Software Requirements Specification (SRS)

For: Coursera Online Learning System
Course: Object-Oriented Analysis and Design / Software Engineering
Prepared by: Swasti Dewan, Garv Sachdeva, Mayank Vij

Date: 17th November,2025

---

1. Introduction

1.1 Purpose

The purpose of this SRS is to describe the functional and non-functional requirements of the **Coursera Online Learning System**. It serves as a detailed guide for developers, designers, and stakeholders to ensure a common understanding of the software's objectives, scope, and expected behavior.

1.2 Scope

The Coursera System is an online learning platform that allows **students** to enroll in courses, **instructors** to create and manage content, and **administrators** to oversee the system.
The software provides course browsing, video lectures, assessments, progress tracking, and certification features.
It follows a **3-tier architecture** and **OOAD modeling principles** for modularity and scalability.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
LMS	Learning Management System
UI	User Interface
DFD	Data Flow Diagram
UML	Unified Modeling Language

OOAD	Object-Oriented Analysis and Design

1.4 References

- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- Coursera Official Website (https://www.coursera.org)
- Sommerville, Ian. *Software Engineering*, 10th Edition

---

2. Overall Description

2.1 Product Perspective

The Coursera System is a standalone web-based application. It follows a **three-tier architecture** consisting of:

- **Presentation Layer:** Web-based UI for users.
- **Application Layer:** Handles business logic and user interactions.
- **Database Layer:** Stores user data, course content, and progress.

2.2 Product Functions

- User registration, login, and profile management.
- Course creation, uploading, and editing by instructors.
- Course search, enrollment, and learning by students.
- Quiz and assessment management.
- Certificate generation after course completion.
- Administrative monitoring and reporting.

2.3 User Characteristics

User	Description
Student	Learns from courses, takes quizzes, and earns certificates.
Instructor	Creates and manages course material.
Administrator	Maintains system and user data.

2.4 Constraints

- Must run on standard web browsers (Chrome, Firefox, Edge).
- Requires stable internet connectivity.
- Data security and user privacy must be ensured.
- The system must support at least 5,000 concurrent users.

2.5 Assumptions and Dependencies

- Users have valid email addresses for account creation.
- Courses are available in video, text, and quiz formats.
- Hosted on cloud infrastructure (AWS / Azure).

---

3. Specific Requirements

3.1 Functional Requirements

ID	Requirement	Description
FR-1	User Registration	The system shall allow users to register using an email ID.
FR-2	Login/Authentication	The system shall authenticate users with credentials.
FR-3	Course Management	Instructors shall be able to create and modify courses.

FR-4	Enrollment	Students shall enroll in available courses.
FR-5	Content Viewing	Students shall view video lectures and course materials.
FR-6	Assessment	The system shall provide quizzes and grade them automatically.
FR-7	Certification	Certificates shall be generated upon course completion.
FR-8	Admin Controls	Admin shall manage users, content, and monitor analytics.

3.2 Non-Functional Requirements

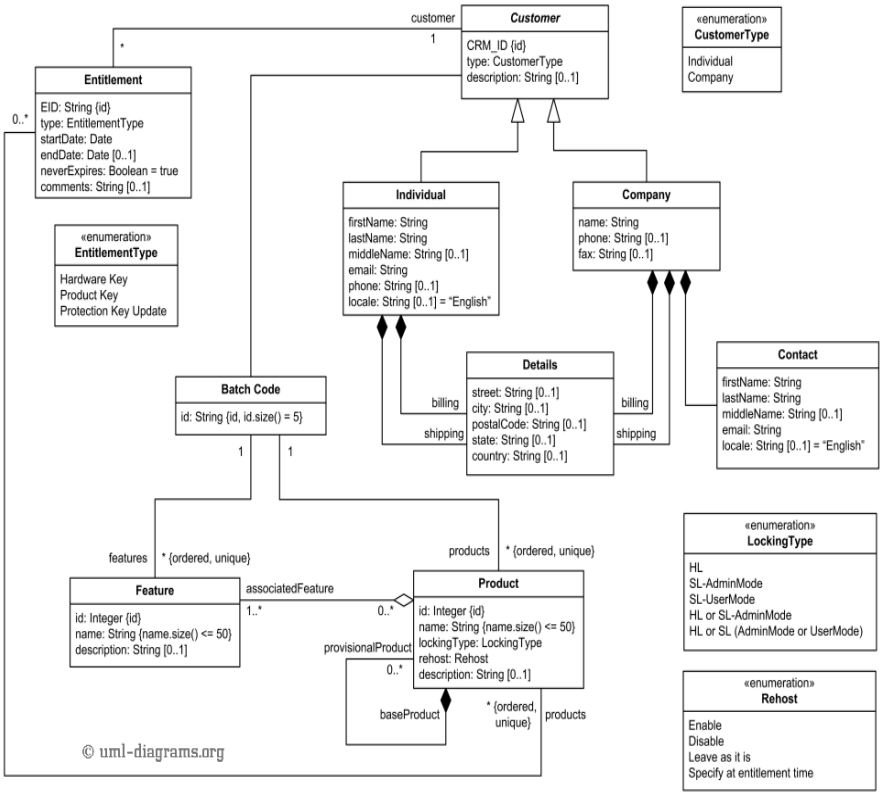
Category	Description
Performance	The system should load within 3 seconds for any page.
Security	Passwords must be encrypted using hashing.
Usability	Interface should be intuitive and easy to navigate.
Scalability	Should handle increasing user base efficiently.
Availability	System uptime must be at least 99%.
Maintainability	Modular architecture to simplify updates and bug fixes.

---

4. System Models

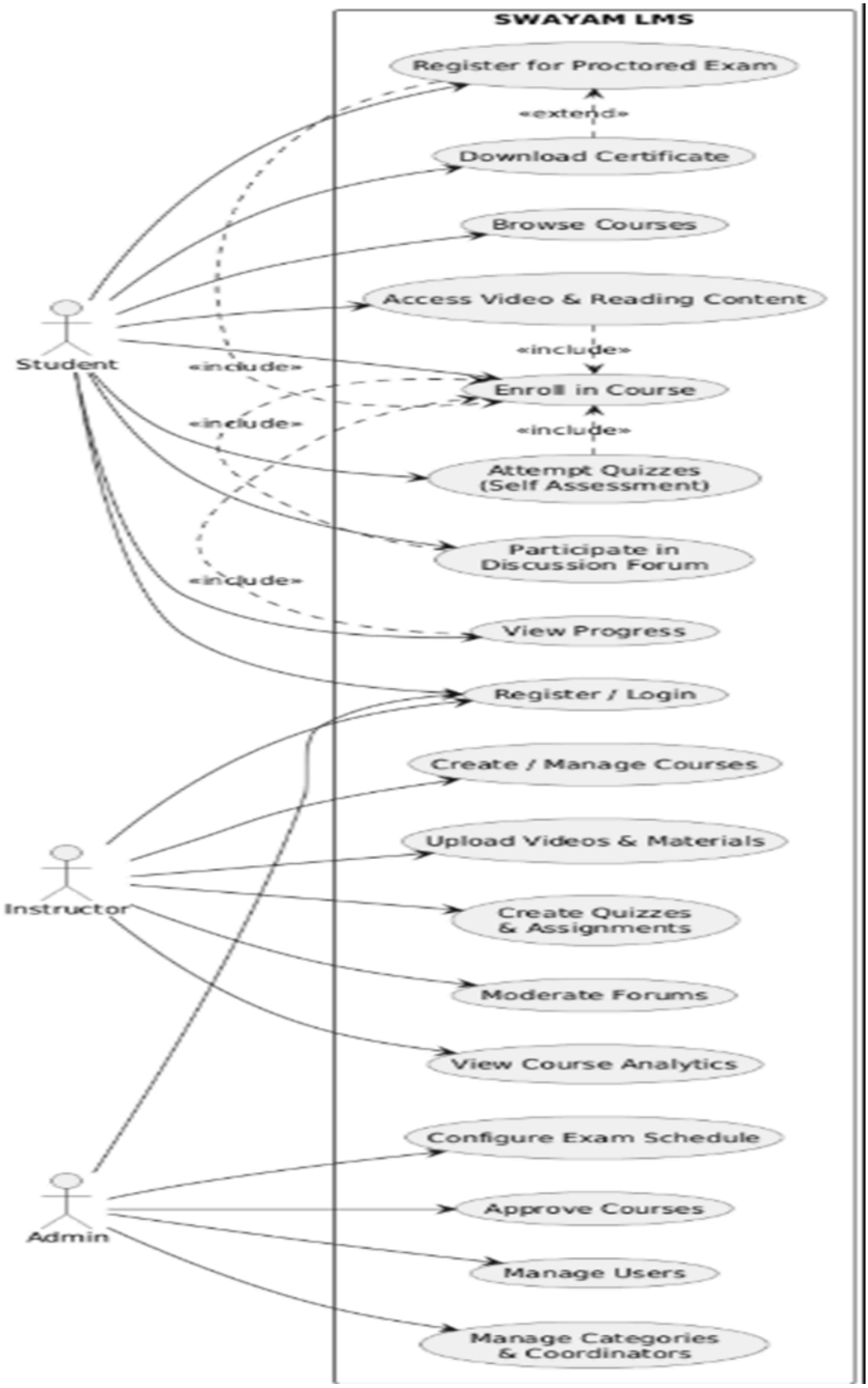
4.1 Structural Modeling

- **Class Diagram:** Represents entities like *User*, *Course*, *Enrollment*, *Assessment*, and their relationships.

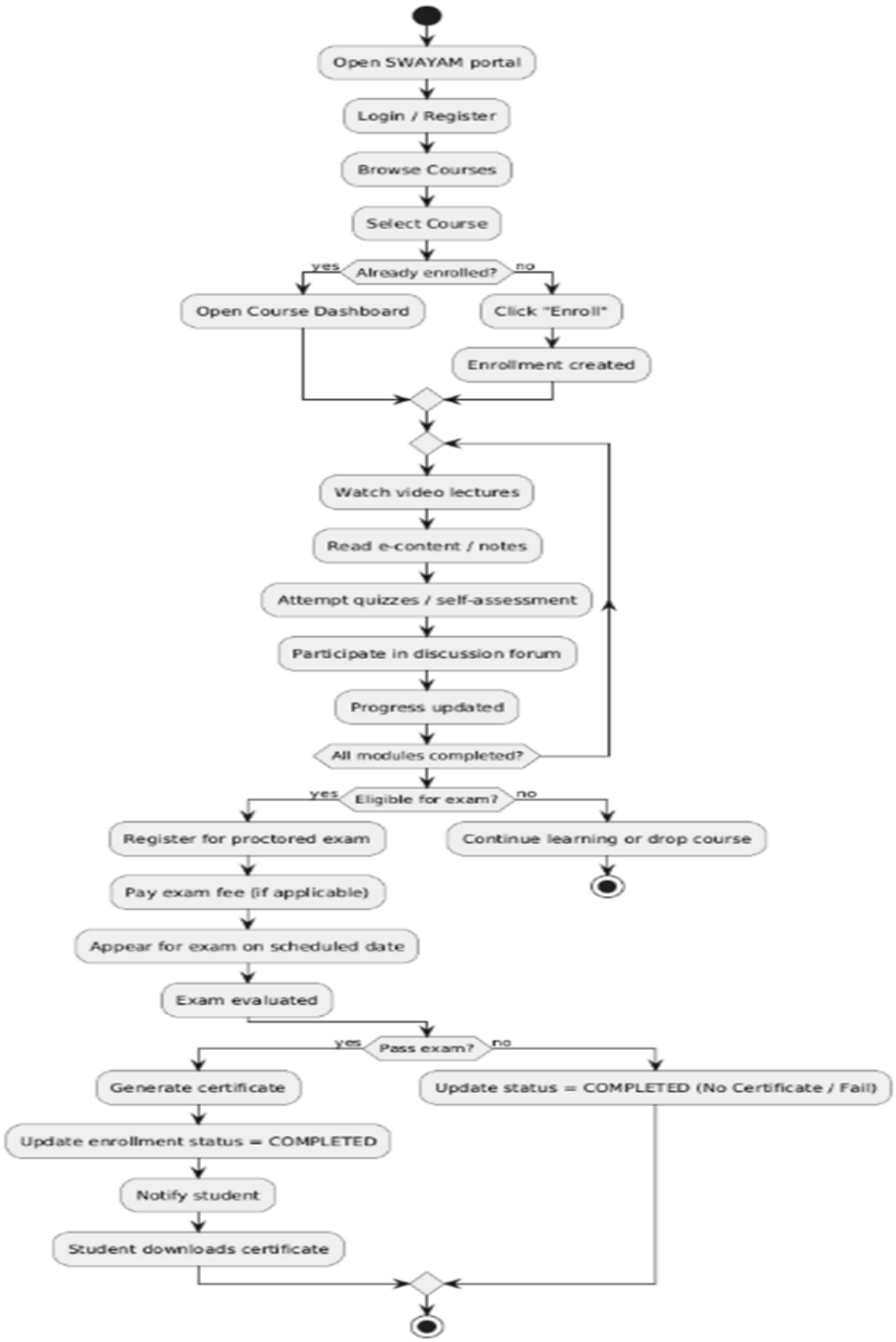


4.2 Behavioral Modeling

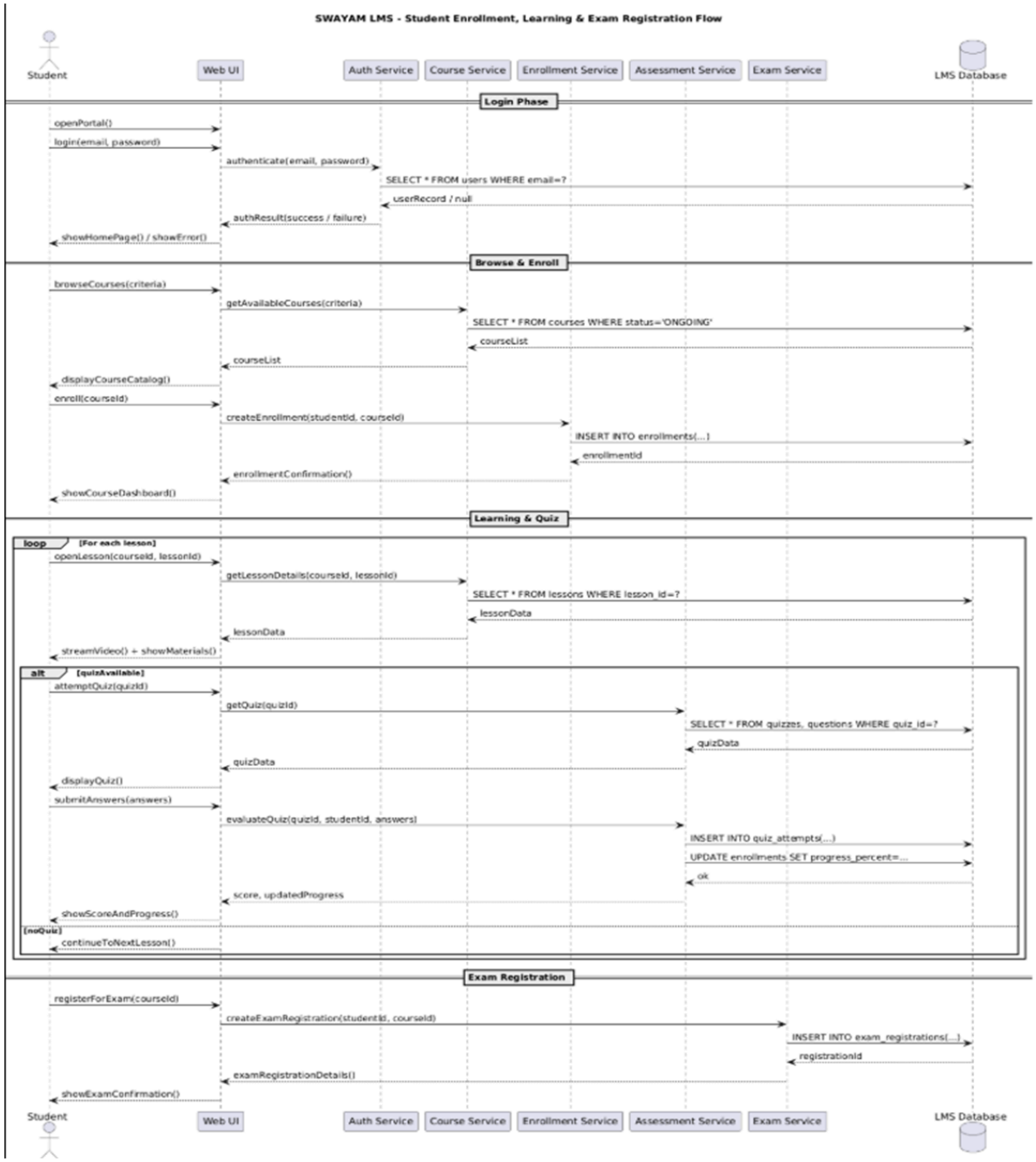
- **Use Case Diagram:** Illustrates interactions between *Students*, *Instructors*, and *Admin*.


SWAYAM LMS
Register for Proctored Exam
«extend»
Download Certificate
Browse Courses
Access Video & Reading Content
«include»
Enroll in Course
«include»
«include»
«include»
Attempt Quizzes
(Self Assessment)
Participate in
Discussion Forum
«include»
View Progress
Register / Login
Create / Manage Courses
Upload Videos & Materials
Create Quizzes
& Assignments
Moderate Forums
View Course Analytics
Configure Exam Schedule
Approve Courses
Manage Users
Manage Categories
& Coordinators
Student
Instructor
Admin

- **Activity Diagram:** Depicts the flow of course enrollment and learning activities.

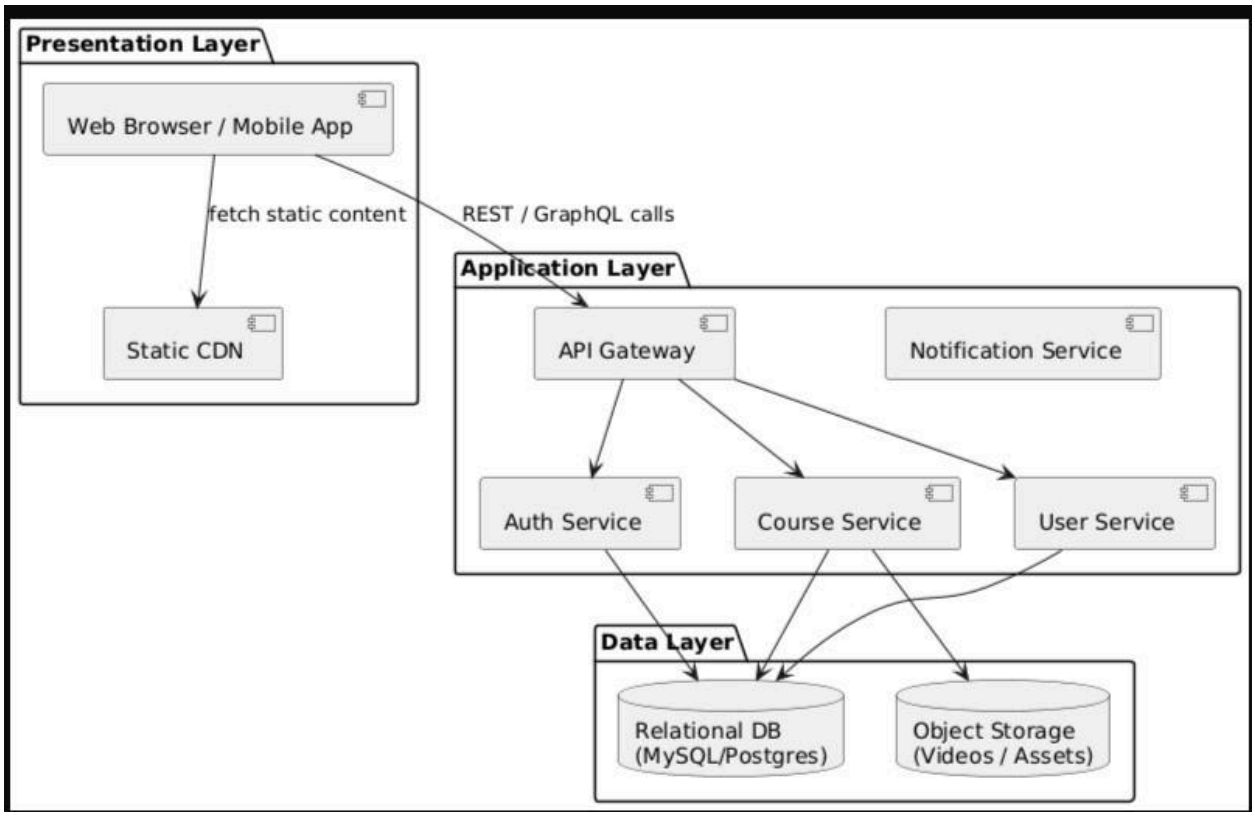


Sequence Diagram: Shows the sequence of operations during enrollment and assessment.



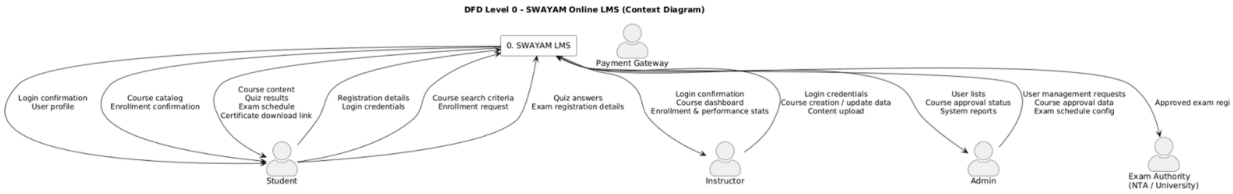
4.3 Architecture Modeling

- **3-Tier Architecture Diagram:** Separates Presentation, Business Logic, and Database layers for clarity and scalability.

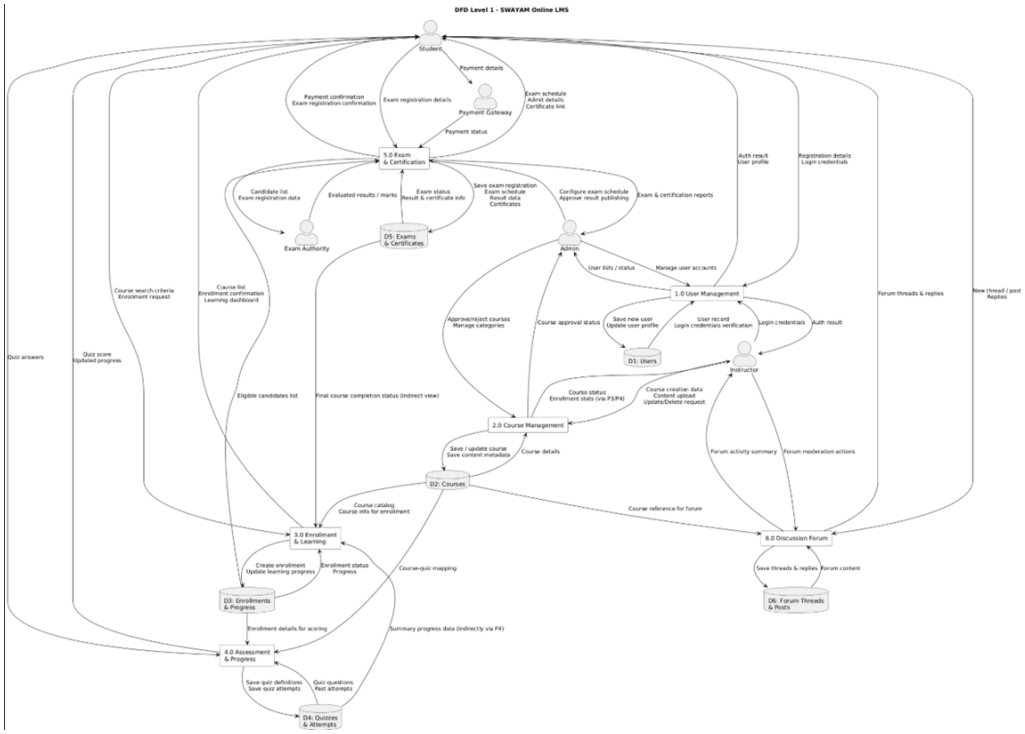


- **Data Flow Diagram (Level 0 & 1):** Shows flow of data between user, system, and database.

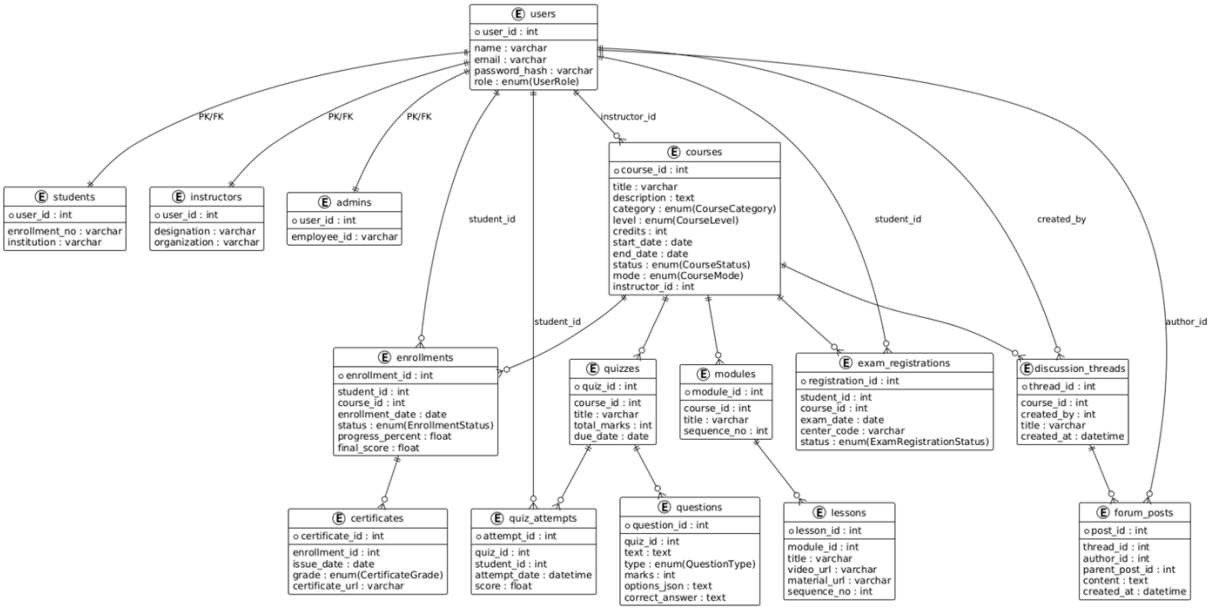
Level 0 :



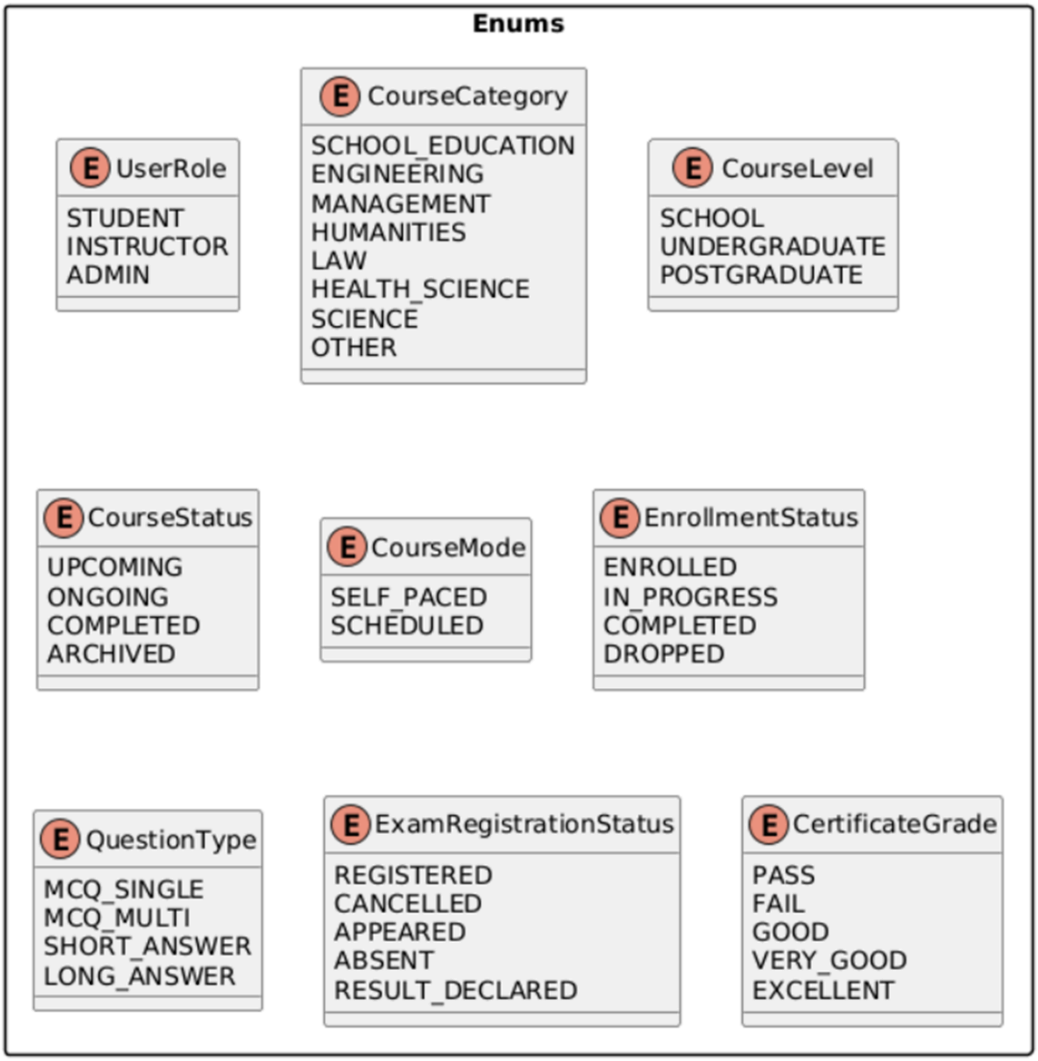
Level 1 :



- **Database Schema:** A database schema defines the structured blueprint of all tables, fields, and relationships used to store and manage data within the system.



- **ENUM Lookup Table :** An enum lookup table stores predefined, fixed categorical values used across the system to ensure consistency, validation, and standardized data representation.



---

5. Other Requirements

- **Reliability:** The system should recover from failures within 5 minutes.
- **Portability:** Should run on Windows, Linux, and mobile browsers.
- **Maintainability:** System code should follow object-oriented design principles for easier debugging and modification.
- **Testing:** Includes unit testing, integration testing, and user acceptance testing.

---

6. Appendices

6.1 UML Diagrams (References)

1. Class Diagram
2. Use Case Diagram
3. Sequence Diagram
4. Activity Diagram
5. Architecture Diagram
6. Data Flow Diagram
7. Testing Flow Diagram
8. Database schema
9. Enum lookup table