Software Requirement Specification

– Online Examination System

Document ID: ASP-CMS-SRS-02.doc

Author(s): RAMAPRABA J

Version: 0.1

Date: April 4, 2024

1. Abstract

Online Examination System is a software solution, which allows any industry or institute to arrange, conduct and manage examinations via an online environment. System helps students to offer a quick and easy way to appear for the test. It also provides the results immediately after the examination. Student can enter to perform exam only with their valid username and password. This examination contains multiple choice questions and appropriate number of options. This provides time limit. The user can see their results after completing the exam. This helps the students to write the exam from far distance and which can provide security and simplicity and other beneficial features to the user. Key features include secure authentication mechanisms, ensuring the integrity and confidentiality of the examination process. Administrators can effortlessly create, schedule, and manage exams, while admin can customize question sets and grading criteria. Students benefit from the convenience of accessing exams remotely, answering questions within specified timeframes, and receiving instant feedback on their performance.

2. Objective and Scope

The objective and scope of an online examination system encompass various aspects aimed at enhancing the efficiency, accessibility, and integrity of the examination process. The scope of the product includes the following basic features:

- The system can be used in educational institutions as well as in corporate world. Can be used anywhere any time as it is a web-based application (user Location doesn't matter). No restriction that examiner has to be present when the candidate takes the test.
- There are different kinds of Users (Admin, Examiner) can log on to website and access, secure authentication system for different types of user access the website. Login screen for entering username and password will be based upon the users.
- Time can be saved by scheduling the exams, if it is available a question bank to store questions for different subjects. A system can be given a mark by checking the students answers, and give the result as soon as students finish his exam.
- The system should have records of students and faculty that can be access to the system which can be used only for the authorized person.
- The system features automated grading functionality, which streamlines the evaluation process and provides instant result.

3. Project End Users

The end users of online examination system are

Examiners: Students or exam candidates are the primary users who take the exams conducted through the online examination system. They interact with the system to access exam instructions, questions, and submit their responses. They may also view their results and performance feedback through the system.

Admin: Admin are responsible for creating and managing exams within the online examination system. They add questions to the question bank, add courses, schedule exams, monitor the exam sessions.

4. Module Description

4.1 Login to the system

Each and every user should be authenticated with a User Name and Password to login into the system. Validations for User Name and Password. user name accepts only alphabets, numbers, dot (.) symbol and underscore (_) symbol and password accept alphabets, numbers and special characters.

4.2 Manage Courses

Add Courses:

Administrators can add new courses to the system by providing details such as course name and description.

Update Courses:

Admin can update existing course details such as the course name and description which is assigned to it.

Delete Courses:

Admin can delete courses that are no longer offered or are obsolete. This action removes the

course and its associated information from the system.

View Courses:

Users can view a list of all available courses in the system, along with their respective details

4.3 Manage Exam

Add Exam:

Admin can add new exams to the system by specifying details such as the exam title, course, description, Time limit and display time to which the exam is assigned.

Update Courses:

Admin can update existing exam details, such as modifying the exam duration, description and associated courses.

Delete Exam:

Admin can delete exams that are no longer relevant or are outdated. This action removes the exam and its associated information from the system.

View Courses:

Users can view a list of all exams available in the system, along with their details.

4.4 Manage Questions

Add Questions:

Admin can add new questions to the system by providing the question content, type (e.g., multiple-choice, true/false), and associated course.

Update Questions:

Admin can update existing questions, such as modifying the question content, type, or associated course.

Delete Questions:

Admin can delete questions that are no longer needed or are incorrect. This action removes the question and its associated information from the system.

View Courses:

Users can view a list of all questions available in the system, along with their details.

4.5 Take Exam

This module is designed to facilitate online exam-taking for students. Upon accessing the system, students can view a list of available exams based on their enrolled courses. They can read exam instructions, navigate through the questions, select or input their answers, and submit their responses within the stipulated time frame. This module ensures a user-friendly and seamless experience for students during the examination process.

4.6 View Result

After completing an exam, students can review their performance and results within this module. The system automatically grades the exam based on predefined criteria and provides students with immediate feedback, including their total score and any correct/incorrect answers.

5. Functional and Non-Functional Requirements

5.1 Functional Requirements

User Authentication: Users should be able to log in securely using unique credentials (e.g., username and password) to access the system.

Exam Creation: Admin should be able to create different types of courses and provide details of courses like define course titles and descriptions.

User Roles and Permissions: Different user roles such as administrators, teachers, and students should have specific permissions and access levels within the system.

Exam Creation: Authorized users should be able to create different types of exams (e.g., multiple-choice, essay, true/false), set time limits, and specify other parameters (e.g., passing score, randomization of questions).

Question Bank Management: The system should allow administrators to create, edit, categorize, and manage a database of exam questions.

Exam Administration: Authorized users should be able to schedule exams, assign them to specific groups or individuals, and monitor exam progress in real-time.

Question Delivery: The system should deliver questions to examinees in the designated format and handle question randomization if required.

Answer Submission: Examinees should be able to submit their answers within the specified time limit. The system should support various question types and answer formats.

Grading: The system should automatically grade objective questions and provide instant feedback to examinees. For subjective questions, it should allow teachers or examiners to manually grade and provide feedback.

Result Generation: After the exam, the system should calculate and display the results, including scores, correct/incorrect answers, and performance analytics.

5.2 Non-Functional Requirements

The system should be fast and responsive, handling multiple users simultaneously without delays. As the number of users and exams increases, the system should be able to scale up its resources seamlessly to accommodate the higher demand without sacrificing performance. The system should be stable and dependable, with minimal downtime or disruptions during exams. It must ensure the confidentiality and integrity of exam data, protecting against unauthorized access or tampering. It should be user-friendly, with intuitive interfaces and clear navigation to enhance the user experience. The system should work seamlessly across different devices, browsers, and operating systems. The system should be available whenever needed, with measures in place to minimize downtime.

6. High Level Design

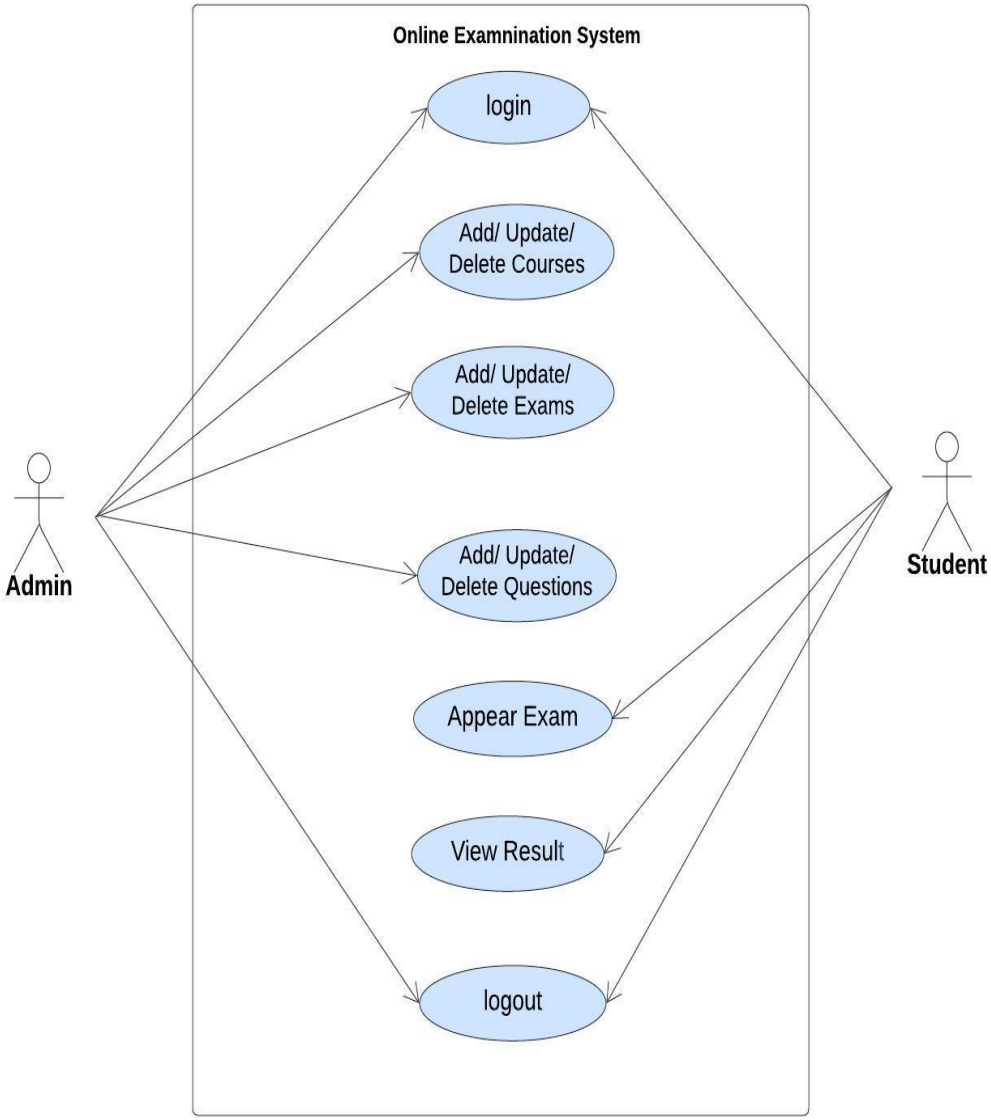
The online examination system will follow a client-server architecture, utilizing a web-based client for user interaction and a centralized server to manage data and handle requests. Multiple user roles, including administrators, teachers, and students, will be supported, each with tailored permissions and access levels. Data will be stored in a relational database system, ensuring integrity and efficient retrieval for user credentials, course information, exam details, questions, and results. Robust security measures such as encryption, authentication, and authorization protocols will be implemented to protect sensitive information and prevent unauthorized access. The system will be designed with scalability in mind, capable of accommodating a large number of concurrent users and future growth through scalable infrastructure and resource management techniques. The user interface will prioritize intuitiveness and user-friendliness, with responsive design principles ensuring accessibility across various devices and screen sizes. Key features will include exam creation and management tools for administrators and teachers, real-time monitoring capabilities, automated grading functionality, and comprehensive reporting and analytics tools to provide insights into student performance and course effectiveness.

7. Lower Level Design

Lower Leve Design (LLD) of online examination system's design will involve defining the database schema with structured tables for user accounts, courses, exams, questions, and results. Secure authentication mechanisms will be implemented, such as username/password authentication or token-based authentication using industry-standard protocols like OAuth. User interface components will be developed using HTML, CSS, and JavaScript frameworks to ensure responsiveness and visual appeal. Server-side logic will be written in programming languages such as Python, Java, or Node.js to handle client requests, interact with the database, and process data. Encryption algorithms like AES or RSA will secure sensitive data stored in the database, while role-based access control rules will restrict access to functionalities and data based on user roles and permissions. Additional features such as exam timers, grading algorithms, and reporting and analytics tools will be integrated, with error handling mechanisms and logging functionality implemented for debugging and monitoring purposes.

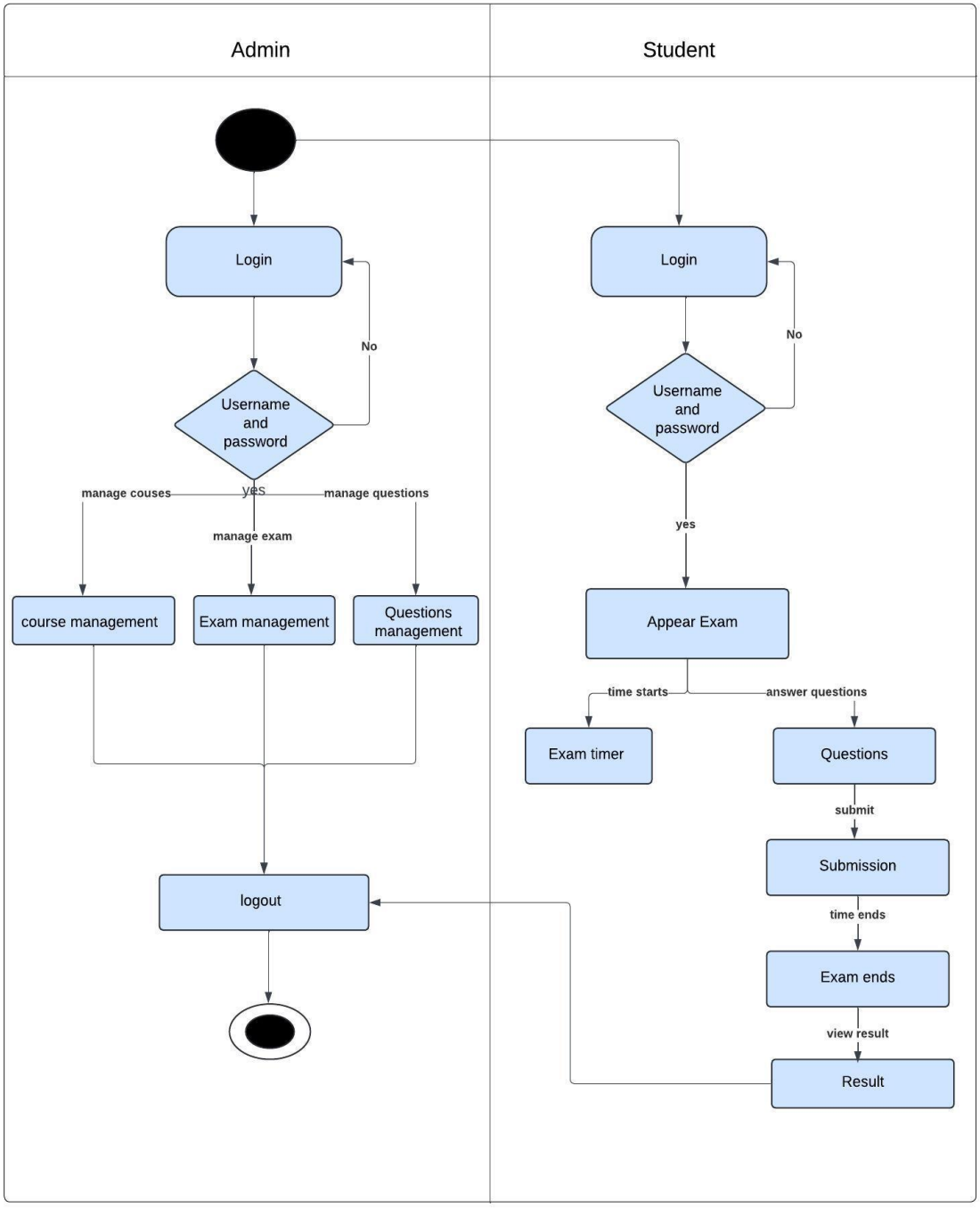
8. Diagrams

8.1 Use Case Diagram:



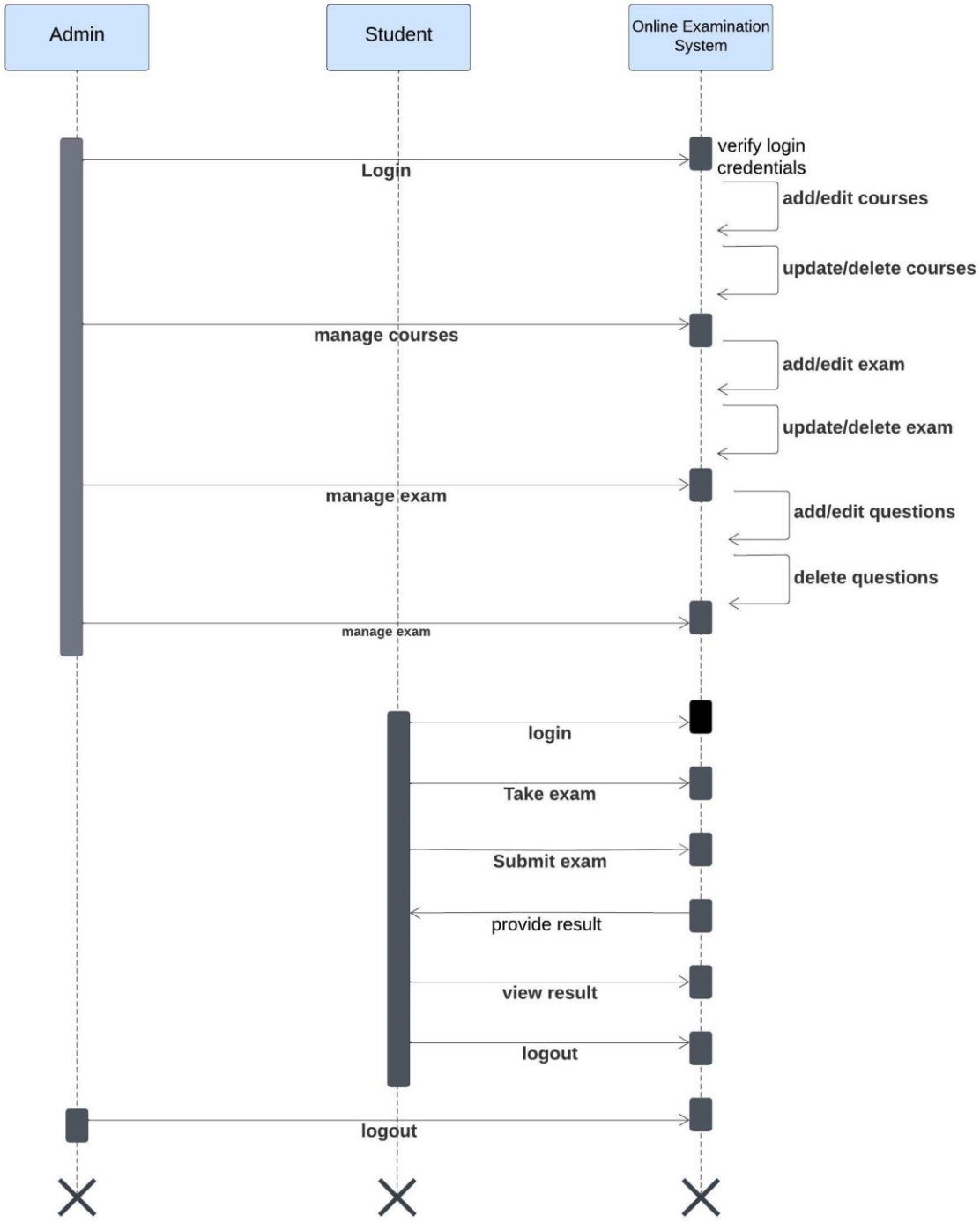
Use case diagram for OES

8.2 Flowchart:



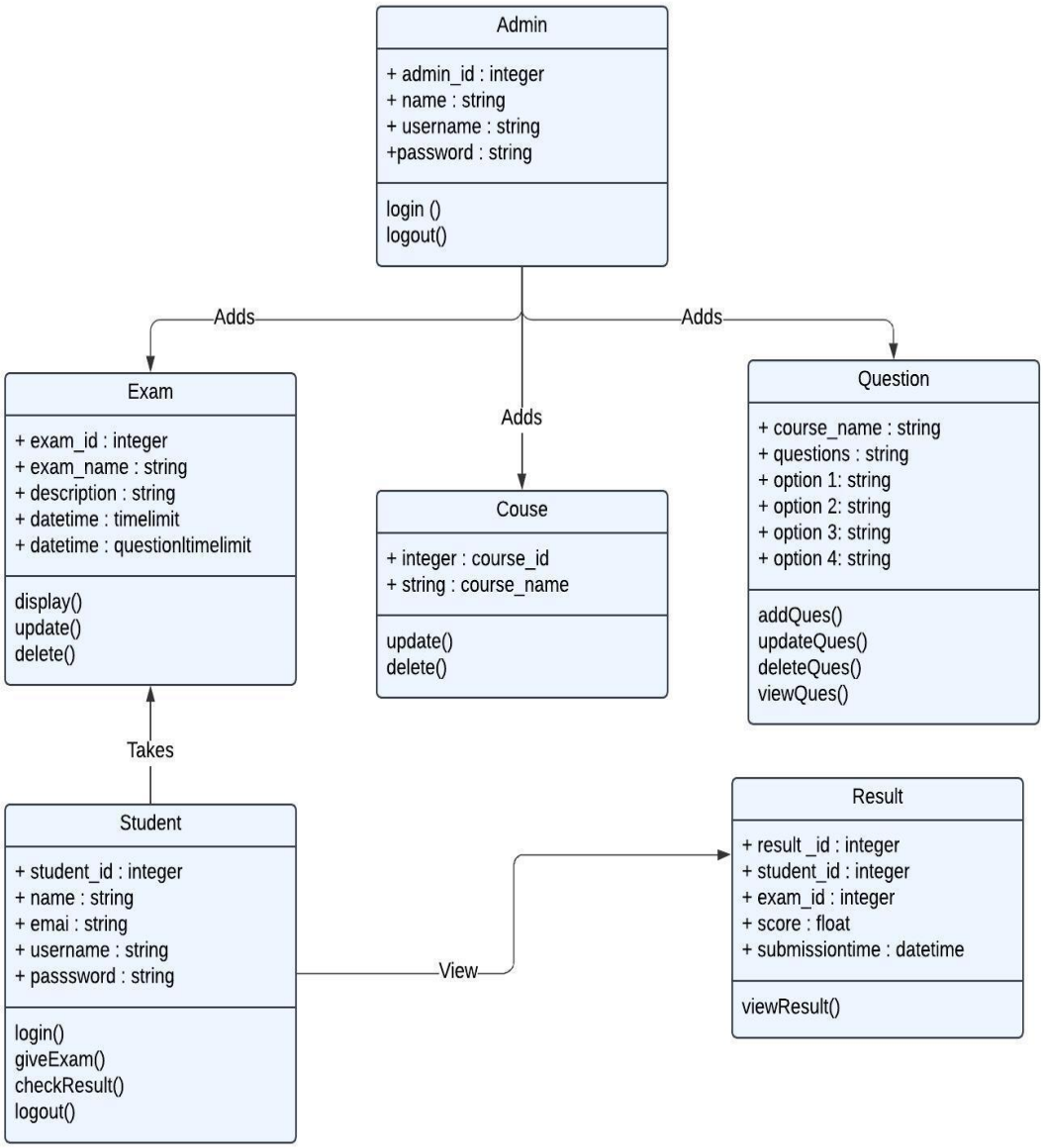
Flowchart for OES

8.3 Sequence Diagram:



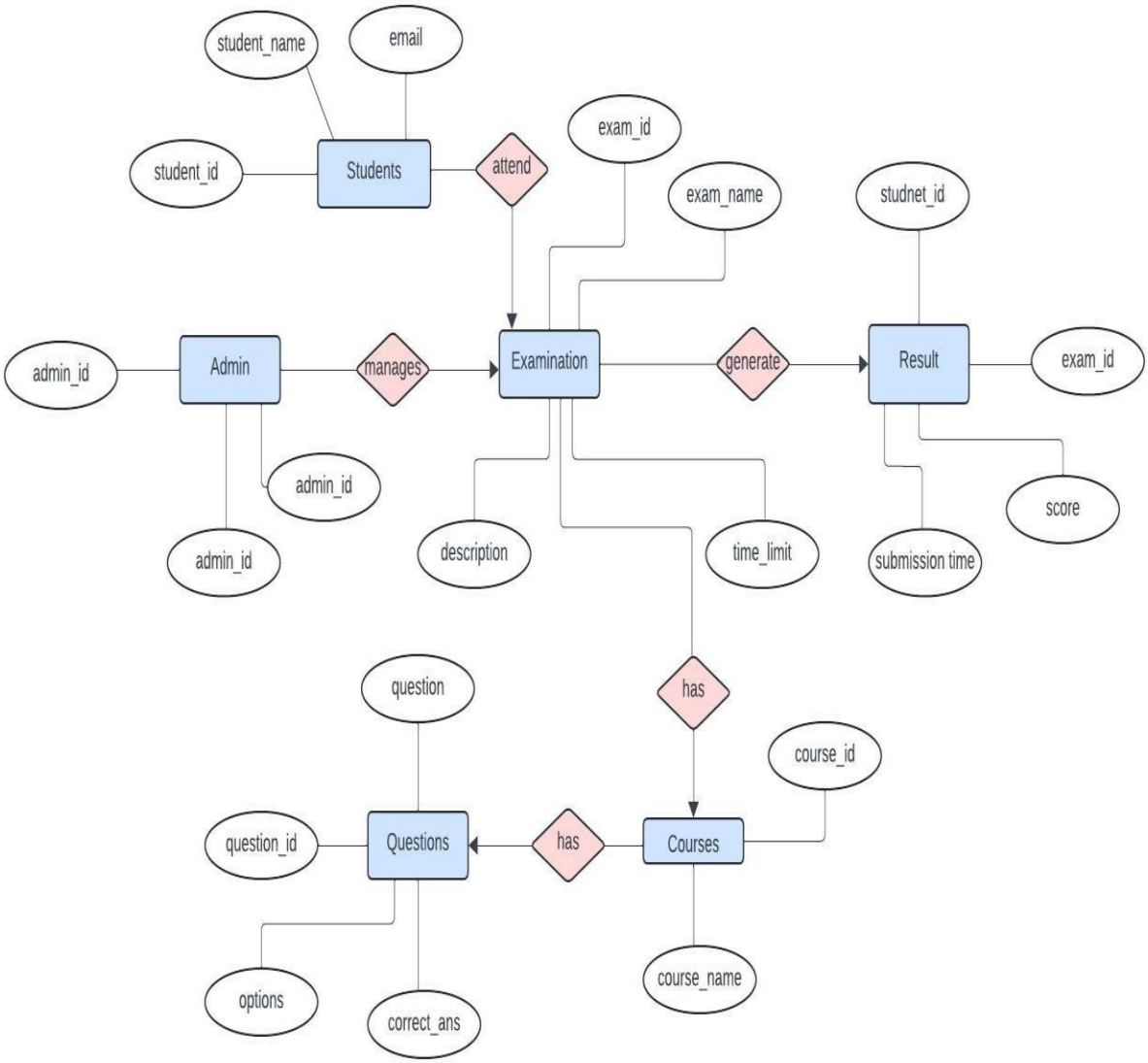
Sequence Diagram for OES

8.4 Class Diagram:



Class Diagram for OES

9. ER Diagrams



ER Diagram for OES

10. Test Cases

Test Case	Test Purpose	Test Condition	Expected Outcome	Actual Result
User Registration	To verify that a user can successfully register in the system.	User provides valid registration details (username, email, password).	User account is created successfully, and the user is redirected to the login page with a success message.	User account is created, and the system displays a success message prompting the user to login.
User Login	To verify that a registered user can log in to the system.	User provides valid login credentials (username/email and password).	User is authenticated and redirected to the main dashboard.	User is successfully authenticated and directed to the main dashboard.
Add courses	To verify the course added in the system	Admin provide name of the course and add it to course	The course is successfully added, and it appears in the list of courses.	The admin successfully adds the course, and it appears in the list of courses.
Delete courses	To verify that a logged-in admin can delete a course.	Admin is logged into the system, and at least one course exists.	The selected course is successfully deleted from the system.	The admin successfully deletes the selected course.
View courses	To verify that a logged-in user can view the list of available courses.	Admin is logged and navigate to courses and view all the added courses	The admin can see the list of available courses with their details.	The admin successfully views the list of available courses.

Add Exam	To verify that a logged-in admin can add a new exam.	Admin add exam by providing exam name, description, time limit and add it	The exam is successfully added for the selected course.	The admin successfully adds the exam for the selected course.
Delete Exam	To verify that a logged-in admin can delete an exam.	Admin is logged into the system, at least one course, and one exam exist.	The selected exam is successfully deleted from the system.	The admin successfully deletes the selected exam.
Add Questions	To verify that a logged-in admin can add a new question to an exam.	Admin logged into the system and provide question with option and correct answer	The question is successfully added to the selected exam.	The admin successfully adds the question to the selected exam.
View Questions	To verify that a logged-in user can view the list of questions for a particular exam.	User is logged into the system, and view the questions and answer of particular exam	The user can see the list of questions associated with the selected exam.	The user successfully views the list of questions for the selected exam.
Take Exam	To verify that a logged-in user can take an exam.	User is logged into the system take exam and submit it	The exam is successfully submitted, and the user can view their result.	The user successfully completes the exam and views the result.