SOFTWARE REQUIREMENTS SPECIFICATION

Syllabus Project SE 302

Version 1.0

Prepared by :

1.Ege Pelindağ (20200602040)
2.Canberk Çoban (20210602022)
3.Ahmet Eren Sırcan(20210602057)
4.Hakan Uzun (20200602051)
5.Mehmet Gülbahar (20210602210)

Lecturer: İlker Korkmaz
November 12, 2023

Contents

Preface

This document represents the initial release (Version 1.0) of the system requirements specification. The primary target audience for this document includes the instructor of the SE302 Principles of Software Engineering course at Izmir University of Economics

1 Introduction

1.1 Purpose

The purpose of the project is to automate the process of handling course syllabuses. Currently, this process is manual, involving the use of work documents to track variable information. The goal is to develop a desktop application that enhances the management of the course syllabuses and keeps track of the changes through a version control system.

1.2 Project Scope

The project scope involves the development of a Java-based desktop application to enhance Syllabus management on the school's website. The primary objectives include implementing a user-friendly interface for creating and modifying Syllabus templates, introducing a version control system to track changes with transparency, and integrating JSON functionality for seamless data interchange. Secondary objectives encompass data import/export features, allowing users to export Syllabus data to JSON files and import JSON files to update syllabuses. Core features consist of a Syllabus management interface and a robust version control system, while additional features include JSON integration and data import/export functionality. Constraints include the development of a desktop application without web-based features and the absence of user authentication requirements. Dependencies involve utilizing appropriate Java libraries for JSON data handling and selecting suitable libraries for the version control system. Assumptions consider users' basic computer skills and compatibility with common desktop environments.

1.3 Overview

The primary objective of the project is to revolutionize the management of course syllabuses by automating the currently manual process. The existing approach relies on traditional work documents to monitor variable information, leading to inefficiencies and a lack of version control. To address these challenges, the project aims to develop a Java-based desktop application that enhances syllabus management on the school's website.

2 System Overview

This section aims to outline the technologies used for the project. A detailed explanation of the utilized technologies provides a crucial understanding of the system's structure.

2.1 Technology Used

The technology and programming language used in the Syllabus Application are as follows:

- Java has been selected as the primary programming language for the development of the Syllabus System, providing a robust and versatile foundation.
- The choice of JavaFX for the Graphical User Interface (GUI) technology ensures a modern and user-friendly interface for Syllabus System users.
- Scene Builder plays a crucial role in the development process by offering a visual layout tool, enabling swift and efficient design of JavaFX application user interfaces.
- GitHub serves as the central hub for version control, facilitating collaboration among developers and providing a transparent platform to monitor the progress of the Syllabus System project.
- The integration of Java, JavaFX, Scene Builder, and GitHub creates a well-rounded development environment, combining a powerful programming language with advanced GUI capabilities and effective version control.
- SQL (Structured Query Language) is employed in the Syllabus System for database management and interaction.
- The integration of SQL enhances the system's ability to store, retrieve, and manage data efficiently.
- SQLite is used for storage in the Syllabus Application.

2.2 General Diagrams

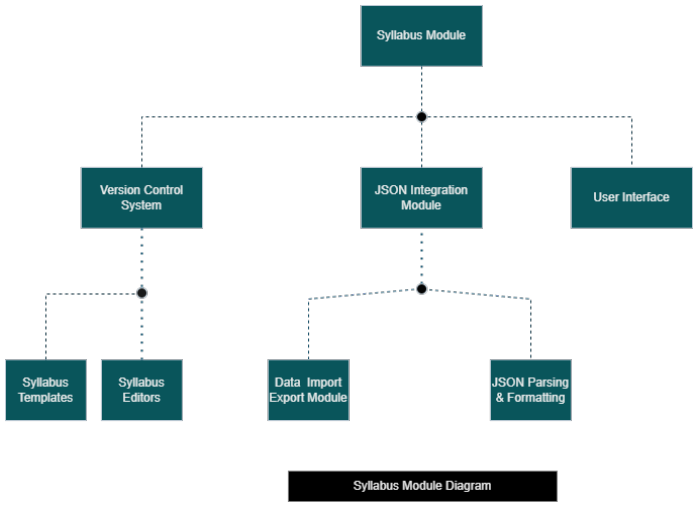

Figure 1: Syllabus Module Diagram

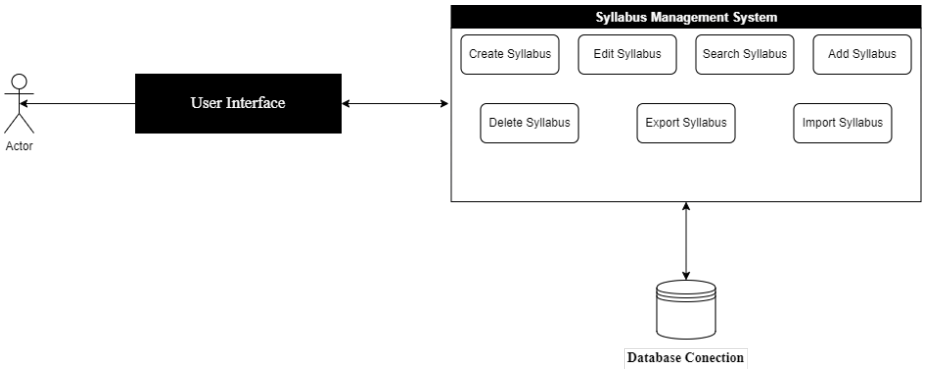

Figure 2: Database Connection

3 Requirements Specification

3.1 User Requirements

3.1.1 Functional User Requirements

- Requirement 1: Users should be able to install the desktop application through an installer without requiring additional installations.
- Requirement 2: The application should facilitate the creation of new syllabi via the program's interface, allowing users to save them locally.
- Requirement 3: The application should be able to display the syllabi templates. So that the user can fill in the templates.
- Requirement 4: Users must be able to edit previously generated syllabi, introduce new fields, and input relevant information.
- Requirement 5: The application should provide the functionality to remove unwanted fields or delete entire syllabi.
- Requirement 6: The search feature should enable users to find syllabi based on course code.
- Requirement 7: The user should be able to display the desired syllabus programs by searching via course code.
- Requirement 8: Users should be able to utilize a version control system within the application to ensure that any modifications made to a syllabus result in the creation of a new version while preserving the previous versions. This functionality is essential for tracking and reviewing modifications, providing users with details on what changes were made and when they were implemented.
- Requirement 9: The program should empower users to convert created syllabi into JSON files and export them to their computers. Additionally, it should support the import of syllabi data from JSON files, allowing users to save the imported information locally within the syllabi program.

3.1.2 Non-Functional User Requirements

- Requirement 1: The application shall be installed on a computer which has only JVM as a desktop application.
- Requirement 2: Internet connection is not required to access the database
- Requirement 3: The program shall support the Windows operating system.
- Requirement 4: The application shall not have any user login interface.
- Requirement 5: The program shall have an info section that explains the application.

3.2 System Requirements

3.2.1 Functional System Requirements

- **Installation**
  Users should be able to install the desktop application through an installer without the need for any additional programs.
- **Managing Data**
  The system shall allow users to interact with existing syllabi, enabling them to select, edit, or delete syllabus documents as needed. This functionality ensures seamless management of syllabi within the system.
- **Syllabus Database**
  Users will have the capability to create new syllabi using the program's interface or save syllabi prepared with the template to the system database.
- **JSON import/Export**
  The program should empower users to convert created syllabi into JSON files and export them to their computers. Additionally, the program should support the import of syllabi data from JSON files, enabling users to save the imported information locally within the syllabi program.
- **Search**
  The system should include a search feature, allowing users to easily find specific syllabi within the database.

3.2.2 Non-Functional System Requirements

Installation Requirements

- The Syllabus Project requires Java 17.0 or later to be installed on the target machine.
- Users should have the flexibility to choose the installation folder for the Syllabus Project.
- Users should have the flexibility to choose the installation folder for the Syllabus Project.
- The Syllabus Project must feature a simple graphical user interface to enhance user productivity and reduce complexity.
- The software should incorporate robust processes for each input to prevent system failures.

User Interaction and Usability

- Considering system memory and disk space, users are advised to manage the number of stored syllabi to optimize performance.
- User can read the Info section to get further information on how to use the application more efficiently.

Performance and Reliability

- System transactions within the Syllabus Project should occur with a delay of no more than 3 seconds to ensure responsiveness. This may change positively or negatively during implementation.
- Uninstalling and reinstalling the program should be a straightforward process for users.

Functionality

- The Syllabus Project software should enable users to edit, add, create, filter, and view syllabi without errors between the interface and the database, ensuring reliable performance.

3.3 Future Enhancements

1. Collaboration Features: In future iterations, the system could include collaboration features, allowing multiple users to work simultaneously.
2. Advanced Version Control: Future enhancements may involve more advanced version control features, such as branch management and side-by-side comparisons.
3. User Authentication: If required, a robust user authentication system could be implemented in future versions to enhance security.
4. Enhanced Compatibility: Ongoing efforts will be made to ensure compatibility with emerging desktop environments and technologies.

4 Expectation and Outcomes

4.1 Expectations

The project aims to automate manual handling of course syllabuses, currently in work documents, through a Java-based desktop app. Objectives include a user-friendly interface for syllabus templates, transparent version control, and JSON functionality. Features include data import/export, a syllabus management interface, and a version control system. Constraints involve desktop app development without web features and no user authentication. Success depends on resolving dependencies, utilizing Java libraries for JSON handling, and ensuring compatibility with common desktop environments. The Syllabus System is developed with Java using JavaFX.

4.2 Outcomes

- User can display old versions of Syllabuses.
- Only the main user can update the syllabus. When the system is edited, it will be possible to see why the old version was edited.
- The user is capable of exporting the Syllabuses as PDF.
- The program can be used in Turkish or English

References

- https://riptutorial.com/Download/javafx.pdf
- https://www.oracle.com/database/what-is-json/
- https://docs.oracle.com/javase/8/javase-clienttechnologies.htm
- https://www.geeksforgeeks.org/javafx-tutorial/