UB - ONLINE LEARNING

(SRS REPORT)

1. Introduction

1.1. Purpose

UB - Online Learning is developed to simplify and enhance the management of online educational content, learner engagement, and course delivery. It serves as an efficient and scalable solution for administrators to manage course listings, monitor student progress, and organize interactive learning activities. The system ensures a seamless experience for learners by providing features such as account registration, personalized course dashboards, progress tracking, and secure certification management. This platform will streamline the workflow for educators and offer a user-friendly interface for students, ensuring an accessible, engaging, and effective online learning experience.

1.2. Scope

The scope of the UB - Online Learning system includes:

- Manage Courses: Allows administrators and instructors to create, edit, and manage course listings, including video lectures, reading materials, assignments, quizzes, and course descriptions.

- User Account Management: Learners can register accounts, update personal profiles, and securely log in. The system ensures account security through password encryption and session management.

- Enrollment and Progress Tracking: Users can enroll in available courses, monitor their learning progress, complete assignments, and view course completion status and certifications.

- Interactive Learning Tools: The platform supports forums, discussion boards, and messaging systems to facilitate interaction between learners and instructors.

- Assessment and Certification: The system allows instructors to create assessments, quizzes, and final tests. Upon completion, learners can receive digital certificates that can be downloaded or shared.

- Payment Integration: Secure integration with third-party payment gateways for course enrollments, supporting transactions via credit cards, e-wallets, and other online payment methods.

- Reporting and Analytics: The system provides comprehensive reports on course performance, learner engagement, and enrollment statistics to support data-driven decision-making for administrators and educators.

1.3. Definition, Acronyms, Abbreviation

- **Admin:** The system user responsible for managing course listings, user accounts, enrollments, assessments, and platform settings. Admins are typically staff members or educators with higher-level access permissions within the system.

- **User (Student):** End-users who register, enroll in courses, access learning materials, and complete assessments on the platform.

- **Payment Gateway:** A third-party service integrated to securely process online payment transactions for paid course enrollments (e.g., credit card, e-wallet).

- **SRS: Software Requirements Specification** — A document that defines the system's functionality, requirements, and technical specifications.

- **LMS: Learning Management System** — A software application for the administration, documentation, tracking, reporting, and delivery of educational courses or training programs.

- **CPD: Continuing Professional Development** — Courses or activities that help learners maintain and enhance their professional skills (if applicable within the platform).

2. Overall Description

2.1. Product Perspective

The proposed UB - Online Learning System is designed to streamline and improve the management of digital courses, user engagement, and academic activities for both administrators and learners. This platform centralizes course creation, student enrollment, progress tracking, assessment management, and communication tools under a unified interface. Every action - whether a course update, assignment submission, or grade notification - is reflected in real-time across the system, ensuring synchronization and a smooth learning experience.

2.2. Software Requirement

- **Front end:**

  - ReactJS (Web version)

  - HTML, CSS, JavaScript

- **Back end:**
  - Node.js
  - MongoDB database

2.3. Hardware Requirement

Minimum (Client Side):

- Smartphone or PC with internet access
- Browser: Chrome, Firefox, Safari, or equivalent
- RAM: 2GB or more

Minimum (Server Side):

- Intel Core i5 processor or equivalent
- 8GB RAM
- 100GB SSD Storage
- Linux-based operating system (Ubuntu 20.04+)

2.4.1. Functional Requirement

REQ#	PRIORITY	DESCRIPTION	RATIONALE	USE CASE
FR001	Critical	Users can register accounts via email, log in securely, and update personal details (e.g., password, profile picture, contact information).	Essential for user authentication and personalization.	Register an account, Login, Manage Profile
FR002	Critical	Users can search/filter courses by keyword, category, or price, view detailed course information, enroll in courses, and process payments securely.	Enhances accessibility and allows users to make informed decisions before enrolling.	Search Courses, View Course Details, Enroll Course,Pay for Course
FR003	High	Admins can manage user accounts (suspend, delete, assign roles) and oversee courses (create/edit/delete	Ensures proper governance and platform management	Manage Courses, Review Reports

REQ#	PRIORITY	DESCRIPTION	RATIONALE	USE CASE
		courses, upload materials, analyze enrollment/feedback reports).		
FR004	High	Learners can submit ratings/reviews for completed courses, and instructors can view/respond to feedback and update course content.	Encourages engagement and quality improvement for courses.	Course Rating and Reviews, Review Course

2.4.2. Non-Functional Requirement

REQ#	PRIORITY	DESCRIPTION	RATIONALE	USE CASE
NFR001	Critical	The platform must handle 10,000+ concurrent users with <2s page load times and scale horizontally during peak enrollment periods.	Ensures smooth performance during high-traffic periods.	Register an account, Login, Search Courses, Enroll Course, Pay for Course
NFR002	Critical	Encrypt sensitive data (e.g., payment info, user credentials) and comply with GDPR or FERPA regulations.	Protects user privacy and secures transactions.	Pay for Course, UC-10: Manage Profile, Register an account, Login
NFR003	High	Ensure seamless functionality across browsers (Chrome, Safari, Firefox) and devices (desktop, mobile, tablet).	Enhances user accessibility and usability.	All use case
NFR004	High	Guarantee 99.9% uptime with redundancy and failover mechanisms.	Minimizes downtime and ensures availability.	Login, Enroll Course, Pay for Course
NFR005	Medium	Meet WCAG 2.1 standards for screen readers, keyboard navigation, and captioning for users with disabilities.	Ensures inclusivity for all users.	Search Courses, View Course Details, Review progress, Manage Notifications
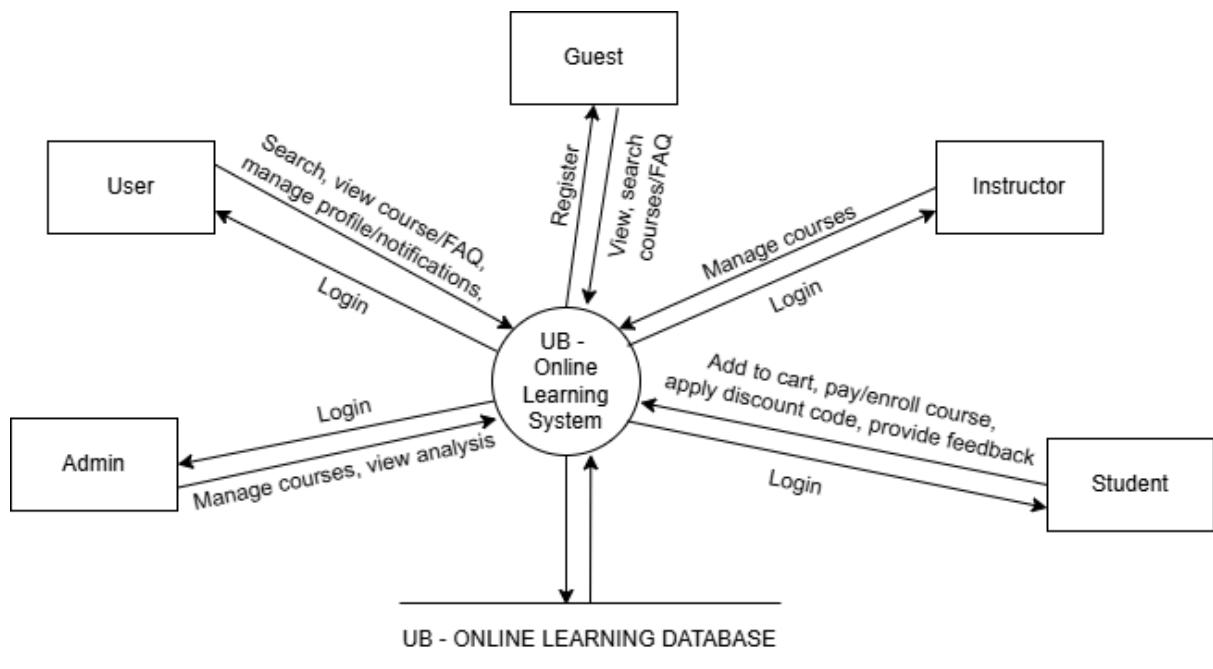
2.5. User Characteristics

-, User (Students): Users with varying levels of technological proficiency who participate in online courses, track their progress, and complete assessments.

-, Administrators: Staff members proficient in system management, responsible for managing users, overseeing course operations, monitoring reports, and ensuring platform functionality.

2.6. Constraints

- The system must comply with data privacy and security regulations such as GDPR and relevant online payment security standards (if payment is integrated).

- The system should support up to 10,000 concurrent users.

- Payment for courses (if applicable) must be processed through third-party online payment gateways.

2.7. Flow Diagram
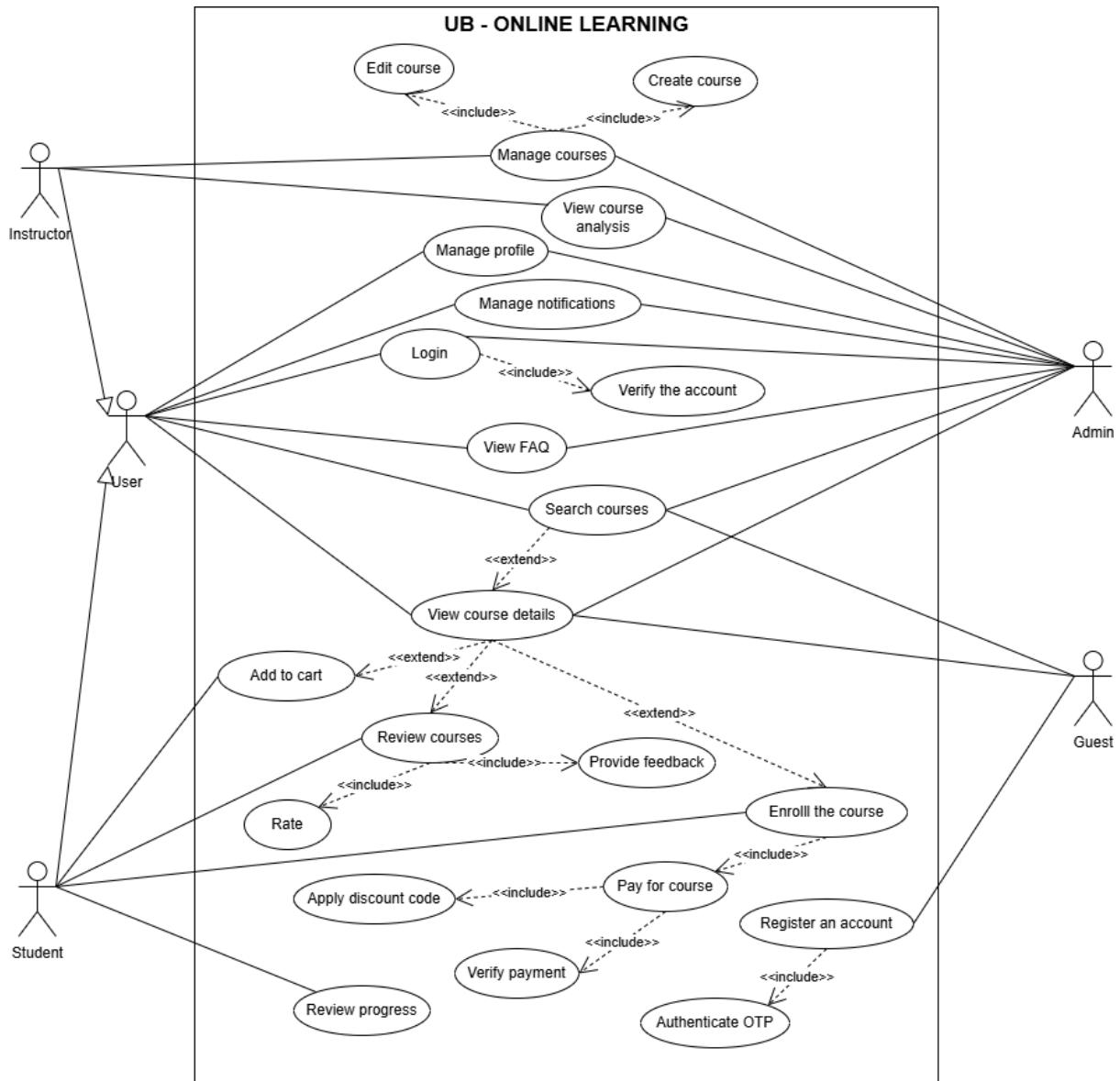
2.8. Use Case List

No	Use Case	Level
1	Register an account	**Must have**
2	Login	**Must have**
3	View Course Analysis	**Should Have**
4	Manage Courses	**Must have**
5	Search Courses	**Must Have**
6	View Course Details	**Must Have**
7	Enroll Course	**Must have**
8	Review Course	**Should Have**
9	Pay for Course	**Must have**
10	Manage Profile	**Must have**
11	Review progress	**Should Have**
12	Manage Notifications	**Should Have**

13	Add to Cart	**Must have**
14	Apply Discount Code	**Should Have**
15	View FAQ	**Should Have**
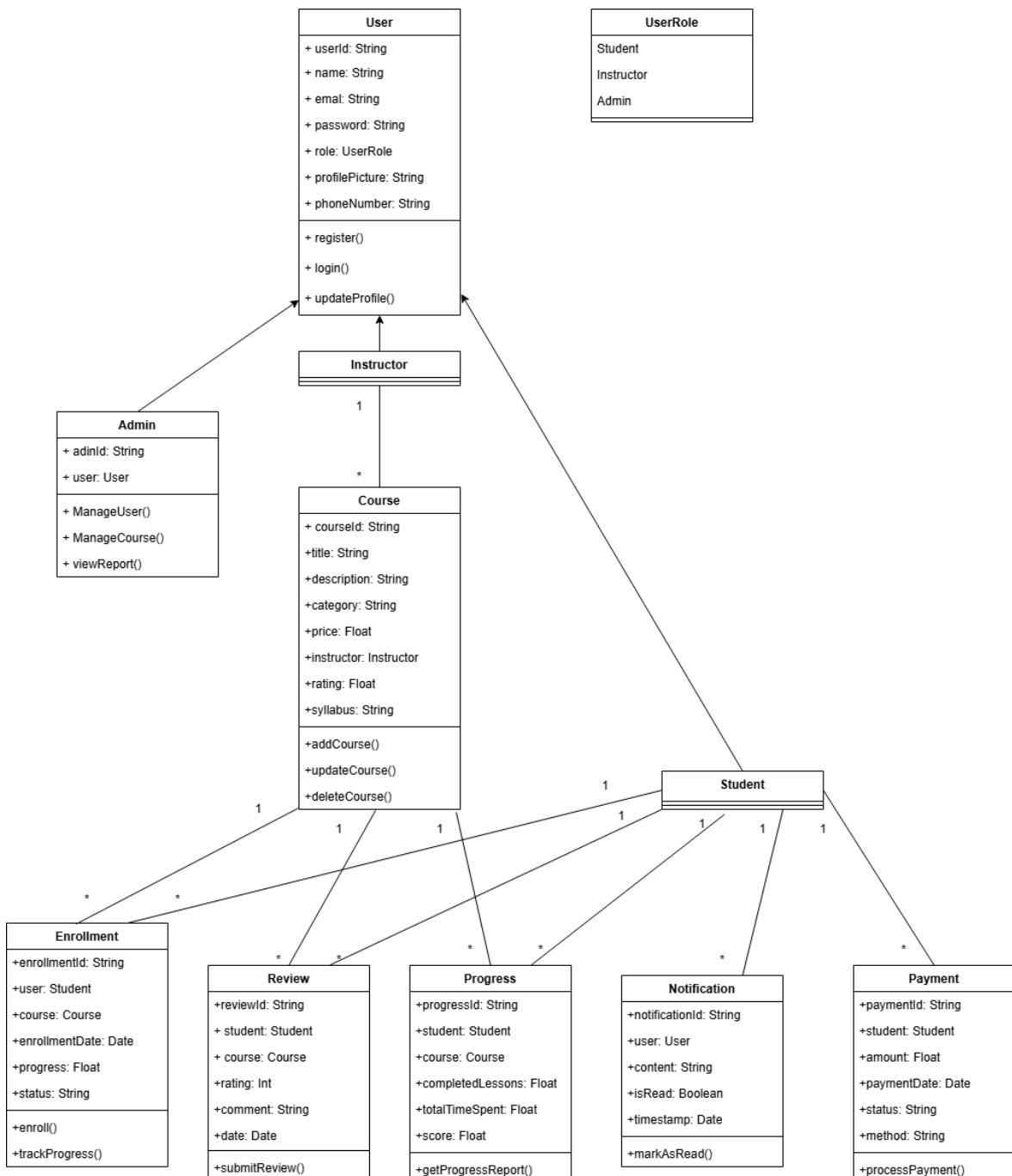
2.9. Application Modelling

2.9.1. Use Case Diagram

Flow chart 1: UB - Online Learning Use Case Diagram with 15 Use Cases

2.9.2. Class Diagram

User

| + userId: String |
| + name: String |
| + emal: String |
| + password: String |
| + role: UserRole |
| + profilePicture: String |
| + phoneNumber: String |
| + register() |
| + login() |
| + updateProfile() |

UserRole

| Student |
| Instructor |
| Admin |

Instructor

Admin

| + adinId: String |
| + user: User |
| + ManageUser() |
| + ManageCourse() |
| + viewReport() |

Course

| + courseId: String |
| +title: String |
| +description: String |
| +category: String |
| +price: Float |
| +instructor: Instructor |
| +rating: Float |
| +syllabus: String |
| +addCourse() |
| +updateCourse() |
| +deleteCourse() |

Student

Enrollment

| +enrollmentId: String |
| +user: Student |
| +course: Course |
| +enrollmentDate: Date |
| +progress: Float |
| +status: String |
| +enroll() |
| +trackProgress() |

Review

| +reviewId: String |
| + student: Student |
| + course: Course |
| +rating: Int |
| +comment: String |
| +date: Date |
| +submitReview() |

Progress

| +progressId: String |
| +student: Student |
| +course: Course |
| +completedLessons: Float |
| +totalTimeSpent: Float |
| +score: Float |
| +getProgressReport() |

Notification

| +notificationId: String |
| +user: User |
| +content: String |
| +isRead: Boolean |
| +timestamp: Date |
| +markAsRead() |

Payment

| +paymentId: String |
| +student: Student |
| +amount: Float |
| +paymentDate: Date |
| +status: String |
| +method: String |
| +processPayment() |

Flow chart 2: UB -Online Learning Class Diagram with 15 Use Cases

2.9.2. Class Diagram Description

The UB - Online Learning system class diagram illustrates the structural relationships between the key entities that make up the platform's domain model. This diagram serves as a blueprint for the system's data architecture and object relationships, guiding both database design and application development.

Core Classes and Their Relationships

User Class Hierarchy:

- The User class serves as the base class with common attributes such as userID, name, email, password, and role. It implements core functionality like authentication, profile management, and notification handling.
- This base class is extended through inheritance by three specialized user types:
    - Student: Students can enroll in courses, track their progress, submit assignments, and provide reviews.
    - Instructor: Instructors create and manage courses, assess student work, and analyze course performance metrics.
    - Admin: Administrators oversee the entire platform, managing users, courses, and system settings.

Course-Related Classes:

- The Course class represents educational offerings with attributes like courseID, title, description, price, category, and duration. Each course is created and managed by an Instructor (1:N relationship), and can be enrolled in by multiple Students.
- CourseContent class maintains the actual learning materials, including videos, readings, quizzes, and assignments associated with a specific course.
- Category class allows for logical organization of courses by subject matter, facilitating effective course discovery and filtering.

Enrollment and Learning Process:

- The Enrollment class forms a critical association between Students and Courses, capturing registration date, completion status, and payment information.
- Progress class tracks a student's advancement through a course, recording completed modules, quiz scores, and time spent.
- Assignment class manages coursework submissions, deadlines, and grades.

Payment and Financial Management:

- The Payment class handles financial transactions with attributes for amount, date, status, and payment method.
- It maintains relationships with both the Enrollment class (showing what the payment was for) and the User class (identifying who made the payment).

- Discount class manages promotional codes and special pricing offers that can be applied to course enrollments.

Feedback and Communication:

- The Review class captures student feedback on completed courses, including ratings, comments, and timestamps.
- Notification class manages system alerts, reminders, and communications to users based on various platform activities.
- FAQ class stores frequently asked questions and their answers to support user self-service.

Key Relationships and Multiplicities

1. An Instructor can create and manage multiple Courses (1:N)
2. A Student can enroll in multiple Courses, and each Course can have multiple Students enrolled (M:N through the Enrollment class)
3. Each Course contains multiple instances of CourseContent (1:N)
4. A Student can submit multiple Reviews, but each Review is associated with exactly one Student and one Course (M:1:1)
5. Each Payment is associated with exactly one Enrollment (1:1)
6. Users can receive multiple Notifications (1:N)
7. Progress records are uniquely associated with a specific Student-Course pair (1:1 with Enrollment)
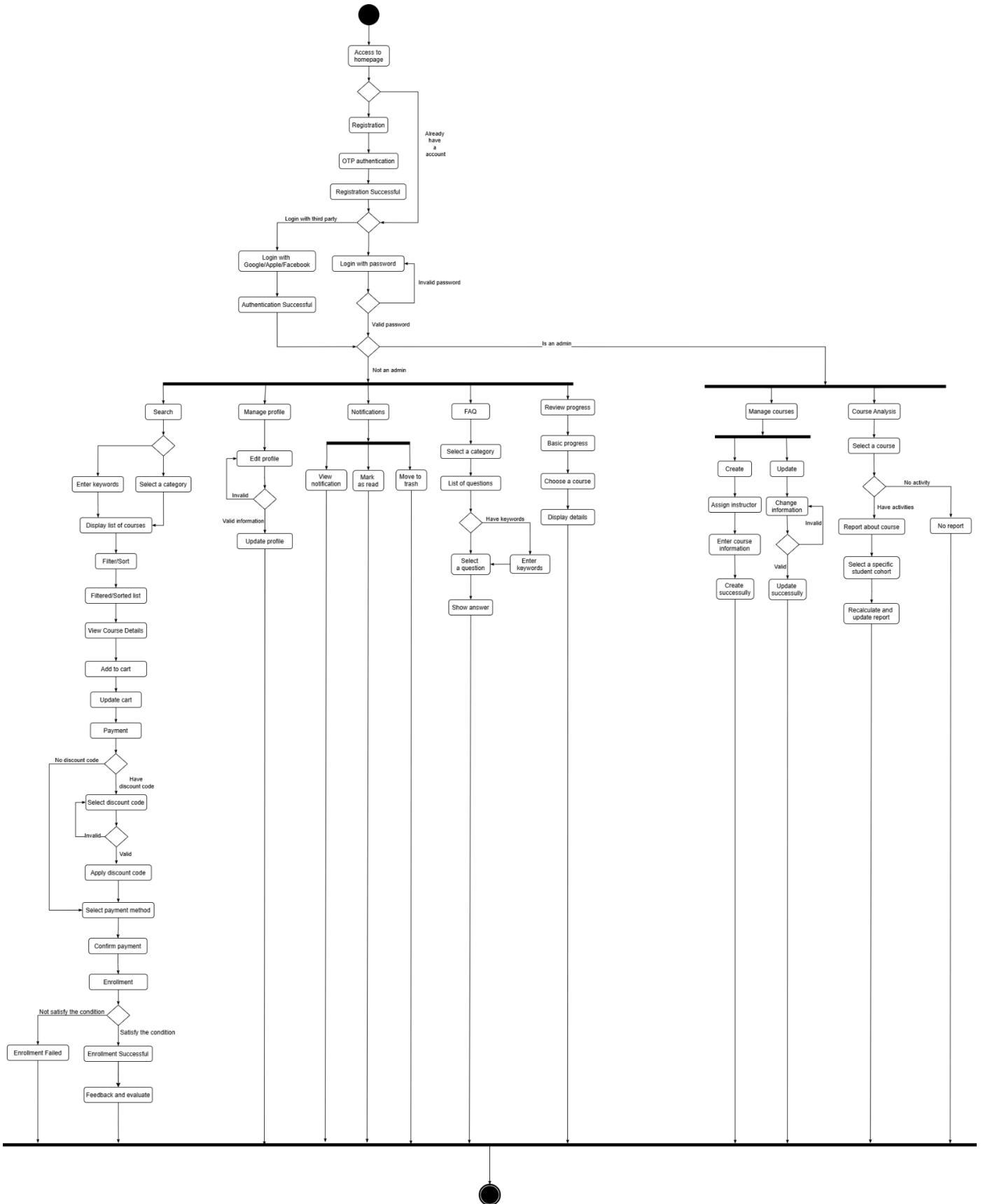
Implementation Considerations

This class model is designed to support the MongoDB document-oriented database architecture specified in the system requirements. The relationships between classes will be implemented through a combination of embedded documents and references, optimizing for both data integrity and query performance.

For the front-end React implementation, these classes will inform the component hierarchy and state management structure, ensuring consistency between the backend data model and user interface. The class diagram particularly supports the implementation of critical use cases such as course enrollment, payment processing, and progress tracking.

This comprehensive domain model ensures that the UB - Online Learning platform can efficiently manage all aspects of the online learning experience while maintaining clear separation of concerns between different system components.

2.9.3. Activity Diagram

Access to homepage

Registration

Already have a account

OTP authentication

Registration Successful

Login with third party

Login with Google/Apple/Facebook

Login with password

Invalid password

Authentication Successful

Valid password

Is an admin

Not an admin

Search — Manage profile — Notifications — FAQ — Review progress

Enter keywords — Select a category

Display list of courses

Filter/Sort

Filtered/Sorted list

View Course Details

Add to cart

Update cart

Payment

No discount code — Have discount code

Select discount code

Invalid — Valid

Apply discount code

Select payment method

Confirm payment

Enrollment

Not satisfy the condition — Satisfy the condition

Enrollment Failed — Enrollment Successful

Feedback and evaluate

Edit profile

Invalid

Valid information

Update profile

View notification — Mark as read — Move to trash

Select a category

List of questions

Have keywords

Select a question — Enter keywords

Show answer

Basic progress

Choose a course

Display details

Manage courses — Course Analysis

Create — Update

Assign instructor — Change information

Enter course information

Invalid

Valid

Create successfully — Update successfully

Select a course

No activity

Have activities

Report about course — No report

Select a specific student cohort

Recalculate and update report

Flow chart 3: UB -Online Learning Activity Diagram with 15 Use Cases
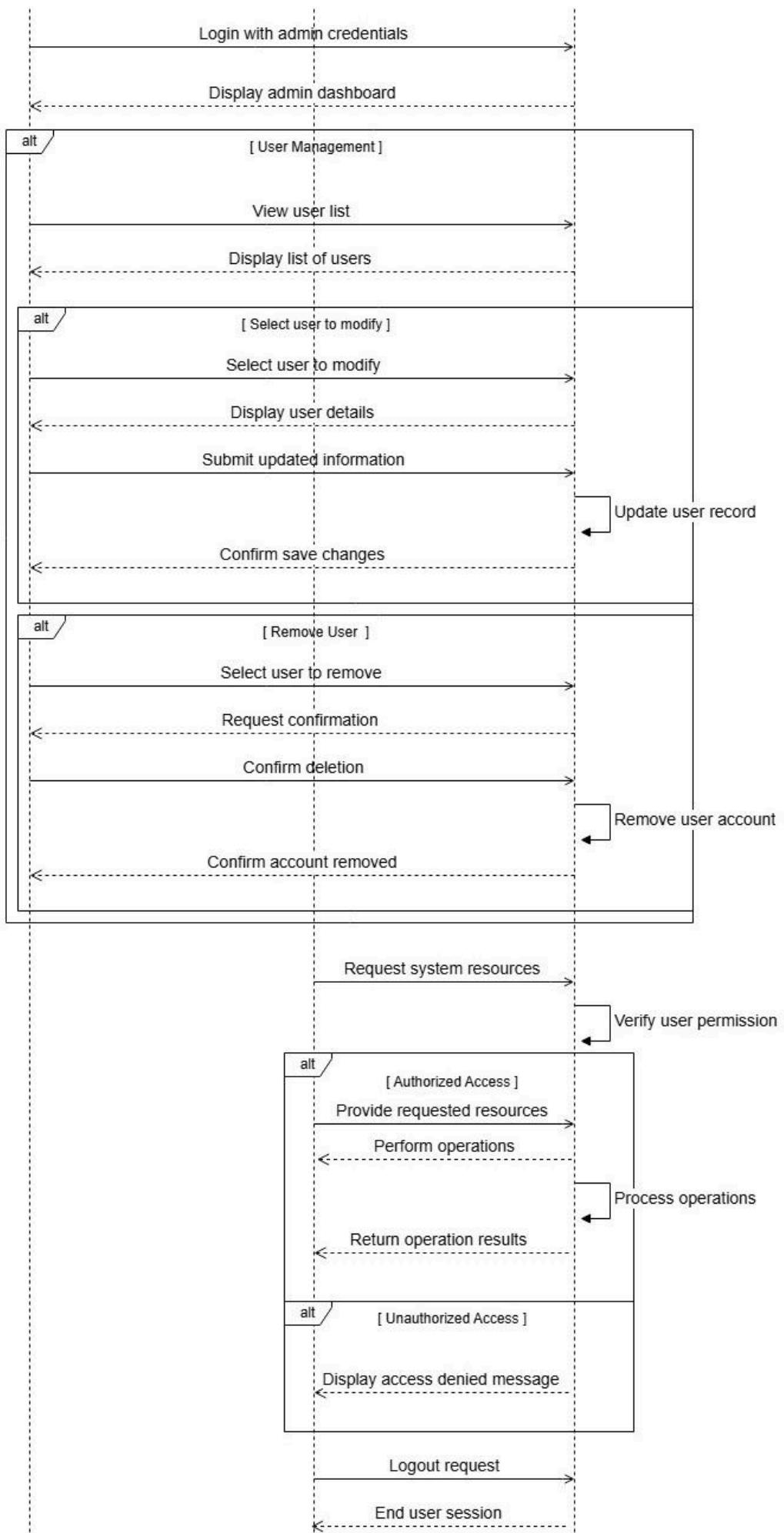
This activity diagram begins with a user initiating interaction by accessing the UB platform. The system then determines the user's role (Student, Administrator) to direct the process flow.

- For a **Student**, the activities can involve browsing and searching for courses, viewing detailed course information, adding courses to a cart , applying discount codes , proceeding to enrollment, and making payments. Once enrolled, students can access learning materials, review their progress, provide course feedback and evaluations, and manage their profile.
- **Administrators** have overarching control, including managing user accounts, overseeing courses, handling notifications, and generating various reports.

Throughout these processes, the system performs various activities like validating input data, processing transactions, updating databases, and generating notifications or error messages. Decision points are crucial, such as checking if a course is available for enrollment, validating payment information, or verifying user credentials. Alternative flows, such as signing up with social media accounts or applying a discount code, and exceptions like invalid data entry or system errors, are also represented. The diagram illustrates how these activities and decision points connect, showing the potential paths a user's interaction can take within the UB - Online Learning system until a specific task is completed or the user session ends.

2.9.4. Sequential Diagram

```
        ADMIN                    USER              DATABASE

                                   │    Request access to system   │
                                   │ ─────────────────────────────▶│
                                   │                                │
                                   │      Present login screen      │
                                   │ ◀─────────────────────────────│
   ┌──────────────────────────────────────────────────────────────────┐
   │ alt │              [ New User ]                                    │
   │                       │    Select Register option     │           │
   │                       │ ─────────────────────────────▶│           │
   │                       │    Display Registration form   │          │
   │                       │ ◀─────────────────────────────│           │
   │                       │    Submit Registration details │          │
   │                       │ ─────────────────────────────▶│           │
   │                       │                          ┌─────┐ Validate user information
   │                       │                          │     │◀──        │
   │  ┌─────────────────────────────────────────────────────────┐      │
   │  │ alt │        [ Valid information ]                        │      │
   │  │                    │                     ┌─────┐ Create User Account
   │  │                    │                     │     │◀──           │
   │  │                    │  Account created successfully │          │
   │  │                    │ ◀─────────────────────────────│          │
   │  ├─────────────────────────────────────────────────────────┤      │
   │  │ alt │        [ Invalid information ]                      │      │
   │  │                    │                                │     │      │
   │  │                    │   Return validation errors     │     │      │
   │  │                    │ ◀─────────────────────────────│     │      │
   │  └─────────────────────────────────────────────────────────┘      │
   ├──────────────────────────────────────────────────────────────────┤
   │ alt │              [ Existing User ]                               │
   │                       │                                │           │
   │                       │    Enter Login credentials     │          │
   │                       │ ─────────────────────────────▶│           │
   │                       │                          ┌─────┐ Authenticate credentials
   │                       │                          │     │◀──        │
   │  ┌─────────────────────────────────────────────────────────┐      │
   │  │ alt │        [ Valid credentials ]                        │      │
   │  │                    │                                │     │      │
   │  │                    │      Grant System access       │     │      │
   │  │                    │ ◀─────────────────────────────│     │      │
   │  ├─────────────────────────────────────────────────────────┤      │
   │  │ alt │        [ Invalid credential ]                       │      │
   │  │                    │                                │     │      │
   │  │                    │     Display error message      │     │      │
   │  │                    │ ◀─────────────────────────────│     │      │
   │  └─────────────────────────────────────────────────────────┘      │
   └──────────────────────────────────────────────────────────────────┘
```

Login with admin credentials

Display admin dashboard

alt [User Management]

View user list

Display list of users

alt [Select user to modify]

Select user to modify

Display user details

Submit updated information

Update user record

Confirm save changes

alt [Remove User]

Select user to remove

Request confirmation

Confirm deletion

Remove user account

Confirm account removed

Request system resources

Verify user permission

alt [Authorized Access]

Provide requested resources

Perform operations

Process operations

Return operation results

alt [Unauthorized Access]

Display access denied message

Logout request

End user session

The sequence diagram below illustrates the core interactions between Admins, Users, and the System, depicting the temporal flow of key processes within the application. It showcases the User Registration and Authentication flow where users can register new accounts or login with existing credentials, complete with validation processes and appropriate feedback mechanisms. For Administrators, the diagram highlights their ability to view, modify, and remove user accounts through a secure interface with confirmation safeguards. Standard user interactions are also represented, demonstrating how authorized users request resources, perform operations based on permissions, and terminate sessions. Throughout these workflows, the system consistently verifies permissions, provides appropriate feedback, and maintains secure interaction patterns—establishing a clear visual representation of the behavioral requirements that will guide development.

2.10. Assumptions and Dependencies

- The system assumes a reliable internet connection.

- Payments depend on third-party gateways.

- The platform will be hosted on a secure VPS or cloud server.

- All users are expected to have a valid email address.