Software Requirements Specification for “Syllabus Manager”

Benhur Okur 20220602059
Doğa Güneş 20210602009
İlayda Buzbuz 20210602017
Tunay Koluaçık 20230602089
Ebru Burhan 20210602208

1 Introduction

Syllabus Manager is a desktop application that simplifies course and classroom management. The user can add or edit courses, remove students from certain courses and also transfer students to other courses and update which classroom should be assigned for each course. It provides a GUI system for the user to work with easily in Java and have an easier time using the information about different courses and classroom data. The first CSV file has all the data of courses and classrooms making it an easy way to handle the data. This self-contained application doesn't require any external services to run.

2 User Requirements

The user requirements can be detailed as follows.

Functional Requirement 1: The user shall be able to create a new course.

Rationale: Managing the educational calendar and managing resources is a basic feature of new course creation. You can configure the basic course details, which will later be used for scheduling, classroom, and student enrolments.

Functional Requirement 2: The user shall be able to change the classroom for a course.
Rationale: Conflicting schedules, room capacity restrictions, or special needs (equipment) may result in the need to move courses around. Users have the choice of changing the assigned classroom, thus it makes the course management more flexible and effective realization of resources.

Functional Requirement 3: The user shall be able to change a student's course enrollment.
Rationale: Students may have to alter their courses due to academic needs, schedule issues, or just personal choices. This allows users to manage and update student enrollment records in one central location making sure records are accurate and up to date.

Functional Requirement 4: The user shall be able to create new courses or standalone meetings and assign them to available classrooms and students.
Rationale: This feature simplifies the configuration process so that users can create courses on a large scale or meetings on a small scale and assign them to specific classrooms and students simultaneously. This makes sure that all required associations are done in a single step, hence saving time and minimizing the risk of mistakes.

Functional Requirement 5: The user shall be able to view a student's, a lecturer’s, or a classroom’s weekly schedule.
Rationale: A clear overview of a weekly schedule ensures that the users can easily spot when and where their course conflicts, attendance tracking, and load management. An excellent feature for planning is schedule visualization, which can provide a great amount of insight to academic advisors and teachers on their end and students on theirs.

Functional Requirement 6: The user shall be able to track attendance for a course.
Rationale: Attendance tracking is important for measuring student engagement, participation, and adherence to attendance policies. The ability to track attendance accurately helps the users for performance retrieval, reporting, and compliance.

Non-Functional Requirement 1: The user interface should be easy to use and organized in such a way that user errors are minimized.
Rationale: An intuitive and ordered UI can minimize the amount of time needed for new users to learn how to use it without error. This is important for improving user satisfaction and workflow, especially among pedagogical staff that are not technically inclined.

3 System Requirements

The system requirements can be detailed as follows.
Functional Requirement 7: The system will read and write CSVs.
Rationale: CSV is a popular data storage and exchange format because of its ease of use due to its generality, and it is compatible with almost any software tool. By supporting CSV file operations, the data import/export will be seamless with any outside system - which also makes the system more universal.

Functional Requirement 8: The system should automatically assign courses to classrooms when importing courses, if possible.
Rationale: Assigning courses to classrooms at the beginning of each term is a time-consuming and difficult task. Doing it manually is prone to errors and can cause delays and stress. Automating it during course import can reduce the workload of administrative staff.

Functional Requirement 9: The System shall provide a website that introduces the application.

Rationale: As a standalone entry in an ecosystem, a website is vital to showcase the app's features and benefits as well as how potential users should use it. This feature includes the full application overview, onboarding on how to use your app, and where to find help. Thus, this helps with gaining new users. Furthermore, having a website to support the application provides better exposure to it and access for users to the system.

Non-Functional Requirement 2: The system must be running on Windows.
Rationale: Windows is a common OS within education and organizations. Making it compatible with Windows increases its accessibility and usability by the system since this makes it easier to adapt to the user infrastructure.

Non-functional Requirement 3: The system shall be in English.
Rationale: The system shall be in English as the language of instruction is English, and it ensures accessibility for international students.

Non-Functional Requirement 4: The system should handle up to 25,000 student records efficiently.
Rationale: The system should be able to work correctly regardless of how big the dataset is and ensure scalability when the number of students increases.

Non-Functional Requirement 5: The system shall ensure data security and privacy for students' personal information.
Rationale: Educational institution's data consists of potentially sensitive student information. It is important to ensure privacy compliance and that the data is protected from unauthorized access.