IT314

SOFTWARE ENGINEERING

Group 16

Online Examination System

Software Requirements Specification

Prof. Saurabh Tiwari

Group Members:

No	Student Id	Name
1	202201204	Meet Mahaliya
2	202201208	Devang Vaghani
3	202201210	Mihir Prajapati
4	202201214	Harshali Dharmik
5	202201225	Gunjan Saroliya
6	202201227	Nishank Kansara
7	202201229	Yash Chaudhari
8	202201241	Divyesh Ramani
9	202201242	Dev Davda
10	202201251	Mehul Vagh

Index

Introduction

SRS (Software Requirements Specification) is a document that serves as a blueprint for software development. It defines the purpose, functionalities, and constraints of the system being developed. For project of website for an **online examination system** the SRS document provides a comprehensive understanding of the requirements from all stakeholders, ensuring alignment and clarity throughout the software development lifecycle.

Problem Description

In the current educational ecosystem, conducting examinations involves various logistical challenges such as scheduling, managing question papers, evaluating answer sheets, and providing results in a timely manner. Traditional examination methods are time-consuming, resource-intensive, and prone to errors. Additionally, students face difficulties in accessing a comprehensive platform for practice and examination.

To address these issues, eExam provides a streamlined, user-friendly online examination system. This system digitizes the entire examination process, enhancing efficiency and transparency. It caters to three types of users:

1. **Admin:** Manages examiners and students, oversees the platform, and monitors upcoming exams.
2. **Examiner:** Creates and manages exams, compiles question banks, and reviews detailed exam reports.
3. **Student:** Practices questions, takes exams, views results, and accesses their profile.

By automating and centralizing these functionalities, eExam aims to reduce administrative overhead, improve accuracy, and provide a seamless experience for all stakeholders.

Scope of the Project

The eExam project is designed to simplify and digitize the examination process while ensuring flexibility, reliability, and scalability. The scope includes the following:

1. **Admin Features**
    - Create, update, and delete examiner and student accounts.
    - View and upcoming exams.
    - Manage the overall system functionality.

2. **Examiner Features**
    - Create and schedule exams with customizable settings.
    - Add, update, and delete questions in the question bank.
    - Generate and view reports of completed exams, including:
        - Total students who attempted the exam.
        - Individual student responses.

3. **Student Features**
    - Practice questions from the question bank.
    - Participate in scheduled exams.
    - View exam results with performance breakdowns.
    - Access and profile information.

4. **Common Features**
    - User authentication and role-based access control.

- Responsive and intuitive user interface for all users.

5. **Technical Scope**
    - The system will be a web-based application, accessible via any modern browser.
    - Secure storage and transmission of sensitive data, including exam results and user details.
    - Scalability to handle a large number of users and exams concurrently.

Technologies Used:

Frontend: React JS, Custom CSS
Backend: Node JS, Express JS
Database: MongoDB
Unit Testing: Mocha
GUI Testing: Selenium IDE
Load Testing: Apache Load Tester

Overall Description of the eExam Project

Users and Stakeholders

1. **Users:**
    - **Admin:** Manages user accounts, view exams, and monitors the system.
    - **Examiner:** Creates exams/questions, manages Exam/question bank and views exam reports.
    - **Student:** Practices questions, takes exams, and views results.

2. **Stakeholders:**
    - Educational institutions, IT administrators, examiners, students and the development team.

Constraints

- **System:** Must handle multiple users simultaneously and ensure secure access.
- **Technology:** Web-based, compatible with standard browsers, and scalable database.
- **Regulations:** Must comply with data protection and institutional policies.

Assumptions and Dependencies

- Users will have basic technical skills and stable internet access.
- Input data (questions, user details) will be accurate.
- Third-party services (e.g., email notifications, analytics) function as expected.

Requirement Analysis

Functional requirements:

1. **Admin Login**
    - The system shall enable administrators to log in with valid credentials.
    - Error messages shall notify administrators of invalid login attempts.

2. **Create Examiner/Student Accounts**
    - Administrators shall be able to create examiner/student accounts by entering details.
    - Examiners/Students shall receive emails upon account creation containing UserID and Password.

3. **Manage Examiner/Student Accounts**
    - Administrators shall be able to add, update and delete examiner/student accounts.
    - Changes to student/examiner accounts shall reflect immediately on the student dashboard.

4. **Manage Question Bank**
    - Administrators shall be able to add, update, and delete questions in the question bank.
    - Questions shall be categorized by subject and type for easier retrieval.

5. **Manage Exams (Admin)**
    - Administrators shall be able to view, edit, and delete scheduled exams.
    - Updates to upcoming exams shall reflect immediately.

6. **View System Performance Reports**
    - Administrators shall be able to view reports detailing system usage, including the number of students, exams, and average scores.

7. **Reset Password**
    - Administrator shall be able reset the password using old password verification.

8. **Send Notifications to Users**
    - Administrators shall be able to compose and send notifications to selected users.
    - Notifications shall appear on the user dashboard or be sent via email.

9. **Examiner Login**
    - Examiners shall be able to log in with valid credentials.
    - Error messages shall notify examiners of invalid login attempts.
    - In case, the Examiner forgets the password then he/she shall be able to reset the password using email verification.

10. **Reset Password**
    - Examiner shall be able reset the password using old password verification.

9. **View all Upcoming Exams**

    - Examiners shall be able see schedules of all upcoming exams to remove conflicts.

11. **Create Exams**
    - Examiners shall be able to create exams by selecting questions from the question bank or adding new ones.
    - Examiners shall set time limits and scoring criteria for exams.

12. **Add New Questions to Question Bank**
    - Examiners shall be able to input question text, options, and correct answers to the question bank.
    - Questions shall be categorized by subjects and difficulty.

13. **Schedule Exams**
    - Examiners shall be able to select dates, times, and durations for exams.

14. **Manage Exams (Examiner)**
    - Examiners shall be able to view, edit, and delete their upcoming exams.
    - All updates to exams shall be saved and reflected in the system.

15. **Grade Descriptive Questions**
    - Examiners shall be able to manually grade descriptive questions and assign scores.
    - Feedback shall be displayed to students after grading.

16. **View Student Performance**
    - Examiners shall be able to view individual student scores and performance reports.
    - Examiners shall be able to search the student with StudentID.

17. **Monitor Live Exams**
    - Examiners shall access a real-time dashboard showing students currently taking exams.

18. **View Profile**
    - Examiners shall be able to view their profile.

19. **Student Login**
    - Students shall log in using valid credentials.
    - Error messages shall notify students of invalid login attempts.
    - In case, the student forgets the password then he/she shall be able to reset the password using email verification.

20. **View Upcoming Exams**
    - Students shall view a list of upcoming exams with details such as date and subject.

21. **Take an Exam**
    - Students shall start an exam at the scheduled time.
    - The system shall automatically save answers during the exam.

22. **View Exam Results**
    - Students shall view their scores, responses, correct answers, and feedback after an exam is submitted.
    - Results shall be displayed in a detailed report.

23. **Practice Questions from Question Bank**
    - Students shall practice questions categorized by subject and difficulty from the question bank.

24. **Receive Notifications for Exams and Results**
    - Students shall receive notifications in their dashboard or via email for upcoming exams and results.

25. **Bookmark Question**
    - Students shall be able to bookmark the question.

26. **View Past Exam Results**
    - Students shall view their history of taken exams along with scores.
    - Past exams shall be available for review but not for retaking.
    - Students shall be able to see their average score of all exams.

27. **Reset Password**
    - Students shall be able reset the password using old password verification.

Non-Functional requirements:

- **Reliability**: Maintain 99.9% uptime by implementing robust error-handling mechanisms and providing intuitive notifications for users when issues occur.
- **Scalability**: Design the system to accommodate growth in both user traffic and content seamlessly.
- **Usability**: Offer a responsive and interactive user interface for a seamless experience across all devices.
- **Security:** Ensure data safety with encryption protocols and secure storage solutions.
- **Compatibility**: The system must function seamlessly across various devices, browsers, and operating systems.
- **Performance**: The system should respond to user actions within 2 seconds under normal load conditions.

Use Case:

1. Admin - Login
2. Admin - Create Student/Examiner Accounts
3. Admin - Manage Student/Examiner Accounts
4. Admin - Manage the Question Bank
5. Admin - Reset Password
6. Admin - View Exams
7. Examiner - Login
8. Examiner - Create Exams
9. Examiner - Add New Questions to the Question Bank
10. Examiner - Reset Password
11. Admin - View System Performance Reports
12. Admin - Send Notifications
13. Student - Login

14. Student - View Upcoming Exams
15. Student - Give an Exam
16. Student - View Exam Results
17. Student - Reset Password
18. Examiner - Schedule Exams
19. Examiner - Manage Exams
20. Examiner - Grade Descriptive Questions
21. Examiner - Monitor Live Exams
22. Student - Practice Questions from the Question Bank
23. Student - Bookmark the Questions
24. Examiner - View Performance Reports
25. Student - Receive Notifications
26. Student - View Previous Exam Results

Use case description:

1. Use Case: Admin Login

Description:
Allows administrators to securely log in to the system to manage various functionalities.

Actors: Admin

Preconditions:

- The admin has an account.
- The admin has access to the platform.

Main Flow:

1. The admin opens the admin login page.
2. The system displays the login form.
3. The admin enters their email and password.
4. The system validates the credentials.
   - Check if the email exists.
   - Verifies if the password is correct.
5. If the credentials are valid, the admin is redirected to the admin dashboard.

Alternative Flow:

4.1. If any required fields are missing, the system prompts the admin to complete the form.

4.2. If the credentials are invalid, an error message is displayed, prompting the admin to retry.

Postconditions:

- Admin is logged in and redirected to the admin dashboard.

2. Use Case: Create Examiner/Student Accounts

Description:
Enables administrators to create accounts for examiners to manage exams.

Actors: Admin

Preconditions:

- The admin is logged in.

Main Flow:

1. The admin navigates to the "Create Examiner/Student" section.
2. The system displays a form to input examiner/student details (e.g., name, email).
3. The admin enters the required details.
4. The system validates the provided details:
    - Checks if the email format is valid.
    - Check if the email is already registered.
5. If valid, the system creates the examiner/student account and sends a notification or email to the examiner/student.
6. The system displays a success message.

Alternative Flow:

4.1. If required fields are missing, the system prompts the admin to complete the form.

4.2. If the email is already registered, the system shows an error message.

Postconditions:

- A new examiner/student account is created.
- Notification or email is sent to the examiner/student.

3. Use Case: Manage Student/Examiner Accounts

Description:
Allows administrators to manage student/Examiner accounts (add, edit, or delete).

Actors: Admin

Preconditions:

- The admin is logged in.

Main Flow:

1. The admin navigates to the "Manage Student/Examiner Accounts" section.
2. The system displays a list of existing student/Examiner accounts.
3. The admin selects an action:
    - **Add:** Enters new student details and submits.
    - **Edit:** Updates details of an existing account.
    - **Delete:** Confirms deletion of a selected account.
4. The system validates the input and performs the requested action.
5. A success message is displayed, and the updated list of student/Examiner accounts is shown.

Alternative Flow:

3.1. If any required fields are missing or invalid during Add/Edit, the system prompts corrections.

4.1. If the action fails (e.g., database error), the system shows an error message.

Postconditions:

- The student/examiner accounts list is updated accordingly.
- Changes are reflected immediately on the student/examiner dashboard.

4. Use Case: Manage Question Bank

Description:
Allows administrators to add, update, or delete questions in the question bank.

Actors: Admin

Preconditions:

- The admin is logged in.

Main Flow:

1. The admin navigates to the "Manage Question Bank" section.
2. The system displays the list of questions with options to add, edit, or delete.
3. The admin selects an action:
   - **Add:** Enters new question details (e.g., text, options, correct answer).
   - **Edit:** Modifies details of an existing question.
   - **Delete:** Confirms the removal of a question.
4. The system validates the input and processes the action.
5. A success message is displayed, and the updated question list is shown.

Alternative Flow:

3.1. If required fields are missing during Add/Edit, the system prompts the admin to complete the details.

4.1. If the action fails, the system displays an error message.

Postconditions:

- The question bank is updated accordingly.

5. Use Case: Reset Password

Description:
Enables admin to reset their password using Old password.

Actors: Admin, Examiner, Student

Preconditions:

- The admin has access to the platform.

Main Flow:

1. The admin clicks on the "Reset Password" in profile.
2. The system displays a password reset form.
3. The admin enters an Old password and New password.
4. The system verifies the Old password.
5. If valid, password reset successfully,and the admin can log in with the new password.

Alternative Flow:

4.1. If the old password is wrong, the system displays an error message of an incorrect old password.

Postconditions:

- The admin's password is updated successfully.
- A log of the password reset is maintained for security.

6. Use Case: Admin - View Exams

Description:
Allows administrators to view all scheduled and upcoming exams in the system for better monitoring and management.

Actors: Admin

Preconditions:

- The admin is logged in.

Main Flow:

1. The admin navigates to the "View Exams" section.
2. The system displays a list of all scheduled exams with details such as:
   - Exam name
   - Scheduled date and time
   - Duration

3. The admin can filter exams by:
   - Subject
   - Difficulty
   - Status (e.g., upcoming, ongoing, completed).
4. The admin can click on an exam to view additional details, such as the list of registered students and scoring criteria.

Alternative Flow:

3.1. If no exams match the filter criteria, the system displays a "No Results Found" message.

Postconditions:

- The admin has a clear overview of the exams.
- The filtered data is displayed as per the selected criteria.

7. Use Case: Examiner - Log In

Description:
Enables examiners to securely log in to their accounts to manage exams and questions.

Actors: Examiner

Preconditions:

- The examiner has an account.
- The examiner has access to the platform.

Main Flow:

1. The examiner clicks on the "Examiner Login" button/link.
2. The system displays the login form.
3. The examiner enters their email and password.
4. The system validates the credentials:
   - Checks if the email exists.
   - Verifies if the password matches.

5. If the credentials are valid, the examiner is redirected to the examiner dashboard.

Alternative Flow:

4.1. If any required fields are missing, the system prompts the examiner to complete the form.

4.2. If the credentials are invalid, an error message is displayed, prompting the examiner to retry.

Postconditions:

- The examiner is logged in and redirected to the dashboard.
- Login attempts are logged for security purposes.

8. Use Case: Examiner - Create Exams

Description:
Allows examiners to create new exams and define their structure, timing, and content.

Actors: Examiner

Preconditions:

- The examiner is logged in.

Main Flow:

1. The examiner navigates to the "Create Exams" section.
2. The system displays a form to input exam details:
    - Exam name
    - Subject
    - Date and time
    - Duration
    - Scoring criteria
    - Instructions for students.

3. The examiner selects questions from the question bank or creates new ones.
4. The examiner submits the form to create the exam.
5. The system validates the input and saves the exam.
6. A success message is displayed, and the examiner is redirected to the list of scheduled exams.

Alternative Flow:

3.1. If any required fields are missing or invalid, the system prompts the examiner to correct the details.

4.1. If there are issues with the database or network, the system displays an error message and suggests retrying.

Postconditions:

- A new exam is created and added to the list of scheduled exams.
- Notifications are sent to registered students (if applicable).

9. Use Case: Examiner - Add New Questions to the Question Bank

Description:

Allows examiners to add new questions to the question bank to ensure a diverse and updated set of questions for exams.

Actors: Examiner

Preconditions:

- The examiner is logged in.

Main Flow:

1. The examiner navigates to the "Add Questions" section of the question bank.

2. The system displays a form to input question details:
    - Question text
    - Answer options (for multiple-choice questions)
    - Correct answer(s)
    - Tags for subject or topic categorization.
3. The examiner fills out the form and submits it.
4. The system validates the input:
    - Ensures no required fields are missing.
    - Checks for duplicate questions (optional).
5. The system saves the question and updates the question bank.
6. A success message is displayed.

Alternative Flow:

4.1. If validation fails (e.g., missing required fields, invalid format), the system prompts the examiner to make corrections.

Postconditions:

- The new question is added to the question bank.

10. Use Case: Examiner - Reset Password

Description:
Allows examiners to reset their password using an old password.

Actors: Examiner

Preconditions:

- The examiner has access to the platform.

Main Flow:

1. The Examiner clicks on the "Reset Password" in profile.
2. The system displays a password reset form.
3. The Examiner enters an Old password and New password.

4. The system verifies the Old password.
5. If valid, password reset successfully,and the Examiner can log in with the new password.

Alternative Flow:

4.1. If the old password is wrong, the system displays an error message of an incorrect old password.

Postconditions:

- The examiner’s password is updated successfully.
- A log of the password reset is maintained for security.

11. Use Case: Admin - View System Performance Reports

Description:
Allows administrators to view detailed reports on system performance, including usage statistics and exam outcomes.

Actors: Admin

Preconditions:

- The admin is logged in.

Main Flow:

1. The admin navigates to the "Performance Reports" section.
2. The system displays a dashboard showing the following:
   - Number of active students and examiners.
   - Number of exams conducted over a specified period.
   - Average student scores and pass percentages.
   - Trends in exam participation over time.
3. The admin can apply filters to view specific data, such as by subject, date range, or performance category.

Alternative Flow:

3.1. If no data matches the filters, the system displays a "No Data Available" message.

Postconditions:

- The admin gains insights into system usage and performance metrics.
- Reports are available for future reference or sharing.

12. Use Case: Admin - Send Notifications (User Story 7)

Description:
Allows administrators to send announcements or notifications to examiners and students.

Actors: Admin

Preconditions:

- The admin is logged in.

Main Flow:

1. The admin navigates to the "Send Notifications" section.
2. The system displays a form where the admin can:
   - Write the notification content.
   - Select recipients (e.g., all users, specific groups like examiners or students).
3. The admin submits the form.
4. The system sends the notification:
   - Displays it on user dashboards.
   - Sends it via email if configured.
5. A success message is shown to the admin.

Alternative Flow:

4.1. If notification delivery fails for some users, the system logs the error and informs the admin.

Postconditions:

- The recipients receive the notification.

13. Use Case: Student - Log In

Description:
Enables students to securely log in to their accounts to access exams and view results.

Actors: Student

Preconditions:

- The student has an account.
- The student has access to the platform.

Main Flow:

1. The student clicks on the "Student Login" button/link.
2. The system displays the login form.
3. The student enters their email and password.
4. The system validates the credentials:
    - Checks if the email exists.
    - Verifies the password.
5. If the credentials are valid, the student is redirected to their dashboard.

Alternative Flow:

4.1. If the credentials are invalid, the system displays an error message and requests retry.

Postconditions:

- The student is logged in and redirected to the dashboard.

14. Use Case: Student - View Upcoming Exams

Description:
Allows students to view a list of their upcoming exams to help them prepare.

Actors: Student

Preconditions:

- The student is logged in.

Main Flow:

1. The student navigates to the "Upcoming Exams" section.
2. The system displays a list of upcoming exams with details such as:
    - Exam name
    - Scheduled date and time
    - Subject

Alternative Flow:
2.1. If no upcoming exams are scheduled, the system displays a "No Upcoming Exams" message.

Postconditions:

- The student is informed about their upcoming exams.

15. Use Case: Student - Take an Exam

Description:
Allows students to take scheduled exams and submit their answers.

Actors: Student

Preconditions:

- The student is logged in.
- The exam is active at the current time.

Main Flow:

1. The student navigates to the "Take Exam" section and selects the exam.
2. The system displays the exam instructions and timer.
3. The student starts the exam by clicking "Start Exam."
4. The system displays the questions one at a time (or in a scrollable format):
    - The student answers the questions.
    - Responses are auto-saved periodically.
5. The student submits the exam by clicking "Submit."
6. The system confirms submission and saves the responses.

Alternative Flow:
4.1. If the student navigates away from the exam, progress is auto-saved.
4.2. If the timer ends, the system auto-submits the exam.

Postconditions:

- The student's responses are recorded and saved.
- The exam status is updated to "Completed."

16. Use Case: Student - View Exam Results

Description:
Allows students to view their performance on completed exams.

Actors: Student

Preconditions:

- The student is logged in.
- The exam has been graded.

Main Flow:

1. The student navigates to the "Exam Results" section.
2. The system displays a list of completed exams with their results.
3. The student clicks on a specific exam to view detailed results, including:
    - Score
    - Correct and incorrect answers
    - Feedback (if applicable)

Alternative Flow:

2.1. If no exams have been graded, the system displays a "No Results Available" message.

Postconditions:

- The student is informed of their performance.

17. Use Case: Student - Reset Password

Description:
Allows students to reset their password using an old password.

Actors: Student

Preconditions:

- The student has access to the platform.

Main Flow:

1. The student clicks on the "Reset Password" in profile.
2. The system displays a password reset form.
3. The student enters an Old password and New password.
4. The system verifies the Old password.
5. If valid, password reset successfully,and the student can log in with the new password.

Alternative Flow:

4.1. If the old password is wrong, the system displays an error message of an incorrect old password.

Postconditions:

- The student’s password is updated successfully.

18. Use Case: Examiner - Schedule Exams

Description:
Allows examiners to schedule exams for a specific date and time.

Actors: Examiner

Preconditions:

- The examiner is logged in.
- A valid exam is created or selected for scheduling.

Main Flow:

1. The examiner navigates to the "Schedule Exam" section.
2. The system displays a list of created exams.
3. The examiner selects an exam to schedule.
4. The system displays a scheduling form where the examiner enters:
    - Date and time of the exam.
    - Duration of the exam.

    - Exam visibility and access settings.
5. The examiner submits the form.
6. The system saves the schedule and sends notifications to the students about the scheduled exam.

Alternative Flow:

5.1. If the selected time conflicts with another exam, the system displays a warning and suggests alternative times.

Postconditions:

- The exam is scheduled, and students are notified.

19. Use Case: Examiner - Manage exams

Description:

Allows the examiner to modify/delete already created exams.

Actor: Examiner

Preconditions:

- Examiner is logged in the system.
- Exam which is to be modified/deleted is already created.

Main Flow:

1. Examiner clicks on the "Upcoming Exam" button from the dashboard.
2. System displays the exams created by the examiner.
3. Examiner clicks on the edit button present below the exam he/she wants to modify.
4. System by displays a dialog box with entered exam details.
5. Examiner modifies with the necessary changes.
6. Examiner clicks on the "Save Exam Details" button to save changes he/she has made.

Alternative Flow:

5.1 If examiner wants to edit questions:

5.1.1 Examiner clicks on the “Edit Questions” button.

5.1.2 System displays Update & Delete buttons in a small dialog box.

5.1.3 Examiner clicks on the update button to modify the selected question or clicks delete to delete the selected question.

5.1.4 Examiner clicks on the save button to save changes in that question.

5.2 If Examiner wants to add a question:

5.2.1 Examiner clicks on the “Add Question” button.

5.2.2 System displays a small dialog box requiring the examiner to fill necessary question details (Question Text, Options, Difficulty, Points, Add Question from Question BAnk, etc).

5.2.3 Examiner clicks on save button to save the entered details.

5.3 If Examiner wants to edit instructions:

5.3.1 Examiner clicks on the “Edit Instruction” button.

5.3.2 System displays a small dialog box displaying already entered instructions.

5.3.3 Examiner clicks on the “Add New Instruction” button to add a new instruction by typing into the displayed text field or clicks on “Delete” to delete the already entered selected instruction.

5.3.4 Examiner clicks on save button to save modified instructions.

5.4 If Examiner wants to Cancel/Delete Exam:

5.4.1 Examiner clicks on the “Cancel Exam” button.

5.4.2 System displays pop up asking examiner to confirm (Yes, Cancel) or cancel (No, Keep).

5.4.3 Examiner clicks on “Yes, Cancel”.

Post Conditions:

- Examiner can view modifications made by clicking on edit button.
- Student can view modifications made by clicking on the “Upcoming Exams” button and then selecting desired exam.

20. Use Case: Examiner - Grade Descriptive Questions

Description:
Allows examiners to manually grade descriptive questions and provide feedback to students.

Actors: Examiner

Preconditions:

- The examiner is logged in.
- The exam containing descriptive questions has been completed by students.

Main Flow:

1. The examiner navigates to the "Grade Responses" section.
2. The system displays a list of completed exams.
3. The examiner selects an exam and views student responses for descriptive questions.
4. The examiner assigns scores and writes optional feedback for each response.
5. The examiner submits the grades.
6. The system updates the scores and notifies the students.

Alternative Flow:
4.1. If the examiner skips grading for some responses, the system saves progress for later completion.

Postconditions:

- Students receive their scores and feedback for descriptive questions.

21. Use Case: Examiner-Monitor Live Exams

Description:
Allows examiners to monitor ongoing exams in real-time.

Actors: Examiner

Preconditions:

- The examiner is logged in.
- An exam is currently active.

Main Flow:

1. The examiner navigates to the "Monitor Live Exams" section.
2. The system displays a real-time dashboard with the following details:
    - List of students currently taking the exam.
    - Progress (e.g., percentage completed) for each student.
    - Submission status (submitted/not submitted).
3. The examiner can:
    - View individual student activity logs (e.g., question navigation, idle time).
    - Send reminders or warnings to students.
4. The examiner monitors the exam until it ends.

Alternative Flow:

2.1. If no exams are currently active, the system displays a "No Live Exams" message.

Postconditions:

- The examiner ensures the exam progresses smoothly and can identify potential issues.

22. Use Case: Practice questions

Description:

Allows students to practice questions from the question bank.

Actor: Student

Preconditions:

- Student is logged in the system.
- More than one questions are added by the examiner in the question bank.

Main Flow:

1. Student click on the “Question Bank” button from the dashboard.
2. System displays the empty question bank page with course names listed.
3. Student selects the course he/she wants to practice the questions from.
4. System by default displays all the types (Easy, Medium, and Difficult) of questions.
5. Student reads through the questions and clicks on the “View” button present below the question text.

6. System displays a dialog box containing question and multiple options to select the answer from.
7. Student selects an option by clicking on it.
8. Student clicks on the submit button.
9. System displays whether the selected option is correct answer or not.

Alternative Flow:

3.1 If student wants to practice from already bookmarked questions:

3.1.1 Student clicks on the “Bookmark” button.

3.1.2 Student reads through the questions and clicks on the “View” button present below the question text. (Applying filter is optional)

3.1.3 System displays a dialog box containing question and multiple options to select the answer from.

3.1.4 Student select an option by clicking on it.

3.1.5 Student clicks on the submit button.

3.1.6 System displays whether the selected option is the correct answer or not.

4.1 If student wants to filter out Easy, Medium, or Hard questions:

4.1.1 Student clicks on the “Filter by Difficulty” button.

4.1.2 System displays dropdown box.

4.1.3 Student selects difficulty level (Easy, Medium, or Hard).

Post Conditions: Student can practice questions before attempting any exam for better performance.

23. Use Case: Bookmark Questions

Description:

Allows students to bookmark preferred questions from the question bank.

Actor: Student

Preconditions:

- Student is logged in the system.
- More than one questions are added by the examiner in the question bank.

Main Flow:

1. Student clicks on the “Question Bank” button from the dashboard.
2. System displays the empty question bank page with course names listed.
3. Student selects the course he/she wants to bookmark the question from.
4. System by default displays all the types (Easy, Medium, and Difficult) of questions.
5. Student reads through the questions and clicks the star icon present in the question card to bookmark that question.

Alternative Flow:

5.1 If student wants to filter out Easy, Medium, or Hard questions.

5.1.1 Student selects required difficulty from “Filter by Difficulty” tab.

Post Conditions:

- All the changes are saved and student can view bookmarked questions by clicking the Bookmark option.
- Student can view bookmarked questions by filtering according to difficulty.

24. Use Case: Examiner - View performance report

Description:

Allows the examiner to view students' performance.

Actor: Examiner

Preconditions:

- Examiner is logged in the system.
- Exam has already finished.

Main Flow:

1. Examiner clicks on the “Past Exam” button from the dashboard.
2. System displays exam results along with necessary exam details.
3. Examiner clicks on the “Search by Student ID” tab to view desired student’s exam results.
4. Examiner types Student ID in the text box.
5. System displays entered student’s exam results.

Alternative Flow:

1.1 If exam was recently conducted:

> 1.1.1 Examiner checks in the “Past 5 Exams” section in the homepage.
>
> 1.1.2 If found, the Examiner clicks on that exam and follows the same steps done in the main flow.

1.2 If examiner wants to view performance report for exam conducted by other professor:

> 1.2.1 Examiner checks in the “Past 5 Exams” section in the homepage.
>
> 1.2.2 If found the desired exam, Examiner clicks on that exam and follows the same steps done in the main flow.

Post Conditions:

- The examiner successfully views the performance report of the selected student or exam.

25. Use Case: View previous exam results

Description:

Allows the student to view his/her previous exam performance.

Actor: Student

Preconditions:

- Students are logged in the system.
- Student has attempted the desired exam for which he/she wants to view the results.

Main Flow:

1. Students click on the “Results” button from the dashboard.
2. System displays exam results along with an overall performance graph and a pie chart.
3. Students click on the exam name of the desired exam.
4. System displays that exam’s result (Answers selected, Correct, and Incorrect answers, etc) along with necessary exam details.

Alternative Flow:

1.1 If the exam was recently conducted:

1.1.1 Student checks in the “Past 5 Exams” section in the homepage.

1.1.2 If found, the student clicks on that exam.

1.1.3 System displays exam results along with an overall performance graph and a pie chart and follows the same steps done in the main flow.

Post Conditions:

- The student successfully views the results of the selected exam, including answers, correct and incorrect responses, and exam details.
- Any charts or performance data (overall graph, pie chart) are displayed and remain accessible during the session until the student navigates away.

26. Use Case: Receive Notifications for Exam Result Published

Description:
Allows students to receive notifications when their exam results are published, ensuring they stay informed about their performance.

Actors: Student

Preconditions:

- The student has an active account on the platform.
- The exam has been graded and results are ready to be published.
- Notifications are enabled for the student's account.

Main Flow:

1. The examiner publishes the exam results in the system.
2. The system generates a notification for all students who took the exam.
3. The system sends the notification to the student's dashboard and/or via email (if email notifications are enabled).
4. The student logs into the platform.
5. The system displays the notification in the notification center or dashboard.
6. The student clicks on the notification to view the detailed exam result.

Alternative Flow:
4.1. If the student has not logged in, the notification remains unread in the dashboard until they access it.

4.2. If email notifications fail (e.g., invalid email address), the system logs the error and retries after a defined interval.

Postconditions:

- The student is notified of the exam result publication.
- Notification delivery events are logged for auditing purposes.

User Stories:

Sr. No.	Title	User story	priority	Acceptance criteria
1	Admin Login	As an Admin, I want to log in so that I can manage the system.	Must have	- Admin can log in with valid credentials. - Error is shown for incorrect credentials.
2	Create Examiner Accounts	As an Admin, I want to create examiner accounts so that examiners can create exams.	Must have	- Admin can input details (name, email) to create an examiner account. - Examiner receives a notification or email about the account creation.
3	Manage Student Accounts	As an Admin, I want to manage student accounts so that students can access the exams.	Must have	- Admin can add, edit, or delete student accounts. - Changes are reflected in the student dashboard immediately.
4	Manage Question Bank	As an Admin, I want to manage the question bank so that the system has a variety of questions for exams.	Must have	- Admin can update, or delete questions. - Questions can be tagged by subject and type.
5	Manage	As an Admin, I want	Must	- Admin can view all

Sr. No.	Title	User story	priority	Acceptance criteria
	Exams	to manage exams so that I can edit or delete exams as needed.	have	scheduled exams. - Admin can edit or remove upcoming exams.
6	View System Performance Reports	As an Admin, I want to view system performance reports so that I can monitor usage and results.	Must have	- Reports include number of students, exams, and average scores.
7	Send Notifications to Users	As an Admin, I want to send notifications to examiners and students so that I can inform them of updates.	Must have	- Admin can write and send announcements to selected users. - Users receive notifications in their dashboard or via email.
8	Examiner Login	As an Examiner, I want to log in so that I can create exams.	Must have	- Examiner can log in with valid credentials. - Error is shown for incorrect credentials.
9	Create Exams	As an Examiner, I want to create exams so that students can take tests.	Must have	- Examiner can select questions from the question bank or create new ones. - Examiner can set time limits and scoring criteria for the exam.
10	Add New Questions to Question Bank	As an Examiner, I want to add new questions to the question bank so that I can tailor exams to my subject area.	Must have	- Examiner can input the question text, options, and correct answers. - Examiner can categorize questions by topic and type.

Sr. No.	Title	User story	priority	Acceptance criteria
11	Schedule Exams	As an Examiner, I want to schedule exams so that students can take them at a specific time.	Must have	- Examiner can select the exam date, time, and duration. - Students receive notifications about the scheduled exam
12	Manage Exams	As an Examiner, I want to manage my exams so that I can edit or delete them if needed.	Must have	- Examiner can view upcoming exams and make changes. - Changes are saved and updated in the system.
13	Grade Descriptive Questions	As an Examiner, I want to manually grade descriptive questions so that I can provide detailed feedback.	Must have	- Examiner can review and assign scores to descriptive answers. - Feedback is displayed to students after grading is complete.
14	View Student Performance	As an Examiner, I want to view student performance so that I can assess the effectiveness of my exams.	Must have	- Examiner can view individual student scores and performance reports.
15	Monitor Live Exams	As an Examiner, I want to monitor live exams so that I can ensure students are completing exams on time.	Must have	- Examiner can view a real-time dashboard showing which students are currently taking the exam.
16	Student Login	As a Student, I want to log in so that I can access exams and view results.	Must have	- Student can log in with valid credentials. - Error is shown for incorrect credentials.
17	View Upcoming	As a Student, I want to view upcoming	Must have	- Student sees a list of upcoming exams with

Sr. No.	Title	User story	priority	Acceptance criteria
	Exams	exams so that I can prepare well.		details such as date and subject. - Student can filter exams by subject or date.
18	Take an Exam	As a Student, I want to take an exam so that I can be graded on my knowledge.	Must have	- Student can start the exam at the scheduled time. - Timer starts when the exam begins, and answers are saved automatically.
19	View Exam Results	As a Student, I want to view my exam results so that I can understand how I performed.	Must have	- Student can see their score, correct answers, and feedback after the exam is graded. - Results are displayed in a detailed report.
20	Practice Questions from Question Bank	As a Student, I want to practice questions from the question bank so that I can prepare for exams.	Must have	- Student can access practice questions by subject or topic.
21	View Past Exam Results	As a Student, I want to view past exam results so that I can track my progress over time.	Must have	- Student can see the history of exams taken with scores. - Past exams are available for review, but cannot be retaken.
22	Receive Notifications for Exams and Results	As a Student, I want to receive notifications for upcoming exams and results so that I stay informed.	Must have	- Student is notified in the dashboard or via email about scheduled exams. - Notification is sent when exam results are available.

Sr. No.	Title	User story	priority	Acceptance criteria
23	Reset Password	As a Student, I want to reset my password so that I can regain access to my account if I forget my credentials.	Must have	- Students can request a password reset link via email. - Students can reset their password using the link.
24	Update Profile Details	As a Student, I want to update my profile details so that my information is accurate and up-to-date.	Must have	- Student can edit fields such as name, email, etc.