FACIAL RECOGNITION ATTENDANCE SYSTEM (FRAMS)

A MINI PROJECT REPORT

Submitted By

1.	**JAYAKANTH S**	**220701101**
2.	**JOHN ALLAN J**	**220701111**
3.	**KAMAL J R**	**220701117**
4.	**KEERTHANA M G**	**220701123**
5.	**KESAVAN M**	**220701129**
6.	**BALAMURUGAN M**	**220701516**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM

CHENNAI-602105

2023 - 24

Table of Contents

1. OVERVIEW:

The Facial Recognition Attendance Management System (FRAMS) is an automated solution for colleges to streamline attendance tracking for students, staff, and drivers. Utilizing advanced facial recognition technology, users can log their attendance daily, ensuring accuracy and convenience. Each user has a personalized account to access daily logs, monthly attendance reports, and attendance percentages. An integrated admin dashboard centralizes all data, providing comprehensive attendance records and generating monthly payroll reports based on attendance for staff and drivers. Users can also indicate On Duty status, which is recorded without marking them absent. The system features secure authentication, data encryption, and a scalable client-server architecture. Key components include user management, attendance tracking, reporting, and notifications. Notifications keep both users and admins informed about attendance and payroll. FRAMS offers a robust and efficient approach to attendance management, enhancing operational efficiency and data accuracy within the college.

1.1. PROBLEM STATEMENTS

1. **High Administrative Costs:** Manual attendance tracking and payroll processing are labor-intensive and costly, consuming significant resources.
2. **Inconvenient Attendance Processes:** Traditional methods, such as paper-based systems or manual entry, are error-prone and inefficient for users and administrators.
3. **Accuracy and Reliability Issues:** Existing systems may lack accuracy, leading to incorrect attendance records and payroll errors.
4. **User Experience Challenges:** Many systems are not user-friendly, causing frustration when accessing and reviewing attendance data.
5. **Security and Privacy Concerns:** Ensuring the security and privacy of sensitive user data, including biometric information, is challenging and requires robust encryption and authentication.

2. BUSINESS ARCHITECTURE:

2.1. BUSINESS OBJECTIVES:

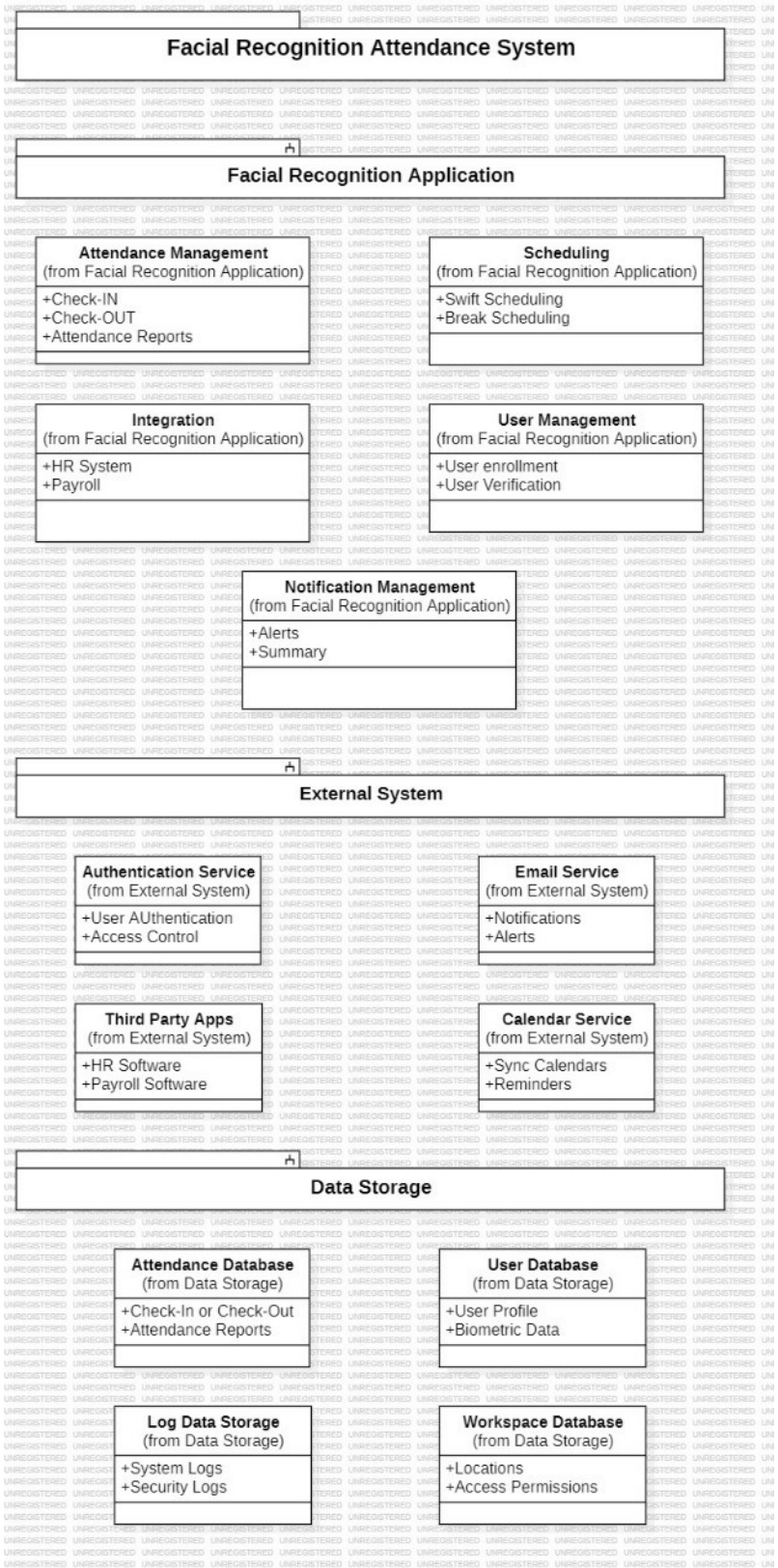
- **Efficiency:** Improve operational efficiency by automating the attendance tracking process.
- **Accuracy:** Enhance accuracy in attendance records, reducing errors associated with manual entry.
- **Security:** Increase security by ensuring only authorized individuals gain access to the premises.
- **Real-Time Monitoring:** Enable real-time monitoring of employee attendance and punctuality.
- **Cost-Effective:** Reduce administrative costs associated with manual attendance management.
- **Data Analysis:** Provide valuable data for analysis of attendance trends, employee punctuality, etc.
- **Integration:** Seamlessly integrate with other systems like payroll, HRMS for streamlined operations.
- **Compliance:** Ensure compliance with labor laws and regulations regarding work hours and breaks.
- **Scalability:** Offer a scalable solution that can grow with the organization's needs.
- **User-Friendly:** Implement a user-friendly system that is easy for employees to use and administrators to manage.

2.2. MARKET ANALYSIS:

MARKET ANALYSIS

EXISTING COMPANY	SYSTEM PROVIDER	IMPLEMENTATION BENEFITS	TARGET AUDIENCE	POTENTIAL IMPROVEMENTS
amazon	Amazon's proprietary system	Improved security, streamlined employee management	Warehouse staff Office employees Security personnel	Enhanced data privacy controls, faster processing speed
DELTA	Enhanced boarding process, reduced wait times	Enhanced boarding process, reduced wait times	Airport staff Passengers Security personnel	Better integration with other biometric systems, improved accuracy
Microsoft	Microsoft Azure Face API	Seamless integration with existing systems, better data analytics	Office employees IT staff Security personnel	Improved data storage solutions, more robust security features

2.3. BUSINESS ARCHITECTURE DIAGRAM:



2.4. TARGET MARKET:

- **Students:** Students are the primary users of the system. It can help track their attendance accurately and efficiently, reducing the chances of proxy attendance.
- **Faculty Members:** Professors, lecturers, and other academic staff can benefit from an automated system that saves time and effort in taking attendance manually.
- **Administrative Staff**: The system can assist administrative staff in maintaining accurate attendance records, which can be crucial for reporting and decision-making.
- **Parents:** In schools, parents can be considered a target audience as they would be interested in the accurate tracking of their child's attendance.
- **IT Department:** The IT department would be involved in the implementation and maintenance of the system, making them a key target audience.

2.5. IMPLEMENTATION:

To implement the Facial Recognition Attendance Management System, we will create a web and mobile application for user interaction. The system will handle user registration, daily check-ins and check-outs via facial recognition, and real-time attendance tracking. An admin dashboard will manage all user data and generate attendance and payroll reports. The database will store user information, attendance logs, and payroll details. Automated payroll calculations and on-duty reporting features will be integrated. The deployment will include setting up a secure, scalable, and reliable environment with monitoring and regular updates to ensure optimal performance and user satisfaction.

2.6. RISK ANALYSIS:

- **Scalability and Performance:**
  Risk: Slow response times with user growth.
  Mitigation: Optimize queries and consider load balancing.

- **Security Vulnerabilities:**
  Risk: Unauthorized access or data breaches.
  Mitigation: Follow secure coding practices and conduct audits.

- **Integration Challenges:**

  Risk: Compatibility issues during system integration.

  Mitigation: Plan for seamless integration and validate APIs.

- **Image Quality Dependency:**

  Risk: Poor image quality leading to false results.

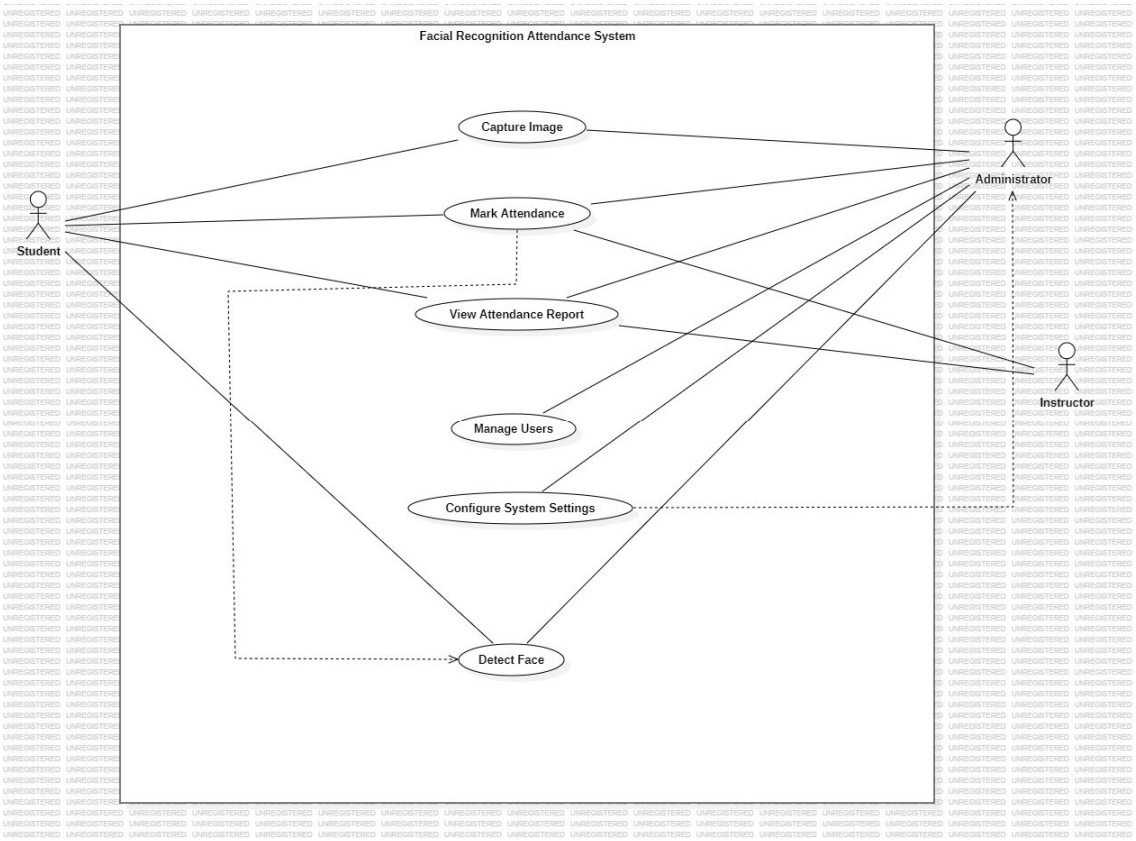
  Mitigation: Use high-quality cameras and preprocess images.

- **Error-Prone Recognition:**

  Risk: Misidentification due to lighting, pose, or expressions.

  Mitigation: Regularly update and fine-tune the recognition model.

2.7. USE CASE DIAGRAM:



3. REQUIREMENTS:

3.1. USER STORIES:

1. **As a system, I should be able to accurately recognize registered faces so that I can identify the person.** [Poker Planning Estimate: 13]

   Acceptance Criteria:

   - The system should recognize and match registered faces with at least 95% accuracy.
   - The system should identify a face within 2 seconds of scanning.

2. **As a system, I should be able to handle different lighting conditions and angles for face recognition so that I can ensure accuracy.** [Poker Planning Estimate: 8]

   Acceptance Criteria:

   - The system should recognize faces in varying lighting conditions (bright light, dim light, backlight).
   - The system should recognize faces from different angles (up to 30 degrees deviation from front-facing).

3. **As a system, I should be able to update the facial data of registered users so that I can improve recognition accuracy.** [Poker Planning Estimate: 5]

   Acceptance Criteria:

   - The system should allow updating of facial data for a user without requiring a complete re-registration.
   - Updated facial data should reflect in the system within 5 minutes of the update.

4. **As a system, I want to mark attendance when a registered face is recognized so that I can keep track of user presence.** [Poker Planning Estimate: 5]

   Acceptance Criteria:

   - Attendance should be marked automatically when a registered face is recognized.
   - The system should log the date and time of recognition accurately.

5. **As a system, I want to handle multiple entries and exits of the same user in a day so that I can accurately log all attendance events.** [Poker Planning Estimate: 8]

   Acceptance Criteria:

   - The system should log each entry and exit event with accurate timestamps.
   - The system should distinguish between multiple entries and exits for the same user.

6. **As a system, I want to provide a manual attendance correction option for administrators so that I can handle errors.** [Poker Planning Estimate: 5]

   Acceptance Criteria:

   - Administrators should have access to a manual correction interface.
   - Changes made by administrators should be logged with details (who made the change, what change was made, timestamp).

7. **As a system, I want to allow administrators to register new users into the system by capturing their facial data so that new users can be recognized.** [Poker Planning Estimate: 5]

   Acceptance Criteria:

   - The system should provide an interface for capturing facial data during registration.
   - The registration process should be completed within 5 minutes per user.

8. **As a system, I want to allow administrators to update or delete user profiles so that user information remains current.** [Poker Planning Estimate: 5]

   Acceptance Criteria:

   - Administrators should be able to update user details (name, role, etc.) and facial data.
   - Administrators should be able to delete user profiles, with a confirmation step to prevent accidental deletions.

9. **As a system, I want to generate daily, weekly, and monthly attendance reports for each user so that attendance can be monitored.** [Poker Planning Estimate: 8]

   Acceptance Criteria:

   - The system should provide options to generate daily, weekly, and monthly reports.
   - Reports should include user name, date, time of entry and exit, and total hours present.

10. **As a system, I want to provide analytics like attendance percentage and frequent latecomers so that administrators can review attendance patterns.** [Poker Planning Estimate: 8]

    Acceptance Criteria:

    - The system should calculate attendance percentage for each user.

11. **As a system, I want to securely store facial data and attendance records so that I can protect user privacy.** [Poker Planning Estimate: 8]

    Acceptance Criteria:

    - Facial data and attendance records should be encrypted in storage and during transmission.
    - Access to data should be restricted based on user roles and permissions.

12. **As a system, I want to have measures in place to prevent unauthorized access to the system and data so that user data remains secure.** [Poker Planning Estimate: 8]

    Acceptance Criteria:

    - The system should implement strong authentication mechanisms (e.g., multi-factor authentication) for accessing sensitive features.
    - User roles and permissions should be clearly defined and enforced.

3.2. EPICS:

1. Facial Recognition System

- Task 1.1.1: Develop an algorithm for face detection.
- Task 1.1.2: Integrate the face detection algorithm with the system.
- Task 1.2.1: Implement functionality to handle different lighting conditions.
- Task 1.3.1: Develop a feature to update facial data of registered users.

2. Attendance Management

- Task 2.1.1: Develop a feature to mark attendance when a face is recognized.
- Task 2.2.1: Implement functionality to handle multiple entries and exits.
- Task 2.3.1: Develop a manual attendance correction feature for administrators.

3. User Registration

- Task 3.1.1: Develop a feature for administrators to register new users.
- Task 3.2.1: Implement functionality for administrators to update or delete user profiles.

4. Reporting and Analytics

- Task 4.1.1: Develop a feature to generate daily, weekly, and monthly attendance reports.
- Task 4.2.1: Implement analytics like attendance percentage, frequent latecomers, etc.

5. System Security and Privacy

- Task 5.1.1: Implement secure storage for facial data and attendance records.
- Task 5.2.1: Develop measures to prevent unauthorized access to the system and data.

6. Notification and Alerts

- Task 6.1.1: Develop a feature to send notifications to users when their attendance is marked.
- Task 6.2.1: Implement alerts for administrators when the same user is detected multiple times within a short period.

3.3. FEATURES:

1. Facial Recognition

- Accurate face detection and recognition
- Handling of different lighting conditions and angles
- Continuous learning and updating of facial data

2. Attendance Management

- Automatic marking of attendance when a registered face is recognized
- Handling of multiple entries and exits of the same user in a day
- Manual attendance correction option for administrators

3. User Registration

- Registration of new users by capturing their facial data
- Updating or deleting user profiles

4. Reporting and Analytics

- Generation of daily, weekly, and monthly attendance reports
- Analytics like attendance percentage, frequent latecomers, etc.

5. System Security and Privacy

- Secure storage of facial data and attendance records
- Measures to prevent unauthorized access to the system and data

3.4. FUNCTIONAL REQUIREMENTS:

- As a user (student/staff/driver), I want to create an account with my unique credentials (such as username, password, and facial biometric data) to access the system.

- As a user, I want to log in securely using my credentials or facial recognition to mark my attendance for the day.

- As a user, I want to have the option to mark myself as "On Duty" for a specific day instead of being marked absent.

- As an admin, I want to have access to a centralized dashboard to monitor attendance records of all users (students, staff, and drivers).

- As an admin, I want to generate monthly attendance reports for individual users to review their attendance percentage.

- As an admin, I want to calculate monthly payroll for staff and drivers based on their attendance records and salary rates.

- As a user, I want to receive notifications regarding attendance marking, monthly reports, and payroll updates.

- As an admin, I want to manage user accounts, including adding, deleting, or modifying user information.

3.5. NON-FUNCTIONAL REQUIREMENTS:

- **Performance:** The system should be capable of processing attendance records and generating reports within a maximum response time of 5 seconds, even under peak loads of up to 1000 simultaneous users.

- **Security:** The facial recognition algorithm should have a false acceptance rate (FAR) of less than 0.1% and a false rejection rate (FRR) of less than 1% to ensure accurate user identification.

- **Accuracy:** The facial recognition system should achieve a recognition accuracy rate of at least 99% to minimize the chances of false identifications or unauthorized access.

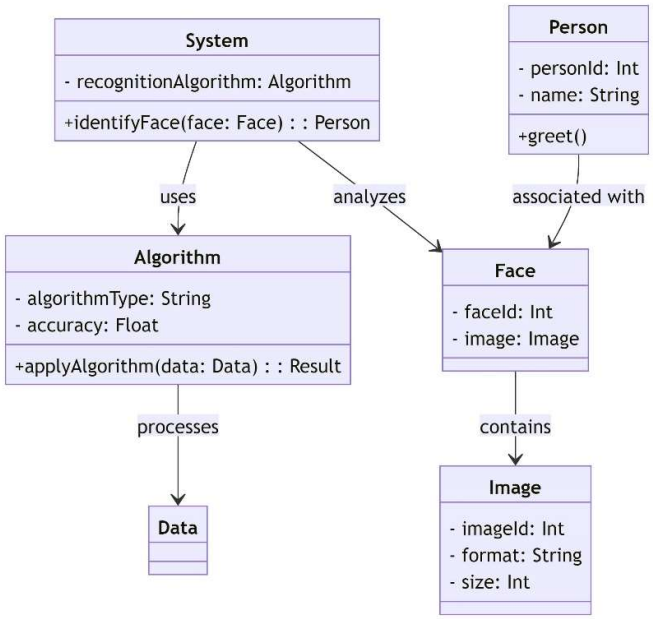
- **Reliability:** The system should have a minimum uptime of 99.9% per month, with automatic failover mechanisms in place to quickly restore service in case of unexpected downtime or server failures.

- **Scalability:** The system architecture should be designed to scale horizontally to accommodate a growing number of users, with provisions for load balancing and dynamic resource allocation to handle increased demand.

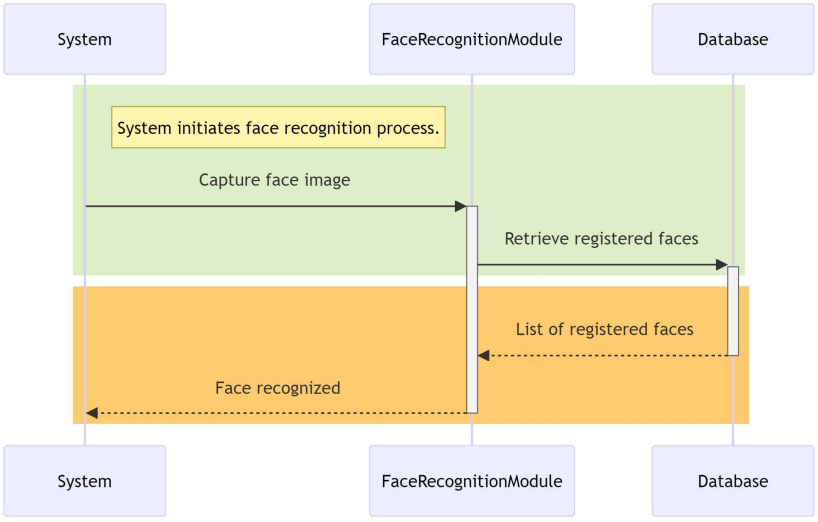
3.6. CLASS DIAGRAM & SEQUENCE DIAGRAM:

1. USER STORY: **As a system, I should be able to accurately recognize registered faces so that I can identify the person.**

CLASS DIAGRAM:

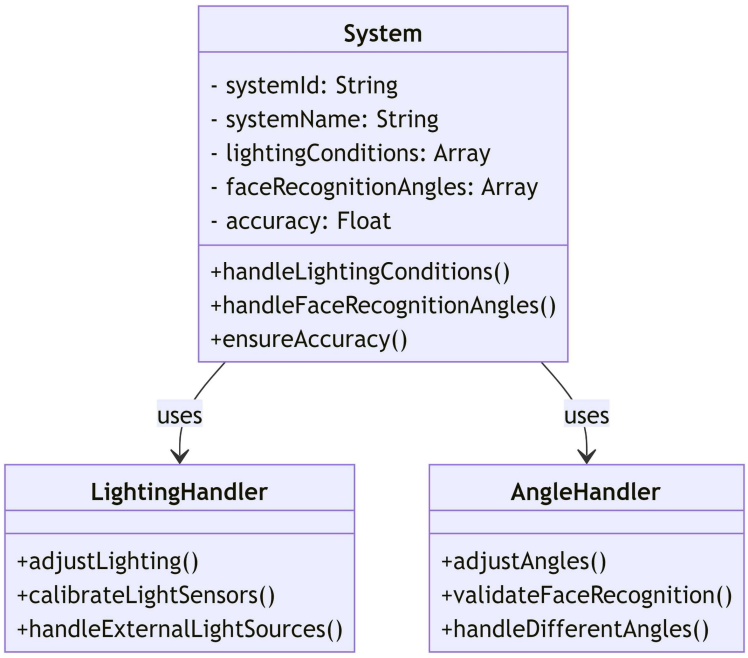


SEQUENCE DIAGRAM:

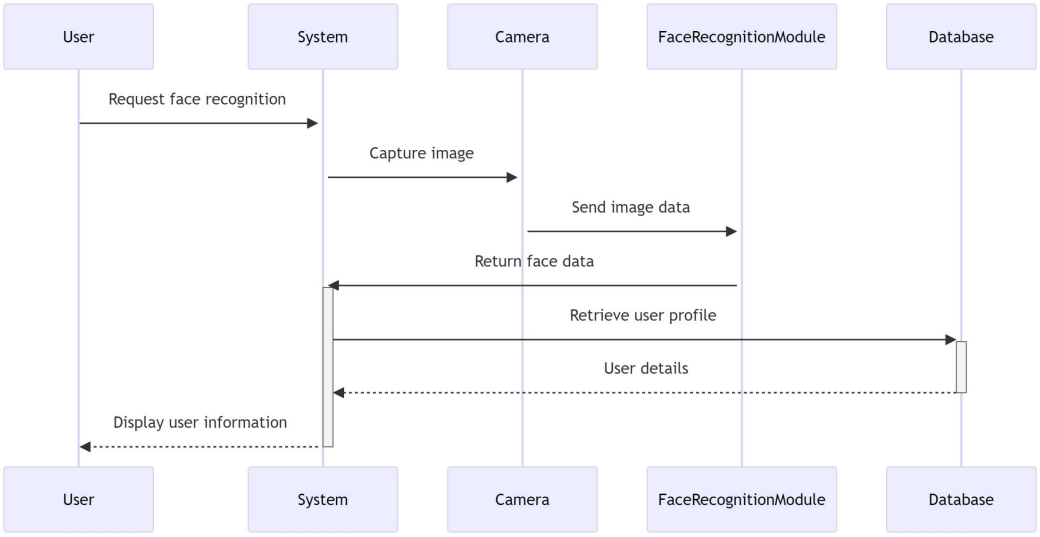


2. USER STORY: **As a system, I should be able to handle different lighting conditions and angles for face recognition so that I can ensure accuracy.**

CLASS DIAGRAM:

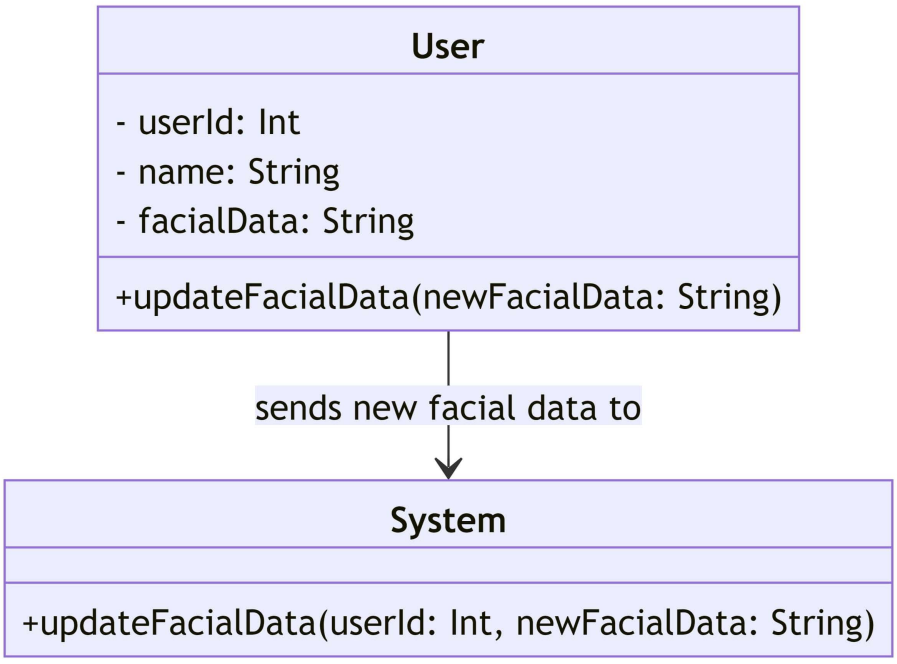


SEQUENCE DIAGRAM:

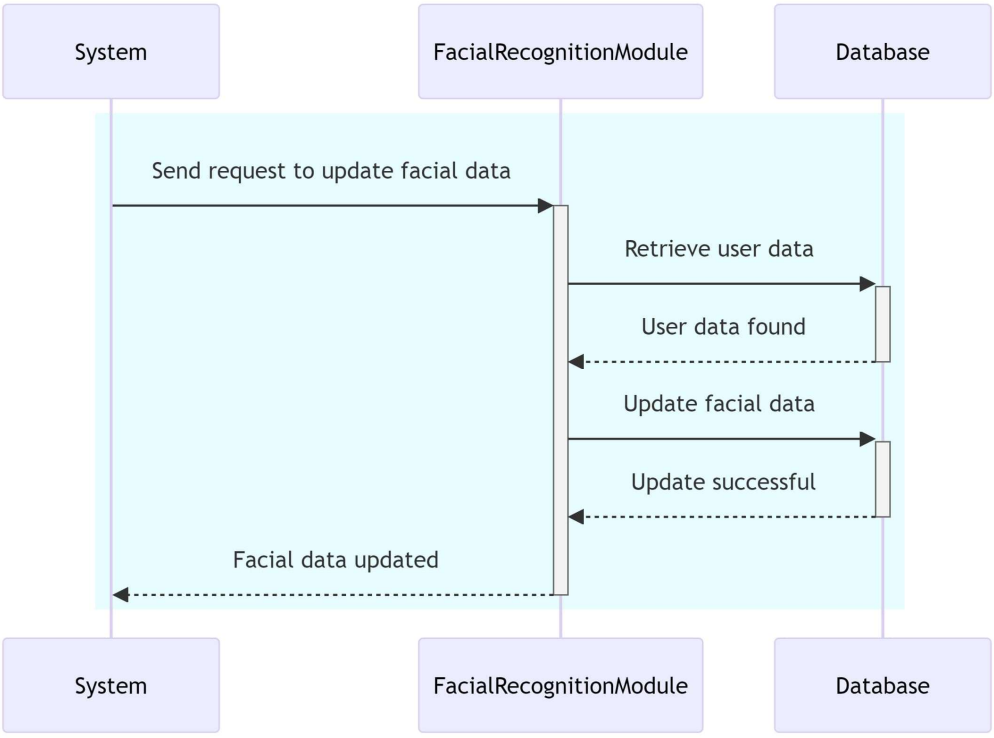


3. USER STORY: **As a system, I should be able to update the facial data of registered users so that I can improve recognition accuracy.**

CLASS DIAGRAM:

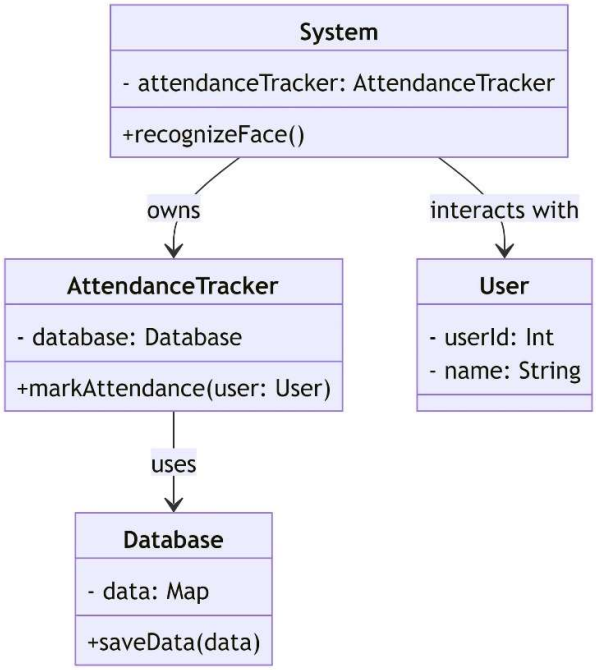


SEQUENCE DIAGRAM:

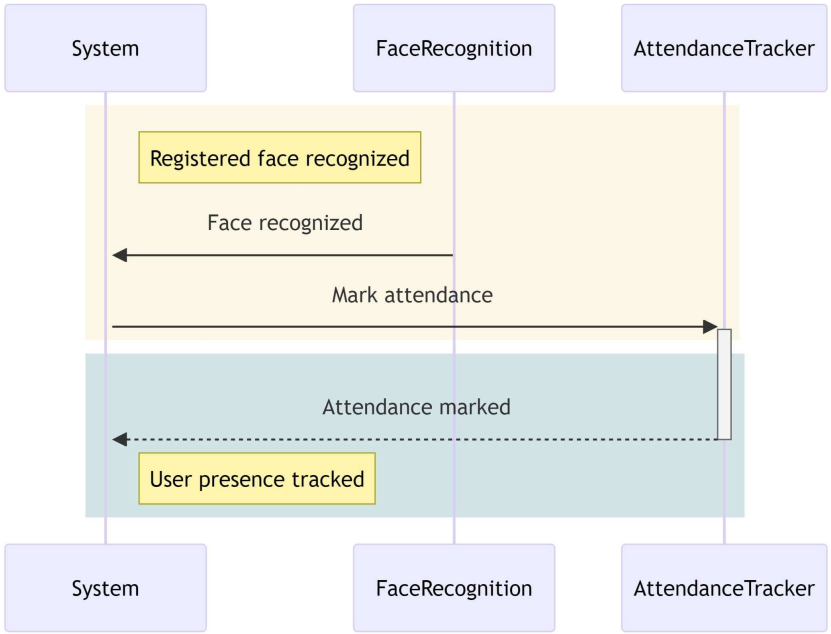


4. USER STORY: **As a system, I want to mark attendance when a registered face is recognized so that I can keep track of user presence.**

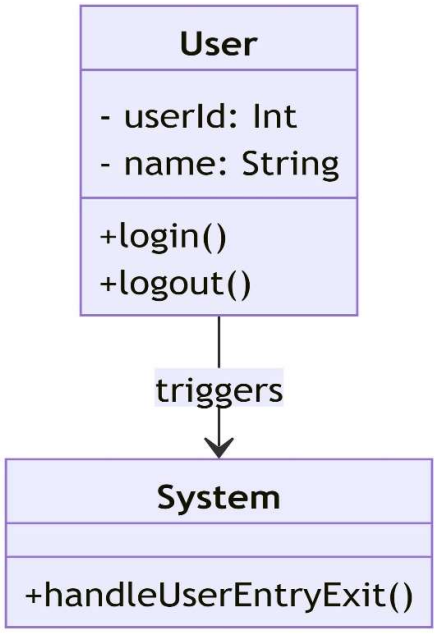
CLASS DIAGRAM:



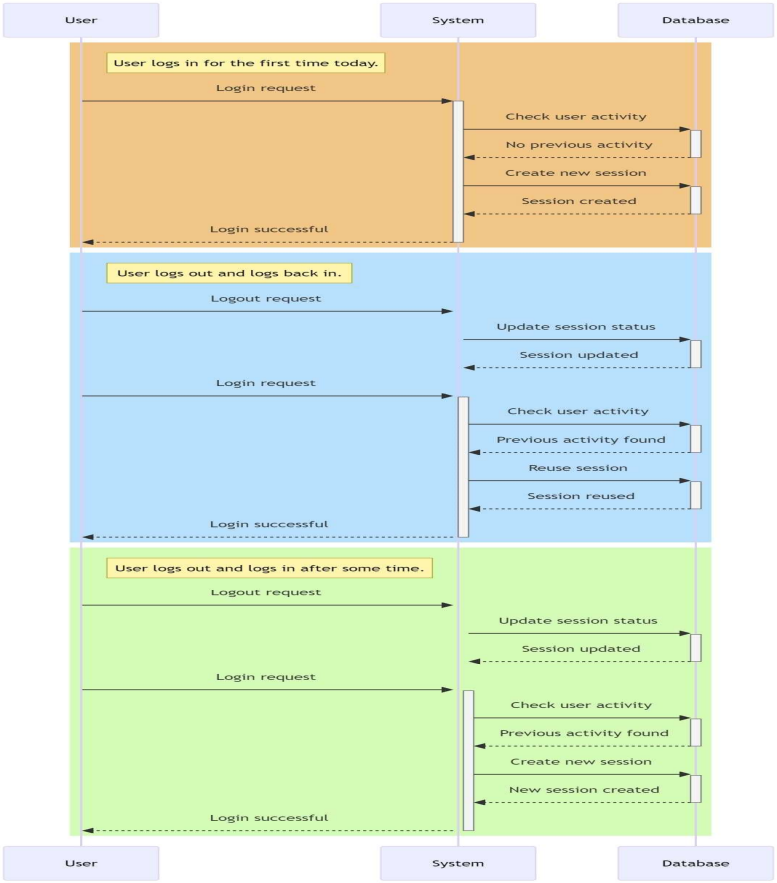
SEQUENCE DIAGRAM:



5. USER STORY: **As a system, I want to handle multiple entries and exits of the same user in a day so that I can accurately log all attendance events.**
CLASS DIAGRAM:

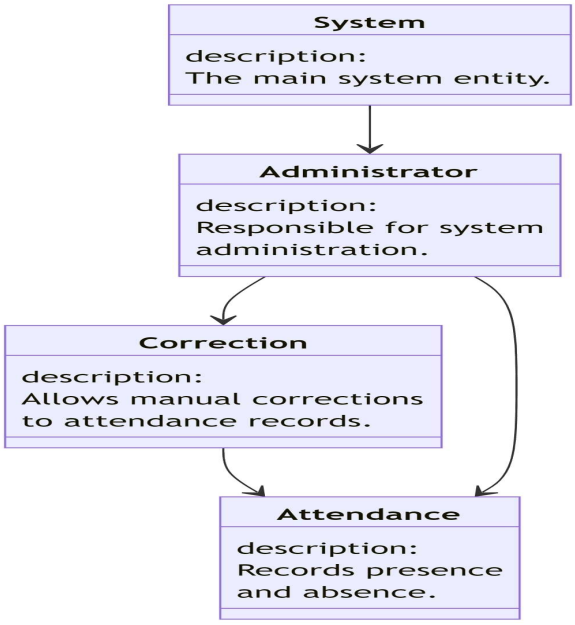


SEQUENCE DIAGRAM:

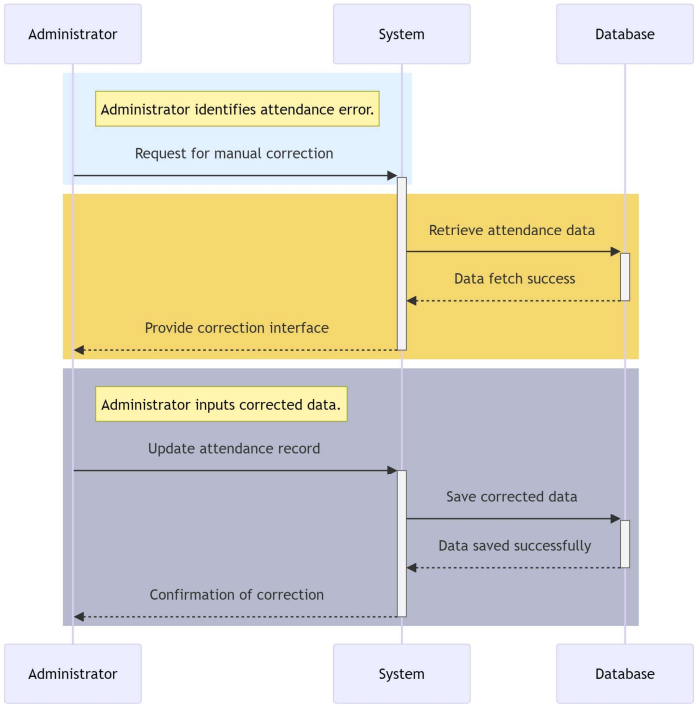


6. USER STORY: **As a system, I want to provide a manual attendance correction option for administrators so that I can handle errors.**

CLASS DIAGRAM:

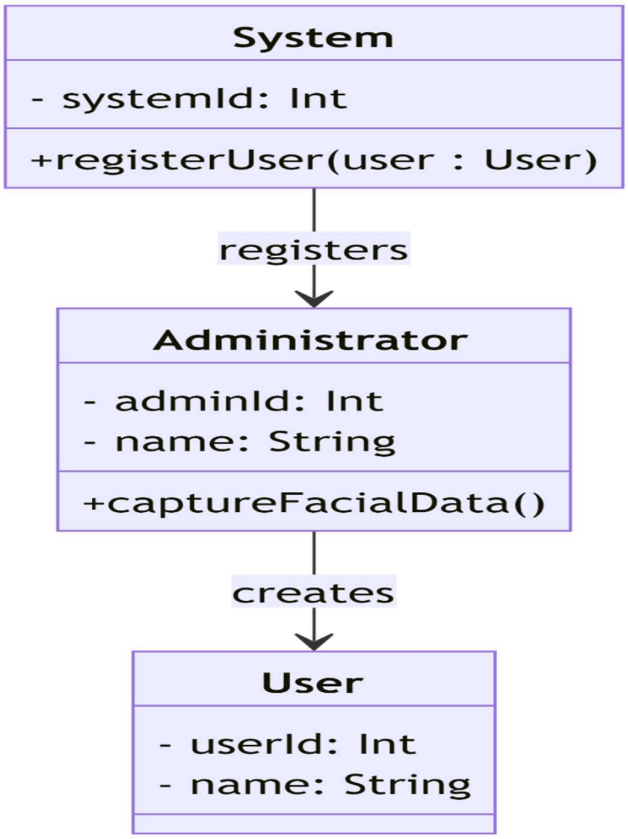


SEQUENCE DIAGRAM:

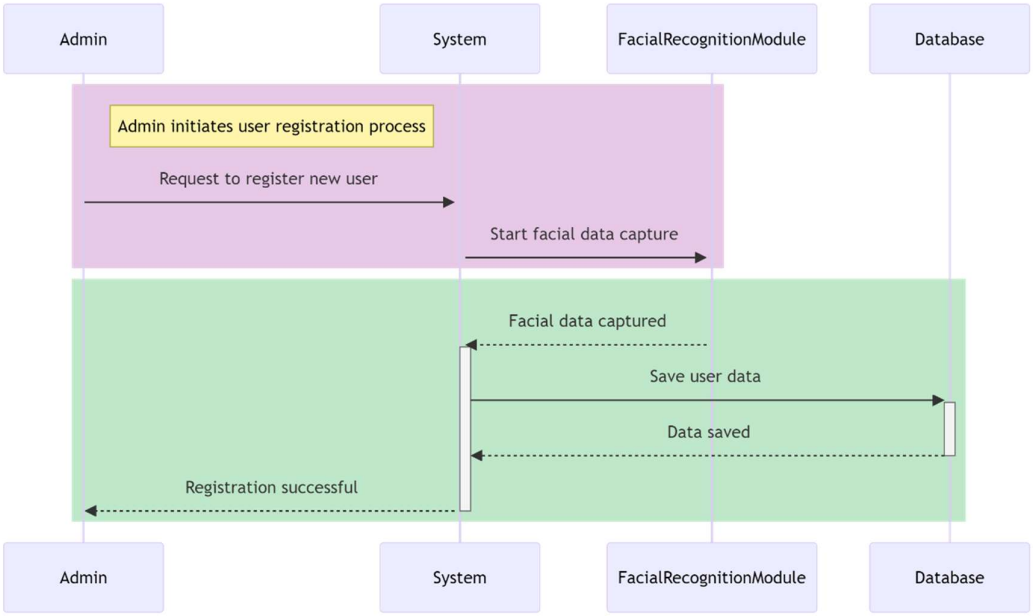


7. USER STORY: **As a system, I want to allow administrators to register new users into the system by capturing their facial data so that new users can be recognized.**

CLASS DIAGRAM:

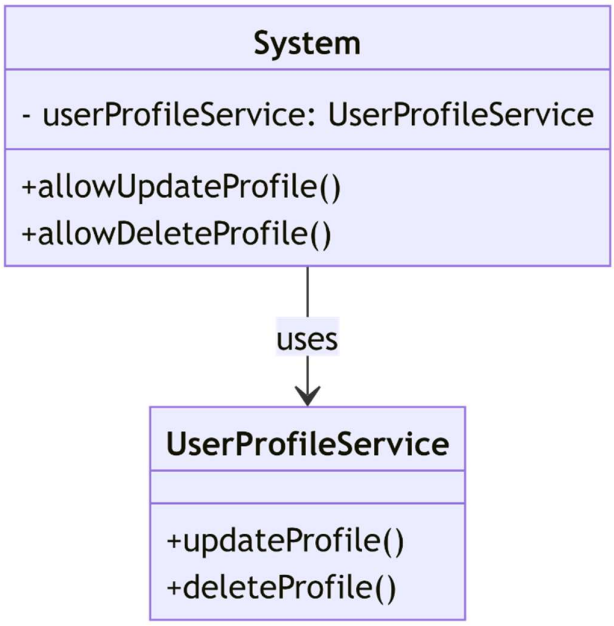


SEQUENCE DIAGRAM:

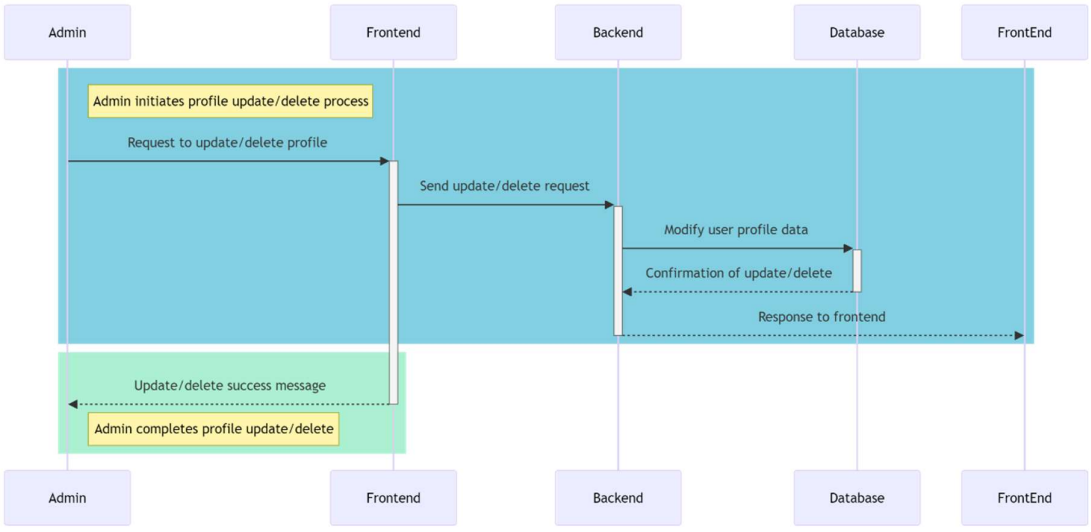


8. USER STORY: **As a system, I want to allow administrators to update or delete user profiles so that user information remains current.**

CLASS DIAGRAM:

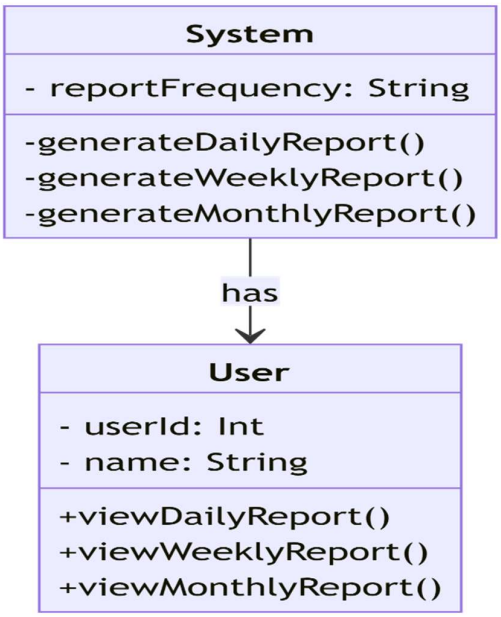


SEQUENCE DIAGRAM:



9. USER STORY: **As a system, I want to allow administrators to update or delete user profiles so that user information remains current.**

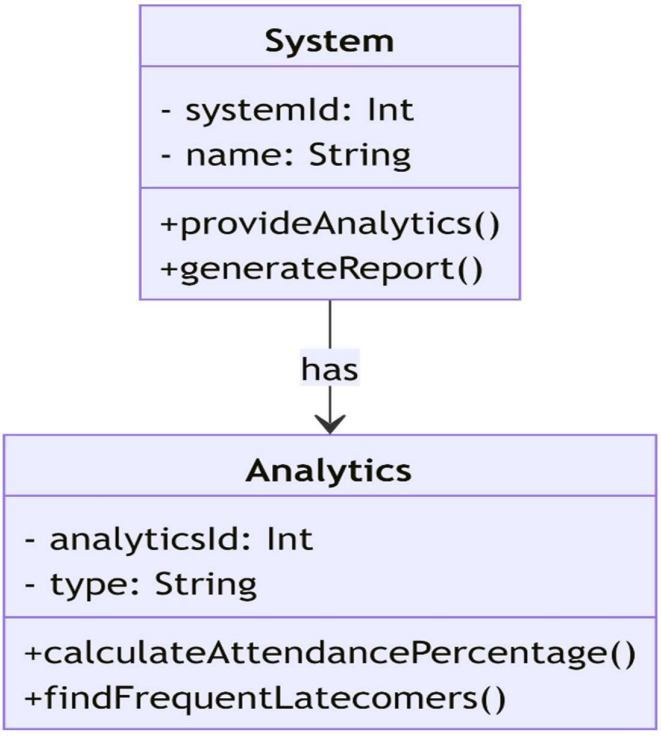
CLASS DIAGRAM:



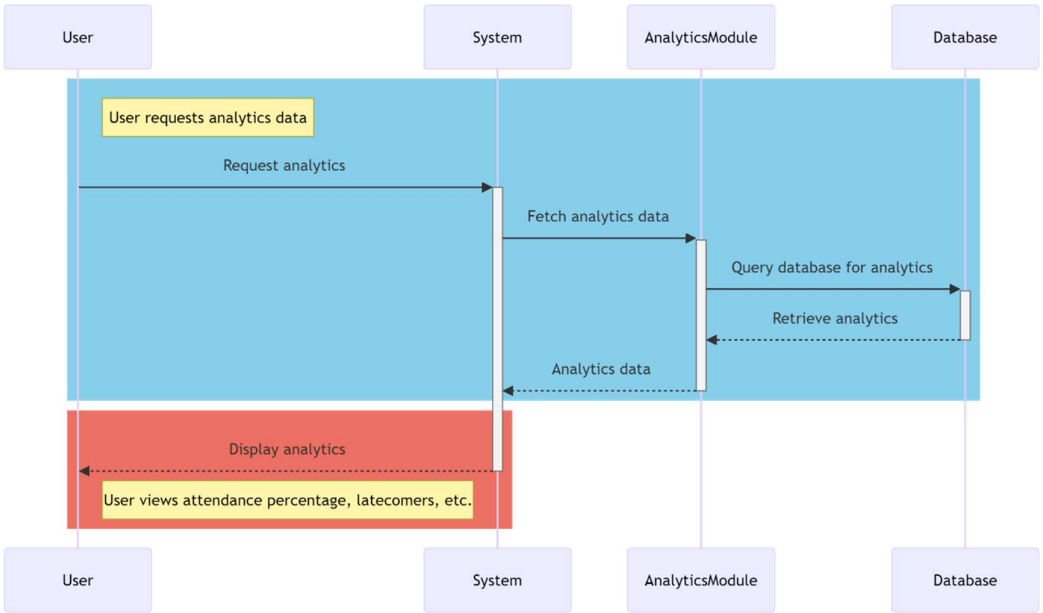
SEQUENCE DIAGRAM:

10. USER STORY: **As a system, I want to provide analytics like attendance percentage and frequent latecomers so that administrators can review attendance patterns.**

CLASS DIAGRAM:

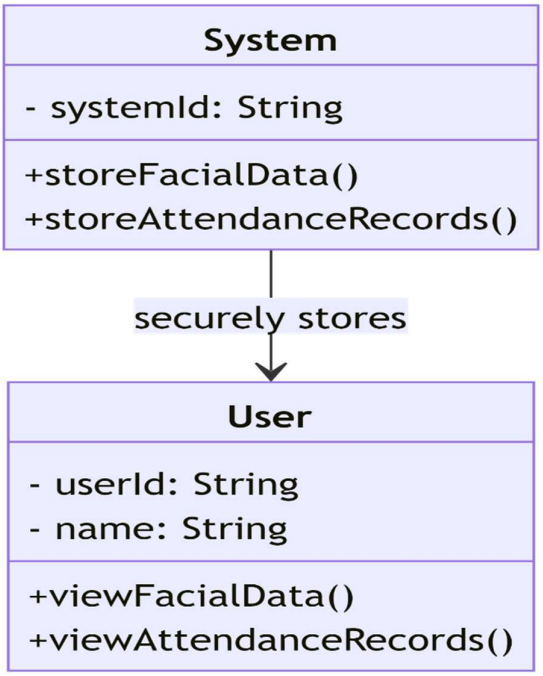


SEQUENCE DIAGRAM:

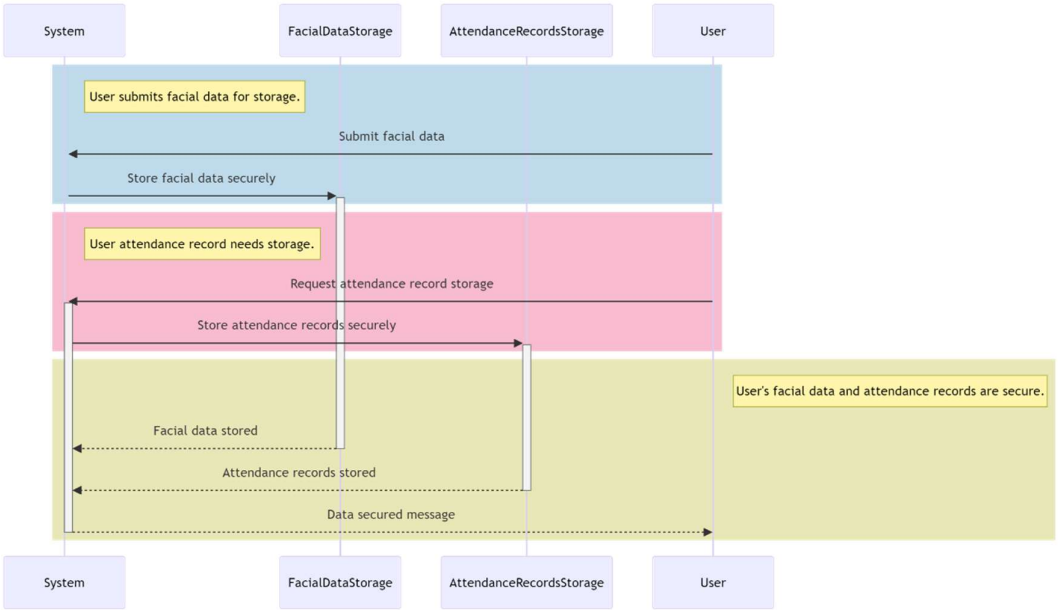


11. USER STORY: **As a system, I want to securely store facial data and attendance records so that I can protect user privacy.**

CLASS DIAGRAM:

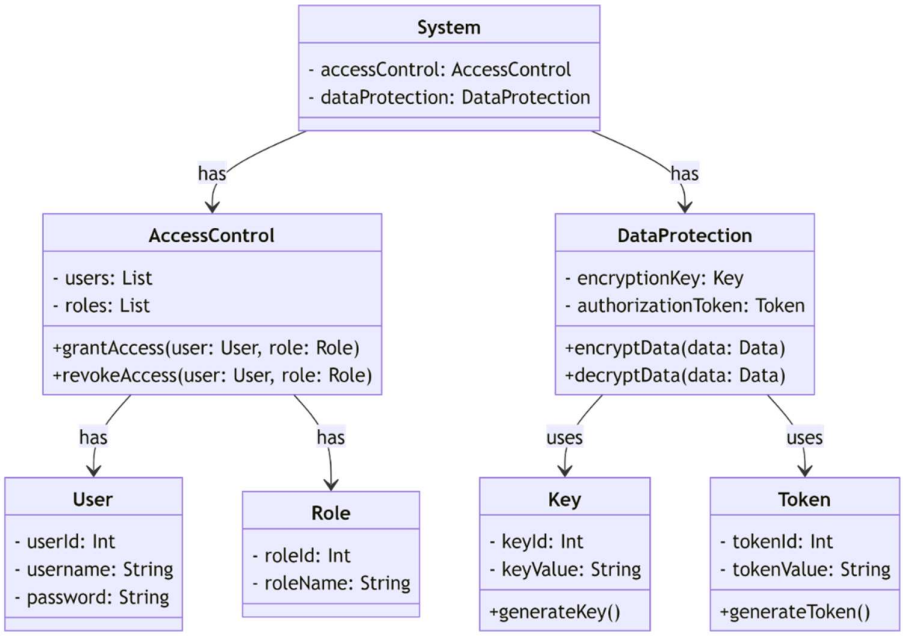


SEQUENCE DIAGRAM:

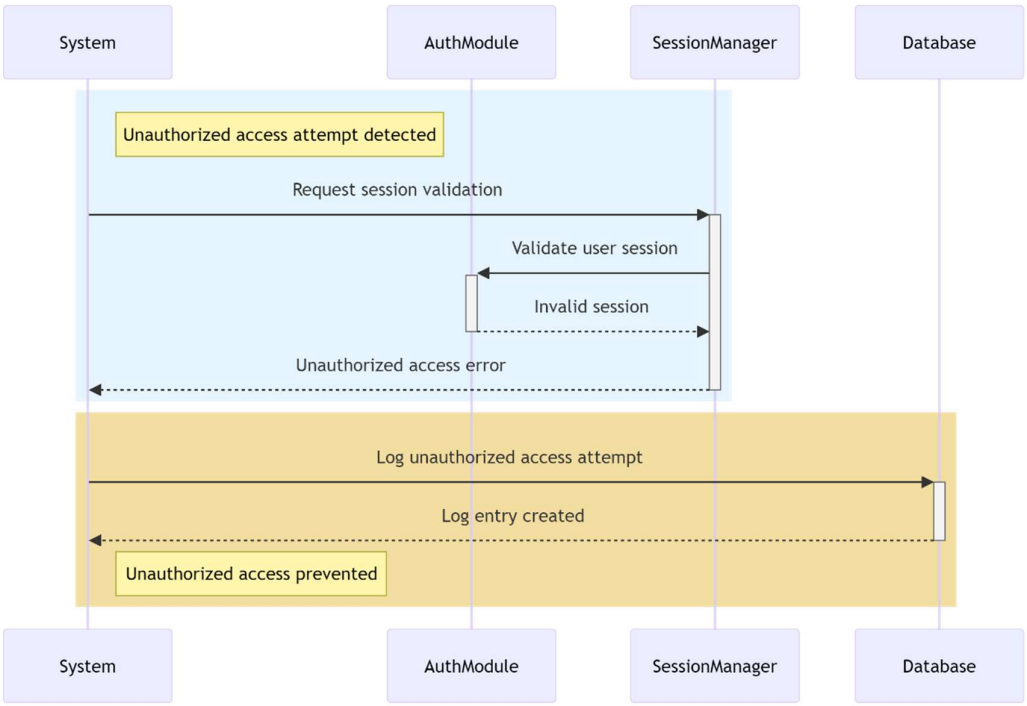


12. USER STORY: **As a system, I want to securely store facial data and attendance records so that I can protect user privacy.**

CLASS DIAGRAM:



SEQUENCE DIAGRAM:



4. ARCHITECTURE & DESIGN:

4.1. MVC ARCHITECTURE:

MVC (Model-View-Controller) architecture is a software design pattern widely used in web and application development. In MVC, the Model represents the data and business logic of the application, encapsulating data management and manipulation operations. The View is responsible for presenting the user interface to the users, displaying data and interacting with the user. The Controller acts as an intermediary between the Model and the View, handling user input, processing requests, and updating the Model accordingly. MVC promotes the separation of concerns, allowing for modular and maintainable code by isolating different aspects of the application logic.

4.1.1. MVC ARCHITECTURE IN FRAMS:

- **Model (M):**

  The Model represents the data and business logic of the application. In the context of FRAMS, the Model would handle tasks such as:

  - Managing user data (e.g., student information, staff details).
  - Storing attendance records and calculating attendance percentages.
  - Handling payroll data and processing salary calculations.

- **View (V):**

  The View represents the presentation layer of the application, responsible for rendering the user interface and displaying data to the users. In FRAMS, Views would handle tasks such as:

  - Displaying login screens for users to log in.
  - Showing attendance reports and payroll summaries in a user-friendly format.
  - Presenting notifications and system messages to users.

- **Controller (C):**

  The Controller acts as an intermediary between the Model and the View, handling user input, processing requests, and updating the Model accordingly. In FRAMS, Controllers would handle tasks such as:

  - Authenticating user credentials during login attempts.

- Processing requests to mark attendance or update user information.
- Generating payroll reports based on attendance data.

4.1.2. WHY MVC OVER OTHER TYPES?

- **Clear Separation of Concerns:**

  MVC ensures that each aspect of FRAMS, from user authentication to attendance management, is encapsulated within its respective component, promoting clarity and maintainability.

- **Scalability for Future Enhancements:**

  The modular structure of MVC allows FRAMS to easily accommodate future enhancements, such as new authentication methods or reporting features, without impacting the entire system.

- **Support for Different User Interfaces:**

  MVC's decoupling of the presentation layer (View) from the underlying logic (Model and Controller) enables FRAMS to have diverse user interfaces for various user roles, ensuring flexibility in UI design and implementation.

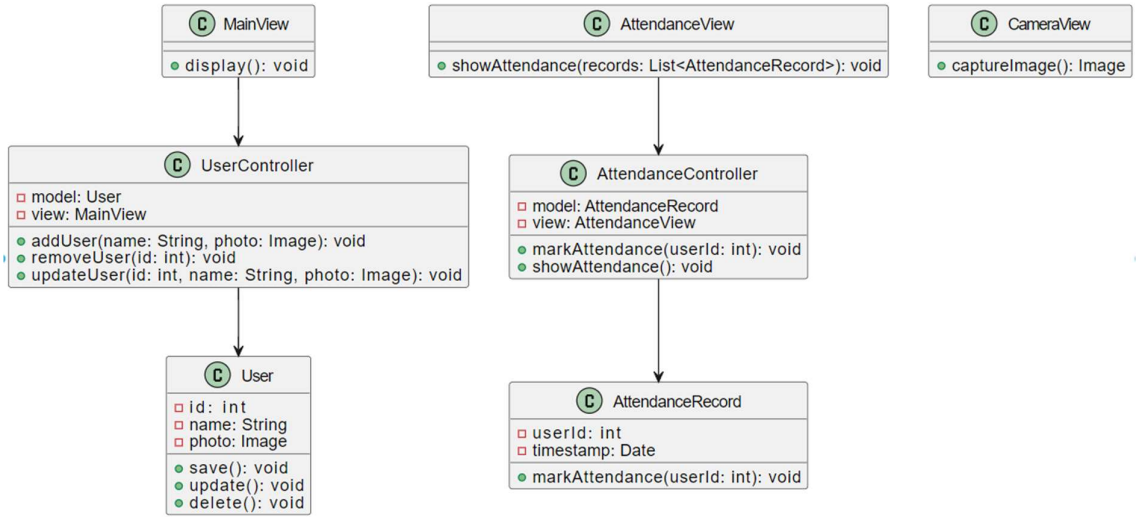
- **Testability and Quality Assurance:**

  With MVC, individual components of FRAMS can be unit tested independently, facilitating thorough quality assurance and reducing the risk of regressions during development.

- **Community Support and Familiarity:**

  Leveraging MVC ensures that developers working on FRAMS can benefit from extensive community support, documentation, and resources, streamlining development efforts and reducing time-to-market.

4.1.3. MVC ARCHITECTURE DIAGRAM:



4.2. DESIGN PRINCIPLES:

- **Single Responsibility Principle:**

  In FRAMS, each component adheres to the SRP by focusing on a single responsibility:

  - Model: Handles data management and business logic related to attendance, user information, and payroll calculations.
  - Controller: Processes user requests, performs authentication, manages attendance marking, and triggers payroll calculations.
  - View: Presents attendance reports, payroll summaries, and user interfaces for interaction.

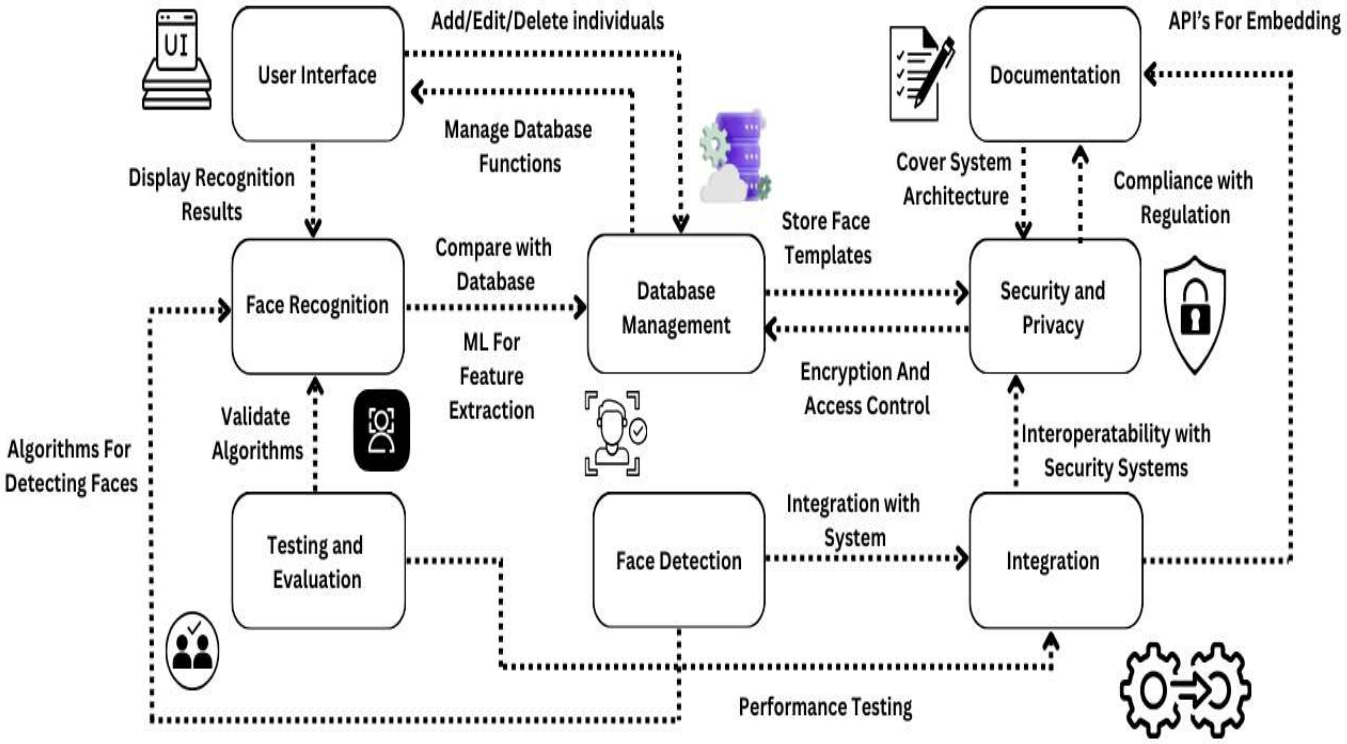
- **Open/Closed Principle:**
  - FRAMS embraces the OCP by allowing for easy extension without modification of existing code.
  - New authentication methods (e.g., biometric, OTP) can be added without altering existing authentication logic.
  - Additional reporting features (e.g., graphical analytics) can be incorporated without changing existing reporting mechanisms.

- **Liskov Substitution Principle (LSP):**
  - FRAMS ensures the LSP by enabling subclasses to be substituted for their base classes without altering system behavior.
  - Different implementations of attendance management can be seamlessly integrated, such as integrating facial recognition or RFID attendance systems.

- **Interface Segregation Principle (ISP):**
  - FRAMS applies the ISP by defining interfaces tailored to specific client needs.
  - The IUserAuthentication interface provides methods for user authentication, while the IAttendanceManager interface defines methods for attendance management. This prevents clients from depending on interfaces they do not use.

- **Dependency Inversion Principle (DIP):**
  - FRAMS utilizes Dependency Injection to adhere to the DIP.
  - The Controller component relies on abstractions (interfaces) rather than concrete implementations, allowing for flexibility and easier testing. For example, the Controller depends on interfaces such as Attendance Manager and User Authentication rather than concrete classes.

4.3. SYSTEM ARCHITECTURE:



5. DEPLOYMENT DIAGRAM:

The deployment diagram for the Facial Recognition Attendance Management System outlines the physical layout of the system's architecture. Client devices, including student, staff, and driver mobile devices or workstations, capture facial images and send them to the central server for processing. The Facial Recognition Server employs sophisticated algorithms to identify and verify faces against stored data. A secure Database Server manages user data, attendance logs, and profile information, ensuring data integrity and privacy through encryption and access controls. An Admin Dashboard Server provides a web-based interface for administrators to manage registrations, view reports, update profiles, and make manual corrections. The network infrastructure, comprising routers, switches, and firewalls, ensures secure and efficient communication between client devices and servers. This deployment configuration enhances the system's reliability, security, and scalability, providing a robust solution for managing attendance through facial recognition.