

Schedule Management System

1. Introduction

Team Details

Name	Roll No	Email
Akshat Batra	IMT2023025	Akshat.Batra@iiitb.ac.in
Dhruv Ramesh Joshi	IMT2023032	dhruvramesh.joshi@iiitb.ac.in
Ankith Kini	IMT2023075	Ankith.Kini@iiitb.ac.in
Arnav Oruganty	IMT2023078	Arnav.Oruganty@iiitb.ac.in
Vaishak Prasad Bhat	IMT2023085	Vaishakprasad.bhat@iiitb.ac.in
Harshit Krishna R	IMT2023613	Harshit.Krishna@iiitb.ac.in

Project Overview

The College Schedule Management System is designed to provide an automated solution for managing and organising academic timetables through a command-line interface (CLI). We take file inputs using File I/O and export our output to a CSV file. This system addresses challenges related to manual scheduling, such as conflicting time slots, classroom allocation, and faculty availability. It ensures no scheduling conflicts, such as double-booking classrooms or faculty, occur.

High-Level Functionality:

- Automated timetable generation.
- Conflict resolution for room and faculty availability.
- Providing batch-wise timetable.
- Scheduling for both academic classes and exams

Scope

Current Capabilities:

- **Make Course Timetable for All Batches in the College:** Develop a fully automated timetable for each batch, avoiding faculty and room allocation conflicts.
 - Implement a common timetable to accommodate all mandatory courses, maintaining consistency across all the batches.

- **Make Timetable for Exams:** Create a separate timetable for exams, ensuring room availability and preventing clashes with regular classes. This schedule will be crafted to suit each batch, allowing students and faculty to focus on exams without impacting the regular curriculum.

Future Capabilities and Additions: (Optional)

- Make a Timetable for Individual Professors based on their availability.
 - Optimising the scheduling algorithm
 - Develop it such that this system can be used to make other schedules as well like the schedule for the sporting events in our college.
-

2. Objectives

- Automate the process of creating class timetables to minimise manual errors.
 - Eliminate scheduling conflicts by ensuring faculty and room availability checks.
 - Allow real-time adjustments in case of faculty or room changes.
-

3. System Overview

Technical Specifications:

- Java for core class definitions and implementations. The algorithm for scheduling has been written in C++. We have used C++ (for class timetable scheduling) and Java (for exam scheduling) to handle input data and make the timetable algorithm. We have used JNI to interact between the C++ and Java components of our project.

Input/Output Requirements:

Input:

- We are giving file inputs - courses, dates and rooms for Exam Scheduling. And courses (batch-wise), professors and rooms for Lecture TimeTable Scheduling.

Output:

- A day-wise exam schedule for all the batches in list format is displayed via the CLI. With the export to CSV option, we get the timetable as a snapshot of a table similar to the one given by our college.
 - Similarly, we get a batch-wise timetable for lectures as a table using the export to CSV option.
-

4. Functional Requirements

Detailed Features:

- Create academic timetables for all batches. (We have assumed all rooms to have the same capacity and that a batch can entirely fit in any room. We have also assumed each batch to have the same number of students.)
- Generate timetables for exams. (We have taken the same assumptions as mentioned above)

Use Cases:

- **Administrator Use Case:** Use the CLI to create and manage timetables. (We can get the output in a tabular format).
 - **Batch Use Case:** Retrieve timetables for specific batches through simple CLI commands. (We can get the output in a tabular format).
-

5. Development Setup

Instructions

Software Requirements:

- C++ compiler (e.g., GCC)
 - Java Development Kit (JDK)
 - IDEs: IntelliJ for Java, Visual Studio for C++
-

6. Workflow

Backend:

- C++ will manage scheduling algorithms and ensure conflict resolution.

- Java will handle the input file reading and pass the required data to Java for processing. (for Exam Scheduling)
- The input and processing takes place in C++ itself for Lectures TimeTable Scheduling.

Output:

- Once we run the program, we will get the output as the timetable for all the batches (both exams and timetable) and simultaneously it will exported to a CSV file which will be properly formatted and ready to deploy wherever required.

7. Testing & Logging

Testing Strategy:

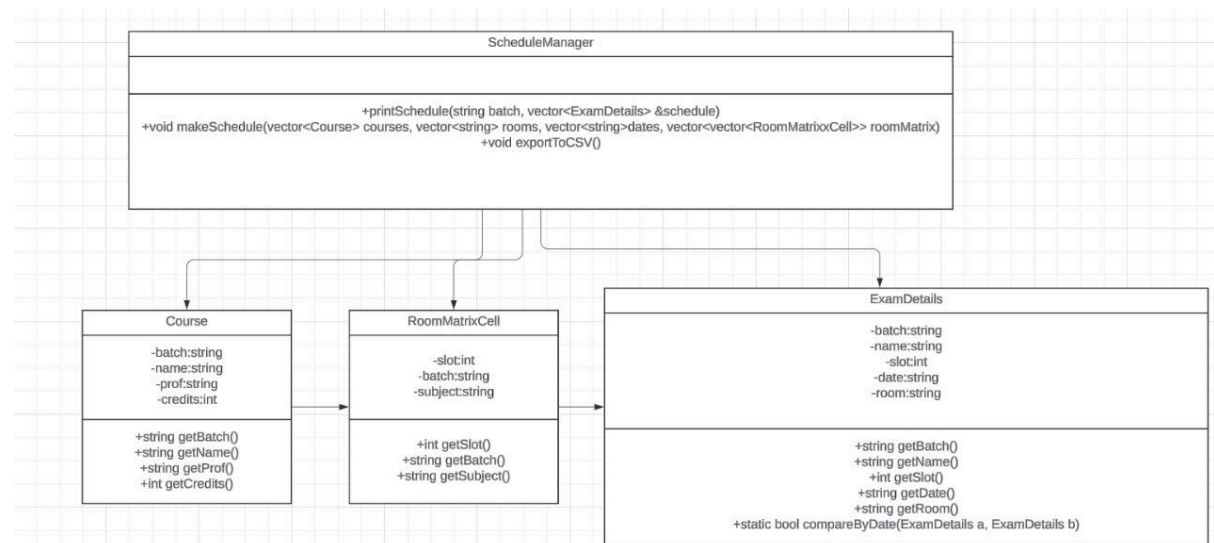
- Unit tests for individual Java and C++ classes and methods.
- Integration tests to verify communication between C++ input handling and Java processing.

Logging Mechanisms:

- Implement logging in Java to track scheduling operations and errors within the CLI system.

8. UML Diagram

Exam Scheduler UML



9. Conclusion

The College Schedule Management System aims to automate the scheduling process through a simple CLI interface, removing the need for manual scheduling and reducing errors. By combining the object-oriented capabilities of Java with C++ for file handling, the system will generate optimised timetables efficiently and ensure that both faculty and room resources are used effectively.