

# **Project Report**

## **Acknowledgment**

I would like to express my sincere gratitude to my mentor, faculty members, and peers who guided me throughout this project. Their invaluable suggestions and continuous support made this project possible.

## **Introduction**

### **Motivation**

Time management is crucial in academic institutions. Automating timetable generation enhances efficiency and minimizes scheduling conflicts.

### **Problem Statement**

Manual timetable scheduling is error-prone and time-consuming. The project aims to develop an automated timetable generation system that optimally assigns lectures based on predefined constraints and priorities.

## **System Analysis**

### **Existing Systems**

These methods often lead to inefficiencies, such as faculty scheduling conflicts, improper room allocation, and unbalanced subject distribution across the week.

### **Scope & Limitations of Existing Systems**

Many systems require extensive manual intervention, reducing their effectiveness in complex institutional settings. Scalability and flexibility remain major challenges, especially in multi-department institutions where scheduling conflicts frequently arise.

### **Project Scope**

The proposed AI-powered timetable generation system aims to automate the scheduling process while ensuring optimized faculty allocation, subject priority management, and real-time adaptability. By leveraging Genetic Algorithms (GA), the system will intelligently distribute subjects across available timeslots, considering faculty availability.

# Requirement Analysis

## 1. Fundamental Requirements

- **User-Friendly Interface:** The system should have an intuitive and responsive web-based interface that allows administrators and faculty to easily input data, view schedules, and make modifications.
- **Multi-Class and Multi-Department Support:** The system should be capable of handling multiple departments, courses, and classes simultaneously, ensuring that institutions of varying sizes can use it effectively.
- **Conflict Resolution:** The AI-based scheduling should detect and resolve conflicts related to faculty unavailability, room allocation, and overlapping subjects.

## 2. Performance Requirements

- **Efficient Processing:** The timetable should be generated within seconds or minutes, even for large datasets involving multiple subjects, classes, and faculty members.
- **Quick Data Retrieval:** The database should efficiently handle queries related to subjects, faculty availability, and room allocations to ensure smooth scheduling operations.
- **Scalability:** The system should be designed to accommodate an increasing number of users, courses, and timetable constraints without significant performance degradation.

## 3. Security Requirements

- **Role-Based Access Control (RBAC):** Different users (e.g., administrators, faculty, students) should have different levels of access to the system. Only authorized users should be able to modify timetables.
- **Data Encryption:** Sensitive information such as faculty schedules and user credentials should be encrypted to protect against unauthorized access.

# Feasibility Study

## 1. Technical Feasibility

The system is developed using Python as the backend programming language, utilizing the Flask framework for web application development.

The AI-based timetable generation leverages Genetic Algorithms (GA) to optimize scheduling, reducing conflicts and ensuring efficient allocation of faculty, classrooms, and subjects.

## 2. Economic Feasibility

The implementation of an automated timetable generation system significantly reduces the costs associated with manual scheduling. Traditionally, timetable creation requires substantial administrative effort, often leading to errors and inefficiencies. By automating the process, institutions can save on:

- Human resource costs, as fewer staff members are required to manage scheduling.
- Time loss due to conflicts and rework, leading to improved overall efficiency.

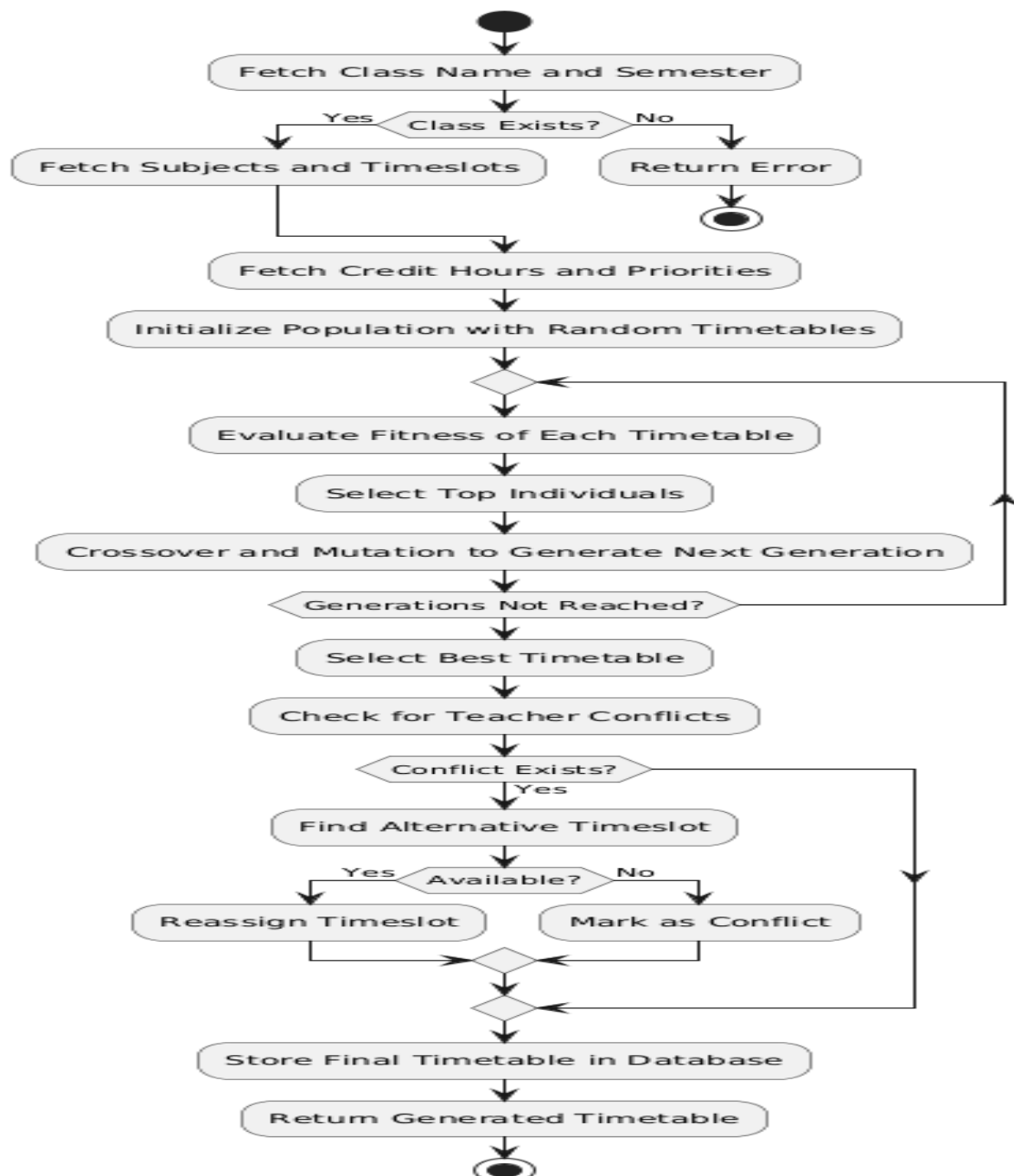
## 3. Operational Feasibility

The system is designed with usability and efficiency in mind, making it highly adaptable for administrators, faculty, and students. Key operational benefits include:

- **Simplified scheduling process:** Users can input constraints such as faculty availability, classroom capacity, and subject priorities, and the system automatically generates an optimized timetable.
- **Conflict-free scheduling:** The AI-driven approach ensures that faculty members are not double-booked and that classrooms are efficiently utilized.
- **Error reduction:** The automated approach minimizes human errors and ensures that no class or subject is omitted.
- **Real-time updates and modifications:** If unforeseen changes occur (such as faculty unavailability), the system can quickly generate an updated timetable.

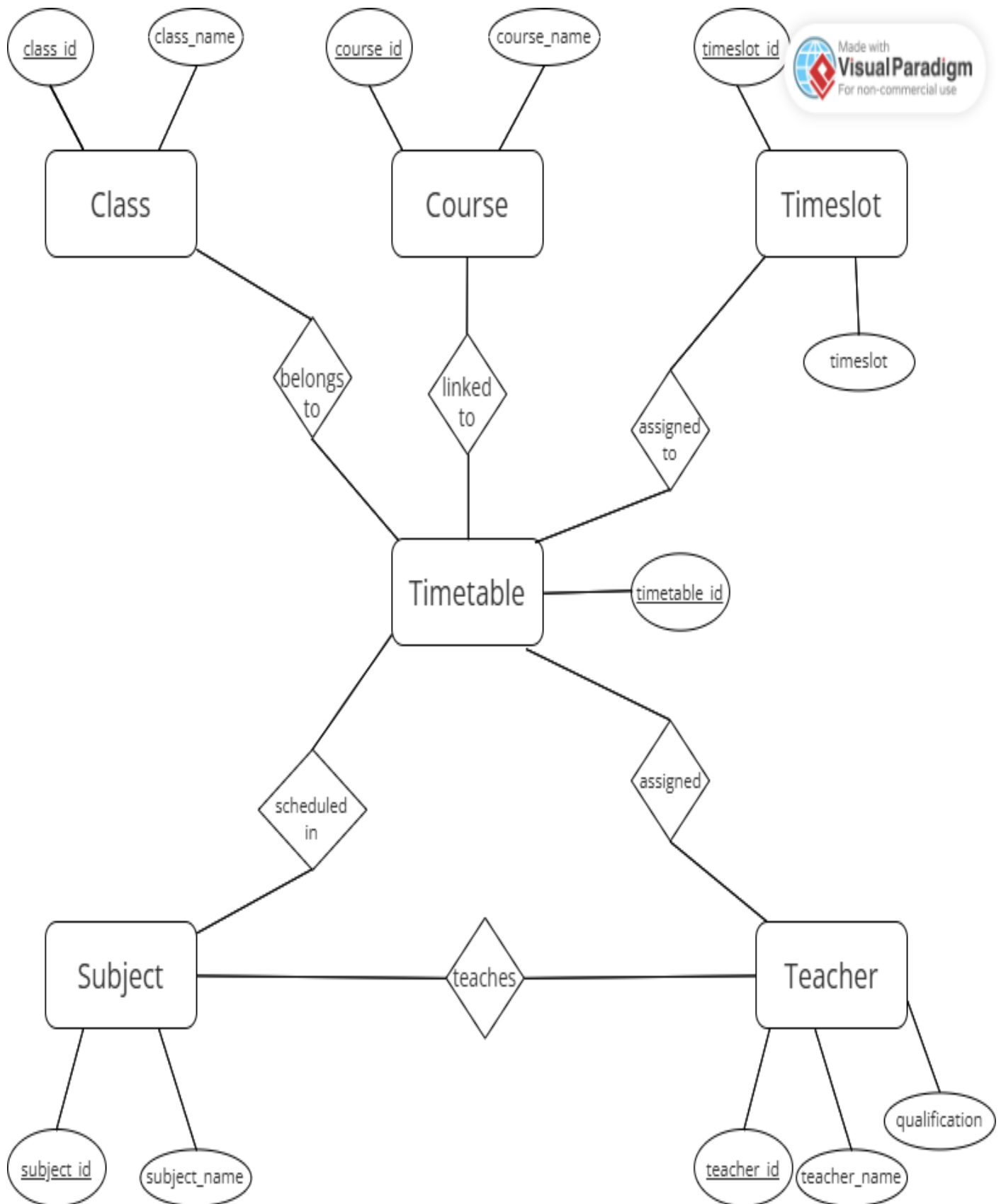
# Genetic Algorithm

The **Genetic Algorithm (GA)** is an optimization technique inspired by the process of natural selection. It is widely used to solve complex problems by mimicking the principles of evolution, including selection, crossover, and mutation. The algorithm starts with a population of possible solutions (individuals), each represented as a chromosome containing a set of parameters. These individuals undergo an iterative process where their fitness is evaluated based on a predefined function. The fittest individuals, which provide the most optimal solutions, are selected for reproduction.

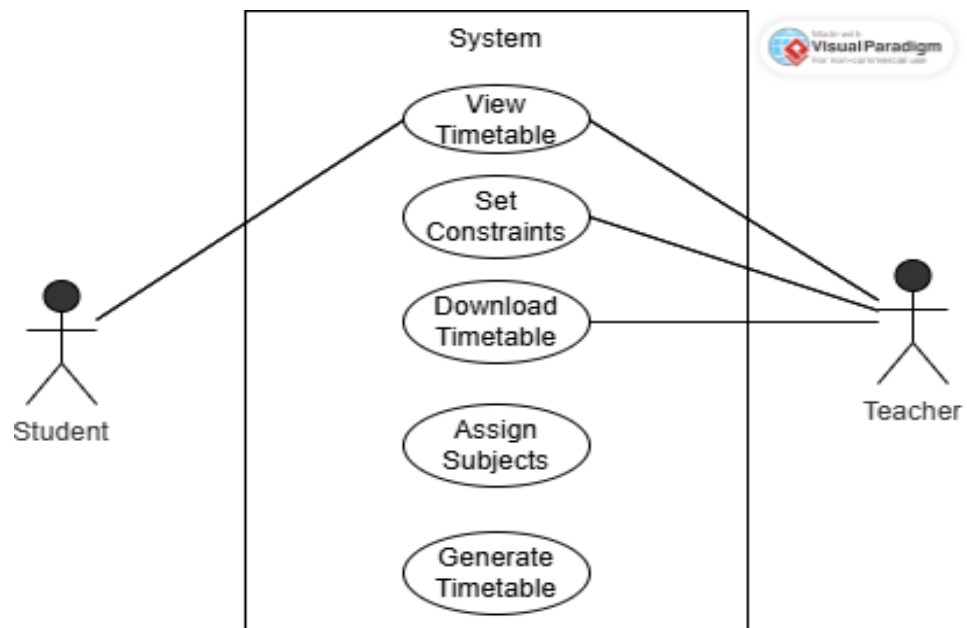


# System Design

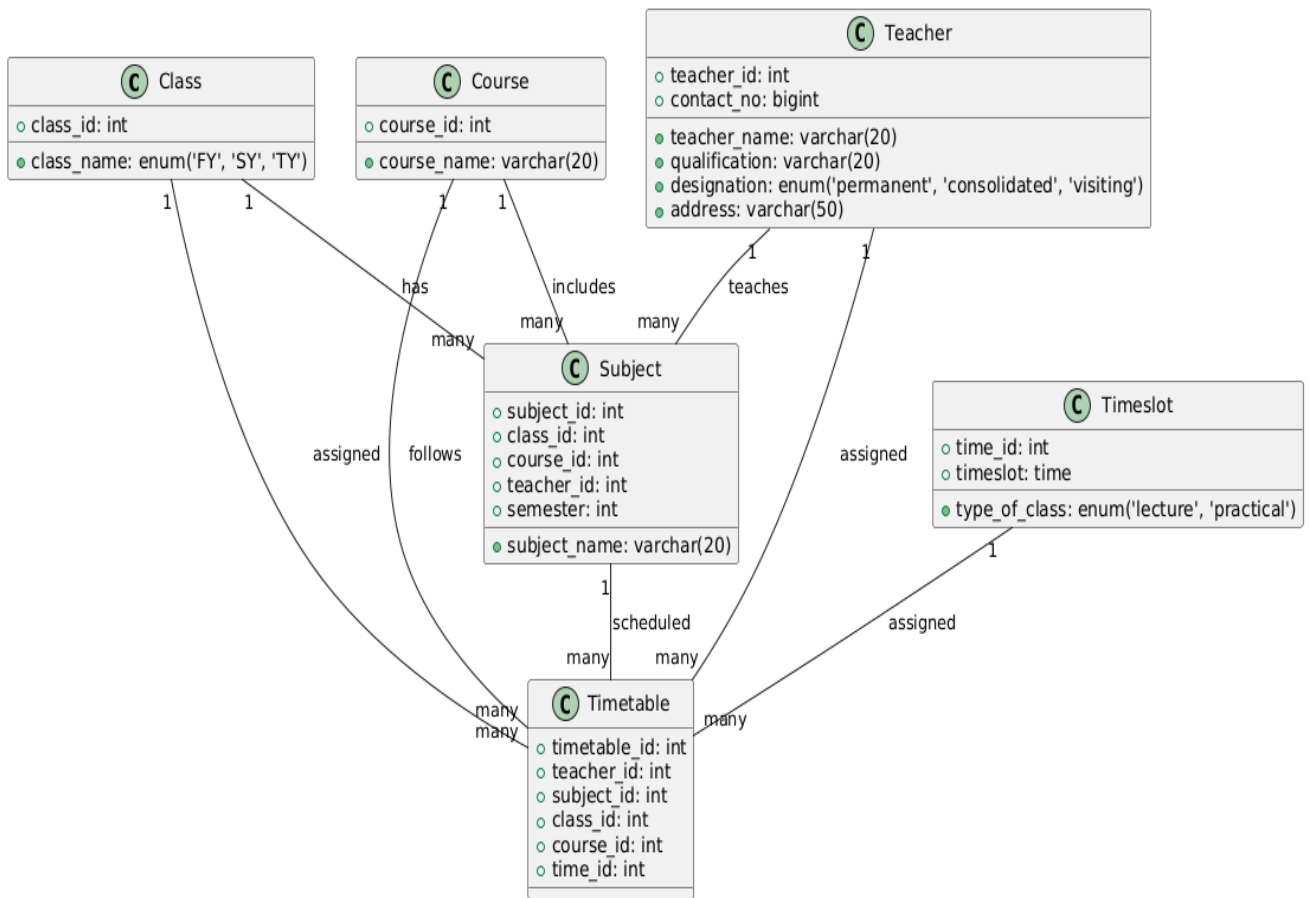
- ER Diagram



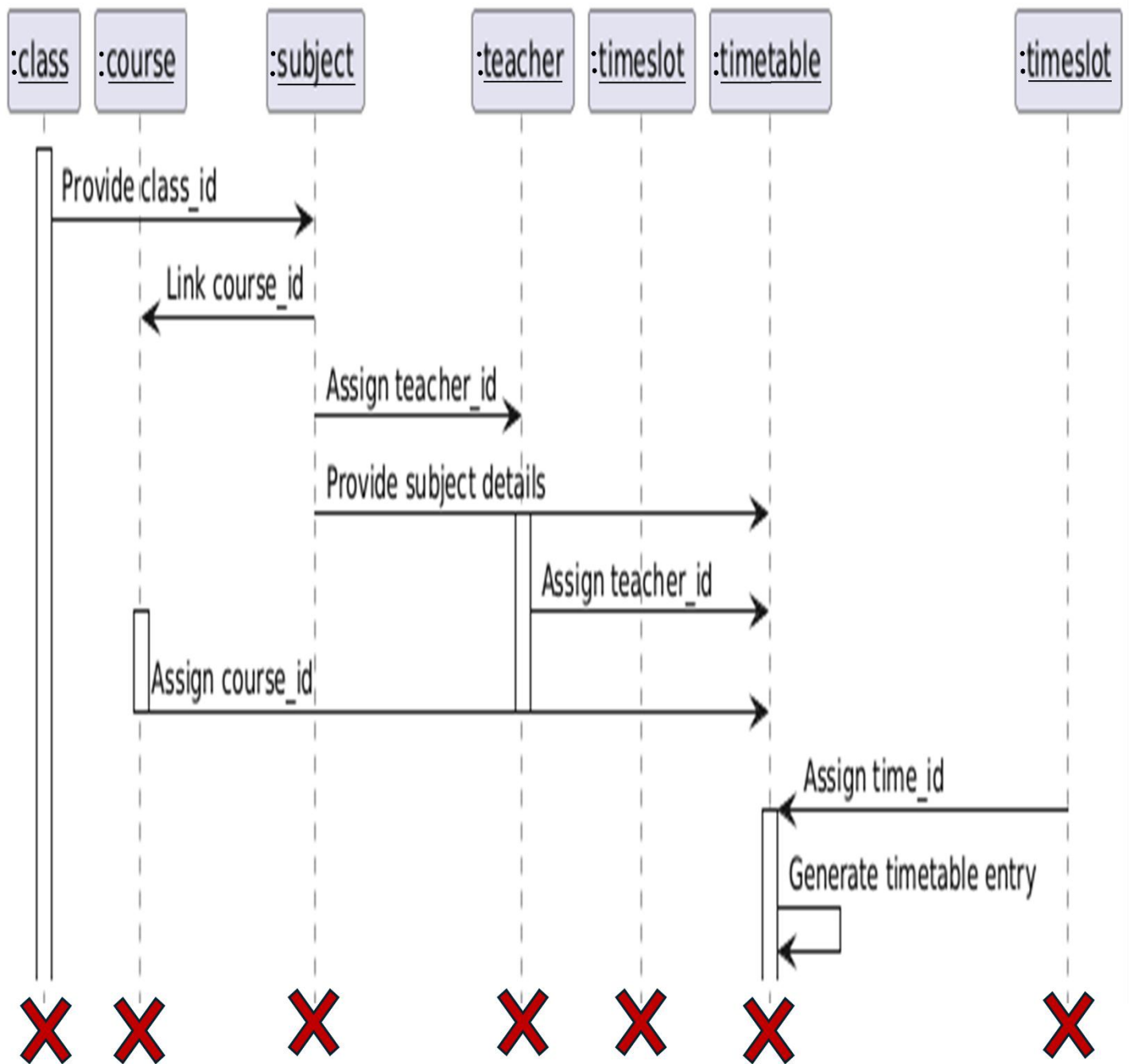
- **Use Case Diagram**



- **Class Diagram**

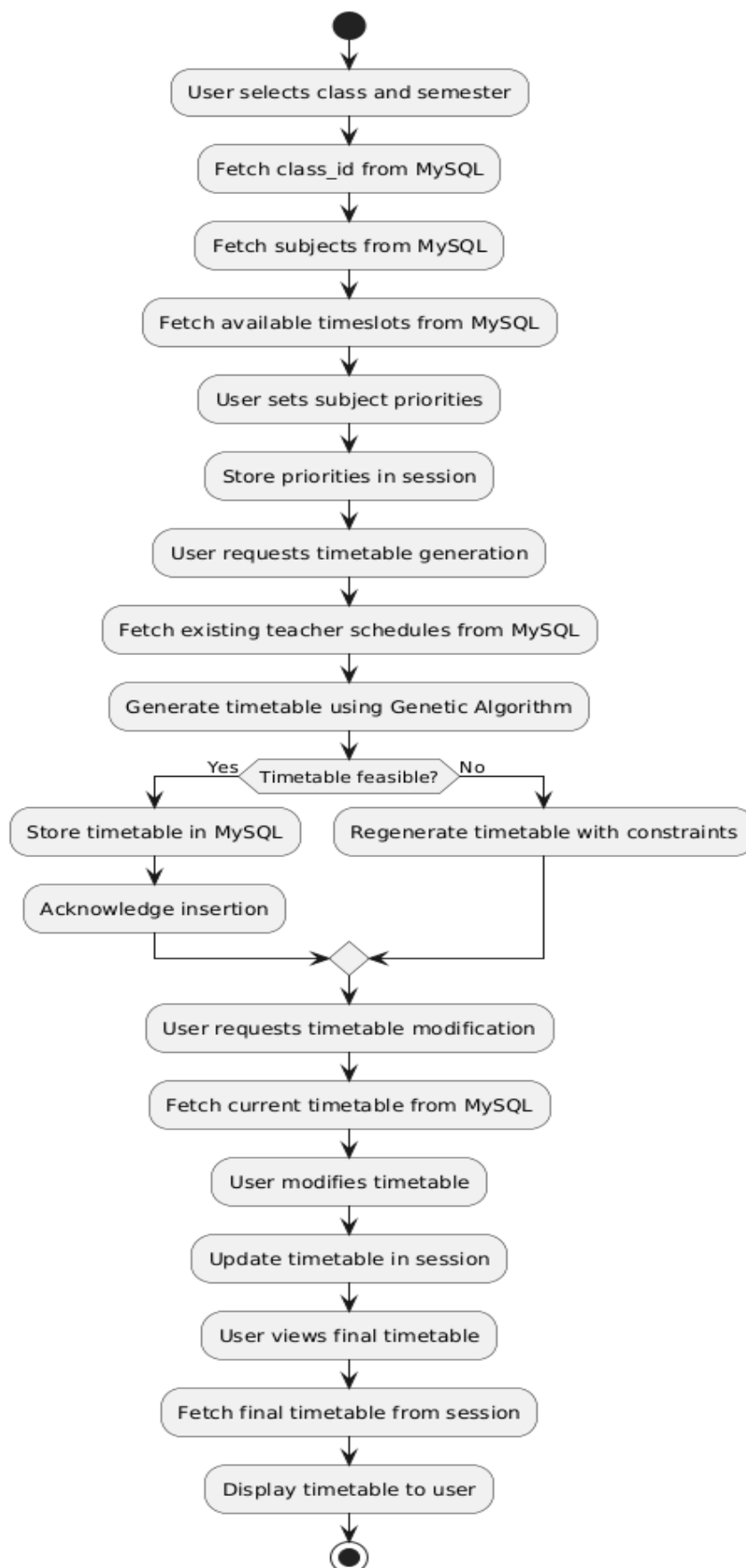


- Sequence Diagram





- **Activity Diagram**



## I/O Screens

### Step 2: Enter Subject Credits

#### Assign Credits

DSA:   
CS 2:   
TCS:   
Math 1:   
Math 2:

[Generate Timetable](#)

### Final Timetable

Timeslot	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
11:30:00	—	TCS	Math 1	—	—	DSA
12:30:00	Math 2	TCS	Math 2	—	Math 1	—
13:30:00	—	—	CS 2	DSA	—	CS 2

 [Print Timetable](#)

[Modify](#)

### Student Timetable

Class Name:

Semester:

[View Timetable](#)

Timeslot	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
11:30:00	—	TCS	Math 1	—	—	DSA
12:30:00	Math 2	TCS	Math 2	—	Math 1	—
13:30:00	—	—	CS 2	DSA	—	CS 2

## Implementation Details

### Software/Hardware Specifications

- Backend: Python
- Database: MySQL
- Platform: Windows

### Test Cases

- Validating correct subject assignments.
- Ensuring conflicts are minimized.
- Checking if all constraints are met.

**Conclusion & Recommendation** The system efficiently automates timetable generation, reducing manual workload and improving accuracy. Future improvements could include AI-based optimizations for scheduling.

### Limitations

- Currently supports only predefined constraints.
- Requires manual adjustments for unforeseen changes.

### Future Scope

- Integration with AI for dynamic scheduling adjustments.
- Web-based interface for broader accessibility.

### Bibliography & References

- Research papers on scheduling algorithms.
- Documentation for Python and MySQL.