

Software Requirements Specification for SE302 Project Team2

Berat Bora Altaş 20210602004

Tuna Demirci 20220602025

Mert Koğuş 20210602037

Betül Özsan 20210602050

Hulki Enes Uysal 20210602212

1 Introduction

The project is going to be an application that can handle course scheduling and maintaining process. The users will be able to fit the timetables of courses into available classes with taking their student count and class capacities into account. The users will be able to inspect weekly schedules for classes, courses and students individually. Users will be able to change course classrooms with available classes. Users will also be able to choose students from courses and create new classes/meetings when needed. The program and everything it needs to run will be delivered with an installer and the program will run locally on the Windows system.

2 User Requirements

The user requirements can be detailed as follows.

Functional Requirement 1: The user shall be able to create a suitable timetable for courses.

Rationale: Creation of weekly timetables without any clashing courses and classrooms with providing information of courses and classrooms the main functionality of the application.

Functional Requirement 2: The user shall be able to add new courses.

Rationale: If a new event, such as a project meeting or a new course, is to be added, the program shall be able to arrange a time and a classroom that does not conflict with the students' schedule that selected by the user.

Functional Requirement 3: The user shall be able to change classes of courses.

Rationale: This is a functionality requested by the customer. The user shall be able to change classrooms of courses without having problems with classroom capacities.

Functional Requirement 4: The user shall be able to view the predefined weekly schedule for any classroom, course, or student, including all relevant details such as times, locations, and participants.

Rationale: Easy access to the user's weekly schedule boosts usability and efficiency since it saves critical time in retrieving information. A clearly visualized time, location, and participant details will enable users to effectively create plans for given courses.

Functional Requirement 5: The user shall be able to edit courses of students.

Rationale: In situations where students choose to withdraw from a course or join a new one, the system shall allow the user to manually transfer the student to courses.

Non-functional Requirement 1: The user shall access all system functionalities through a detailed and comprehensive user manual.

Rationale: The system shall be employed effectively by all categories of users, from the complete beginner to the expert in computing, with a comprehensive user manual.

3 System Requirements

The system requirements can be detailed as follows.

Functional Requirement 1: The software shall provide a graphical user interface.

Rationale: The system shall offer a user-friendly and easy-to-use graphical user interface (GUI). It shall effectively visualize every feature included within the application and enable user to operate the system only using the graphical interface.

Functional Requirement 2: The system shall be able to distribute courses among the available classrooms.

Rationale: The distribution process shall take classroom capacity and course schedule into account. The capacity of each classroom must be equal to or greater than the number of students enrolled in the course. The timing of the course must be considered to prevent scheduling two courses in the same classroom at the same time.

Functional Requirement 3: The program shall have a persistent data system.

Rationale: Changes made by users shall be stored in a persistent data system to save users' progress to load upon restart and resume the process.

Functional Requirement 4: The program shall be able to read data from given input files.

Rationale: The system shall support the import of data from CSV files to be used to fill critical datasets such as course and classroom details.

Non-functional Requirement 1: The system shall be developed using TypeScript, React, Tailwind and Electron 3.5.

Rationale: The React-based user interface will be developed using a modular architecture based on functional components for better reuse and easier maintenance. TypeScript is going to be used to ensure type safety, allowing better developer productivity through static typing and advanced tooling. Tailwind CSS is going to be used for utility-first styling; Electron would be utilized to enable desktop application development that smoothly integrates web technologies within native desktops. This ensures a robust, scalable, maintainable frontend architecture.

Non-functional Requirement 2: The program shall be in Turkish.

Rationale: The system will be created for use in Izmir University of Economics. The system will be in Turkish to ensure the user can understand it without any language barriers, making this choice the most suitable for the project.

Non-functional Requirement 3: The program shall run on Windows operating systems.

Rationale: Target users of the system are using Windows as their operating system, so the system shall be developed with this in mind. The software shall have an installer, and the installer shall deploy all the necessary files and libraries to execute the program.