

Software Requirements Specification (SRS) for Student Notes Repository Web Application

Table of Contents

1. Introduction

- Purpose
- Scope
- Definitions, Acronyms, and Abbreviations
- References

2. Overall Description

- Product Perspective
- Product Features
- User Classes and Characteristics
- Operating Environment
- Design and Implementation Constraints
- Assumptions and Dependencies

3. Specific Requirements

- Functional Requirements
- Non-Functional Requirements
- Interface Requirements

4. Appendices

- Glossary
 - Analysis Models
 - Issues List
-

1. Introduction

1.1 Purpose

The purpose of this SRS is to provide a detailed description of the Student Notes Repository Web Application, which will allow students from various backgrounds to upload, download, and share their notes. This document will outline the functional and non-functional requirements, user characteristics, and system constraints.

1.2 Scope

The web application will serve as a centralized repository for student notes. Key features include note uploading, downloading, sharing with friends or the public, and note organization. The application will support various file formats and provide a user-friendly interface for students.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS:** Software Requirements Specification
- **UI:** User Interface
- **API:** Application Programming Interface

1.4 References

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications
- W3C Web Content Accessibility Guidelines (WCAG)

2. Overall Description

2.1 Product Perspective

The Student Notes Repository Web Application is a standalone product designed to be accessed via web browsers. It will be built using modern web technologies and will integrate with cloud storage services for note storage.

2.2 Product Features

- User Registration and Authentication
- Note Uploading and Downloading

- Note Sharing (Public and Private)
- Note Organization (by subject, date, etc.)
- Search and Filter Notes
- User Profiles
- Rating and Commenting on Notes
- Notifications for New Notes or Updates
- Mobile-Friendly Interface
- Secure Data Storage and Transfer
- Version Control for Notes
- Integration with Cloud Storage Services
- Collaborative Note Editing

2.3 User Classes and Characteristics

- **Students:** Primary users who will upload, download, and share notes.
- **Teachers:** Can upload and share notes, provide additional resources.
- **Administrators:** Manage user accounts, monitor content, ensure compliance with guidelines.

2.4 Operating Environment

- Web Browsers: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge
- Operating Systems: Windows, macOS, Linux, iOS, Android

2.5 Design and Implementation Constraints

- The application must comply with data protection regulations such as GDPR.
- The application should support multiple languages.
- Scalability to handle a large number of users and notes.

2.6 Assumptions and Dependencies

- Users have access to the internet and a modern web browser.
- Cloud storage service availability and reliability.
- User willingness to create and manage their accounts.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 User Registration and Authentication

- Users must be able to register using an email address or social media accounts.
- Users must be able to log in using their credentials.

3.1.2 Note Uploading

- Users must be able to upload notes in various formats (PDF, DOCX, TXT).
- Users must be able to categorize notes by subject and tags.

3.1.3 Note Downloading

- Users must be able to download notes uploaded by themselves or shared by others.

3.1.4 Note Sharing

- Users must be able to share notes publicly or with specific users.
- Users must be able to set permissions for shared notes (view, edit).

3.1.5 Search and Filter

- Users must be able to search for notes by keyword, subject, or author.
- Users must be able to filter notes by date, rating, or popularity.

3.1.6 User Profiles

- Users must have profiles where they can manage their uploaded notes and see their activity.

3.1.7 Rating and Commenting

- Users must be able to rate and comment on notes shared by others.

3.1.8 Notifications

- Users must receive notifications for new notes, comments, or updates to their shared notes.

3.1.9 Collaborative Note Editing

- Multiple users must be able to edit a note simultaneously with changes tracked.

3.2 Non-Functional Requirements

3.2.1 Performance

- The system should support up to 10,000 concurrent users without performance degradation.
- Notes should upload/download within 5 seconds on a high-speed internet connection.

3.2.2 Security

- User data must be encrypted in transit and at rest.
- Regular security audits must be conducted to identify and fix vulnerabilities.

3.2.3 Usability

- The interface must be intuitive and easy to navigate.
- The application must be accessible to users with disabilities.

3.2.4 Reliability

- The system must have 99.9% uptime.
- Data backups must be taken daily and stored securely.

3.3 Interface Requirements

3.3.1 User Interface

- The application must have a responsive design that works on both desktops and mobile devices.
- The application must have a clean and modern UI design.

3.3.2 API Interface

- An API must be provided for programmatic access to the notes repository.
- The API must follow RESTful principles.

4. Appendices

4.1 Glossary

- **Concurrent Users:** Number of users using the application at the same time.
- **Responsive Design:** Design that adapts to different screen sizes and orientations.

4.2 Analysis Models

- Use case diagrams
- Entity-relationship diagrams

4.3 Issues List

- Identify and document any known issues or challenges in the development of the application.

This SRS provides a comprehensive guide to the development of the Student Notes Repository Web Application, ensuring that all stakeholders have a clear understanding of the project requirements and expectations.