Software Requirements Specification (SRS) for JAMC Project

1. Introduction

1.1 Purpose

The purpose of this document is to define the technical requirements for the development of the JAMC platform. This document outlines the functionalities, constraints, and design considerations for the system to ensure it meets stakeholder and business goals as outlined in the BRD.

1.2 Scope

JAMC is an educational platform designed for high school students and teachers in Vietnam. It provides tools for managing classes, facilitating interactive Q&A sessions, and delivering additional learning resources through monetizable Modules. The platform aims to enhance teacher efficiency and improve student engagement through a personalized and flexible learning experience.

1.3 Definitions, Acronyms, and Abbreviations

- **OAuth:** Open Authorization protocol for secure third-party authentication.
- **Module:** A collection of educational content, including videos, PDFs, and quizzes, created by teachers.
- **Q&A System:** A structured feature for students and teachers to collaborate on course-related questions and answers.
- **GDPR:** General Data Protection Regulation for data privacy compliance.
- **BRD:** Business Requirements Document.
- **SRS:** Software Requirements Specification.

1.4 References

- Business Requirements Document (BRD) for JAMC Project.
- Educational technology guidelines and standards in Vietnam.
- Personal Data Protection Law

2. Overall Description

2.1 Product Perspective

JAMC integrates the capabilities of a Learning Management System and a Q&A platform. It supports class management, self-paced learning through Modules, and interactive student-teacher collaboration.

2.2 Product Features

The platform includes:

- Role-based access for students, teachers, and administrators.

- Tools for creating and managing classes.
- An interactive Q&A system for course-related discussions.
- Module creation and monetization capabilities for teachers.
- Dashboards for tracking progress and analytics.

2.3 User Classes and Characteristics

- **Teachers:** Digitally literate educators managing classes and student interactions.
- **Students:** High school students with varying levels of familiarity with online learning tools.
- **Administrators:** Platform managers responsible for moderating content and managing users.

2.4 Constraints

- The platform must adhere to Vietnamese data privacy laws and GDPR.
- Development must prioritize essential features; advanced features are reserved for future iterations.

3. Functional Requirements

3.1 User Registration and Onboarding

- Users can register as students, teachers, or administrators.
- Supports OAuth and email-based authentication.
- Role-specific onboarding with tailored user interfaces.

3.2 Class Management (Teacher Role)

- Teachers can create, edit, and delete classes.
- Tools for scheduling and tracking student attendance and progress.

3.3 Module Management

- Teachers can create Modules with videos, PDFs, quizzes, and assignments.
- Monetization options for Modules with configurable pricing.

3.4 Q&A System

- Students can post questions tagged with relevant topics or subjects.
- Teachers and students can answer questions, with upvoting/downvoting for quality content.
- Teachers can select the best answer for guidance.

3.5 Progress Tracking

- Students view course completion progress and badges.
- Teachers monitor individual and class-wide progress through dashboards.

4. Non-Functional Requirements

4.1 Performance

- The platform should support 1,000 concurrent users with minimal latency (<500ms response time).

4.2 Security

- Implement TLS encryption for all data transmissions.
- Role-based access control for user permissions.

4.3 Usability

- Ensure the interface is intuitive and responsive across devices.
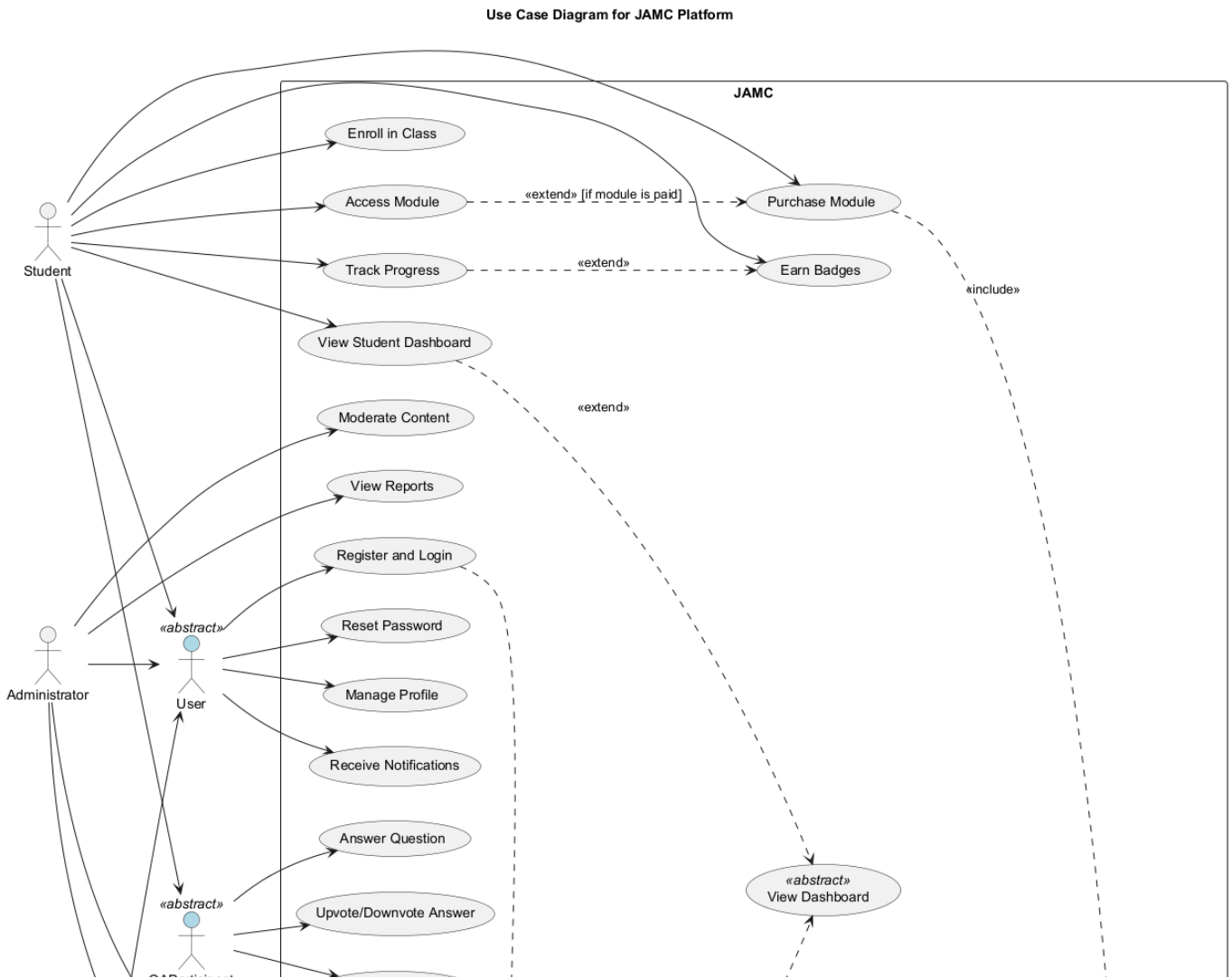- Support Vietnamese and English languages.

4.4 Reliability

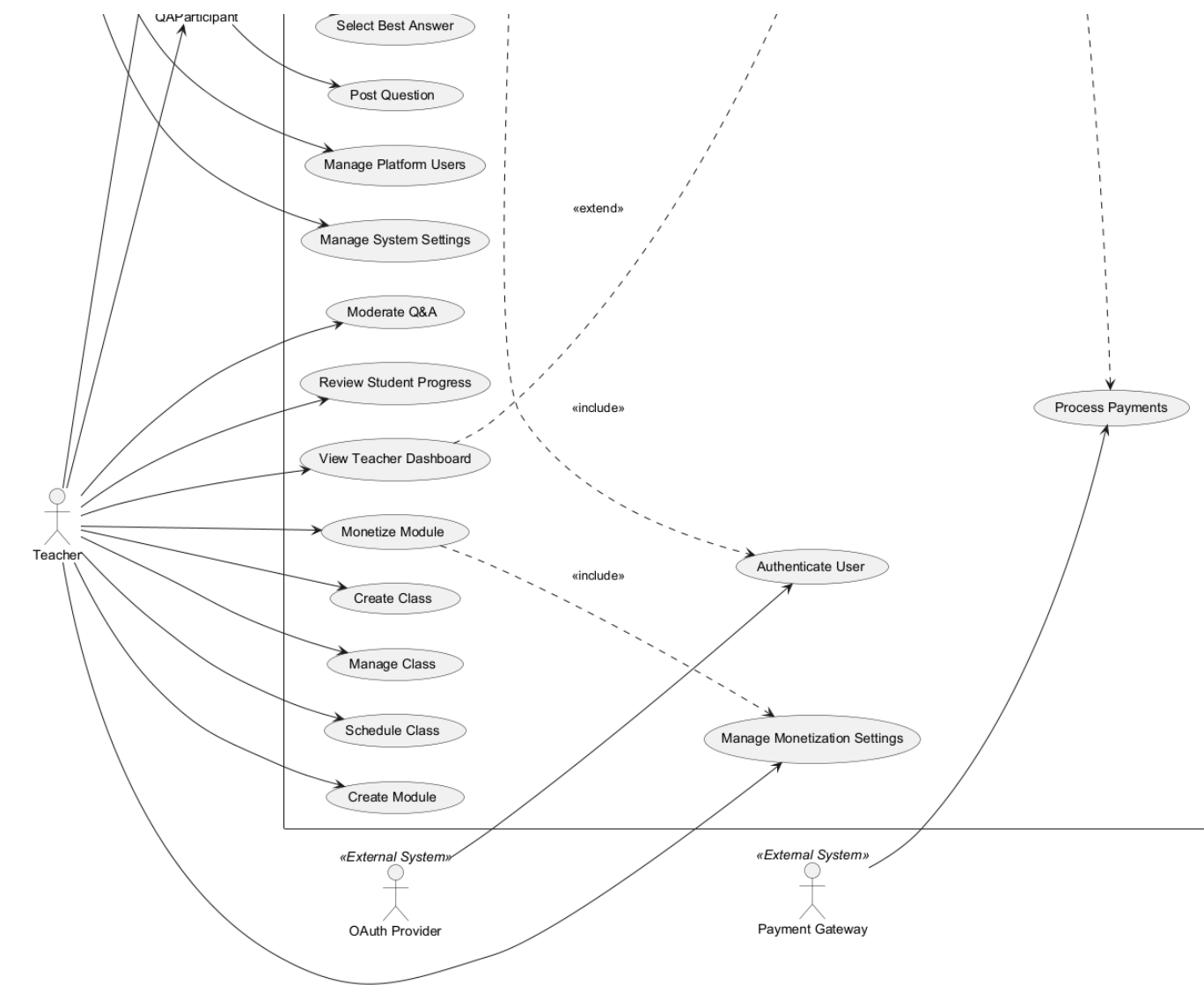- Target system has high uptime, especially during peak school hours.

4.5 Scalability

- The system must scale to support 10,000 users within one year.

5. System Models and Diagrams

5.1 Use Case Diagram

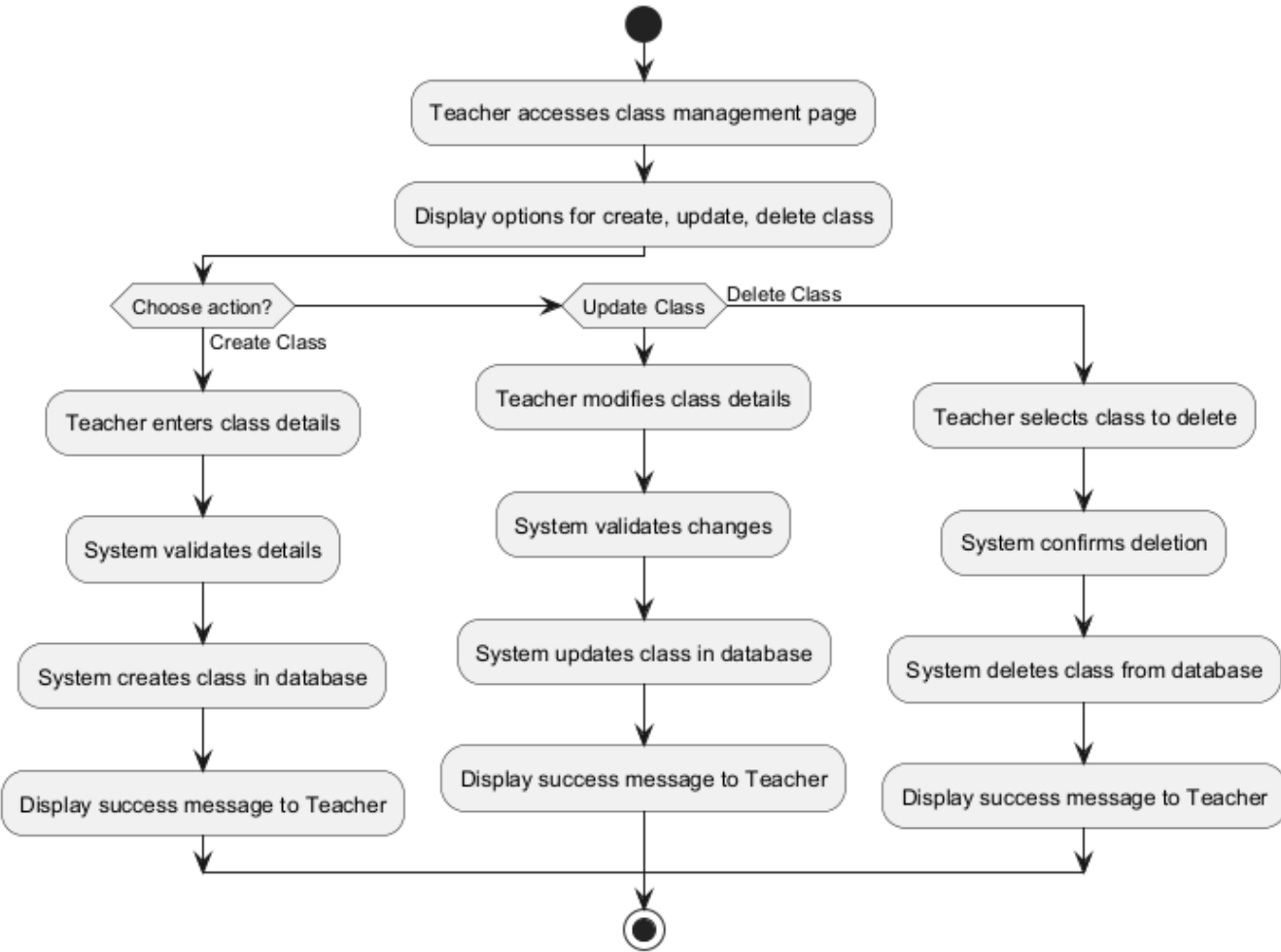Use Case Diagram for JAMC Platform

This diagram illustrates the key actors (Student, Teacher, Administrator) and their interactions with the JAMC platform, including core functionalities like class enrollment, module management, and Q&A participation.
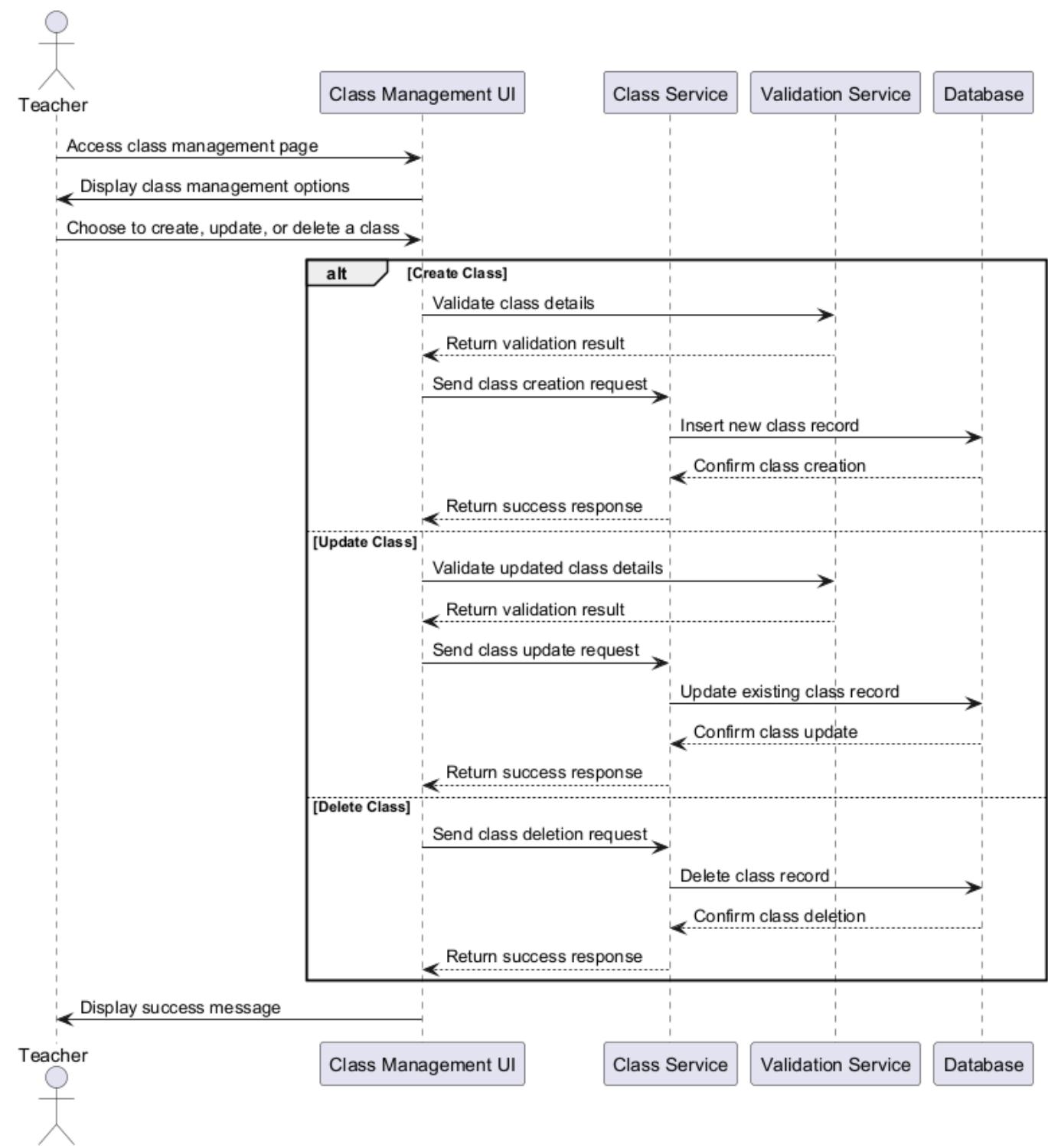
5.2 Class Management Diagrams

5.2.1 Class Management Activity Diagram

This diagram illustrates the flow of activities when teachers manage their classes, including creation, updates, and deletion of classes.
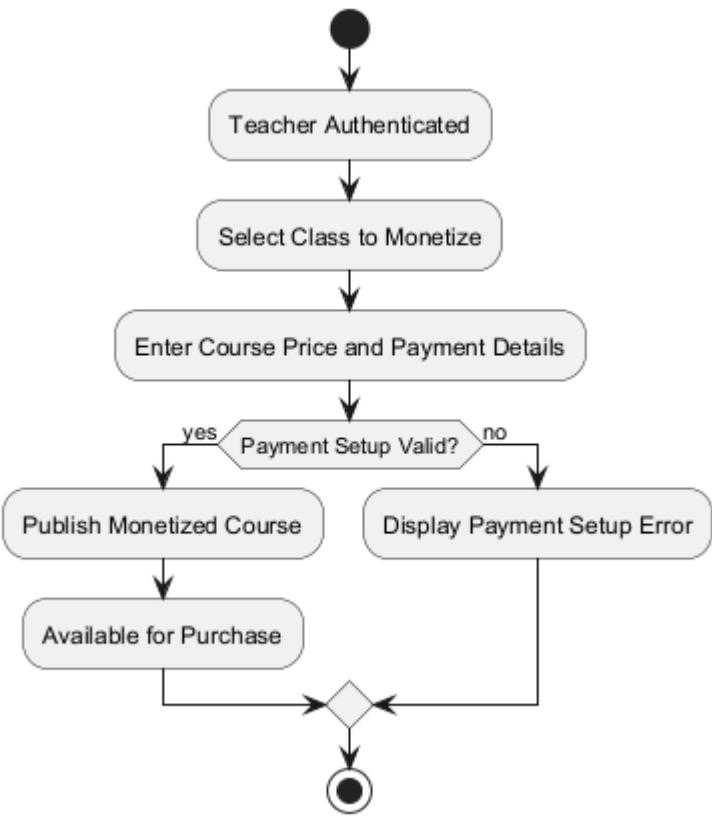
5.2.2 Class Management Sequence Diagram

This sequence diagram shows the interactions between the Teacher, UI components, and backend services during class management operations.

5.3 Course Monetization Diagrams
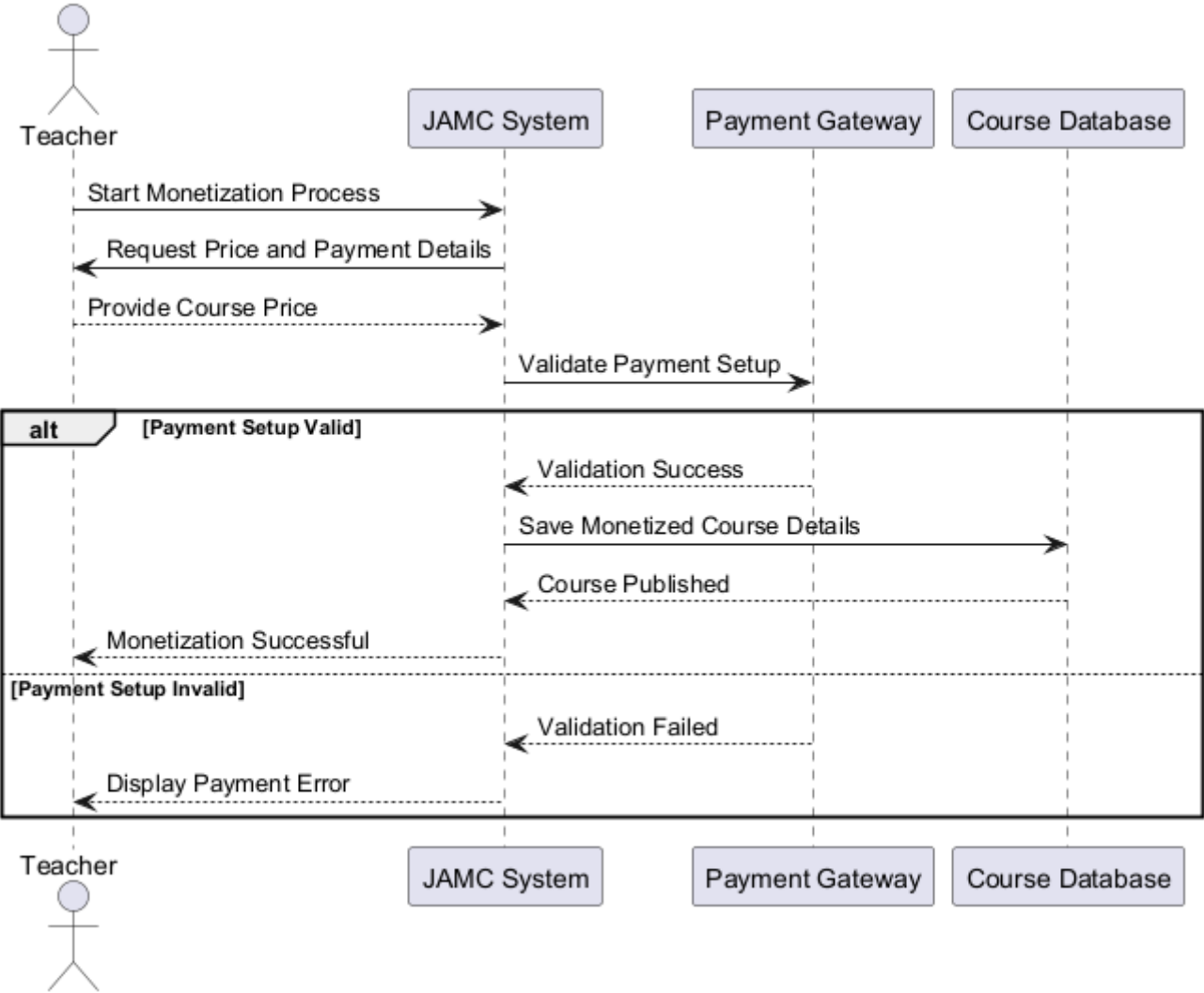
5.3.1 Course Monetization Activity Diagram

This diagram shows the process flow for teachers monetizing their courses.

5.3.2 Course Monetization Sequence Diagram

This sequence diagram illustrates the interactions between various system components during the course monetization process.

5.4 Administrative Control Diagrams

5.4.1 Admin Controls Activity Diagram
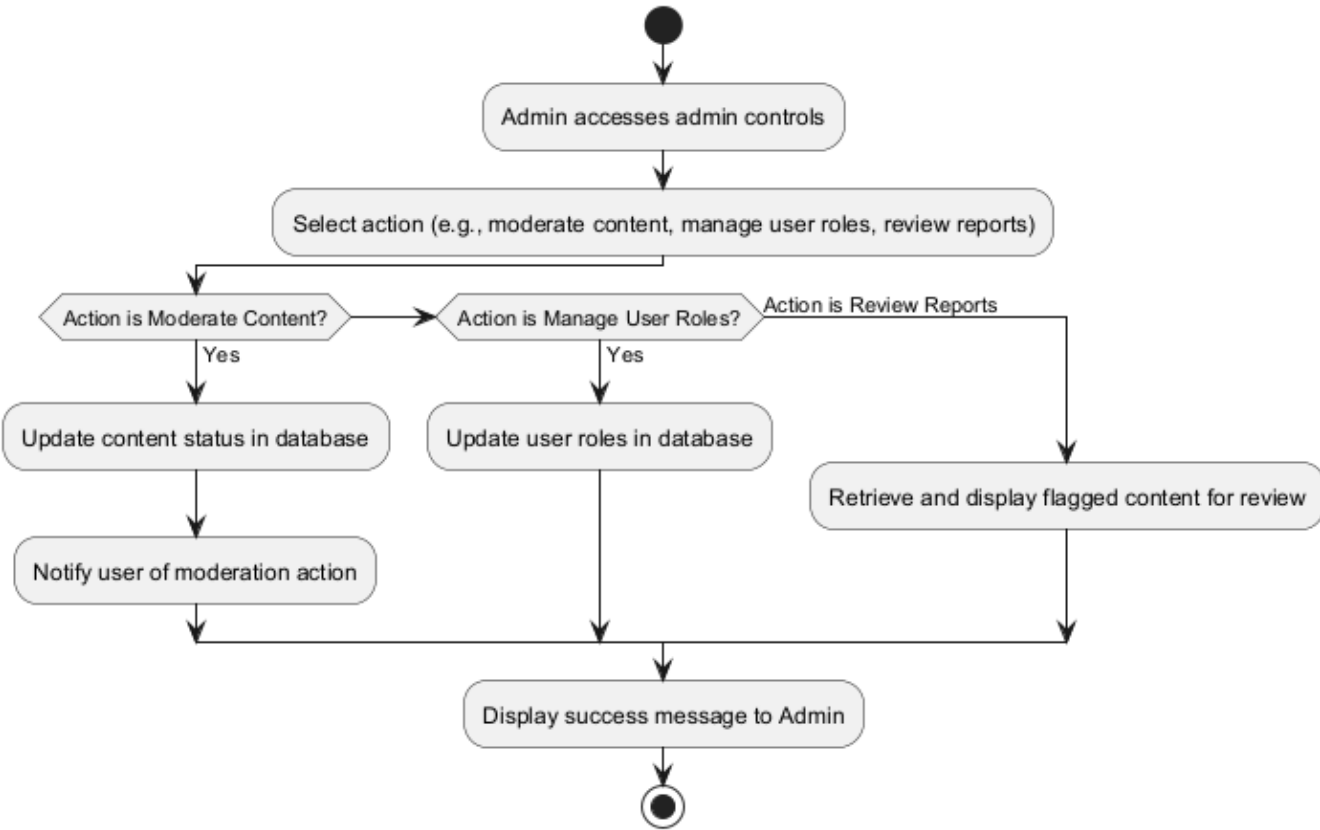
This diagram shows the flow of activities for administrative tasks such as content moderation and user management.

5.4.2 Admin Controls Sequence Diagram
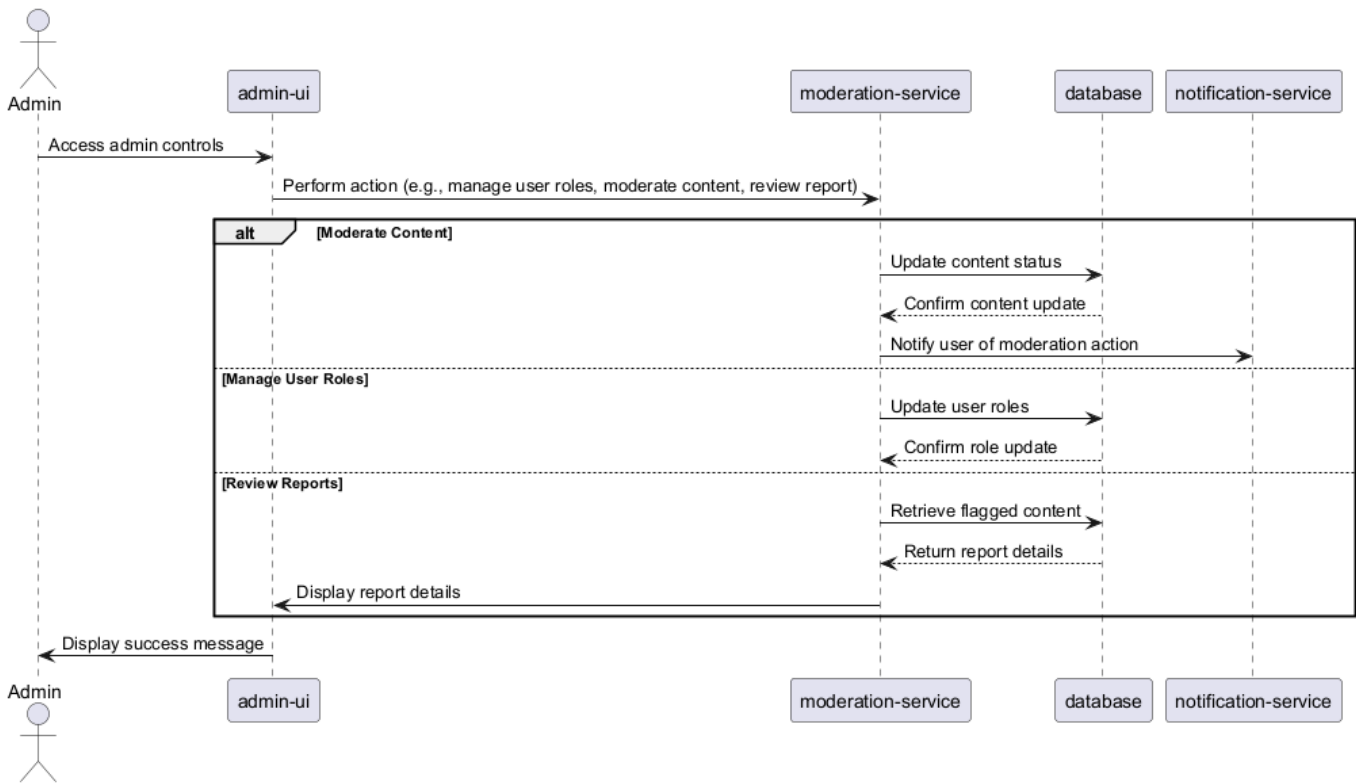
This sequence diagram details the interactions between administrators and the system during administrative operations.

5.5 Class Creation and Management Diagrams

5.5.1 Class Creation Activity Diagram

This diagram illustrates the detailed flow of activities during class creation and management.

5.5.2 Class Creation Sequence Diagram

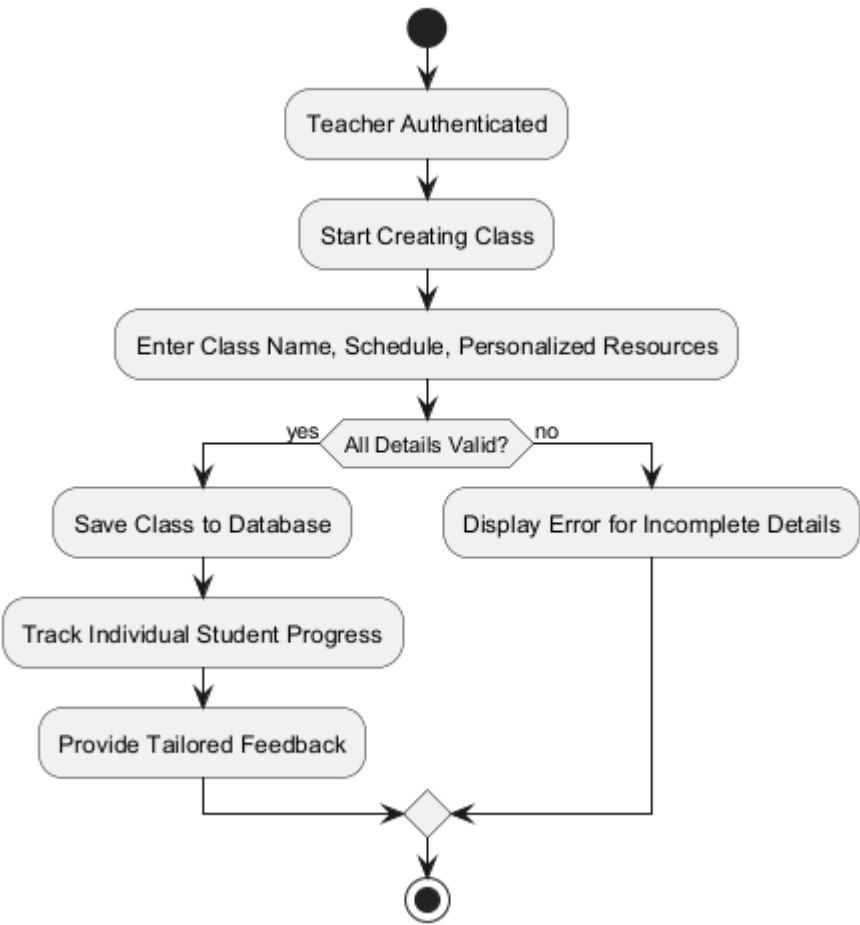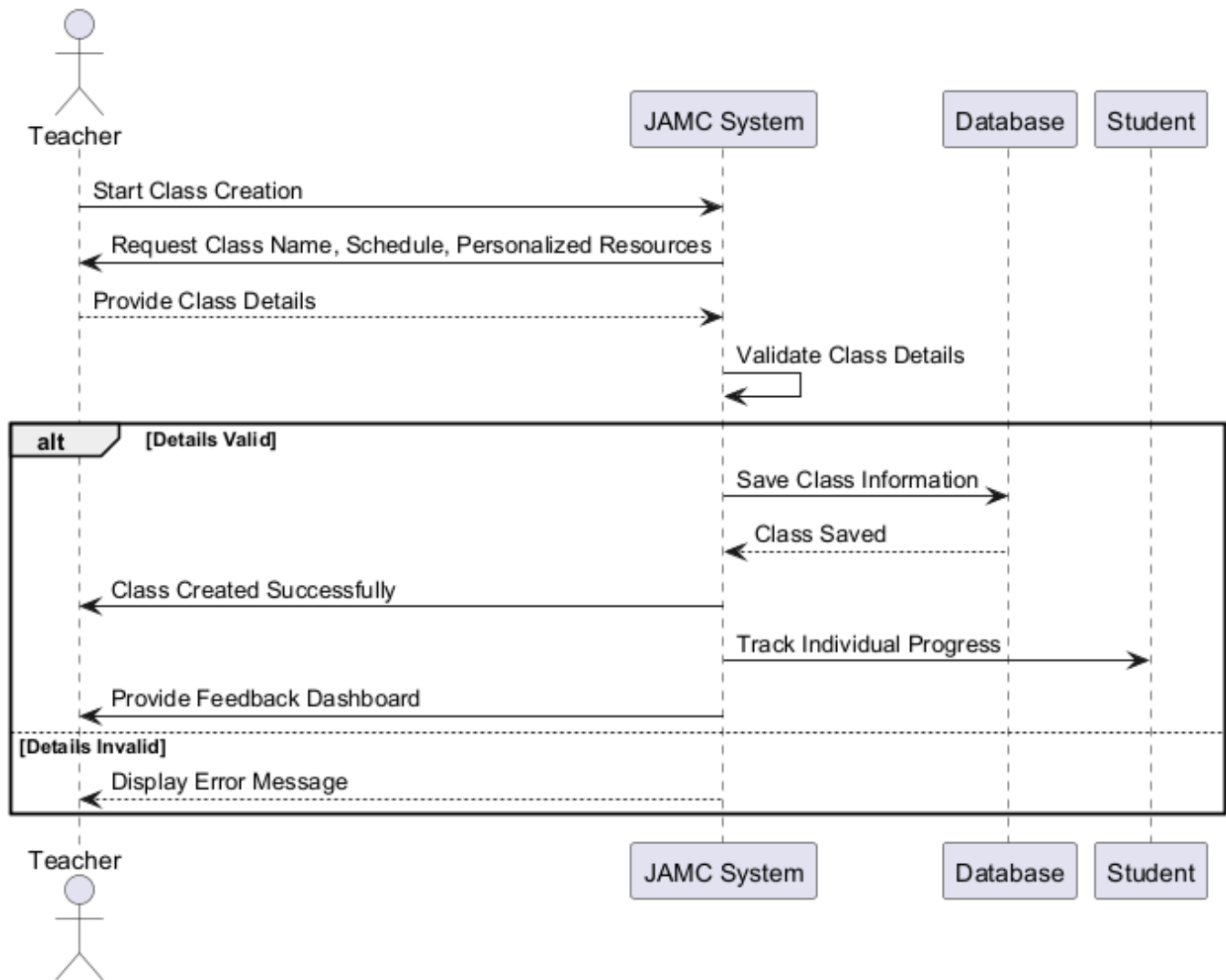This sequence diagram shows the interactions between teachers and the system during the class creation process.

---

6. Data Requirements

6.1 Overview

The JAMC platform requires a robust data management system to store and handle user information, course content, interactions, and transactions securely and efficiently. The data requirements are centered around ensuring data integrity, consistency, and compliance with data protection regulations, specifically Vietnamese data privacy laws and GDPR.

6.2 Data Entities and Attributes

Below are detailed descriptions of critical data entities, their attributes, and relationships essential for the platform's functionalities.

6.2.1 User

Description: Represents all users on the platform, including students, teachers, and administrators.

Attributes:

- **id**: String (Unique identifier)
- **name**: String (Full name of the user)
- **email**: String (Unique email address)
- **emailVerified**: DateTime? (Timestamp of email verification)
- **image**: String? (URL to the user's profile image)
- **role**: Role (User role: ONBOARDING, STUDENT, TEACHER, ADMIN)
- **password**: String? (Hashed password)
- **createdAt**: DateTime (Account creation timestamp)
- **updatedAt**: DateTime (Timestamp of the last update)

Relations:

- **accounts**: Authentication accounts (OAuth, etc.)
- **sessions**: Active user sessions
- **studentProfile**: Student-specific profile (if role is STUDENT)
- **teacherProfile**: Teacher-specific profile (if role is TEACHER)
- **questions**: Questions posted by the user
- **answers**: Answers provided by the user
- **questionVotes**: Votes cast on questions
- **answerVotes**: Votes cast on answers
- **ratings**: Course ratings given by the user

6.2.2 StudentProfile

Description: Contains student-specific information and relations.

Attributes:

- **id**: String (Unique identifier)
- **userId**: String (References User.id)
- **createdAt**: DateTime (Profile creation timestamp)
- **updatedAt**: DateTime (Timestamp of the last update)

Relations:

- **enrollments**: Courses the student is enrolled in
- **progresses**: Progress records in courses and lessons
- **groupMembers**: Group memberships for collaborative learning
- **certificates**: Achievements and badges earned

6.2.3 TeacherProfile

Description: Contains teacher-specific information and relations.

Attributes:

- **id**: String (Unique identifier)
- **userId**: String (References User.id)

- **createdAt**: `DateTime` (Profile creation timestamp)
- **updatedAt**: `DateTime` (Timestamp of the last update)
- **verificationDocuments**: `String[]` (Documents for identity verification)

Relations:

- **courses**: Courses created by the teacher
- **revenues**: Revenue records from course monetization

6.2.4 Course

Description: Represents educational content created by teachers, which can be monetized.

Attributes:

- **id**: `String` (Unique identifier)
- **title**: `String` (Course title)
- **description**: `String` (Course description)
- **rating**: `Float` (Average rating, default 0)
- **difficulty**: `DifficultyLevel?` (EASY, MEDIUM, HARD)
- **subject**: `String?` (Subject area)
- **createdAt**: `DateTime` (Creation timestamp)
- **updatedAt**: `DateTime` (Last update timestamp)

Relations:

- **teacher**: Creator (References TeacherProfile.id)
- **lessons**: List of lessons included
- **enrollments**: Student enrollments
- **progresses**: Student progress records
- **ratings**: Reviews and ratings
- **certificates**: Certificates awarded upon completion
- **revenues**: Monetization records
- **resources**: Additional course materials

6.2.5 Lesson

Description: Individual units of learning within a course.

Attributes:

- **id**: `String` (Unique identifier)
- **title**: `String` (Lesson title)
- **content**: `String` (Lesson content)
- **videoUrl**: `String?` (URL to lesson video)
- **order**: `Int` (Position in course sequence)
- **createdAt**: `DateTime` (Creation timestamp)
- **updatedAt**: `DateTime` (Last update timestamp)

Relations:

- **course**: Associated course (References Course.id)
- **progresses**: Progress records for students
- **questions**: Related Q&A entries

6.2.6 Enrollment

Description: Records of students enrolled in courses.

Attributes:

- **id**: `String` (Unique identifier)
- **accessType**: `AccessType` (FREE or PAID)
- **status**: `String` (Enrollment status: PENDING, ACTIVE, etc.)
- **createdAt**: `DateTime` (Enrollment timestamp)
- **updatedAt**: `DateTime` (Last update timestamp)

Relations:

- **student**: Enrolled student (References StudentProfile.id)
- **course**: Enrolled course (References Course.id)

Constraints:

- **Unique Enrollment**: A student cannot have multiple enrollments in the same course.

6.2.7 Progress

Description: Tracks student progress within courses and lessons.

Attributes:

- **id**: `String` (Unique identifier)
- **lessonCompletion**: `Boolean` (Completion status)
- **quizResults**: `Json` (Results of quizzes)
- **flaggedStatus**: `FlaggedStatus` (SLOW_PROGRESS, FAST_PROGRESS, NORMAL)
- **createdAt**: `DateTime` (Record creation timestamp)
- **updatedAt**: `DateTime` (Last update timestamp)

Relations:

- **student**: Student making progress (References StudentProfile.id)
- **course**: Associated course (References Course.id)
- **lesson**: Associated lesson (References Lesson.id)

6.2.8 Question

Description: Represents questions posted in the Q&A system.

Attributes:

- **id**: String (Unique identifier)
- **title**: String (Question title)
- **problemDetails**: String (Detailed description)
- **attemptedSolution**: String (Attempted answer by the asker)
- **priorityScore**: Float (Priority level, default 0)
- **visibility**: Visibility (PUBLIC or PRIVATE)
- **difficulty**: DifficultyLevel? (EASY, MEDIUM, HARD)
- **createdAt**: DateTime (Creation timestamp)
- **updatedAt**: DateTime (Last update timestamp)

Relations:

- **author**: User who posted (References User.id)
- **lesson**: Associated lesson (if any)
- **votes**: Votes on the question
- **tags**: Tags associated with the question
- **comments**: Comments on the question
- **answers**: Provided answers
- **bestAnswer**: Selected best answer (References Answer.id)
- **files**: Attached files (images, PDFs)

6.2.9 Answer

Description: Responses to questions in the Q&A system.

Attributes:

- **id**: String (Unique identifier)
- **content**: String (Answer content)
- **createdAt**: DateTime (Creation timestamp)
- **updatedAt**: DateTime (Last update timestamp)

Relations:

- **author**: User who answered (References User.id)
- **question**: Associated question (References Question.id)
- **votes**: Votes on the answer
- **comments**: Comments on the answer
- **questionBestAnswer**: If selected as the best answer

6.2.10 Vote (QuestionVote & AnswerVote)

Description: Records votes (upvote/downvote) on questions and answers.

Attributes:

- **id**: String (Unique identifier)

- **voteType**: VoteType (UPVOTE or DOWNVOTE)
- **createdAt**: DateTime (Vote timestamp)

Relations:

- **user**: Voter (References User.id)
- **question**: For QuestionVote (References Question.id)
- **answer**: For AnswerVote (References Answer.id)

Constraints:

- **Unique Vote**: A user can vote only once per question or answer.

6.2.11 Comment (QuestionComment & AnswerComment)

Description: Comments on questions and answers.

Attributes:

- **id**: String (Unique identifier)
- **content**: String (Comment text)
- **createdAt**: DateTime (Creation timestamp)
- **updatedAt**: DateTime (Last update timestamp)

Relations:

- **author**: Commenter (References User.id)
- **question**: For QuestionComment (References Question.id)
- **answer**: For AnswerComment (References Answer.id)

6.2.12 Tag

Description: Categorizes questions for easier navigation.

Attributes:

- **id**: String (Unique identifier)
- **name**: String (Tag name, unique)

Relations:

- **questionTags**: Association with questions

6.2.13 Certificate

Description: Represents achievements awarded to students.

Attributes:

- **id**: String (Unique identifier)

- **achievement**: `AchievementType` (CERTIFICATE or BADGE)
- **dateIssued**: `DateTime` (Issuance date)

Relations:

- **student**: Recipient (References StudentProfile.id)
- **course**: Associated course (References Course.id)

6.2.14 Revenue

Description: Tracks earnings from course monetization.

Attributes:

- **id**: `String` (Unique identifier)
- **earnedAmount**: `Float` (Total earnings)
- **studentEnrollment**: `Int` (Number of enrolled students)
- **createdAt**: `DateTime` (Record creation timestamp)
- **updatedAt**: `DateTime` (Last update timestamp)

Relations:

- **teacher**: Earning teacher (References TeacherProfile.id)
- **course**: Associated course (References Course.id)

6.2.15 Resource

Description: Additional materials for courses.

Attributes:

- **id**: `String` (Unique identifier)
- **title**: `String` (Resource title)
- **url**: `String` (Link to the resource)
- **type**: `String` (Type of resource, e.g., PDF, VIDEO)
- **description**: `String?` (Optional description)
- **order**: `Int` (Ordering in course)
- **createdAt**: `DateTime` (Creation timestamp)
- **updatedAt**: `DateTime` (Last update timestamp)

Relations:

- **course**: Associated course (References Course.id)

6.3 Data Relationships

The data entities are interconnected to support platform functionalities:

- **User** can be a **Student** or **Teacher**, identified by their **role**.

- **StudentProfile** and **TeacherProfile** are linked to **User** by **userId**.
- **Course** is created by a **TeacherProfile** and includes multiple **Lessons**.
- **Enrollment** links **StudentProfile** and **Course**.
- **Progress** tracks **StudentProfile**'s advancement in **Course** and **Lesson**.
- **Question** and **Answer** are associated with **User**, **Lesson**, and **Tags**.
- **Vote** entities record user interactions with **Question** and **Answer**.
- **Certificate** records achievements for **StudentProfile** in **Course**.
- **Revenue** tracks earnings for **TeacherProfile** from **Course**.

6.4 Data Constraints

- **Uniqueness Constraints:**

  - **User.email** must be unique.
  - **Tag.name** must be unique.
  - **Enrollment** must be unique per **studentId** and **courseId**.
  - **Rating** must be unique per **userId** and **courseId**.
  - **QuestionVote** and **AnswerVote** must be unique per user and question/answer.

- **Referential Integrity:**

  - All foreign keys must reference existing records.
  - Deletion of records must consider cascading effects (e.g., deleting a **Course** should cascade to **Lesson**).

- **Validation Rules:**

  - **Password** must meet security standards (e.g., minimum length, complexity).
  - **Rating.rating** must be within an acceptable range (e.g., 1 to 5).
  - **Content fields** (e.g., **Question.title**, **Answer.content**) cannot be empty.

- **Access Control:**

  - Data access must be restricted based on user roles.
  - Private data (e.g., **Question** with **visibility** set to PRIVATE) must only be accessible to authorized users.

6.5 Data Privacy and Security

- **Compliance:**

  - Adhere to Vietnamese data privacy laws and GDPR.
  - Collect and process only necessary personal data.

- **Encryption:**

  - Store sensitive data (e.g., passwords, tokens) using strong encryption.
  - Use TLS/SSL for all data transmission.

- **Data Retention and Deletion:**

  - Define data retention policies in compliance with regulations.

- Implement mechanisms for data deletion upon user request.

- **Audit Trails:**

    - Maintain logs of critical operations (e.g., login attempts, data changes).
    - Ensure logs are secure and tamper-proof.

- **User Consent:**

    - Obtain explicit consent for data collection and processing.
    - Provide clear privacy policies and terms of service.

- **Data Backup and Recovery:**

    - Regularly backup data to prevent loss.
    - Implement disaster recovery plans.

6.6 Scalability Considerations

- **Database Optimization:**

    - Index frequently queried fields (e.g., **User.email**, **Course.title**).
    - Use efficient query strategies to handle large datasets.

- **Data Partitioning:**

    - Consider sharding or partitioning strategies for scaling.

- **Caching Mechanisms:**

    - Implement caching for read-heavy operations (e.g., fetching popular courses).

6.7 Data Migration and Integration

- **Third-Party Integration:**

    - Data exchange with OAuth providers for authentication.
    - Integration with payment gateways for monetization.

- **Migration Strategies:**

    - Plan for data migration in case of system upgrades or changes.

6.8 Future Data Requirements

- **Advanced Analytics:**

    - Collect data for future implementation of analytics and reporting features.
    - Ensure data collected now can support advanced features planned for later iterations.

- **AI and Personalization:**

    - Store user interaction data to enable personalized recommendations in the future.
    - Ensure data collected complies with privacy regulations.

7. Constraints and Dependencies

7.1 Constraints

- The system must comply with Vietnamese and international data privacy laws.
- Limited to core functionalities in the initial release; advanced analytics are deferred.

7.2 Dependencies

- Integration with third-party OAuth providers for authentication.
- Hosting on a cloud platform for scalability and uptime guarantees.

8. Validation and Acceptance Criteria

- Functional tests for core features (e.g., Q&A posting, class creation).
- Performance testing to ensure the platform can handle concurrent users.
- Feedback collection from pilot schools for usability validation.