Software Requirements Specification

for

Flashcard Studying System (EduCards)

Version 2

Prepared by: Marko Ilioski, 211518
Nina Kraleva, 211502
Petar Pejoski, 211551
Martin Patcev, 211524
Drin Kadriu, 211505

Group: FIVE

30.03.2025

Table of Contents

1. Introduction

As technology becomes an integral part of education, both students and teachers rely more on digital tools for learning and progress tracking. However, managing assignments, monitoring student performance, and ensuring effective engagement can be challenging without a centralized system. To address these challenges, the **Flashcard Studying System - EduCards** has been introduced. This system is designed to enhance the learning experience by providing a structured platform where students can access course materials, complete assignments (quizzes), and track their progress, while teachers can efficiently monitor student performance, provide feedback, and adapt their teaching methods accordingly.

1.1. Purpose

This document provides an overview of requirements, limitations, and critical areas, fostering a clear understanding and successful implementation. It is essential for developers, stakeholders and testers involved in the online flashcard studying system software development.

Importance for Developers: Developers play a crucial role in implementing the software. They should carefully review this document to understand the requirements, restrictions, and risks. This ensures correct development and addresses concerns.

Importance for Testers: Testers ensure software quality and functionality. They should read this document to understand requirements and limitations, creating comprehensive test plans. They should note any risks or concerns mentioned.

Importance for Stakeholders: Stakeholders should review this document to ensure their needs are considered and provide feedback. It helps align requirements and expectations.

1.2. Scope

The software to be created is called **“EduCards”** and is a platform designed to enhance the learning experience for both students and teachers. It will facilitate interactive learning by allowing students to access study materials, complete quizzes, and track their progress. Teachers will be able to monitor student performance, provide feedback, and adapt their teaching strategies accordingly.

1.3. Definitions, acronyms, abbreviations

- ***Progress Tracking*** – A feature that allows students to monitor their learning progress and receive insights on their strengths and areas for improvement.
- ***Assignment Management*** – A system that enables teachers to assign and review student work efficiently.
- ***Real-Time Analytics*** - Data-driven insights into student performance, helping teachers adjust their teaching strategies based on individual or class-wide needs.
- ***Interactive Learning*** - A method of studying that incorporates digital tools such as quizzes, flashcards, and multimedia content to enhance student engagement.
- ***User Dashboard*** - A personalized interface where students can view their assignments, progress reports, and study materials, while teachers can track student performance and manage coursework.
- ***Educator*** - A teacher or instructor who uses the platform to assign tasks, assess student performance, and provide feedback.
- ***Student*** - A learner who uses the system to complete assignments, track progress, and engage with study materials.

1.4. References

- https://press.rebus.community/requirementsengineering/back-matter/appendix-c-ieee-830-template/ - template link
- https://press.rebus.community/requirementsengineering/back-matter/appendix-d-ieee-830-sample/ - template sample
- Tim Chagnon, Joe Kokinda, Andrew Meinert, Don Naegely “Software Requirements Specification

1.5. Overview

The latter part of the document is divided into two sections, Section 2 and Section 3. Section 2 includes an overall description of the system such as the product perspective, the user interface of the software, features that the system will offer and constraints on the requirements. Section 3 focuses on the specific requirements and external interfaces of the system.

2. Overall Description

2.1. Product perspective

The system enables teachers to upload educational materials and automatically generates study flashcards to help students learn, while creating a competitive environment with a leaderboard system.

Software interface - EduCards will run on Windows, macOS, and Linux operating systems. Additional software components are yet to be chosen.

Hardware interface - The software will run on a CPU yet to be chosen. The main hardware component will be the user's computer.

2.1.1. User interfaces

User - the user interface should prioritize simplicity, such that 99% of the users will face no difficulties and won't require assistance when answering quizzes and attaching documents.

2.2. Product features

- ***User Registration*** **-** The software allows users to create accounts and register with their personal information.
- ***User Login*** **-** Registered users can log into the software using their credentials to access their accounts and use the features provided.
- ***Profile Management*** **-** Users can manage their profiles by updating personal details, such as name, email address, and profile picture.
- ***Document Upload*** **-** The software includes an upload feature that allows professors/teachers to upload study materials.
- ***Flashcard Generation*** **-** The software generates flashcards based on study material uploaded for a specific subject/course.
- *Quiz* – The software creates quizzes for a specific subject/course made from generated flashcards.
- ***Leaderboard Overview*** – The software includes a leaderboard where users can track the progress and see scores from quizzes.

2.3. User characteristics

- **Professors/Teaching Assistants:** Users who can upload study materials for their subject/course. They should know how to upload and edit documents. They most likely will not require any training to use the program.

- **Students:** Users should be enrolled in the specific subject/course so they can access the study materials in the form of quizzes with flashcards. They most likely will not require any training to use the program.

2.4. Constraints

- The travel grant system must be compatible on the operating systems Windows, macOS, and Linux.
- The travel grant system must be compatible on all desktop browsers (Chrome, Firefox, Safari, etc.).
- The system should be capable of supporting a minimum of 500 simultaneous users during peak usage periods without significant performance degradation.

3. Specific Requirements

3.1. Functional Requirements

1. The system shall support registration for both teachers and students, allowing them to create accounts with relevant information.
2. The system shall display an error message if the input information in the field is invalid.
3. The system shall implement token-based authentication to ensure secure access to the application for all users.
4. The system shall assign distinct roles for teachers and students, with teachers having administrative privileges over documents and students having viewing rights.
5. The system shall allow teachers to upload PDF documents to the platform for sharing and access by students.
    5.1. The system shall have a separate field where the studying material can be uploaded.
6. The system shall allow the user to choose if they want to use Macedonian, Albanian or English while using the application.
7. The system shall provide a document browsing feature, where users (teachers and students) can view a list of available documents with filtering options (e.g., by subject, document type).
8. The application shall generate flashcards with questions and optional answers based on the uploaded study material.
9. The user shall be able to choose on what subject/course they would like to answer flashcards.
10. The system shall enable the user to track the progress of the students in their subject/course.
11. The system shall enable the user to view the points after a quiz.
12. The system shall allow teachers to manage their documents by providing functionality to update or delete their uploaded documents.
13. The system shall allow students to bookmark flashcards for later review, with a dedicated view to access all bookmarked flashcards.

14. The system shall allow teachers to set a time limit for quizzes, after which the quiz auto-submits with answers provided up to that point.
15. The system shall provide a dark mode option, with user preferences saved across sessions.

3.2. Performance Requirements

1. The system shall support a minimum of 500 simultaneous users accessing it without a noticeable decrease in performance or response time.
2. The system shall have an API response time under 500ms for most operations.
3. The system should support uploading documents up to 20MB size.
4. The system should complete flashcard generation within 30 seconds for average documents.
5. The system shall generate and download exports within 5 seconds.
6. The system should be developed using React for front-end development, Python for back-end development, and PostgreSQL for the database.

3.3. Security Requirements

1. The system shall implement token-based authentication for all API endpoints to ensure secure access control.
2. The system shall use role-based access control (RBAC) to differentiate access levels, such as teachers vs. students, based on user roles.
3. The system shall ensure the secure storage of user credentials and uploaded documents, employing strong encryption methods.
4. The system shall securely handle OpenAI API keys, ensuring they are not exposed in client-side code or logs.
5. The system shall perform thorough validation of all user inputs to prevent security vulnerabilities like SQL injection and cross-site scripting (XSS).
6. The system shall provide secure error handling by delivering error responses that do not expose sensitive information to the user.

3.4. Quality Requirements

1. The system shall have comprehensive logging for debugging and monitoring purposes, capturing key events and errors.
2. The system shall implement regular database backup procedures to ensure data

integrity and recovery.

3. The system shall include a mechanism for applying updates and patches, ensuring smooth and timely maintenance of the system.

4. The system shall incorporate performance and error monitoring to detect and resolve issues proactively.

5. The system shall be designed to scale effectively with increasing user load, ensuring consistent performance as the user base grows.