

# SRS FOR ONLINE EXAM MANAGEMENT SYSTEM

## **1.INTRODUCTION:**

The advancement of digital education demands scalable and secure platforms for conducting academic assessments. Traditional examination methods pose challenges in terms of logistics, manual evaluation, and accessibility. The **Online Exam Management System (OEMS)** aims to transform this landscape by providing a centralized, efficient, and secure platform for conducting examinations remotely or on campus.

### **1.1 Purpose:**

The Online Exam Management System (OEMS) is designed to streamline and automate the examination process for academic institutions and training organizations. Its primary purpose is to enable secure, efficient, and scalable exam delivery and result management through a web-based platform.

### **1.2 Document Convention:**

This document adheres to IEEE Standard 830-1998 for Software Requirements Specifications. It uses standard terminology and organized sections to clearly outline functional and non-functional requirements, interfaces, and overall system descriptions.

### **1.3 Intended Audience and Reading System:**

The SPMS is designed for individuals across various professions and industries who rely on software tools to perform daily tasks.

This document is intended for Developers, Project Managers, Quality Assurance Teams, Institutional Stakeholders, Students and Faculty (as end users)

### **1.4 Product Scope:**

The OEMS will include:

- Secure login for admins, instructors, and students
- Exam creation and scheduling by instructors
- Timed online exams with multiple question types
- Auto-evaluation for MCQs and manual grading for descriptive responses
- Result publication and reporting
- Plagiarism checks and activity monitoring
- Accessible UI for ease of use across devices

### **1.5 References:**

- IEEE SRS Template Guidelines
- Research on online assessment platforms
- Academic integrity and proctoring best practices
- Data security standards like ISO 27001, OWASP

## **2. FEASIBILITY STUDIES**

Feasibility studies help determine the practicality and viability of the proposed Online Exam Management System in various dimensions. These studies ensure that the system is worth pursuing in terms of technology, cost, operations, and legality.

### **2.1 Technical Feasibility**

The Online Exam Management System is technically feasible using modern, widely supported, and scalable technologies. The application can be developed using a layered architecture consisting of frontend, backend, and database layers.

#### **Frontend Technologies:**

- ReactJS or Angular can be used to develop a responsive and dynamic user interface.
- HTML5, CSS3, and JavaScript will handle the basic structure, styling, and interactivity.
- Bootstrap or TailwindCSS can be used for UI components and responsiveness.

#### **Backend Technologies:**

- Node.js (with Express) for asynchronous, scalable backend services.
- Django (Python) for rapid development and built-in admin functionalities.
- Laravel (PHP) for MVC structure and security features.

#### **Database:**

- MySQL or PostgreSQL for relational data storage such as user accounts, exams, questions, and results.
- MongoDB can be optionally used for storing logs and session data due to its flexibility with unstructured data.

#### **Deployment & Hosting:**

- Cloud platforms like AWS, Render, Heroku, or Firebase provide scalable hosting, CI/CD pipelines, and database-as-a-service.
- Docker can be used for containerized deployment.

These tools and platforms are open-source, well-documented, and supported by large developer communities, ensuring ease of development, bug resolution, and long-term maintainability.

## **2.2 Economic Feasibility**

From an economic standpoint, the system is highly feasible as it offers long-term cost benefits:

### **Low Initial Investment:**

- The use of open-source tools and frameworks eliminates licensing fees.
- Cloud services like Render or Vercel offer free-tier hosting for small-scale applications.

### **Reduced Operational Costs:**

- Automating the exam process eliminates costs associated with printing papers, logistics, exam centers, and manual evaluation.

### **Cost of Human Resources:**

- Once deployed, the system minimizes the need for administrative staff, thereby reducing recurring HR expenses.

### **ROI (Return on Investment):**

- The system can serve multiple academic batches across semesters without additional cost, offering excellent ROI for institutions.

## **2.3 Operational Feasibility**

Operational feasibility assesses how well the system meets the needs of users and institutions in real-world conditions.

### **Ease of Use:**

- \* Designed with a user-friendly interface for students, instructors, and admins.
- \* Tutorials, tooltips, and responsive dashboards will ensure a smooth user experience.

### **Adaptability:**

- Can be integrated with existing Learning Management Systems (LMS).
- Modular design supports future expansion such as webcam-based proctoring or AI plagiarism detection.

### **Training Requirements:**

- Minimal training needed due to intuitive interfaces.
- Training sessions for staff can be conducted via recorded tutorials or live webinars.

### **Reliability:**

- Automated exam creation, auto-evaluation, and centralized result management ensure consistent performance and reduce the scope of human errors.

## **2.4 Legal Feasibility**

Legal feasibility ensures that the system adheres to all laws and regulations relevant to digital data, education, and examinations.

### **Data Privacy Compliance:**

The system will comply with global data protection laws such as:

- GDPR (General Data Protection Regulation – EU)
- FERPA (Family Educational Rights and Privacy Act – USA)
- IT Act (India)

### **Academic Integrity:**

- Incorporates plagiarism checks and logging mechanisms to uphold academic ethics.
- Exam monitoring tools will log IP addresses, timestamps, and actions to discourage malpractice.

### **Security & Consent:**

- Collects only necessary personal information with user consent.
- Data storage and processing will be secure and transparent, maintaining user trust and legal compliance.

## **3. REQUIREMENT ANALYSIS**

Requirement analysis is the critical phase in the software development life cycle where the needs of all stakeholders are identified, analyzed, and documented to ensure the system meets its intended purpose. In the context of the Online Exam Management System (OEMS), this phase helps in understanding what different users expect from the system and how the system will fulfill those expectations.

The goal of requirement analysis is to bridge the gap between what users want and what the development team will build. It outlines the functional behavior of the system, the interactions between users and the software, and the data flow across modules.

### **3.1 Stakeholders and Roles**

Understanding the roles and responsibilities of each stakeholder helps in designing tailored interfaces and allocating appropriate access privileges. The OEMS involves the following key stakeholders:

#### **1. Administrator**

- The Administrator holds the highest level of access in the system and is responsible for platform-level control and supervision.

**Responsibilities:**

- User Management: Create and manage accounts for students, instructors, and other admins.
- Role Assignment: Define and control access permissions based on user roles.
- Configuration Management: Set up institution-specific configurations such as semester timelines, exam rules, and grading policies.
- Monitoring and Auditing: View activity logs, manage security alerts, and ensure compliance with institutional policies.
- Report Generation: Generate system-wide reports on exams, user activity, performance statistics, and attendance.

**2. Instructor / Faculty**

Instructors are academic staff members who conduct the examinations and oversee student assessments.

**Responsibilities:**

- Exam Creation: Design and configure exams by specifying exam titles, durations, total marks, and availability windows.
- Question Management: Add, edit, and organize questions in the exam. Questions may be of multiple types including MCQs, short answers, and essay-type.
- Exam Scheduling: Assign exams to particular student groups or courses with defined time slots.
- Evaluation: Review student submissions, especially for subjective-type questions, and assign marks accordingly.
- Feedback and Results: Publish results, provide individual feedback, and respond to student queries related to the exam.

**3. Student**

Students are the end users who take the exams and interact with the system primarily through the exam and result interfaces.

**Responsibilities:**

- Account Access: Log into the system securely using assigned credentials.
- Exam Participation: View scheduled exams, read instructions, and attempt them within the assigned duration.
- Submission: Submit their answers before the deadline; system should auto-submit if time expires.
- Result Access: View graded results, examiner comments, and performance analytics post-evaluation.

## 4. System

The system is the software backbone that performs automated functions and ensures smooth operations of the application.

### Responsibilities:

- Authentication: Verify login credentials and enforce role-based access control.
- Automation: Auto-grade objective questions like MCQs using predefined answers.
- Data Management: Store and retrieve user data, exam submissions, results, and logs.
- Notifications: Send reminders or alerts via email, SMS, or in-app notifications.
- Logging and Monitoring: Record activities such as login/logout, exam start/end, and unusual behaviors like multiple tab switches (for anti-cheating).

## 3.2 User Stories

User stories are short, simple descriptions of a feature told from the perspective of the user. These help developers understand the user's needs in plain language.

### Student User Stories

- As a student, I want to log in securely so that I can access my personalized dashboard.
- As a student, I want to view the list of upcoming exams and their scheduled time so that I can prepare accordingly.
- As a student, I want to receive reminders before my exam starts so I don't miss any tests.
- As a student, I want to take the exam online within a fixed time limit and be able to navigate between questions easily.
- As a student, I want to view my grades and feedback after evaluation so that I can understand my performance.

### Instructor User Stories

- As an instructor, I want to create exams with different types of questions so I can effectively assess students.
- As an instructor, I want to schedule exams for specific groups of students so that they are organized properly.
- As an instructor, I want to evaluate students' responses, especially descriptive ones, and assign marks.
- As an instructor, I want to view analytical reports on exam performance to assess the overall result trends.
- As an instructor, I want to send feedback or messages to students after the exam to improve learning outcomes.

## **Administrator User Stories**

- As an admin, I want to manage user accounts so that only authorized users can access the system.
- As an admin, I want to configure system settings such as exam rules and grading policies.
- As an admin, I want to view audit logs and track system usage to ensure platform integrity and compliance.
- As an admin, I want to generate comprehensive reports of exams, user activities, and overall system performance.
- As an admin, I want to receive alerts about system anomalies or security threats so that timely actions can be taken.

## **4. SOFTWARE REQUIREMENT SPECIFICATION (SRS)**

A Software Requirements Specification (SRS) is a comprehensive document that outlines and describes the complete set of software requirements for a system or application. It acts as a formal agreement between stakeholders (such as clients, users, and developers) on what the software is expected to do and how it should behave.

### **4.1 Functional Requirements**

#### **1. User Registration and Authentication:**

- Role-based login system.
- Secure password encryption (bcrypt).
- Forgotten password recovery via OTP/email.

#### **2. Exam Management:**

- Create, edit, delete exams with time limits.
- Schedule exams and assign them to specific batches or students.

#### **3. Question Bank:**

- Supports multiple question types: MCQ, true/false, fill-in-the-blanks, descriptive.
- Randomized question order per student for MCQs.

#### **4. Exam Interface:**

- Timer countdown and autosave of answers.
- Navigation between questions.
- Submission alert before time expiry.

## **5. Auto-Evaluation & Manual Grading:**

- MCQs are auto-evaluated.
- Instructors manually review and score subjective answers.
- Negative marking and partial scoring supported.

## **6. Result Management:**

- Generate result sheets per student or batch.
- Export results in PDF/Excel.
- Students can view results and instructor feedback.

## **7. Notification System:**

- Email/SMS for scheduled exams, results, and password resets.

## **8. Activity Logging and Proctoring (Optional):**

- Logs student login, submission time, and IP address.
- Detects tab switching and inactivity.

## **9. Feedback System:**

- Students can rate exams or provide feedback.
- Instructors can leave review comments on subjective answers.

## **4.2 Non-Functional Requirements**

### **1. Performance Requirements:**

- Response time should be < 2 seconds under normal load.
- Should support 1000+ concurrent users.

### **2. Scalability:**

- System must scale both vertically (adding resources) and horizontally (adding servers).
- Microservices architecture preferred for large-scale adoption.

### **3. Security:**

- Role-based access control.
- HTTPS for secure data transmission.



Encrypted storage for user data and exam submissions.

#### **4. Maintainability:**

- Clean codebase with modular architecture.
- Comprehensive documentation for developers and users.

#### **5. Usability:**

- Intuitive interface, mobile-compatible.
- Accessibility compliant (e.g., WCAG for screen readers).

#### **6. Reliability and Availability:**

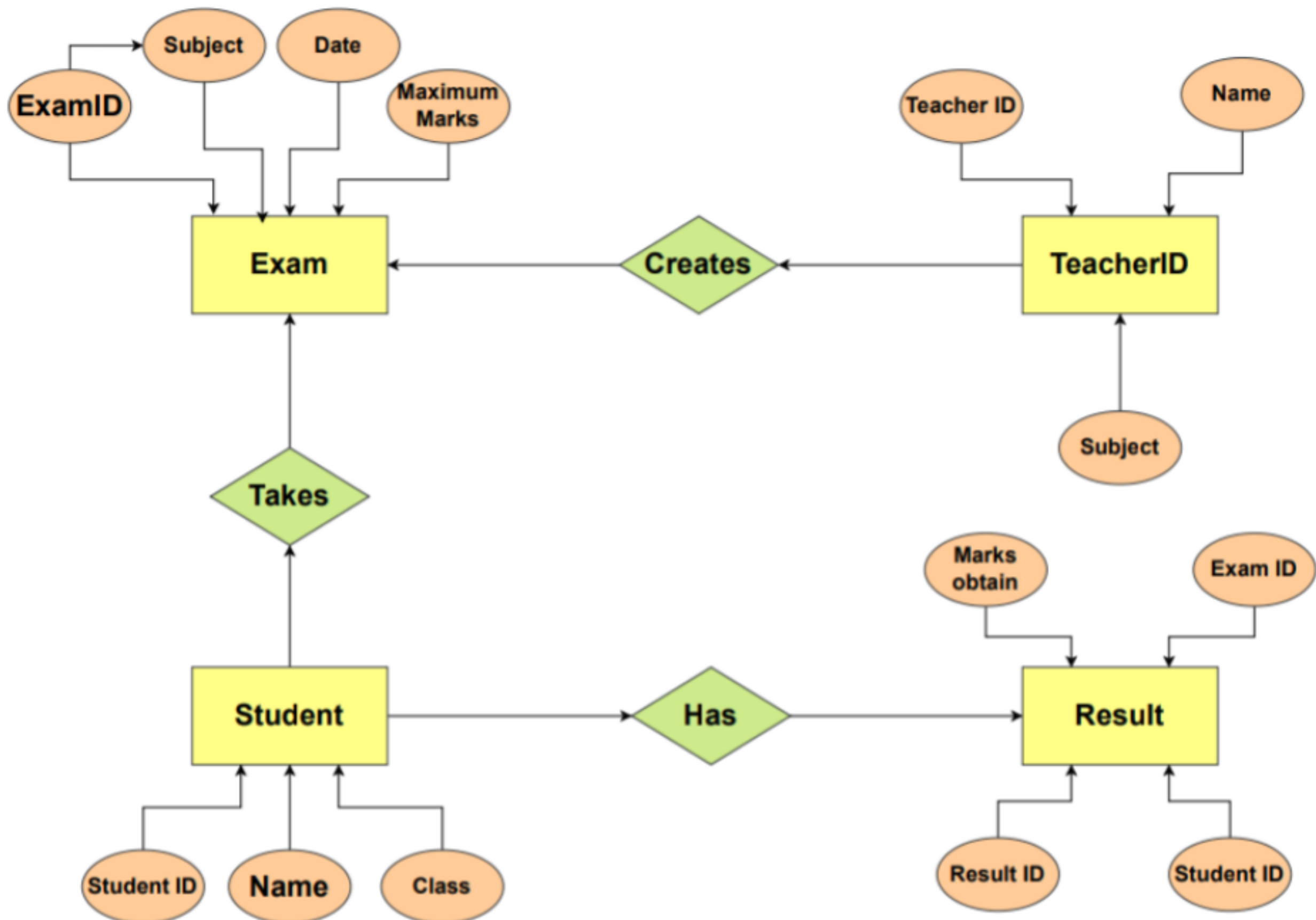
- System availability of at least 99.9%.
- Backup mechanisms for exam and result data.

#### **7. Legal and Ethical Compliance:**

- Conformance with educational data laws.
- Ensures fairness, non-discrimination, and academic integrity.

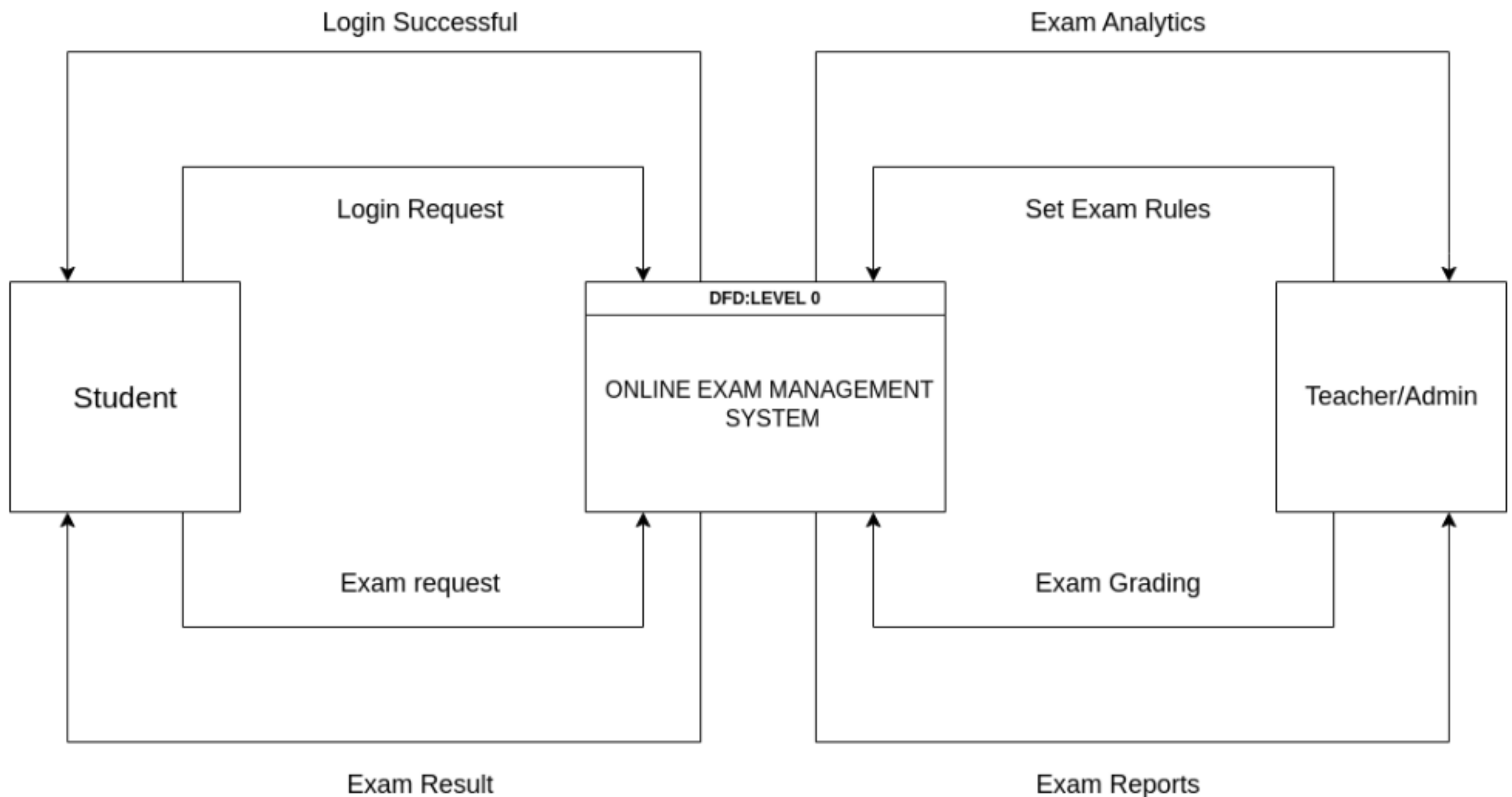
# **Online Exam Management System**

## **ER Diagram**



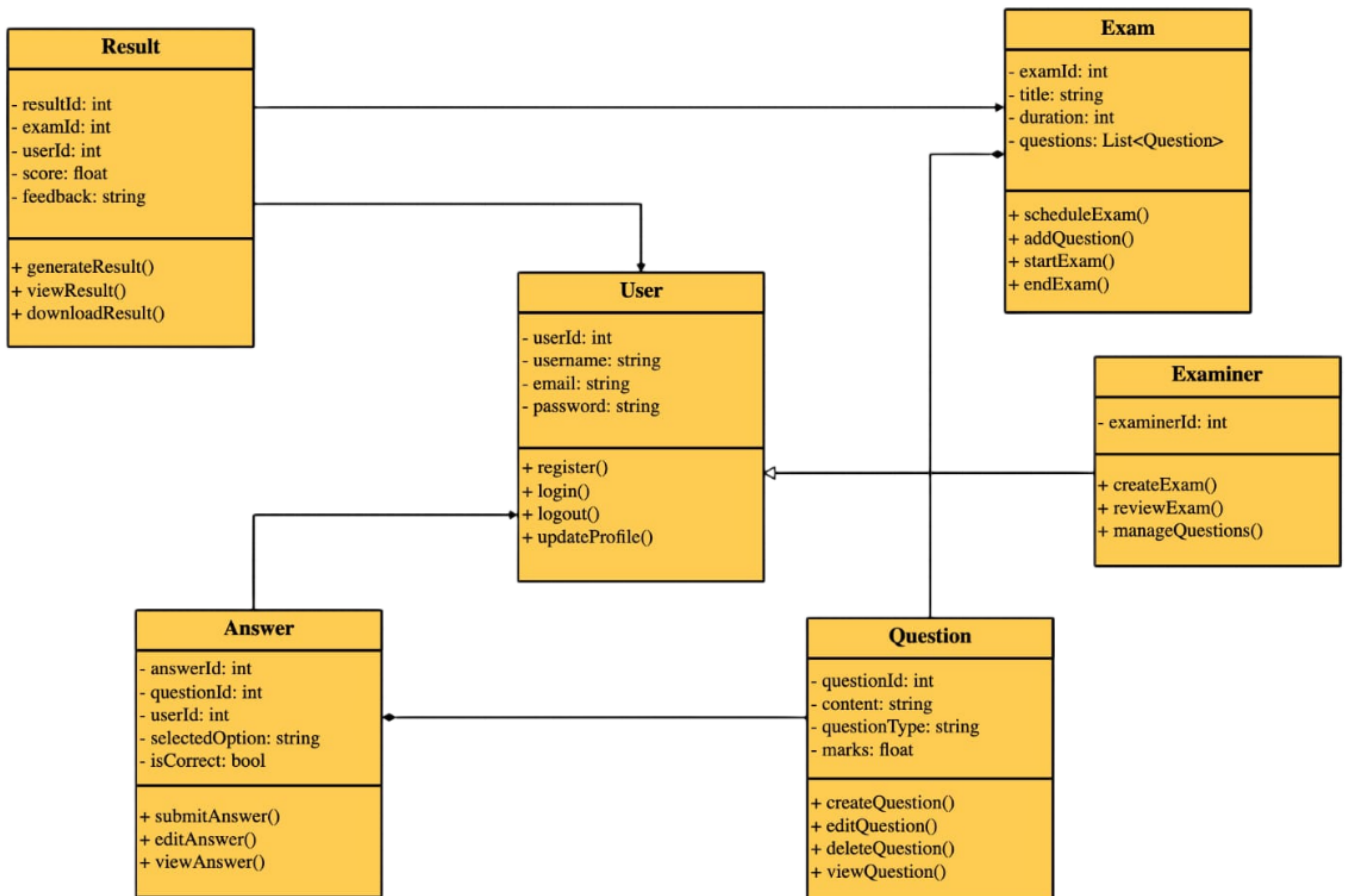
# ***Online Exam Management System***

## **DFD: Level 0 Diagram**



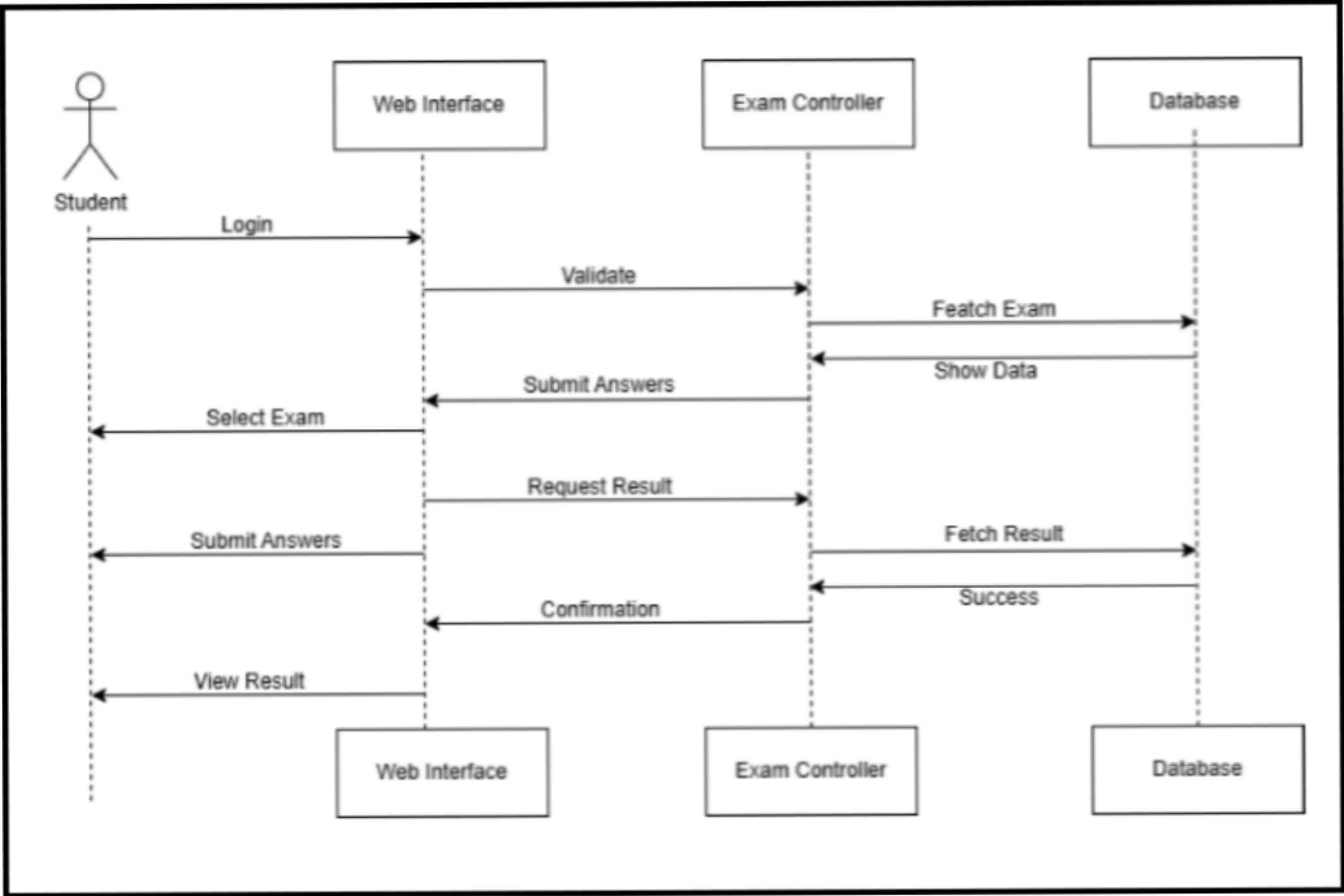
# Online Exam Management System

## Class Diagram



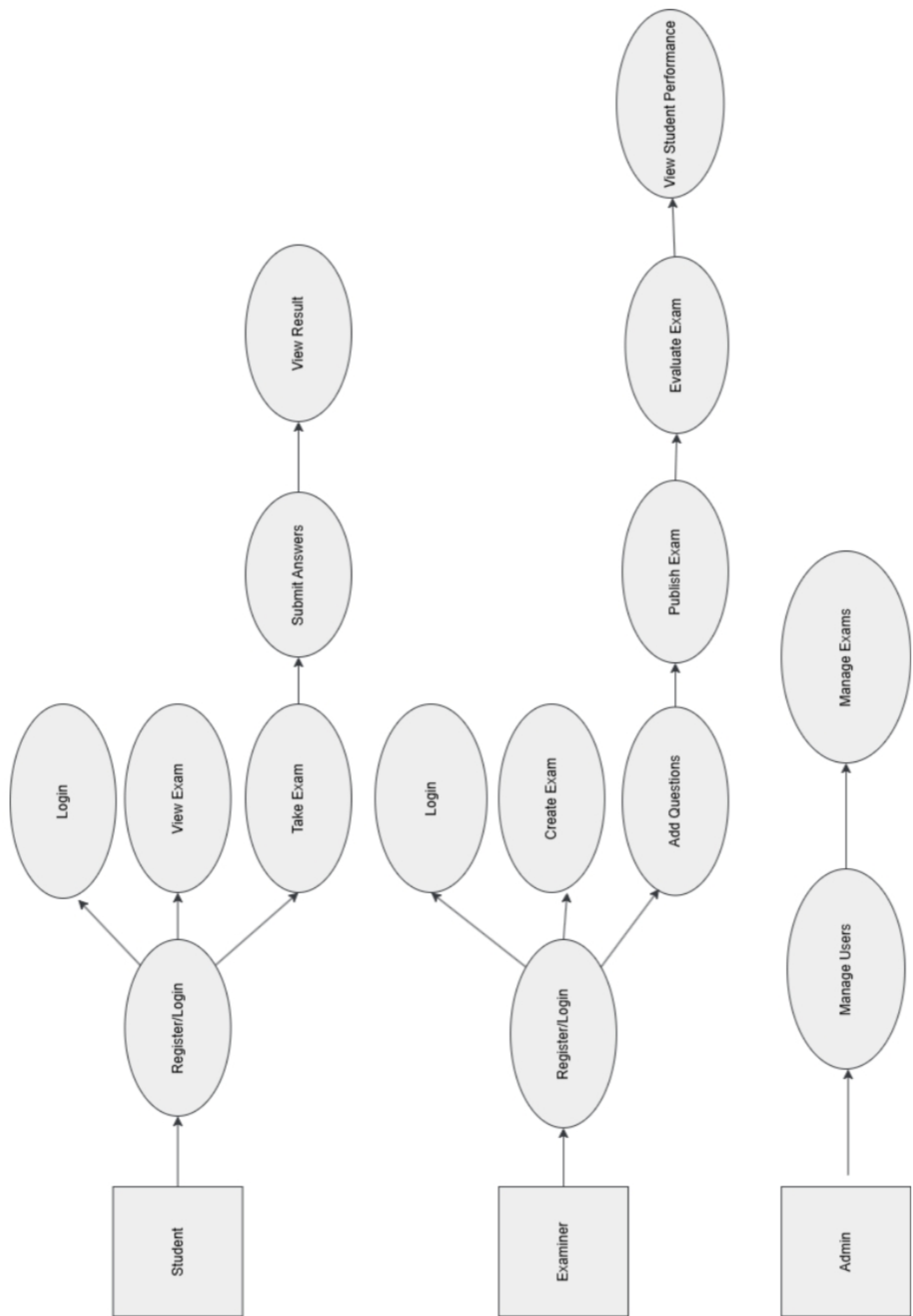
# Online Exam Management System

## Sequence Diagram



# *Online Exam Management System*

## Use Case Diagram



# Online Exam Management System

## Activity Diagram

