



Master Advanced Machine Learning and Multimedia Intelligence

Developing a University Timetabling System

A Step-by-Step Guide Using Spring Boot, Java, CPLEX and MySQL, Based on Integer Programming Formulation

Mohammed El Idrissi Laoukili

Supervisor:

Prof. LOQMAN CHAKIR

Academic Year: 2024/2025

Table of Contents

Summary	3
General Introduction	Error! Bookmark not defined.
I. Application modeling and database design	Error! Bookmark not defined.
1. General features of the model	Error! Bookmark not defined.
2. Special features of the model	Error! Bookmark not defined.
3. Core Entities and Relationships	Error! Bookmark not defined.
4. Database diagram	Error! Bookmark not defined.
5. Extending the Database to a Pro Version	Error! Bookmark not defined.
6. Database pro version diagram:	Error! Bookmark not defined.
7. SQL Queries for Data Retrieval and Management	Error! Bookmark not defined.
8. Data variables	Error! Bookmark not defined.
II. Development Tools & project structure	Error! Bookmark not defined.
1. Tools:	Error! Bookmark not defined.
2. Project structure:	Error! Bookmark not defined.
III. Testing and Debugging Approach	Error! Bookmark not defined.
IV. Applying the Proposed Model	Error! Bookmark not defined.
1. Timetable Diagram	Error! Bookmark not defined.
2. Constraints for the IP model	Error! Bookmark not defined.
2.1. Uniqueness constraints	Error! Bookmark not defined.
2.2. Completeness constraints	Error! Bookmark not defined.
2.3. Consecutiveness constraints	Error! Bookmark not defined.
2.4. Repetitiveness constraints	Error! Bookmark not defined.
2.5. Pre-assignment constraints	Error! Bookmark not defined.
2.6. Objective function for the IP model	Error! Bookmark not defined.
2.7. Running the program using on simple data	Error! Bookmark not defined.
3. Proposed Real-World Datasets for Testing	Error! Bookmark not defined.
3.1. Lower-Grade Dataset:	Error! Bookmark not defined.
3.2. Higher-Grade Dataset:	Error! Bookmark not defined.
V. Running timetabling application with spring boot	Error! Bookmark not defined.
Conclusion	4

Summary

This report represents the outcome of our work over the past month. The objective of this project is to design and implement a web application for generating university timetables. The methodology adopted was inspired by the research presented in the article, *"An Integer Programming Formulation for a Case Study in University Timetabling"* by S. Daskalaki, T. Birbas, and E. Housos. The article provided insights into modeling university timetabling as a 0-1 integer programming problem, addressing operational constraints and optimization to ensure feasible and quality schedules.

Before entering the development phase, we began by analyzing requirements and defining specifications, guided by the concepts in the article. Following this, we conducted a detailed design based on the Unified Modeling Language (UML). This study allowed us to establish the technical choices adopted in the development process and to create a structured work plan, enabling us to achieve good results while adhering to quality standards.

This report also demonstrates how the proposed system was incrementally developed, starting with simple terminal-based data demonstrations, progressing to database integration, and finally presenting a full-stack implementation using Spring Boot, Java, and MySQL.

Conclusion

This project has been a challenging yet rewarding experience that deepened my understanding of solving real-world problems using technology. The university timetabling problem, while seemingly straightforward, proved to be complex due to the numerous constraints that needed to be considered, such as class schedules, teacher availability, and room capacity.

I approached this challenge methodically. Initially, I decided to simplify the process by bypassing the use of a database and directly sending inputs from the React frontend to the Spring Boot backend for processing. This decision allowed me to focus on the core logic of timetable generation without being bogged down by CRUD operations.

The complexity of the project came from identifying and applying the constraints correctly. Each constraint represented a unique challenge, and balancing them required careful planning and optimization. To address this, I avoided CRUD operations and frontend data input entirely, instead opting to instantiate the required data directly within the program using a dedicated **TimetablingData** class. This approach allowed me to focus entirely on building the logic and ensuring the constraints were properly formulated before implementing a solver to generate feasible schedules. Leveraging tools like CPLEX made it possible to handle these constraints effectively and produce an optimized timetable.

This project taught me the importance of breaking down a problem into manageable parts. By focusing on the essentials first, such as constraint formulation and logic implementation, I was able to build a strong foundation before improving the system with a simple yet intuitive user interface. The output, displayed at *localhost:8080/api/timetable/solve*, provided a clear and user-friendly view of the generated timetable, showcasing the practical value of the solution.

Overall, I found this project both intellectually stimulating and deeply fulfilling. It reinforced my problem-solving skills, my ability to tackle complexity, and my confidence in applying advanced tools and frameworks like Spring Boot and CPLEX. It was **not just** about building a solution, **it was about learning how to navigate challenges, think critically, and deliver results.**

Resources:

 GitHub Repository: <https://github.com/elidrissilaoukili/timetabling>

 Project Website : <https://elidrissilaoukili.github.io/timetabling>