PES
UNIVERSITY

Department of Computer Science & Engineering

Software Requirements Specification

Mini Project - Software Engineering

UE22CS341A

Project Title: University Assignment Portal

Team Details:

Member 1:

Name: Arjun N R

SRN: PES2UG22CS910

Member 2:

Name: Mohammed Hassan

SRN: PES2UG22CS317

Date of Submission: 09-09-2024

GitHub Link: Assignment Portal Git

Corse Instructor:

Prof. Rohith Vaidya K

Table of Contents

1 Introduction

1.1 Purpose

The purpose of the University Assignment Portal web application is to streamline and enhance the management of assignments within a university setting. The document covers the entire scope of the product, detailing the functionalities and features required to facilitate assignment management, submission, grading, and feedback.

1.2 Intended Audience

Students: The portal will serve as the main platform for students to access assignment details, submit their work, track progress, and receive feedback. The user-friendly interface and streamlined submission process will enhance their assignment management experience.

Professors: The portal will enable professors to efficiently create and distribute assignments, monitor submissions, grade student work, and provide timely feedback. The automation of these processes will save time and improve the overall grading workflow.

Administrators: The portal will offer administrators oversight into the assignment management process, allowing them to track assignment statuses, generate reports, and ensure the system's smooth operation.

1.3 Product Scope

The University Assignment Portal is designed to serve the needs of students, professors, and administrators within a university setting. The portal will help students by providing a centralized platform to access assignment details, submit their work seamlessly, track their progress, and receive timely feedback. The user-friendly interface and streamlined submission process aim to enhance their overall assignment management experience.

For professors, the portal will offer a more efficient way to create and distribute assignments, monitor student submissions, grade student work, and provide constructive feedback. The automation of these processes will lead to considerable time savings and streamline the grading workflow.

Additionally, administrators will benefit from the portal's oversight capabilities, enabling them to track assignment statuses, generate insightful reports, and ensure the system's seamless operation.

1.4 Reference

SE 2024 - Project-Guidelines with deliverables details.pdf. Provided by PES University, Department of Computer Science and Engineering.
https://drive.google.com/file/d/1M9eYaBUHZfxZQ_Ee8JweKaCGopfC35mf/view

2 Overall Description

2.1 Product Perspective

The University Assignment Portal is a new, self-contained product designed to address the challenges associated with traditional assignment management methods within a university setting. It aims to replace the fragmented use of various platforms and communication channels, such as email, learning management systems, and physical notice boards, by providing a centralized platform for all assignment-related activities.

The portal will integrate seamlessly with the university's existing infrastructure, potentially interfacing with the student information system and learning management system to ensure data consistency and avoid redundancy.

2.2 Product Functions

The University Assignment Portal will offer a range of functionalities to support the assignment lifecycle:

1. Assignment Management:
    i. Creation and uploading of assignments by professors.
    ii. Specification of deadlines, instructions, and associated materials
    iii. Organization and categorization of assignments by course
2. Submission and Tracking:

   i. Submission of assignments by students

   ii. Real-time tracking of submission status

   iii. Confirmation of successful submissions

3. Grading and Feedback:

   i. Review and grading of submissions by professors.

   ii. Provision of detailed feedback to students

   iii. Secure storage of grades and feedback

4. User Management and Access Control:

   i. Creation and management of user accounts

   ii. Role-based access control for students, professors, and administrators

5. Notification System:
   Notifications for new assignments, upcoming deadlines, and grade releases are given.

6. Report Generation:
   Generation of reports on assignment statuses, grades, and feedback.

2.3 User Classes and Characteristics

The University Assignment Portal will cater to three primary user classes:

1. Students: The largest user group, comprising undergraduate and postgraduate students across various disciplines. They will primarily use the portal to access assignment details, submit their work, and receive feedback. They are expected to have basic computer literacy and internet skills.

2. Professors: Faculty members responsible for creating and managing assignments, grading submissions, and providing feedback. They will require a good understanding of the portal's functionalities to effectively utilize its features.

3. Administrators: University staff responsible for overseeing the system's operation, managing user accounts, and generating reports. They will need a comprehensive understanding of the system's features and configurations.

The most important user classes to satisfy are the students and professors, as they are the primary users who will directly interact with the system on a regular basis.

2.4 Operating Environment

The University Assignment Portal will be a web-based application, accessible through standard web browsers (Chrome, Firefox, Safari, Edge) on various devices. The system will initially be hosted on a local server to facilitate development and testing. The specific hardware and operating system configurations of the server will depend on the chosen hosting solution, both for the initial local setup and the eventual cloud deployment.

The portal will be designed to be compatible with modern web technologies and frameworks, ensuring smooth operation across different browsers and devices.

2.5 Design and Implementation Constraints

The development of the University Assignment Portal will be subject to certain constraints:

1. Technology Stack: The project will utilize the following technologies:
    I. Frontend: HTML, CSS, JavaScript, Bootstrap
    II. Backend: Python (Flask)
    III. Database: MySQL
2. Security: The system must adhere to strict security standards to protect sensitive student and assignment data. This includes implementing robust authentication, authorization, and data encryption mechanisms.
3. Usability: The portal should be user-friendly and intuitive, catering to users with varying levels of technical expertise. The interface should be accessible and adhere to web accessibility guidelines.
4. Scalability: The system should be designed to oversee a large number of users and assignments, ensuring optimal performance even during peak usage periods.
5. Integration: The portal may need to integrate with existing university systems, such as the student information system and learning management system. The development team will need to ensure seamless data exchange and compatibility with these systems. (tentative)
6. Time and Resources: The project will be constrained by the available development time and resources. The team will need to prioritize features and

functionalities based on their importance and feasibility within the given constraints.

7. Maintenance: The system should be designed for easy maintenance and updates, allowing for future enhancements and bug fixes.

2.6 Assumptions and Dependencies

The following assumptions and dependencies have been identified:

1. User Adoption: The effectiveness of the portal hinges on its acceptance and utilization by students and professors. It is assumed that both groups will be willing to transition from traditional assignment management methods to the new platform.
2. Technical Infrastructure: The initial reliance on a local server assumes that it possesses the necessary capacity to handle the expected user load and data volume. The subsequent migration to a cloud-based infrastructure assumes the availability of suitable cloud services and the technical expertise to manage the transition effectively.
3. Integration with Existing Systems: Third-Party Components: The project may rely on third-party libraries or components for certain functionalities, such as authentication, file storage, or notification services. The assumption is that these components will be compatible with the chosen technology stack and will continue to be supported and maintained throughout the portal's lifecycle.

3 External Interface Requirements

3.1 User Interface

The University Assignment Portal will provide an intuitive and accessible web-based user interface (UI) designed to accommodate the distinct needs of students, professors, and administrators. The UI will follow modern usability standards, ensuring responsiveness across various devices and platforms, enhancing the user experience for all roles.

1. Homepage: Upon successful authentication, users will be directed to a personalized homepage based on their roles (Student, Professor, Administrator). The homepage will present an overview of relevant tasks such as upcoming assignments, pending submissions, and system notifications.

2. Student Dashboard: The student dashboard will display all active assignments categorized by course, with submission status, deadlines, and links for uploading work.
3. Professor Dashboard: Professors will have access to an interactive dashboard that allows them to create, edit, and manage assignments. They will be able to view submission details, grade assignments, and provide detailed feedback.
4. Administrator Dashboard: Administrators will have a system overview with options to generate reports, manage user roles, and oversee the overall assignment process.
5. Assignment Submission Interface: Students will submit assignments through an intuitive form that accepts various file formats.
6. Grading Interface: Professors and students will use a secure grading interface where professors can review submissions, enter feedback, and store grades, while students receive and review the same.

3.2 Software Interfaces

The University Assignment Portal will interface with various software systems to ensure smooth operation, data consistency, and secure access control. The following software interfaces will be employed.

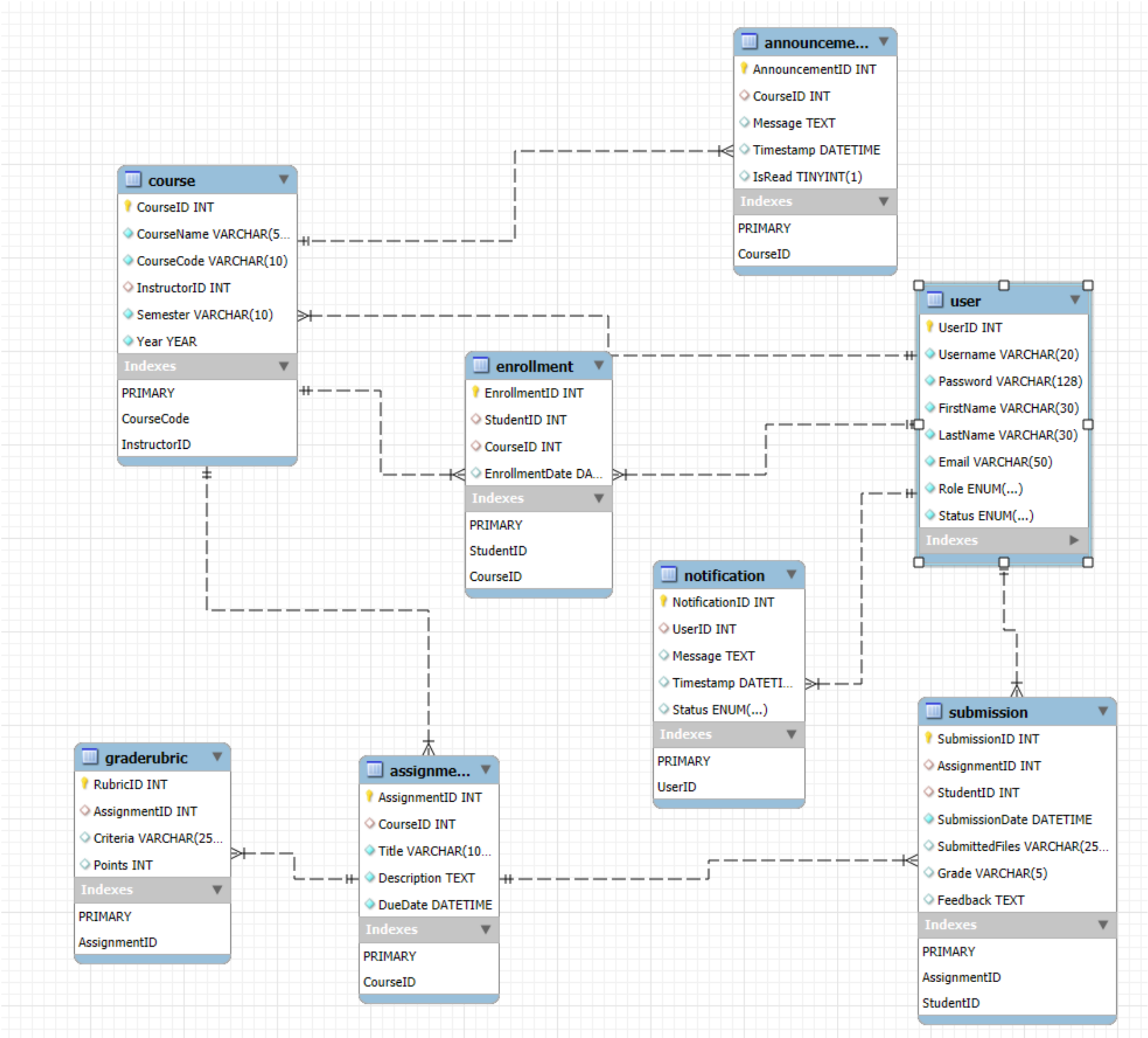
1. Student Information System (SIS): The portal will integrate with the university's existing SIS to retrieve student details, course enrolments, and other relevant information. This will ensure accurate assignment distribution and proper role management.
2. Authentication System: The portal will leverage the university's centralized authentication system for user login. A Single Sign-On (SSO) mechanism, based on OAuth or OpenID Connect, will be employed to securely authenticate students, professors, and administrators.
3. Database Management System (DBMS): A relational database system (e.g., MySQL) will be used to store all assignment-related data, including user information, assignment details, submissions, and feedback. The portal will implement secure CRUD operations through a RESTful API for data manipulation and retrieval.
4. File Storage System: A secure file storage system will be used to manage the upload and retrieval of assignment submissions. The system may rely on local server storage or cloud storage solutions depending on scalability requirements.

3.3 Communications Interfaces

The University Assignment Portal will employ the following communication protocols and interfaces to ensure secure and efficient data exchange between the system, external services, and users:

1. HTTP/HTTPS Protocol: The primary communication protocol between client browsers and the server will be HTTPS to ensure encrypted data transfer and secure communications. All sensitive data, such as user credentials, will be transmitted securely to protect against eavesdropping and man-in-the-middle attacks.
2. RESTful APIs: The portal will expose a set of RESTful APIs for communication between the frontend and backend services. These APIs will manage all operations related to assignments, submissions, and user management, ensuring data consistency across the system. The API will support common HTTP methods (GET, POST, PUT, DELETE) and will follow industry standards for secure communication.
3. WebSocket: For real-time updates and notifications (e.g., new assignments or grade releases), the portal will utilize WebSocket-based communication. This will provide an enhanced user experience, allowing instant updates without requiring page reloads.
4. SMTP Protocol: The portal will use the SMTP protocol for sending automated email notifications related to assignment submissions, deadlines, and grade releases. The system will be integrated with the university’s email infrastructure or third-party email service providers to ensure reliable and timely delivery of notifications.
5. OAuth/OpenID Connect: The portal will integrate with the university’s existing authentication mechanisms using OAuth or OpenID Connect protocols. This will allow secure, token-based authentication for users, reducing the risk of credential theft and enhancing the overall security of the system.
6. SFTP/FTP: For file transfers related to assignment submissions or batch uploads, the system may employ SFTP (Secure File Transfer Protocol) or FTP depending on the security requirements. This will ensure safe and efficient file exchange between the client systems and the server.

4 Analysis Models (Relational Schema Diagram)



5 System Features

1. Assignment Creation and Management
   This feature enables professors to create and manage assignments efficiently. Professors can upload assignment details, set deadlines, provide instructions, and attach relevant resources.

   Inputs: Assignment title, description, due date, and supplementary materials.
   Outputs: Assignments are made available to students.

2. Assignment Submission
   Students can submit their assignments through a user-friendly interface.

   Inputs: Assignment files and submission metadata.
   Outputs: Submission confirmations and status updates are communicated to students.

3. Grading and Feedback
   Professors can review student submissions, provide detailed feedback, and assign grades using predefined grading rubrics.
   Inputs: Submission content, feedback text, and grading criteria.
   Outputs: Grades and feedback are securely stored and accessible to students upon completion.

4. User Roles and Access Control
   The system includes role-based access control to manage user permissions and ensure appropriate access to various functionalities. This feature supports students, professors, and administrators.

   Inputs: User credentials and assigned roles.
   Outputs: Access permissions are enforced according to user roles, allowing appropriate feature access.

5. Report Generation
   Reports can be generated on assignment status, submission rates, and grading information.

   Inputs: Report criteria, such as assignment type, course, and date range.
   Outputs: Generated reports provide insights into assignment and grading metrics, supporting administrative functions.

6 Other Nonfunctional Requirements

6.1 Performance Requirements

The system must maintain responsiveness, with page load times in normal conditions. It should manage concurrent users without performance degradation. Data processing tasks, such as grading and report generation, should be managed properly. The system must be scalable to accommodate increased user numbers.

6.2 Safety Requirements

The system must ensure secure data storage through periodic backups, protecting against data loss due to hardware failure or accidental deletion. Access must be restricted to authenticated users.

6.3 Security Requirements

Login processes must include encrypted password handling, with data encryption applied to assignment submissions and grades during transmission and storage. Role-based access control must restrict user permissions based on their roles.

6.4 Software Quality Attributes

The system must prioritize maintainability, ensuring ease of updates, bug fixes, and feature expansions. Usability should be a focus, providing a user-friendly and intuitive interface for all users. It must be dependable, portable and compatible with various platforms.

6.5 Business Rules

Professors are restricted to modifying or grading assignments for other courses. Students are limited to submitting assignments for courses they are enrolled in. Administrators can create, modify, or delete users, courses per university policies. The professors will strictly enforce assignment deadlines.

Appendix A: Glossary

1. CRUD Operations: Create, Read, Update, Delete - fundamental database operations.

2. RESTful API: An architectural style for designing networked applications, often used for communication between frontend and backend.

3. OAuth/OpenID Connect: Protocols for authorization and authentication, enabling secure access to protected resources.

4. WebSocket: A communication protocol providing full-duplex communication channels over a single TCP connection, often used for real-time updates.

5. SMTP: Simple Mail Transfer Protocol, used for sending email.

6. SFTP/FTP: Secure File Transfer Protocol/File Transfer Protocol, used for transferring files between systems.

7. WCAG 2.1: Web Content Accessibility Guidelines, ensuring web content is accessible to people with disabilities.

8. GDPR: General Data Protection Regulation, a legal framework that sets guidelines for the collection and processing of personal information from individuals within the European Union.

9. SSO: Single Sign-On, an authentication scheme allowing a user to log in with a single ID and password to any of several related, yet independent, software systems.

10. Man-in-the-middle attack: A type of cyberattack where a malicious actor intercepts and potentially alters communication between two parties without their knowledge.

11. Data Encryption: The process of converting information or data into a code, especially to prevent unauthorized access.

12. Role-Based Access Control (RBAC): A method of regulating access to computer or network resources based on the roles of individual users within an enterprise.

13. Horizontal Scaling: The ability to handle increased workload by adding more machines to the existing pool of resources.

14. API (Application Programming Interface): A set of rules and specifications that software programs can follow to communicate with each other.

15. Concurrency: The ability of a system to handle multiple requests or tasks simultaneously.

16. Data Integrity: The accuracy, completeness, and consistency of data throughout its lifecycle.

17. Usability: The ease with which users can learn and use a product to achieve their goals.

18. Maintainability: The ease with which a software system or component can be modified to correct faults, improve performance, or other attributes.

19. Portability: The ease with which a software system or component can be transferred from one hardware or software environment to another.

20. Peak Usage Periods: The times when the system is expected to experience the highest number of concurrent users and requests.

Appendix B: Field Layouts

1. User (Could be Student/Professor/Administrator)

Field	*Length*	*Data Type*	*Description*	*Is Mandatory*
User ID	10	Numeric	Unique identifier for each user	Y
Username	20	String	Username for login	Y
Password	128	String	Encrypted password for authentication	Y
First Name	30	String	User's first name	Y
Last Name	30	String	User's last name	Y

Email	50	String	User's email address	Y
Role	10	String	User's role (Student, Professor, Administrator)	Y
Status	10	String	User's account status (Active, Inactive)	Y

2. Course

Field	*Length*	*Data Type*	*Description*	*Is Mandatory*
Course ID	10	Numeric	Unique identifier for each course	Y
Course Name	50	String	Name of the course	Y
Course Code	10	String	Course code (e.g., CS101)	Y
Instructor ID	10	Numeric	ID of the professor teaching the course	Y
Semester	10	String	Semester in which the course is offered	Y
Year	4	Numeric	Year in which the course is offered	Y

3. Assignment

Field	Length	Data Type	Description	Is Mandatory
Assignment ID	10	Numeric	Unique identifier for each assignment	Y
Course ID	10	Numeric	ID of the course the assignment belongs to	Y
Title	100	String	Title of the assignment	Y
Description	500	String	Detailed description of the assignment	Y
Due Date	9	Date	Date and time when the assignment is due	Y

Attachments	-	File	Files associated with the assignment (instructions, resources)	N

4. Submission

Field	Length	Data Type	Description	Is Mandatory
Submission ID	10	Numeric	Unique identifier for each submission	Y
Assignment ID	10	Numeric	ID of the assignment the submission belongs to	Y
Student ID	10	Numeric	ID of the student who made the submission	Y
Submission Date		Datetime	Date and time of submission	Y
Submitted Files	-	File	Files submitted by the student	Y
Grade	5	Numeric or String	Grade assigned by the professor (can be numeric or letter grade)	N
Feedback	500	String	Feedback provided by the professor	N

Note: The field lengths and data types mentioned are just tentative and can be adjusted based on the specific requirements further development. Additionally, we may need to add more entities and fields to refine the design.

Appendix C: Requirements Traceability Matrix

Note:

"Code file reference" column in the Requirements Traceability Matrix (RTM) cannot be filled out at this stage because the code has not yet been developed. The RTM is a living document that evolves alongside the project. Once the code is developed, we can update the RTM.

Sl No	*Req Id*	*Brief description of requirement*	*Architecture reference*	*Design Reference*	*Code file reference*	*Test case Id*	*Test Case Validation*
1	Req-001	Relational Schema Diagram	DBMS MySQL	Er Diagram	PNG Diagram	Tst 1	Verify ER diagram accuracy, entity relationships, attribute definitions
2	Req-002	Database Formation (table, relations etc)	DBMS MySQL	Er Diagram	SQL Code	Tst 2	Execute SQL code, verify table creation, data integrity, relationships
3	Req-003	Frontend – Login page	Web Frontend	Login Wireframe			
4	Req-004	Frontend – Student Page interface	Web Frontend	Student Wireframe			
5	Req-005	Frontend – Professors Page interface	Web Frontend	Professor Wireframe			
6	Req-006	Frontend – Admin terminal interface	Web Frontend	Admin Wireframe			
7	Req-007	Backend	Web Backend	Backend			