ACADEMIC MANAGEMENT SYSTEM

Supervisor
Ma'am Alina Munir

Submitted by
Muhammad Salman
{2021_SE_04}
Numan Ahmed
{2021_SE_24}
Farjad Waseem
{2021_SE_56}

Department of Computer Science,
University of Engineering and Technology, Lahore, New-Campus.
[4-October-2022]

TABLE OF CONTENTS

1. Project Description

Our **Academic Management Software** is designed to efficiently manage student profiles, attendance, and fees, while addressing the critical need to minimize proxy attendance in educational institutions.

Key Features:

- **Student Profile Management:** Academic administrators can easily create and update student profiles, including guardian contact information, monthly fees, and due dates. Biometric fingerprint scans are securely stored for student identification.

- **Biometric Attendance Tracking:** Students can mark their attendance using a biometric fingerprint scanner, ensuring accurate and real-time attendance records. Automated notifications are sent to guardians if a student is absent after a specified threshold time.

- **Comprehensive Reporting:** Administrators have access to detailed reports on student attendance and fee-related data. They can set due dates for late fees and efficiently manage fee collections. The system generates reminder notifications for fee payments to prevent missed deadlines.

- **Guardian Notifications:** Guardians receive notifications regarding their child's attendance, including arrival and departure times, as well as absence alerts, via SMS or email.

- **Fee Management:** Students or guardians can pay fees through the administrative office or bank deposits using vouchers. Paid vouchers are submitted to the administrator for fee updates, and the system generates and tracks them. Administrators can record fee payments, whether in cash or through bank transactions. Automated notifications inform guardians about the fee payment and chosen method.

- **Voucher Generation:** The software generates vouchers for student fees, ensuring a streamlined fee collection process.

Summary: Our innovative system offers comprehensive solutions to manage student fees, accurately track attendance through biometric technology, and promptly notify guardians about attendance and fee-related matters. Additionally, it addresses the critical issue of minimizing proxy attendance in educational institutions, enhancing the overall efficiency of academic administration.

2. Problem Statement and Analysis

Problem Domain:

In educational institutions, administrators face student attendance and fee reminder challenges. Due to these challenges, they must provide timely information about absentees on a daily basis and send fee reminders at the beginning of each month to parents.

Problem Significance:

The significance lies in saving administrators time and improving efficiency by automating attendance and fee reminders, which currently consume about **20%** of their daily work hours.

Business Rules:

SMS sending adheres to PTA regulations for each notification type to ensure efficient parent communication via SMS

- Max 750 SMS per day
- Max 250 SMS per day
- Max 150 SMS every 15 minutes

3. Requirement Analysis

Functional Requirements

A. User Requirements

1. **Student Profile Management:**

    - Administrators should be able to easily create, update, and view student profiles.
    - Student profiles should include information such as guardian contact details, monthly fees, and due dates.

2. **Biometric Attendance Tracking (Attendance Mode):**

    - Students should be able to use a biometric fingerprint scanner to mark their attendance accurately in attendance mode.
    - The system should provide real-time attendance records in attendance mode.
    - Automated notifications should be sent to guardians if a student is absent beyond a specified threshold in attendance mode.

3. **Comprehensive Reporting (Administrative Mode):**

    - Administrators should have access to detailed reports on student attendance and fee-related data in administrative mode.
    - They should be able to set due dates for late fees and efficiently manage fee collections in administrative mode.
    - Reminder notifications for fee payments should be generated to prevent missed deadlines in administrative mode.

4. **Guardian Notifications (Administrative Mode):**

    - Guardians should receive notifications about their child's attendance, including arrival and departure times and absence alerts in administrative mode.
    - Notifications should be sent via SMS and email, allowing guardians to stay informed in administrative mode.

5. **Fee Management (Administrative Mode):**

    - Students or guardians can pay fees through the administrative office or bank deposits using vouchers in administrative mode.
    - Paid vouchers are submitted to the administrator for fee updates in administrative mode.
    - The system generates and tracks paid vouchers, allowing administrators to record fee payments, whether in cash or through bank transactions in administrative mode.

- Automated notifications are sent to guardians to confirm fee payments and the chosen payment method in administrative mode.

B. System Requirements

1. Programming Language:

- The system will be developed using Java.

2. Database:

- The system will require a database to store student profiles, attendance records, fee-related data, and notifications.
- The system will use Mongo-DB as its database

3. Biometric Fingerprint Scanner Integration (Attendance Mode):

- Integration with biometric fingerprint scanners is necessary for accurate attendance tracking in attendance mode.
- Biometric Fingerprint Scanner can be
- Digital Persona U are U 4500 Fingerprint reader
- R307 Fingerprint Reader Optical Module Sensor

4. Third-Party SMS API (Administrative Mode):

- Integration with a third-party SMS API is essential for sending SMS notifications in administrative mode.
- The system should be capable of sending a maximum of 750 SMS messages per day, 250 SMS messages per day for each specific notification type, and 150 SMS messages every 15 minutes, in compliance with PTA regulations.

5. User Authentication:

- Secure user authentication and authorization mechanisms should be in place to ensure data privacy and security.
- Password-based authentication is required to switch from attendance mode to administrative mode.

6. Reporting and Notification Generation (Administrative Mode):

- The system should have the capability to generate detailed reports and notifications in various formats, including SMS and email in administrative mode.

7. **User Interface:**

    - The user interface should be user-friendly and accessible to administrators, teachers, and guardians.

8. **Security Measures:**

    - The system should implement security measures to protect sensitive student data, biometric information, and communication channels.

9. **Scalability and Performance:**

    - The system should be scalable to accommodate a growing number of users and students.
    - It should be designed for optimal performance to handle the data and notification load efficiently.

10. **Backup and Recovery:**

    - Data backup and recovery mechanisms should be in place to ensure data integrity and availability in case of system failures.

Non Functional Requirements

1. **Performance:** The system should be responsive and It should provide real-time attendance tracking and generate notifications promptly.

2. **Security:** The system should not display student's biometric information. Password-based authentication should ensure access control to the administrative mode.

3. **Usability:** The user interface should be intuitive and user-friendly, making it easy for administrators, and students to use the system effectively.

4. **Scalability:** The system should be designed to accommodate a growing number of users and students as educational institutions expand.

5. **Compliance:** The system must adhere to PTA regulations regarding SMS messaging limitations, ensuring compliance with maximum SMS limits per day and per 15 minutes.

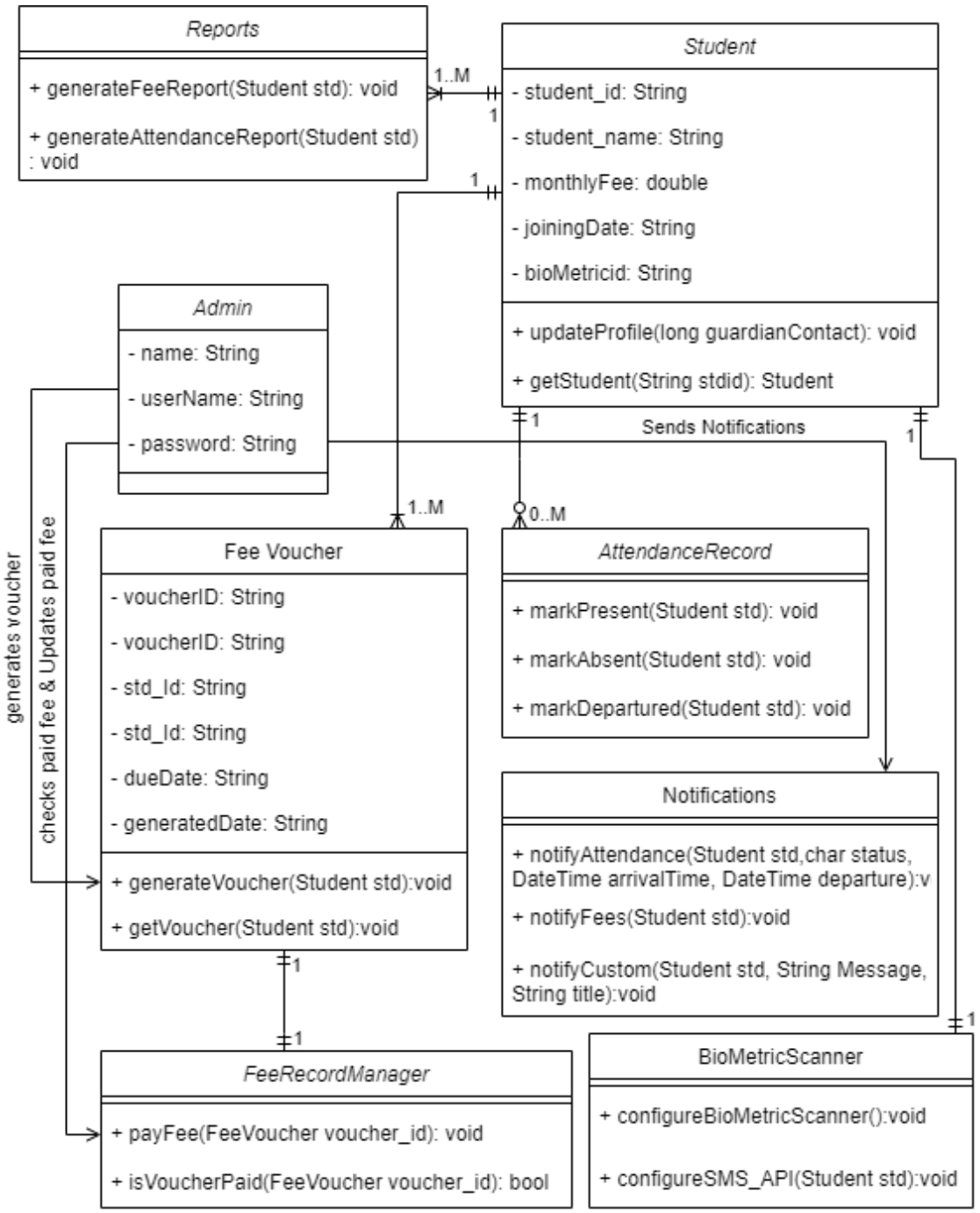
6. **Reliability:** The system should be reliable, with backup and recovery mechanisms in place to ensure data integrity and availability in case of system failures.

7. **Interoperability**: The system should be compatible with standard hardware and operating systems to facilitate easy deployment and maintenance.

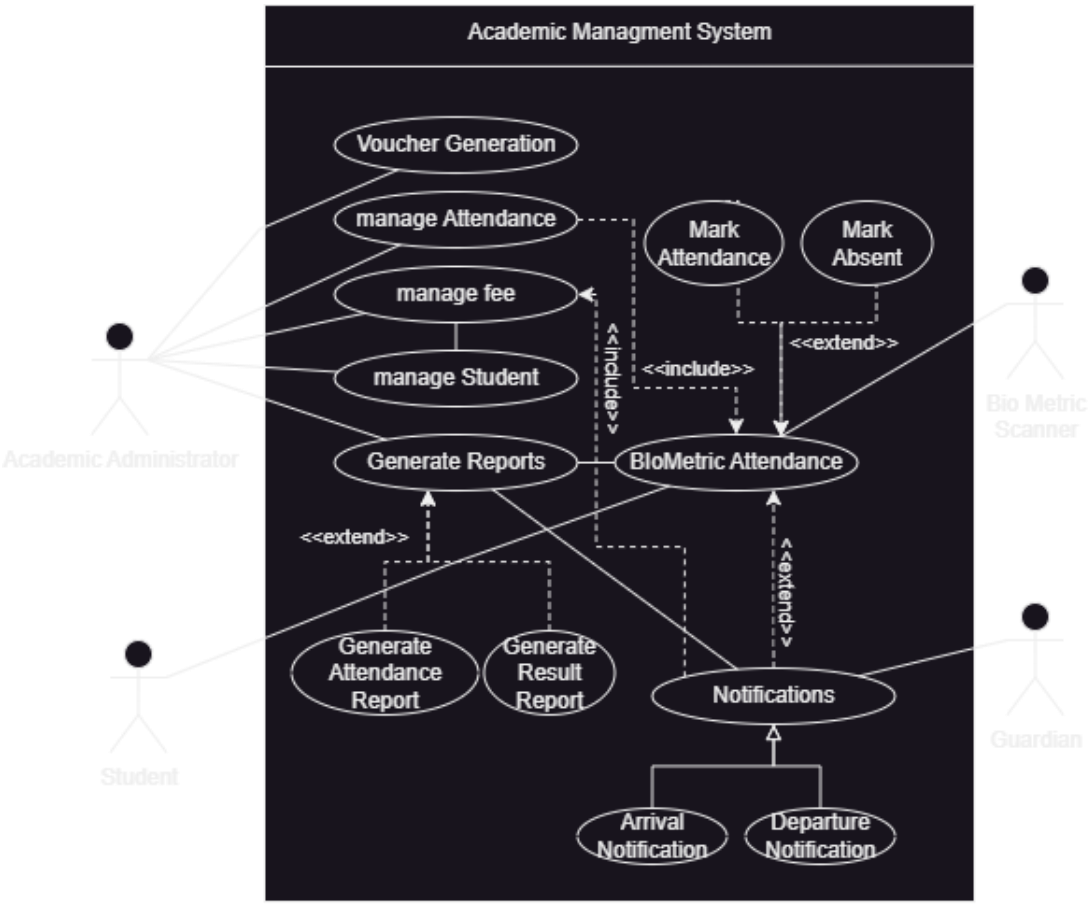
8. **Availability:** The system should be available and accessible to users with minimal downtime for maintenance or updates.

4. System Design:

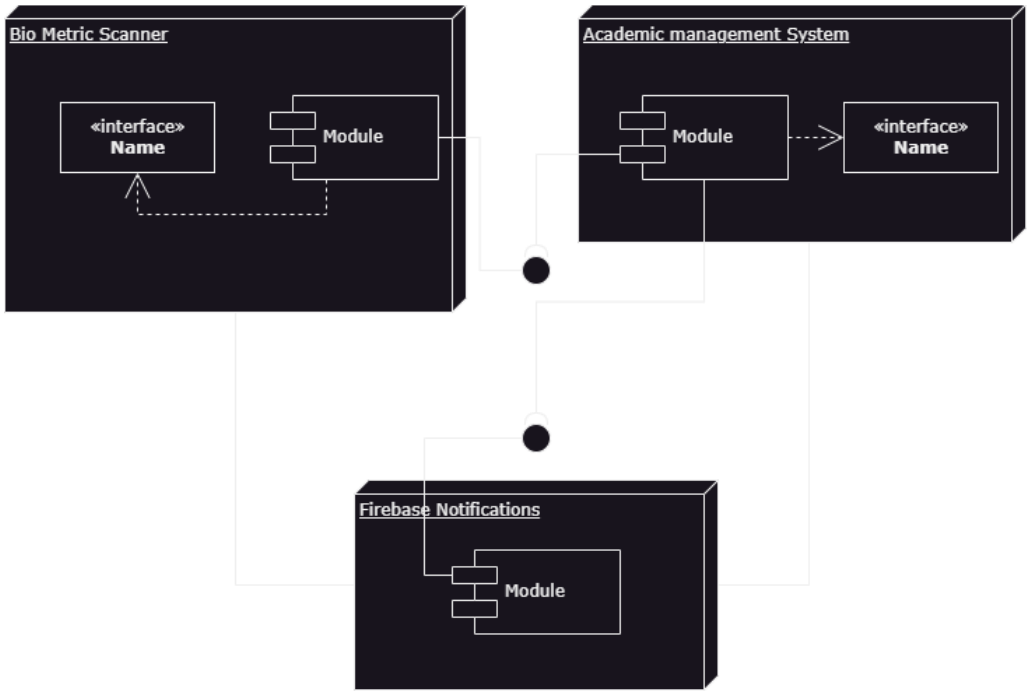
I. Class Diagram:



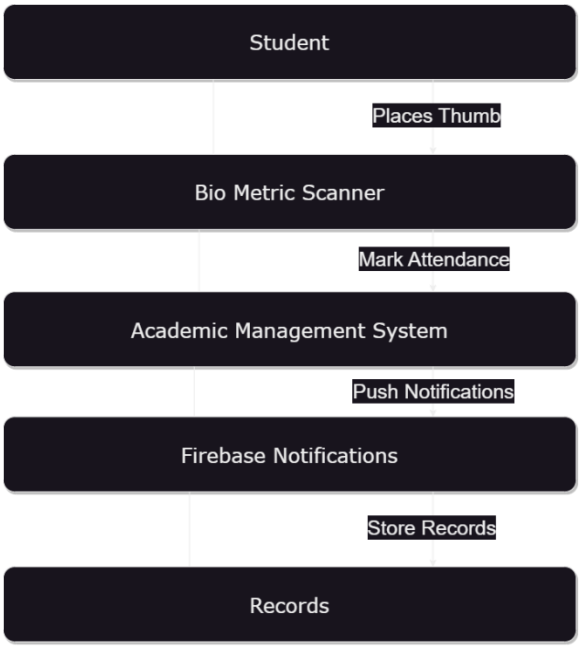
II. Use-Case Diagram:



III. Component Diagram:



IV. Collaboration Diagram:



V. Wireframes:

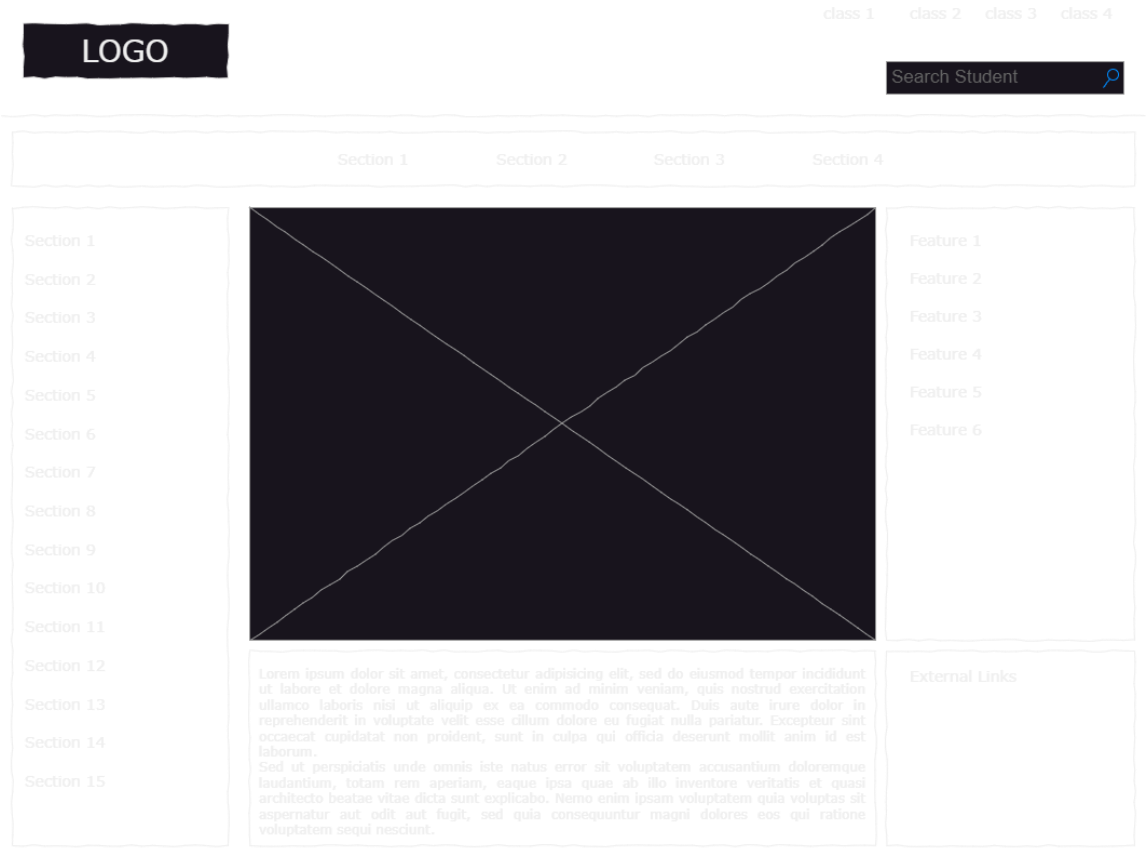


Figure 1.1

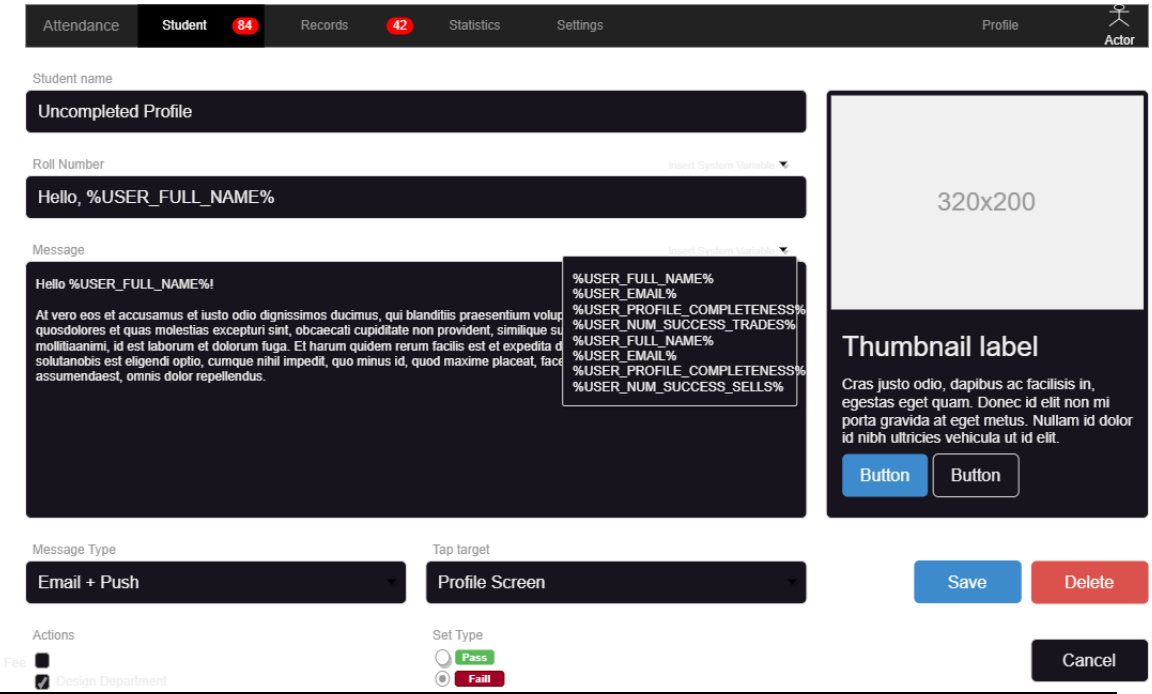
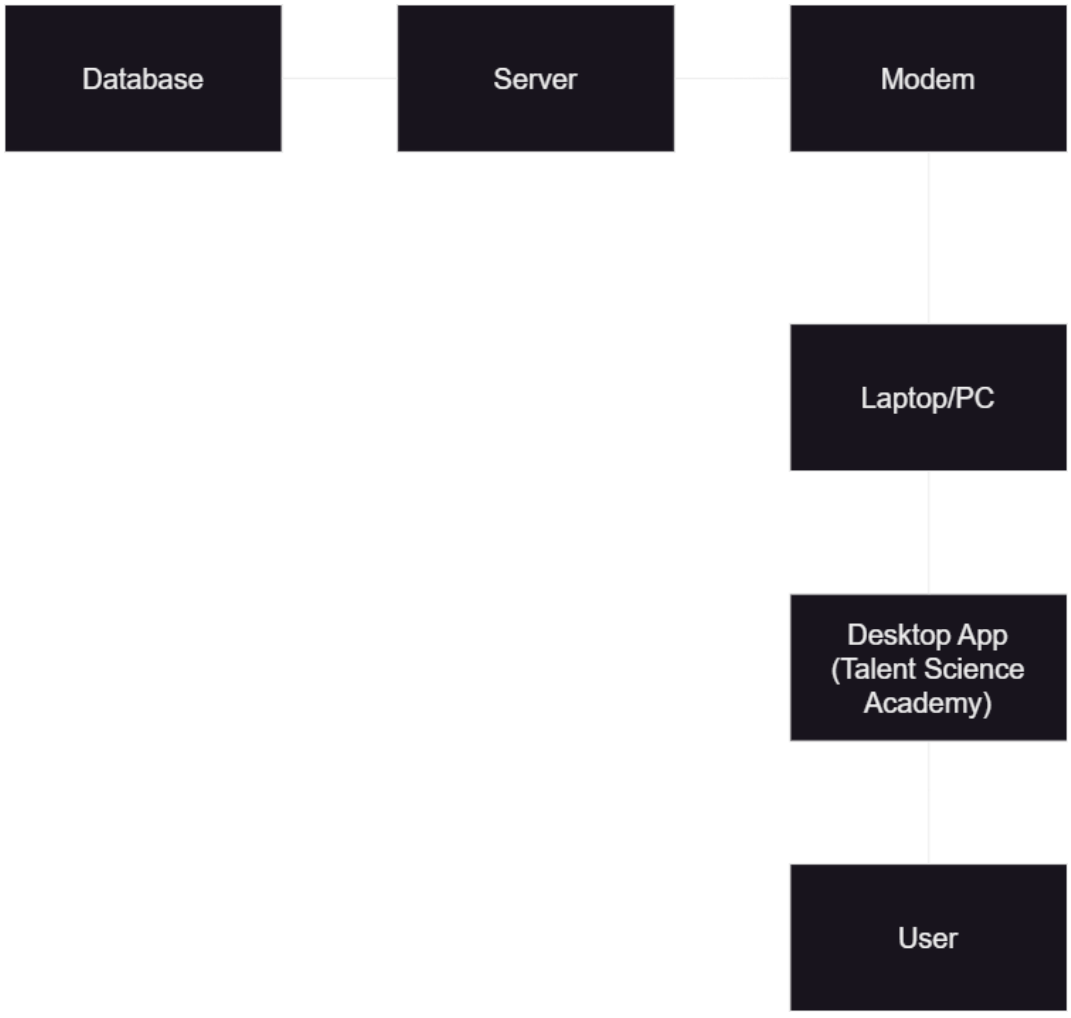
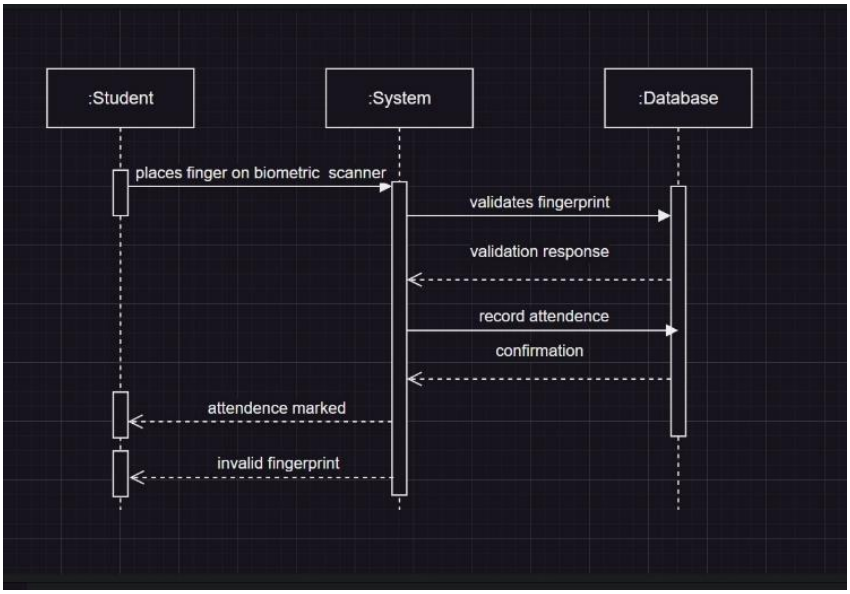


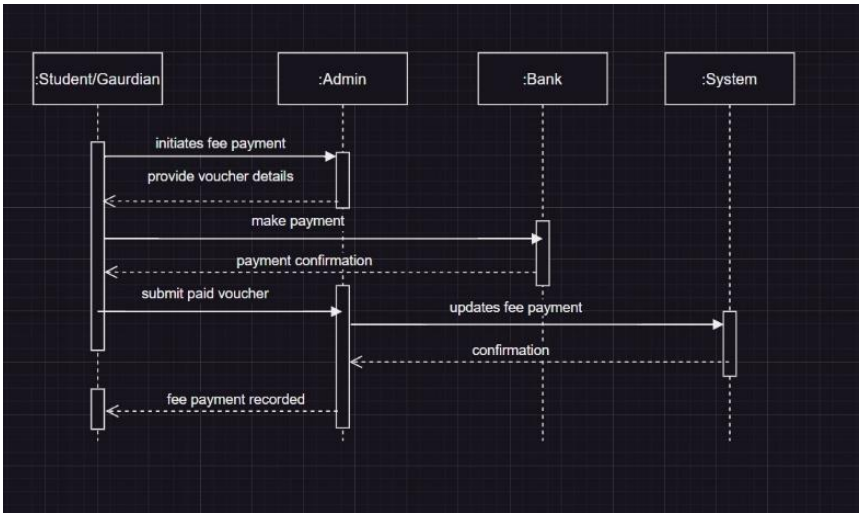
Figure 1.2

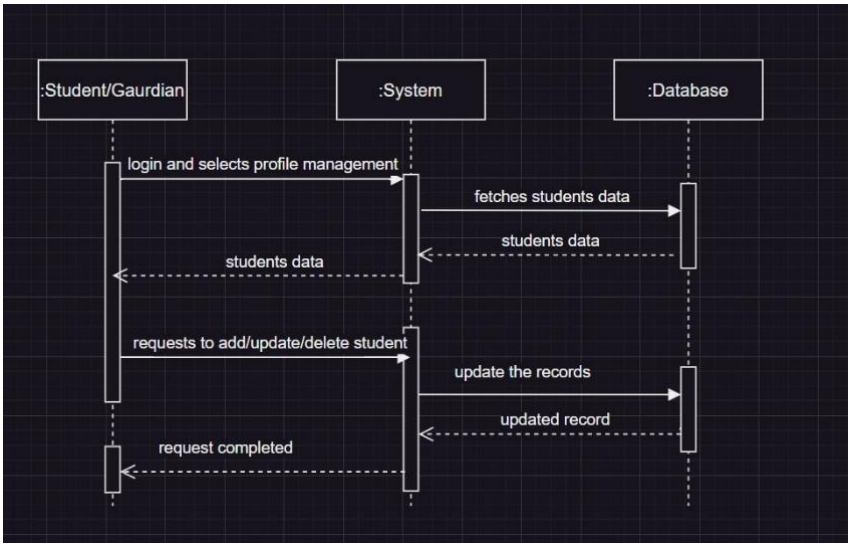
VI. Deployment Diagram:



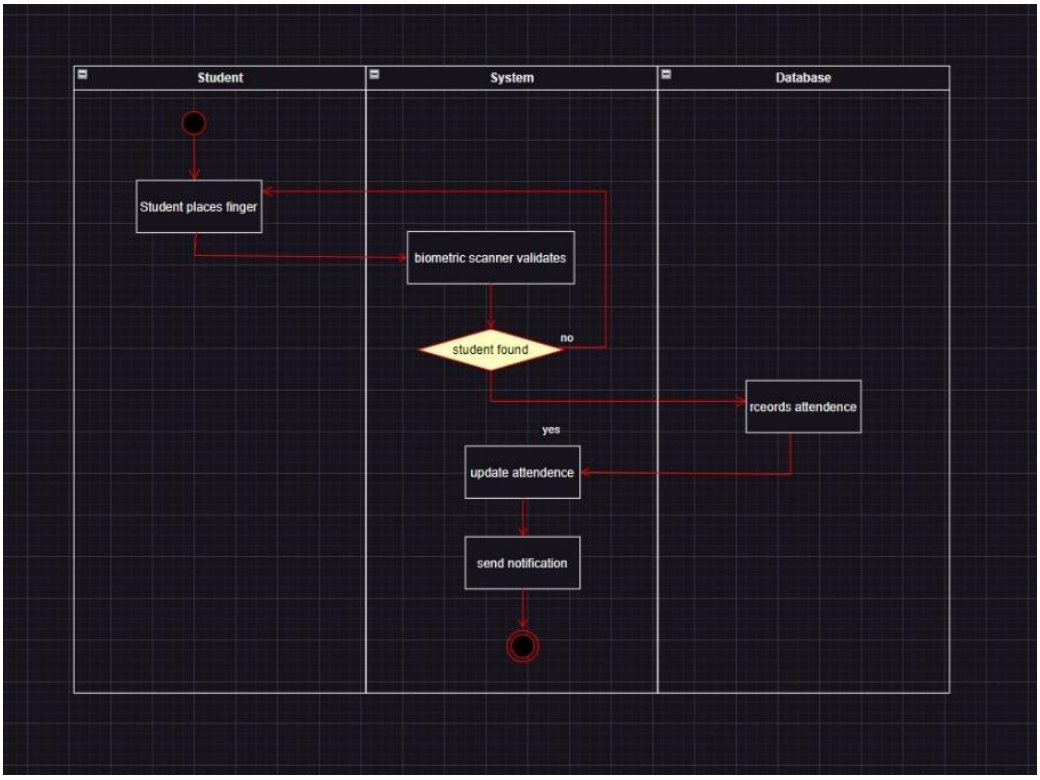
VII. Sequence Diagram

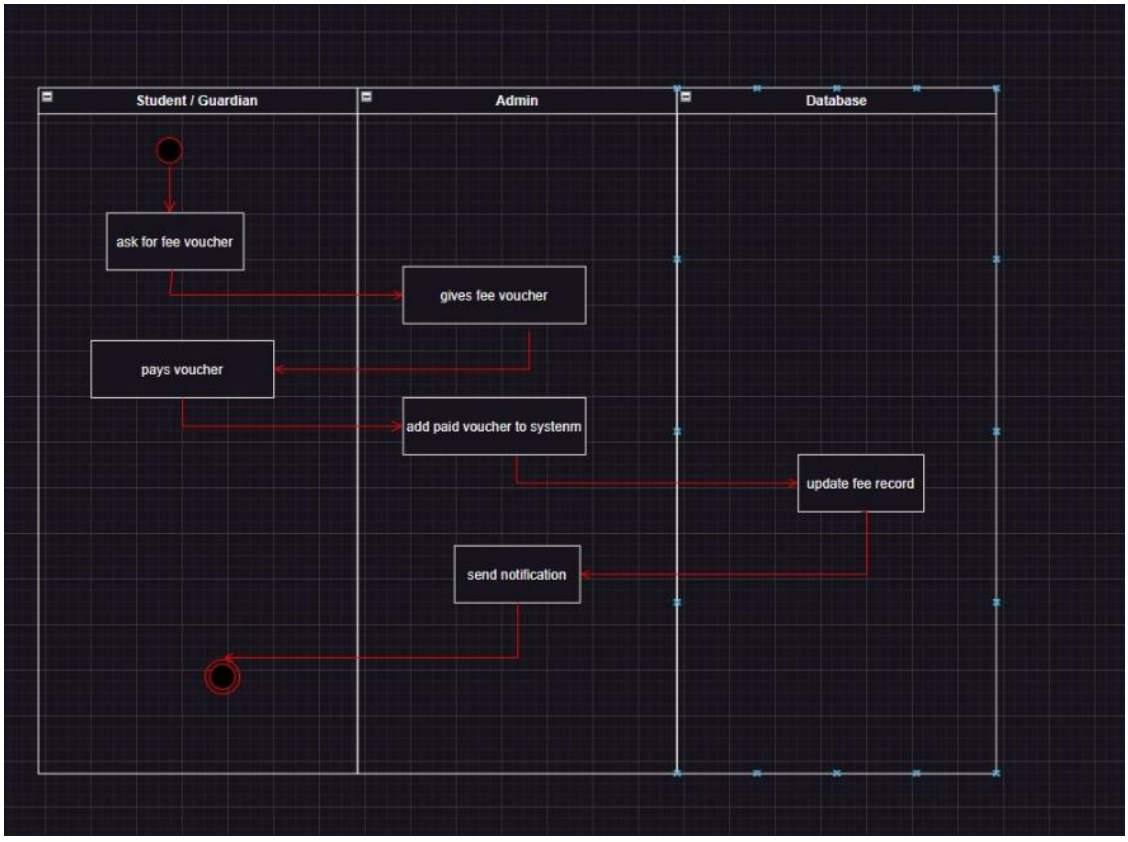


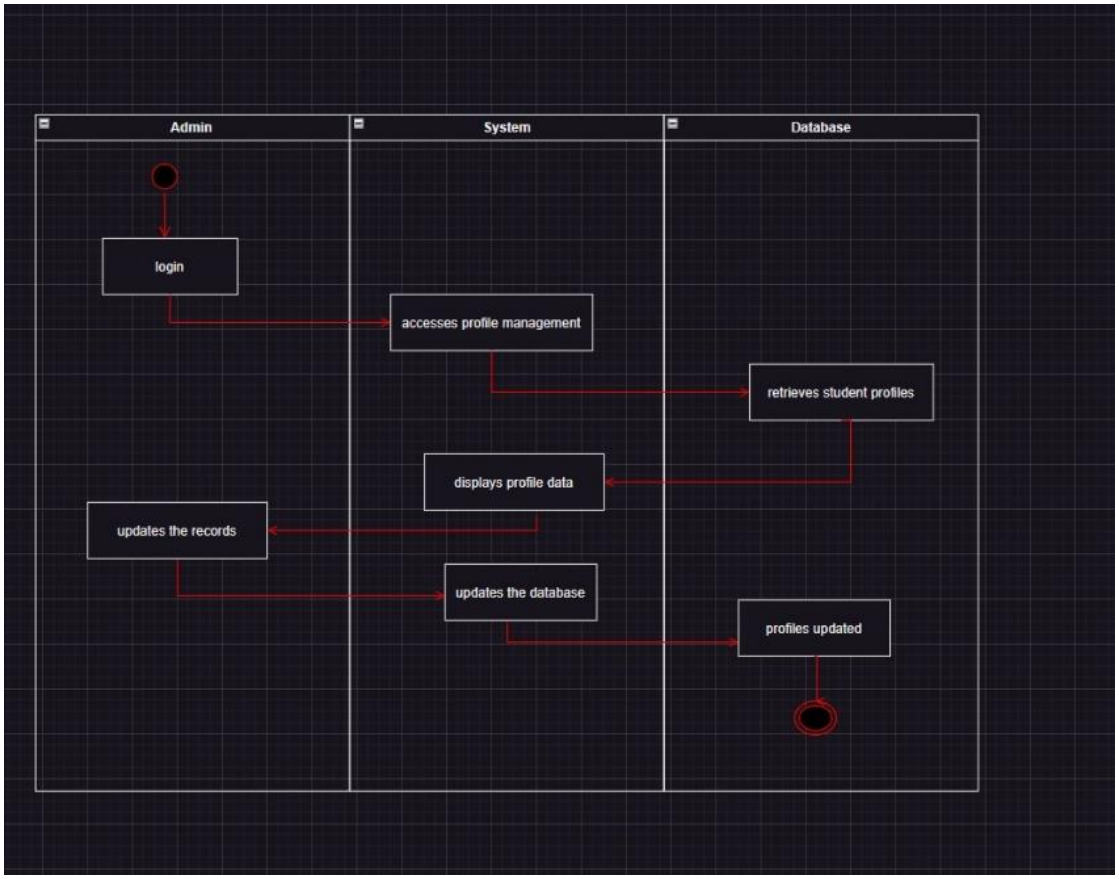




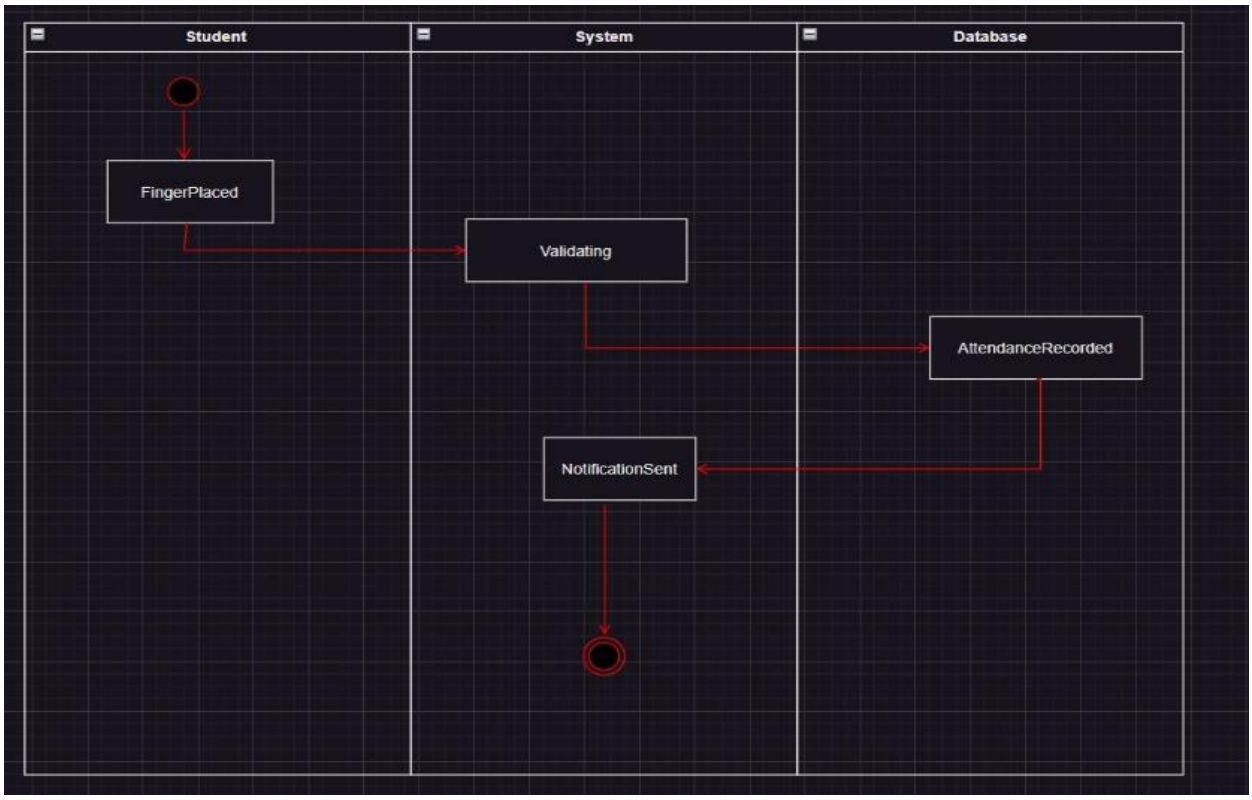
VIII. Activity Diagram

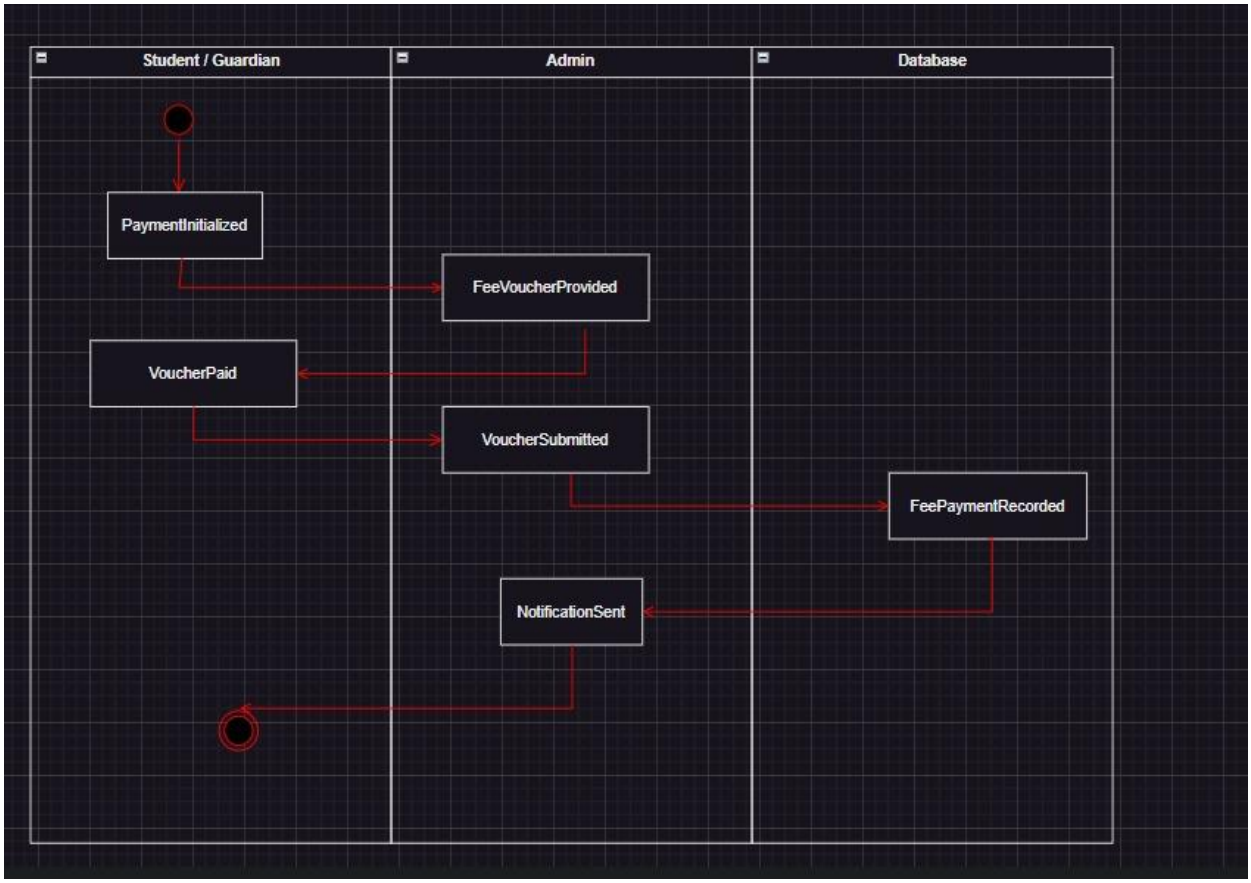


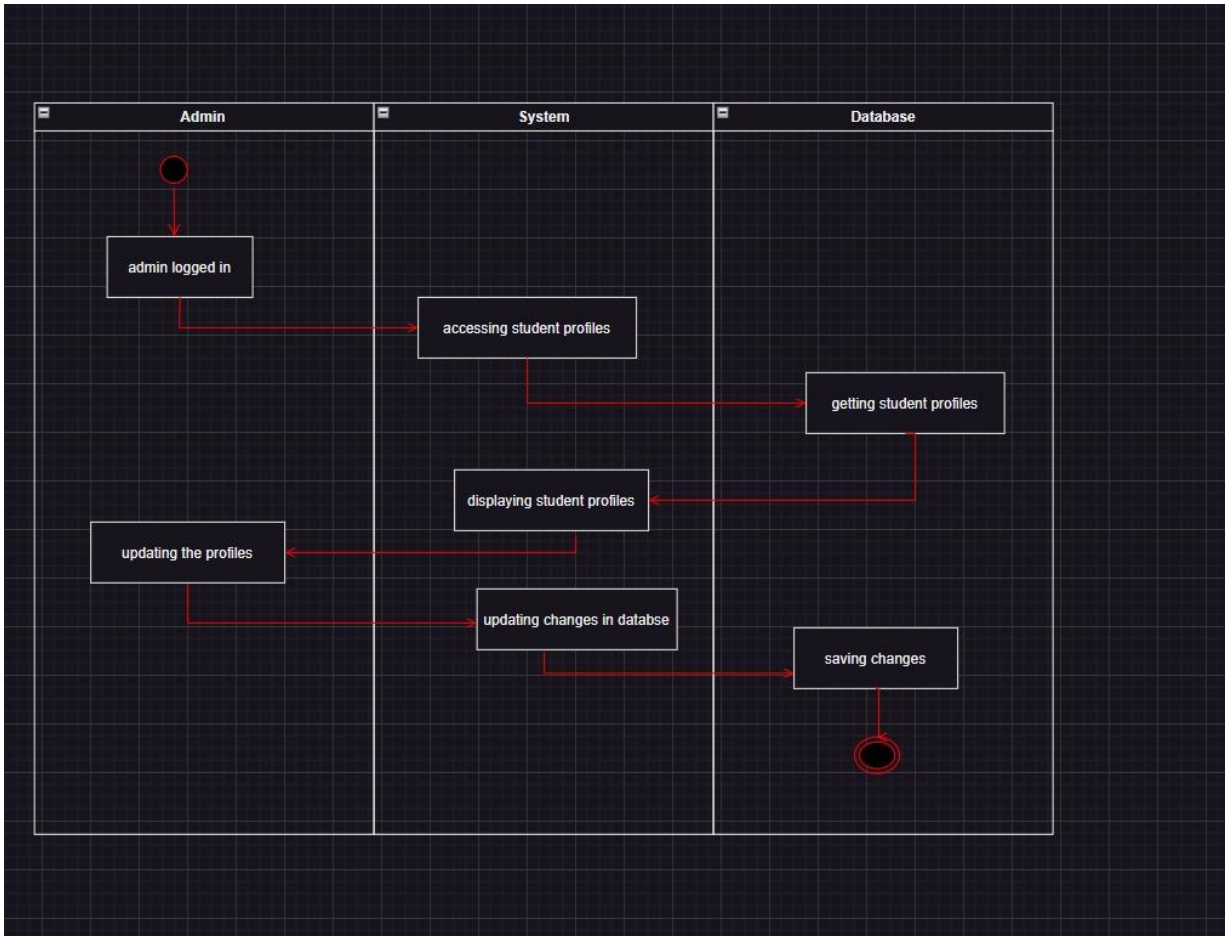




IX. State Diagram




Student / Guardian
Admin
Database
PaymentInitialized
FeeVoucherProvided
VoucherPaid
VoucherSubmitted
FeePaymentRecorded
NotificationSent


Admin
System
Database
admin logged in
accessing student profiles
getting student profiles
displaying student profiles
updating the profiles
updating changes in databse
saving changes

5. Milestones, Goals & Deliverables:

I. Milestones:

Milestones represent significant achievements or stages in the development process.

1. **Project Kickoff**
   a. Define project scope, goals, and requirements.
   b. Formulate a project plan and timeline.
   c. Assemble the project team.
2. **Requirements Specification Complete:**
   a. Finalize and document functional and non-functional requirements.
   b. Ensure alignment with stakeholder expectations.
3. **UI/UX Design Approval:**
   a. Present and obtain approval for the user interface and user experience design.
   b. Ensure the design aligns with user expectations and usability standards.
4. **Backend Development Complete:**
   a. Develop the backend components for message handling, user profiles, and profile history.
   b. Implement secure user authentication mechanisms.
5. **Frontend Development Complete:**
   a. Develop the user interface components for student management, profile creation, and profile updates.
   b. Ensure a responsive and user-friendly front end.
6. **Initial Testing and Bug Fixing:**
   a. Conduct initial testing to identify and fix bugs.
   b. Ensure basic functionality is stable.
7. **Student Management Completed:**
   a. Implement Student management and updating features.
   b. Enable attendance mechanism.
8. **Messaging Features Completed:**
   a. Implement Score cards functionality, including sending and receiving result notifications.
   b. Ensure real-time push notification capabilities.
9. **Performance Optimization:**
   a. Optimize app performance for responsiveness and scalability.
   b. Address any performance bottlenecks.

These milestones represent a logical progression in the development and release of an academic management system with the specified features.

Each milestone signifies a crucial stage in the development process, and their successful completion contributes to the overall success of the project.

II. Goals:

The goals of an academic management system with features like sending notifications, managing students, and preserving their results typically revolve around providing a seamless and secure communication experience for the user. Following are the goals of academic management system:

a. Analytics & Reporting
b. Real-Time Data Updates
c. Automated Attendance Tracking
d. User-friendly Interface
e. Result record maintenance
f. Push Notifications for Attendance alerts
g. Scalability and Performance

All these objectives work together to produce a feature-rich academic administration system that improves automation of attendance, offers a satisfying user experience, and protects the privacy and security of user data. The system's overall success and uptake are influenced by the accomplishment of these objectives.

III. Project Deliverables:

Deliverables for this system with the functionalities (managing student records, automated attendance, sending notifications) often span several software development stages and comprise a variety of artefacts.

a. Project Proposal
b. Requirements Specification
c. System Architecture Design
d. Database Schema
e. User Documentation

These goals collectively aim to create a feature-rich academic management system that enhances Attendance automation, provides a positive user experience, and ensures the security and privacy of user data. The successful achievement of these goals contributes to the overall success and adoption of the system.

6. Gantt Chart:

Steps	First Week	Second Week	Third Week	Fourth Week	Fifth Week	Sixth Week	Seventh Week	Eighth Week
Project Description								
Problem Statement and Objective								
Requirement								
Class Diagram								
ER Diagram								
3NF DB								
UML Diagrams								
Documentation								