
Project: University Exam Scheduling Optimization Platform

Context and Problem

In a faculty with more than **13,000 students**, distributed across **7 departments** and over **200 academic programs** (each consisting of **6 to 9 modules**), the manual creation of exam timetables frequently leads to major conflicts. These issues include:

- Overcrowded lecture halls due to varying room capacities
- Classrooms limited to a **maximum of 20 students** during exam periods
- Scheduling conflicts for students and professors
- Limited availability of specialized equipment

To address these challenges, this project aims to **design a relational database system combined with an automatic optimization algorithm** capable of generating **conflict-free exam schedules in under 45 seconds (ideally)**.

Learning Objectives

- Master **complex relational database modeling** with multiple real-world constraints
 - Implement **advanced analytical queries** (conflict detection, optimization logic)
 - Optimize performance on **large-scale datasets** (estimated **130,000+ registrations**)
 - Develop a **functional web interface** for visualization and demonstration
-

Actors and Functionalities

Actor	Main Functionalities
Dean / Vice-Dean	Strategic overview of exam scheduling, lecture hall occupancy, conflict rates per department, final timetable validation, and academic KPIs (professor workload, room usage rate, etc.)
Exam Administrator (Planning Department)	Automatic exam timetable generation, conflict detection, and resource optimization
Department Head	Department-level validation, statistics, and conflict analysis by program

Actor	Main Functionalities
Students / Professors	Personalized exam timetable consultation with filtering by department or program

Critical Constraints to Model

- **Students:** Maximum of **one exam per day**
 - **Professors:** Maximum of **three exams per day**
 - **Rooms and lecture halls:** Must respect **real capacity limits**
 - **Priority rules:**
 - Professors supervise exams from their own department first
 - **Fair workload distribution:**
 - All professors must have an **equal number of proctoring duties**
 - Additional academic and logistical constraints as required
-

Technologies Used

- **Database Management System:** MySQL
- **Backend:** Python (optimization algorithms, conflict detection)
- **Frontend:** Streamlit (interactive web interface and visualization)