Design Report

AAK Academy Management System

Course:
Software Engineering (CSC-225)

Team Members:

Salman Safdar (NUM-BSCS-2024-70)
Muhammad Haris Ajmal (NUM-BSCS-2024-49)
Farwa Imran (NUM-BSCS-2024-22)

Submission Date: January 18, 2026

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The objective of this Design Report is to translate the functional and non-functional requirements of the AAK Academy Management System into a comprehensive technical blueprint. This system is designed to automate academic and administrative operations for Matric and Intermediate programs at AAK Academy.

The design focuses on key features including:

- Subject selection with real-time fee calculation
- Automated fee calculation with tiered discounts (10% and 20%)
- Live class integration via Zoom API
- Comprehensive quiz and assignment management
- Role-based access control for Students, Teachers, and Administrators

This report serves as the final bridge between requirement gathering and system implementation, providing detailed architectural and design specifications that will guide the development phase.

Chapter 2

Design Assumptions and Constraints

To ensure the system is built within practical limits and realistic expectations, the following parameters have been identified and documented.

2.1 Assumptions

The following assumptions form the basis of the system design:

- **User Roles:** The design assumes three primary actors: Students (9th–12th grade), Teachers (subject experts), and Administrators (system managers).
- **Connectivity:** Users are assumed to have a minimum 2 Mbps broadband connection for reliable access to learning materials and Zoom sessions.
- **Browser Standards:** It is assumed that users will access the system through modern browsers such as Chrome 80+, Firefox 75+, or Safari 12+.

2.2 Constraints

The following constraints limit the scope and implementation of the system:

- **Technology Stack:** The development is constrained to open-source tools including HTML5, CSS3, JavaScript, and MySQL to maintain low licensing costs.
- **Timeline:** Project implementation must be completed within the academic semester schedule.
- **Language:** The initial release is constrained to support English only.
- **Security:** Authentication is restricted to email-based recovery, and the system must lock accounts after 5 consecutive failed login attempts.

Chapter 3

Key Design Decisions

The following architectural and design choices were made to satisfy system goals and ensure maintainability, scalability, and performance:

- **Process Decomposition in DFDs:** Processes are decomposed by user role (Student, Teacher, Admin) to ensure that the core logic for the Automated Discount System remains independent from general user profile management.
- **Class Relationships:**
  - *Composition:* Used for the Quiz-Question relationship; questions cannot exist without a parent quiz.
  - *Aggregation:* Used for the Course-Student relationship, as students exist independently of a specific course enrollment.
- **Functional Distribution:** Functionality is distributed across multiple sequence diagrams (e.g., separate flows for "Fee Payment" and "Quiz Attempt") to prevent over-complexity in a single model.
- **Third-Party Integration:** We decided to integrate the Zoom API for live classes rather than building a custom video solution to ensure stability and professional features.

Chapter 4

Requirements–Design Traceability

This chapter maps key requirements from the Software Requirements Specification (SRS) to their corresponding design artifacts, ensuring complete coverage and traceability throughout the development process.

4.1 Traceability Matrix

Table 4.1 presents the comprehensive mapping between functional and non-functional requirements and their design implementations.

Table 4.1: Requirements–Design Traceability Matrix

ID	Requirement Description	Design Artifact
FR001	Student Registration: Allow new students to create accounts with validation.	Use Case: Register; DFD: Student Registration
FR002	Teacher Registration: Allow admin to create faculty accounts.	Use Case: Register (Faculty); DFD: Faculty Registration
FR003	User Login: Authenticate users based on roles.	Use Case: Login; Sequence Diagram: Login Interaction
FR004	Access Control: Enforce role-based permissions.	Design Decision: RBAC; Class Diagram: User Roles
FR008	Subject Selection: Allow subject choice with real-time fee calculation.	Use Case: Manage Subjects; Activity Diagram: Subject Selection
FR011	Online Quiz System: Provide quiz taking with timers.	Use Case: Attempt Quizzes; Class Diagram: Quiz Class
FR013	Live Classes: Integrate with Zoom for virtual classrooms.	Use Case: Join Live Classes; Component Diagram: Zoom API
FR029	Automated Discount: Apply 10% for 3 subjects and 20% for 6 subjects.	Use Case: Fee Calculation; DFD: Fee Information
FR030	Report Generation: Generate academic and financial reports.	Use Case: Generate Reports; DFD: Student Results
NFR001	Performance: Page load time under 3 seconds.	System Architecture: Web Tier Optimization

Continued on next page

Table 4.1 – continued from previous page

ID	Requirement Description	Design Artifact
NFR018	Security: Passwords stored using bcrypt hashing.	Design Decision: Secure Auth; Class Diagram: User Attributes
NFR053	Backup: Weekly full and daily incremental backups.	Use Case: System Configuration; Maintenance Plan

Chapter 5

System Design Diagrams

This chapter provides a detailed technical description of each diagram required for the AAK Academy Management System Design Report. These diagrams visualize system behavior and structure while directly supporting the SRS requirements.

5.1 Use Case Diagram

The Use Case Diagram (Figure 5.1) serves as the high-level overview of system functionality.

- **Actors:** It identifies three primary human actors (Student, Faculty, and Admin) and external systems like the Zoom API and Payment Gateway.
- **Core Interactions:** It maps essential tasks such as Register, Login, Manage Subjects, and Join Live Classes.
- **Logical Links:** It uses `<<include>>` relationships to show that Manage Subjects automatically triggers Fee Calculation with tiered discounts.

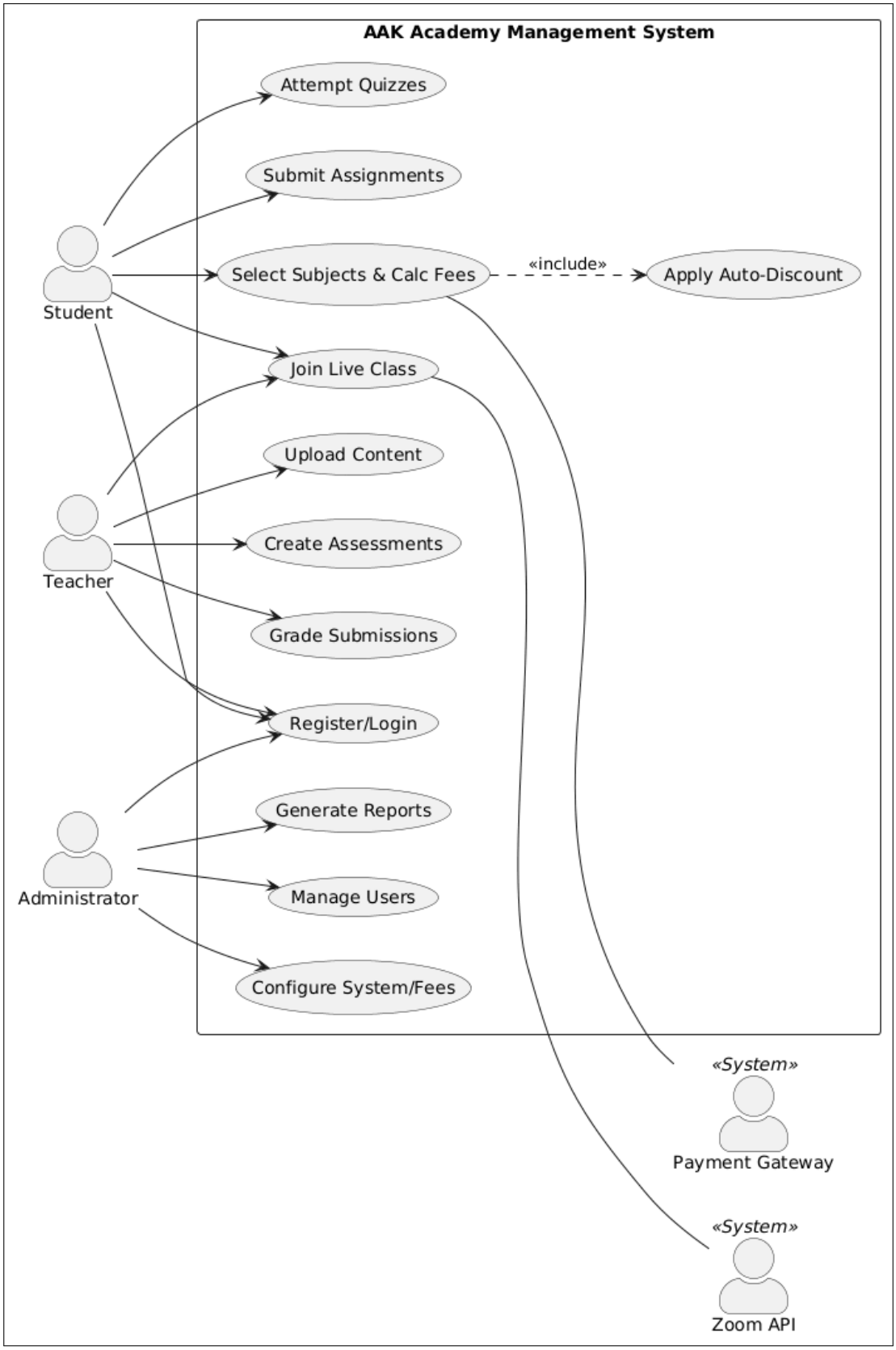


Figure 5.1: Use Case Diagram for AAK Academy Management System

5.2 Data Flow Diagrams (DFD)

Data Flow Diagrams illustrate how data moves through the system, starting from a high-level view down to specific processes.

5.2.1 Level 0 (Context Diagram)

Figure 5.2 shows the entire system as a single process interacting with external entities. For example, it shows Student providing "paid fee" and receiving "marks and result".

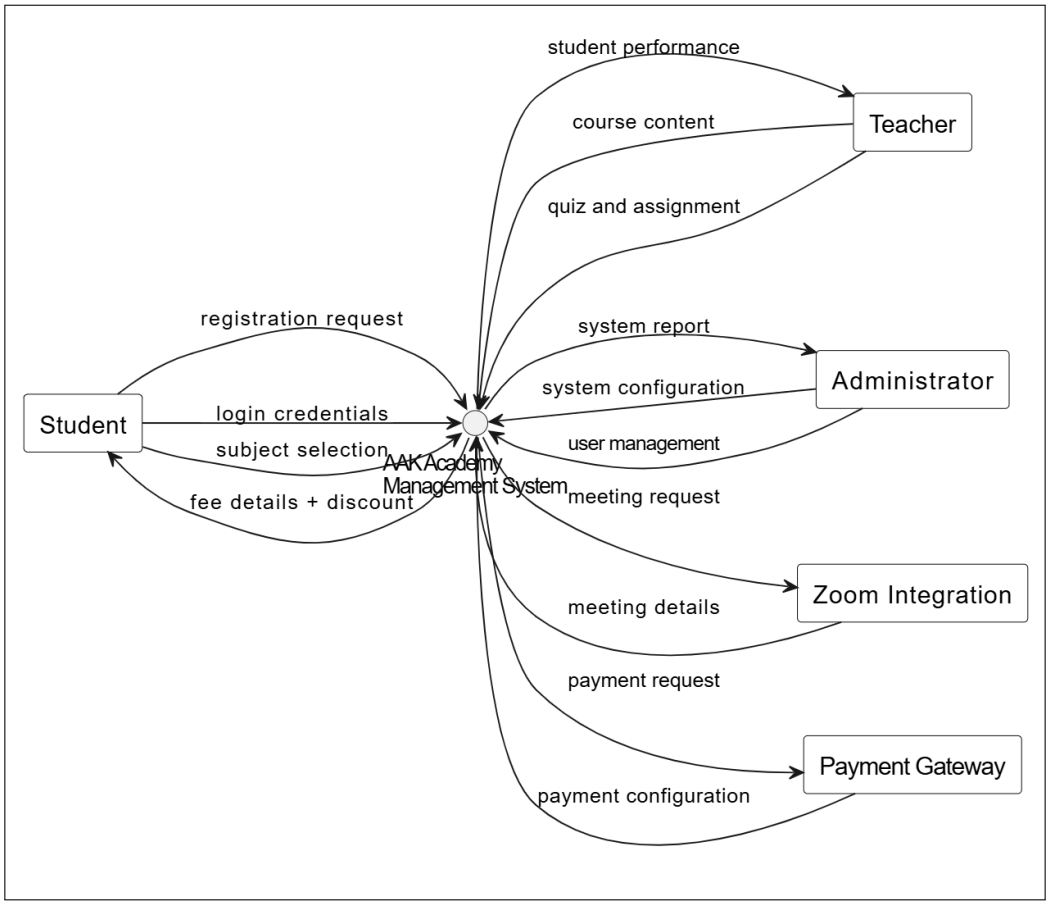


Figure 5.2: Data Flow Diagram – Level 0 (Context Diagram)

5.2.2 Level 1 DFD

Figure 5.3 breaks the system into major functional modules like User Management, Academic Management, and Financial Management.

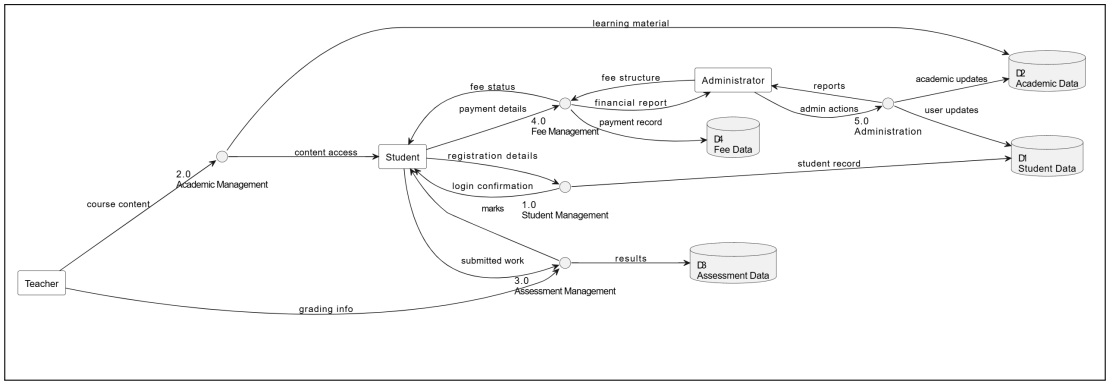


Figure 5.3: Data Flow Diagram – Level 1

5.2.3 Level 2 DFD

Figure 5.4 provides the most detail, such as the specific steps inside the Fee Calculation process where the 10% or 20% discount logic is applied.

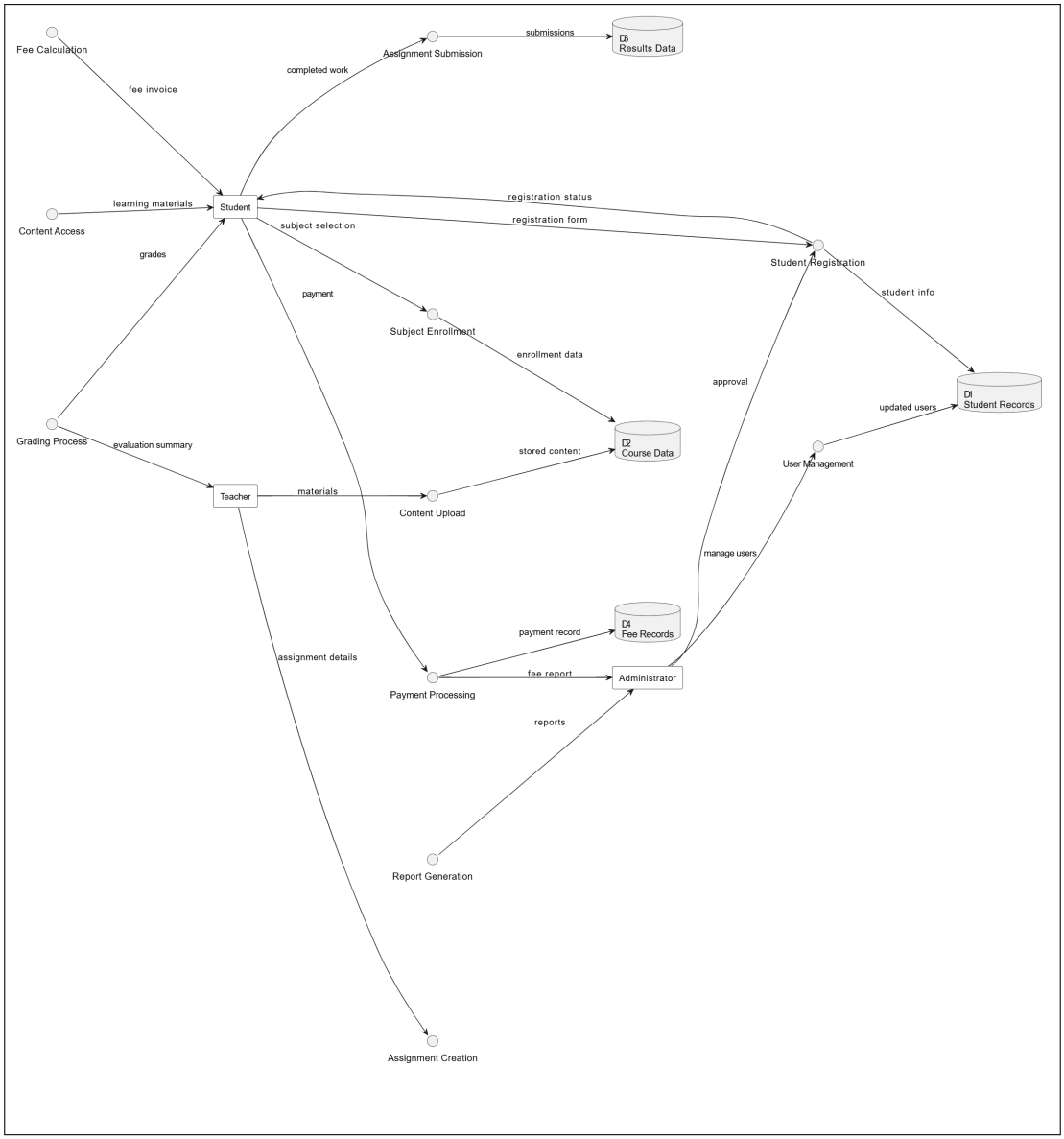


Figure 5.4: Data Flow Diagram – Level 2 (Fee Calculation Detail)

5.3 Activity Diagrams

Activity diagrams represent the operational workflows and decision logic for various tasks within the system.

5.3.1 Student Workflow

Figure 5.5 models the path from login to subject selection, highlighting the decision nodes for applying 10% (for 3 subjects) or 20% (for 6 subjects) automatic discounts.

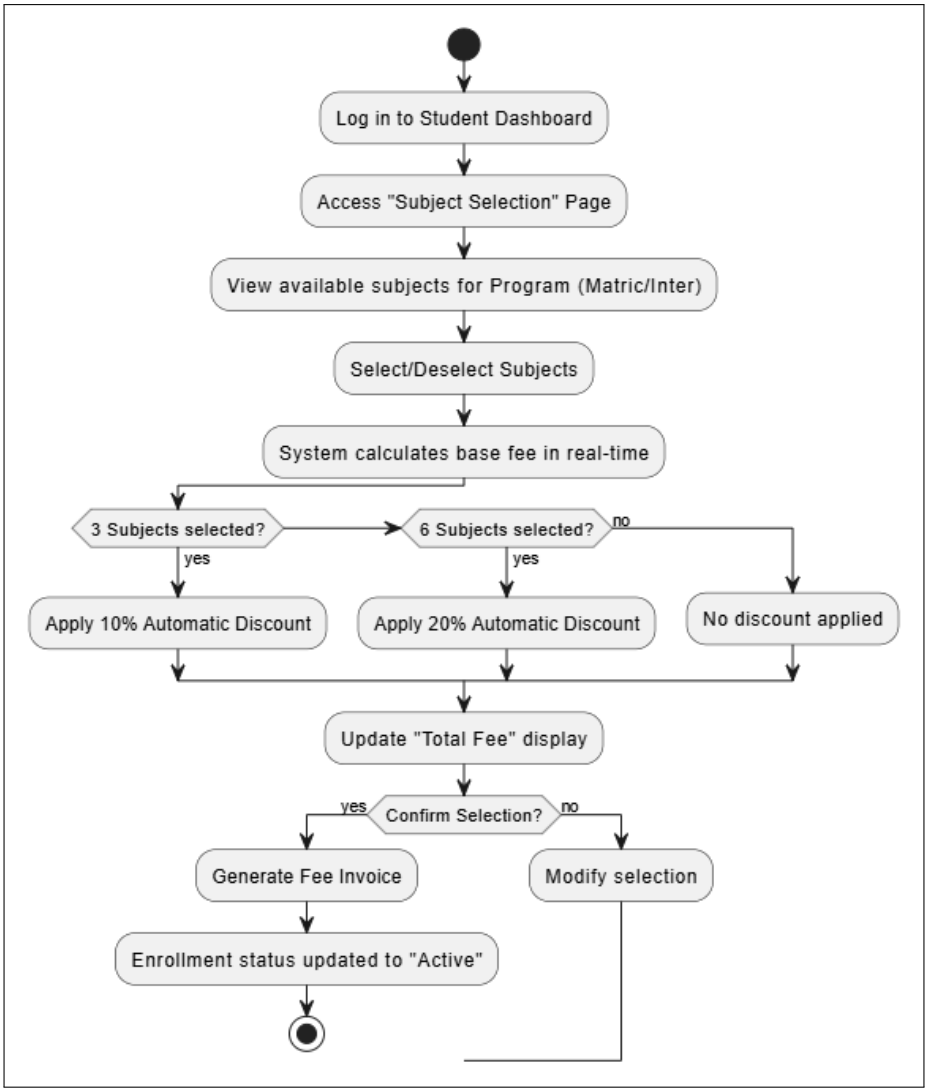


Figure 5.5: Activity Diagram – Student Workflow

5.3.2 Teacher Workflow

Figure 5.6 shows the process of uploading course content, including a validation step to ensure file types (PDF/Video) are correct before notifying students.

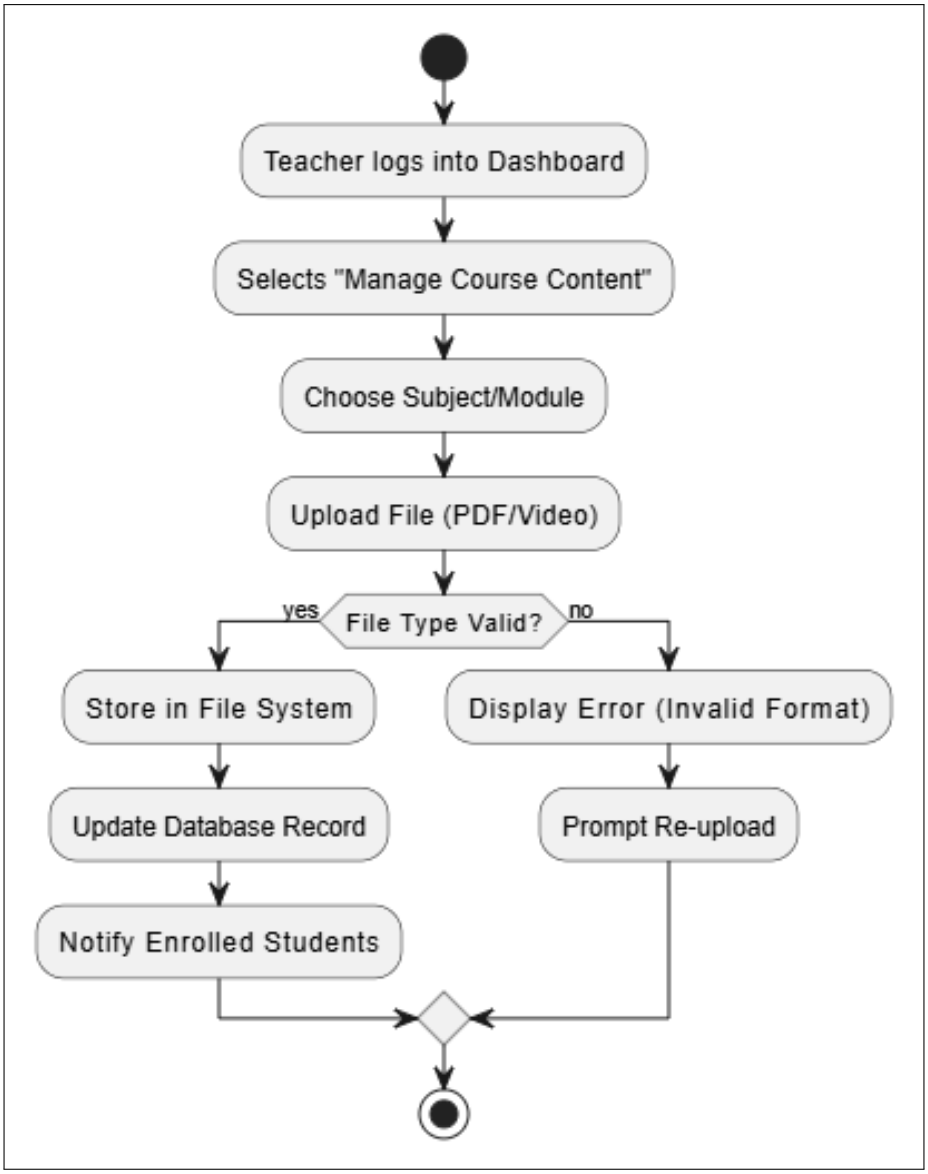


Figure 5.6: Activity Diagram – Teacher Workflow

5.3.3 Admin Workflow

Figure 5.7 details the registration approval process, from receiving a notification to verifying documents and generating a Student ID.

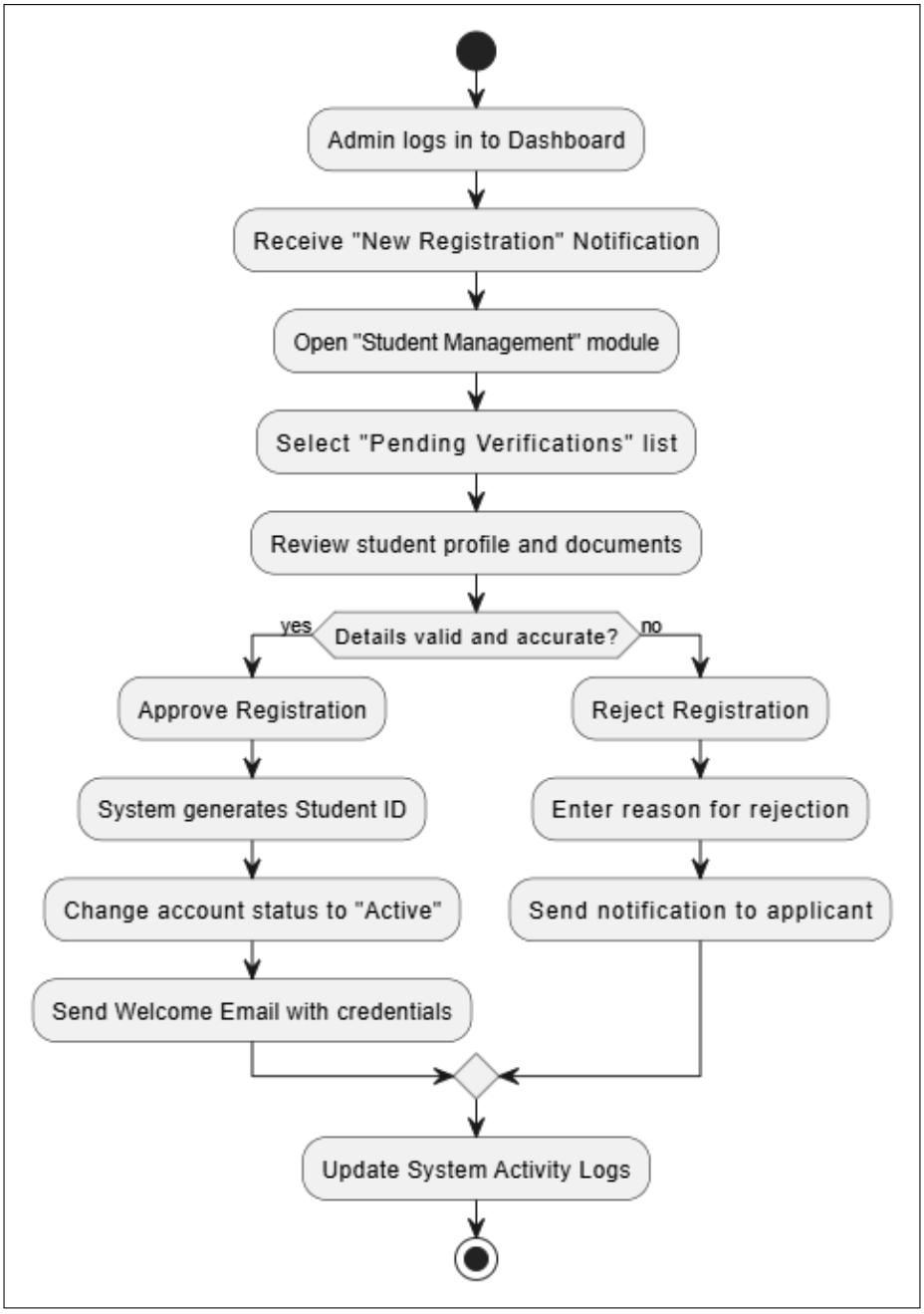
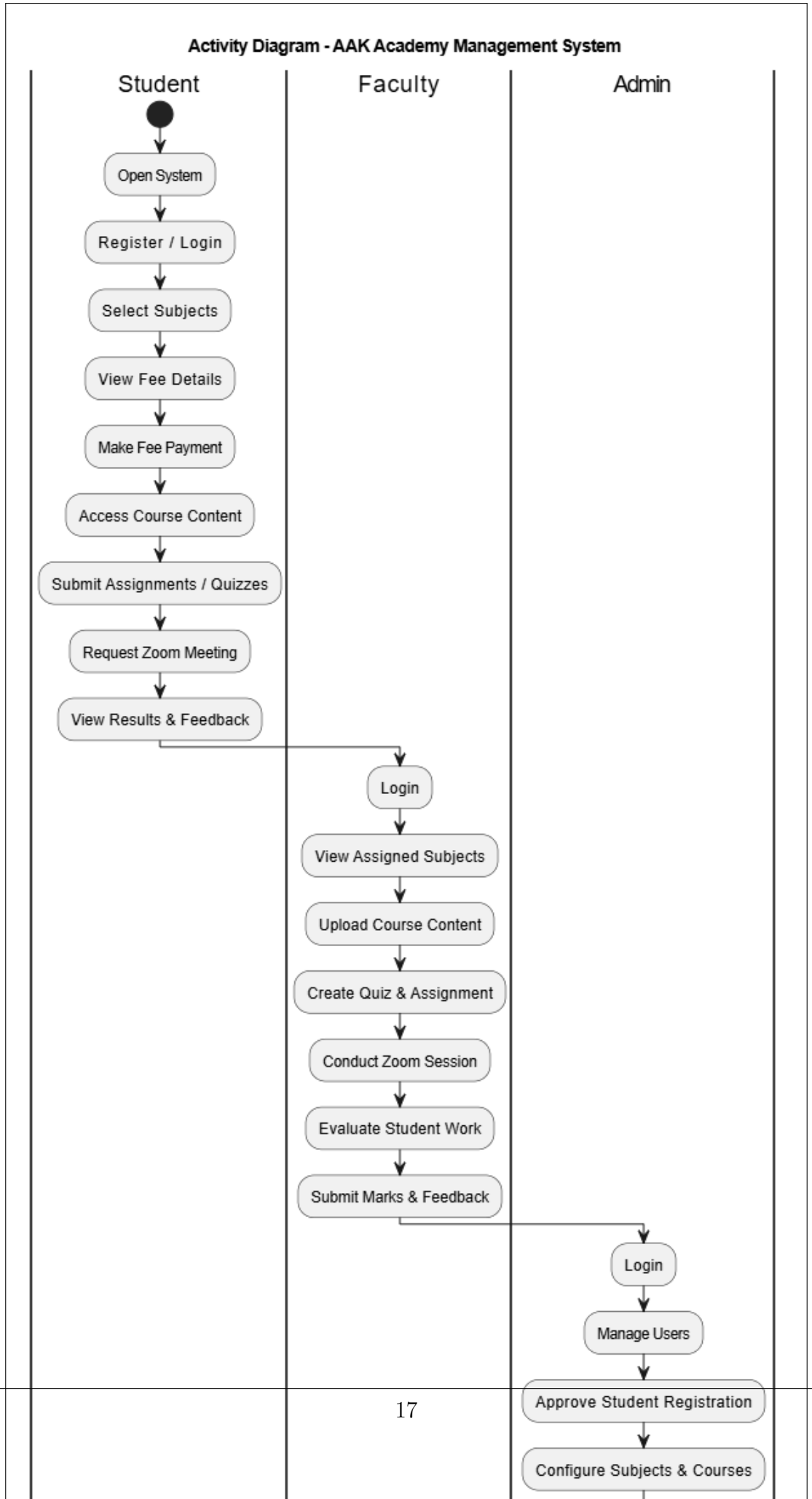


Figure 5.7: Activity Diagram – Admin Workflow

5.3.4 Overall System Activity

Figure 5.8 presents the comprehensive system activity diagram showing the interaction of all workflows.


Activity Diagram - AAK Academy Management System
Student
Faculty
Admin
Open System
Register / Login
Select Subjects
View Fee Details
Make Fee Payment
Access Course Content
Submit Assignments / Quizzes
Request Zoom Meeting
View Results & Feedback
Login
View Assigned Subjects
Upload Course Content
Create Quiz & Assignment
Conduct Zoom Session
Evaluate Student Work
Submit Marks & Feedback
Login
Manage Users
Approve Student Registration
Configure Subjects & Courses

5.4 Sequence Diagrams

Sequence Diagrams model the time-ordered interactions between objects for key functionalities. These are used for complex tasks to show exactly how the User, Interface, Controller, and Database communicate.

5.4.1 Student Registration and Login

Figure 5.9 illustrates the interaction flow for student registration and authentication processes.

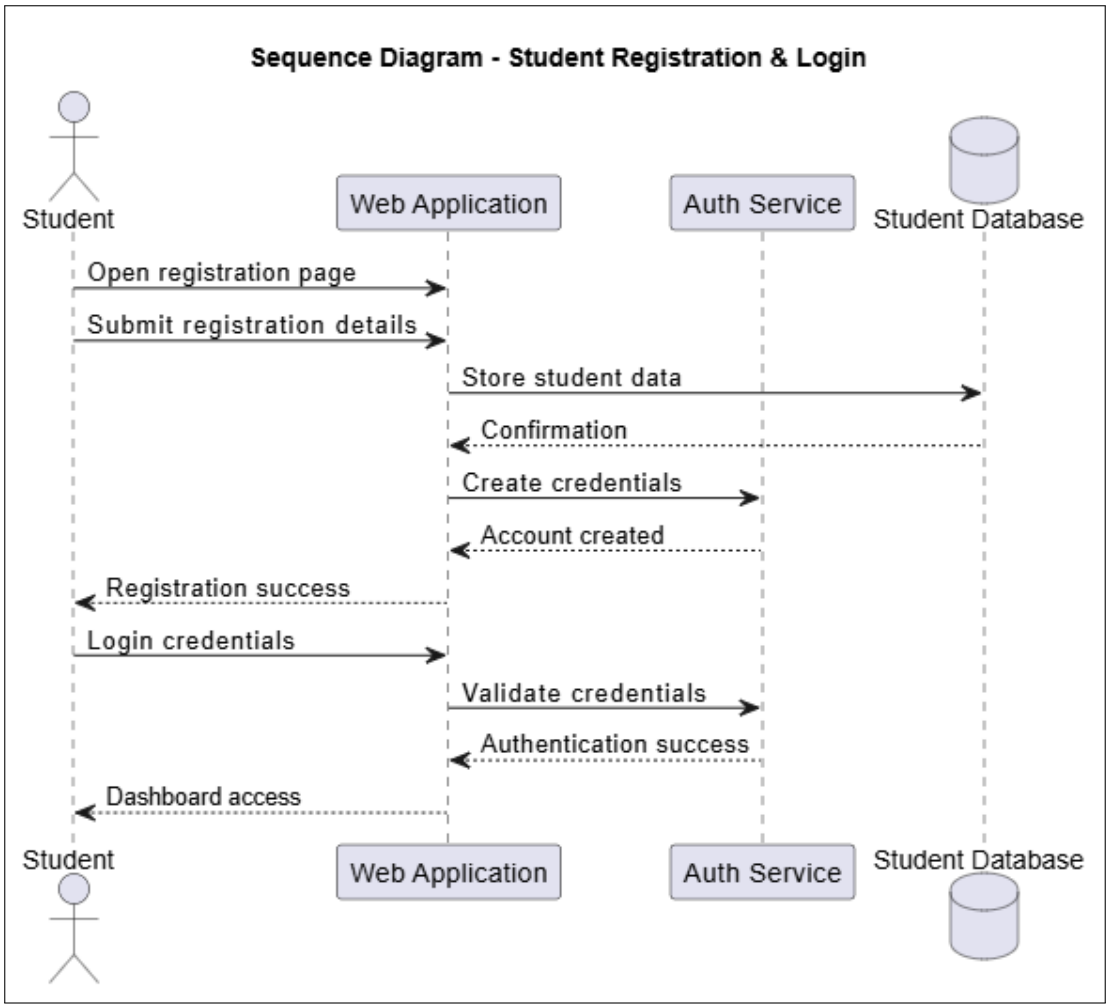


Figure 5.9: Sequence Diagram – Student Registration and Login

5.4.2 Fee Payment and Administration

Figure 5.10 demonstrates the sequence of operations involved in fee calculation, payment processing, and administrative oversight.

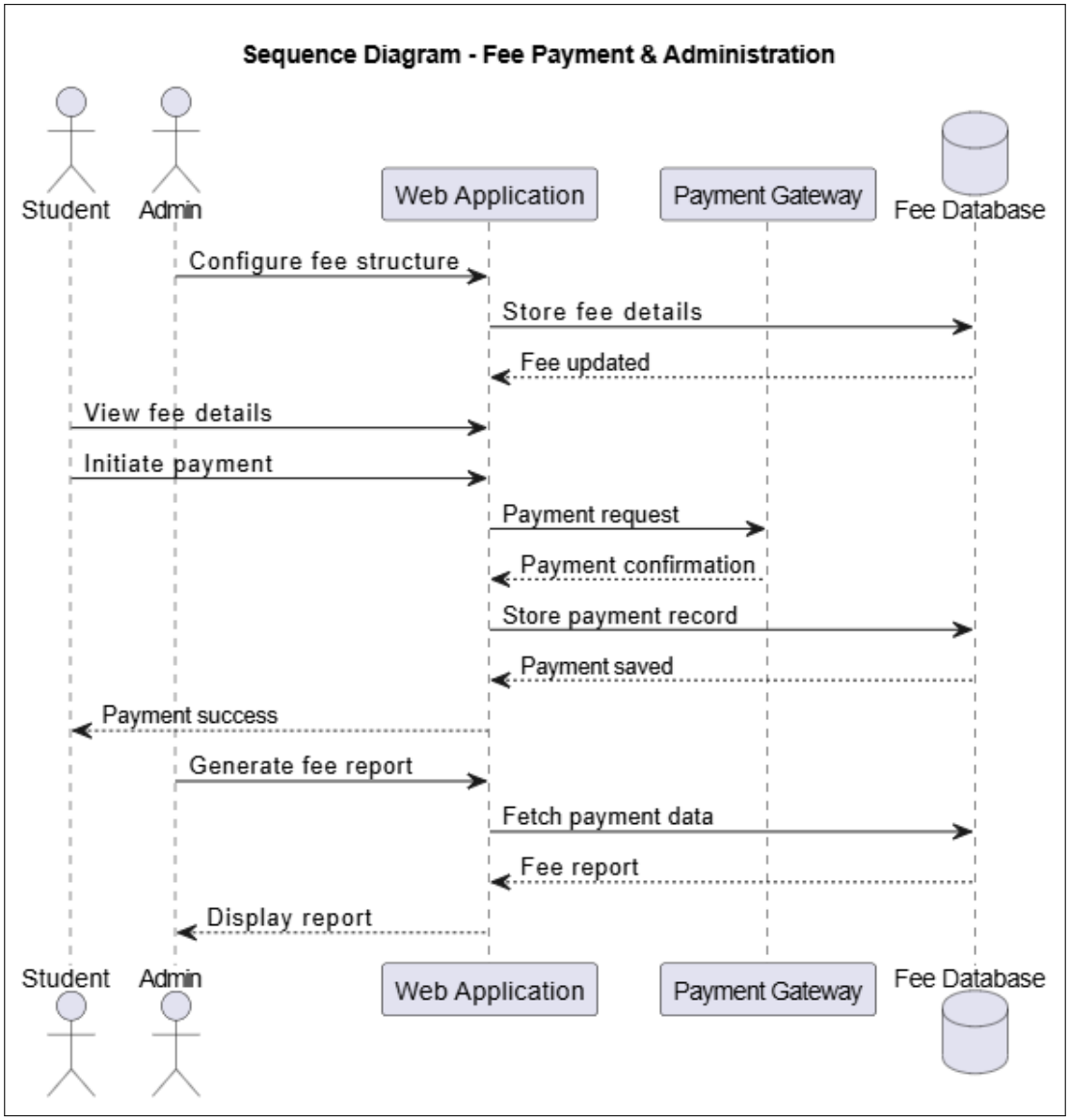


Figure 5.10: Sequence Diagram – Fee Payment and Administration

5.4.3 Course Content Management

Figure 5.11 shows the workflow for teachers uploading and managing course materials.

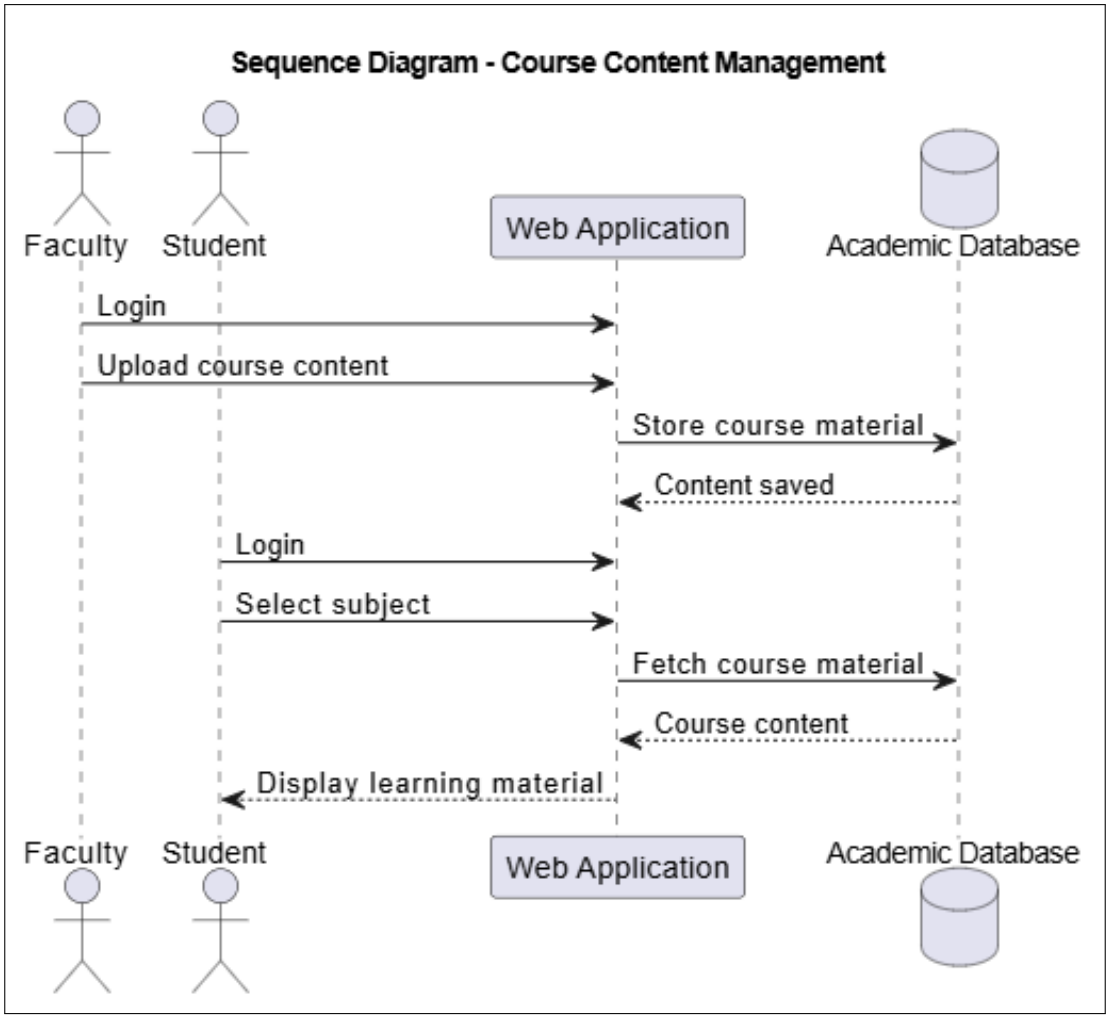


Figure 5.11: Sequence Diagram – Course Content Management

5.4.4 Assignment Submission and Assessment

Figure 5.12 depicts the complete cycle of assignment submission by students and grading by teachers.

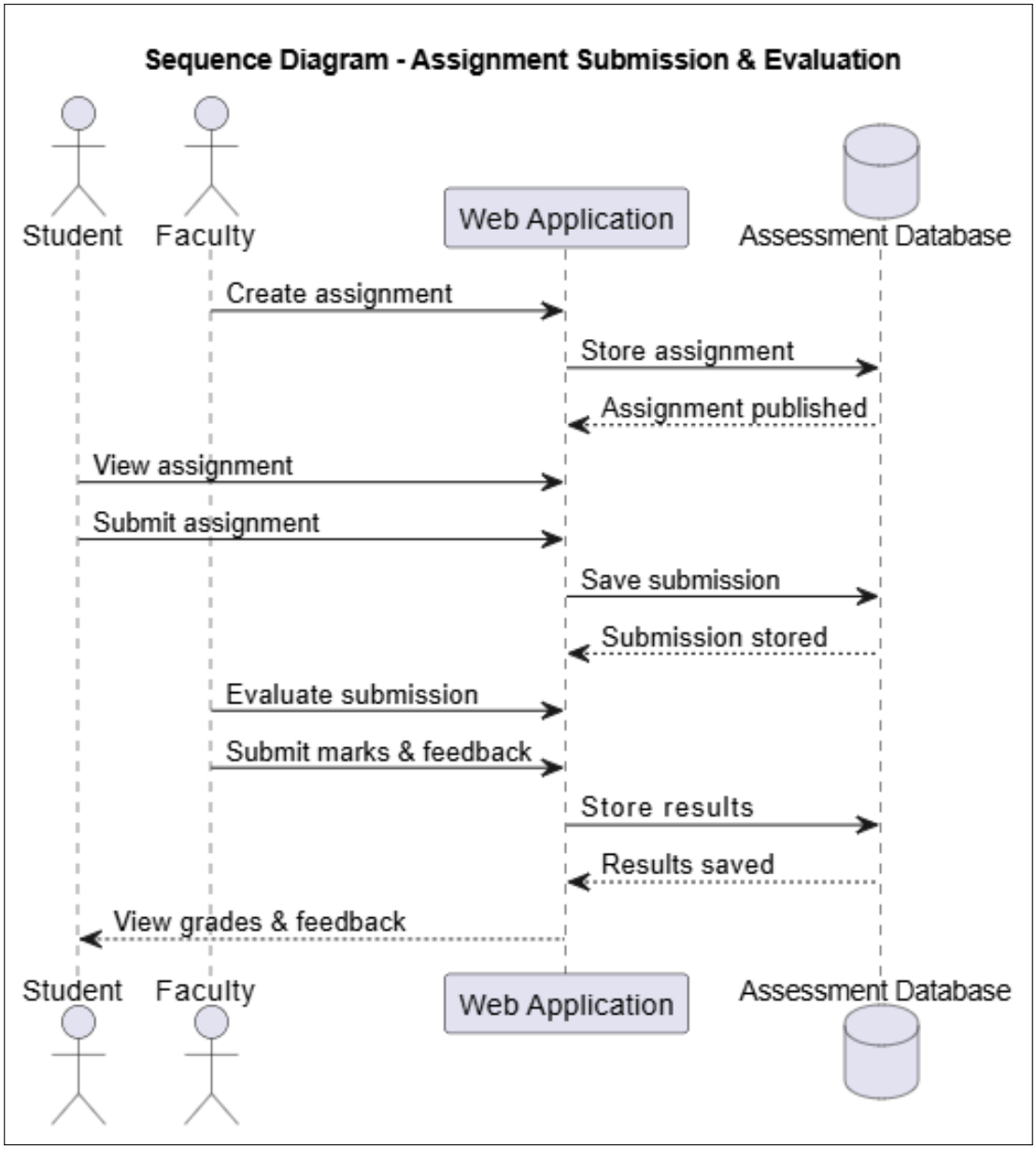


Figure 5.12: Sequence Diagram – Assignment Submission and Assessment

5.4.5 Complete System Sequence Diagram

Figure 5.13 presents the comprehensive sequence diagram showing all major system interactions.

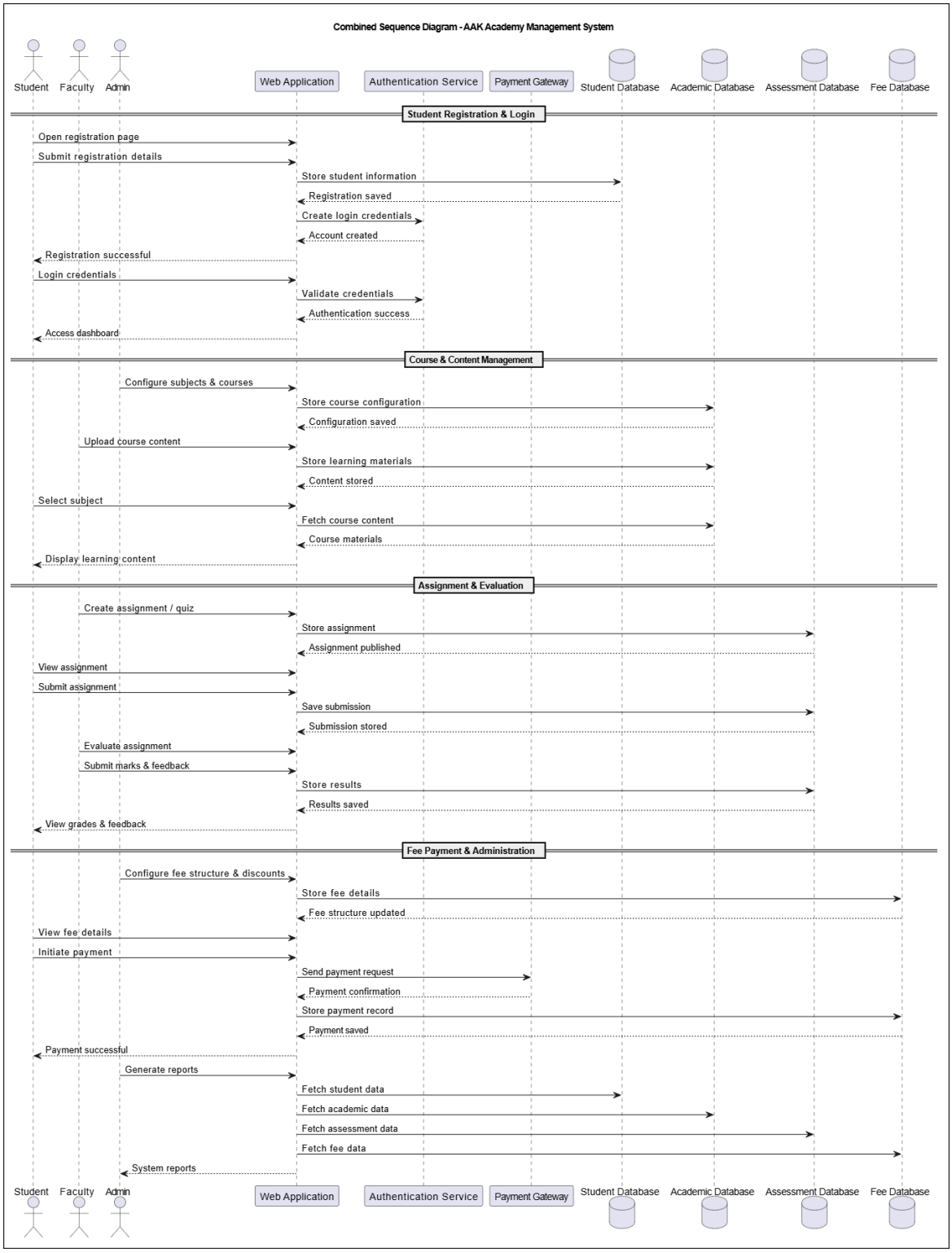


Figure 5.13: Complete AAK Academy System Sequence Diagram

5.5 Class Diagram

The Class Diagram (Figure 5.14) reflects the static structure of the system, including attributes and relationships.

- **Structure:** It features a User base class with subclasses for Student, Teacher, and Admin to handle Role-Based Access Control (RBAC).
- **Relationships:** It uses Composition for the Enrollment and Subject relationship and Aggregation for CourseContent managed by Teachers.

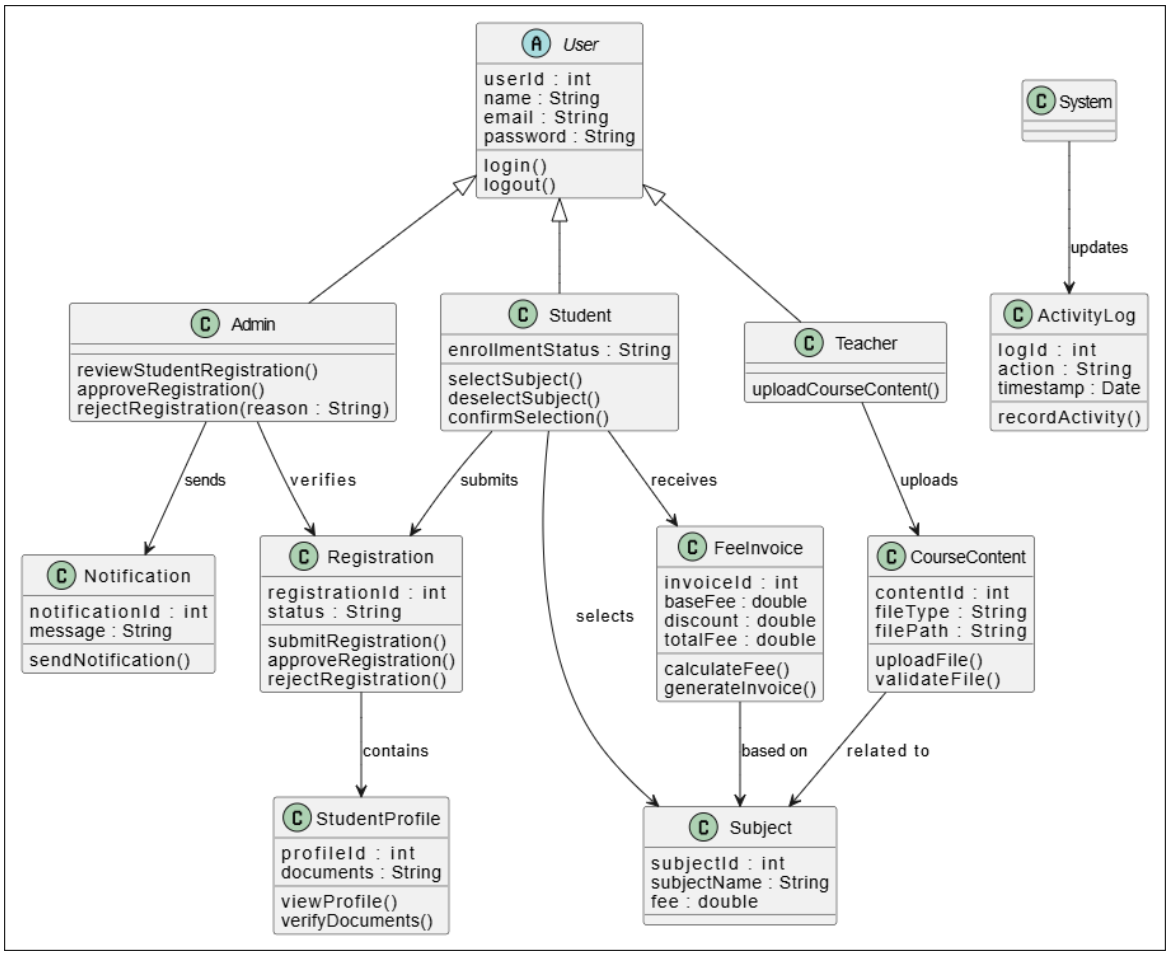


Figure 5.14: Class Diagram for AAK Academy Management System

5.6 Component Diagram

The Component Diagram (Figure 5.15) illustrates the high-level software components and their dependencies.

- **Modular Design:** It shows the relationship between the Web UI, the Business Logic Layer (handling discounts and grading), and the Database Tier.
- **Integrations:** It highlights dependencies on external components like the Zoom REST API for live class functionality.

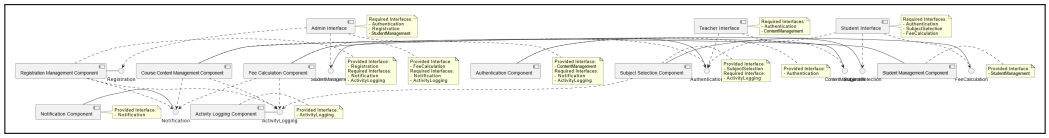


Figure 5.15: Component Diagram showing System Architecture

Chapter 6

Project Resources

This chapter provides links to external resources and repositories associated with the AAK Academy Management System project.

6.1 Repository and Prototype Links

- **GitHub Repository:** https://github.com/salmansafdarr/SOFTWARE-ENG-PROJECT
- **Figma Interactive Prototype:** https://www.figma.com/proto/ydfJnbavV8oPTpYEaPAm69/AAK-Academy?node-id=0-1&t=1t8M02Iw2xl4fXUc-1