Software Requirements Specification

for

Education Management System

Version 1.0

Prepared by

Omar Ahmed	202220090
Ahmed Mohammed	202220124
Ahmed Mohammed Mordi	202220309
Mohammed Abdalla	202220109
Mahmoud Mohammed Emara	202220101

Instructor:	*Mohammed Karam*
Course:	Software engineering
Teaching Assistant:	*Mohammed Karam*
Date:	*14/11/2024*

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the requirements for the Educational Management System (EMS). This system aims to provide an efficient way for managing educational courses, assignments, and communication among users, including Doctors, Teaching Assistants, and Students. It will serve as a guideline for the development team to implement features that meet stakeholder expectations and project objectives.

1.2 Document Conventions

This document uses the following conventions:

- ***Bold text**: Section headings and important terms.*
- *Italic text: Emphasis or specific notes.*
- `Monospaced text`*: Code snippets or inline technical terms.*
- *Terms such as **Doctor**, **TA**, and **Student** will be capitalized throughout the document to refer to specific user roles.*

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- ***Developers**: To understand the functional and non-functional requirements of the system.*
- ***Project Managers**: To ensure the project aligns with the scope and requirements.*
- ***Stakeholders**: To validate that the requirements meet their needs.*
- ***Testers**: To create test cases based on the defined requirements.*

1.4 Product Scope

The Educational Management System (EMS) is designed to:

- *Provide a platform for managing educational activities such as course creation, assignment handling, and grade management.*
- *Allow Doctors to create and manage courses and assignments, and track student progress.*
- *Enable Teaching Assistants to assist in grading and course management.*
- *Allow Students to register for courses, view course details, submit assignments, and check their grades.*

1.5 References

The following resources were used in the preparation of this document:

- ***Project Outline**: "Educational Management System" by Mostafa Saad Ibrahim.*

- ***IEEE Standard**: IEEE 830-1998, Recommended Practice for Software Requirements Specifications.*

2. Overall Description

2.1 Product Perspective

The Educational Management System (EMS) is a desktop application aimed at providing a basic yet functional educational platform. It is a standalone system developed initially as a desktop application using C# for the backend.

This product is designed to be a simplified version of existing Learning Management Systems (LMS), such as **Ibn Alhaitham***, but tailored for educational environments where simplicity and ease of use are prioritized. The system will help automate course management tasks and facilitate interactions between Doctors, Teaching Assistants, and Students.*

2.2 Product Functions

The main functions of the Educational Management System include:

- ***User Authentication:** Supports login and signup for Doctors, Teaching Assistants, and Students.*
- ***Course Management:***
  - *Doctors can create, edit, and delete courses.*
  - *Doctors can assign Teaching Assistants to courses.*
  - *Students can register for available courses and view registered courses.*
- ***Assignment Handling:***
  - *Doctors and Teaching Assistants can create assignments, view submissions, and grade assignments.*
  - *Students can view assignments, submit solutions, and check their grades.*

2.3 User Classes and Characteristics

The system identifies the following user classes:

- ***Doctor (Instructor):***
  - *Can create and manage courses and assignments.*
  - *Can assign grades and communicate with Teaching Assistants and Students.*
  - *Has access to view and generate course statistics.*
- ***Teaching Assistant (TA):***
  - *Assists the Doctor in managing courses.*
  - *Can help in grading assignments and providing feedback to Students*
- ***Student:***
  - *Can register for courses, view course details, and check assignments.*
  - *Can submit assignment solutions and view feedback from Doctors and TAs.*

2.4 Operating Environment

The Educational Management System will operate in the following environment:

- ***Operating System:** Windows (Windows Form App)*
- ***Development Language:** C# (using Dotnet)*
- ***Hardware Requirements:** Standard PC with a dotnet environment*
- ***Memory Requirements:** Requires minimal system memory for in-memory data storage*

2.5 Design and Implementation Constraints

***Desktop App:** .using dotnet and WF GUI.*

WF = Windows Form

2.6 User Documentation

User documentation will include:

- ***Installation Guide:** Download exe file in www.test.com*
- ***User Manual:** Detailed explanations of system features, including sample commands and usage scenarios for Doctors, TAs, and Students.*
- ***Frequently Asked Questions (FAQ):** A list of common issues and solutions for troubleshooting.*

2.7 Assumptions and Dependencies

- *The system assumes that all users have basic knowledge of computers applications.*
- *It is assumed that user data (e.g., email addresses and passwords) is input correctly and validated as per standard formats.*
- *The system depends on a dotnet environment*
- *Future extensions may require a transition from in-memory data storage to file-based or database storage for persistent data.*

3. External Interface Requirements

3.1 User Interfaces

The Educational Management System (EMS) is a desktop application.

- ***Main Menu:***
  - *Options for login and signup for Doctors, TAs, and Students.*

- ***Doctor Interface:***
    - *A main menu for Doctors with options to:*
        - *List Courses*
        - *Create Course*
        - *View Course*
        - *Log out*
    - *Course-specific menu options:*
        - *List Assignments*
        - *Create Assignment*
        - *View Assignment*
        - *Back to main menu*
- ***Teaching Assistant Interface:***
    - *A main menu with options to:*
        - *View Assigned Courses*
        - *Assist in Assignment Grading*
        - *Communicate with Doctors and Students*
        - *Log out*
- ***Student Interface:***
    - *A main menu with options to:*
        - *Register for Courses*
        - *List Registered Courses*
        - *View Course Details*
        - *Submit Assignment*
        - *Check Grades Report*
        - *Log out*

3.2 Hardware Interfaces

The EMS does not require any specialized hardware. It is designed to work on standard PCs or laptops with basic hardware capabilities.

- ***Minimum Requirements:***
    - ***Processor:*** *1 GHz or higher*
    - ***Memory:*** *2 GB RAM*
    - ***Storage:*** *100 MB available disk space*

3.3 Software Interfaces

The EMS interacts with the following software components:

- ***Windows Form App***
- ***File System (for Data Persistence):***

  - *The application may use basic text files for storing user data, course information, and assignment details when saving data to disk (optional feature).*
  - *File I/O operations will involve reading from and writing to simple text files.*
- ***Operating System:***
  - *The system is compatible with Windows.*

3.4 Communications Interfaces

The initial version of the EMS does not require internet or network connectivity. However, basic communication features are included within the application:

- ***Internal Messaging:***
  - *Users (Doctors and TAs) can send messages or announcements to each other and to Students within the application.*
  - *Messages are handled internally and are displayed in the console output.*
- ***Announcements:***
  - *Course announcements can be posted by Doctors and are visible to all Students enrolled in the course.*

Future versions may include network capabilities for online data storage and real-time communication.

4. System Features

4.1 System Feature 1: User Authentication

Description

The system supports secure user authentication, allowing users to sign up, log in, and log out. The users include Doctors, Teaching Assistants (TAs), and Students.

Priority

High

Functional Requirements

- ***4.1.1** The system shall allow users to sign up by providing a username, email, and password.*
- ***4.1.2** The system shall validate email format during sign-up.*
- ***4.1.3** The system shall encrypt user passwords before storing them in memory.*
- ***4.1.4** The system shall provide a login interface where users can enter their username and password.*
- ***4.1.5** The system shall display an error message if the login credentials are incorrect.*
- ***4.1.6** The system shall allow users to log out and return to the main menu.*

4.2 System Feature 2: Course Management

Description

This feature allows Doctors to create and manage courses, and enables Students to register for courses and view course details.

Priority

High

Functional Requirements

- ***4.2.1*** *The system shall allow Doctors to create courses by entering the course name and code.*
- ***4.2.2*** *The system shall enable Doctors to assign Teaching Assistants (TAs) to specific courses.*
- ***4.2.3*** *The system shall allow Students to register for courses from a list of available courses.*
- ***4.2.4*** *The system shall prevent Students from registering for the same course more than once.*
- ***4.2.5*** *The system shall allow Doctors and TAs to view a list of Students registered in each course.*
- ***4.2.6*** *The system shall allow Students to view their registered courses and access course summaries.*

4.3 System Feature 3: Assignment Handling

Description

The assignment feature allows Doctors and TAs to create, view, and grade assignments, and enables Students to submit their assignment solutions.

Priority

High

Functional Requirements

- ***4.3.1*** *The system shall allow Doctors and TAs to create assignments by specifying the title, description, and due date.*
- ***4.3.2*** *The system shall allow Students to view the list of assignments for their registered courses.*
- ***4.3.3*** *The system shall allow Students to submit solutions for assignments before the due date.*

- ***4.3.4*** *The system shall allow Doctors and TAs to view submitted solutions and provide grades and feedback.*
- ***4.3.5*** *The system shall generate a grades report for each course, showing Student performance on assignments.*

4.4 System Feature 4: Communication and Announcements

Description

This feature allows users (Doctors and TAs) to communicate via internal messaging and make announcements for course updates.

Priority

Medium

Functional Requirements

- ***4.4.1*** *The system shall allow Doctors to post announcements visible to all Students in a course.*
- ***4.4.2*** *The system shall allow TAs to reply to announcements made by Doctors.*
- ***4.4.3*** *The system shall provide an inbox feature where users can view internal messages from other users.*
- ***4.4.4*** *The system shall notify users of new messages or announcements upon login.*

4.5 System Feature 5: Grades Report

Description

The Grades Report feature provides Students with an overview of their performance across all registered courses and assignments.

Priority

Medium

Functional Requirements

- ***4.5.1*** *The system shall allow Students to view a grades report showing the total score for each registered course.*
- ***4.5.2*** *The system shall display detailed assignment scores for each course, including submitted assignments and those pending evaluation.*
- ***4.5.3*** *The system shall allow Doctors and TAs to export a course’s grades report for analysis or record-keeping.*

5. Other Nonfunctional Requirements

Nonfunctional requirements define the system's overall qualities and constraints, focusing on aspects like usability, reliability, and scalability. This section outlines requirements that are not directly related to specific functionalities but are critical for the user experience and overall performance.

5.1 Performance Requirements

Performance requirements specify how well the software must perform under certain conditions, including response times, throughput, and resource usage.

5.2 Safety Requirements

Safety requirements ensure that the software operates without causing harm to users, systems, or data. These are crucial in systems where failures can lead to significant risks or damages.

5.3 Security Requirements

Security requirements focus on protecting the software from unauthorized access, data breaches, and other security threats.

5.4 Software Quality Attributes

Software quality attributes describe the desired qualities of the system, often using the ***ISO/IEC 25010*** *standard, which includes aspects like reliability, maintainability, and usability.*

5.5 Business Rules

Business rules define specific policies, procedures, or constraints that the software must adhere to in order to align with the organization's processes and goals.

6. Other Requirements

This section outlines additional requirements that do not fit into the functional or non-functional categories but are essential for the successful deployment of the system.

6.1 Legal and Regulatory Requirements

- ***6.1.1*** *The system must comply with data protection and privacy laws, such as GDPR, to ensure that user information is securely stored and handled.*
- ***6.1.2*** *User data (e.g., email, personal information) must not be shared without consent, and encryption should be used to protect sensitive data.*

6.2 Performance Requirements

- ***6.2.1*** *The system should be able to handle at least 100 simultaneous users without performance degradation on standard hardware.*
- ***6.2.2*** *User actions (e.g., login, course registration, assignment submission) should have a response time of less than 2 seconds.*

6.3 Security Requirements

- ***6.3.1*** *The system shall implement password encryption to protect user credentials.*
- ***6.3.2*** *The system shall enforce strong password policies (minimum 8 characters, including numbers and special characters).*
- ***6.3.3*** *User sessions should expire after a period of inactivity (e.g., 15 minutes) for added security.*

6.4 Maintainability Requirements

- ***6.4.1*** *The codebase should follow modular design principles to facilitate easy updates and bug fixes.*
- ***6.4.2*** *The system should be well-documented, including in-code comments and a developer guide.*

6.5 Portability Requirements

- ***6.5.1*** *The system shall be compatible with major operating systems (Linux, Windows, macOS).*
- ***6.5.2*** *The system should not rely on any OS-specific features, ensuring ease of portability.*

Appendix A: Glossary

This section defines key terms used throughout the SRS document.

- ***LMS (Learning Management System):*** *A software platform designed to manage educational content, track student progress, and facilitate communication between instructors and students.*
- ***Doctor:*** *A user role representing the instructor or course creator.*
- ***Teaching Assistant (TA):*** *A user role that assists the Doctor in managing courses and assignments.*

- ***Student:*** *A user role that interacts with the system by registering for courses, submitting assignments, and viewing grades.*
- ***Assignment:*** *A task or project created by the Doctor or TA that Students must complete and submit.*
- ***Course:*** *A collection of educational content and assignments, managed by a Doctor and potentially supported by TAs.*

Appendix B: Analysis Models

This section provides diagrams and models representing the system's architecture and user interactions. For a console-based system like this, we suggest including:

- ***Use Case Diagrams:*** *Showing the interactions between user roles (Doctor, TA, Student) and system features (e.g., login, course management, assignment submission).*
- ***Flowcharts:*** *Illustrating the sequence of actions for key features such as user login, course registration, and assignment handling.*
- ***Data Flow Diagrams (DFDs):*** *Representing the flow of information between the system components.*

You can create these diagrams using tools like draw.io or Lucidchart.

Appendix C: To Be Determined List

This section lists any details that are not finalized and need further clarification or decisions.

Item	***Description***
Database Integration	*Future versions may include file-based or database storage for data persistence.*
Network Capabilities	*The possibility of adding online features for real-time collaboration.*
User Interface Enhancements	*A potential transition from a console-based to a GUI-based application.*