HACETTEPE UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

BM384 Software Engineering Lab - 2023 Spring

SOFTWARE REQUIREMENTS SPECIFICATION

April 9, 2023

Student Name :	**Student Number:**
Duygu Sena Öztürk	2200356056
Esad Boran	21827206
Gamze Ergin	2200356827
Mustafa Emir Peker	2200356011
Yusuf Koca	2200356013

TABLE OF CONTENTS

ICES4HU
System-Wide Requirements Specification

Introduction

A software requirements specification (SRS) document is a detailed description of the functional and non-functional requirements of a software system. It is a formal document that outlines what the software must do and how it should perform.

The SRS document includes a comprehensive list of features, capabilities, and constraints that the software must satisfy. It is used as a blueprint for designing, building, and testing the software. The SRS document is created at the beginning of the software development process and is used as a reference throughout the development lifecycle. Our software project “ICES4HU”’’s system-wise needs described in this paper. ICES4HU is an online instructor and course evaluation system specific to Hacettepe University. The software is briefly explained in seven parts. These are system-wide functional requirements, system qualities, system interfaces, business roles, system constraints, system compliance, and system documentation.

1. System-Wide Functional Requirements

1.1 Auditing:

Auditing is the process of evaluating an organization, system, or process to determine its compliance with established guidelines, standards, or requirements. It is an independent, objective review of the activities and operations of an entity to ensure that they are operating following relevant laws, regulations, policies, and procedures.

- The system will keep the IP addresses of the users to follow any anomaly in their accounts.
- The surveys should be checked if any questions are appropriate.
- If there are any used slang words in the evaluation forms, the users can be banned.
- If there are mistakes in user authorizations, they should be detected.

1.2 Authorization:

Authorization is the process of determining what actions an authenticated user is allowed to perform within a system or application. It is the second step in the process of providing access to a system or service and is essential for ensuring the security and integrity of sensitive information.

- There are different responsibilities for each different type of user. The system control access to system resources based on user roles and permissions.

1.3 Authentication:

Authentication is the process of verifying the identity of a user, system, or device. It is the first step in the process of providing access to a system or service and is essential for ensuring the security and integrity of sensitive information.

- The system is for the users that enrolled in the system. This is being checked with their mail extensions. If it belongs to Hacettepe University they can log in.

1.4 Data Management

Data management is the ability to create, read, update, and delete data within the system, as well as manage data backups and recovery. Our system holds information about courses, students, instructors, and their evaluations. Data management plays a large role in our system.

- Data storage
- Data retrieval
- Data validation
- Data update
- Data deletion

1.5 Reporting and Analytics:

The ability to generate reports and perform data analysis based on the system's data.

- After every successful evaluation all of the statics and results can be viewed by users.
- These evaluation results can be used to find the success rate in courses using analytics.

1.6 Licensing

Licensing is an important aspect of system-wide functional requirements. It refers to the rules and conditions governing the legal use of software and includes the terms and conditions for obtaining, using, and distributing software licenses.

- In this project there will be no license used.

1.7 Security

Security in software projects can be said as the ability to protect sensitive data and comply with relevant regulations and standards.

- The system holds anonymous answers from students. They should be invisible to only admin and department managers.
- User's personal information should be protected in case of any attack.
- The system should enroll users if only the admin approves it.

2. System Qualities

System qualities are an essential component of system-wide functional requirements in a software requirements specification (SRS) document. Qualities represent the URPS in the FURPS+ classification of supporting requirements.

2.1 Usability

Usability refers to the ease with which users can learn to use a system, navigate through it, and perform tasks efficiently and effectively. The usability requirements typically include the following:

Ease of learning: The system should be easy to learn and use, with clear and concise instructions and the minimal training required for users.

Navigation: The system should be easy to navigate, with clear and intuitive menus, buttons, and icons that guide users to the information and features they need.

User interface design: The system should have a user-friendly interface design, with consistent layout and formatting, use of color and font, and clear labeling and instructions.

Accessibility: The system should be accessible to users with disabilities, with features such as alternative text for images, keyboard navigation, and support for assistive technology.

Error prevention and handling: The system should prevent errors and provide clear and helpful error messages when they occur, with suggestions for how to correct them.

Efficiency: The system should allow users to perform tasks quickly and efficiently, with shortcuts and other time-saving features.

User feedback: The system should provide feedback to users, such as progress indicators, confirmation messages, and notifications, to keep them informed of the system status and progress.

Effective usability design is critical to the success of any software system, as it ensures that the system is easy to learn and use and that users can perform tasks quickly and efficiently.

2.2 Reliability

Reliability refers to the ability of a system to perform consistently and accurately over time, without unexpected failures or errors. The reliability requirements typically include the following:

Availability: The system should be available when needed, with minimum downtime or outages, and quick recovery in the event of a failure.

Fault tolerance: The system should be designed to tolerate and recover from failures, with features such as redundancy, backup systems, and error handling.

Data integrity: The system should ensure the integrity of data, with features such as data validation, error checking, and error correction.

Testing and validation: The system should be thoroughly tested and validated to ensure that it meets reliability requirements, with testing methods such as stress testing, load testing, and failure testing.

Effective reliability design is critical to the success of any software system, as it ensures that the system is available, stable, and consistent and that users can rely on it to perform as expected.

2.3 Performance

Performance refers to the ability of a system to process data and transactions quickly and efficiently and to meet the organization's performance requirements. The performance requirements typically include the following:

Response time: The system should have fast response times, with minimum delay between user requests and system responses.

Throughput: The system should have high throughput, with the ability to handle a large number of transactions or requests simultaneously.

Scalability: The system should be scalable, with the ability to handle increasing volumes of data, users, or transactions without degrading performance.

Resource utilization: The system should use resources efficiently, such as CPU, memory, and disk space, and avoid excessive usage that can lead to performance degradation.

Performance metrics: The system should have performance metrics that can be monitored and measured, such as CPU usage, memory usage, and response time, to ensure that performance requirements are being met.

Effective performance design is critical to the success of any software system, as it ensures that the system can handle the expected load, process data and transactions quickly and efficiently, and meet the organization's performance requirements.

2.4 Supportability

Supportability refers to the ability of a system to be easily supported, maintained, and updated by the organization or team responsible for it. The supportability requirements include the following:

Maintainability: The system should be designed to be easily maintained, with clear documentation, well-organized code, and modularity that allows for easy updates and changes.

Upgradability: The system should be designed to be easily upgraded, with support for new features, technology updates, and compatibility with other systems or platforms.

Compliance: The system should be compliant with relevant standards, regulations, and security protocols, to ensure that it can be properly supported and maintained.

Effective supportability design is critical to the success of any software system, as it ensures that the system can be easily maintained and updated and that users can rely on it to perform as expected.

3. System Interfaces

3.1 User Interfaces

The user interface (UI) is an important aspect of any software system, as it is the primary way in which users interact with the system.

3.1.1 Look & Feel

The UI should be visually appealing and easy to navigate, with a design that is appropriate to user expectations.

- The users could easily use the web without needing any help.

3.1.2 Layout and Navigation Requirements

- The UI should have a clear and intuitive navigation structure, with menus and navigation elements that are easy to understand and use.
- The UI should have a navigation history feature, such as a "back" button, to allow users to easily navigate back to previous screens, and evaluations.
- It should have interactive elements, such as buttons and links, that allow users to easily navigate through the system.
- It should have a clear and logical screen layout, with elements arranged in a way that is easy to understand and navigate.

3.1.3 Consistency

- The UI should have a consistent design language, which includes colors, fonts, and other design elements that are used consistently throughout the system.
- The UI should use standard UI components, such as buttons, checkboxes, and dropdown menus, that are consistent with industry standards and user expectations.
- The UI should have consistent interaction patterns, such as hover effects, button styles, and animations, that make it easy for users to understand how to interact with the system.
- The UI should use consistent terminology throughout the system, to avoid confusion and ensure that users understand the system's functions and features.

3.1.4 User Personalization & Customization Requirements

- The UI should provide customizable settings that allow users to modify the appearance and behavior of the UI.
- It should allow users to set personalized preferences, such as language, font size, and color schemes.
- It should allow users to create personalized dashboards, which display relevant information and metrics. The students can revisit their evaluations and instructors can list, and sort their surveys, etc.

3.2 Interfaces to External Systems or Devices

3.2.1 Software Interfaces

In our web project system we are using Java with Spring Framework for backend development and React for frontend development. For the database, MySQL will be used.

3.2.2 Hardware Interfaces

There is no hardware interface in the system.

3.2.3 Communications Interfaces

There will be no communications interface on this system.

4. Business Rules

Business rules, as they define the specific policies, procedures, and regulations that must be followed by the system. Business rules typically describe how the system should behave under certain conditions or circumstances, and they help ensure that the system meets the organization's business objectives.

4.1 Registration Rule

4.1.1 This is a web project that will be only for Hacettepe University employees and students. They have to prove they are a part of the University.

4.1.2 System members can be added to the system by admin. There are 4 different types of system membership. Admin, student, instructor, department manager.

4.1.3 The user can log in to the system with the desired e-mail address.

4.2 Secrecy of evaluation

4.2.1 The answers of students to surveys should not be open to all system members.

4.2.2 It can be viewed by Department Manager or Admin but instructors can not see their answers. It’s important to protect transparency in evaluation.

4.3 Code of Conduct

4.3.1 If any user uses slang words in the evaluation form, the admin should be able to ban this user. The banned user cannot submit another evaluation form to other instructors or courses.

4.3.2 Managing evaluation forms uploaded by the instructors or department manager. For example, s/he should delete the question whose context is not appropriate.

5. System Constraints

- Our system is a web application that is designed to be compatible with all web-based platforms, allowing users to access it through their preferred device or operating system. The system will be developed using the Java language, which provides a robust and reliable foundation for building complex web applications.
- For the front end, we will be utilizing React, a popular and powerful framework that allows for efficient and responsive user interfaces.
- The system's data will be managed using MySQL, a widely used relational database management system known for its scalability and reliability.
- To ensure proper version control and collaboration among the development team, changes and configuration management actions will be tracked using GitHub flow. This will allow us to maintain a centralized repository of the system's codebase and easily manage any changes or updates made to it.
- In terms of language, the system will be primarily English-based to ensure consistency.

6. System Compliance

6.1 Licensing Requirements

The software system called ICES4HU is being developed by students who are currently enrolled in the BBM384 class. No license is required for its development because it is being designed by the students themselves. In this project, we are using what we are learning in BBM382 Software Engineering lectures and BBM384 Software Engineering Laboratory class. The development team, “Winx Perileri” is working with the following coding principles and software engineering principles.

6.2 Legal, Copyright, and Other Notices

The system being developed by the student group named “Winx Perileri”. All codes belong to the team members but we are creating this project open-source. That means that anyone can use, copy, distribute, and modify the system according to their needs. As a result, users have the freedom to customize and improve the software according to their preferences, without any legal restrictions. Users must acknowledge and understand the risks associated with using an open-source system and take responsibility for any legal implications or consequences that may arise. Despite this, the group members have taken all necessary precautions to ensure that the system adheres to the highest standards of software engineering practices.

6.3 Applicable Standards

For the ICES4HU software development process, our team is committed to following a set of applicable standards to ensure the quality and efficiency of the project.

Firstly, we will adhere to coding standards, ensuring that all code is written in a consistent and readable manner. We will use IEEE coding standards to ensure that our code is easily maintainable and understandable by other developers.

Furthermore, we will be using an object-oriented design pattern for the software, complying with SOLID principles. This will ensure that our software is modular, easy to maintain, and scalable for future developments.

We will follow clean and readable code standards, ensuring that our code is easy to understand and modify.

Our team is committed to developing the project according to security and web protocols and standards. This will ensure that the software is secure and reliable for all users.

7. System Documentation

This software is designed for Hacettepe University academicians and students. You can use this system with 4 different memberships: student, admin, instructor, and department manager.

The user manual is an essential document that provides guidance and assistance to system users. It is the responsibility of the system developers to create a comprehensive user manual that outlines how to use the system, what to do if a problem arises, and answers to frequently asked questions. The user manual will be developed clearly and concisely, with easy-to-understand language that is accessible to all users, regardless of their technical knowledge.

The user manual will include step-by-step instructions on how to use the various features of the system, such as how to create an account, how to log in, and how to navigate through the different sections of the system. It will also provide troubleshooting tips and solutions to common problems that users may encounter while using the system. The user manual will include a section on frequently asked questions, which will provide answers to common queries that users may have about the system. This will help users quickly find the information they need and reduce the number of support requests.

8. Distribution of Tasks

In our software requirements specification document, we divided the subjects equally among group members.
The members and their parts in this document are listed below:

System-wide functional requirements - Yusuf Koca
System Qualities - Esad Boran
System Interfaces - Duygu Sena Öztürk
Business Roles -Mustafa Emir Peker
System Constraints, Compliance, Documentation - Gamze Ergin