduling, the output is still valid, but hard constraints are those which if we violate them; the timetable is no longer valid. The search space of a timetabling problem is too vast, many

solutions exist in the search space and few of them are not feasible. Feasible solutions here mean those which do not violate hard constraints and as well try to satisfy soft constraints. We need to choose the most appropriate one from feasible solutions. Most appropriate ones here mean those which do not violate soft constraints to a greater extent.

In this project hard-constraints have been taken care of strictly and it has been ensured that soft-constraints are as well followed as much as possible.

Soft-constraints (flexible):

- More or less equal load is given to all faculties.
- Required time (hours per week) is given to every Batch.

Hard-constraints (rigid):

- There should not be any single instance of a faculty taking two classes simultaneously.
- A class group must not have more than one lectures at the same time.
- A genetic algorithm maintains a population of candidate solutions for the problem at hand, and makes it evolve by iteratively applying a set of stochastic operators.

- Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype or the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions.

- Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible.

- The evolution usually starts from a population of randomly generated individuals and happens in generations.

- In each generation, the fitness of every individual in the population is evaluated, multiple individuals are selected from the current population (based on their fitness), and modified (recombined and possibly mutated) to form a new population.

- The new population is then used in the next iteration of the algorithm.

- Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

- If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

Before we can use a genetic algorithm to solve a problem, a way must be found of encoding any potential solution to the problem. This could be as a string of real numbers or, as is more typically the case, a binary bit string. We will refer to this bit string from now on as the chromosome. A typical chromosome may look like this:

10010101110101001010011101101110111111101

At the beginning of a run of a genetic algorithm a large population of random chromosomes is created. Each one, when decoded will represent a different solution to the problem at hand. Let's say there are N chromosomes in the initial population. Then, the following steps are repeated until a solution is found.

- Test each chromosome to see how good it is at solving the problem at hand and assign a Fitness Score accordingly. The fitness score is a measure of how good that chromosome is at solving the problem to hand.

- Select two members from the current population. The chance of being selected is proportional to the chromosomes fitness. Roulette Wheel Selection is a commonly used method.

- Dependent on the Crossover rate crossover the bits from each chosen chromosome at a randomly chosen point.

- Step through the chosen chromosomes bits and flip dependent on the Mutation rate.

- Repeat step 2, 3, 4 until a new population of N members has been created.

- Keep repeating until required fitness is achieved.

Motivation

Timetable creation for the FFCS(Fully Flexible Credit System) is a very arduous and time consuming task. To create timetable it takes lots of patience and man hours. Time table is created for various purposes like to organize lectures in school and colleges, to create timing charts for train and bus schedule and many more. To create timetable it requires lots of time and manpower .In our project we have tried to reduce these difficulties of generating timetable by Genetics Algorithm. By using Genetic algorithm we are able to reduce the time required to generate time table and generate a timetable which is more accurate, precise and free of human errors. The first phase of FFCS contains all the common compulsory classes of the VIT, which are scheduled by a central team. The second phase of FFCS registration contains the individual departmental classes. Presently this timetable is prepared manually by students, by manipulating those of earlier years, with the only aim of producing a feasible timetable.

Aim of the proposed work

The system will greatly reduce time and effort spent on generating a schedule, therefore giving more time for the administrator to manage the institute. Multiple educational institutes can adopt the system and benefit by having a computed scheduling solution. It will help ease the management of school and possibly reduce expenses by having a more utilized schedule. The system is built to support scheduling for levels lower than college. Developing the system will greatly help the researcher towards improving on computer science due to implementation of multiple algorithms. It will also help provide a step for the growing artificial intelligence community. The research is going to be included in the portfolio which can display the skills of the researcher in tackling complex problems. Researchers can use this research as a source for their research. They may use this approach on solving optimization problems or even create a more efficient version. This research comes with helpful data on understanding the performance and process which will help future researchers.

Objective of the proposed work

- To allow students to create their account and enter their subject preferences.
- To allow user to select time and faculty.
- To allow user to feed the database of available faculty, courses and rooms.

Report Organization

The report will be giving us an idea of all similar applications and how this particular application stands out from the rest .

Followed by this, we will make a requirements Analysis and Design for the proposed system including the requirement analysis, requirement specification and verification, and functional as well as non functional - Requirements This will be followed by the design of the proposed system, . the architectural model and also the proposed process flow model This is . followed by a high level design and a corresponding detailed design for the same.

The design phase is followed by implementation and testing, and finally concluded by specifying the limitations and further scope of the project .

Literature Survey

Survey of the existing models

Trying to develop a software which helps to generate timetable for VIT automatically. By looking at the existing system we can understand that timetable generation is done manually. Manually adjust the timetable when any of the faculty is absent, and this is the big challenge for the Automatic Timetable Generator that managing the timetable automatically when any of the faculty is absent.

FET is free software, licensed under the GNU Affero General Public License version 3. Fully automatic generation algorithm, allowing also semi-automatic or manual allocation Platform independent implementation, allowing running on GNU/Linux, Windows, Mac and any system that Qt supports Flexible modular XML format for the input file, allowing editing with an XML editor or by hand (besides FET interface) Import/export from CSV format. The resulted timetables are exported into HTML, XML and CSV formats. Flexible students structure, organized into sets: years, groups and subgroups. FET allows overlapping years and groups and non-overlapping subgroups. You can even define individual students (as separate sets). Each constraint has a weight percentage, from 0.0% to 100.0% (but some special constraints are allowed to have only 100% weight percentage).

Summary Gaps identified in the Survey

The survey is has been mostly limited to software's being used in India and the survey results may vary widely if conducted in a different country due to shift in the stakeholders, or rather the people who would be benefitted by the software.

Proposed System Requirements Analysis and Design

Introduction

The purpose of the Software Requirements Specification is to describe the specific requirements of the Time-Table Generator project that are to be met by the implementation effort of Genetic Algorithm. Included with the description of the requirements is a description of any constraints or assumptions that the project is working within.

This document also provides a description of any project dependencies that need to be explicitly expressed. Along with the requirements descriptions, it is also the purpose of this document to

describe any performance requirements that need to be met. If there are any standards that need to be considered when developing the software are also listed.
Lastly, the purpose of this document is to communicate the system attributes of the Time-Table Generator. These system attributes include reliability, availability, scalability, maintainability, and portability.

Requirement Analysis

Functional Requirements

Product Perspective

Timetable creation for the FFCS(Fully Flexible Credit System) is a very arduous and time consuming task. To create timetable it takes lots of patience and man hours. Time table is created for various purposes like to organize lectures in school and colleges, to create timing charts for train and bus schedule and many more. To create timetable it requires lots of time and manpower .In our project we have tried to reduce these difficulties of generating timetable by Genetics Algorithm. By using Genetic algorithm we are able to reduce the time required to generate time table and generate a timetable which is more accurate, precise and free of human errors. The first phase of FFCS contains all the common compulsory classes of the VIT, which are scheduled by a central team. The second phase of FFCS registration contains the individual departmental classes. Presently this timetable is prepared manually by students, by manipulating those of earlier years, with the only aim of producing a feasible timetable.

Product Features

- **Login Authentication**

The timetable generator project, a proper authenticated id for each and every student and as well as for teachers and the staff through which they will login and with their secure passwords and the system will verify the authentication of the user and its typed password. This is the first step for accessing the time table generator so firstly user has to set his password for the login and it should pass all the constraints of the password setting for successful signup.

Functional Requirements

Purpose: User authentication.

Input: Provide username and the chosen password.

Processing: Provided username and password should be verified through the database.

Output: If the provided username and password is correct then user shall be able to use the Time Table Generator.

Stimulus Response

User Action	System Response
1. Fill username and password	
2. Press the login button.	
	3. Verify the entered username and password from the database.
	4. If the given credentials are correct then take the user in his workspace.

- **Timetable Generation**

Making a class schedule is one of those NP hard problems. The problem can be solved using a heuristic search algorithm to find the optimal solution, but it only works for simple cases. For more complex inputs and requirements, finding a considerably good solution can take a while, or it may be impossible. This is where genetic algorithms come in to the game.

Functional Requirements

Purpose: Receives the user id and the password then the user choices that he made for selecting the courses and the desire slots and the faculties and the buildings for the particular courses.

Input: User Email id- Password, User ID number(Registration Number), Course Title, Course Code, Slot(Timing includes).

Processing: If the user id and password is correct then let him enter to the time table generating stage then after filling its preferred course title/ course code and selecting his time slot including the building and the teacher name for that particular subject. The system will confirm the request and will allocate that particular teacher and the course with the corresponding time slot.

Output: The chosen course and the faculty with the timing slot and the building will be added to the user time table if there is no clash between any two subjects and slot timing.

Stimulus Response

User Actions	**System Actions**
	(1) Receive user id and the password
	(2) Check the login authentication.
	(3) Will provide the GUI or the platform to the user according to its need.
	(4) Will create the time table base using the Genetic Algorithm.
	(5) Fetch all the modules for the making of the time table generation possible including the constraints.
	(6) Gather all the data regarding the user, courses, teachers, building, time slots and their availability.
	(7) All undesirable courses and the previous memories will be deleted and will made a new time table

(8) User will be able to scale the time table of its own choice and demands and needs.	
(9) User will be able to manipulate its choices and will be able to edit them at any point of time.	
	(10) Saves the users Time Table if it click for save the time table.

- **Discussion Portal**

This feature allows the user to post and read the various question & answers in a public discussion portal.

Functional Requirements

Purpose: Post and replying to a query in the discussion portal.

Input: Text Message, Donar's ID, Subject of the question (optional), and question or answer.

Processing: If the proper formatting of question or answer is require then ask the user to rewrite.

Output: Question/answer is displayed on the public discussion portal within a single thread.

Stimulus Response

User Action	System Response
1).Write a question/answer and click enter	
	2). Check the formatting of the written message.

	3). It will post the message in public.
4). User will be able to answer the questions and can ask more.	

User Characteristics

The following table identifies and describes the different users of the Time-Table Generator software. The information gathered about the different users of the system helped define what the software needs to do. Also, these users are referenced in the requirements and diagrams.

User	**Description**
Student	Student is the main user for the project which is going to use this time table for the max.
Teacher	Teacher is the side user of the software which is going to set his/her time table so that it can be followed by the students.
Admin	Admin is the governing object of the software. It help in making the best suited timetable for the students and is trying to run the software in its best efficiency.
Head of Department(HOD)	HOD is the timetable maker for the teachers, which will assign courses, slots and batches to them. He is the

Domain Requirements

Domain requirements are the requirements which are characteristic of a particular category or domain of projects. The basic functions that a system of a specific domain must necessarily exhibit come under this category. For our case, in a timetable generator system, the functionality of being able to make timetable from the given courses and faculty as well as modifying the timetable as per user's need.

Domain requirements are important because they often reflect fundamentals of the application domain If these requirements are not . satisfied, it may be impossible to make the system work satisfactorily.

User Requirements and Product Specific System Requirements

The follow is a table of the requirements that the system SHALL meet.

STAKEHOLDER	REQUIREMENT
Student	● Student SHALL be able to sign up or login. ● Student SHALL be able to select courses. ● Student SHALL be able to view all the slots. ● Student SHALL be able to view the teachers available for each courses. ● Student SHALL be able to ask questions on discussion portal. ● Student SHALL be able to modify timetable. ● Student SHALL be able to file wish list. ● Student SHALL be able to download timetable.

	● Student SHALL be able to give the feedback.
Admin	● Admin SHALL be able to login. ● Admin SHALL be able to list or modify the guidelines. ● Admin SHALL be able to modify timetable. ● Admin SHALL be able to access the discussion portal.
Teachers	● Teachers SHALL be able to sign up or login. ● Teachers SHALL be able to select their courses. ● Teachers SHALL be able to access the discussion portal. ● Teachers SHALL be able to provide details of their free hours.
HOD	● HOD SHALL be able to provide the list of courses to be offered in the semester. ● HOD SHALL be able to assign the slots to the teachers. ● HOD SHALL be able to issue the set of guidelines to be followed. ● HOD SHALL be able to re-conduct the course allotment.

Non Functional Requirements

Non-functional requirements describe the constraints on the services and/or functions offered by the system and constraints on the development process and standards. These requirements are grouped into three main types: Product requirements which specify product behavior, Organizational requirements which are derived from policies and procedures in the customer's and developer's organization, and External requirements which are derived from factors external to the system and its development process. The TTGS System must have the following attributes:

Usability

The system must have an easy to use and understandable user interface. This will ensure that the user, for which this system is intended, will be able to achieve a level of proficiency with the system with minimum effort in a very short period of time.

Efficiency

Response time, memory utilization, processing time and other performance factors should be optimal in the system. In the event that only the most minimum of system resources are available, the system should be able to function optimally and it should not make wasteful use of system resources.

Reliability

The system's functions and services should be available to its users most of the times. The rate of failure and/or the probability of it not being available should both be very low. Failures in the system should not occur too often, at least a longer time space between them.

Dependability

In the event of the system failing due to any reason, it should not cause any physical or economic damage to the customers, end users or the developers.

Robustness

The time taken for the system to restart after a system failure should be very short. The percentage of user and system events that cause failures should be very minimal. The probability of data being corrupted due to a failure should be very minimal as well.

Maintainability

Since software specifications change all the time as business needs change for organizations, the TTGS system has been designed in such a way that it easily evolves meeting the customers' changing demands. The system should be easy to maintain so as to cut on the maintenance costs.

Flexibility

The system should be modifiable depending on the changing needs of the user. Such modifications should not entail extensive reconstructing or recreation of software. It should also be portable to different computer systems.

Security

This is a very important aspect of the design and should cover areas of hardware reliability, fall back procedures, physical security of data and provision for detection of fraud and abuse. System design involves logical design first and then physical construction of the system. The logical design describes the structure and characteristics of features, like the outputs, inputs, files, database and procedures. The physical construction, which follows the logical design, produces actual program software, files and a working system.

Portability

The system should be installed or re-installed somewhere else with little or no modification. The system shall be cross platform and be compatible with a number of common architectures and should also be able to run on a variety of operating systems.

Consistency

The system shall maintain a constant look and appearance and exhibit high levels of consistency among its various modules and components.

System Requirements

H/W Requirements

This project has no specific hardware requirements It only requires a . personal computer with a decent processor.

S/W Requirements

The project requires a python compiler, preferably the 3.7 version of python. Following are the dependencies that should be installed in python:

- PyQT5
- Numpy
- psutil

Design of the proposed system

Introduction

System design is the specification or construction of a technical, computer-based solution for the business requirements identified in system analysis. It gives the overall plan or model of a system consisting of all specification that give the system its form and structure (i.e. the structural implementation of the system analysis).
The objective of designing the new system is to provide an automated timetabling system that will generate a feasible timetable.

High level Design

Process Model

We are using Throw Away Model because it will complete our project as fast as possible. We will create a project prototype to either ensure a feature is technically feasible or to show the feature to stakeholders or potential users in order to gather feedback from them. We will not use

the prototype in future use which will help us to focusing on the actual feature, while leaving aside aspects such as maintainability of the code, style, design patterns or testing. This makes it possible to finish the prototype very fast, without affecting negatively the technical debt of the final product. This model will definitely help us to understand the end user requirements and hence develop a better requirements specification for the project.
The other models were deemed as unsuitable for the following reasons:

- We are NOT using Waterfall Model because our end user requirements are not fully clear and there is high amount of risk and uncertainty is there in the case of modifying the project to add new features in the software. Also there will be a high workload in the end of the time period and it won't allow us to face the challenges of developing the project from the beginning. It will be difficult to show the progress in the project review.

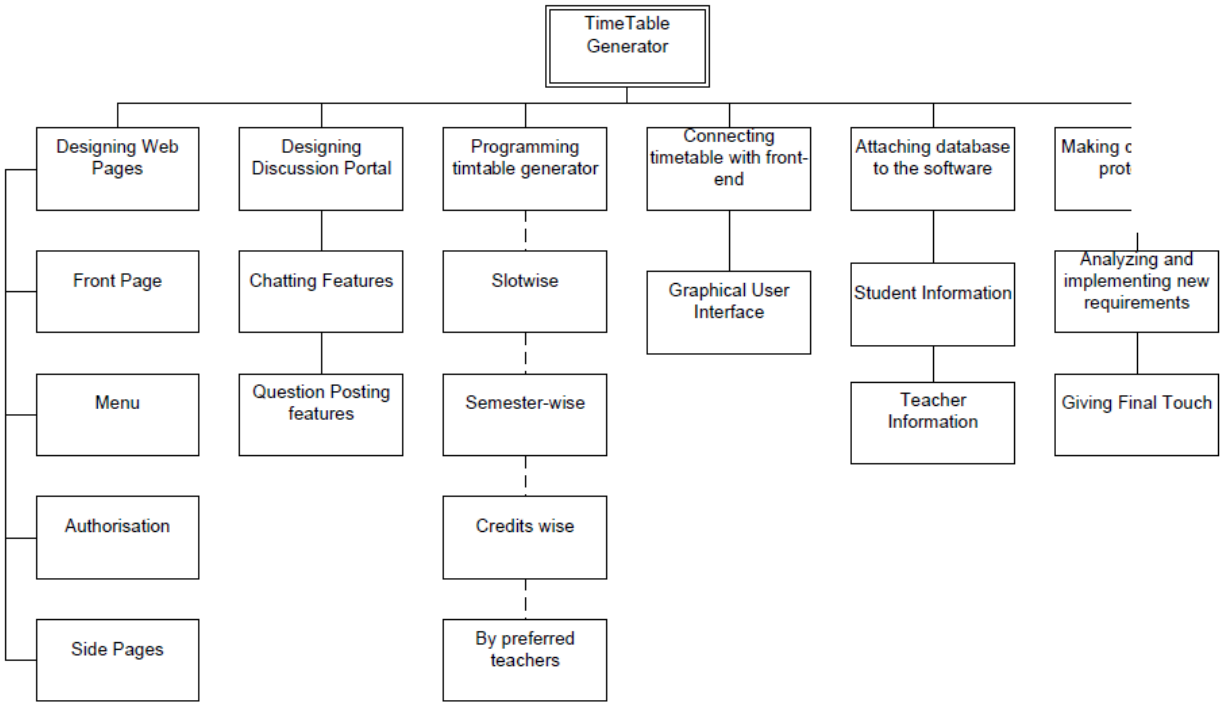
- We are NOT using Incremental Model because our requirements are not fully clear and we cannot foreseen all the requirements in the short planning stage. Constant management is required which cannot be provided in the team of two members. For our small project it not convenient to use this model.

- We are NOT using Spiral Model because our project is small and it is ineffective in this. There are not so much of critical risks associated to our project. Lots of documentation of the project is required for every cycle which is unnecessary for the small projects. Spirals can be go on indefinitely if for example a certain user can't manage to get all the required subjects in the semester and he doesn't want to study the available subjects. So this adding and dropping the subjects will continue indefinitely.

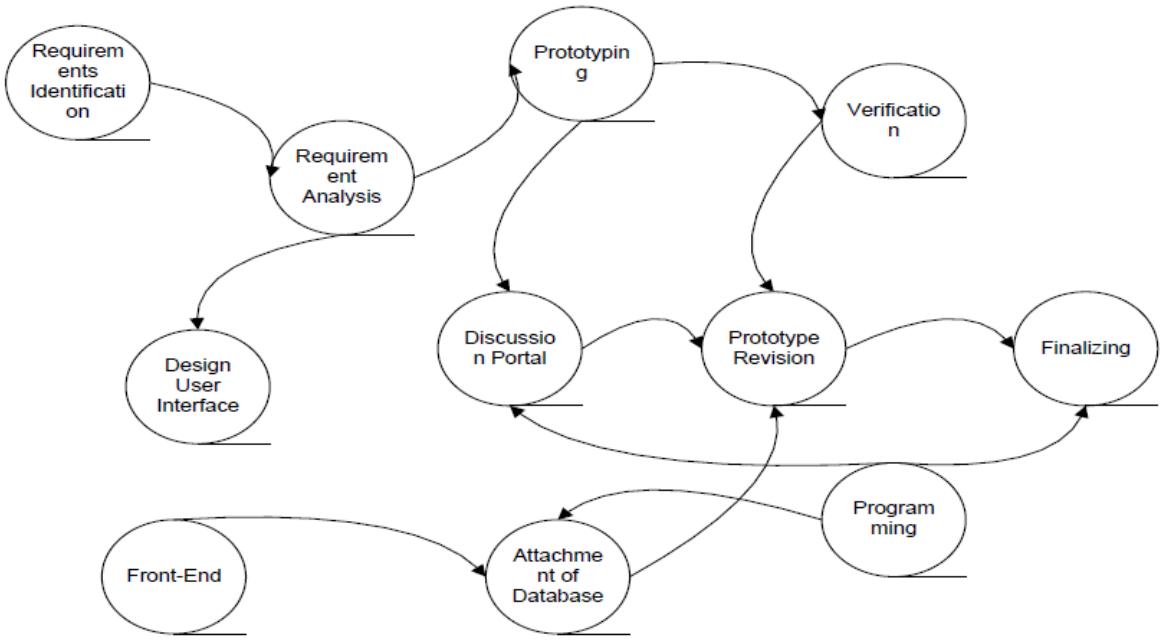
- We are NOT using Exploratory Model because after starting to develop the software we do not want to leave the complete project and start from the beginning if a new feature need to be added. For example, we a student want slot-wise filter in the software, we do not want to just leave all the progress and start making from the start.

- We are NOT using Component based Model because we don't have any components of the project available already. If we had components it would cost compromises in the project and hence result in the decrease of functionalities of the project.

Module Decomposition

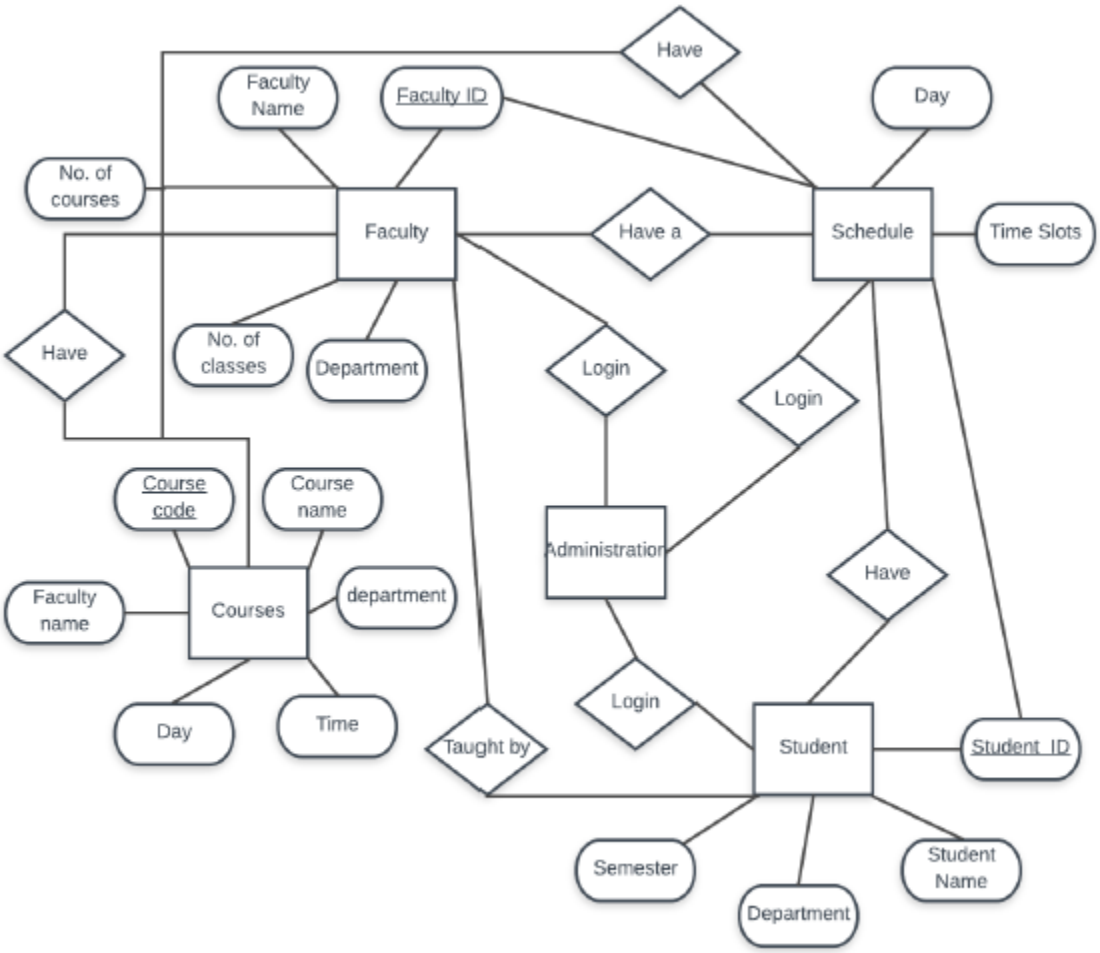


Activity Network

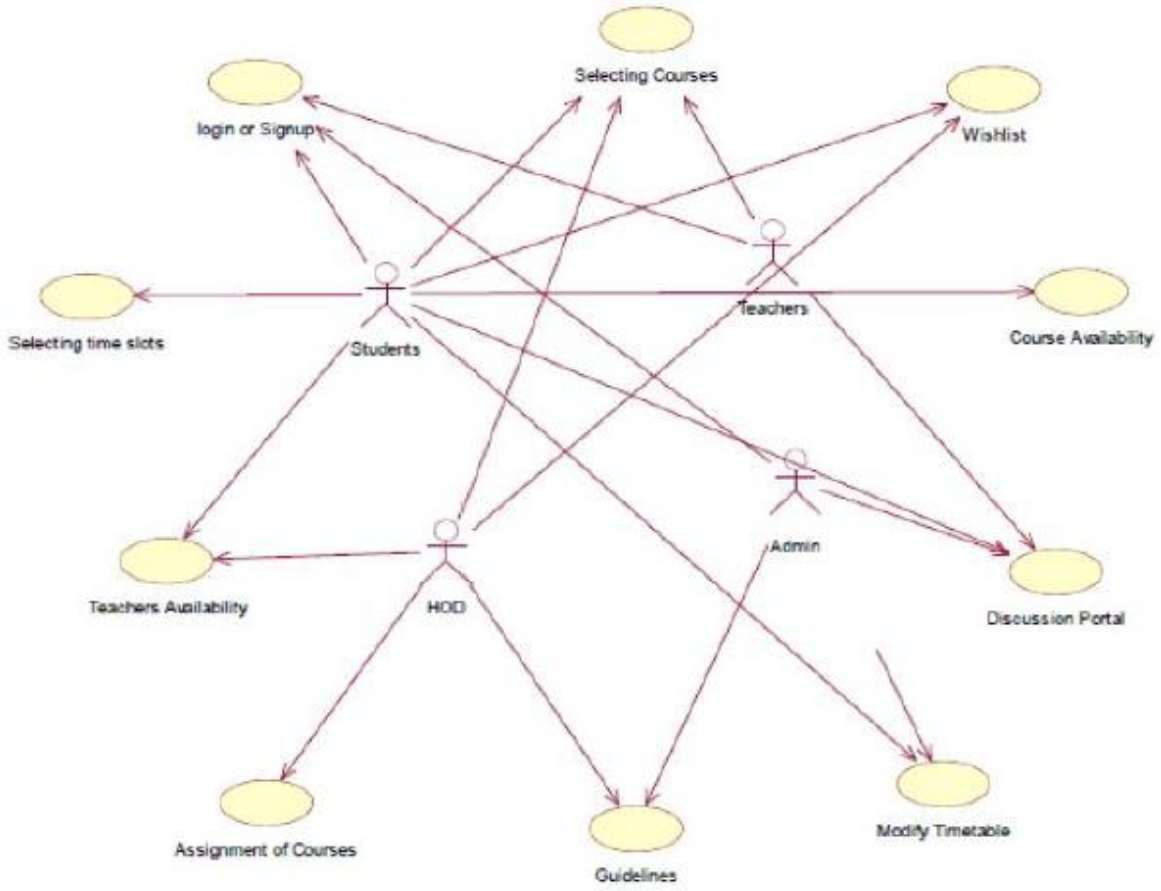


Detailed Design (ER Diagram/UML Diagram/Mathematical Modeling)

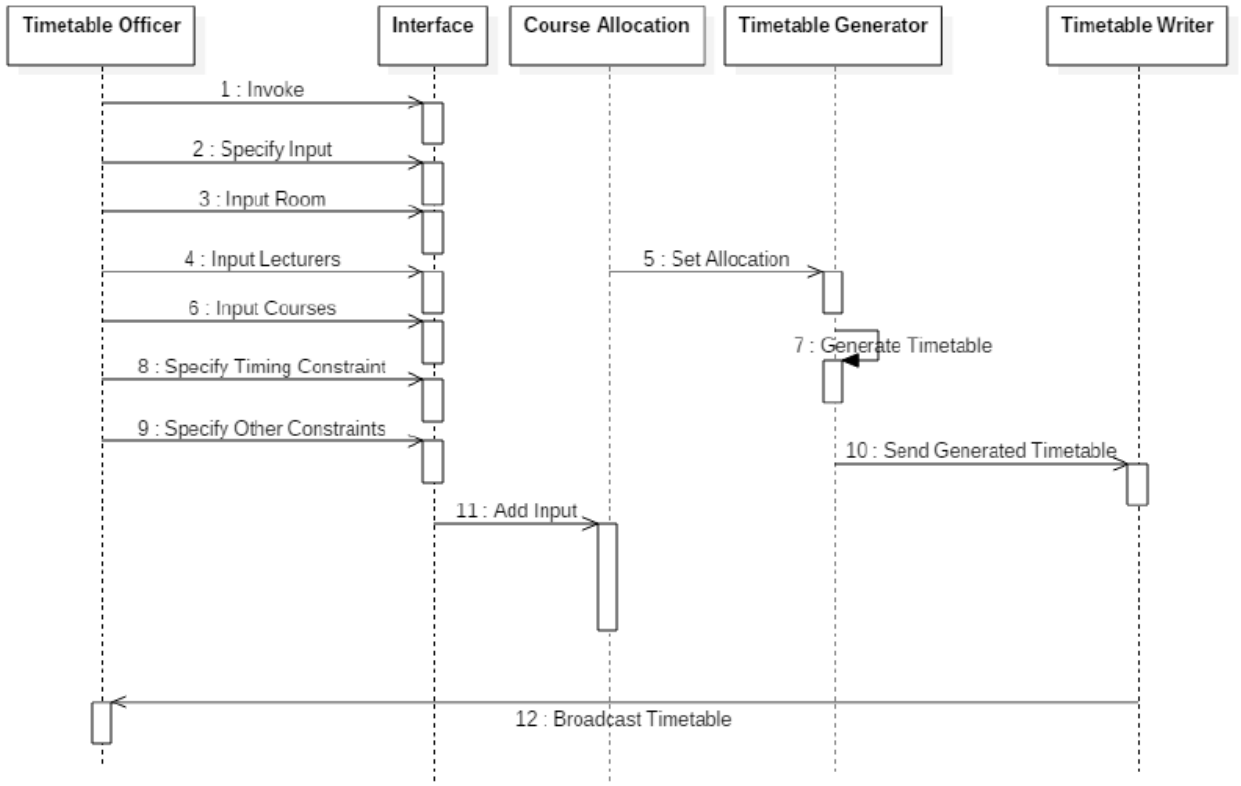
ER Diagram



Use-Case Diagram

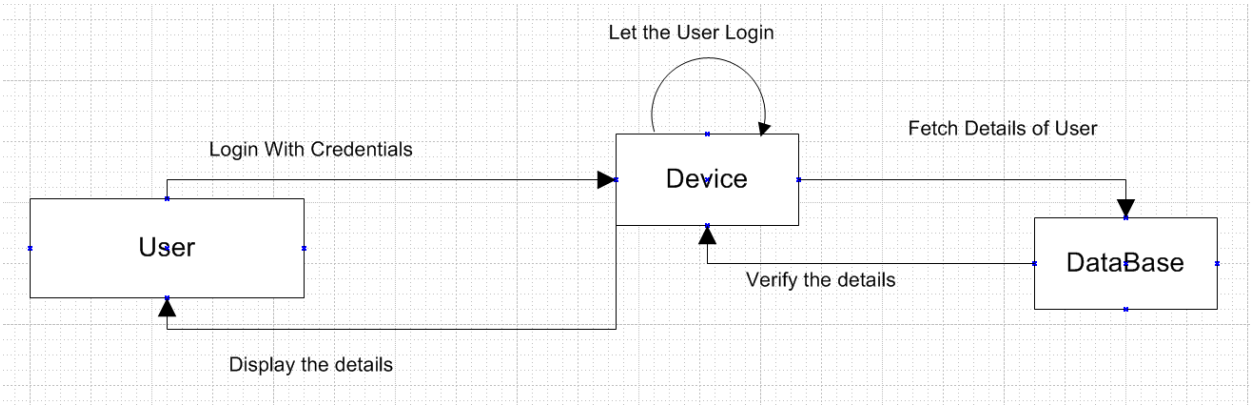


Sequence Diagram

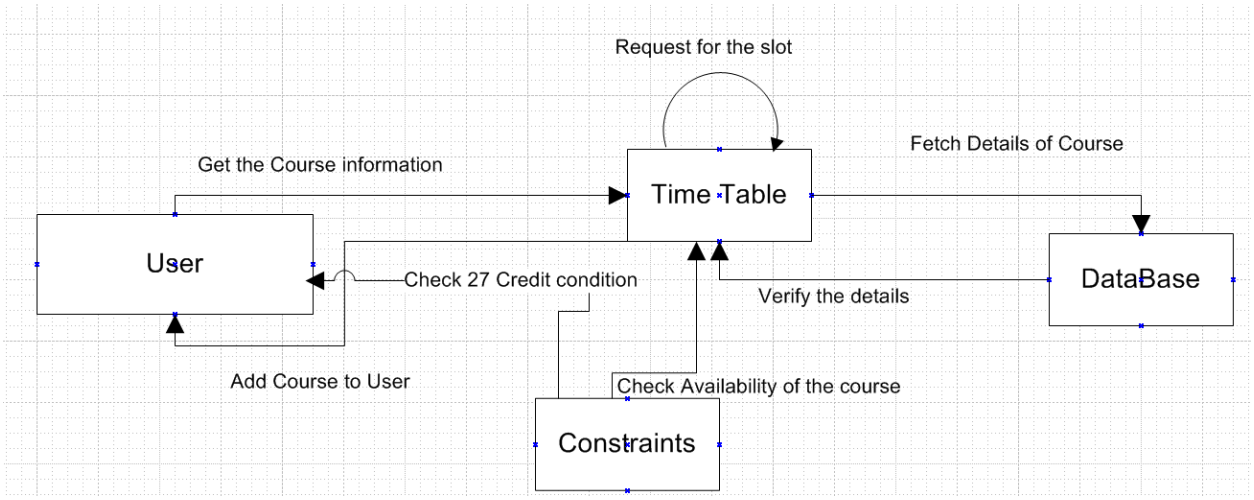


Collaboration Diagram

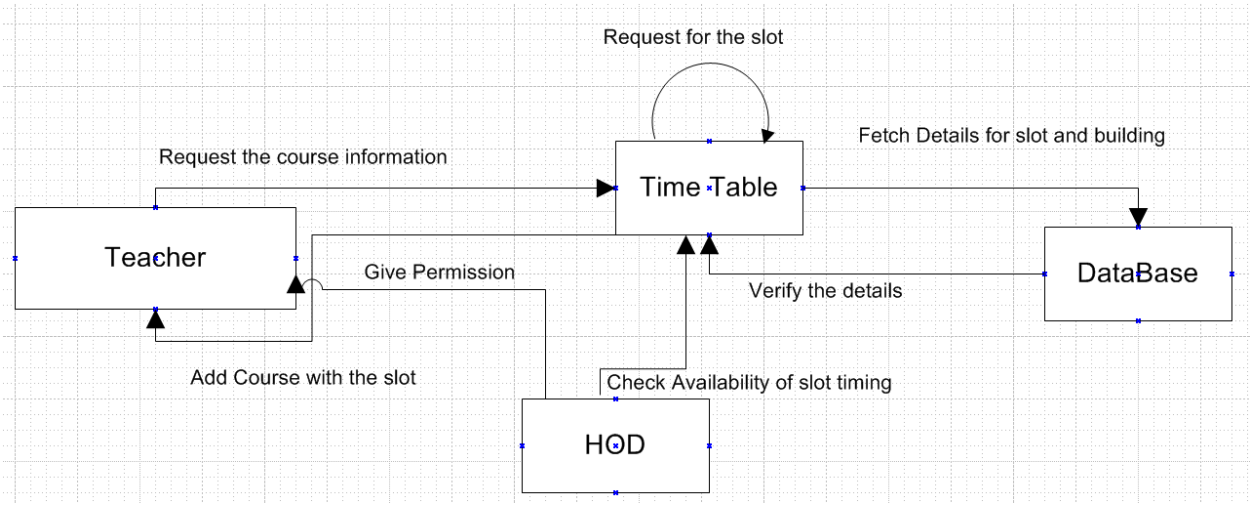
Login



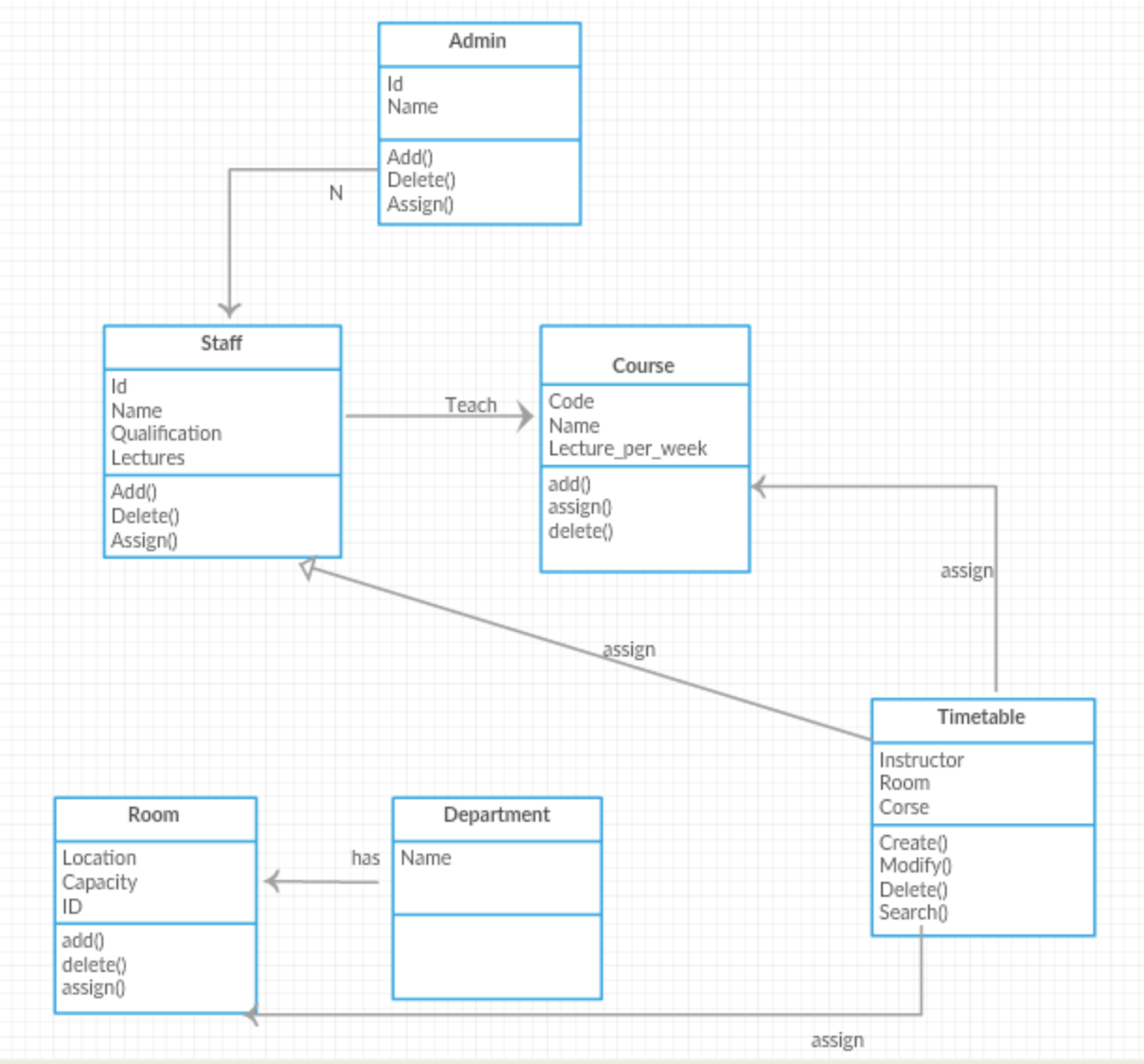
Choosing course



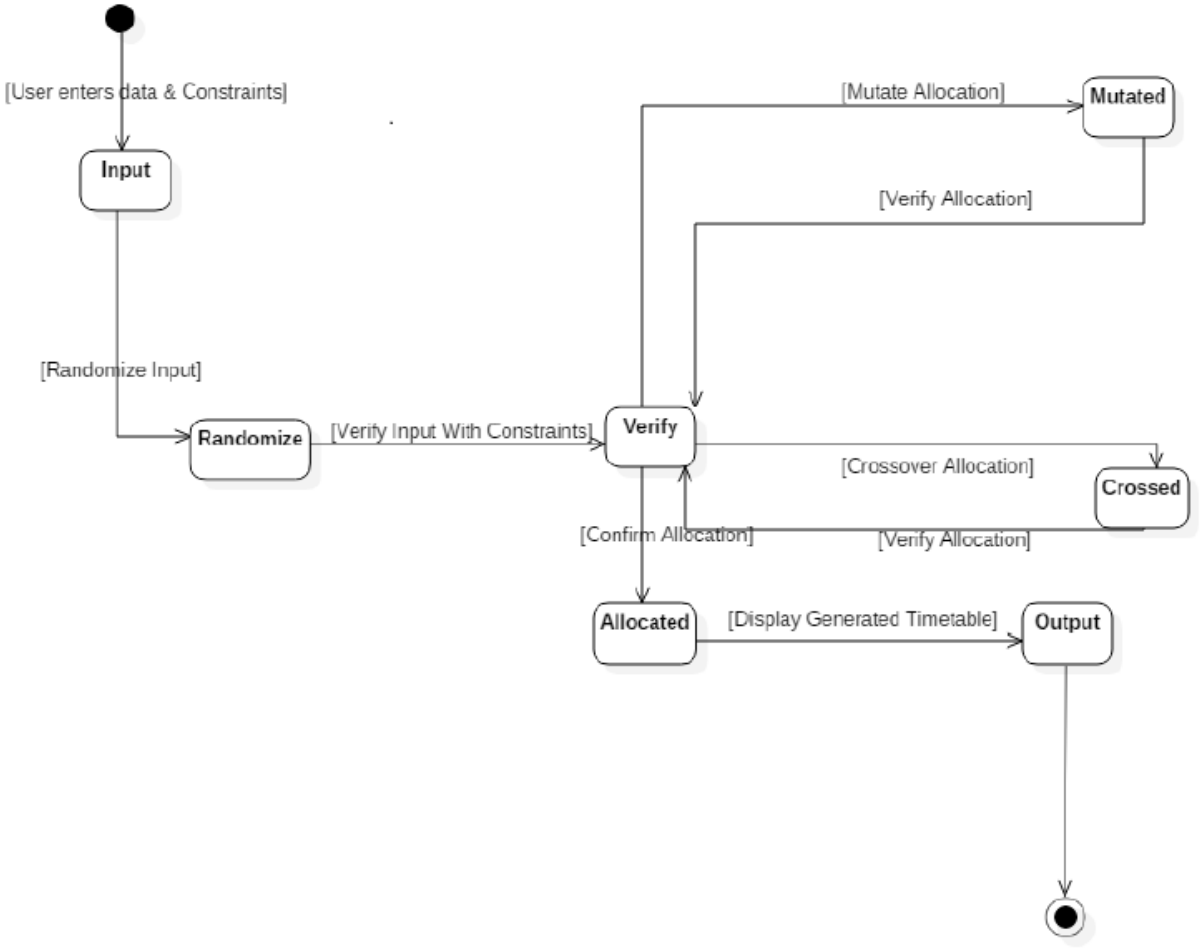
Teacher Courses



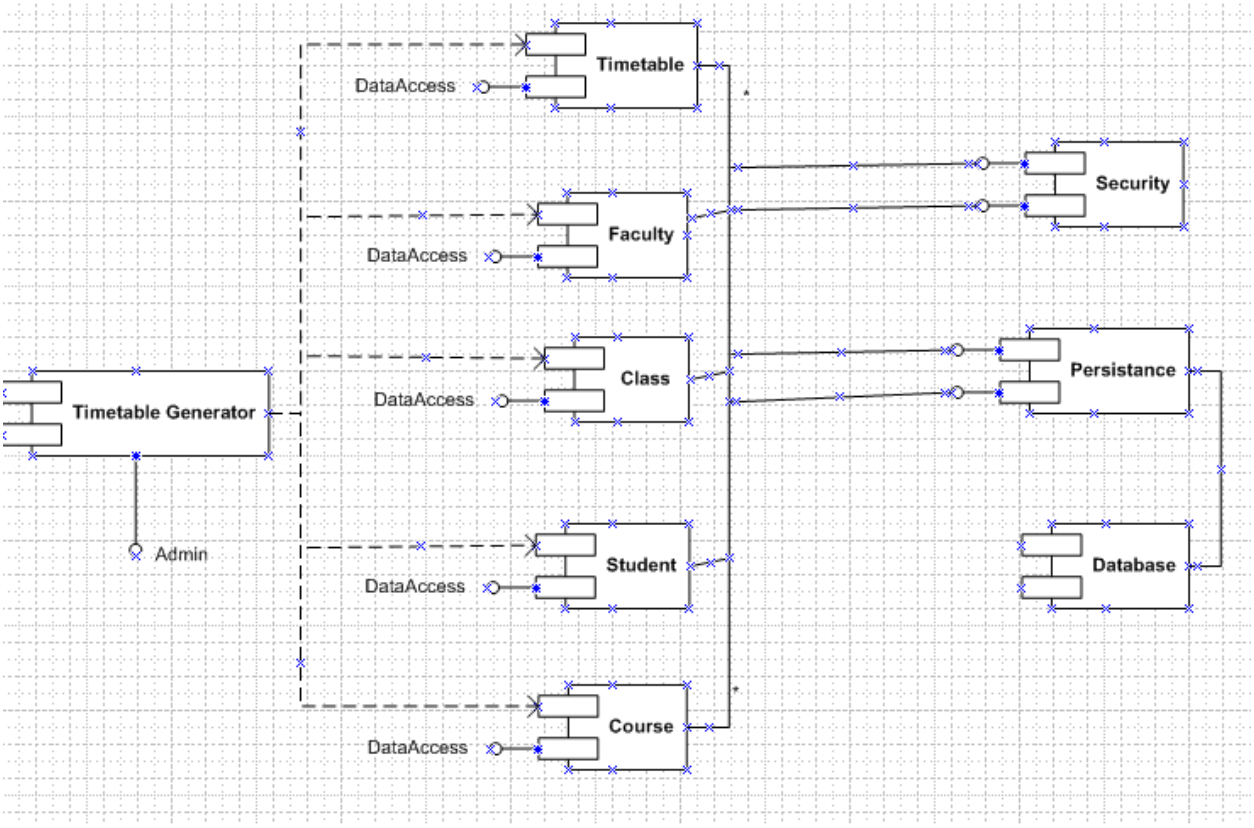
Class Diagram



State Diagram



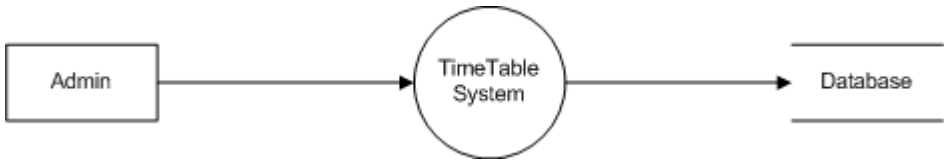
Component Diagram



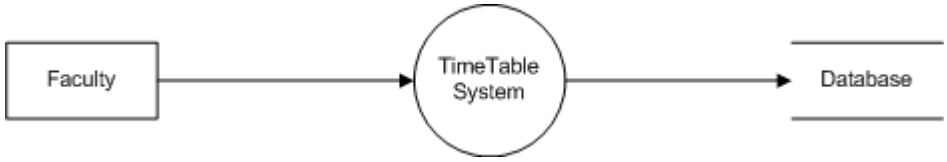
Data flow Diagram

Level - 0:

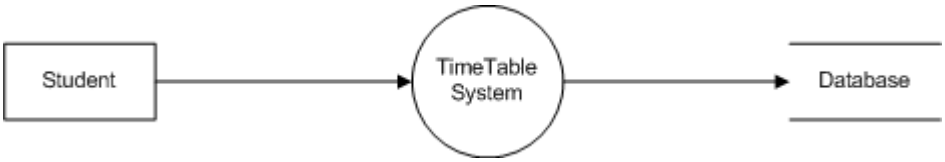
Admin Module:



Faculty Module:

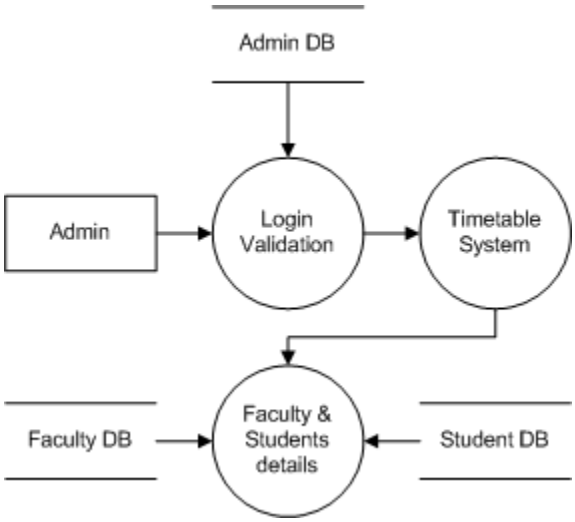


Student Module:

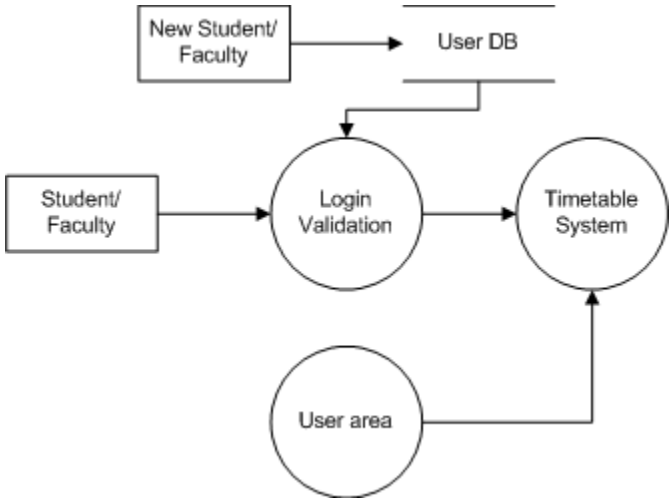


Level - 1:

Admin Module:



Student/Faculty Module:

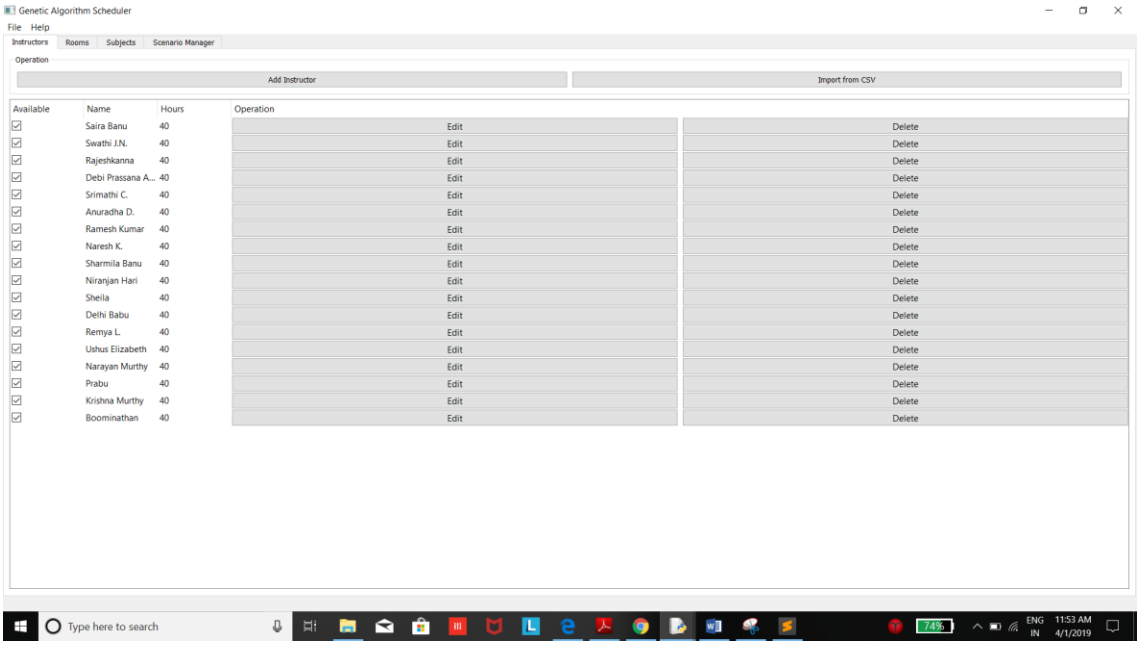


Implementation and Testing

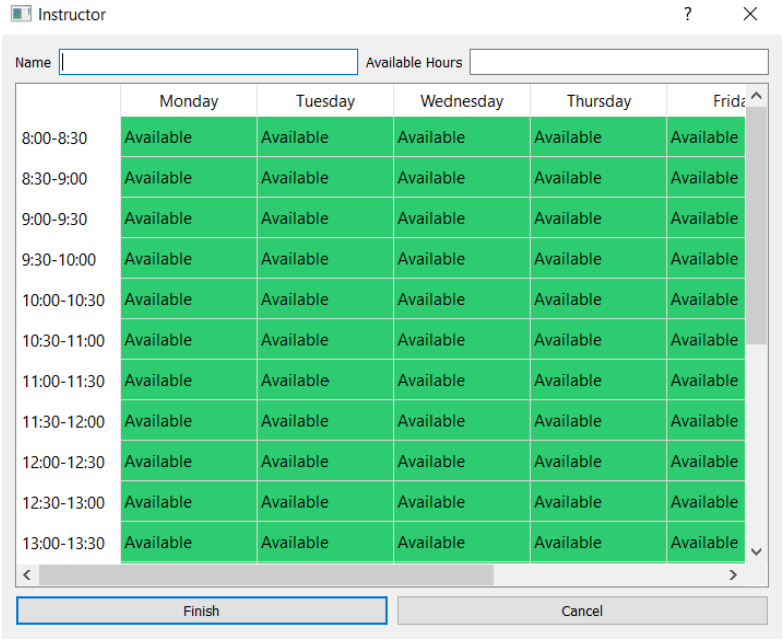
Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
1	Add Instructor	Name, Available hours, Available/Not available	Faculty registered Successfully	Faculty registered successfully	Pass
2	Add Course	Course_name, Course_Code, hours/week, faculty_select	Course registered successfully	Couse registered successfully	Pass
3	Search Course	Couse_code, Course_name	Course availability	Course Availability	Pass
4	Timetable Generation	Select start and end time, select algorithm parameters, Generate	Timetable	Timetable	Pass
5	Modify Timetable	Select desired course, faculty or slot	Timetable modified	Modified Timetable	Pass
6	Export Timetable	Click on the export button	Timetable downloaded	Timetable downloaded	Pass

Testing

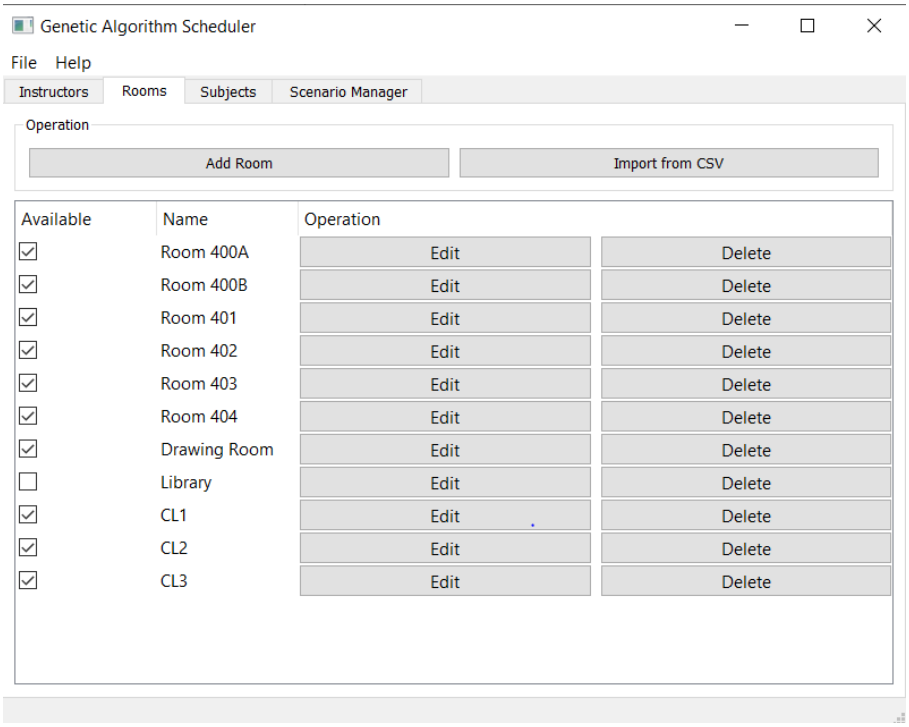
Add Instructor:



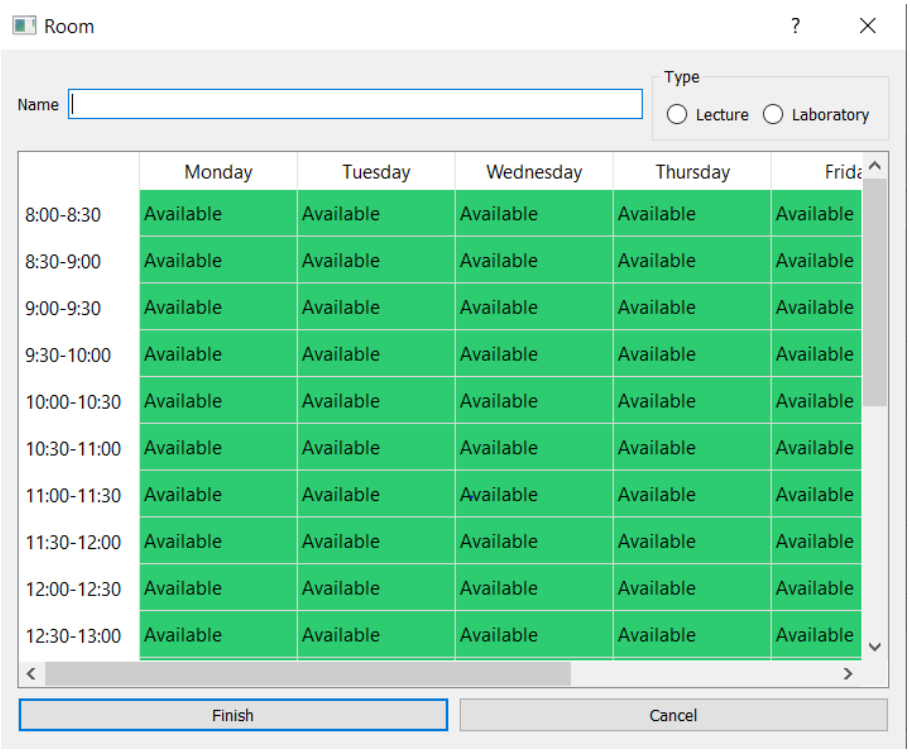
Add Instructor Manually



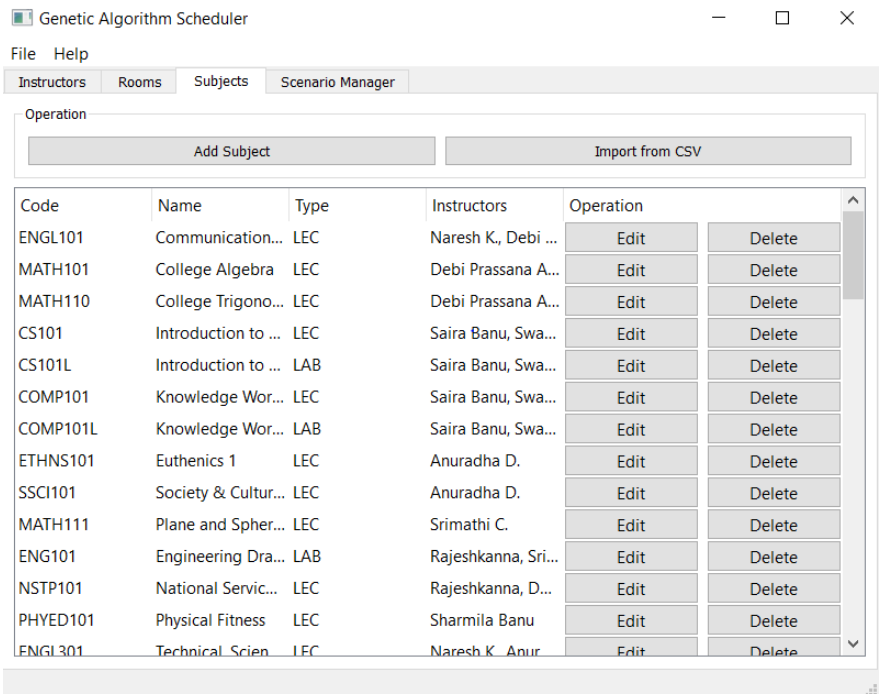
Add Rooms:



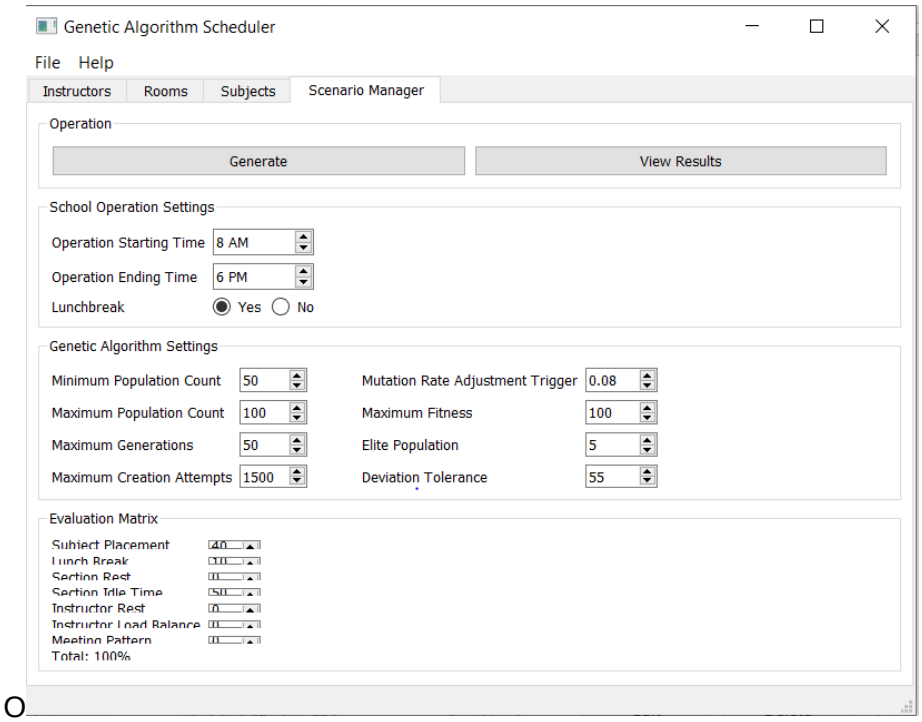
Add Rooms Manually:



Add Subjects:

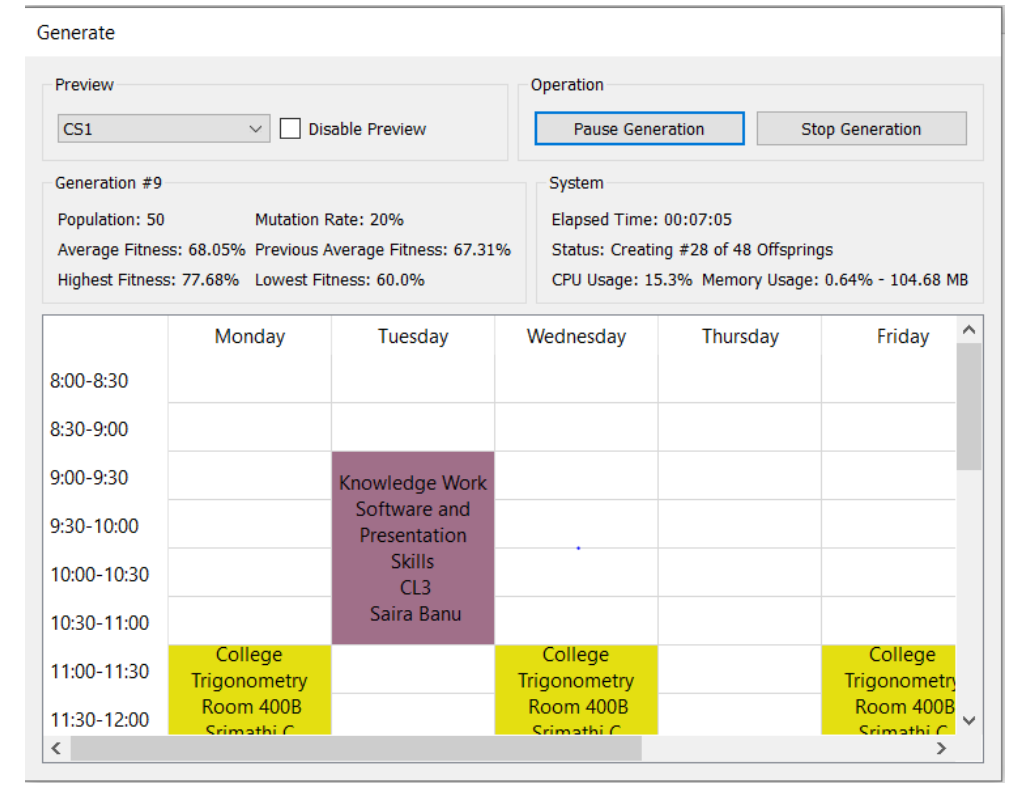


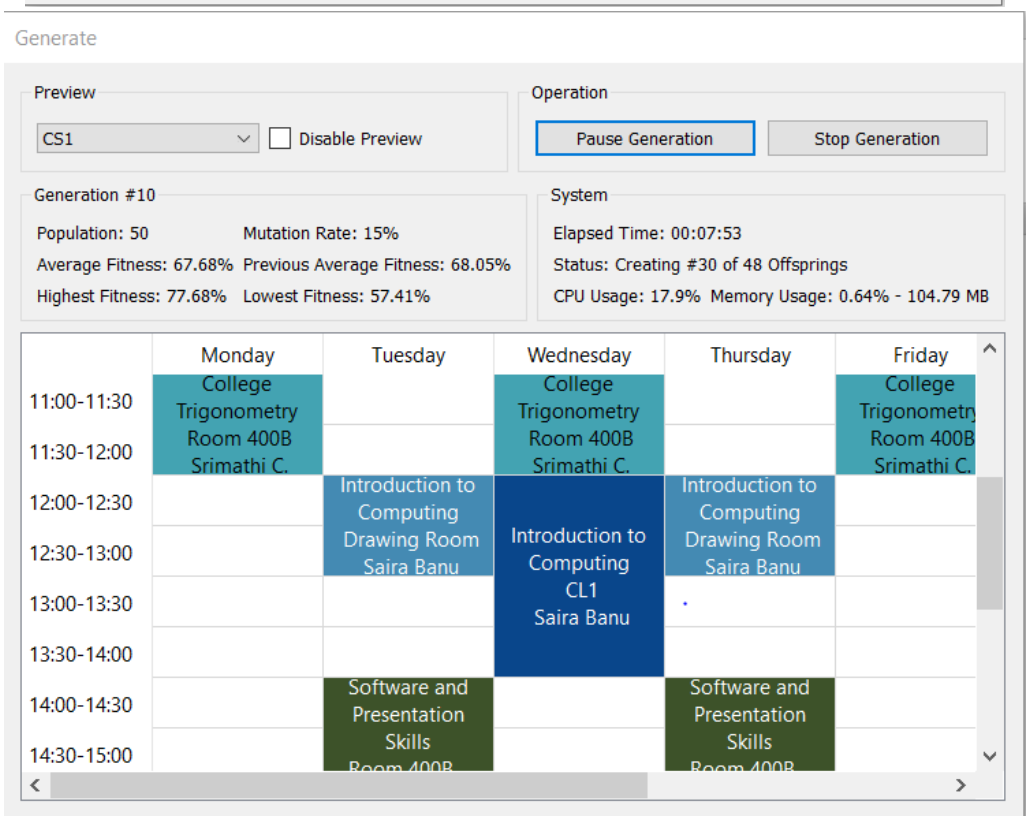
Scenario Manager:

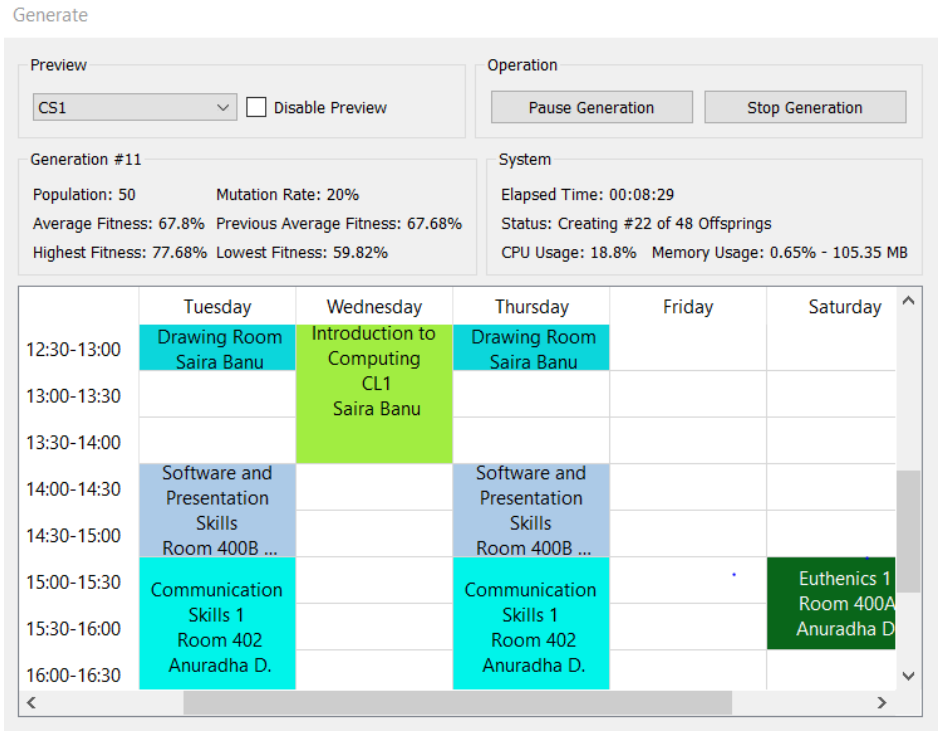


O

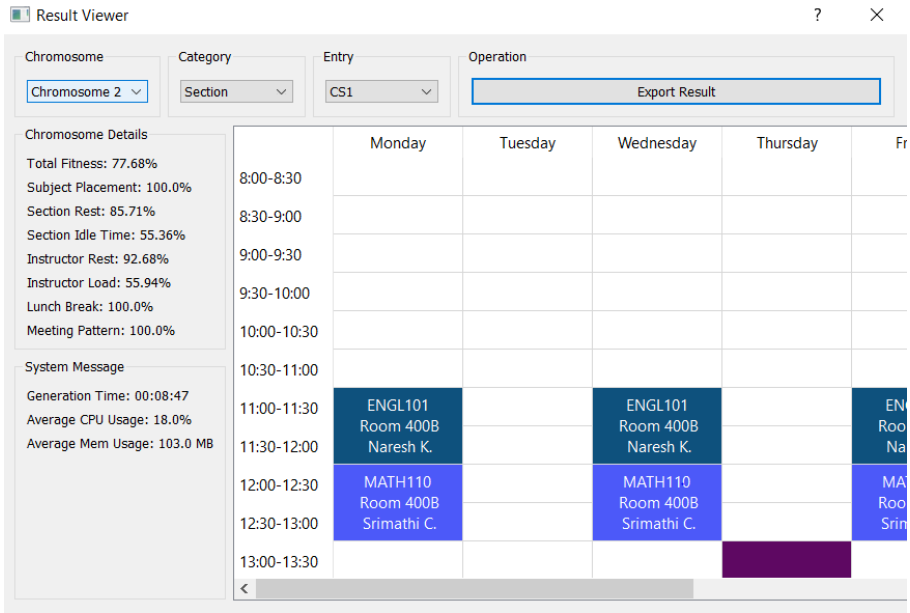
OUTPUT:

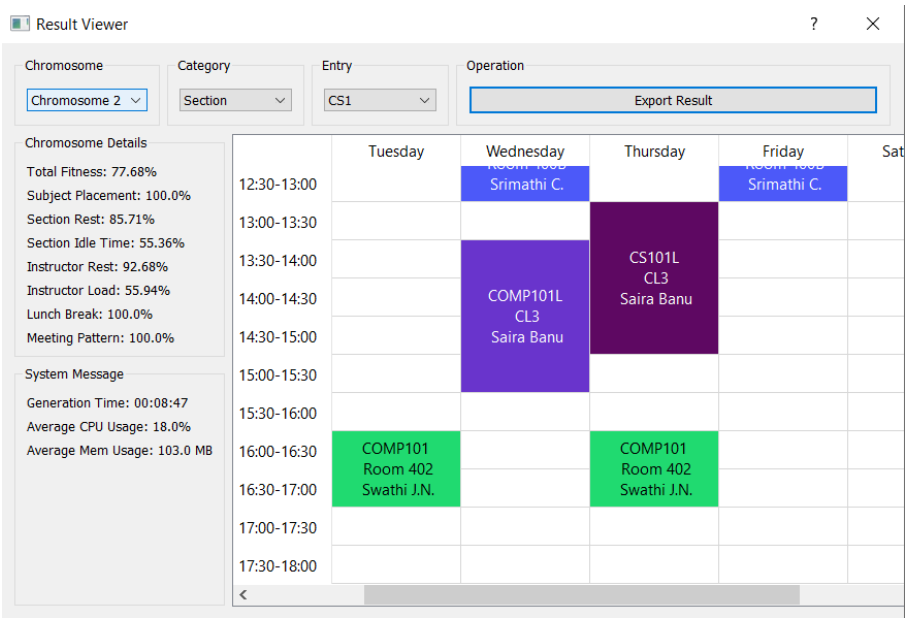

Generate
Preview
CS1
Disable Preview
Operation
Pause Generation
Stop Generation
Generation #9
Population: 50
Mutation Rate: 20%
Average Fitness: 68.05%
Previous Average Fitness: 67.31%
Highest Fitness: 77.68%
Lowest Fitness: 60.0%
System
Elapsed Time: 00:07:05
Status: Creating #28 of 48 Offsprings
CPU Usage: 15.3%
Memory Usage: 0.64% - 104.68 MB
Monday
Tuesday
Wednesday
Thursday
Friday
8:00-8:30
8:30-9:00
9:00-9:30
9:30-10:00
10:00-10:30
10:30-11:00
11:00-11:30
11:30-12:00
Knowledge Work
Software and
Presentation
Skills
CL3
Saira Banu
College
Trigonometry
Room 400B


Generate
Preview
CS1
Disable Preview
Operation
Pause Generation
Stop Generation
Generation #10
Population: 50
Mutation Rate: 15%
Average Fitness: 67.68%
Previous Average Fitness: 68.05%
Highest Fitness: 77.68%
Lowest Fitness: 57.41%
System
Elapsed Time: 00:07:53
Status: Creating #30 of 48 Offsprings
CPU Usage: 17.9%
Memory Usage: 0.64% - 104.79 MB
Monday
Tuesday
Wednesday
Thursday
Friday
11:00-11:30
11:30-12:00
12:00-12:30
12:30-13:00
13:00-13:30
13:30-14:00
14:00-14:30
14:30-15:00
College
Trigonometry
Room 400B
Srimathi C.
Introduction to
Computing
Drawing Room
Saira Banu
Introduction to
Computing
CL1
Saira Banu
Software and
Presentation
Skills



Chromosome 2:





Conclusion

The usage of models was meant to test the capabilities of the artificial intelligence. Judging by the performance of results compared to models, the system was able to generate solutions that have at least 80% fitness (basing on set of highest solutions per scenario). The problem lies on the inability of artificial intelligence to perform minor corrections on adjusting schedules (see generation results category “C”). Nevertheless, the algorithm was able to cater for majority of the entries on the evaluation matrix.

The results have shown that the system can provide valid solutions that can be used. However, it does not provide complete automation. There are still scenarios that would require the operator to adjust some entries to create a perfect solution.

The system was also designed to be simple and straightforward. This eliminates any confusion caused by scattered user interface controls and makes usage of the software fully utilized.

Limitations

The large amount of combinations for testing in order to find an accurate evaluation for the application has proven to be far from possibility. However, it can be concluded that from the

conditions provided, the system was able to generate results that despite being imperfect still remains valid and acceptable given the number of constraints imposed to it.

· The solutions that the system will provide will heavily depend on the running configuration and evaluation matrix.

The complete evaluation for the system will remain hard to solve as the freedom for the configuration of the algorithm has provided a large amount of combinations.

Future Work

- Though this web-app serves as a basic time table generator, there is a lot more which could be done to make this project even better in terms of consideration of soft constraints like professor giving preference to particular class.

- The up-gradations I look up to currently will be Classroom size considerations, lab facility consideration and multiple subject selection for faculty. I will try to bring the following up-gradations very soon.

- More features such as schedule print for individual faculty etc. would also be involved to make this more useful as a final product.

References

- http://www.javatpoint.com
- http://www.ai-junkie.com/ga/intro/gat3.html
- http://www.obitko.com/tutorials/genetic-algorithms/encoding.php
- https://en.wikipedia.org/wiki/Genetic_algorithm
- https://www.researchgate.net
- Automatic Timetable Generation using Genetic Algorithm-International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 2, February 2015