Software Requirements Specification for

Online Student Materials Website

Table of Contents

1. Introduction 2

1.1 Purpose of this document 2

1.2 Scope of Project 2

1.3 Intended Audience 2

1.4 References 2

1.5 Overview 2

2. General Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.2 User Classes and Characteristics 2

2.3 Operating Environment 2

2.4 General Constraints 2

2.6 Assumptions and Dependencies 2

3. Specific Requirements 2

3.1 Functional Requirements 2

3.1.1 User Interface

3.1.2 Architecture

3.1.3 Data Entry

3.1.4 Report Generation

3.2 Performance Requirements 2

3.3 Design Constraints 2

3.4 Security Requirements 2

3.5 Reliability 2

# Introduction

## Purpose of this document

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed overview of our web-based platform, designed to enable students from various universities to access and share educational materials related to their curriculum. This includes syllabi, textbooks, lecture notes, exam papers, and other relevant study materials. The document outlines functional and non-functional requirements, intended users, system features, constraints, and interaction with external applications. This SRS aims to ensure that developers, project managers, and stakeholders have a clear understanding of the system's capabilities, facilitating a streamlined development process that meets the project's objectives and user needs.

## Scope of Project

The project, herein referred to as "EduResource Hub," is a web-based platform that allows university students to effortlessly search, download, and upload academic materials corresponding to their specific course requirements. The website will feature a comprehensive database of educational resources categorized by university, semester, course, and type of material. Key functionalities include user registration and authentication, advanced search capabilities, material upload and download, user feedback systems, and administrative moderation tools. The scope encompasses the development of a responsive web application that supports diverse user devices, ensuring accessibility and usability across different platforms.

## Intended Audience

The primary audience for the EduResource Hub includes:

* **University Students:** Seeking access to a wide range of study materials for their academic courses.
* **Educational Institutions:** Universities and colleges that wish to make their curriculum materials easily accessible to their students.
* **Faculty Members:** Looking to share educational content with students and enhance the learning experience.
* **Developers and Technical Staff:** Responsible for developing, maintaining, and updating the website.
* **Project Stakeholders:** Including investors, educational policymakers, and anyone interested in the project's success and alignment with educational standards.

## References

Here, list any documents, websites, standards, or other SRS documents that were used as references in the preparation of this SRS.

## Overview

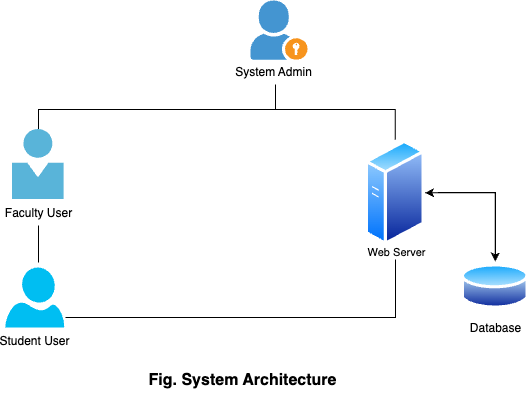
The remainder of this document is structured as follows: Section 2 details the overall description of the project, including product perspective, product functions, user characteristics, constraints, assumptions and dependencies. Section 3 elaborates on the specific requirements, including system features, user interaction, functional requirements, performance requirements, and security requirements. Appendices and indexes provide supplementary information and a glossary of terms used throughout the document.

# General Description

Building upon the structure outlined in the introduction, the General Description section of the SRS document further elaborates on the EduResource Hub platform. This section aims to provide a comprehensive understanding of the product's environment, functionality, user interaction, and constraints.

## Product Perspective (Add Image)

EduResource Hub is designed as a self-contained, web-based platform that integrates seamlessly with existing educational ecosystems. It operates independently but can potentially link with university databases and educational resource repositories to facilitate the exchange of information. The system is envisioned as a central hub for academic resources, enabling students to find and share materials without navigating through multiple systems.

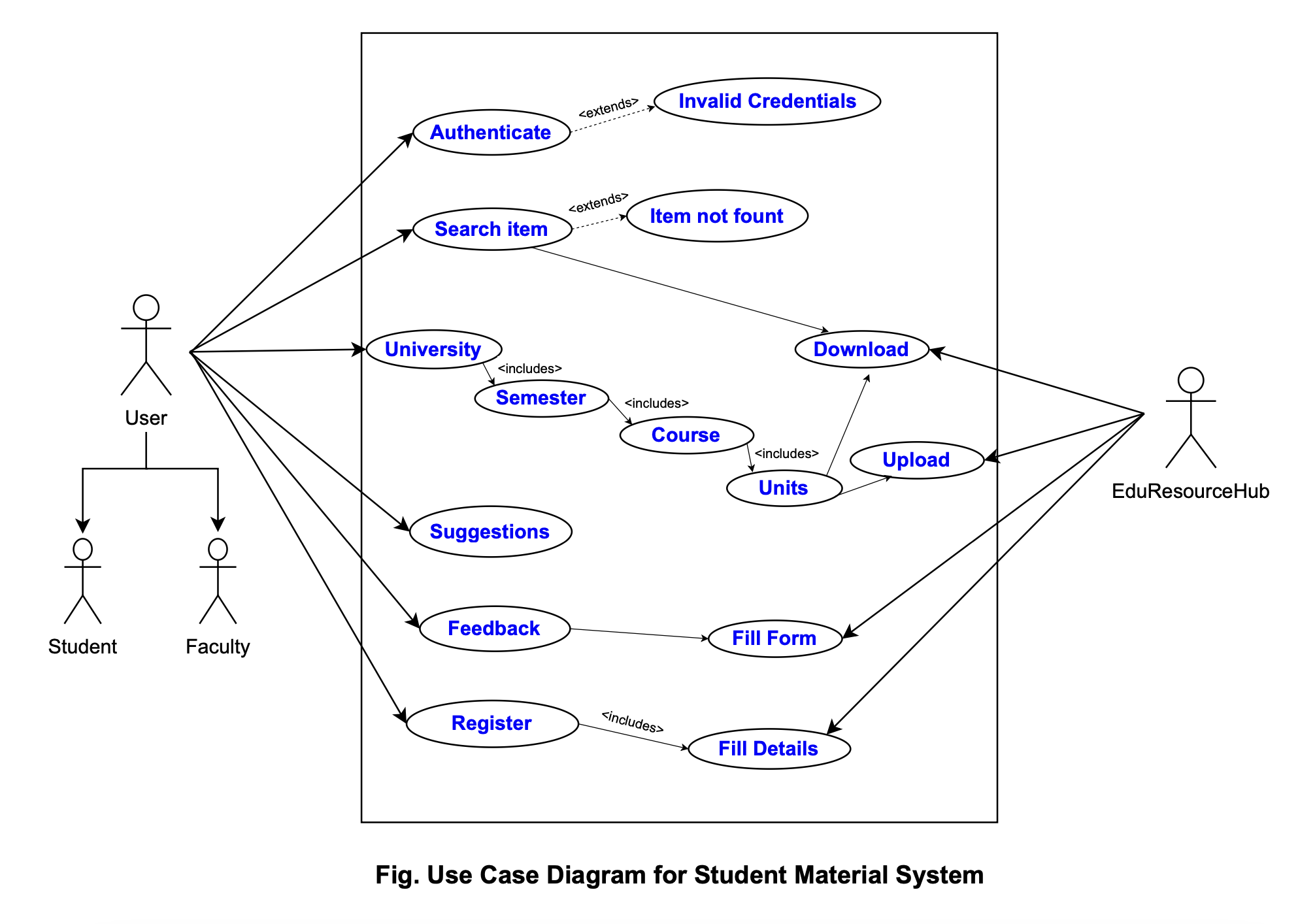


Its architecture is designed to support scalability, allowing for the addition of new universities, courses, and materials as the platform grows. Interoperability with educational standards and systems (e.g., Learning Tools Interoperability (LTI)) is considered to enhance its utility and adoption.

## Product Functions

EduResource Hub offers a wide range of functionalities tailored to its users' needs:

* **User Registration and Authentication:** Secure sign-up/login mechanisms for students, faculty, and administrators.
* **Advanced Search Capabilities:** Users can search for materials based on university, semester, course, and document type.
* **Material Upload and Download:** Facilitates the uploading and downloading of study materials, including syllabi, lecture notes, textbooks, and exam papers.
* **Course and Material Management:** Allows administrators and faculty to manage courses, including adding new materials and moderating user uploads.
* **User Profiles and Customization:** Users can customize their profiles and settings to receive recommendations and updates on relevant materials.
* **Feedback and Rating System:** Users can rate materials and provide feedback, enhancing the quality and relevance of resources.



## User Classes and Characteristics – Change to Table

* **Students:** Seeking convenient access to study materials. They require a user-friendly interface to quickly find and download content.
* **Faculty Members:** Interested in sharing educational content with their students and monitoring the usage and feedback of their materials.
* **Administrators:** Responsible for managing the platform's content, user accounts, and system security. They need robust tools to oversee platform operations effectively.
* **Technical Users:** Include developers and IT staff who maintain and update the website. They are interested in system performance, scalability, and integration capabilities.

## Operating Environment

|  |  |  |
| --- | --- | --- |
| Category | **Requirement** | **Description** |
| **Device** | Mobile browser, PC browser | Website should be accessible and functional on both mobile and PC browsers |
|  | Minimum screen resolution | 320x480 for mobile, 1024x768 for PC |
|  | Operating systems | Android 5.0+, iOS 10+, Windows 10+, macOS 10.12+ |
| **Internet Connection** | Download speed | Minimum of 2 Mbps for efficient browsing and downloading |
|  | Upload speed | Minimum of 1 Mbps for uploading materials |
|  | Connection stability | Reliable internet connection with minimal latency |
| **Hardware** | Mobile device | Minimum of 2GB RAM, 16GB storage |
|  | PC | Minimum of 4GB RAM, 50GB storage |
| **Software** | Web browser | Modern browser with JavaScript and cookies enabled |
|  | PDF reader | Ability to open and view PDF files |
|  | File compression/decompression software (optional) | For uploading/downloading compressed files (e.g., ZIP, RAR) |
| **Skills** | Basic computer literacy | Ability to navigate websites, search for information, upload/download files |
|  | Email address | Required for account creation and communication |
| **Accessibility** | Screen reader compatibility | Website should be accessible to users with visual impairments |
|  | Keyboard navigation | Website should be fully navigable using keyboard controls |
|  | Color contrast | Website should meet WCAG guidelines for color contrast |

## General Constraints

* **Compliance and Security:** Must comply with data protection laws (e.g., GDPR) and implement robust security measures to protect user data.
* **Internet Dependency:** Full functionality requires an internet connection, limiting access in areas with poor connectivity.
* **Content Quality Control:** Ensuring the accuracy and reliability of uploaded materials poses a significant challenge.
* **Intellectual Property Rights:** Must navigate copyright laws and university policies regarding the distribution of educational materials.

## Assumptions and Dependencies

* **User Participation:** The platform's success is contingent upon active participation from both students and faculty in uploading and sharing materials.
* **Stable Internet Access:** Assumes users have consistent access to the internet to utilize the platform's features.
* **University Cooperation:** The platform may depend on partnerships with universities to ensure a comprehensive database of materials and to facilitate integration with existing systems.
* **Technological Advancements:** Assumes that the current technological environment will remain stable, without significant disruptions that could necessitate major platform overhauls.

# Specific Requirements

Specific requirements in an SRS outline the detailed, precise functionalities and limitations of a system. They dive deeper than the broader statements of functional requirements, providing concrete specifications for each feature. Instead of stating "search for materials," a specific requirement might detail supported search filters, preview types, and sorting options.

## Functional Requirements

### User Interface

* **Intuitive Navigation:** Clear and concise menu options, allowing users to easily navigate between different sections of the website (e.g., Home, Search, Upload).
* **Responsive Design:** Interface automatically adjusts to the screen size and resolution of the device being used.
* **Searching for Resources and Browsing Experience:** Search materials by university, semester, course, and file type. Preview documents or explore detailed metadata for other files. Refine your search with sorting and filtering by upload date, rating, or popularity.
* **Uploading with Ease:** Provide a user-friendly interface for uploading study materials. Implement file size and format restrictions (e.g., maximum size, allowed file types). Offer optional fields for additional information (e.g., keywords, learning objectives).
* **Downloading:** Enable easy download of materials in their original format. Track download statistics for content analysis and user behavior insights. Implement a download queue or history management system.

* **User Profile:** update your profile, view uploaded materials and download history, and manage your contributions with ease.
* **Community Features (Optional):** update your profile, Implement a rating system for uploaded materials. Include a forum or discussion board for students to exchange questions and insights. Offer a "like" or "bookmark" feature for saving favorite materials.

### Architecture

Client

Server

Database

**3-Tier Architecture**

* Implement a secure and scalable three-tier architecture using appropriate technologies.
* Employ a relational database (e.g., MySQL) to store user data, material metadata, and uploaded files.
* Utilize a web server (e.g., nginx) and a programming language (e.g., NodeJS) to develop the website backend.
* Consider implementing a caching mechanism for improved performance and scalability.
* Design the architecture to be modular and easily adaptable for future functionalities.

### Data Entry

* Establish data entry protocols for ensuring consistency and quality of uploaded materials.
* Implement validation rules to prevent invalid data from entering the system.
* Consider user-friendly data entry methods for uploading materials
* Design forms for metadata entry that are intuitive and efficient to complete.

### Report Generation

* Develop reports (optional) to track website usage, user activity, and material popularity.
* Generate reports for administrators to analyze user behavior and content trends.
* Offer downloadable reports in various formats (e.g., PDF, CSV).

## Performance Requirements

<The data dictionary defines the composition of data structures and the meaning, data type, length, format, and allowed values for the data elements that make up those structures. In many cases, you're better off storing the data dictionary as a separate artifact, rather than embedding it in the middle of an SRS. That also increases its reusability potential in other projects.>

## Design Constraints

<The data dictionary defines the composition of data structures and the meaning, data type, length, format, and allowed values for the data elements that make up those structures. In many cases, you're better off storing the data dictionary as a separate artifact, rather than embedding it in the middle of an SRS. That also increases its reusability potential in other projects.>

## Security Requirements

<If your application will generate any reports, identify them here and describe their characteristics. If a report must conform to a specific predefined layout you can specify that here as a constraint, perhaps with an example. Otherwise, focus on the logical descriptions of the report content, sort sequence, totaling levels, and so forth, deferring the detailed report layout to the design stage.>

## Reliability

<If relevant, describe how data is acquired and maintained. State any requirements regarding the need to protect the integrity of the system's data. Identify any specific techniques that are necessary, such as backups, checkpointing, mirroring, or data accuracy verification. State policies the system must enforce for either retaining or disposing of data, including temporary data, metadata, residual data (such as deleted records), cached data, local copies, archives, and interim backups.>

them at the end.>