Development of an Automated Class Schedule Allocation Method

By

011181077	**Tanvir Hasan Emon**
011181090	**Partho Protime Sarker**
011181097	**Efat Ahmed Shohel**
011182104	**Joyosree Acharjee Tithi**
011182051	**Tahmina Tanjin Sharna**
011181187	**Farhana Afroz**

Submitted in partial fulfilment of the requirements
of the degree of Bachelor of Science in Computer Science and Engineering

July 19, 2022

Department of Computer Science and Engineering
United International University

Abstract

Course selection and section selection is a major problem for open credit universities like United International University. In the time of registration for next trimester the students have to face a lot of trouble. They need to choose their course and section on their own and most of the time they can not make a suitable class routine for them. This paper proposes a new approach to solve this problem which is an automated class schedule allocation method. The system consists of two parts. First part is about collecting necessary data. Our system needs taken course data from students and section list of offered courses from the university authority. The second part is our system, which is responsible for the optimization of the students class routine based on students selected courses and the university authority provided section list for those courses.Here we will try to resolves some constraints like reducing time gap between classes, day class or morning class preferences. Students can choose their preference while generating the class schedule. Our initial objective is to provide an effective and useful system which can generate an optimal class routine for students.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the whole project in a systematic and brief manner. This chapter explains project overview, motivation, objectives, methodology and project outcomes as well. It will give a proper overall understanding of the project.

1.1 Project Overview

Students of different trimesters need to select courses of their own choice in an open credit university. They need to choose their selections against each course on their own from the university provided section list, the process is known as "Advising". It is very burdensome and also stressful for students to create a suitable class routine manually. Most of the time, two or more classes conflict with one another, which makes it impossible to register for both and also the universities have to develop a fixed schedule for courses with limited capacity. As a result, students can not take their desired section, and sometimes they even have to drop the course. To reduce the complexity of the advising, we will develop Automated Class Schedule Allocation method. Our system will analyze collected data and will generate an optimal class schedule based on student time preference.

1.2 Motivation

We began this project with the goal to provide a solution that will solve the difficulties a student face during advising process. We have analyzed an automated class scheduling system can solve those kinds of problem. From that thought we decided that we need to do something about it. Our proposed system makes section selection very much easy and efficient as compared to manual system. It can also save a lot of valuable time of our students.

1.3 Objectives

The primary objectives of our project are :

- To develop an automated class schedule method that will analyze data and will provide a optimal class schedule based on students time preference.
- To make our system efficient that the run-time of our system will be very low so that it can generate result in a short amount of time.
- To design an automated class scheduling system and test its functionality in terms of accuracy, data handling, stability and adaptability in making class schedules.

1.4 Methodology

Students pre-advising data and university offered course and section routine will be used as input. We have analyzed different algorithms like Genetic Algorithm, Evolutionary Algorithm, Ant Colony Algorithm, Heuristic Approach which are used to solve schedule generating problems. After analyzing these algorithms we will try to develop our own algorithm that will generate optimal class routine. Our system will provide all possible conflict free routines. Among those routines students will be able choose their preferred one, they will even be able to apply their time preference while choosing the routine. Our offered schedule will also calculate the time gap between classes, so one can get the idea of how much leisure time he/she is going to spend in a day in the campus.

1.5 Project Outcome

Our goal is to create an system that will provide an automated class schedule/routine for our students based on their preferable time. Our system will reduce those problems that students are facing for years during registration time. Our system will be able to generate routine automatically based on learning data but if any student wants to select sections manually he/she also can do that as well. So our project outcome will be very much useful to students and as well as university authority too.

1.6 Organization of the Report

- Chapter 1 of this report provides a brief introduction of the project by talking about project overview, motivation, objective, methodology and project outcome.
- Chapter 2 of this report provides background knowledge, literature review, related research and gap analysis of those papers.
- Chapter 3 of this report provides a overview of functional and non-functional requirements, context and data flow diagram of the project, demo ui design, project plan and task allocation.
- Chapter 4 of this report provides knowledge about our project environment, testing, evaluation and results.

- Chapter 5 of this report provides a knowledge about compliance with the standards, design constraints, cost analysis and complex engineering problem.
- Chapter 6 of this report provides knowledge about project summary, limitations and future work.

Chapter 2

Background

This chapter will give readers the background knowledge needed for our project. It also gives readers a brief idea about related papers to understand the problem.

2.1 Preliminaries

For understanding this project and report, we will keep some knowledge about some topics and subjects. For better understanding we will also discuss about them :

Scheduling

In the computer system, scheduling is the act of allocating resources for performance. Here, resources mean computer processor, random access memory (RAM), networks links etc. Task can be threads, processes or data flows. Schedule activities are performed by some systematic ways called scheduler.

Genetic Algorithm

A genetic algorithm (GA) is a type of heuristic search method that is used to address problems involving search and optimization. Because they employ previous data to direct the search to the best performing region within the solution space, these algorithms are more intelligent than random search algorithms.

Web Course Registration

Our Online Student Record has a link to Web Registration. Before our registration appointment, you can pre-select and save the courses and sections we want to take. It's also where we will register for classes whenever our registration appointment time comes around. We can utilize Web Registration to add or drop classes after we've registered for a semester.

Evolutionary Algorithm

An evolutionary algorithm is a subset of evolutionary computation, a generic population-based metaheuristic optimization method in artificial intelligence. EAs use biological evolution-inspired mechanisms like reproduction, mutation, recombination, and selection.

2.2 Literature Review

A literature review is an overview of the previously published works on a specific topic. Here is the literature review of our project :

2.2.1 Similar Applications

Different universities uses different kinds of process for their advising and registration. Some of them are listed here as similar applications :

Brac University

In this university, suppose they have 100 students per course. So they create some group of around 20 students. The university authority gives a section and course in the first group and its running like recursion. When they reach the last group many sections have booked and the students can not take the courses. But if they see there are around at least 10 students are remaining they open a new section that stays at the same time in previous sections. Otherwise, they drop the course. But the main problem is their server become also slow when the advising start.[1]

North South University

In this university students can take course before selecting the sections. After selecting course then they can select their sections but in the process of selecting sections their server became very slow.[2]

East West University

In this university they publish section list and course list at the same time while advising. The students are permitted to take course and section between a limited amount of time. If they somehow fail to take course in that time then the course will be drop.[3]

American International University of Bangladesh

In that university they provide advantage to those students who has completed more credits to select course and section. After that they gives course and section for rest of the students. In that result for those who has completed less credits they face to take course and sometimes they need to drop the course as well.[4]

2.2.2 Related Research

We have read and analyze some papers those are related to our project. Those papers are categorised based on the used algorithm :

Genetic Algorithm

In this paper [5], they present an ILP-based approach to generating university course schedules. it show how to formulate the UCSP as an ILP problem and study the possibility of solving the UCSP using (i) advanced SAT-based 0-1 ILP algorithms and (ii) generic-based ILP algorithms. It compare the performance of both algorithms and provide empirical results showing that generic-based ILP solvers tend to outperform SAT-based ILP solvers. The ILP models are solved using advanced genericbased and Boolean satisfiability based ILP solvers. The goal is to find a schedule that satisfies the university's rules, yet optimizes the use of the existing facilities such as minimizing the capacity to enrollment ratio in a class. The approach was tested on different cases with various sizes and showed promising results. The approach is complete and will find the best possible schedule, or will indicate that no schedule exists that meets the current university rules.

In this paper [6], they try to manage class room,lecture and time to get an optimized schedule. And they try to solve this problem using genetic algorithm by building a library.It gives a good optimised schedule. They use DVM and UYB for find out a solution for scheduling problem. UYB case gives a fastest optimized solution.For interchange mutation and DVM they apply on two mutation method to find out which is best and the result is DVM is better then interchange mutation.It is appropriate for optimal solution. Lecturing schedule computation of UBY is basically for afternoon expect study program.They try to make a advantage and disadvantage table for make sure is library has good efficiency or not.

The significance of this paper [7] is to design a scheduling algorithm based on improving the existing artificial intelligence algorithm and for the scheduling problem, which applies to the current scheduling principles and can not only make reasonable arrangements for large-scale scheduling data but also improve the efficiency of the scheduling staff and reduce the redundant workload. Therefore, the use of computer scheduling has the advantages of saving time and labor and high quality and effectively reduces the tedious scheduling tasks, which plays a crucial role in the information and intelligent construction and development of universities. Because of the stochastic nature of genetic algorithms, the conflict of constraints arising from scheduling should be detected in time, and the constraints are interrelated, so the conflict characteristics should be fully considered and solved; many Complexity researchers study the scheduling problem only based on the single schedule of the course but fail to relate the scheduling problem to the scheduling problem in actual teachings, such as the scheduling of different times and classrooms each week for courses with different weekly credit hours and the arrangement of the beginning and end weeks of teaching.

In this paper [8], they worked with small instance of timetable problem which required scheduling 100 events in 45 time slots. To solve this they used tournament selection methods. Tournament selection is a method of selecting an individual from a population of individuals in a Genetic Algorithm. They used tournament selection II and tournament selection V to check the quality of offspring generated. They found considerable improvement in solution generated with tournament selection V. As stated genetic Algorithm starts with initializing the population with some random solutions. After initialization we evaluated these random solutions to determine the survival capacity of solution. Selection is an important activity used here to select proper parents to generate an individual called offspring which improves probability of survival of good solution.

Evolutionary Algorithm

In this paper [9], we can see in Alexander Technological Educational Institute of Thessaloniki they have some problem in course registration system. So, they aim to solve this and this paper present the optimized course registration system. They maintain their registration system based on web. First of all, they allow students to submit their course preference and they store this course preference data into a database. They have administration who have the control power of the system and they use some administrative tools for their operation. Then they use evolutionary algorithm in those collected data to provide optimized course schedule. Their result is very much satisfying. Their system can allocate most of the places and the percentage of higher than 97 percent. They also have request served rate is more than 90 percent.

This paper [10] focuses on the issues of time table of course (curriculum) scheduling at the university level. Students and faculty members face many problems when preparing courses and exam plans. The goals of this work are to create a suitable timetable system for the required courses and to find feasible solutions under many constraints. The primary step to achieve the goals mentioned above is to discover the problems faced by faculty and staff when preparing the plan, collect relevant information about the problems to analyze it. In this paper they have compared output from Genetic algorithm with the Memetic algorithm also, which is basically an extension of the GA. It uses a local search technique to reduce the likelihood of premature convergence. After the comparison it can be seen easily that GA and MA are both capable of approaching the level where no collision occurred. However, in case of MA, when there is no clash, the evolution happened very slowly, which can most likely be due to a local limit, as a result of which it has to wait for a suitable mutation. While in a GA, after they achieved a no-clash timetable, the evolution continued at the same rate, and thee are always capable of obtaining a more acceptable mutation.

Tabu Search Algorithm

In this paper [11], they try to generate course and exm schedule by using Tabu Search Algorithm.There have some constraint,and they try to divided these problems into two types.Most important is timetabling problem, and they use some techniques to solve this problem.There are Tabu Search, Simulated Annealing, Ant Colony Optimization etc. They try to find a favourable solution and minimize the constraint problem. Like hard constraint, soft constraint. They have a goal,which is make course and exam schedule within a short time by using Tabu Search.

Ant Colony Optimization

In this paper [12], The faculty of management of Universitas Komputer Indonesia has a problem about arrangement of schedule.They has a request for lecturer and lack of classroom which has impact on other.For solving this problem, the researchers mentioned a solution to get optimal collision of schedule by using some variables. They are using Ant colony Algorithm to solve this problem. By using Ant colony Algorithm, research has made efficient and appropriate process and time reduction. The performance of ant colony Algorithm is given scheduling process more proficient and considered about impact on schedule. Process scheduling at Universitas Komputer Indonesia has regulated in optimization scheduling has always been following by regulation stated.Whenever management was changed,it means that the process scheduling would be changed too.

Heuristic Approach

In this paper [13], Anirudha Nanda, Manisha P. Pai, and Abhijeet Gole developed an algorithm using heuristic approach to solve the school time tabling problem . Their proposed algorithm solve the problem solves the problem while giving importance to teachers availability for given time slot. This algorithm takes a number of subjects, number of teachers, subjects every teacher takes, number of days in a week for which the timetable needs to be set, number of time slots in a day and the maximum lectures a teacher can conduct in a week as user input. It initially uses randomly generated subject sequence without any repetition of subjects over a day to make a temporary time table. After this, the teacher availability is checked for each of the subjects allocated for the respective slot. Every time a teacher is available for the subject at the allocated slot, the subject and the teacher are entered into the output data structure and marked as final. Before the allocation of this subject to the output data structure, it checks the number of maximum lectures a teacher can conduct. If the teacher has been allocated more than the allowed maximum lectures, the subject is moved into a Clash data structure. To resolve this clash data, this data structure revisited to allocate the subjects in it to an available time slot in the day and if it is not possible to allocate the clash subject, subjects are moved to the Day-Clash data structure. When sequence for the next day is generated preference is given to the subjects under Day-Clash. The output of the implemented algorithm gen-

erates a time slot. In each time table, all time-slot is filled with, a teacher and a unique combination of subjects without any repetition of subjects. This algorithm incorporates a number of techniques, aimed to improve the efficiency of the search operation. Given the generality of the algorithm operation, it can further be adapted to more specific scenarios, e.g. University, examination scheduling and further be enhanced to create railway time tables.

Aprori Mining Algorithm

In this paper [14], we can see Jordan University of Science and Technology was facing course timetabling problem. To solve this timetable problem, they proposed some different approaches. Here they collect instructors' preference data and some other constrains to generate a schedule which will be conflict free. They collect those data from the computer science department and other related departments in Jordan University of Science and Technology. In this paper [14] they finally solve the timetable problem and get a satisfactory result. They use Aprori mining algorithm to solve this problem. They also work with some constrains that will identify the task priorities. Their system will reduce the effort and time for the university authority those are related to this system. They have only tested their system in Computer Science department courses but they are highly expecting that their system will perform well in other department courses as well. They also said that they can achieve better performance by using student's preferences.

2.3 Gap Analysis

We have read and analyze some papers related to our project and we found those gaps in them :

- In paper [13, 14], they solved the problem from teacher's point of view/preference, but if we consider students point of view/preference as well, the system will be more efficient.
- In paper [14], we can see that no. of sections after generated schedule is less when it is done by the system than the no. of generated sections when it is done manually.
- In paper [9, 14], they only focused on generating conflicting free lecture schedules, but they did not consider any other parameters to optimize the scheduling problem like reducing the time gap between two classes or any kind of affinity.
- Paper [13], does not have sufficient information or data to support their progress.

2.4 Summary

Now this chapter gives the proper view of background work. Each topics describe properly with description about theirs tasks. This chapter cleans all the doubts and present the

accurate working procedure of our project.

Chapter 3

Project Design

Good design thinking plays a very important role in every successful project. From this chapter, readers will get an idea about how we implemented our project step by step.

3.1 Requirement Analysis

Requirements analysis is an interactive process in which the expectations and needs of each stakeholder are explored. Analyzing the requirement helps the developer to understand stakeholder demands and facilitate system design. Since our goal is to develop an automated class generating system that provides an optimal class routine while giving importance to students time preference. So to analyze the requirements of our project, we have talked to our university students personally and asked about their expectations and need from our system. We also reviewed some papers related to our project and analyzed those paper and tried to find gaps. After analyzing those papers and students requirements we have listed our project requirements in two parts Functional and Non-Functional. The Functional and Non-functional Requirements of our project are mentioned in sub section 3.1.1 .

3.1.1 Functional and Nonfunctional Requirements

Functional Requirements:

- Generating optimal class routine
- Reducing Time gap
- Generating routine while focusing students time preferences like day class or morning class.

Our main functional requirement is generating an optimal class routine for each student. For this it will require the student's data and university provided timetable or section list. Our second functional requirement is reducing time gap between classes and third one is to provide a routine focusing students time preferences.

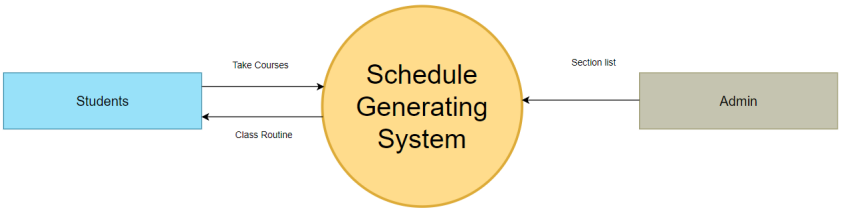

Figure 3.1: Context Diagram

Nonfunctional Requirements:

- Performance
- Storage
- Reliability

Our first non-functional requirement is minimizing the processing time of the system for better performance. Second is ensuring that our system uses less amount of storage. Another non-functional requirement is ensuring the reliability of the system by providing optimal class routine.

3.1.2 Context Diagram

In figure 3.1, we can see the context diagram of our project. In our system, there will be two kinds of users one is students another one is admin. Students will take course and our system will take the selected course data. Then admin will provide section list data and our system will collect this data as well. Then our Scheduling generating system will provide a optimal class routine to the students.

3.1.3 Data Flow Diagram Level 1

In figure 3.2, we can see the data flow diagram of our project. From this diagram we can see that we have two external entity those are student and university authority. We have three process those are Course Selection System, Create section list, class schedule allocation system. First of all student will take course in Course selection system then our system will collect those data. Then University authority will provide section description in create section list process and our system will collect those data as well. After that our system will analyze course data and section data and provide a optimized class routine to the student.

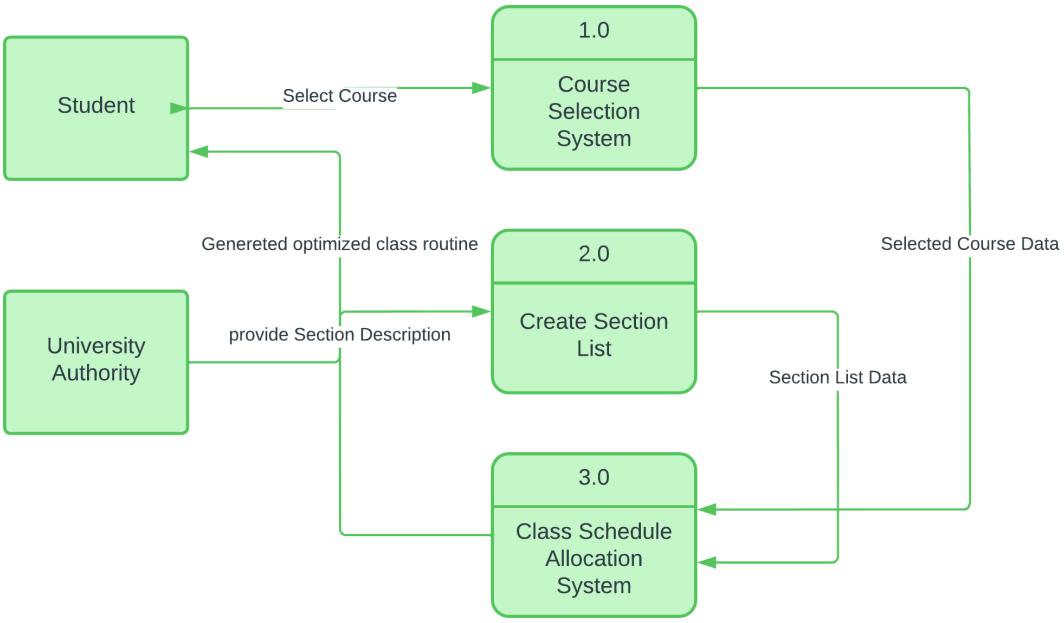

Figure 3.2: Data Flow Diagram Level 1

3.1.4 UI Design

Figure 4.1, figure 3.4 and 3.5 are the demo visualization of the user interface of our project. In that figure 4.1, we will collect data from user. Here user can enter how many courses and which courses they are want to take. They can also enter there time preference for day time class or morning time class. After taking those input our algorithm will provide suggested schedule which will be conflict free schedule that student can take. In figure 3.4, we can see the process. After that our system will generate another schedule which will be preferable schedule. In figure 3.5 we can see the process.

3.2 Detailed Methodology and Design

Here we have tried to explain how the algorithm of our project works through a flowchart 3.6. To use this system, initially user has to input the number of courses he/she wants to take, the selected courses and time preference for the course schedule. After taking all the inputs,the system reads the selected course from a section list initially imported in the system and brings all the section assigned against the selected courses and stores. Then we used an algorithms to generate the all possible conflicts free optimized schedule. The algorithm generates all possible combination of available sections of each course.First it creates a routine and then checks whether the day conflicts among the selected sections and if the day conflicts it discard that particular routine and choose the next routine. But if the day does not conflict then it applies another condition and checks whether the time conflict. If the time conflicts, it discard that routine and choose the next routine. But if

```
Enter how many course you want to take = 7
Enter course Code = CSE 465/CSE 4165
Enter course Code = CSE 481/CSE 4181
Enter course Code = CSI 321/CSE 3421
Enter course Code = CSE 3812/CSI 342
Enter course Code = CSI 421/ CSE 4521/     CSE 4621
Enter course Code = CSI 422
Enter course Code = IPE 3401/IPE 401

1.Day Class
2.Morning Class
Enter your preferrence = 1
```

Figure 3.3: Demo UI Input

```
Suggested Schedule :
Suggested Schedule  1  :
['EEE 2113/ CSE 113' 'Electrical Circuits' 'A' 'Sat T' '11:40 AM - 1:10 PM']
['EEE 2123/ CSE 123' 'Electronics' 'B' 'S W' '10:05 AM - 11:35 AM']
['EEE 2124/ CSE 124' 'Electronics Lab' 'B' 'T' '8:30 AM - 11:00 AM']
['EEE 4261' 'Green Computing' 'A' 'S W' '11:40 AM - 1:10 PM']
['PHY 2105/ PHY 105' 'Physics' 'A' 'S W' '1:30 PM - 3:00 PM']
['PHY 2106 /PHY 106' 'Physics Lab' 'E' 'Sat' '8:30 AM - 11:00 AM']
Time Gap Of Schedule  1  :
Total Time Gap =  2  Hour  10 Minutes In One Week.
Saturday Time Gap =  40  Minutes.
Sunday Time Gap =  25  Minutes.
Tuesday Time Gap =  40  Minutes.
Wednesday Time Gap =  25  Minutes.
```

Figure 3.4: Demo UI Suggested Schedule Output

```
Your Preferable Schedule :
Your Day Class Schedule  1  :
['EEE 2113/ CSE 113' 'Electrical Circuits' 'C' 'S W' '10:05 AM - 11:35 AM']
['EEE 2123/ CSE 123' 'Electronics' 'D' 'Sat T' '10:05 AM - 11:35 AM']
['EEE 2124/ CSE 124' 'Electronics Lab' 'E' 'W' '2:00 PM - 4:30 PM']
['EEE 4261' 'Green Computing' 'A' 'S W' '11:40 AM - 1:10 PM']
['PHY 2105/ PHY 105' 'Physics' 'E' 'Sat T' '11:40 AM - 1:10 PM']
['PHY 2106 /PHY 106' 'Physics Lab' 'F' 'Sat' '2:00 PM - 4:30 PM']
Time Gap Of Day Schedule  1  :
Total Time Gap =  2  Hour  0 Minutes In One Week.
Saturday Time Gap =  55  Minutes.
Sunday Time Gap =  5  Minutes.
Tuesday Time Gap =  5  Minutes.
Wednesday Time Gap =  55  Minutes.
```

Figure 3.5: Demo UI Preferable Schedule Output

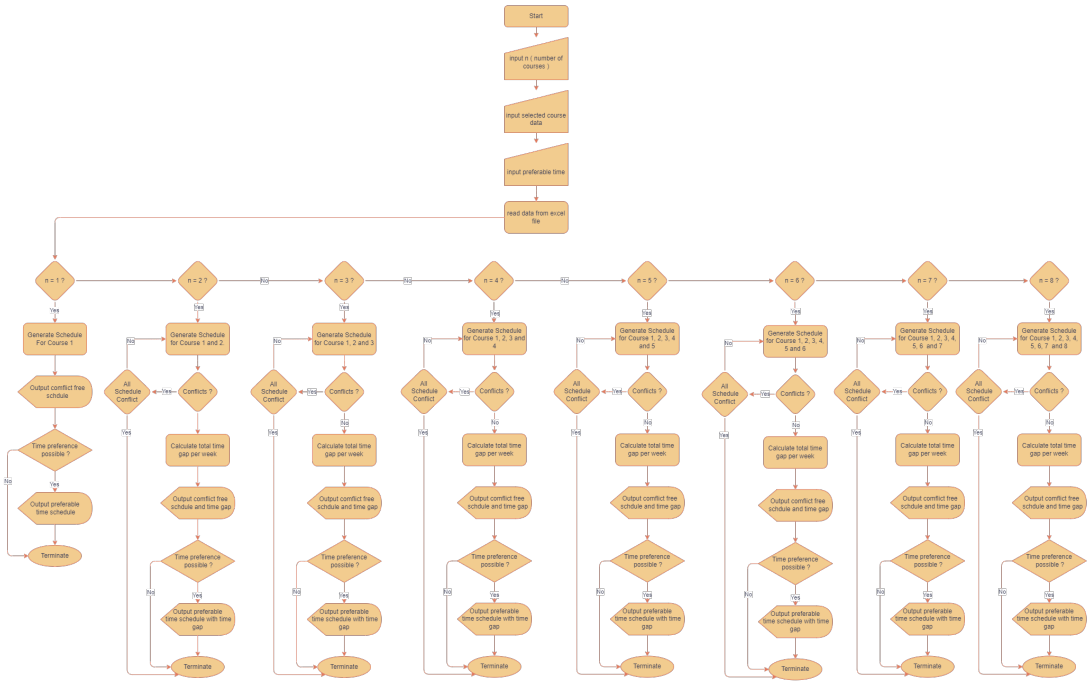

Figure 3.6: Flowchart Of Automated Class Scheduling System

the time does not conflict, the routine is selected as a possible output and it calculate the overall leisure time of the whole week for that particular schedule/routine. After checking all the possible combination of generated routines , the all possible conflicts free schedule, calculated leisure time and his/her preferred time routines are shown to the users as final output.

3.3 Project Plan

Figure 3.7 and figure 3.8 are the project plan of our project. From those figures we can see there are some tasks name and task deadline. From figure 3.7 we can see we have some tasks like literature review, gap analysis, methodology, report writing etc. We blocked the time limit for those tasks so that we can finish those tasks in time. In figure 3.8 we have some other tasks like database design, ui and ux development, algorithm development, testing and deployment. We allocation some specific time for those tasks as well. From those two figures readers can have a brief idea about our project plan.

3.4 Task Allocation

We have total six members. Here is the task allocation of our project:

- **Tanvir Hasan Emon :** Algorithm Development, Implementation, Testing, Literature Review, Report Writing.

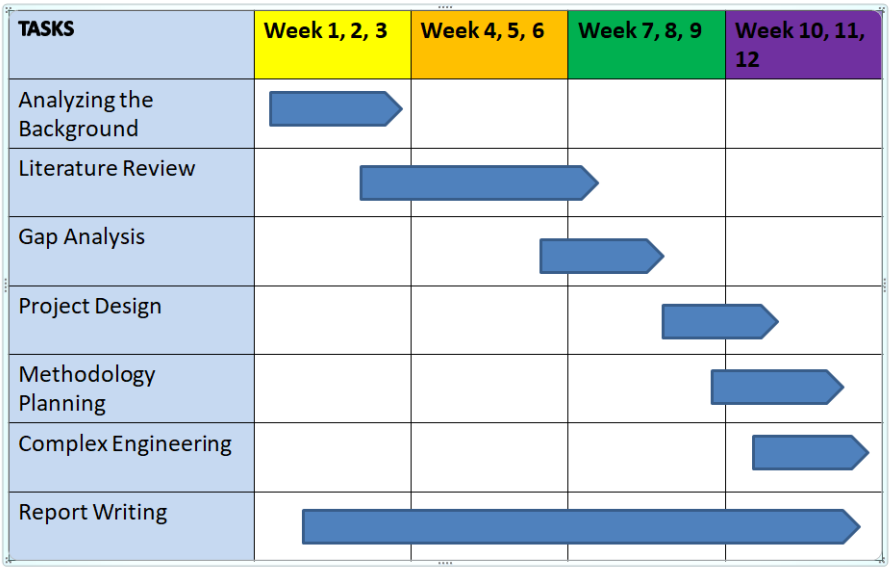

Figure 3.7: Project Plan 1

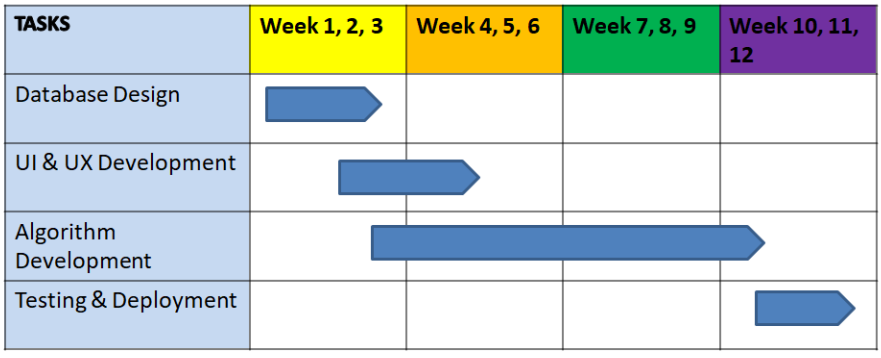

Figure 3.8: Project Plan 2

- **Farhana Afroz :** Algorithm Development, Implementation, Testing, Literature Review, Report Writing.
- **Partho Protime Sarker :** Data Prepossessing, Testing, Literature Review, Report Writing.
- **Efat Ahmed Shohel :** Presentation Slide, Literature Review, Report Writing.
- **Joyosree Acharjee Tithi :** Presentation Slide, Literature Review, Report Writing.
- **Tahmina Tanjin Sharna :** Literature Review, Report Writing, Presentation Slide.

3.5 Summary

In this chapter we discussed about our project requirements include functional requirements and non functional requirements. We also discussed about Context diagram, Data flow diagram, UI design, project plan and task allocation. This chapter Will help reader to have better understanding about our project.

Chapter 4

Implementation and Results

In this chapter, readers will have a brief knowledge about the implementation and results of our project. Here we talked about our algorithm, result, project environment setup, testing and result evaluation.

4.1 Environment Setup

In that section, we talked about our environment setup.

4.1.1 Python Programming Language

Python is a object oriented and high level programming language. For our project, we installed Python 3.10 in our device. We have done our implementation using Python because it is easy to use,understand and it has vast amount of libraries.In our project we used NumPy library.

4.1.2 Google Colaboratory

Google Colaboratory is a virtual environment where we can write and execute code through browser. For using colaboratory, it does not require any predefined setup. It can work its full functionality in a browser using google storage and virtual ram. In our project we use Google Colaboratory as our primary IDE.

4.1.3 Google Drive

Google Drive is a virtual storage system. We can store our files there and use them when we need to. In our project we use Google Drive to store our section list data and to import it in our implementation.

4.2 Testing and Evaluation

In this section we talked about our input, algorithm, algorithm evaluation and testing.

```
Enter how many course you want to take = 7
Enter course Code = CSE 465/CSE 4165
Enter course Code = CSE 481/CSE 4181
Enter course Code = CSI 321/CSE 3421
Enter course Code = CSE 3812/CSI 342
Enter course Code = CSI 421/ CSE 4521/     CSE 4621
Enter course Code = CSI 422
Enter course Code = IPE 3401/IPE 401

1.Day Class
2.Morning Class
Enter your preferrence = 1
```

Figure 4.1: User Provided Input

4.2.1 Input

Here we talked about our input.

User Provided Input

Here we will take some data from user. First of all, user will provide how many courses he/ she wants to take. After that user will provide which courses they are going to take. After that user can also provide there time preference. They can choose day time class schedule or morning time class schedule. In figure 4.1, we can see the process.

University Provided Section Information

We import university provided section information in our implementation. It has course code, course name, course section, course day and time. Those are essential information in our project to provide an optimal class schedule. In figure 4.2, we can see university provided section information file.

4.2.2 Algorithm Development

Here we talked about our implemented algorithm. Our implementation can provide optimal class schedule between 1 to 8 courses. If user takes 1 course then we don't need to apply any algorithm. We will provide direct class for them.

2 Courses

If user wants to take 2 courses c1 and c2. Then our Algorithm will provide the solution. Here is the developed algorithm for 2 courses :

	A	B	C	D	E
1	**Course Code**	**Title**	**Section**	**Day**	**Time**
2	ACT 2111/ACT 111	Financial and Managerial Accounting	A	Sat T	8:30 AM - 10:00 AM
3	ACT 2111/ACT 111	Financial and Managerial Accounting	B	S W	8:30 AM - 10:00 AM
4	BDS 1201	History of the Emergence of Bangladesh(2nd Tri)	A	Sat T	11:40 AM - 1:10 PM
5	BDS 1201	History of the Emergence of Bangladesh(2nd Tri)	B	Sat T	1:30 PM - 3:00 PM
6	BDS 1201	History of the Emergence of Bangladesh(2nd Tri)	C	S W	11:40 AM - 1:10 PM
7	BDS 1201	History of the Emergence of Bangladesh(2nd Tri)	D	S W	1:30 PM - 3:00 PM
8	BDS 1201	History of the Emergence of Bangladesh(2nd Tri)	E	S W	8:30 AM - 10:00 AM
9	BDS 1201	History of the Emergence of Bangladesh(2nd Tri)	F	Sat T	8:30 AM - 10:00 AM
10	BDS 1201	History of the Emergence of Bangladesh(2nd Tri)	G	S W	1:30 PM - 3:00 PM
11	BDS 1201	History of the Emergence of Bangladesh(2nd Tri)	H	Sat T	1:30 PM - 3:00 PM
12	BDS 1201	History of the Emergence of Bangladesh(2nd Tri)	I	Sat T	10:05 AM - 11:35 AM
13	BDS 1201	History of the Emergence of Bangladesh(2nd Tri)	J	S W	10:05 AM - 11:35 AM
14	BIO 3105	Biology for Engineers	A	S W	11:40 AM - 1:10 PM
15	BIO 3105	Biology for Engineers	B	Sat T	11:40 AM - 1:10 PM
16	BIO 3105	Biology for Engineers	C	Sat T	10:05 AM - 11:35 AM

Figure 4.2: University Provided Section Information

```
For i in range(0,len(c1)) :
    For j in range(0,len(c2):
        If c1 and c2 both theory :
            If no day and time conflict :
                result
        Elif c1 and c2 both lab :
            If no day and time conflict :
                Result
        Else : ( 1 theory 1 lab )
            If c1 theory c2 lab :
                If no day and time conflict :
                    Result
            Elif c1 lab c2 theory :
                If no day and time conflict :
                    Result
```

After providing conflict free schedule we also provide some schedule based on user time preferences. Suppose user want day class then there will be no class in 8.30 A.M. in their preferred schedule. On the other hand, if user want to take morning class then there will be no class after 1.30 P.M.

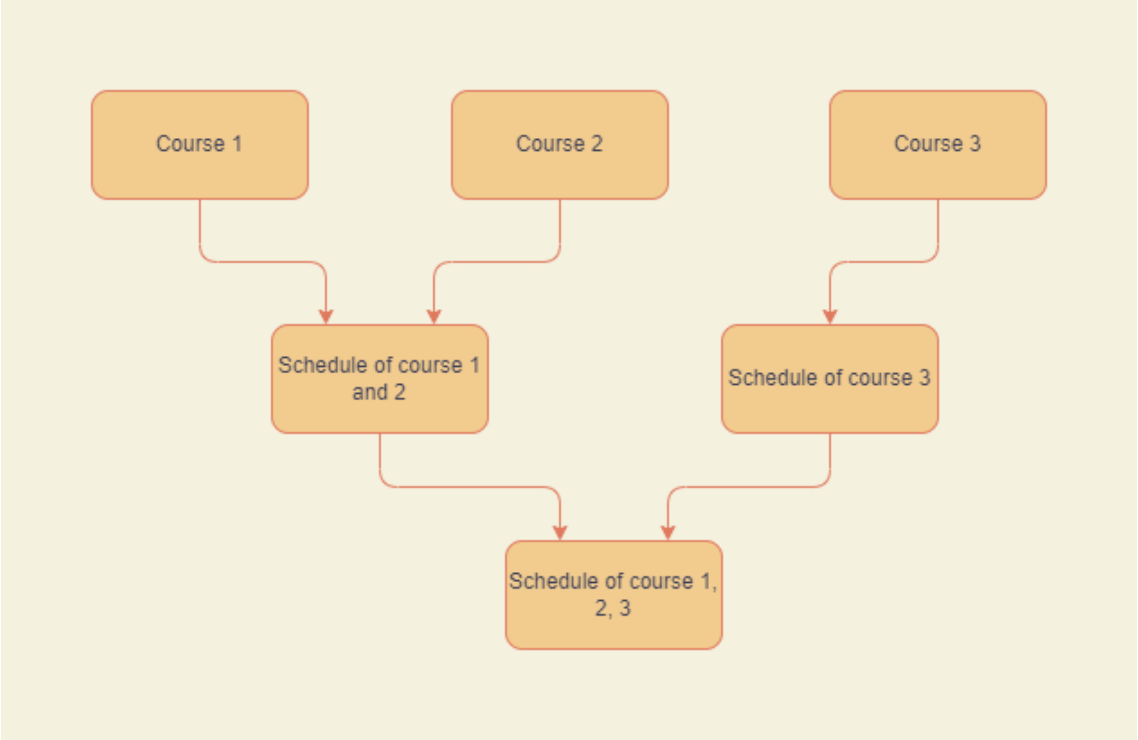

Figure 4.3: Stages of Algorithm for 3 courses

3 Courses

In figure 4.3, we can see the stages of algorithm for 3 courses.

4 Courses

In figure 4.4, we can see the stages of algorithm for 4 courses.

5 Courses

In figure 4.5, we can see the stages of algorithm for 5 courses.

6 Courses

In figure 4.6, we can see the stages of algorithm for 6 courses.

7 Courses

In figure 4.7, we can see the stages of algorithm for 7 courses.

8 Courses

In figure 4.8, we can see the stages of algorithm for 8 courses.

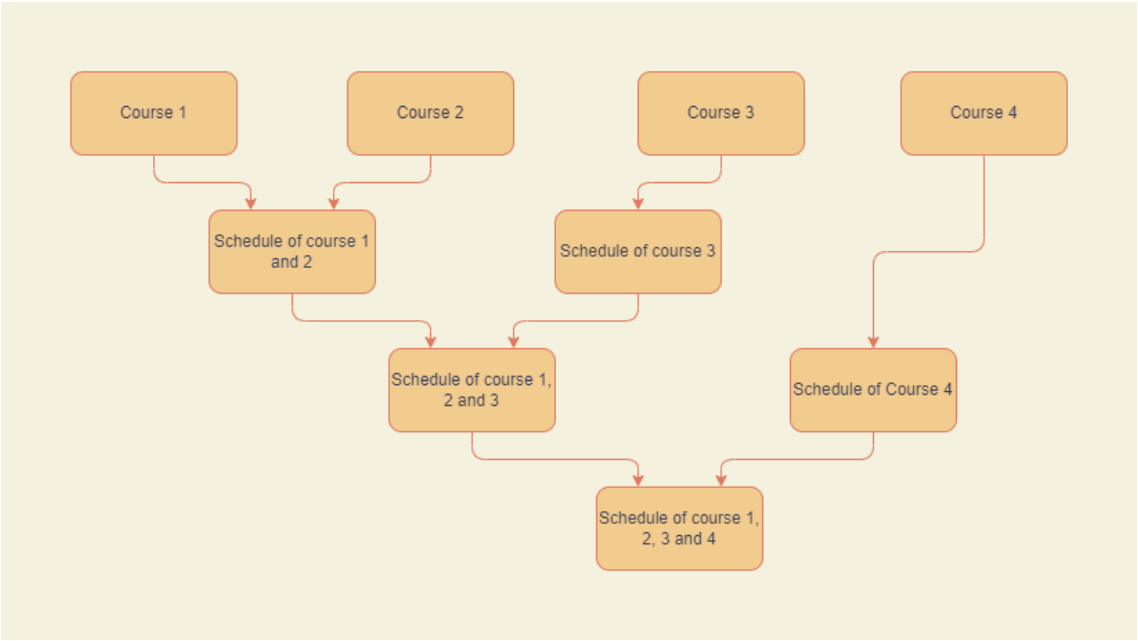

Figure 4.4: Stages of Algorithm for 4 courses

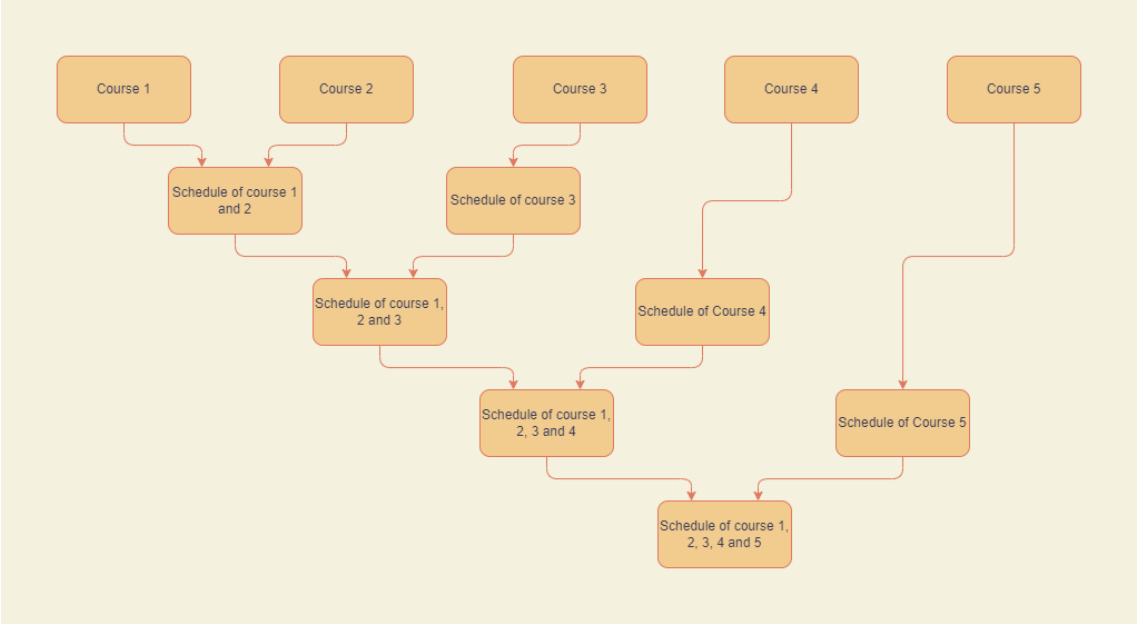

Figure 4.5: Stages of Algorithm for 5 courses

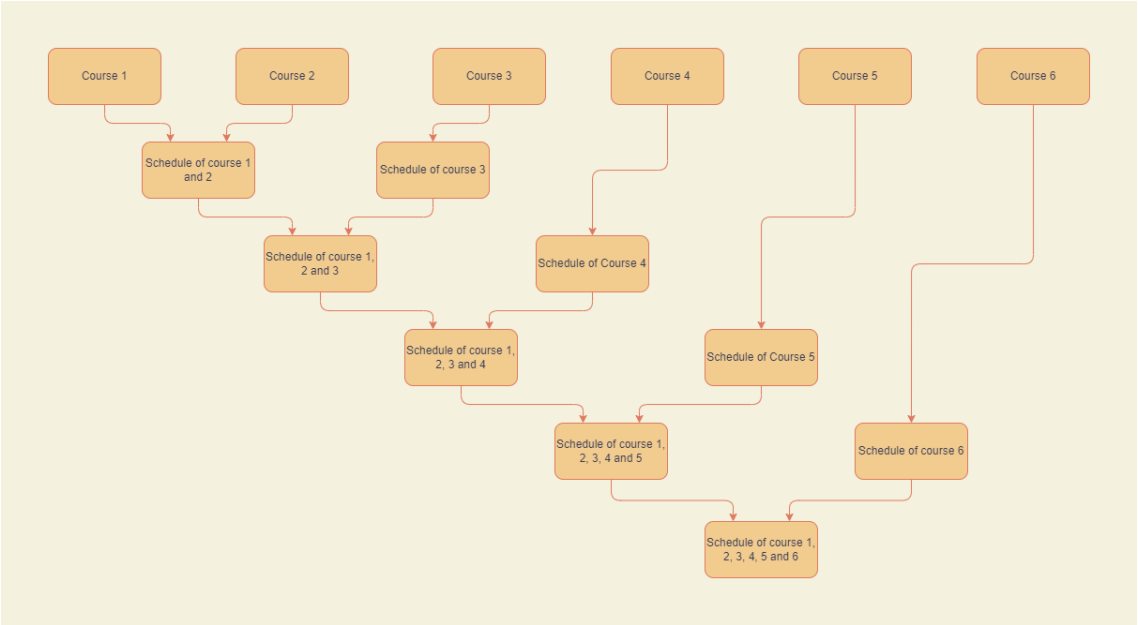

Figure 4.6: Stages of Algorithm for 6 courses

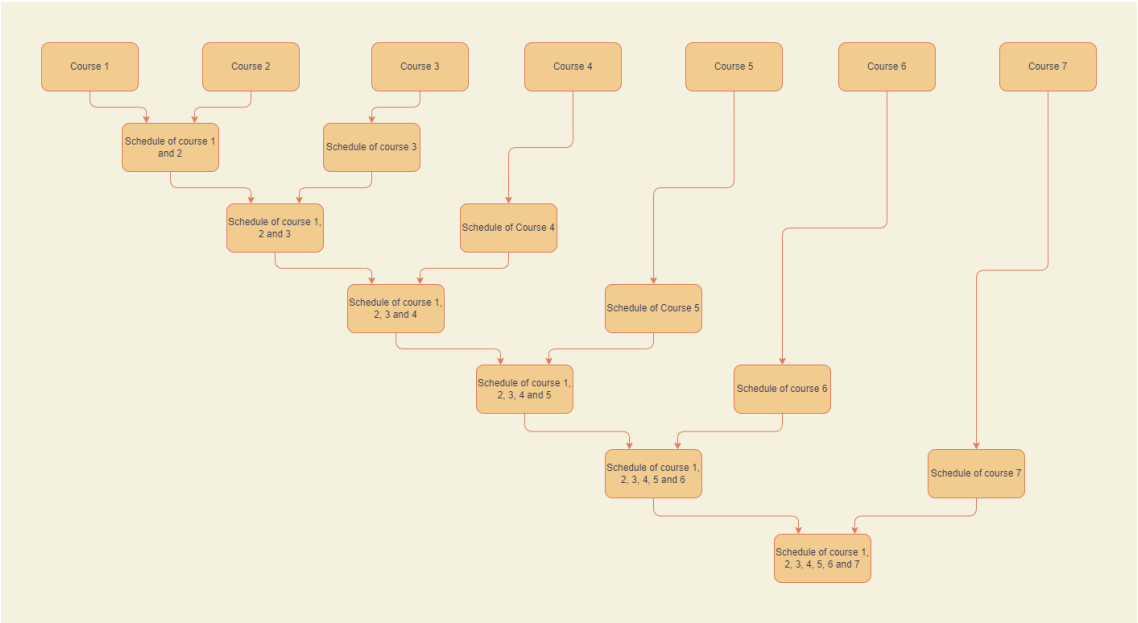

Figure 4.7: Stages of Algorithm for 7 courses

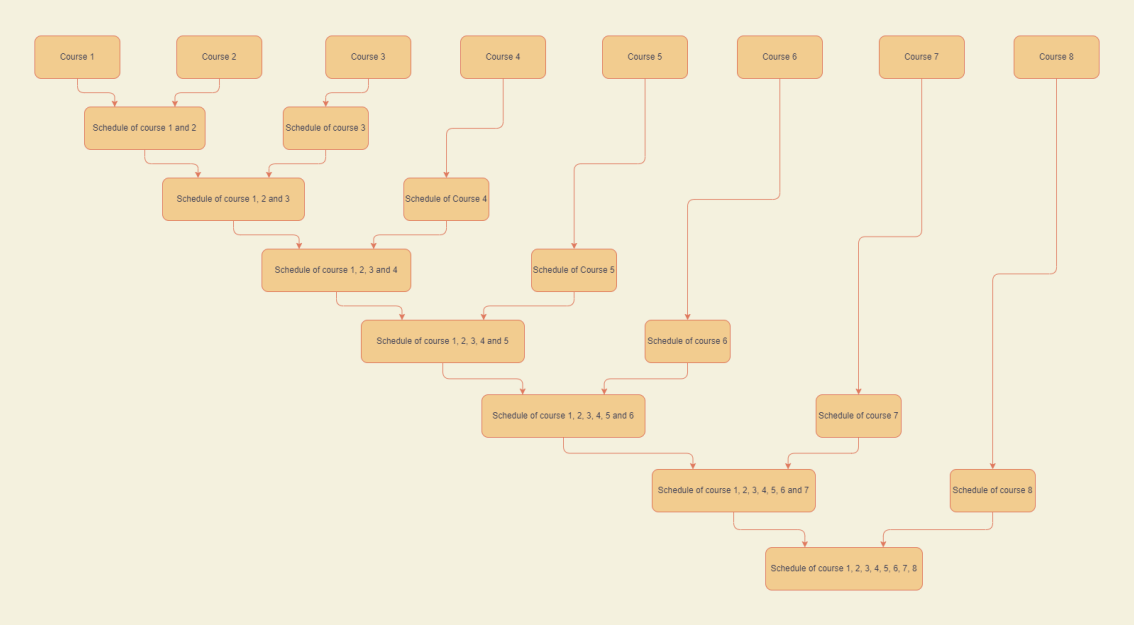

Figure 4.8: Stages of Algorithm for 8 courses

4.2.3 Algorithm Evaluation

In literature review we analyze some papers. From those papers, we learn that there are so many algorithms to solve scheduling problem. From those algorithms we take Genetic Algorithm for implementation. But in our implementation phase we build our own algorithm for solving our problem which is inspired from Genetic Algorithm. Our algorithm can work up to 8 courses and provide solution. Our algorithm time complexity is $O(n^2)$. Our algorithm can generate schedule less than 1 minute in worse case scenario.

4.2.4 Testing

We have tested our system with multiple inputs and for all types of inputs it is able to generate conflict free class schedule with 100% accuracy if there is any conflict free schedule possible. In figure 3.4 and 3.4 we have tested our system with 6 courses and it was able to provide all possible conflict free solutions which were accurate.

4.3 Results and Discussion

4.3.1 Results

Figure 4.9 shows the output for 3 courses. Figure 4.10 shows the output for 4 courses. Figure 4.11 shows the output for 5 courses. Figure 4.12 shows the output for 7 courses.

We have tested our system with several inputs and for all types of inputs our system is able to generate conflict free class schedule with 100% accuracy if there is any conflict free schedule possible. In our output it shows the detailed about the schedule like course code, course name, class day, section name, class time. Our solution is able to calculate

```
Suggested Schedule :
Suggested Schedule  1  :
['CSE 429/CSE 4329' 'Digital System Design ' 'A' 'Sat T' '10:05 AM - 11:35 AM']
['CSE 430 ' 'Digital System Design Lab ' 'B' 'W' '8:30 AM - 11:00 AM']
['CSE 469/PMG 4101' 'Project Management' 'A' 'Sat T' '11:40 AM - 1:10 PM']
Time Gap Of Schedule  1  :
Total Time Gap =  0  Hour  10 Minutes In One Week.
Saturday Time Gap =  5  Minutes.
Sunday Time Gap =  0  Minutes.
Tuesday Time Gap =  5  Minutes.
Wednesday Time Gap =  0  Minutes.

Suggested Schedule  2  :
['CSE 429/CSE 4329' 'Digital System Design ' 'A' 'Sat T' '10:05 AM - 11:35 AM']
['CSE 430 ' 'Digital System Design Lab ' 'B' 'W' '8:30 AM - 11:00 AM']
['CSE 469/PMG 4101' 'Project Management' 'B' 'S W' '11:40 AM - 1:10 PM']
Time Gap Of Schedule  2  :
Total Time Gap =  0  Hour  40 Minutes In One Week.
Saturday Time Gap =  0  Minutes.
Sunday Time Gap =  0  Minutes.
Tuesday Time Gap =  0  Minutes.
```

Figure 4.9: Result of 3 courses schedule

the time gap among classes for each day and also for overall week. It will help the student to know more about their schedule. Our system also provides scope for student to choose options between morning classes and day classes. All of these features will make it easier for student to choose a more optimal class routine than the manual one.

4.3.2 Discussion

Though our system can provide conflict free schedule with perfect accuracy but it is still not ready for real life usages. Now it is in a prototype stage. For real life usages it needs to integrate with Ucam and web server. As our system works perfectly so integration with Ucam will not be difficult.

4.4 Summary

We have analyzed several algorithms which are used to solve schedule generating problem. After analyzing those algorithm we developed an algorithm inspired from Genetic Algorithm. Our algorithm is able to generate conflict free optimized class schedule for at most 8 courses with 100% accuracy. It can provide class schedule based on student time preference and also calculate the total time gap among classes in a week, which will help student to choose an optimal routine.

```
Suggested Schedule :
Suggested Schedule  1  :
['EEE 2113/ CSE 113' 'Electrical Circuits' 'A' 'Sat T' '11:40 AM - 1:10 PM']
['EEE 2123/ CSE 123' 'Electronics' 'B' 'S W' '10:05 AM - 11:35 AM']
['EEE 2124/ CSE 124' 'Electronics Lab' 'B' 'T' '8:30 AM - 11:00 AM']
['EEE 4261' 'Green Computing' 'A' 'S W' '11:40 AM - 1:10 PM']
Time Gap Of Schedule  1  :
Total Time Gap =  0  Hour  50 Minutes In One Week.
Saturday Time Gap =  0  Minutes.
Sunday Time Gap =  5  Minutes.
Tuesday Time Gap =  40  Minutes.
Wednesday Time Gap =  5  Minutes.

Suggested Schedule  2  :
['EEE 2113/ CSE 113' 'Electrical Circuits' 'A' 'Sat T' '11:40 AM - 1:10 PM']
['EEE 2123/ CSE 123' 'Electronics' 'B' 'S W' '10:05 AM - 11:35 AM']
['EEE 2124/ CSE 124' 'Electronics Lab' 'B' 'T' '8:30 AM - 11:00 AM']
['EEE 4261' 'Green Computing' 'B' 'S W' '8:30 AM - 10:00 AM']
Time Gap Of Schedule  2  :
Total Time Gap =  0  Hour  50 Minutes In One Week.
Saturday Time Gap =  0  Minutes.
Sunday Time Gap =  5  Minutes.
Tuesday Time Gap =  40  Minutes.
Wednesday Time Gap =  5  Minutes.
```

Figure 4.10: Result of 4 courses schedule

```
Suggested Schedule :
Suggested Schedule  1  :
['PHY 2105/ PHY 105' 'Physics' 'A' 'S W' '1:30 PM - 3:00 PM']
['PHY 2106 /PHY 106' 'Physics Lab' 'A' 'W' '11:00 AM - 1:30 PM']
['MATH 2205/ STAT 205' 'Probability and Statistics' 'B' 'Sat T' '11:40 AM - 1:10 PM']
['SOC 2101/SOC 101' 'Society, Technology and Engineering Ethics/Society, Environment and Engineering Ethics' 'B' 'Sat T' '3:00 PM - 4:30 PM']
['CSI 229 ' 'Numerical Methods' 'A' 'Sat T' '10:05 AM - 11:35 AM']
Time Gap Of Schedule  1  :
Total Time Gap =  3  Hour  50 Minutes In One Week.
Saturday Time Gap =  115  Minutes.
Sunday Time Gap =  0  Minutes.
Tuesday Time Gap =  115  Minutes.
Wednesday Time Gap =  0  Minutes.

Suggested Schedule  2  :
['PHY 2105/ PHY 105' 'Physics' 'A' 'S W' '1:30 PM - 3:00 PM']
['PHY 2106 /PHY 106' 'Physics Lab' 'A' 'W' '11:00 AM - 1:30 PM']
['MATH 2205/ STAT 205' 'Probability and Statistics' 'C' 'Sat T' '11:40 AM - 1:10 PM']
['SOC 2101/SOC 101' 'Society, Technology and Engineering Ethics/Society, Environment and Engineering Ethics' 'B' 'Sat T' '3:00 PM - 4:30 PM']
['CSI 229 ' 'Numerical Methods' 'A' 'Sat T' '10:05 AM - 11:35 AM']
Time Gap Of Schedule  2  :
Total Time Gap =  3  Hour  50 Minutes In One Week.
Saturday Time Gap =  115  Minutes.
Sunday Time Gap =  0  Minutes.
Tuesday Time Gap =  115  Minutes.
Wednesday Time Gap =  0  Minutes.
```

Figure 4.11: Result of 5 courses schedule

```
Suggested Schedule :
Suggested Schedule  1  :
['CSE 465/CSE 4165' 'Web Programming Lab' 'A' 'S' '2:00 PM - 4:30 PM']
['CSE 481/CSE 4181' 'Mobile Application Developmen Lab' 'A' 'W' '2:00 PM - 4:30 PM']
['CSI 321/CSE 3421' 'Software Engineering' 'B' 'Sat T' '10:05 AM - 11:35 AM']
['CSE 3812/CSI 342' 'Artificial Intelligence Lab' 'E' 'Sat' '2:00 PM - 4:30 PM']
['CSI 421/ CSE 4521/     CSE 4621' 'Computer Graphics' 'A' 'Sat T' '8:30 AM - 10:00 AM']
['CSI 422' 'Computer Graphics Lab' 'B' 'W' '11:00 AM - 1:30 PM']
['IPE 3401/IPE 401' 'Industrial Management/Industrial and Operational Management' 'C' 'Sat T' '11:40 AM - 1:10 PM']
Time Gap Of Schedule  1  :
Total Time Gap =  1  Hour  40 Minutes In One Week.
Saturday Time Gap =  60  Minutes.
Sunday Time Gap =  0  Minutes.
Tuesday Time Gap =  10  Minutes.
Wednesday Time Gap =  30  Minutes.

Suggested Schedule  2  :
['CSE 465/CSE 4165' 'Web Programming Lab' 'A' 'S' '2:00 PM - 4:30 PM']
['CSE 481/CSE 4181' 'Mobile Application Developmen Lab' 'A' 'W' '2:00 PM - 4:30 PM']
['CSI 321/CSE 3421' 'Software Engineering' 'C' 'S W' '11:40 AM - 1:10 PM']
['CSE 3812/CSI 342' 'Artificial Intelligence Lab' 'D' 'Sat' '11:00 AM - 1:30 PM']
['CSI 421/ CSE 4521/     CSE 4621' 'Computer Graphics' 'A' 'Sat T' '8:30 AM - 10:00 AM']
['CSI 422' 'Computer Graphics Lab' 'A' 'T' '11:00 AM - 1:30 PM']
['IPE 3401/IPE 401' 'Industrial Management/Industrial and Operational Management' 'B' 'S W' '10:05 AM - 11:35 AM']
Time Gap Of Schedule  2  :
Total Time Gap =  3  Hour  0 Minutes In One Week.
Saturday Time Gap =  60  Minutes.
Sunday Time Gap =  5  Minutes.
Tuesday Time Gap =  60  Minutes.
Wednesday Time Gap =  55  Minutes.
```

Figure 4.12: Result of 7 courses schedule

Chapter 5

Standards and Design Constraints

In this chapter readers will get a brief knowledge about which standards we followed for our project. Then we discussed about the constraints of our project. Then we discussed about cost analysis and revenue plan as well.

5.1 Compliance with the Standards

The following are the standards that we followed in our project:

Software Development Life Cycle

Here we will follow ISO/IEC 12207 standard which is an international standard for software life cycle processes. Here we will use Agile Software development life cycle model.

Coding Standard

Here we will use Python for our project so we will also use PEP8 which is style guide for python code.

Version Controlling Standard

For version controlling we will use git and some best practice like commit changes atomically, commit file with a single purpose etc.

Ethical Standard

Here we will use some ethical standards like users data privacy standard, data confidentiality standard etc.

5.2 Design Constraints

We have listed some constraints of our project. These all constraints are listed while our project is in developing stage.

5.2.1 Economic Constraint

Since we are making an automated class schedule generating system so our concern is whether we will be able to sell our system to educational organizations or not. Will students be benefited from our system in the long run. We will try to overcome this problem so that students can take course and selecting their sections without any difficulties.

5.2.2 Environmental Constraint

Our system will provide automated class schedule so we are making a software-based project. For using our software, we will need some machine which will consume energy. It can be a constraint for our environment but it will be very little.

5.2.3 Ethical Constraint

In our project we will work with students selected course data, previous selected section data and university provided section list data. Those are confidential information and we need to make sure that those information will not be leaked and will not happen any data breach.

5.2.4 Social Constraint

Our system will be very much helpful for students and university authority. It will generate an automated class routine so that students don't need to select it manually. So, our system will have acceptability in our society and there is nothing in our system that can violate our culture.

5.2.5 Sustainability

Our system is an essential need for students. It can reduce the registration hassle nearly zero. This system will be useful for next couple of years. So, we can say our system is sustainable. We will also maintain our system and provide frequent update to make our system useful in the future as well.

5.3 Cost Analysis

In Table 5.1, We estimated the total cost of our project. Here total cost of our project will be 217200 taka where employee salary will be 187200 taka and maintenance cost will be 30000 taka.

In table 5.2 Here is the revenue analysis of our project. Here we estimate that we will sell our product at around 280000 taka then our profit will be 62800 taka from 1 organization.

Expenses	Unit	Number of Unites	Cost
Single Employee Salary	Per Hour	1 Hour	50 Taka
	Per Day	8 Hour	4000 Taka
	Per Month	208 Hour	10400 Taka
	3 Month	624 Hour	31200 Taka
Six Employee Salary	3 Month	3744 Hour	187200 Taka
Maintenance Cost	1 Month		10000 Taka
Maintenance Cost	3 Month		30000 Taka
Total Cost of Project (Salary and Maintenance)	3 Month		217200 Taka

Table 5.1: Budget and Cost Analysis

Number of Unit	Selling Price	Cost	Revenue
1	280000 Taka	217200 Taka	62800 Taka
5	1400000 Taka	1086000 Taka	314000 Taka

Table 5.2: Revenue Analysis

We will sell it at least 5 different educational organizations and then our estimated profit will be 314000 taka.

5.4 Complex Engineering Problem

In this section we talked about why and how our project is a complex engineering problem.

5.4.1 Complex Problem Solving

To be a complex engineering problem a project must have some criteria. Here 5.3 we discussed about those criteria and how those are related to our project.

The mapping of complex engineering problem solving according to our project :

Table 5.3: Mapping with complex problem solving.

P1 Dept of Knowledge	P2 Range of Conflicting Requirements	P3 Depth of Analysis	P4 Familiarity of Issues	P5 Extent of Applicable Codes	P6 Extent of Stakeholder Involvement	P7 Interdependence
√	√	√		√	√	√

P1 : Depth of Knowledge

Here Table 5.4, we find out some knowledge profile those are related to our project :

Table 5.4: Mapping with different knowledge profile P1.

K1 Natural Science	K2 Mathematics	K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practise	K7 Comprehension	K8 Research Literature
		√	√	√	√		√

K3 : Engineering Fundamentals

- System Analysis Design, Database Management System, Web Technology all are in Engineering Fundamentals. Here we are developing a application where those fundamentals are needed.

K4 : Specialist Knowledge

- Software Development Life Cycle knowledge, Programming knowledge, Database Design knowledge those are in Specialist knowledge and those are needed for our project.

K5 : Engineering Design

- Following Software Development Life Cycle as Design strategy is Engineering Design. Here we will use Agile Mythology for designing our project.

K6 : Engineering Practise

- Here we will follow some engineering practise like Agile model for software development, git for version control, PEP8 for python coding.

K8 : Research Literature

- Here we reviewed more than 10 papers and wrote a well defined literature review. So our project fulfill this section as well.

P2 : Range of Conflicting Requirements

- **Technical Issue :** The system may take a little bit long time while processing a large number of data.

P3 : Depth of Analysis

- Our Project problem does not have obvious solution and here we will provide a solution that will be unique.
- Our project has its originality and we are going to implement some unique features that requires abstract thinking.

P5 : Extent of Applicable Codes

- Maintaining some ethics such as keeping privacy of our client information.
- Apply appropriate technique, resources and IT tools like version controller, project management tools.

P6 : Extend of Stakeholders Involvement

- In our project we have different kinds of stakeholders like students, teachers, university authority and developers.

P7 : Interdependence

- Since we are going to make a class scheduling system so we need to collaborate with university authority, students and department office as well.

5.4.2 Engineering Activities

In this section, we talked about engineering activities. In Table 5.5 we showed engineering activities those are related to our project.

Table 5.5: Mapping with complex engineering activities.

A1 Range of resources	A2 Level of Interaction	A3 Innovation	A4 Consequences for society and environment	A5 Familiarity
√	√		√	

A1 : Range of Resources

In our project we use different kinds of resources like people, different kinds of engineering tools like git, project management tools, and a lot of engineering knowledge.

A2 : Level of Interaction

We collect lots of data and research about schedule generating system and papers. After that we talked about some university authority those are related to registration process and also talked with few developers about our project so that we can have basic idea how can we implement that.

A4 : Consequences for Society and Environment

Though we are trying to solve registration hassle for our student. So our project will be very much helpful for student to registering their course and also for university authority whose are related to registration process. From our project student will be able to do hassle free registration.

5.5 Summary

This chapter is about which standard we have followed and which constraints are related to our project. This chapter also shows cost analysis and also complex engineering problem specification.

Chapter 6

Conclusion

In this chapter we tried to provide some information about our project which are project summary, project limitations and the future work we are willing to do in future.

6.1 Summary

In open credit system university, we are very familiar in a word that is advising. In advising process student choose their classes time and sections based on the university provide section list. But most of the time it is very stressful for a student because the advising time is very short and the students can not match up all the classes time. On the other hand, two or more classes time are conflict with one another. As a result, they drop the course. For this reason, we are trying to develop a method which is make an automated class schedule. Which is more optimized. In here optimized means less time gap between classes and students can take sections in their preferable time. So, our main goal is to develop an automated class schedule method that will analyze advising data and create an optimal class schedule based on student's time preference.. For developing this method we have analyzed different types of algorithms like ant colony algorithm, genetic algorithm etc. We hope this system will help all the students.

6.2 Limitation

For making this system we have some limitations. First of all budget, as a student we have limitation of money. Our fund is not sufficient enough. Another limitation is collecting data. It is very difficult to collect data and most important thing is maintaining data security. Because most of the data contains the personal information of a student. In our project, we are solving the problem from students perspective.We can not work for teachers perspective because of time limitation. We believe that a teacher also needs some relaxation after one class. Other side now a days there are so many students in a class so the teacher can not take care of all the students. For the time limitation we are not solving those problem in our project. We hope we will make a system, that is single system but

solve multiple task.

6.3 Future Work

We are working at this moment from the students perspective. But in future we will work from teachers perspective also. Because we will try that a teacher can get sufficient amount of time between classes to relax after a class and can also take some preparation for the next class. And give the equal number of classes all day in a week. We will also try to work on maintaining a moderate number of students in a class. Because now in a class there is so many students, that's why teacher cannot take care all of the student. We have also planned to develop a feature in our system that will assign a teacher in that course which he is expert based on students' feedback. It will be more effective and helpful for all the students. We hope we will be able to develop a useful system that will be very much helpful for university authority, faculties and students.

References

[1] Brac university website. `https://www.bracu.ac.bd/`.

[2] North south university website. `http://www.northsouth.edu/`.

[3] East west university website. `https://www.ewubd.edu/`.

[4] East west university website. `https://www.aiub.edu/`.

[5] Ahmed Wasfy and F Aloul. Solving the university class scheduling problem using advanced ilp techniques. In *IEEE GCC Conference*, 2007.

[6] David Kristiadi and Rudy Hartanto. Genetic algorithm for lecturing schedule optimization. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 13(1):83–94, 2019.

[7] Jing Xu. Improved genetic algorithm to solve the scheduling problem of college english courses. *Complexity*, 2021, 2021.

[8] Rajaram Ambole and Dinesh Hanchate. Class timetable scheduling with genetic algorithm. *International journal of computer sceince and Technology*, 4:371–375, 12 2013.

[9] Panagiotis Adamidis and Georgios Kinigopoulos. Optimization of student personal course schedules with evolutionary algorithms. *BALCOR 2011*, page 146, 2011.

[10] Mohit Kumar Kakkar, Jajji Singla, Neha Garg, Gourav Gupta, Prateek Srivastava, and Ajay Kumar. Class schedule generation using evolutionary algorithms. In *Journal of Physics: Conference Series*, volume 1950, page 012067. IOP Publishing, 2021.

[11] Tanzila Islam, Zunayed Shahriar, Mohammad Anower Perves, and Monirul Hasan. University timetable generator using tabu search. *Journal of Computer and Communications*, 4(16):28–37, 2016.

[12] Rangga Sidik, Mia Fitriawati, Syahrul Mauluddin, and A Nursikuwagus. A schedule optimization of ant colony optimization to arrange scheduling process at certainty variables. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 9(12):318–323, 2018.

[13] Anirudha Nanda, Manisha P Pai, and Abhijeet Gole. An algorithm to automatically generate schedule for school lectures using a heuristic approach. *International journal of machine learning and computing*, 2(4):492, 2012.

[14] Ibrahim Aljarah, Ayiad Salhieh, and Hossam Faris. An automatic course scheduling approach using instructors' preferences. *International Journal of Emerging Technologies in Learning (iJET)*, 7(1):24–32, 2012.