## Class Planning and Management System – Initial Problem Description, Solution and Plan

## **Medet Murzakhanov**

## **Introduction**

The Class Planning and Management System aims to tackle the common scheduling challenges universities face, including managing complex constraints while providing a seamless, user-friendly experience. Even while many existing systems perform well on the backend, they often lack the flexibility and intuitive design required to meet the demands of modern academic environments, particularly in hybrid learning settings. In this project I’m going to attempt to develop a solution that combines robust backend functionality with a customizable, responsive frontend, enhancing the overall user experience for students, faculty, and administrators.

## **Problem Description**

In my experience, university scheduling systems, such as the one in our institution, often manage backend tasks like constraints (class sizes, room availability, and course prerequisites) quite well. However, the main issue arises in the **frontend** of these systems, where the user experience does not meet modern standards. The interface is often cumbersome, unintuitive, and lacking in features that would simplify scheduling tasks for students, faculty, and administrators.

* **Who**: The people most affected by these limitations are students, faculty, and administrators. Students need a system that allows them to quickly and efficiently navigate their schedules, while faculty rely on accessible tools to manage their teaching schedules. Administrators are responsible for overseeing room allocations and ensuring the system follows university policies. A difficult-to-use interface slows down this entire process.
* **What**: While the backend is solid in handling constraints, the **frontend design** is outdated and inefficient, causing frustration for users. Additionally, there is a lack of integration with important platforms like Canvas and Microsoft Teams, which are good for supporting hybrid and online learning environments.
* **When**: These issues become most noticeable at the start of the semester, when students and faculty are engaging with the system the most for course registration and schedule adjustments. However, I am assuming that the difficulties persist throughout the academic year whenever changes or room reassignments are required.
* **Where**: The limitations exist primarily in the frontend of the university’s central scheduling system. Despite a well-functioning backend, the user interface fails to provide an efficient, intuitive experience, which hinders the overall usability of the system.
* **Why**: The system’s backend architecture is sound, but the frontend was not designed with modern user experience or flexibility in mind. As the needs of academic institutions evolve—especially with the rise of hybrid learning—the lack of integration and user-friendly design becomes increasingly problematic. This results in additional administrative burden and frustration for students and faculty alike.

## **Goals and requirements**

The main goal of this project is to create a scheduling system that can handle complex university timetabling challenges while being intuitive for users. I want the system to be both scalable and adaptable, meaning it can grow and change as needed. Here’s how I plan to achieve that:

* Managing Constraints (Hard and Soft):
  + One of the key requirements is that the system needs to manage both hard and soft constraints. Hard constraints are things like room availability and class size limits that absolutely have to be followed, while soft constraints are preferences, such as trying to avoid scheduling back-to-back classes for students or aligning faculty availability. By using a combination of constraint programming for hard rules and genetic algorithms for more flexible scheduling, I’ll be able to create a timetable that meets the needs of both the institution and its users.
* Integration with Digital Tools
  + Seamless integration with a variety of digital tools—such as Canvas, Microsoft Teams, and other platforms—will streamline the scheduling process for both students and instructors. This includes automatic updates to class schedules and access to online meeting links or resources, providing support for both in-person and online learning environments. As the project progresses, additional integrations may be introduced based on evolving requirements.
* Real-Time Data Management:
  + I’m planning to use PostgreSQL to handle structured data and Redis as a caching layer to ensure the system can process large volumes of data quickly and provide real-time updates. This means that if a schedule change happens, it’ll be reflected immediately across the platform.
* User-Configurable Notifications:
  + I know how important it is for users to stay informed about schedule changes, but I also want to give them control over how they receive notifications. That’s why I’ll include an option for users to customize their notifications, whether they want alerts for class changes, conflicts, or upcoming meetings.
* Customizable and Mobile-Friendly Interface:
  + Since users will be accessing the system on different devices, I’m designing the interface to be fully mobile-friendly. On top of that, users will be able to personalize their dashboards, showing them the info that matters most—whether it’s a calendar view or a list of upcoming classes.
* Student Self-Allocation of Optional Modules:
  + Another important feature I’m adding is the ability for students to allocate optional modules themselves. At the same time, the system will automatically assign mandatory modules to streamline the enrolment process. This approach gives students flexibility without overloading the administration.
* Constraint-Checking Mechanism:
  + I’m building a constraint-checking mechanism into the system to ensure that every schedule is validated before it’s finalized. This will prevent conflicts and ensure that all constraints are met, making the timetabling process smoother and more reliable.

**Success Criteria**

To measure the success of the project, I’ll be focusing on a few key areas:

* **Resource Utilization**: I want to ensure that classrooms and other resources are used as efficiently as possible, so I’ll be tracking how well the system manages room allocation.
* **Schedule Adherence**: The system needs to stick to the scheduling constraints and preferences set by the users, so I’ll be measuring how often schedules align with these requirements.
* **User Satisfaction**: I plan to collect feedback through surveys to see how students, faculty, and administrators feel about the system’s ease of use and effectiveness.
* **Conflict Resolution**: I’ll be keeping an eye on how quickly and effectively the system can resolve scheduling conflicts.
* **Flexibility**: The system’s ability to handle last-minute changes will be another important measure of success. I’ll be looking at how adaptable it is to evolving requirements.

In addition to this, range of tests will be ran that will provide comprehensive verification that the system meets functional, performance and user requirements for a successful launch:

* Unit testing
* Integration testing
* Performance testing
* Security testing
* Browser Compatibility Testing

## **Expected Project Development Plan**

|  |  |  |
| --- | --- | --- |
| Phase | Tasks | Timeline |
| Research planning | Requirement gathering, tool selection, initial scoping | October 2024 (Weeks 4-5) |
| System Design | Finalizing system architecture, UI/UX design, database schema | October – December 2024 |
| Initial Development | Building core features: backend functionality, database setup, frontend framework | December 2024 – February 2025 |
| Integration Phase | Integrating external tools (Canvas, Microsoft Teams), real-time updates, notifications | February 2025 |
| Testing & Iteration | Unit testing, system validation, resolving constraints, user feedback cycle | March 2025 |
| Refinement & Optimization | Performance optimization, refining UI, resolving bugs based on user feedback | Early April 2025 |
| Final Prototype & Demonstration | Final system presentation, gathering final feedback, making last adjustments | April 2025 (Before 28 April 2025) |