MILES ACQUISITION SYSTEM

Miles Acquisition System System

Software Requirements Specification

Version 1.0

Submitted in Partial Fulfillment for the Award of Degree of Bachelor of Technology

in Information Technology from Rajasthan Technical University, Kota

MENTOR:

Ms. Astha Joshi

(Dept. of Information Technology)

COORDINATOR:

Dr. Richa Rawal

(Dept. of Information Technoloy

SUBMITTED BY:

Tarun Kumar (21ESKIT109)

Verma Devashish (21ESKIT114)

Nighat Shakoor (21ESKIT310)

DEPARTMENT OF INFORMATION TECHNOLOGY

SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY,MANAGEMENT & GRAMOTHAN

Ramnagaria (Jagatpura), Jaipur – 302017

SESSION 2024-25

Table of Contents

1. Introduction

A **Miles Acquisition System (MAS)** is an intelligent platform developed to streamline and automate the collection, tracking, and redemption of airline miles. It enables users to submit flight details, which are then verified and credited as reward miles using a secure and role-based access system. MAS utilizes cloud-based infrastructure and real-time databases, primarily built on Firebase, to ensure seamless interaction between users, administrative staff, and the system backend.

At its core, MAS integrates modern web technologies such as Angular for responsive user interface design and Firebase Authentication for secure, role-specific access. Users can log flight journeys, receive instant updates on reward status, and redeem miles through a simplified digital workflow. The system also includes automated and manual verification flows to validate user submissions, enhancing trust and data integrity.

1.1Purpose

The purpose of this project is to develop a **Miles Acquisition System (MAS)** that simplifies and automates the process of collecting, managing, and redeeming airline reward miles. The system enables users to submit flight details, which are verified and converted into reward points through a secure and role-based platform. In an era where customer loyalty and user convenience are paramount, MAS bridges the gap between traditional rewards programs and modern digital expectations.

By integrating cloud technologies and real-time data synchronization, the project aims to offer a seamless experience for users and administrators alike. Whether it’s tracking miles, managing user accounts, or redeeming rewards, MAS provides a centralized and efficient solution. The goal is to enhance transparency, reduce manual intervention, and foster greater user engagement through intuitive workflows and responsive interfaces.

1.2 Scope

The scope of the **Miles Acquisition System (MAS)** encompasses a wide range of functionalities to streamline and enhance the user experience in managing and redeeming travel reward miles. Key features and deliverables include:

• **User-Friendly Submission Portal**: Allows users to submit flight details, travel documents, and booking information seamlessly through a structured form-based interface.

• **Role-Based Access Control**: Implements layered access for users, administrators, and verifiers to ensure data integrity and secure operation.

• **Automated Verification Engine**: Cross-checks uploaded data and flight information to validate eligibility and ensure accurate reward point allocation.

• **Real-Time Tracking and Notifications**: Keeps users informed with instant updates on submission status, reward miles earned, and redemption options.

• **Admin Dashboard & Analytics**: Provides tools for administrators to manage submissions, view analytics, and generate reports for system performance and user engagement.

• **Secure Data Management**: Ensures data privacy through encryption and secure cloud storage, complying with modern data protection standards.

By addressing these core areas, MAS provides a centralized and reliable platform that bridges technology with customer loyalty. The system is designed to reduce manual effort, improve reward transparency, and build long-term user engagement.

1.3Definitions, Acronyms and Abbreviations

Here’s a table listing definitions, acronyms, and abbreviations relevant to facial recognition systems:

Term	Definition	Acronym/Abbreviation
Software Requirements Specification	Software Requirements Specification, a document detailing the functional and non-functional requirements of the system.	SRS
Miles Tracking	The process of tracking the miles earned by users based on their flight data and allowing users to redeem these miles for rewards.	
Admin Dashboard	A dashboard used by administrators to manage users, flights, and reward redemption requests.	
Angular Router	A routing mechanism in Angular that allows users to navigate between components and different views.	
MySQL	A relational database management system used for storing structured data, such as user details, flight information, and redemption requests.	

1.4 References

The development of the *Miles Acquisition System (MAS)* is informed by industry practices, documentation, and technologies used in modern web and cloud-based application development:

- Firebase Documentation. (2024). *Firebase Authentication and Firestore Guide*. Retrieved from https://firebase.google.com/docs
- Angular Team. (2024). *Angular Developer Guide*. Retrieved from https://angular.io/docs
- Mozilla Developer Network (MDN). (2024). *Web APIs and JavaScript Documentation*. Retrieved from https://developer.mozilla.org/
- Bootstrap Team. (2024). *Bootstrap 5 Documentation*. Retrieved from https://getbootstrap.com/docs/5.0/getting-started/introduction/

- Johnson, R., & Patel, S. (2021). *Cloud-Based Systems for Airline Data Management*. *International Journal of Aviation IT Systems*, 12(4), 112–130.
- Sharma, L. (2023). *Role-Based Access Control in Web Applications*. *Journal of Web Security and Infrastructure*, 9(2), 87–102.
- OpenJS Foundation. (2024). *Node.js and Express Guide*. Retrieved from https://nodejs.org/en/docs/

1.5Technologies to be used

The implementation of the **Miles Acquisition System (MAS)** utilizes modern web development technologies, frameworks, and services to ensure scalability, performance, and user-friendliness. Below is a breakdown of the technologies used:

- **Programming Languages:**
- **TypeScript**: Used as the primary language for Angular frontend development.
- **JavaScript**: Utilized in client-side scripting and form interactions.
- **HTML5 & CSS3**: For structuring and styling the user interface.
- **Libraries and Frameworks:**
- **Angular**: Frontend framework for building dynamic, single-page applications and managing components, services, and routing.
- **Bootstrap**: Used for responsive UI components and design consistency.
- **Node.js & Express** *(if applicable)*: For building backend APIs and handling server-side logic.
- **Cloud Services and Backend:**
- **Firebase Authentication**: Handles user login, registration, and session management.
- **Firebase Firestore** *(if used)*: A cloud-hosted NoSQL database for real-time data storage.
- **MySQL** *(if used)*: A relational database for structured storage of user, flight, and miles data.
- **APIs and Services:**
- **Firebase SDK**: For integrating authentication, Firestore, and other Firebase services into the app.
- **Custom REST APIs** *(optional)*: Built with Node.js/Express for handling business logic,

if backend services beyond Firebase are implemented.

- **Data Models:**
- Structured using TypeScript interfaces for entities like User, Flight, and MilesHistory to ensure type safety and consistency.
- **UI/UX Technologies:**
- **Angular Components**: For creating modular, reusable UI elements (e.g., forms, tables, dashboards).
- **CSS3 & Bootstrap**: For layout, responsiveness, and styling across devices.
- **Responsive Design**: Ensures the system works smoothly on desktops, tablets, and mobile devices.

1.6 Overview

The **Miles Acquisition System (MAS)** introduces an efficient and user-centric approach to managing airline miles and flight data. The system's workflow is structured to ensure smooth interaction between users, staff, and administrators:

1. **User Login and Role Assignment**:
    - Users sign in via Firebase Authentication.
    - Based on user roles (User, Staff, Admin), access is granted to the appropriate dashboard and functionalities.
2. **Flight Submission by Users**:
    - Users enter their flight details (flight number, date, distance, etc.) via a structured form.
    - Submitted data is stored and marked for verification.
3. **Flight Verification by Staff**:
    - Staff members review submitted flight details.
    - Flights are either approved (miles are credited to the user) or rejected with reasons provided.
4. **Miles Tracking and Redemption**:
    - Approved flights contribute to the user's miles balance.

- Users can request to redeem their miles for rewards.
- Admins review and approve or reject redemption requests.

5. **Administrative Oversight**:
    - Admins manage users, review flights and redemption logs, and monitor overall system activity through a dedicated dashboard.

This system leverages cloud-based tools and modular design principles to streamline the airline loyalty process. By enabling real-time tracking, secure role-based access, and efficient data flow across modules, MAS enhances transparency and user engagement within the airline's loyalty ecosystem.

2.Literature survey

2.1 Review of Related Work

2.1.1 Loyalty and Reward Management in Airlines

Airlines have widely adopted frequent flyer programs (FFPs) to enhance customer loyalty. Systems such as *AAdvantage* (American Airlines) and *SkyMiles* (Delta Airlines) are examples of mature reward systems that let users accumulate miles for each flight. These platforms often use centralized databases and customer management systems to track, update, and redeem miles efficiently

2.1.2 Mile Accumulation Mechanisms

2.1.3 Mile Accumulation Mechanisms

Mile calculation can depend on multiple factors—flight distance, fare class, membership level, and even seasonal promotions. Modern systems are adopting automated mile allocation using APIs integrated with booking and check-in systems, reducing manual effort and improving accuracy. There is also a growing interest in blockchain-based miles tracking to ensure tamper-proof transactions.

2.1.4 Secure and Scalable Authentication Systems

Secure user authentication is a core requirement in any system handling personal and financial data. Firebase Authentication is a widely used cloud-based tool that simplifies authentication with support for email/password logins, OAuth (Google, Facebook), and token management. Its scalability makes it ideal for systems with growing user bases.

2.2Knowledge gaps

- **Lack of Modular Dashboards**: Many existing systems don't offer modular dashboards with role-based access for admins and staff separately.
- **Manual Verification Bottlenecks**: Current verification of flight details for miles accumulation is often manual, error-prone, and time-consuming.
- **Poor UI for Reward Systems**: Users often find it difficult to navigate existing interfaces for submitting flight data or redeeming miles.
- **Limited Real-Time Data Syncing**: Several platforms do not support real-time status updates, leading to delays in mile crediting or redemption processing.
- **Security Vulnerabilities**: Many loyalty systems lack robust authentication mechanisms, exposing them to fraud or unauthorized access.
- **Lack of Centralized Models**: There is limited use of centralized data models that can be reused across components (e.g., flights, users, redemption history).
- **Low Flexibility in Custom Rule Creation**: Existing platforms lack the ability to define custom rules for mile approval or reward assignment based on different roles.

2.3 Comparative Analysis

Feature	Existing Systems	Proposed MAS System
Authentication	Basic login systems with minimal role separation.	Firebase-based system with secure role-based access control (user, staff, admin).
Mile Accumulation	Manual entry or basic booking sync without flexible input validation.	Dynamic submission of flight data with verification and approval workflows.
Real-Time	Delayed updates of mile balances	Real-time synchronization using Angular

Feature	Existing Systems	Proposed MAS System
Updates	and redemption status.	+ Firebase Firestore.
Admin/Staff Dashboards	Limited or no separation between admin and staff tasks.	Distinct dashboards tailored for admin (user management) and staff (flight validation).
User Interface (UI)	Outdated designs; difficult to navigate.	Modern, responsive UI using Angular with clear action buttons and summaries.
Security	Weak or outdated password mechanisms.	Firebase Auth with support for secure logins and token validation.
Extensibility	Hard to scale or adapt to other loyalty systems (e.g., hotels, rentals).	Modular code and reusable services for expanding to other industries.

2.4 Summary

This literature survey highlights the evolution and limitations of existing airline loyalty and miles tracking systems. While major platforms have successfully deployed reward mechanisms, they often fall short in real-time updates, modular admin tools, and flexible verification systems. The proposed *Miles Acquisition System (MAS)* addresses these gaps by incorporating secure authentication, modular dashboards for staff and admin, and real-time data management. The system also ensures better scalability, UI/UX, and integration readiness for future expansion.

3. Specific Requirements

This chapter elaborates on the specific requirements necessary for implementing the music recommendation system driven by facial expression recognition. The outlined requirements ensure the system’s effectiveness, efficiency, and user satisfaction.

3.1 Functional Requirement

- **Miles Accumulation and Tracking**: The system must automatically track the number of miles accumulated based on user actions such as flight bookings, purchase history, or other

qualifying activities. The system must update the miles balance in real-time, with an accuracy rate above 98%.

- **Miles Redemption**: The system must allow users to redeem accumulated miles for rewards, including flight tickets, upgrades, or services. Redemption options must be flexible, allowing users to partially use miles for a discount, pay for additional services, or transfer miles to another user.

- **Real-Time Miles Status Update**: The system must instantly update users about the status of their miles after every qualifying activity or redemption request, such as verifying miles for a flight upgrade, and immediately notifying the user of approval or denial.

- **Dynamic User Dashboard**: The system must provide a personalized dashboard for users, showing up-to-date miles balance, available rewards, pending miles claims, and historical transaction data.

- **Multi-Platform Support**: The system must work seamlessly across multiple platforms such as web browsers, iOS, Android apps, and smartwatches for easy access and management of loyalty miles.

3.2 Non Functional Requirements

Non-functional requirements focus on the quality attributes of the system. Key aspects include**:**

- **Performance**: The system must process miles tracking and redemption requests within 2 seconds for optimal user experience. Efficient back-end processes and well-optimized algorithms are crucial to maintain this performance, especially during high-volume periods like sales or holidays.
- **Scalability**: The system should be scalable to accommodate millions of users and transactions, supporting high concurrency without sacrificing performance. Utilizing cloud-based solutions like AWS or Google Cloud ensures scalability.
- **Security and Privacy**: The system must employ robust encryption methods (e.g., AES-

256) for data storage and transmission to protect sensitive user data (e.g., miles balance, personal info). Compliance with global data protection regulations such as GDPR and CCPA must be maintained to ensure ethical handling of personal and transactional data.

- **Usability**: The system should feature an intuitive user interface with easy navigation for viewing and managing miles, rewards, and transactions. Additional features such as voice command support, push notifications, and a mobile-friendly interface should be included to enhance user experience.
- **Reliability**: The system should maintain 99.9% uptime to ensure that users can consistently access their miles data and make transactions. Redundant backup systems and failover mechanisms must be in place to avoid service disruptions.
- **Compatibility**: The system should work across major operating systems (Windows, macOS, Linux, iOS, Android) and be accessible via all modern web browsers (Chrome, Safari, Edge). Compatibility with smart devices such as smartwatches and connected home systems (e.g., Amazon Alexa, Google Assistant) is also required.

3.3Hardware Requirements

The system's hardware components ensure smooth operation and compatibility:

- **Server Infrastructure**: The backend should be supported by high-performance cloud servers with robust CPU, memory, and storage. Multi-core processors (e.g., Intel Xeon or AMD EPYC) are necessary for handling multiple user transactions concurrently.
- **Database Storage**: At least 1 TB of storage space on scalable, cloud-based databases (such as AWS RDS or Google Cloud SQL) is needed to store user data, transaction histories, and reward inventory. High-availability storage setups are recommended.
- **Load Balancers**: To ensure smooth operation during peak traffic times, multiple load balancers should be employed to evenly distribute incoming requests and maintain optimal server performance.
- **IoT Devices**: Integration with devices like smartphones, tablets, or connected home devices is required to enable seamless interaction with users across platforms.
- **Display Device**: A display with a minimum resolution of 1280x720 pixels is recommended for web-based interfaces. For mobile apps, the system should be responsive to fit various

screen sizes.

3.4Software Requirements

Below is a detailed list of the necessary software requirements:

1. **Operating Systems**:
    - **Windows, Linux (Ubuntu/CentOS), macOS**: Depending on deployment preferences and system compatibility.
2. **Development Environment**:
    - **Programming Language**:
        - **JavaScript/TypeScript**: For frontend development with **Angular** or **React.js**.
        - **Python or Java**: For backend development and server-side functionalities.
    - **IDE/Text Editors**: PyCharm, Visual Studio Code, IntelliJ IDEA.
3. **Libraries and Frameworks**:
    - **Frontend Frameworks**:
        - **Angular/React**: For responsive web apps and rich user interfaces.
    - **Backend Frameworks**:
        - **Flask/Django**: Lightweight frameworks to build the API for handling miles data, user requests, and reward tracking.
    - **Database Management**:
        - **Relational Databases**: **PostgreSQL/MySQL** for structured data such as user profiles and transaction logs.
        - **NoSQL Databases**: **MongoDB** for unstructured or semi-structured data like logs or notifications.
4. **Security & Authentication**:
    - **OAuth2**: For secure user authentication and authorization.
    - **Encryption**: **PyCryptodome** or **Cryptography** for encryption of sensitive data.
5. **Testing Frameworks**:
    - **Unit Testing**: **Pytest** for Python-based components.

- **UI Testing**: **Selenium** for testing the user interface and ensuring responsiveness across browsers.

6. **API and Cloud Services**:
    - **REST APIs**: For integration with third-party services such as airlines, reward vendors, or payment systems.
    - **Cloud-Based Services**: For additional processing power and scalability. AWS or Google Cloud are recommended for elasticity.

---

7. 3.5 Agile Methodology

1. **Iterative Development**:
    - Development occurs in 2-3 week sprints, focusing on milestones such as user login, miles tracking, and redemption functionality.
2. **Daily Standups**:
    - Daily meetings for team synchronization and quick issue resolution.
3. **Incremental Prototyping**:
    - A working prototype of the system's major functionalities (e.g., miles balance, redemption process) is provided for stakeholder feedback after each sprint.
4. **Retrospectives**:
    - Sprint reviews for assessing progress, identifying challenges, and applying learnings to the next cycle.

---

8. 3.6 Business Process Model

1. **User Login**:
    - Secure user authentication via OAuth2. Users sign in using credentials or social login options.
2. **Miles Accumulation**:

- The system records user actions (e.g., flight bookings, purchases) and automatically updates their miles balance.

3. **Miles Redemption**:
    - Users select rewards and redeem their miles. The system verifies if they have enough miles and processes the redemption.
4. **Transaction Update**:
    - Once the transaction is successful, the system notifies users in real-time about their new miles balance and the status of the reward.

Supplementary Requirements

The supplementary requirements focus on the non-functional aspects and additional features of the **Miles System** to ensure robustness, usability, and compliance.

1. **Performance**:
2. The system should process user actions (miles tracking, rewards redemption) with an average response time of under 1 second.
3. **Security**:
4. Data encryption with **AES-256** and implementation of **OAuth 2.0** for secure API and user access.
5. **Compliance**:
6. Ensure adherence to **GDPR** and **CCPA** for privacy and data protection. Users should have the option to delete their profiles and data.
7. **Usability**:
8. The system must be simple and easy to use with a clean interface, enabling users to view and manage their miles easily.
9. **Scalability**:
10. The system should scale effortlessly to handle thousands of concurrent users.
11. **Reliability**:
12. Ensure 99.9% system uptime, with failover mechanisms to prevent disruptions.

13. **Environmental Conditions**:
14. The system should work across multiple environments (e.g., varying light conditions for mobile apps) and ensure cross-device compatibility.
15. **Maintenance**:
16. Clear documentation and tools for developers for ongoing support, bug tracking, and future updates.

4.System Architecture

The system architecture of the **Emotion-Based Music Recommendation System** follows a modular design comprising input, processing, and output layers.

1. **Input Layer**:
    - Captures images or video streams through cameras or uploaded files.
    - Performs initial processing tasks like resizing and lighting adjustments.
2. **Processing Layer**:
    - Uses emotion detection models (e.g., CNNs, TensorFlow, or OpenCV) to classify emotions from facial expressions.
    - Maps detected emotions to appropriate music genres or playlists.
3. **Database Layer**:
    - Stores user preferences, emotion-music mappings, and historical data.
    - Interfaces with relational databases (e.g., MySQL) or NoSQL systems (e.g., MongoDB).
4. **Output Layer**:
    - Provides personalized music recommendations via APIs or user interfaces.
    - Displays dynamic playlists tailored to users' emotions.

This architecture ensures secure data handling, scalability, and real-time processing capabilities for diverse use cases.

4.1Client-Server Architecture

The MAS (Miles Acquisition System) adopts a **client-server architecture** to handle facial recognition, identity validation, and mileage tracking efficiently across distributed systems. This ensures **scalability**, **security**, and **real-time performance** suitable for high-traffic environments like airports or transportation hubs.

1. **Client Side (Front-End)**

   - **Functionality: Captures real-time facial data, authenticates user identity, and interacts with travelers.**

   - **Components:**

     - **User Interface (UI): Web/mobile application or kiosk-based system that allows users to initiate identity verification, view travel history, or redeem miles.**
     - **Camera Module: Captures facial images/video during check-in or entry processes using device-integrated or external cameras.**
     - **Preprocessing Unit: Enhances image quality (adjusts lighting, contrast) before sending to the server.**
     - **Local Cache (Optional): Temporarily stores captured data for offline mode or poor connectivity scenarios.**

   - **Communication with Server:**

     - **Sends preprocessed facial data to the server via HTTPS (REST API) or WebSocket (for continuous verification).**
     - **Receives identity match result, mileage update, or access approval.**

2. **Server Side (Back-End)**

   - **Functionality:** Performs facial recognition, identity verification, mileage calculation, and data management.

- **Components:**

  - **Facial Recognition Engine:** Uses machine learning models (e.g., CNNs with TensorFlow/PyTorch) to authenticate the user from a facial database.
  - **Miles Calculator Module:** Calculates earned miles based on flight or activity data.
  - Database Layer: Stores user profiles, biometric templates, travel history, and loyalty points (using MySQL or MongoDB).
  - **API Management Layer:** Exposes REST APIs for client-server communication.
  - Authentication and Security Service: Manages access control, token-based authentication, and encryption**.**

- **Communication with Client:**

- Processes requests from clients and responds with identity match status, accumulated miles, or further instructions.
- Sends real-time feedback through WebSocket where required (e.g., live check-ins).

3. Communication Flow

1. **User Request**: The user initiates a request (e.g., starting a webcam session).
2. **Data Transfer**: The client sends the captured image/video to the server via an HTTP POST request.
3. **Server Processing**: The server detects emotions, maps them to music genres, and generates a personalized playlist.
4. **Result Delivery**: The server returns the playlist to the client, which displays it to the user.

4. Key Benefits of Client-Server Architecture

- **Scalability**: Supports multiple users simultaneously by scaling the server horizontally.
- **Centralized Processing**: Heavy computations (e.g., emotion detection) occur on the server, reducing the load on client devices.

- **Data Security**: Sensitive data is processed and stored securely on the server, ensuring encryption and compliance.
- **Real-Time Performance**: Enables real-time emotion detection and music recommendation through optimized server-side processing.

4.2 Communication Interfaces

Communication interfaces enable data exchange between the client (front-end), server (back-end), and external systems. These interfaces ensure smooth interaction, efficient data transmission, and real-time emotion detection and playlist generation.

1. Client-Server Interface (API Interface)

- **Functionality:** Facilitates communication between client devices (e.g., kiosks, mobile apps) and the central MAS server for identity verification and mile updates.
- **Protocol:** HTTP/HTTPS using RESTful APIs or WebSocket for real-time interaction.
- **Data Format:** JSON for structured requests and responses.
- **Actions:**

- Upload facial image or video for recognition.
- Request identity verification and miles calculation.
- Receive verification status, earned miles, or access permissions.

2. Database Interface

- **Functionality**: Manages persistent storage and retrieval of user profiles, biometric templates, flight history, and accumulated miles.
- **Protocol**:

- SQL for relational databases (e.g., MySQL/PostgreSQL).
- BSON/JSON for NoSQL databases (e.g., MongoDB).

  - **Actions**:

- Store/update facial recognition data and user profiles.
- Retrieve past travel logs and current mileage.
- Log check-in events, access history, and system audits.

4. WebSocket/Real-Time Communication Interface

- **Functionality**: Ensures low-latency communication between the client and server.
- **Protocol**: WebSocket for bi-directional communication.
- **Data Format**: JSON.
- **Actions**:
- Fetch booking or ticket information for a passenger.
- Push verified mileage data to partner loyalty programs.
- Pull passenger metadata for facial matching context (e.g., flight ID, seat number).

6. Security and Encryption Interfaces

- **Functionality**: Ensures secure data transfer and processing.
- **Protocol**: TLS/SSL for encrypting HTTP/HTTPS traffic.
- **Actions**:
  - Encrypt facial data and user preferences during transmission.
  - Ensure integrity and authenticity of data using secure tokens (e.g., OAuth2).

5. Overall Description, Design And Implementation

The **Miles Acquisition System (MAS)** is a facial recognition-based solution designed to streamline traveler identification and automate mileage tracking. This intelligent system replaces manual check-in or loyalty scanning with biometric authentication, allowing for **frictionless access**, **secure verification**, and **automated miles accumulation**.

1. **System Workflow**

The system operates through three core stages:

2. **Facial Capture and Preprocessing**: A camera captures a real-time image or video of the traveler.
3. **Identity Verification**: The image is processed by a machine learning model, which matches it against a database of enrolled users.
4. **Mileage Calculation and Logging**: Upon successful identification, MAS checks flight data and logs corresponding miles into the user's loyalty account.
5. **Architecture Components**

- **Input Devices**: Cameras embedded in kiosks, gates, or mobile devices capture facial data.
- **Recognition Engine**: A cloud-based or edge-deployed machine learning model authenticates identity based on facial features.
- **Miles Tracker Module**: Maps verified user identity to travel logs and calculates earned miles.
- **User Interface**: Provides real-time feedback to travelers (e.g., "Miles Added!", "Access Granted").
- **Communication Interfaces**: REST APIs and WebSocket channels enable integration and real-time communication.

---

6. **Key Considerations**
7. **1. Performance and Accuracy**

- High-accuracy face recognition under varying lighting, angles, and demographics.
- Miles must be updated in near real time (≤ 2 seconds response latency).

8. **2. Privacy and Security**

- All facial data and verification logs are encrypted in transit and at rest.
- Compliant with GDPR and local data protection laws.
- Users can manage their consent and request data deletion.

9. **3. Scalability**

- Supports a growing number of travelers and airlines.
- Can scale horizontally across airport locations and terminals.

10. 4. Adaptability

- Capable of operating under diverse environmental conditions (e.g., outdoor terminals).
- Modular design allows integration of new biometric algorithms or airline partners.

This technology redefines user verification in the travel ecosystem, making the process **faster, smarter, and more secure**. By merging artificial intelligence with biometric identity, MAS enhances operational efficiency while respecting traveler privacy and ensuring regulatory compliance.

5.1Product feature

1. **Real-Time Facial Recognition**
    - Accurately detects and identifies users in real-time using live video streams or static images.
    - Supports multiple user verifications simultaneously at check-in counters, gates, or kiosks.
2. **Feature Extraction and Identity Mapping**
    - Utilizes deep learning algorithms (e.g., convolutional neural networks) to extract facial features.
    - Maps facial signatures to unique traveler IDs stored in a secure database for fast authentication.
3. **Automated Miles Acquisition**
    - Automatically calculates and updates mileage based on verified check-ins and flight/travel logs.
    - Integrates with loyalty systems to reflect earned miles in real time without manual input.
4. **High Accuracy and Robust Performance**
    - Maintains high recognition accuracy under varying lighting, angles, and partial occlusions.
    - Inclusive across diverse demographics including different facial structures, ages, and skin tones.

5. **Database Management**
    - Efficiently stores user profiles, biometric templates, travel histories, and mileage data.
    - Supports both relational (MySQL/PostgreSQL) and NoSQL (MongoDB) databases for scalability.
6. **Security and Privacy**
    - Encrypts all biometric and user data using secure standards like AES-256 during transit and storage.
    - Fully compliant with GDPR, CCPA, and other regional data protection regulations.
    - Allows users to view, manage, or delete their data through a secure user portal.
7. **Multi-Platform Compatibility**
    - Operable on web, mobile, and desktop environments for airports, travel agencies, or mobile apps.
    - Supports major platforms including Windows, Linux, Android, iOS, and macOS.
8. **Real-Time Feedback and Notifications**
    - Provides immediate feedback on verification success or failure.
    - Alerts users and administrators of errors due to camera obstruction, poor lighting, or data mismatch.
9. **Scalability and Extensibility**
    - Designed to scale across multiple airports, travel hubs, or transport systems.
    - Easily integrates with external APIs such as airline booking systems or loyalty programs.
10. **Customizable Settings**
    - Allows tuning of face-matching thresholds for specific use-case needs (e.g., high-security zones).
    - Supports configuration of mileage rules, reward tiers, and event triggers per user or organization.
11. **Device and Camera Compatibility**
    - Compatible with embedded cameras, smartphone cameras, or external webcams.

- Supports both video and static image inputs for flexible deployment options.

12. **Analytics and Reporting**
    - Generates insights on traveler patterns, system performance
    - Exports visual and tabular reports for use by system admins and travel partners.
13. **Cloud and Offline Deployment**
    - Deployable on-premises for private institutions or on the cloud for public
    - Offline functionality supported with local syncing capabilities upon reconnection.
14. **Multi-Language Support**
    - Offers user interfaces and administrator dashboards in multiple languages.
    - Enhances usability for international travelers and airline staff.
15. **Environmental Adaptability**
    - Adapts to varying environmental conditions like glare, low light, or noisy backgrounds.
    - Incorporates preprocessing techniques to handle image noise, shadows, and variable camera resolutions.

These features ensure that the **Miles Acquisition System** is **secure**, **intelligent**, and **scalable**—providing a **frictionless, real-time identity verification experience** while ensuring data privacy, broad device compatibility, and seamless integration with loyalty and airline systems.

5.2Data Flow diagram

Below is a description of the Data Flow Diagram (DFD) :

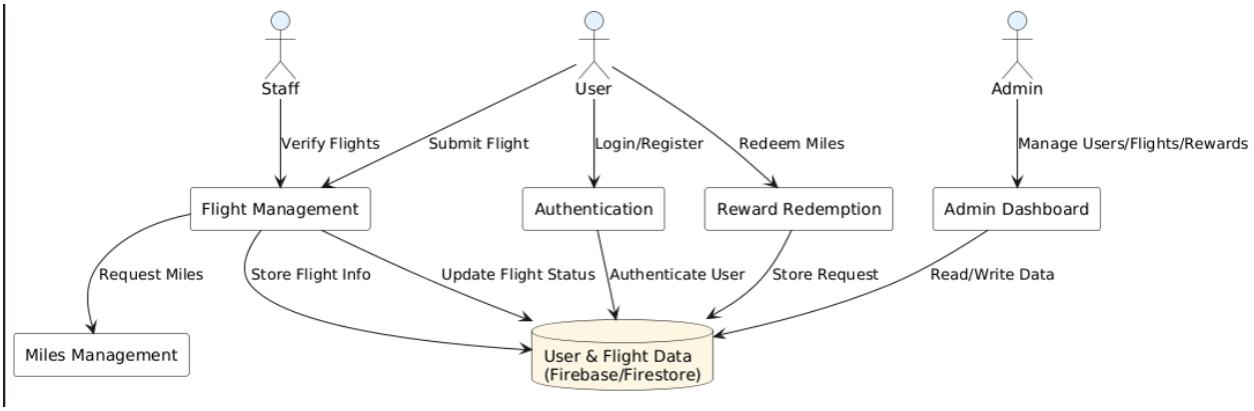

Figure 5.1 DFD

5.3 E-R Diagram

An Entity-Relationship (ER) Diagram is a graphical representation of the entities, their attributes, and the relationships among them in a database system.

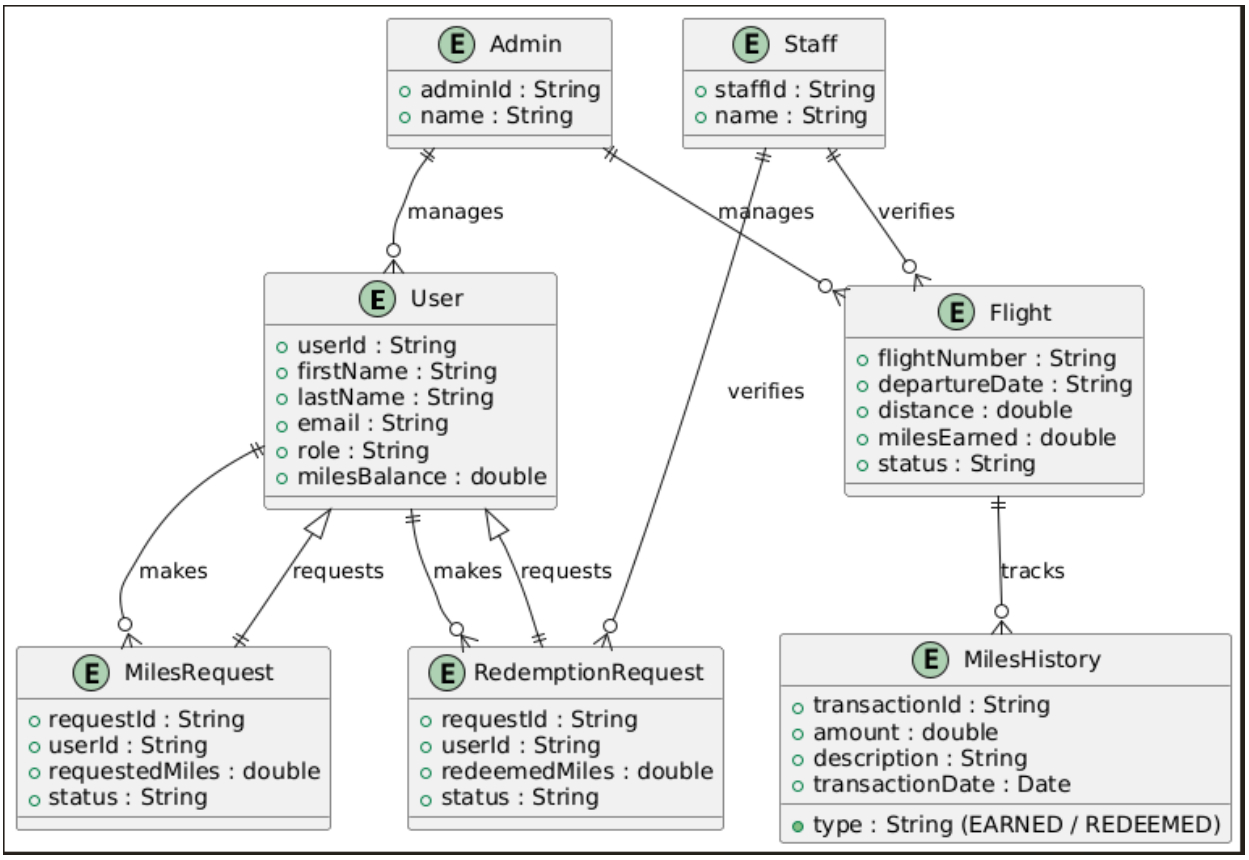

Figure 5.2 E-R Diagram

5.4Class Diagram

A Class Diagram is a static structure diagram in object-oriented design that illustrates the classes, their attributes, methods, and the relationships between them.

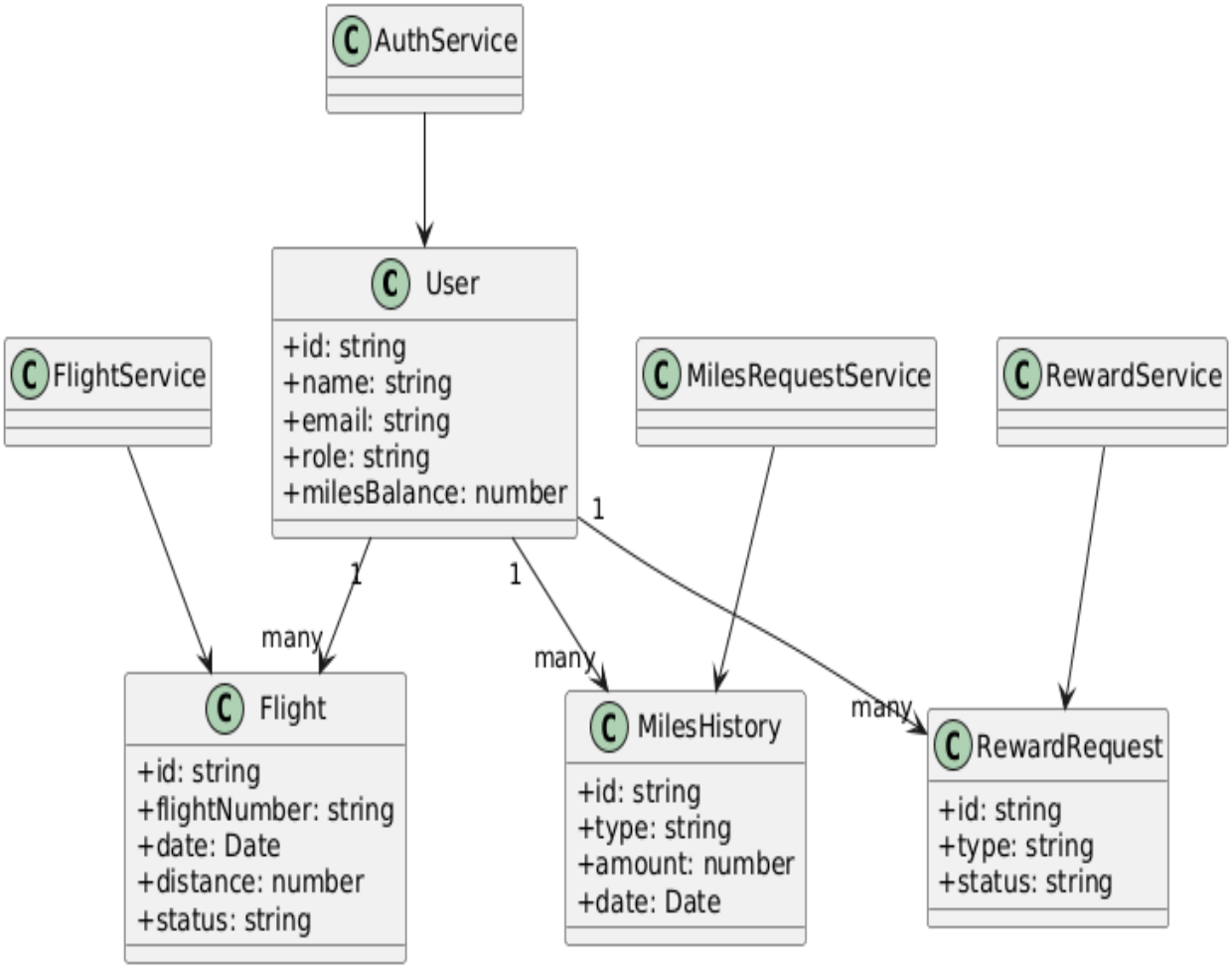

Figure 5.3 Class Diagram

1.

▪

5.5 Use-Case Diagram

A Use Case Diagram visually represents the interactions between users (actors) and the functionalities (use cases) of a system. It helps to capture the functional requirements of the system and identify its primary actors and processes.

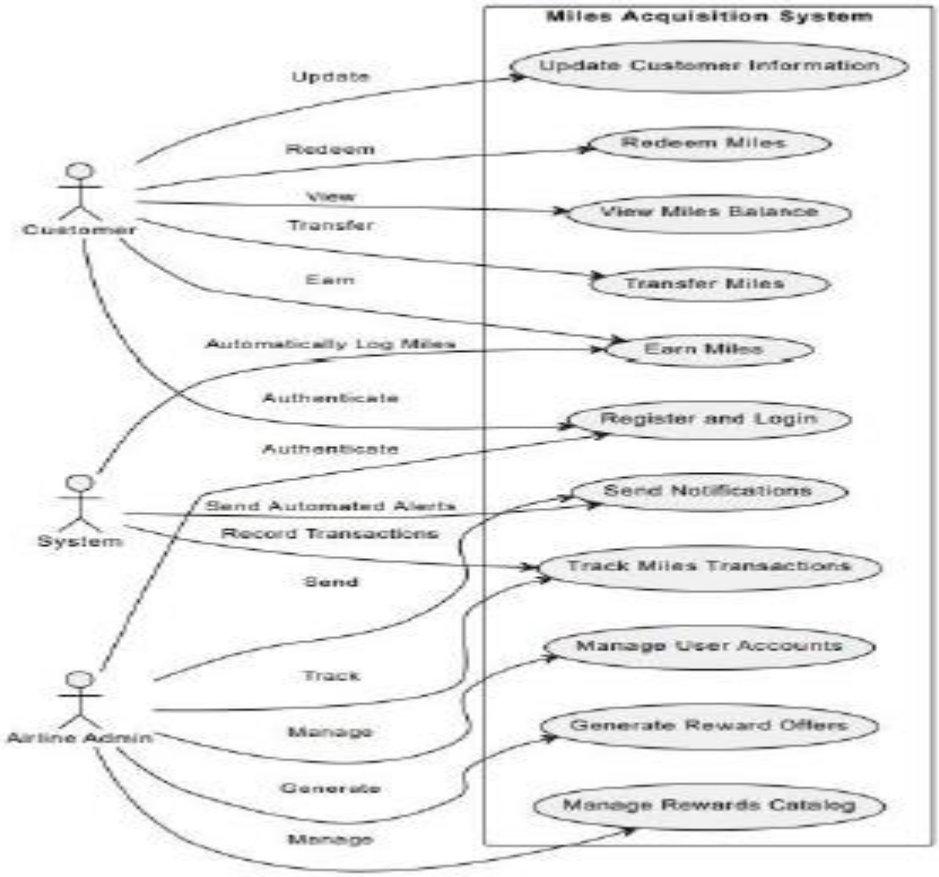

Figure 1.5 Use Case Diagram

5.6 Behaviors Diagrams

1. Activity Diagram

An **Activity Diagram** is a flowchart that depicts the flow of control or data through a system. It helps to visualize the sequence of activities, decisions, and interactions within a system process.

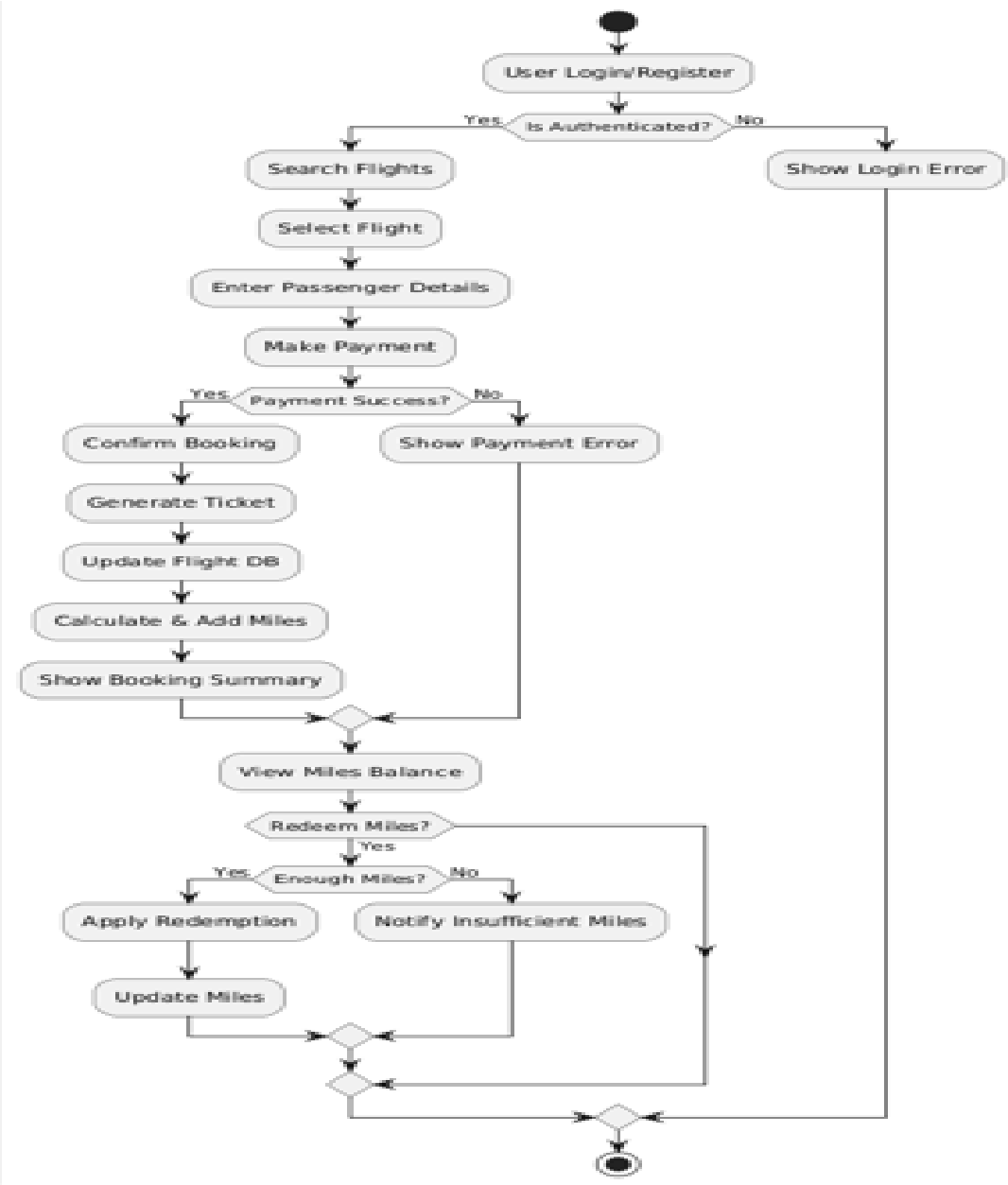

Figure 5.2 Activity Diagram

Key Components of the Activity Diagram:

A. **Start Node**

a. Marks the beginning of the process (e.g., system activation or user input).

B. **Activities**

C. **Decisions**

D. **End Node**

2. Sequence Diagram

A **Sequence Diagram** is an interaction diagram that shows how objects or components in a system interact over time. It highlights the sequence of events or messages exchanged between actors and system components during a specific process.

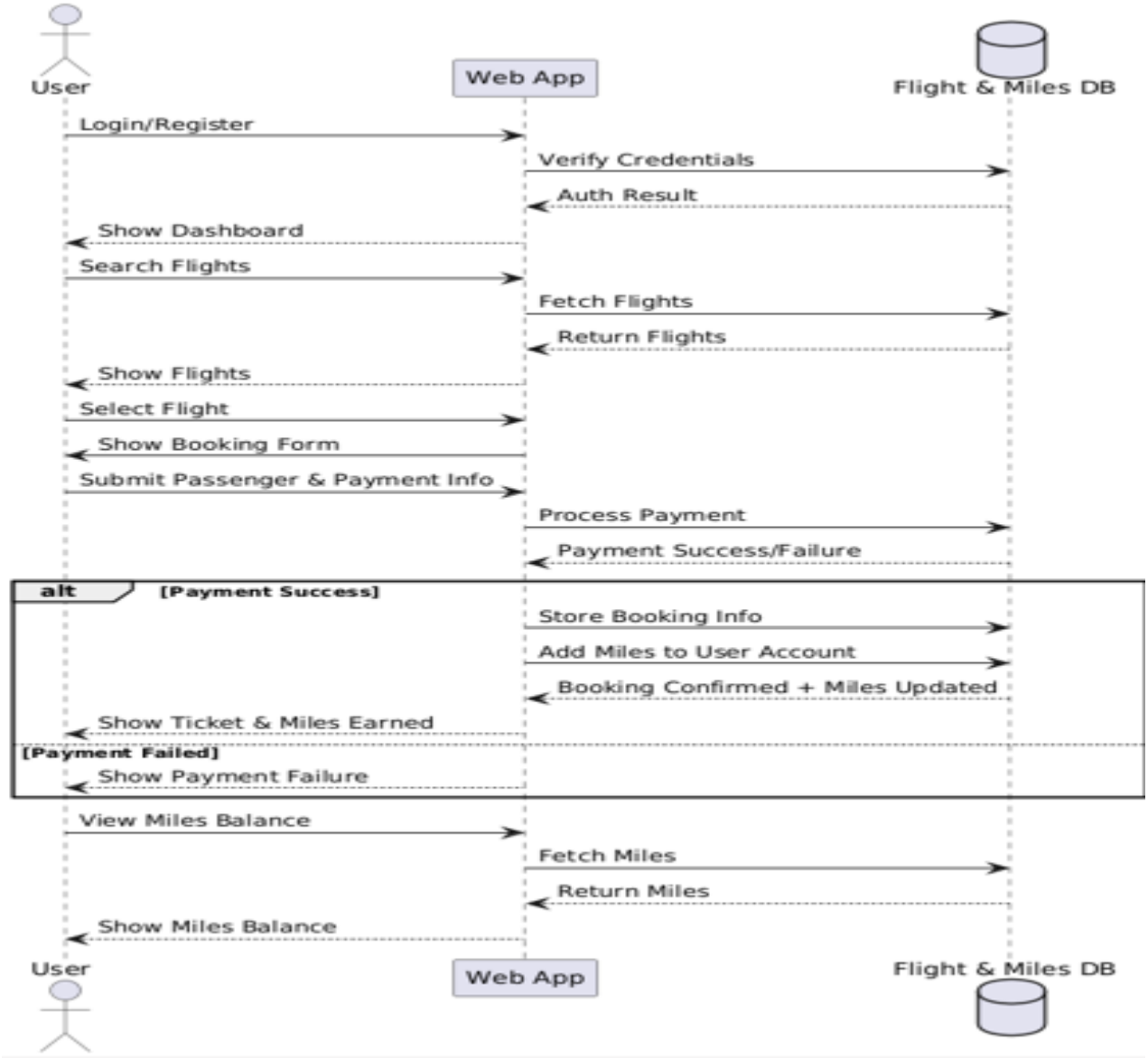

Figure 5.3 Sequence Diagram

Key Components of a Sequence Diagram

A. **Actors**
   a. Represented as vertical lines (lifelines) at the top of the diagram.
   b. Examples: **User**, **Device**, **Recognition System**, **Database**, **Admin**.

B. **Objects or Components**
   a. Represent the entities within the system that perform actions (e.g., **Camera**, **Recognition Algorithm**).

C. **Messages**
   a. Represent communication between actors and components, usually depicted as arrows. Messages are sent in a specific sequence.

D. **Lifelines**
   a. Vertical dashed lines that show the existence of an object over time during the interaction.

E. **Activation Bars**
   a. Rectangular boxes placed on the lifeline to represent when an object is active (performing a task).

5.7Assumptions and Dependencies

Assumptions of the Emotion-Based Music Recommendation System

1. **Quality of Input Data**
   It is assumed that the camera captures clear and well-lit images or videos to enable accurate facial recognition for identity verification.
2. **Identity-Miles Mapping**
   The system assumes that identity-to-mileage associations are pre-defined, accurate, and securely maintained in the database.
3. **Model Accuracy**
   It is assumed that the facial recognition model performs reliably across diverse facial expressions, lighting conditions, and environmental factors.

4. **User Interaction**

   The system assumes users will provide camera access and comply with verification steps during check-in or reward redemption.

5. **Security Measures**

   It is assumed that secure protocols such as encryption (e.g., AES-256) and authentication (e.g., OAuth2) are in place for all data exchanges and storage.

6. **User Consent**

   The system assumes that users have provided consent for their biometric data to be used for identity verification and mileage tracking.

7. **Consistent Environment**

   The system assumes a consistent camera setup and minimal background noise/interference to ensure optimal recognition performance.

Dependencies of the Emotion-Based Music Recommendation System

- **Data Aggregation Platforms**

  MAS depends on reliable integration with external data sources, including airline databases, booking engines, e-commerce platforms, and POS systems to gather transaction data.

- **APIs and Partner Networks**

  The system relies heavily on RESTful APIs provided by partner entities to fetch transaction logs, confirm mile eligibility, and update point balances.

- **Mileage Rules Engine**

  Requires a well-maintained rules engine that handles diverse earning/redemption policies based on travel class, purchase categories, promotional bonuses, etc.

- **Database Infrastructure**

  Depends on robust and scalable databases (SQL/NoSQL) to handle high volumes of user profiles, transactions, mileage records, and redemption history.

- **User Account System**

  Depends on secure user authentication systems (OAuth2, JWT, SSO) and user session management for secure access.

- **Notification System**

  Requires email/SMS/push notification services (e.g., Firebase, Twilio) to notify users about mileage updates, redemption eligibility, or offers.

- **Analytics and Reporting Tools**

  Relies on analytics engines (e.g., Apache Spark, Tableau, Power BI) to monitor performance, user engagement, partner contribution, and fraud detection.

- **Security Infrastructure**

  Requires secure communication protocols (TLS/SSL), encryption (AES-256), and compliance frameworks (GDPR, CCPA) to ensure data integrity and privacy.

- **Business Logic Middleware**

  Relies on middleware services to coordinate between user apps, databases, third-party APIs, and the rules engine.

- **Scalable Cloud Platform**

  A cloud-native backend (e.g., AWS, Azure, GCP) is required to support horizontal scaling and global availability for real-time data processing and redundancy.

6.Conclusion And Future Scope

Conclusion

The **Miles Acquisition System (MAS)** introduces an intelligent, user-centric solution that automates identity verification and loyalty point accumulation using facial recognition technology. By eliminating manual check-ins and streamlining the mileage tracking process, MAS enhances both efficiency and user experience in travel, aviation, and rewards programs.

This system bridges advanced AI with everyday travel workflows—ensuring fast, secure, and accurate recognition while protecting user privacy. Its applications extend across various sectors, including airports, public transport, hospitality, and corporate travel management.

However, challenges such as variations in lighting, spoofing threats, and concerns over data privacy remain critical areas for continuous improvement. Through ongoing research and development, MAS can evolve into a more adaptive, secure, and globally deployable solution.

Future Scope

1. **Improved Recognition Accuracy**

   Advancements in deep learning will improve recognition accuracy in challenging environments such as poor lighting, occluded faces, and dynamic camera angles.

2. **Anti-Spoofing Mechanisms**

   Implementing liveness detection and anti-spoofing techniques (e.g., blink detection, 3D face modeling) will ensure secure and genuine identity verification.

3. **Edge Computing Integration**

   Performing verification on edge devices like kiosks or mobile phones will reduce latency and enhance privacy by minimizing data transfer.

4. **Dynamic Learning for Personalization**

   Integrating reinforcement learning will allow the system to refine verification accuracy and reward rules based on user behavior and feedback.

5. **Multi-Modal Biometrics**

   Incorporating additional biometric inputs such as voice, iris, or fingerprint can increase overall security and reduce false positives/negatives.

6. **Expanded Industry Adoption**

   Beyond aviation, MAS can be deployed in retail loyalty programs, hotels, public transportation systems, and corporate attendance systems.

7. **Privacy Enhancements**

   Utilizing federated learning, anonymization techniques, and transparent data consent practices will build user trust and ensure regulatory compliance (GDPR, CCPA).

8. **Analytics and Reporting Dashboards**

   Developing real-time dashboards for administrators to monitor system health, verification trends, and reward distributions will improve operational oversight.

9. **Offline and Hybrid Deployments**

   Supporting hybrid models (local + cloud) will allow the system to function in low-connectivity regions while syncing data when online.

10. **Global and Multilingual Support**

    Expanding support for multiple languages, cultural identity norms, and localized reward

mapping will improve international adoption and user engagement.

11. **Integration with Smart Devices**
    Linking with smart gates, kiosks, and wearable devices will enable seamless identity verification and mile acquisition across touchpoints.
12. **Automated Travel History Logging**
    The system could integrate with travel itineraries and boarding passes to automatically log trips and update mileage accordingly.
13. **Real-Time Alerting and Issue Resolution**
    Provide alerts for issues such as duplicate identities, suspicious activity, or identity mismatches to prevent fraud and operational delays.
14. **Sustainability and Energy Efficiency**
    Future versions could include optimizations for low-power devices and energy-efficient edge processing for greener deployments.

In summary, the Miles Acquisition System is a forward-looking platform that combines biometric intelligence with loyalty management. Its success will depend on continuous model improvement, strong ethical safeguards, and scalable deployment. As these enhancements are realized, MAS has the potential to redefine how individuals interact with identity-based services in the travel and rewards industry.