Software Requirements Specification
for
Timetable Manager

Alp Koçak - 20220602056
Efe Serin - 20210602055
Doğa Orhan - 20210602043
Sude Teslime Daka - 20220602207

1 Introduction

The "Timetable Manager" is a desktop application designed to help users create and manage course schedules, classroom allocations, and student enrollments in an educational setting. The application will support functionalities such as adding courses, assigning students to classes, swapping classrooms, and viewing weekly schedules for both students and classrooms. It will operate as a standalone application that doesn't require integration with other software or services.

2 User Requirements

The user requirements can be detailed as follows:

2.1 Functional Requirements

- **Functional Requirement 1:** The user must be able to add a course.
  Rationale: Adding courses is a fundamental function since this will be a scheduling application. The user must be able to define the course name, description, and other course parameters.

- **Functional Requirement 2:** The user must be able to assign classrooms to courses compatible with the capacity.
  Rationale: The application must ensure classrooms are appropriately allocated based on the number of enrolled students to prevent over placing. If a classroom is tiny, the system should look for an available classroom that meets the capacity requirements.

- **Functional Requirement 3:** The user must be able to enroll students in courses without schedule conflicts.
  Rationale: To make sure students are not enrolled in classes that overlap, the application must look for any possible conflicts with scheduling. This feature will allow students to schedule their activities more efficiently.

- **Functional Requirement 4:** The user must be able to search for a student's course schedule by week.
  Rationale: With lots of students, it is essential for the user to quickly access individual schedules, allowing them to verify or adjust enrollments as needed.

- **Functional Requirement 5:** The user must be able to check classroom schedules for any given week.
  Rationale: This feature allows the user to monitor classroom usage across the week, helping with resource allocation and resolving potential conflicts.

- **Functional Requirement 6:** The user must be able to perform a classroom swap, checking availability for that classroom.
  Rationale: In case of room changes, the application will ensure that the target classroom has

enough capacity for the course being changed. If the capacity is inadequate, the system will notify the user.

- **Functional Requirement 7:** The user must be able to add new sections to a course if the classroom capacity is full.
  Rationale: For cases where the number of students exceeds a classroom's capacity, the system will allow the creation of additional sections. New students can be enrolled in these sections to ensure all students are assigned to a sections.

- **Functional Requirement 8:** The user must be able to save, load, and update the timetable data.
  Rationale: Users will be able to save the current state of schedules, load previous timetables, and make necessary updates.

- **Functional Requirement 9:** The user must be able to import classrooms and courses data in CSV file format.
  Rationale: This enables quick data import from external sources, saving time and reducing errors.

3 System Requirements

The system requirements can be detailed as follows:

3.1 Functional Requirements

- **Functional Requirement 10:** The system must be able to load a previously saved timetable from a file.
  Rationale: This prevents data loss and enables users to access and continue working on previously made timetables.

- **Functional Requirement 11:** The system must be able to merge schedules if two timetables need to be combined.
  Rationale: It is useful to merge the two schedules so that there are no duplicate entries and conflicts can be avoided.

- **Functional Requirement 12:** The software must have a manual that will be displayed with a "Help" menu item.
  Rationale: There must be a manual on how to use the software and this manual must be accessible from the "Help" menu on your application.

3.2 Non-functional Requirements

- **Non-functional Requirement 1:** The system must be compatible with Windows operating systems.
  Rationale: The target platform is Windows, although the programming language should ideally allow portability to other platforms.

- **Non-functional Requirement 2:** Default language must be in English.
  Rationale: Using English ensures accessibility to a broader audience and facilitates use by non-Turkish speakers.

- **Non-functional Requirement 3:** The system must utilize SQLite as its file-based database.
  Rationale: The program will utilize queries to efficiently manage its database.