Online Educational Website

Software Requirements Specification

08-04-2024

Hardik Mishra

Prepared for
Continuous Assessment 3
Spring 2024

Revision History

Date	Description	Author	Comments
<date>	<Version 1>	<Your Name>	<First Revision>

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the requirements for the development of the EduTech website. It is intended for software engineers, developers, and other stakeholders involved in the design and implementation of the software product. The SRS serves as a guideline for understanding the scope and functionality of the website.

1.2 Scope

1.2.1 Software Product Overview

The software product to be produced is the EduTech website, an online platform that offers courses on various subjects. The website will provide users with the ability to browse courses, enroll in courses, and interact with instructors and other students.

1.2.2 What the Software Will Do

- Provide a user-friendly interface for browsing and searching courses
- Allow users to enroll in courses and track their progress
- Provide a platform for instructors to create and manage courses
- Facilitate communication between instructors and students

1.2.3 What the Software Will Not Do

- The software will not provide certification or accreditation for completed courses
- The software will not offer courses in all subjects; it will focus on a specific set of topics

1.2.4 Application of the Software

The software is intended to be used by students seeking to learn new skills or enhance their knowledge in specific subjects. It is also intended for instructors who wish to create and offer courses to a wider audience.

1.3 Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification
- HTML: Hypertext Markup Language
- CSS: Cascading Style Sheets
- JS: JavaScript

1.4 References

- IEEE Guide to Software Requirements Specifications
- HTML Specification
- CSS Specification
- JavaScript Documentation
- Bootstrap Classes
- Angular

1.5 Overview

The rest of the SRS document contains detailed requirements for the EduTech website, including functional and non-functional requirements. It is organized into sections that cover different aspects of the software product, such as user interfaces, database design, and security. Each section provides specific requirements that must be met during the development of the website.

2. General Description

2.1 Product Perspective

The EduTech website is a standalone product that is not directly related to any other existing products or projects. However, it may interact with external systems such as payment gateways for processing transactions or third-party authentication services for user login.

2.2 Product Functions

The software will perform the following key functions:

- User registration and login
- Course browsing and searching.
- Enrolment in courses
- Course creation and management for instructors
- Communication between instructors and students

2.3 User Characteristics

The eventual users of the EduTech website will vary in their technical proficiency, educational backgrounds, and learning objectives. Users may include:

- Novice learners seeking introductory courses.
- Intermediate learners looking to deepen their knowledge in specific subjects.
- Advanced learners pursuing advanced topics or certifications.
- Instructors who are subject matter experts and course creators

2.4 General Constraints

The development of the EduTech website is subject to the following constraints:

- Compliance with web standards and best practices for usability, accessibility, and security
- Availability of hosting infrastructure to support the website's performance and scalability requirements.
- Integration with external services such as payment gateways and third-party APIs
- Compatibility with modern web browsers and devices
- Adherence to budget and timeline constraints for development and deployment

2.5 Assumptions and Dependencies

The requirements stated in this SRS are based on the following assumptions and dependencies:

- Availability of required software and hardware components, including web servers, databases, and development tools
- Accessibility of external resources such as course materials, media assets, and instructional content
- Compatibility with third-party services and APIs for features like payment processing and authentication
- Stability of underlying technologies and frameworks used for website development.

- Cooperation and collaboration from stakeholders, including users, instructors, and administrators, throughout the development process.

3. Specific Requirements

This section outlines the detailed functional and non-functional requirements of the EduTech website. Each requirement is categorized, numbered, and structured according to its importance and stability.

3.1 Functional Requirements

3.1.1 User Management

a) **User Registration**: Users must be able to create accounts by providing necessary information such as username, email, and password.

b) **User Authentication**: Registered users should be able to log in securely using their credentials.

3.1.2 Course Management

a) **Course Creation**: Instructors must be able to create new courses, including adding course materials, descriptions, and prerequisites.

b) **Course Enrollment**: Users should be able to enroll in available courses, subject to any enrollment restrictions set by instructors.

c) **Communication**: Users and instructors must have means to communicate within the platform, such as discussion forums or messaging features.

3.1.3 Content Management

a) **Content Upload**: Instructors should be able to upload various types of course content, including text, images, videos, and documents.

b) **Content Organization**: Content within courses should be organized logically, allowing users to navigate easily between sections and modules.

3.2 Non-Functional Requirements

3.2.1 Performance

a) **Response Time**: The website must respond to user interactions within acceptable time frames, typically less than 2 seconds.

b) **Scalability**: The system should be scalable to accommodate increasing numbers of users and courses without significant degradation in performance.

3.2.2 Security

a) **Data Encryption**: User data, including passwords and personal information, must be stored and transmitted securely using encryption protocols.

b) **Access Control**: Role-based access control (RBAC) should be implemented to restrict access to sensitive features and information based on user roles (e.g., admin, instructor, student).

3.2.3 Usability

a) **Intuitive Interface**: The user interface should be intuitive and user-friendly, allowing users to navigate the website easily and find relevant information.

b) **Accessibility**: The website should adhere to accessibility standards, ensuring that users with disabilities can access and use the platform effectively.

3.2.4 Reliability

a) **System Uptime**: The system should be available and operational 24/7, with minimal downtime for maintenance and updates.

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface of the EduTech website should be intuitive, visually appealing, and easy to navigate. It should support the following features:

- **Login Page**: A simple and secure login page where users can enter their credentials to access the platform.
- **User Dashboard**: A personalized dashboard for each user displaying enrolled courses, progress, notifications, and account settings.
- **Course Pages**: Clear and organized course pages with sections for course materials, discussions, assignments, and quizzes.
- **Content Viewer**: Interfaces for viewing various types of content, including text, images, videos, and documents.
- **Communication Tools**: User-friendly interfaces for communication, such as messaging systems, discussion forums, and chat features.

3.1.2 Hardware Interfaces

The EduTech website should be accessible through standard hardware devices commonly used for web browsing, including:

- **Desktop Computers**: Support for popular web browsers such as Chrome, Firefox, Safari, and Edge.
- **Laptops**: Compatibility with various laptop models running supported web browsers.
- **Tablets**: Responsive design to ensure usability on tablets of different screen sizes.
- **Smartphones**: Mobile optimization for seamless access and interaction on smartphones.

3.1.3 Software Interfaces

The website should integrate with various software components and platforms to enhance functionality and user experience, including:

- **Content Management Systems (CMS)**: Interfaces for instructors to manage course content easily.
- **Messaging APIs**: Integration with messaging APIs for real-time communication features.
- **Authentication Services**: Integration with authentication services for secure user login and session management.

3.1.4 Communications Interfaces

The EduTech website should support communication protocols for seamless interaction between users and the platform, including:

- **HTTP/HTTPS**: Hypertext Transfer Protocol for transmitting data between web servers and clients securely.
- **Email**: Integration with email services for sending notifications, password recovery emails, and other communication.
- **APIs**: RESTful APIs for external integrations with third-party services and applications.

3.2 Functional Requirements

This section outlines the specific features and functionalities of the EduTech website.

3.2.1.1 Introduction

Users should be able to enroll in courses offered on the platform.

3.2.1.2 Inputs

- **User Selection**: User selects a course from the list of available courses.
- **Payment Information**: User provides payment details if the course is paid.

3.2.1.3 Processing

- **Validation**: The system validates the user's selection and payment information.
- **Enrollment**: Upon successful validation, the user is enrolled in the selected course.

3.2.1.4 Outputs

- **Enrollment Confirmation**: Confirmation message indicating successful enrollment in the course.

3.2.1.5 Error Handling

- **Invalid Selection**: If the selected course is not available or does not exist, an error message is displayed.
- **Payment Failure**: If payment processing fails, an error message is displayed, and the user is prompted to retry or choose an alternative payment method.

3.2.2 Course Progress Tracking

3.2.2.1 Introduction

The system should track the progress of users enrolled in courses.

3.2.2.2 Inputs

- **User Activity**: User interacts with course materials, completes assignments, and takes quizzes.

3.2.2.3 Processing

- **Progress Calculation**: The system calculates the user's progress based on completed activities and assessments.

3.2.2.4 Outputs

- **Progress Report**: Users can view their course progress through a visual representation, such as a progress bar or percentage completion.

3.2.2.5 Error Handling

- **Data Synchronization Error**: If there is an error in synchronizing user activity data, an error message is displayed, and the user is advised to refresh the page or try again later.

3.5 Non-Functional Requirements

3.5.1 Performance

- **Response Time**: 95% of user interactions, such as page loading and form submissions, shall be processed within 3 seconds.
- **Throughput**: The system shall support a minimum of 1000 concurrent users without significant degradation in performance.

3.5.2 Reliability

- **Mean Time Between Failures (MTBF)**: The system shall have a MTBF value of at least 30 days.
- **Error Handling**: The system shall gracefully handle errors and recover from failures without data loss or corruption.

3.5.3 Availability

- **System Uptime**: The system shall be available for use 99.9% of the time, excluding scheduled maintenance periods.
- **Redundancy**: Critical system components shall have redundancy measures in place to ensure uninterrupted service in case of hardware or software failures.

3.5.4 Security

- **Data Encryption**: All sensitive user data shall be encrypted both in transit and at rest using industry-standard encryption algorithms.
- **Access Control**: User authentication and authorization mechanisms shall be implemented to ensure that only authorized users can access protected resources.
- **Audit Trails**: The system shall maintain audit trails to track user activities and detect any unauthorized access attempts.

3.5.5 Maintainability

- **Code Maintainability**: Code shall be well-documented and follow best practices to facilitate ease of maintenance and future enhancements.
- **Modularity**: The system shall be designed with modular components to enable easy updates and modifications without affecting other parts of the system.

3.5.6 Portability

- **Platform Compatibility**: The system shall be compatible with major web browsers (Chrome, Firefox, Safari, Edge) and operating systems (Windows, macOS, Linux).
- **Deployment Flexibility**: The system shall support deployment on both cloud-based and on-premises infrastructure to accommodate varying deployment requirements of clients.

3.7 Design Constraints

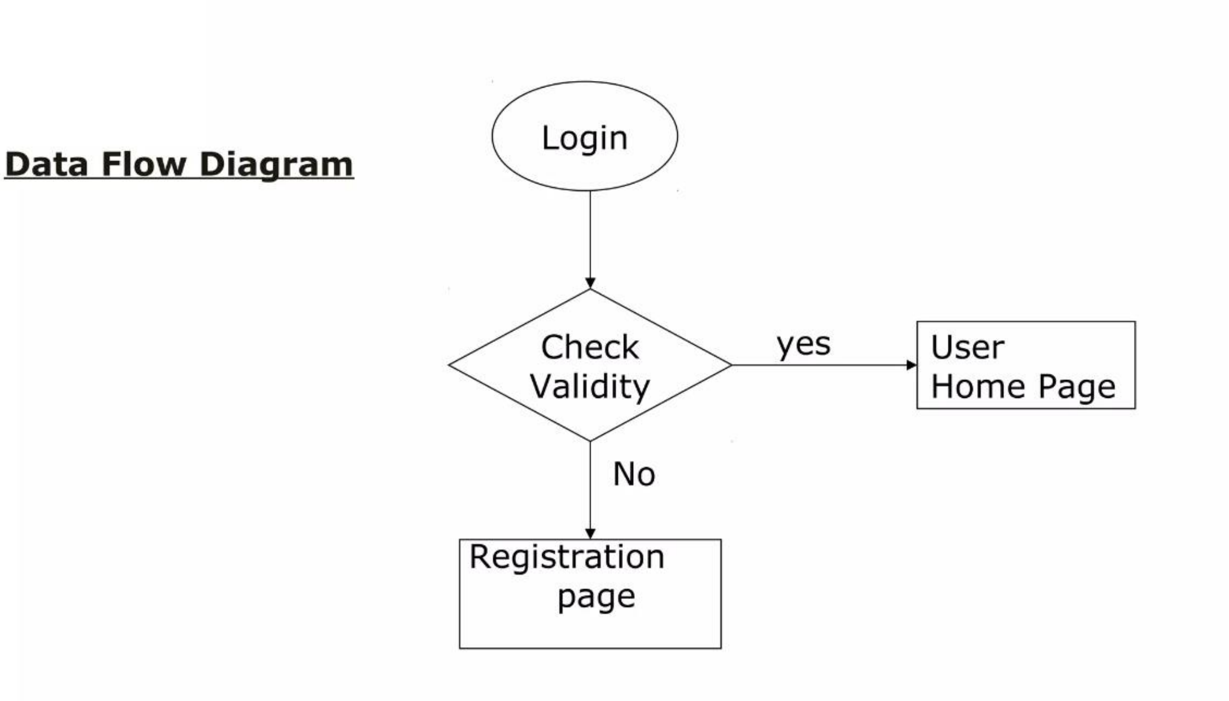
- **Regulatory Compliance**: The software project must adhere to relevant industry standards and regulations, including but not limited to GDPR, COPPA, and FERPA, to ensure the protection of user data and privacy.
- **Company Policies**: The software project must comply with company policies regarding data handling, security measures, and code quality standards.
- **Hardware Limitations**: The software design must take into account any limitations imposed by the hardware infrastructure on which it will be deployed, such as memory constraints, processing power, and network bandwidth.

3.9 Other Requirements

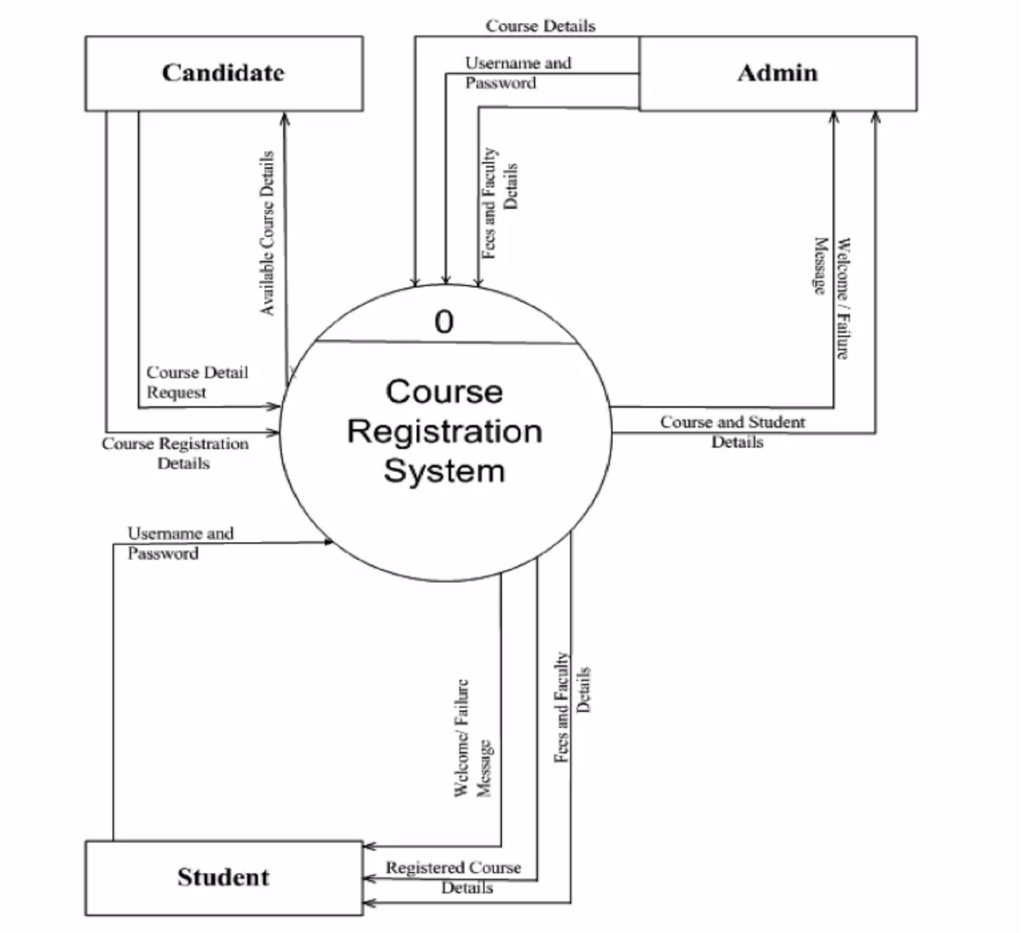
- **Documentation**: The project must include comprehensive documentation covering system architecture, user manuals, API documentation, and developer guides.
- **Training**: A training program must be developed to familiarize users and administrators with the software's features and functionalities.
- **Support**: A support mechanism, such as a help desk or knowledge base, must be established to address user queries and technical issues post-deployment.
- **Scalability**: The software architecture must be designed to accommodate future scalability requirements, allowing for easy expansion to support a growing user base and additional features.
- **Compatibility**: The software must be compatible with existing third-party tools and systems used by the client organization, ensuring seamless integration and interoperability.
- .

4. Analysis Models

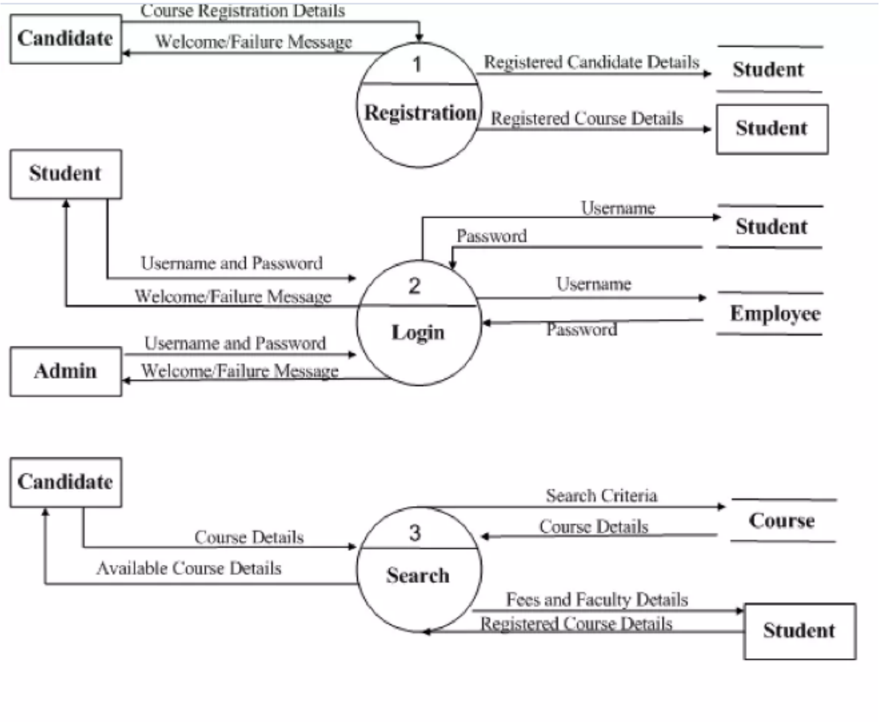
4.1 Data Flow Diagrams (DFD)



Data Flow Diagram



Context-diagram



Introduction: Data Flow Diagrams (DFDs) are graphical representations that illustrate the flow of data within a system. They consist of processes, data stores, data flows, and external entities, showing how data moves through the system and how it is processed and stored.

GitHub Link:

https://github.com/hardikm06/Ed-Tech-Website.git

A. Appendices

A.1 Appendix 1

This appendix contains the initial conceptual documents for the software project, including high-level system architecture diagrams, conceptual models, and preliminary user interface designs. While these documents provide valuable context for understanding the system, they do not constitute formal requirements unless explicitly referenced within the SRS.

A.2 Appendix 2

Appendix 2 includes marketing materials such as product brochures, promotional videos, and market analysis reports. While these materials offer insights into the intended market positioning and target audience of the software product, they are not considered part of the formal requirements documented in the SRS unless explicitly specified otherwise.