B.M.S COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

OOMD AAT Report on

Online Examination System

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:
M Ananta Naga Rajesh (1BM21CS098)
Mahavir Nahata (1BM21CS100)
Manasa S (1BM21CS101)
Manav Take (1BM21CS102)

Faculty In charge

Dr. Pallavi G B

Assistant Professor

Department of Computer Science and Engineering
B.M.S College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
2023-2024

B.M.S COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, M Ananta Naga Rajesh (1BM21CS098), Mahavir Nahata (1BM21CS100), Manasa S (1BM21CS101) and Manav Take (1BM21CS102), students of 6th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this OOMD Mini Project entitled "MEDWAY" has been carried out in Department of CSE, BMS College of Engineering, Bangalore during the academic semester March - July 2023. I also declare that to the best of our knowledge and belief, the OOMD AAT report is not from part of any other report by any other students.

Signature of the Candidate

M Ananta Naga Rajesh (1BM21CS098)

Mahavir Nahata (1BM21CS100)

Manasa S (1BM21CS101)

Manav Take (1BM21CS102)

BMS COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the OOMD AAT titled “**Online Examination System”** has been carried out by M Ananta Naga Rajesh (1BM21CS098), Mahavir Nahata (1BM21CS100), Manasa S (1BM21CS101) and Manav Take (1BM21CS102), during the academic year 2023-2024.

Signature of the Faculty in Charge

Table of Contents

1. INTRODUCTION

Software engineering is a systematic and disciplined approach to developing, designing, testing, and maintaining software systems. It involves the application of engineering principles to the entire software development process, ensuring the creation of reliable, efficient, and scalable software solutions. Software engineers employ various methodologies, tools, and techniques to manage the complexity of software development, meet user requirements, and deliver high-quality products.

Object-oriented modeling is a key aspect of software engineering that revolves around organizing and designing software systems using the principles of object-oriented programming (OOP). In OOP, software is structured as a collection of objects, which are instances of classes encapsulating data and behavior. Object-oriented modeling emphasizes encapsulation, inheritance, and polymorphism, fostering modular and reusable code. This approach enhances code maintainability, promotes a clear understanding of system architecture, and facilitates collaboration among development teams. By representing real-world entities as objects and modeling their interactions, object-oriented modeling provides a powerful and flexible framework for creating robust and adaptable software systems.

1.1 Purpose

The purpose of this SRS document is to describe the requirements for an Online Examination System. This system will be used to facilitate online exams for educational institutions, ensuring a secure, efficient, and user-friendly experience for both students and examiners.

1.2 Scope

The Online Examination System will support the following functionalities:

- User Registration and Authentication
- Exam Creation and Management
- Exam Scheduling
- Exam Participation
- Result Generation and Management
- System Administration

2. PROBLEM STATEMENT

The problem statement of an online examination system typically revolves around addressing various challenges and requirements associated with conducting exams over the internet.

Here is a generalized problem statement:

Title: Development of an Efficient Online Examination System

In the rapidly evolving landscape of education, there is a growing demand for a seamless and secure online examination system that can cater to the needs of educational institutions, students, and administrators. The current manual examination processes are plagued with inefficiencies, logistical challenges, and security concerns. This project aims to design and implement a robust online examination system that overcomes these issues and provides a user-friendly and reliable platform for conducting exams.

3. Software Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- **Login Page:** Allows users to log in with their credentials.
- **Registration Page:** New users can register by providing necessary details.
- **Dashboard:** Provides users with an overview of their activities and available functionalities.

3.1.2 Hardware Interfaces

- None specifically required; the system will operate on standard web-enabled devices.

3.1.3 Software Interfaces

- Integration with email servers for notifications.
- Possible integration with payment gateways for exam fees.

3.1.4 Communications Interfaces

- HTTPS protocol for secure data transmission.

3.2 Functional Requirements

3.2.1 User Registration and Authentication

- **FR1:** The system shall allow users to register by providing a username, password, and email address.
- **FR2:** The system shall send a verification email upon registration.
- **FR3:** The system shall allow users to log in using their credentials.
- **FR4:** The system shall allow users to reset their password using their email.

3.2.2 Exam Management

- **FR5:** The system shall allow instructors to create, edit, and delete exams.
- **FR6:** The system shall allow instructors to add, edit, and delete questions in a question bank.
- **FR7:** The system shall support various question types, including multiple choice, true/false, and short answer.

3.2.3 Exam Scheduling

- **FR8:** The system shall allow instructors to schedule exams by setting a start date and time, end date and time, and duration.
- **FR9:** The system shall notify students of upcoming exams via email.

3.2.4 Exam Participation

- **FR10:** The system shall allow students to take exams within the scheduled time frame.
- **FR11:** The system shall auto-submit exams when the time limit is reached.
- **FR12:** The system shall allow students to review their answers before final submission.

3.2.5 Result Management

- **FR13:** The system shall automatically grade objective-type questions.
- **FR14:** The system shall allow instructors to manually grade subjective-type questions.
- **FR15:** The system shall generate results and provide feedback to students.

3.2.6 System Administration

- **FR16:** The system shall allow admins to manage user accounts and roles.
- **FR17:** The system shall allow admins to configure system settings.

3.3 Non-Functional Requirements

3.3.1 Performance Requirements

- **NFR1:** The system should support at least 500 concurrent users without performance degradation.
- **NFR2:** The system should load pages within 3 seconds under normal conditions.

3.3.2 Security Requirements

- **NFR3:** The system shall use HTTPS for secure communication.
- **NFR4:** The system shall encrypt sensitive data such as passwords.

3.3.3 Usability Requirements

- **NFR5:** The system shall have an intuitive user interface.
- **NFR6:** The system shall provide help and support documentation.

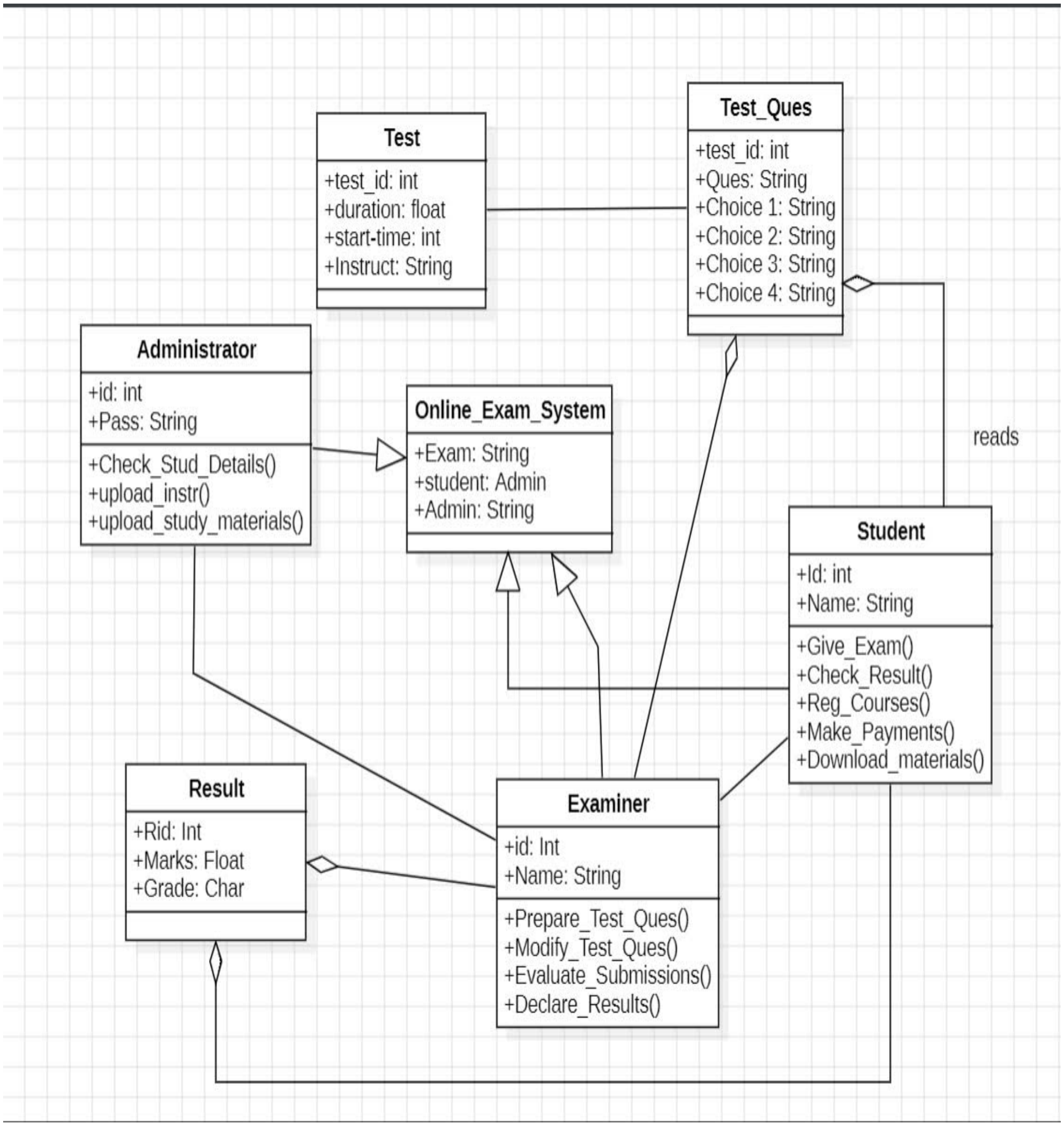
3.3.4 Reliability Requirements

- **NFR7:** The system should have 99.9% uptime.
- **NFR8:** The system should automatically back up data daily.

3.3.5 Maintainability Requirements

- **NFR9:** The system shall be modular to allow easy updates and maintenance.
- **NFR10:** The system shall include comprehensive logging for debugging and audit purposes.

4. CLASS MODELING



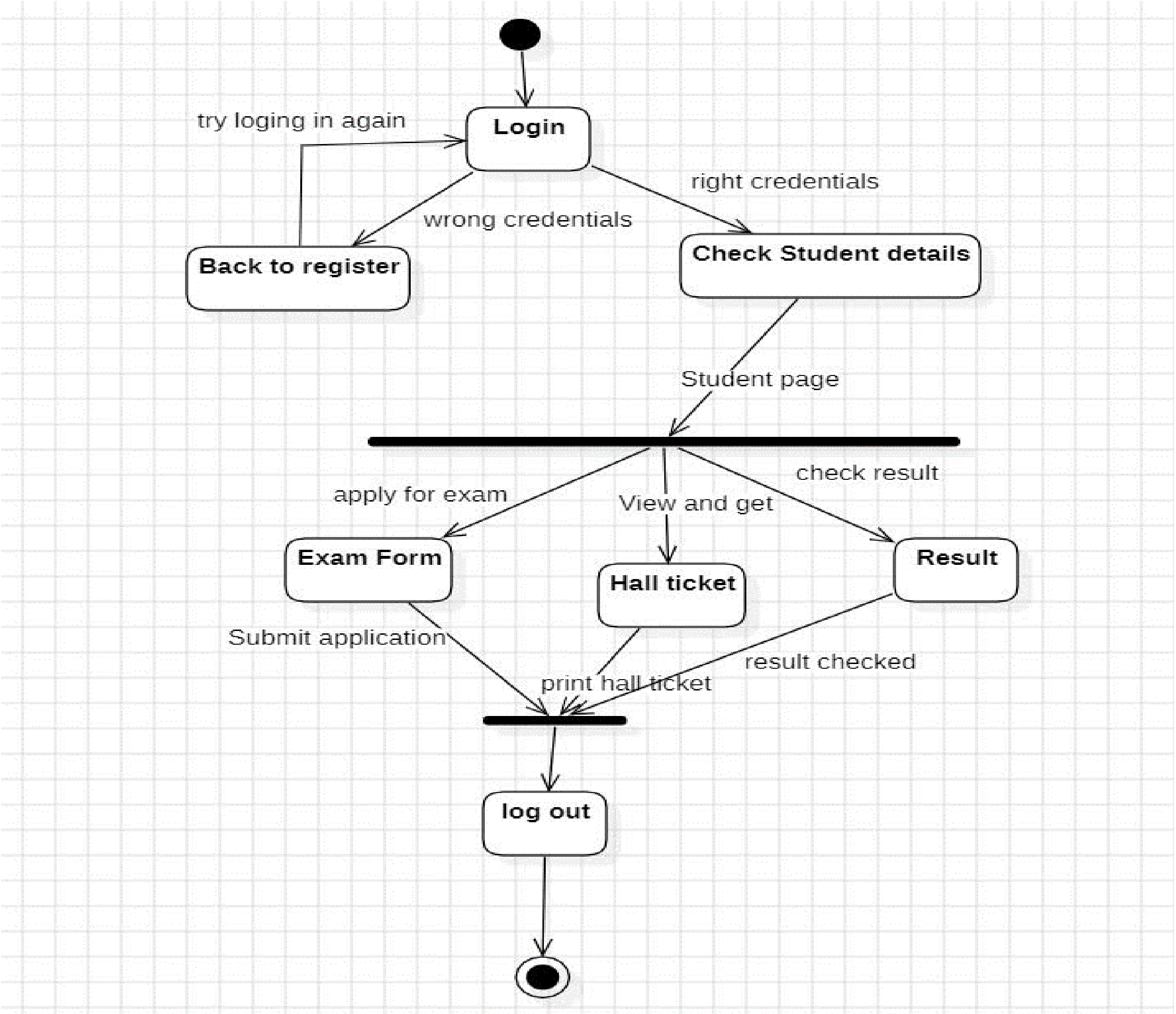
Class diagrams are a fundamental aspect of Unified Modeling Language (UML) used in software engineering to depict the static structure of a system. They provide a visual representation of classes, their attributes, methods, and the relationships between them. Here's a brief note on class diagrams:

In the context of a online examination, a class diagram can help illustrate the structural elements of the system. Here's a conceptual representation using a class diagram:

Class Diagram for Online Examination:

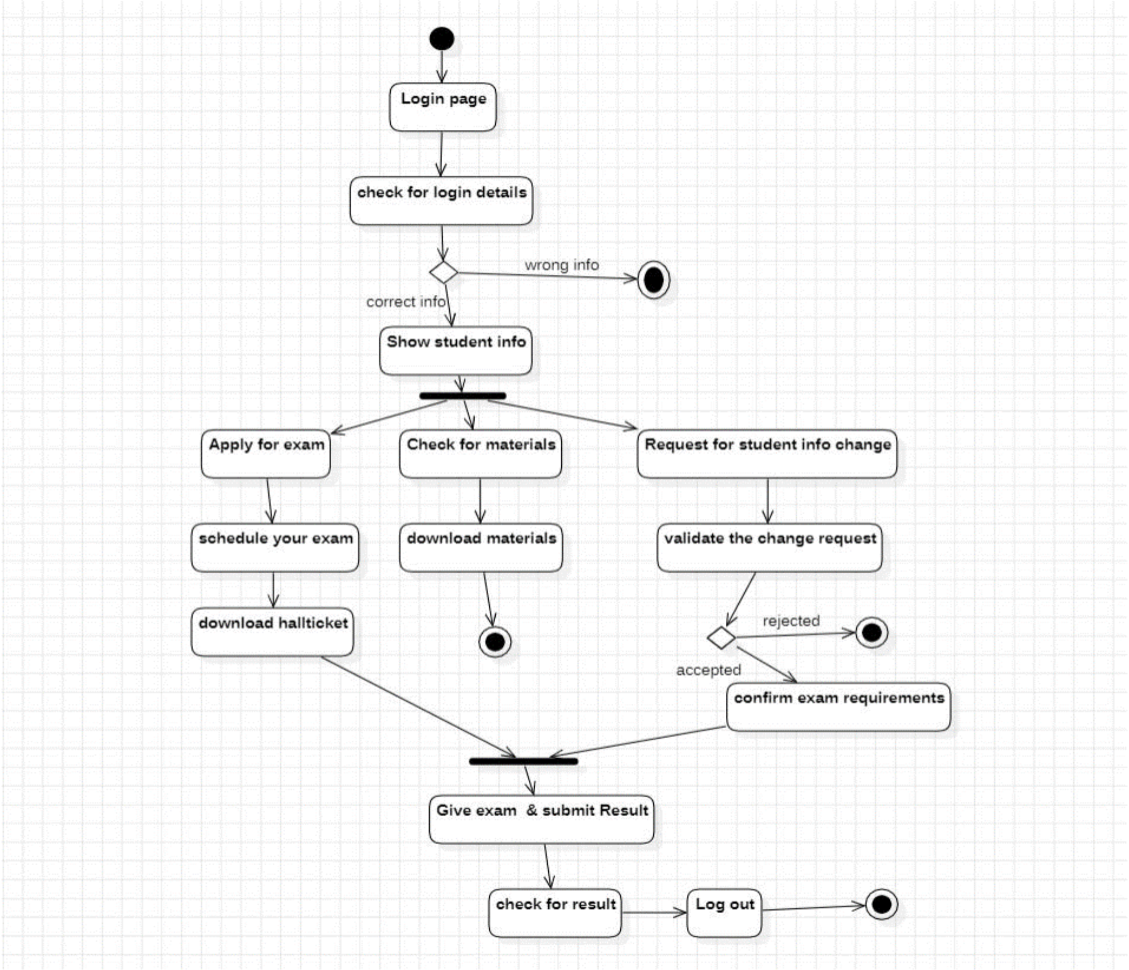
- The central entity is “**Online_Exam_System**” with attributes including Exam, student, and Admin.
- “**Administrator**” is connected to “**Online_Exam_System**”, having attributes like id and Pass. Functions include checking student details and uploading instructions or study materials.
- “**Test**” entity has attributes such as test id, duration, start time, and instruction. It’s linked to the Online Exam System.
- “**Test_Ques**” is another entity with attributes including test id and choices for questions. It reads from the Test entity.
- The “**Student**” entity has attributes like id and Name; functions include giving exams, checking results, registering for courses making payments and downloading materials. It reads from the Test_Ques entity.
- The “**Examiner**” has an ID attribute along with a name attribute; it’s responsible for preparing test questions modifying them evaluating submissions declaring results
- “**Result**” is connected to “**Examiner**”, containing RID Marks Grade as its attributes.

5. STATE DIAGRAM



This image is a flowchart that outlines the process for a student to log in, check details, apply for an exam, and log out on a website or application. The flowchart is structured with labelled rectangles and diamonds connected by arrows, indicating the flow of actions and decisions. It begins with "Login" where there are two outcomes: "try logging in again" if unsuccessful or proceed to "Check Student details" if successful. If login is unsuccessful due to wrong credentials, there's an option to "Back to register". After checking student details, it proceeds to the "Student page" where one can "apply for exam". The "Exam Form" step involves submitting an application. Post-application submission leads to "Hall ticket" where students can view and print their hall tickets. Students can also check their results from the student page which leads them to "Result" after their results have been checked. The final step is "log out".

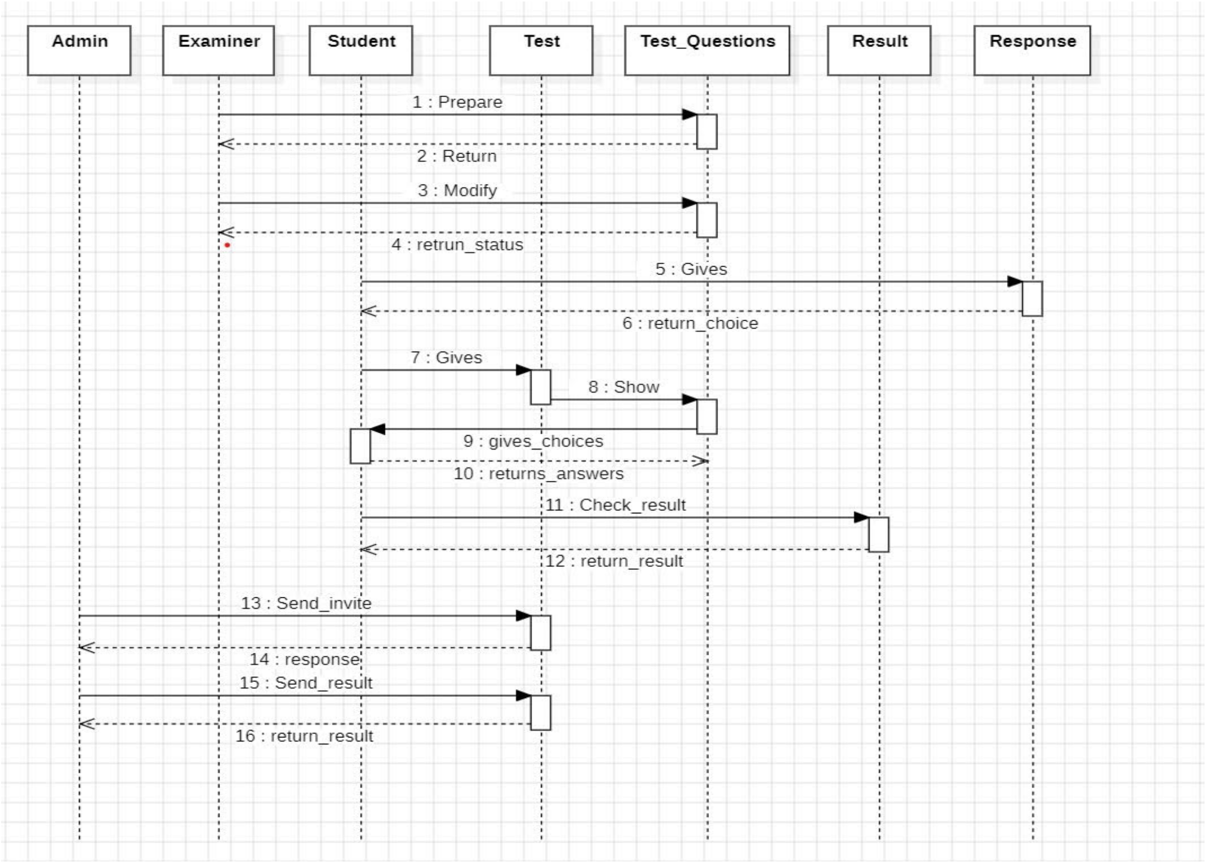
6. ACTIVITY DIAGRAM



This image is a flowchart that outlines the process for students logging in, viewing their information, applying for an exam, and checking their results on a particular platform.

The flowchart is set against a grid background. It begins with the “Login page” where users are prompted to enter their login details. There’s a decision point where if the login details are correct, it proceeds to “Show student info”; if wrong, it loops back to re-enter login details. From “Show student info”, there are three paths: “Apply for exam” leads to scheduling the exam and downloading the hall ticket before proceeding to “Give exam & submit Result”. “Check for materials” allows users to download materials. “Request for student info change” leads to validating the change request which can either be accepted or rejected. If accepted, it confirms exam requirements before moving forward. After giving the exam and submitting results, users can check for results and then log out.

7. SEQUENCE DIAGRAM



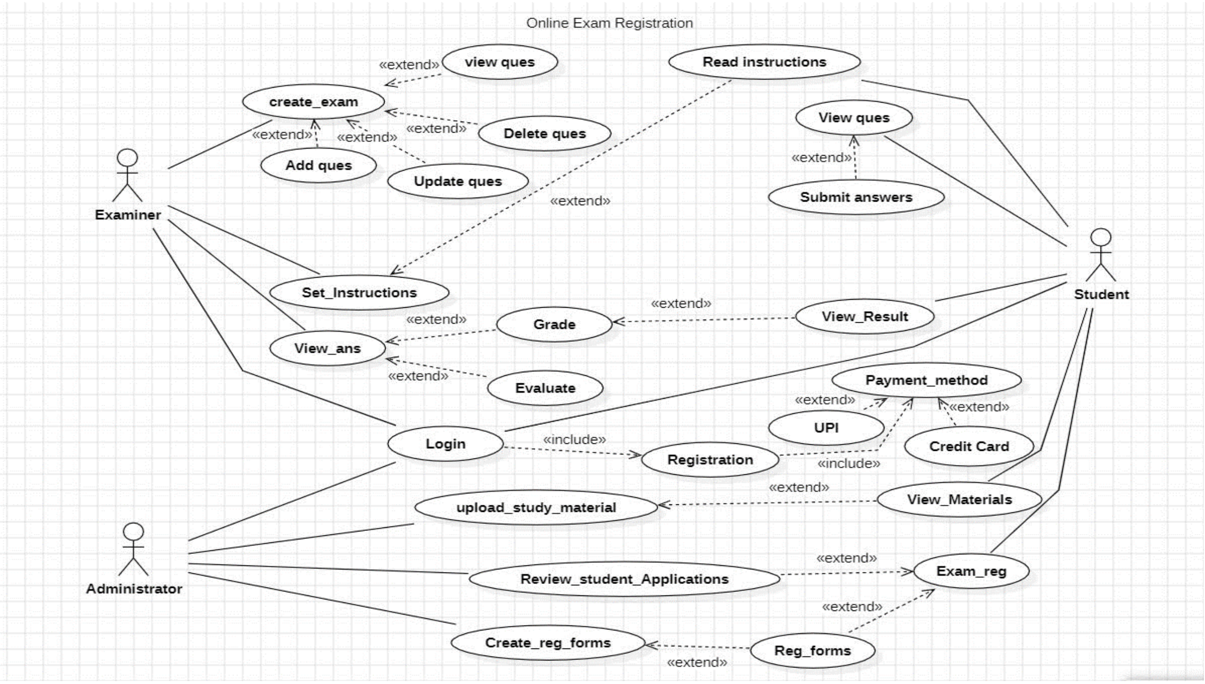
A sequence diagram in the context of a social media platform serves as a dynamic visualization tool that illustrates the interactions between various components, primarily focusing on the chronological order of messages and actions exchanged among users and the system. This diagram provides a step-by-step representationof how different entities collaborate in real-time.

It shows the process flow of a test preparation and result generation system involving Admin, Examiner, Student, Test, Test_Questions, Result, and Response.

The diagram is on a grid background with vertical lifelines labelled Admin, Examiner, Student, Test, Test_Questions, Result and Response. The Admin interacts with the Examiner by preparing (1), returning (2), modifying (3) and returning status (4). The Examiner gives the test questions to the Student (5) and returns choice to the Test (6). The Test shows choices to the Student (8) after receiving them from the Examiner who gives choices through interaction with Test_Questions (7).

The Student returns answers to the Test_Questions (10), which then checks results and returns them to Result entity(11). Results are returned from Result entity to Response entity(12). Admin sends an invite to Response entity(13), receives response(14) and sends result back to it(15). Finally return result is sent back from Response entity to Admin(16).

8. USE CASE DIAGRAM



Use case diagrams are a type of Unified Modeling Language (UML) diagram used in software engineering to visually represent the interactions between various actors (users or external systems) and a system. These diagrams provide a high-level overview of the system's functionalities and the ways in which users or external entities interact with it.

There are three main entities involved: **Examiner**, **Administrator**, and **Student**. The **Examiner** has tasks like creating exams, viewing questions, adding or updating questions, setting instructions, grading and evaluating answers. The **Administrator** is responsible for logging in, uploading study material, reviewing student applications, and creating registration forms. The **Student** is involved in reading instructions, viewing questions, submitting answers, viewing results, and choosing payment methods for exam registration.

Arrows indicate interactions between tasks or steps in processes like “extend” or “include”. Payment methods listed include UPI and Credit Card

CONCLUSION

In conclusion, Unified Modelling Language (UML) diagrams serve as indispensable tools in the software development lifecycle, offering a standardized and visual means of communication among stakeholders. UML diagrams, including use case diagrams, class diagrams, activity diagrams, sequence diagrams, and state machine diagrams, collectively provide a comprehensive and structured representation of a system's architecture, functionality, and dynamic behaviour. These diagrams facilitate effective collaboration between development teams, designers, and non-technical stakeholders by providing a common visual language.They aid in requirements analysis, system design, and documentation, fostering a clearer understanding of complex systems. UML diagrams enhance the development process by serving as blueprints for coding, testing, and maintenance,ultimately contributing to the creation of robust, scalable, and well-designed software systems. Their versatility makes them an integral part of modern softwareengineering, guiding the development of systems that meet user needs while adhering to industry standards and best practices. UML diagrams provide a comprehensive and visual representation of the online examination system. They enable stakeholders to understand the system's structure, functionality, and behaviour. These diagrams serve as valuable tools for communication among development teams, stakeholders, and anyone involved in the design and implementation of the online examination system. They help ensure that the system meets the desired requirements and functions smoothly.

BIBLIOGRAPHY

1. STARUML SOFTWARE
2. Chat.openai.com
3. https://creately.com
4. https://www.coursera.org/learn/software-engineering-modeling-software-systems-using-uml
5. https://ieeexplore.ieee.org/document/5260494