Software Requirements Specification (SRS)

CourseHub - Student Course Management System

Prepared by:

Abdur Rafi Abdullah - 203 1353 042

Nurjahan Bint Hamid - 221 2396 042

Md. Fahim Hossain Khan - 223 2359 042

Table of Contents

1. Introduction

1.1 Purpose

Universities around the country that offer different courses often use manual process of student course registration and management, which is very tedious. With CourseHub, our online web based student course management system, we intend to automate this process for all universities.

This Software Requirements Specification(SRS) describes the functional and non-functional requirements of CourseHub. The document will act as a guideline for the development team to design, build, test and maintain the system. Our purpose is to simplify the management of courses, enrollments and grades for both students and administrators within a university.

1.2 Intended Audience

- **Developers**
- **Testers**
- **Managers**
- **Marketing Department**

1.3 Intended Use

- **Developers** can use this document to implement the system according to the requirements.
- **Testers** can validate the system against the specified functionality.
- **Managers** can monitor progress and make sure the product's features are aligning properly with set objectives.
- **Marketing Department** can view the information in the SRS to make informed decision on how to create successful advertisement campaigns for the target users.

1.4 Product Scope

i) <u>Purpose:</u>

CourseHub will provide an online web based student registration system which will be used to provide a dashboard for displaying course information, grades and enable a means for registering for courses online. Students can register, log in, view course details, enroll in available courses and check their grades. Administrators can manage courses such as adding, editing and deleting them, assign grades

and monitor enrollments.

ii) Objectives and Benefits:

The objectives of CourseHub will be to automate course enrollment and grade management, provide role-based access for students and administrators and simplify academic data management using a secure and responsive interface. CourseHub provides a centralized for online academic course management that reduces the manual workload for both administrators and students as well as improving transparency and accuracy when it comes to registering for courses as well as submitting, editing and viewing grades for those courses.

1.5 Rise Definitions

Due to CourseHub being a fully online web based platform there are potential risks to consider when managing and developing this system. The following key risks have been identified along with how to mitigate them:

i) Data Loss:

Grades or database records may be lost during system crashes or accidental deletion. To mitigate this, regular backups of the project and database must be kept on GitHub or local drives.

ii) Incomplete Features:

Some planned features might not be finished before the deadline due to knowledge, skill and/or technology gaps as well as time constraints. To mitigate this, tasks should be divided among team members and the focus should be on completing core features first.

iii) Login or Authentication Errors:

Errors in the login system may prevent users such as students or admins from accessing their accounts. To mitigate this, the login and registration systems should be tested carefully and use simple input validation.

iv) User Interface (UI) Issues:

The UI interface built may not be fully responsive or user-friendly. To mitigate this, the system should be repeatedly tested on different devices and feedback for improvement should be collected from students and admins who use the system.

v) Database Connectivity Problems:

The system may fail to connect to the MySQL database due to server issues. To mitigate this, we must check database connection settings and ensure our MySQL server is running properly.

2. Overall Description

2.1 User Classes and Characteristics

Given below are the user classes and characteristics as well as some assumptions for each.

User Class | Characteristics

i) Student The students from a university that has implemented CourseHub into their system will use it to register for courses, view course details, check their grades and view their academic information. Students that want to register for a course need not call or go to their university and administrator. He or she will register online.

ii) Admin The administrators of the university that has implemented CourseHub into their system will use it for managing courses, submitting grades, viewing list of enrolled students in detail and adding or removing courses. They are responsible for managing the overall system.

2.2 User Needs

- **Students** need a responsive, simple and easy to use dashboard to view and enroll into courses as well as check their grades and other academic information.
- **Admins** need responsive and secure tools as well as special admin privileges to manage courses efficiently, submit grades and monitor student performance.
- **University** needs reliable and secure system to manage their courses and grades.

2.3 Operating Environment

- **Operating system:** Any OS(Windows, macOS) that supports internet browsing.
- **Frontend:** HTML, CSS & JavaScript that runs in any modern web browser (Firefox, Chrome).
- **Backend:** Python (3.10 or higher), FastAPI framework
- **Database:** MySQL(8.0 or higher)

2.4 Constraints

- The system must be developed using FastAPI, MySQL, HTML, CSS and Javascript.
- It should be able to handle concurrent user sessions without performance degradation.
- Only authorized administrators with special privileges can modify courses or grades.
- The system must comply with university data privacy standards.

2.5 Assumptions

i) Computer and Internet Access:

Both Students and administrators are assumed to have access to a computer and have stable internet access.

ii) English language skills:

Both students and administrators are assumed to have sufficient English language skills.

iii) Computer and Internet literacy:

Both students and administrators are assumed to have sufficient internet usage skills and computer literacy

iv) Grade knowledge and responsibility:

Administrators are assumed to have access to grades given by the faculties of each course. They are responsible for grade accuracy.

v) Course knowledge and responsibility:

Administrators are assumed to have sufficient knowledge on course detail in order to modify courses accurately and responsibly.

vi) Unique user ID:

Each user (both students and admins) have unique login ID credentials to distinguish each user.

vii) Database limits:

The database can handle a moderate number of users concurrently (maximum of 1000 concurrent users).

3. Requirements

3.1 Functional Requirements

i) User(Student and Administrator) Registration & Login:

As a User

I want to register and log in securely

So that, I can access my dashboard.

Confirmation

- User can register for an account.
- If he/she provide the required registration details(Username, phone number and address) accurately, set password and authenticate successfully an account will be created.
- If he/she have successfully created an account, the user can now login securely using his/her username and password.
- If he/she have successfully logged in he/she can view his/her dashboard according to his/her role(Student dashboard for students and Admin dashboard for administrators)

ii) Administrator Course Management:

As an Administrator

I want to add, edit and/or delete courses

So that, I can manage courses.

Confirmation

- Administrator can successfully login to the administrator dashboard using username and password
- If he/she successfully access his/her dashboard he/she can perform CRUD operations on courses through a web form.

iii) Administrator Grade Submission:

As an Administrator

I want to assign or update grades for each students enrolled courses

So that, I can responsibly maintain accurate grade records.

Confirmation

- Administrator has to successfully login and access the admin dashboard.
- He/she can then update the grade in the enrollment record.

iv) Student Course Enrollment

As a Student

I want to enroll or leave enrollment from courses

So that, I can choose courses to take for the semester.

Confirmation

- Student has to has to successfully login and access the student dashboard.
- He/she can then see available courses and update his/her enrollment.

v) Student Course Details

As a Student

I want to view all my enrolled courses in detail

So that, I can track my academic progress.

Confirmation

- Student has to has to successfully login and access the student dashboard.
- Student dashboard displays the student's enrolled courses retrieved from the database in detail.

vi) Student Grade Details

As a Student

I want to view my grades from each course

So that, I can evaluate my performance.

Confirmation

- Student has to has to successfully login and access the student dashboard.

- Student can then view his/her grades along with course details.

3.2 Non-Functional Requirements

i) Performance Requirements:

- Response to any kind of interaction must take no longer than 2 seconds to appear on the screen.
- The system should respond quickly for basic operations.

ii) Reliability Requirements:

- System should target an uptime of 99%.
- Ensure that the system is stable and not prone to errors.

iii) Security Requirements:

- System must be password protected to ensure protection of user data and prevent unauthorized access.

iv) Usability Requirements:

- Easy to understand user interface(UI) for ease of access to information and enrollment

v) Compatibility Requirements:

- System should be compatible with most modern OS and browsers.

vi) Prevent duplication

- Prevent duplicate enrollments of courses. Each course should be unique.