"MILES ACQUISITION SYSTEM"

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of Information Technology

Project Mentor:

Ms. Astha Joshi

Assistant Professor

Submitted By:

Tarun Kumar
(21ESKIT109)
Verma Devashish
(21ESKIT114)
Nighat Shakoor
(21ESKIT310)

Department of Information Technology
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota
Session 2024-2025

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that Mr Tarun Kumar, a student of B.Tech (Information Technology) VIII semester has submitted his Project Report entitled Miles Acquisition System under my guidance.

Mentor	**Coordinator**
Ms. Astha Joshi	Dr. Richa Rawal
Assistant Professor	Associate Professor
Signature :	Signature :

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that Ms. Nighat Shakoor, a student of B.Tech (Information Technology) VIII semester has submitted her Project Report entitled Miles Acquisition System under my guidance.

Mentor	**Coordinator**
Ms. Astha Joshi	Dr. Richa Rawal
Assistant Professor	Associate Professor
Signature :	Signature :

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that Mr Verma Devashish, a student of B.Tech (Information Technology) VIII semester has submitted his Project Report entitled Miles Acquisition System under my guidance.

Mentor	**Coordinator**
Ms. Astha Joshi	Dr. Richa Rawal
Assistant Professor	Associate Professor
Signature :	Signature :

DECLARATION

We hereby declare that the report of the project entitled ”Emotify” is a record of an original work done by us at Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur under the mentorship of Ms.Astha Joshi (Dept. of Information Technology) and coordination of Dr. Richa Rawal (Dept.of Information Technology). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech) in the Department of Information Technology.It has not been submitted anywhere else, under any other program to the best of our knowledge and belief.

Team Members **Signature**

Tarun Kumar (21ESKIT109)

Verma Devashish (21ESKIT114)

Nighat Shakoor (21ESKIT310)

Acknowledgement

A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization.We take this opportunity to express our gratitude to all those who have been helping us in making this project successful.

We are highly indebted to our faculty mentor Ms. Astha Joshi .He has been a guide, motivator source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We also thank our project coordinator Dr. Richa Rawal for her co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction.

We would also like to convey our sincere thanks to Dr. Anil Chaudhary, HOD, Department of Information Technology, for facilitating, motivating and supporting us during each phase of development of the project.Also, we pay our sincere gratitude to all the Faculty Members of Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur and all our Colleagues for their co-operation and support.

Last but not least we would like to thank all those who have directly or indirectly helped and cooperated in accomplishing this project.

Team Members:

Tarun Kumar (21ESKIT109)

Verma Devashish (21ESKIT114)

Nighat Shakoor (21ESKIT310)

Table of Content

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Problem Statement and Objective

- Airline companies often struggle with efficiently tracking and rewarding customers for their air travel due to manual processes and lack of centralized systems.
- Users find it difficult to monitor and redeem their accumulated miles, leading to underutilization of loyalty benefits.
- Existing systems lack dedicated platforms for staff to verify flight-related data and manage mile requests systematically.
- The objective of this project is to develop a comprehensive *Miles Acquisition System (MAS)* that streamlines the process of flight data submission, miles calculation, and reward redemption.
- The system enables users to earn and redeem miles through a structured web-based platform and allows staff/admins to manage, verify, and approve the data and requests efficiently.
- Additionally, the platform ensures secure authentication using Firebase and provides role-based access control to maintain operational integrity.

1.2 Literature Survey / Market Survey / Investigation and Analysis

- **Customer Loyalty Systems:** Modern airline loyalty programs require digital platforms that offer transparency and ease of use in tracking earned miles.

- **Web-based Flight Tracking Systems:** Prior systems have proven the effectiveness of digital portals in handling flight data and reward calculations but often lack real-time verification workflows.
- **Firebase Authentication:** Widely adopted for secure, scalable user authentication, Firebase allows seamless integration into Angular-based web apps.
- **Role-Based Dashboards:** Research supports the use of separate dashboards for different roles (user, staff, admin) to enhance manageability and data flow control.
- **Miles Redemption Mechanisms:** Studies reveal that automating reward redemptions boosts user engagement and satisfaction within loyalty programs.

1.3 Introduction to Project

The *Miles Acquisition System (MAS)* is a web-based application aimed at streamlining the airline miles tracking and redemption process. Traditional systems often lack centralized dashboards for verification and reward management, making the process inefficient for both users and staff. This project introduces a role-based system with Firebase Authentication that allows users to submit flight data and track their miles in real-time. Staff members can verify submitted flight information, while admins manage users, approve miles requests, and oversee reward redemptions. By using Angular for the frontend and Firebase for authentication, the system ensures scalability, security, and a clean user experience across different modules.

1.4 Proposed Logic / Algorithm / Business Plan / Solution

- **User Authentication:** Firebase Authentication is used for secure login/logout, session handling, and user state management. Role-based redirection ensures users land on the appropriate dashboard (user, staff, admin).
- **Flight Submission and Verification:** Users input flight details, which are sent to the backend. Staff members validate these entries using predefined criteria (e.g., flight number, departure date, and distance) and approve or reject them.

- **Miles Calculation and Tracking:** Miles earned are calculated based on the flight distance and stored in the user's profile. A history log is maintained using the MilesHistory model.
- **Reward Redemption:** Users can request to redeem miles. Admins review and act upon these requests via the admin dashboard.
- **Role-based Dashboards:** Admins have access to all users, requests, and flight data. Staff members only view and act on flight verification tasks. Each dashboard displays only the relevant modules and features for the role.

This approach ensures efficient management of the entire miles acquisition and redemption process in a secure, scalable, and user-friendly manner.] The MAS system is designed using modular architecture with components dedicated to different user roles and responsibilities. The logical flow of the system is structured as follows:

- **User Authentication:** Firebase Authentication is used for secure login/logout, session handling, and user state management. Role-based redirection ensures users land on the appropriate dashboard (user, staff, admin).
- **Flight Submission and Verification:** Users input flight details, which are sent to the backend. Staff members validate these entries using predefined criteria (e.g., flight number, departure date, and distance) and approve or reject them.
- **Miles Calculation and Tracking:** Miles earned are calculated based on the flight distance and stored in the user's profile. A history log is maintained using the MilesHistory model.
- **Reward Redemption:** Users can request to redeem miles. Admins review and act upon these requests via the admin dashboard.
- **Role-based Dashboards:** Admins have access to all users, requests, and flight data. Staff members only view and act on flight verification tasks. Each dashboard displays only the relevant modules and features for the role.

This approach ensures efficient management of the entire miles acquisition and redemption process in a secure, scalable, and user-friendly manner.

1.5 Scope of the Project

- A secure login system with Firebase Authentication and route protection using Angular guards.
- Dashboards for each role (user, staff, admin) displaying relevant data and controls.
- A flight submission and verification module allowing staff to approve/reject flight requests.
- A miles tracking module for users to view their accumulated miles and transaction history.
- A reward redemption system where users can submit requests and admins can manage them.
- Integration with Firebase (and optionally Firestore/MySQL) for storing user, flight, and reward data.

The project can be expanded to include APIs for flight validation, reward inventory integration, and analytics tools for admin insights. It serves as a scalable solution for airline loyalty programs.] The scope of this project is to build a centralized, role-based *Miles Acquisition System (MAS)* that automates and simplifies the process of accumulating, verifying, and redeeming airline miles. The system supports both end users and staff/admin operations, ensuring transparency and efficiency. Deliverables include:

- A secure login system with Firebase Authentication and route protection using Angular guards.
- Dashboards for each role (user, staff, admin) displaying relevant data and controls.

- A flight submission and verification module allowing staff to approve/reject flight requests.
- A miles tracking module for users to view their accumulated miles and transaction history.
- A reward redemption system where users can submit requests and admins can manage them.
- Integration with Firebase (and optionally Firestore/MySQL) for storing user, flight, and reward data.

The project can be expanded to include APIs for flight validation, reward inventory integration, and analytics tools for admin insights. It serves as a scalable solution for airline loyalty programs.

Chapter 2
Software Requirement Specification

2.1 Overall Description

This project focuses on developing a *Miles Acquisition System (MAS)* — a web-based platform for an airline company that enables users to earn, track, and redeem flight miles while allowing administrators and staff to verify flight submissions and manage user data. The system aims to streamline the entire miles lifecycle, from submission to redemption, while enforcing role-based access controls for different types of users (regular users, staff, and administrators).

The application includes real-time tracking of miles based on submitted flight data, a structured verification and approval process by staff and admins, and redemption options for eligible users. With clearly defined modules and dashboards, MAS ensures secure and efficient management of user interactions, flight details, and reward redemptions.

This solution improves operational transparency and customer engagement in the airline's loyalty program, supporting both business goals and customer satisfaction through efficient digital workflows.

2.1.1 Product Perspective

2.1.1.1 System Interfaces

The MAS consists of multiple interdependent modules:

- **Authentication Module:** Manages login and session validation for users, staff, and admins using Firebase Authentication.
- **User Management Module:** Enables admins to manage user pro-

files, miles balances, and access roles.

- **Flight Management Module:** Allows users to submit flight data and enables staff/admins to verify and update flight statuses.
- **Miles and Redemption Module:** Tracks miles earned from verified flights and handles reward redemption requests.
- **Dashboard Interfaces:** Role-specific dashboards for users, staff, and administrators to interact with data based on their access level.

These interfaces operate together to ensure secure, role-based interactions throughout the system.

2.1.1.2 User Interfaces

The system offers dedicated, user-friendly dashboards for different roles:

- **User Dashboard:** Enables flight submission and viewing of earned miles.
- **Staff Dashboard:** Displays submitted flights for verification and enables staff to approve/reject flight requests.
- **Admin Dashboard:** Facilitates the management of users, flights, and reward redemptions with full control features.

All dashboards are built with Angular and designed to be responsive and intuitive.

2.1.1.3 Hardware Interfaces

As a web application, MAS is hardware-independent but interacts with:

- **Web Browser:** Used to access the application.
- **Server/Cloud Environment:** (If applicable) for deploying back-end services and databases.

Minimum client hardware requirements include a modern browser and internet connectivity.

2.1.1.4 Software Interfaces

The MAS interacts with the following software services and APIs:

- **Firebase Authentication:** For managing user logins and sessions.
- **Angular Framework:** Used for frontend development and routing.
- **Custom Services (FlightService, MilesRequestService, RewardService):** Handle API interactions and business logic.
- **Database (Firebase Firestore or MySQL):** Stores user data, flight details, and miles history.

These software interfaces work together to deliver a secure and consistent user experience.

2.1.1.5 Communications Interfaces

MAS uses secure HTTPS protocols to interact with Firebase and any backend services. It relies on:

- Secure REST APIs for data retrieval and updates.
- Firebase's real-time database communication (if applicable).
- Token-based authentication for session management.

Internet access is mandatory for all operations.

2.1.1.6 Memory Constraints

The application is lightweight on the frontend but may have standard backend resource requirements. Suggested specifications:

- **Client RAM:** Minimum 2GB (4GB preferred).
- **Backend Hosting:** Dependent on Firebase or cloud infrastructure for scalable usage.

The app is optimized for browser execution and does not impose heavy memory loads.

2.1.1.7 Operations

Primary operations of MAS include:

- User authentication and role-based redirection.
- Submission of flight data by users.
- Verification of flight data by staff.

- Approval or rejection of miles and rewards by admins.
- Real-time updates to dashboards and mile balances.

Operations are driven by user roles and designed for efficiency and traceability.

2.1.1.8 Project Functions

The system offers the following key functions:

- Secure user login and logout.
- Flight data submission and tracking.
- Verification and approval of submitted flight details.
- Miles accumulation and redemption.
- Role-based dashboards for streamlined management.

These functions contribute to a robust digital loyalty platform.

2.1.1.9 User Characteristics

MAS is designed for:

- Regular users submitting and redeeming miles.
- Staff verifying flight information.
- Admins managing the overall system.

Users are expected to have basic internet and device usage skills.

2.1.1.10 Constraints

System limitations include:

- Dependence on internet access for all modules.
- Firebase usage quotas or service limits (if applicable).
- Role-based access control enforcement.
- Potential delay in real-time updates on large data sets.

2.1.1.11 Assumptions and Dependencies

Project assumptions:

- Firebase and any integrated backend services will remain available.
- Users have access to the internet and a modern web browser.
- Authentication and data management follow industry-standard security practices.

The system depends on Angular, Firebase services, and reliable hosting infrastructure.

Chapter 3

System Design Specification

3.1 System Architecture

The architecture of the Miles Acquisition System (MAS) follows a layered, modular web-based design to ensure secure access, smooth operations, and scalable performance across different user roles. The high-level system architecture comprises the following components:

- **Frontend Layer (Angular):** Provides the user interface for users, staff, and administrators. Built using Angular with routing, form validation, and REST API consumption.
- **Authentication Module (Firebase):** Manages secure login, registration, and session handling via Firebase Authentication. Each user role (User, Staff, Admin) is routed to the appropriate dashboard.
- **Flight Submission Miles Module:** Allows users to submit flight information. Staff validate submitted data, and Admins finalize approvals and update mile balances.
- **Reward Redemption Module:** Enables users to redeem earned miles. Admins manage redemption requests and update statuses accordingly.
- **Database Layer (Firebase Firestore):** Stores user credentials,

flight details, earned miles, and reward history in structured documents.

- **API Services:** Angular services (e.g., FlightService, MilesRequestService) abstract the logic of interacting with Firebase backend, ensuring maintainability and code separation.

3.2 Module Decomposition Description

Each main module of MAS is decomposed into functional submodules for maintainability and clarity:

1. **Authentication Subsystem**
    - *Login/Signup Module*: Provides Firebase-based login and registration.
    - *Role-based Redirection*: Directs users to appropriate dashboards based on roles (user, staff, admin).
2. **User Dashboard Subsystem**
    - *Flight Submission Form*: Accepts flight number, date, ticket proof, etc.
    - *Miles Tracker*: Displays user's current and historical earned miles.
    - *Redemption Request Interface*: Enables users to request redemption.
3. **Staff Dashboard Subsystem**

- *Verification Panel*: Lists pending flight submissions for validation.
- *Approval/Rejection Tools*: Allows staff to approve/reject flights with optional comments.

4. **Admin Dashboard Subsystem**
    - *Flight Verification Management*: Oversees all flights (approved, rejected, pending).
    - *Miles Allocation Engine*: Allocates or adjusts miles based on flight data.
    - *Redemption Management*: Reviews and processes reward redemption requests.
5. **Database Integration Subsystem**
    - *Firebase Firestore*: Stores structured data in collections (Users, Flights, Miles, Redemptions).
    - *Real-time Sync*: Automatically updates dashboards on data changes.
6. **Service Layer (Angular Services)**
    - *FlightService*: Handles CRUD operations on flight data.
    - *MilesRequestService*: Manages mile-related requests and balances.
    - *RewardService*: Processes redemption data and status updates.

This layered decomposition ensures that responsibilities are well-defined and independent, enabling scalable development and ease of testing.

3.3 High Level Design Diagrams

3.3.1 Use Case Diagram

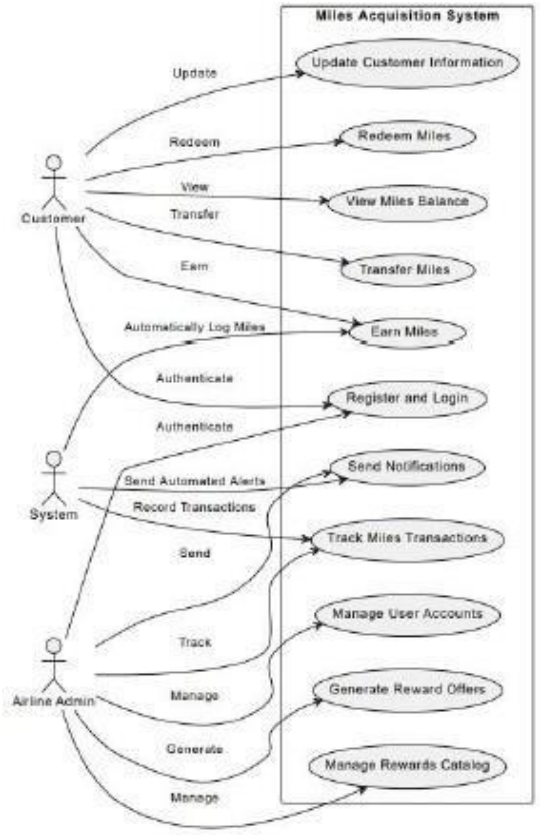

Figure 3.1: Use Case Diagram for Miles Acquisition System

3.3.2 Activity Diagram

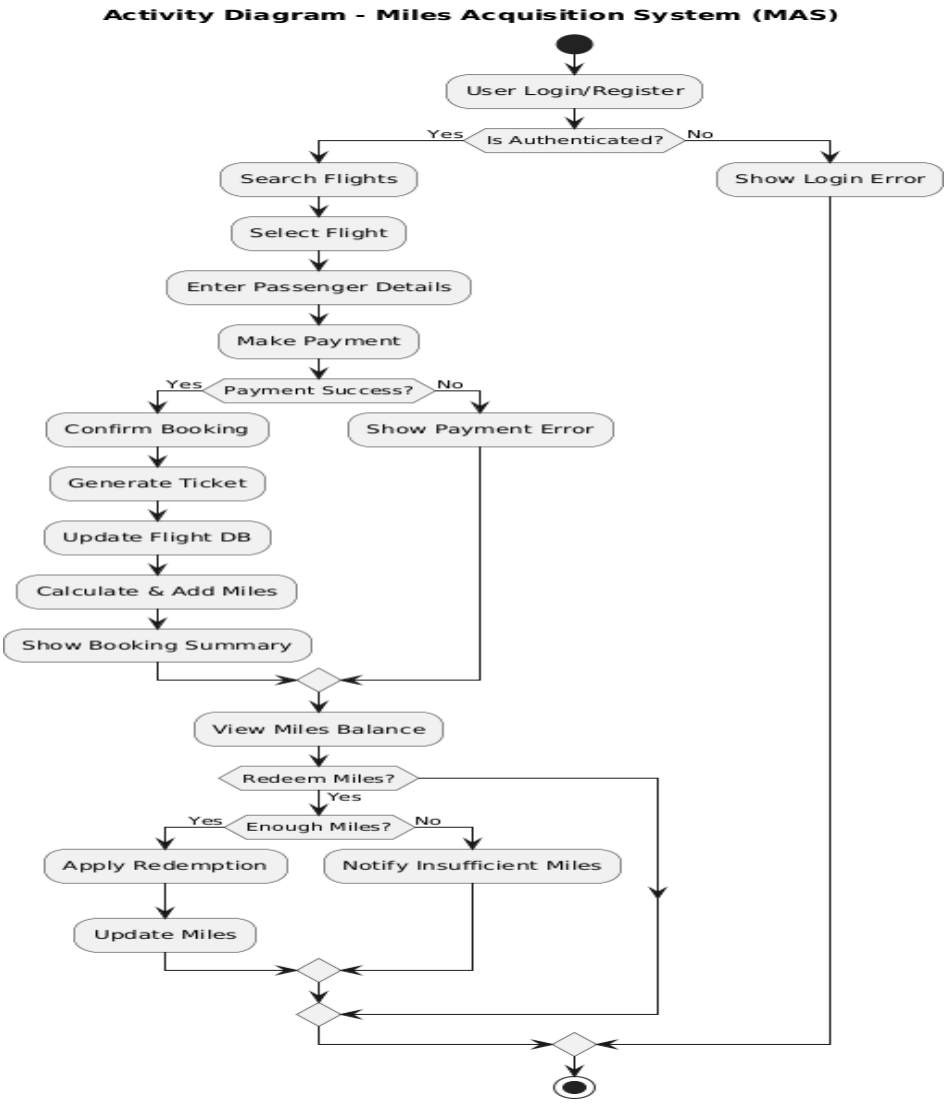

Figure 3.2: Activity Diagram for Miles Acquisition System

3.3.3 Data-Flow Diagram

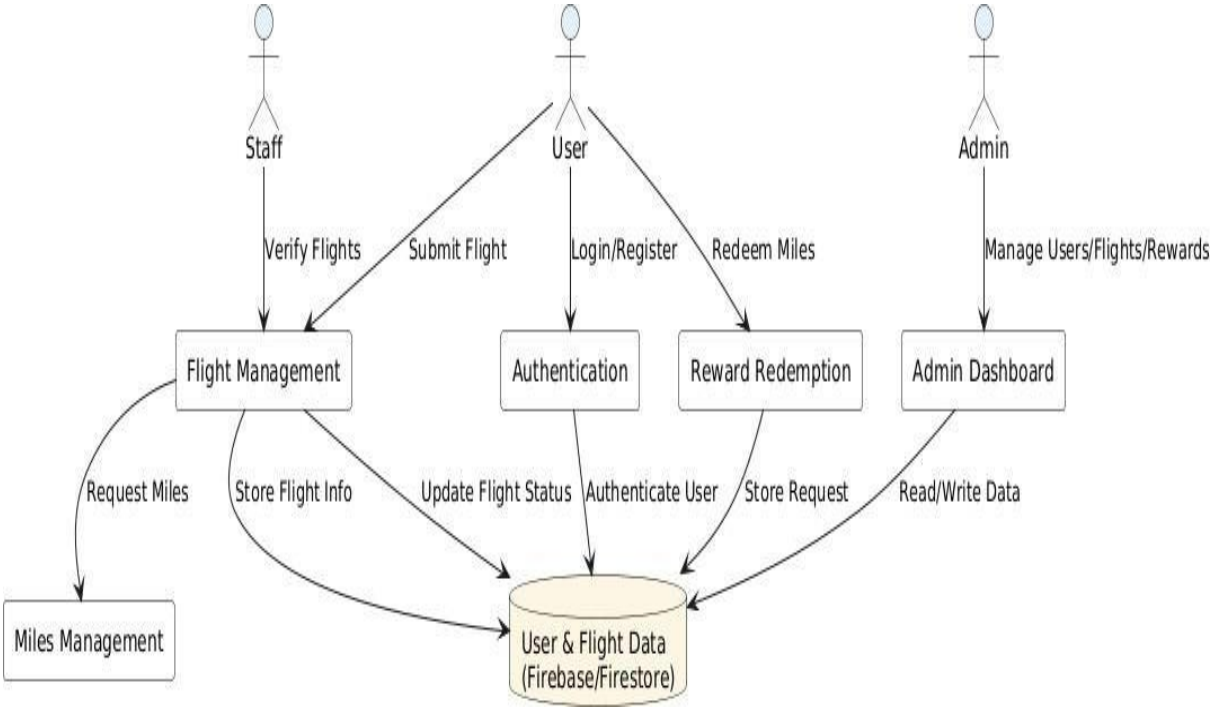

Figure 3.3: Data Flow Diagram for Miles Acquisition System

3.3.4 Class Diagram

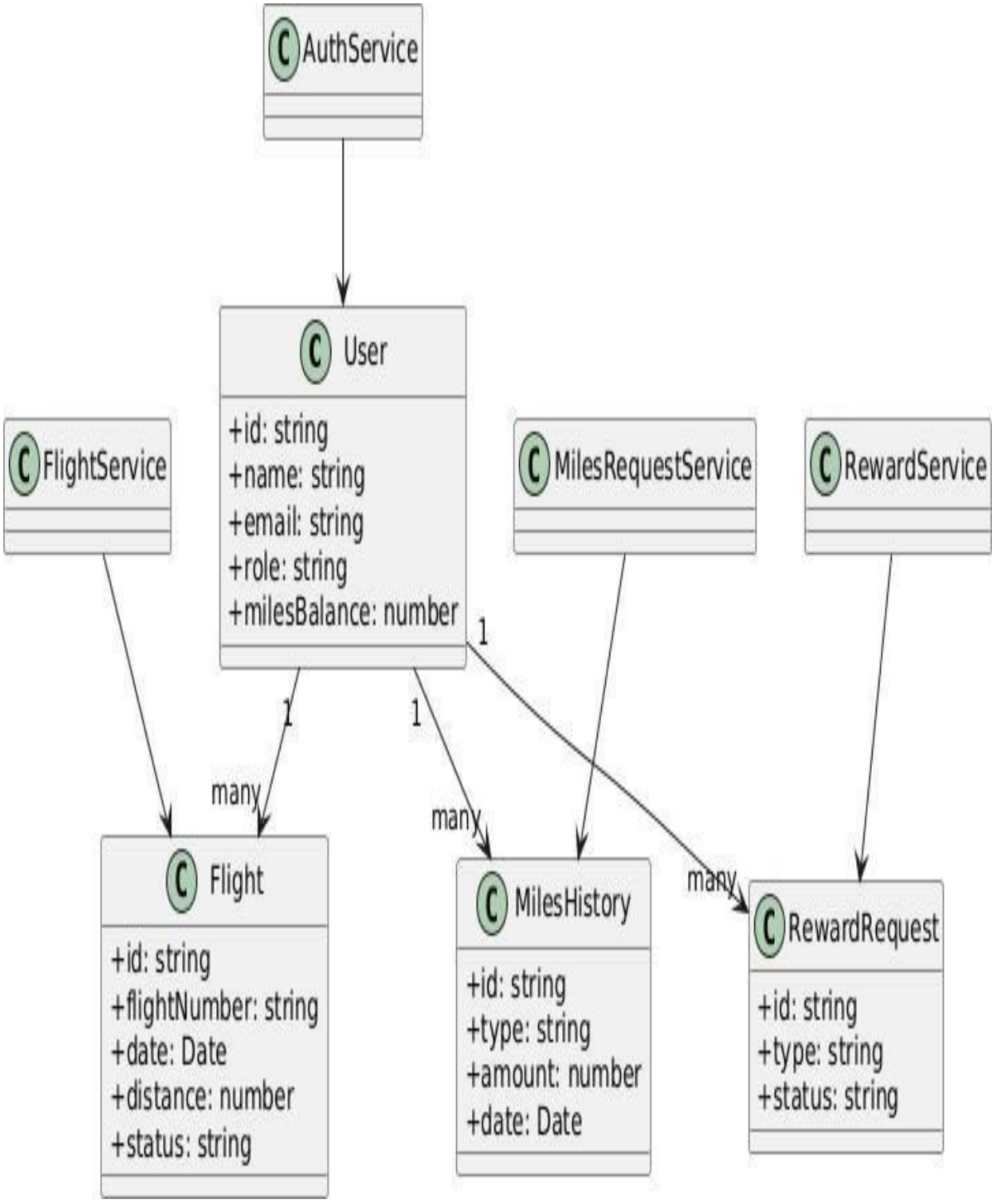

Figure 3.4: Class Diagram for Miles Acquisition System

3.3.5 ER-Diagram

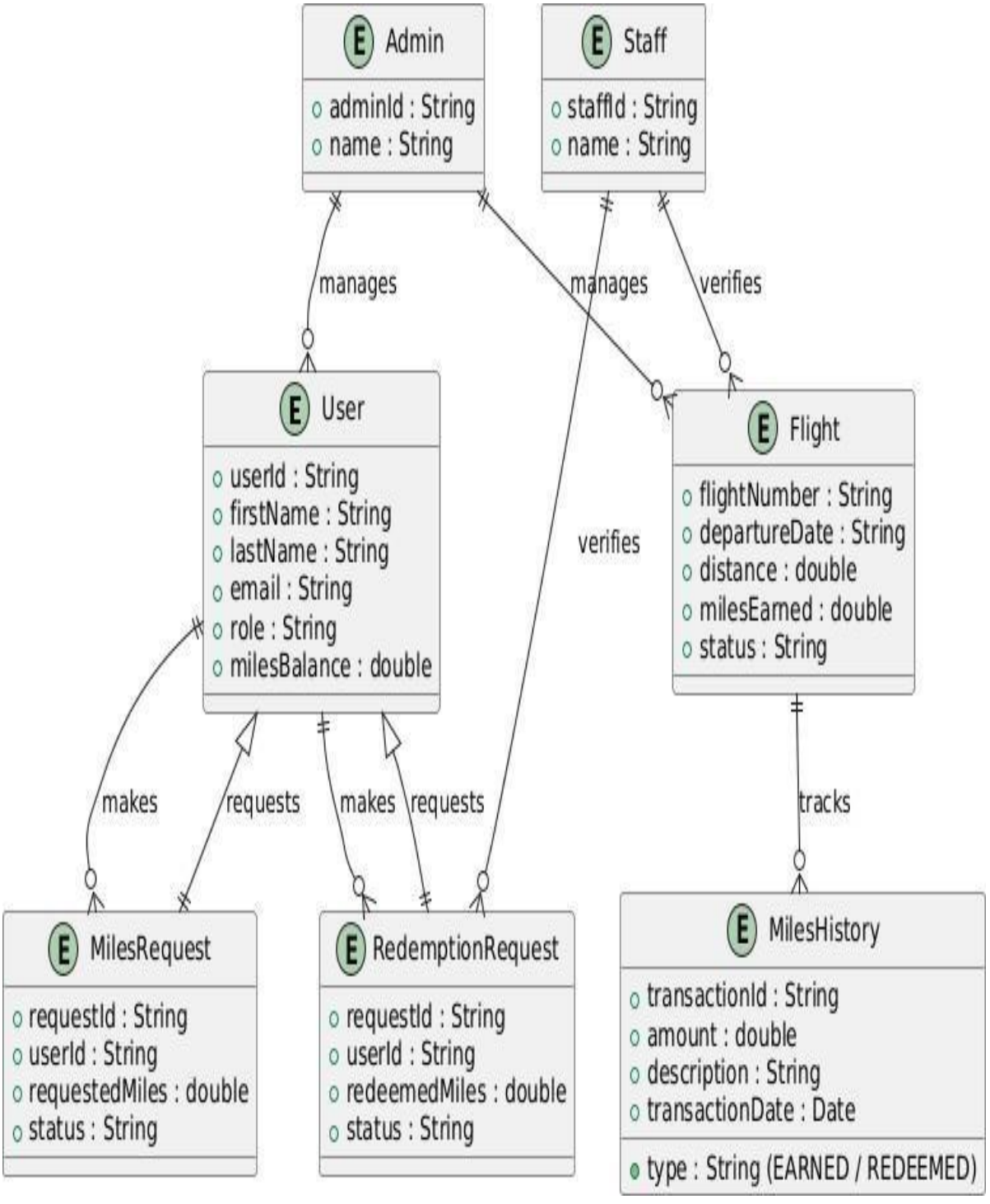

Figure 3.5: Entity-Relationship Diagram for Miles Acquisition System

Chapter 4
Methodology and Team

4.1 Introduction to Agile Framework

Agile is an iterative and incremental software development methodology that emphasizes flexibility, collaboration, and rapid delivery of working software. Unlike the linear Waterfall model, Agile divides the project into short, time-boxed iterations called “sprints” (typically 1–4 weeks), each producing a potentially shippable product increment.

Key characteristics of Agile include:

- **Iterative Development:** Features are developed in small slices and continuously refined.
- **Customer Collaboration:** Stakeholders provide feedback at the end of every sprint to steer priorities.
- **Adaptive Planning:** Plans are revisited and adjusted each sprint based on working software and feedback.
- **Cross-functional Teams:** Team members with diverse skills deliver end-to-end functionality together.
- **Continuous Improvement:** Regular retrospectives identify process enhancements for the next sprint.

A typical Agile sprint cycle comprises:

1. **Product Backlog Refinement:** The Product Owner updates and prioritizes user stories and tasks.
2. **Sprint Planning:** The team selects a set of high-priority backlog items, breaks them into tasks, and commits to a sprint goal.
3. **Development & Daily Stand-up:** Team members implement and test features while meeting daily to synchronize and remove blockers.
4. **Sprint Review:** The team demos the working increment to stakeholders for feedback and acceptance.
5. **Sprint Retrospective:** The team reflects on what went well, what didn't, and agrees on actionable improvements.
6. **Release (Optional):** After one or more sprints, a stable product increment may be deployed to end users.

The diagram below illustrates the flow of an Agile sprint cycle:

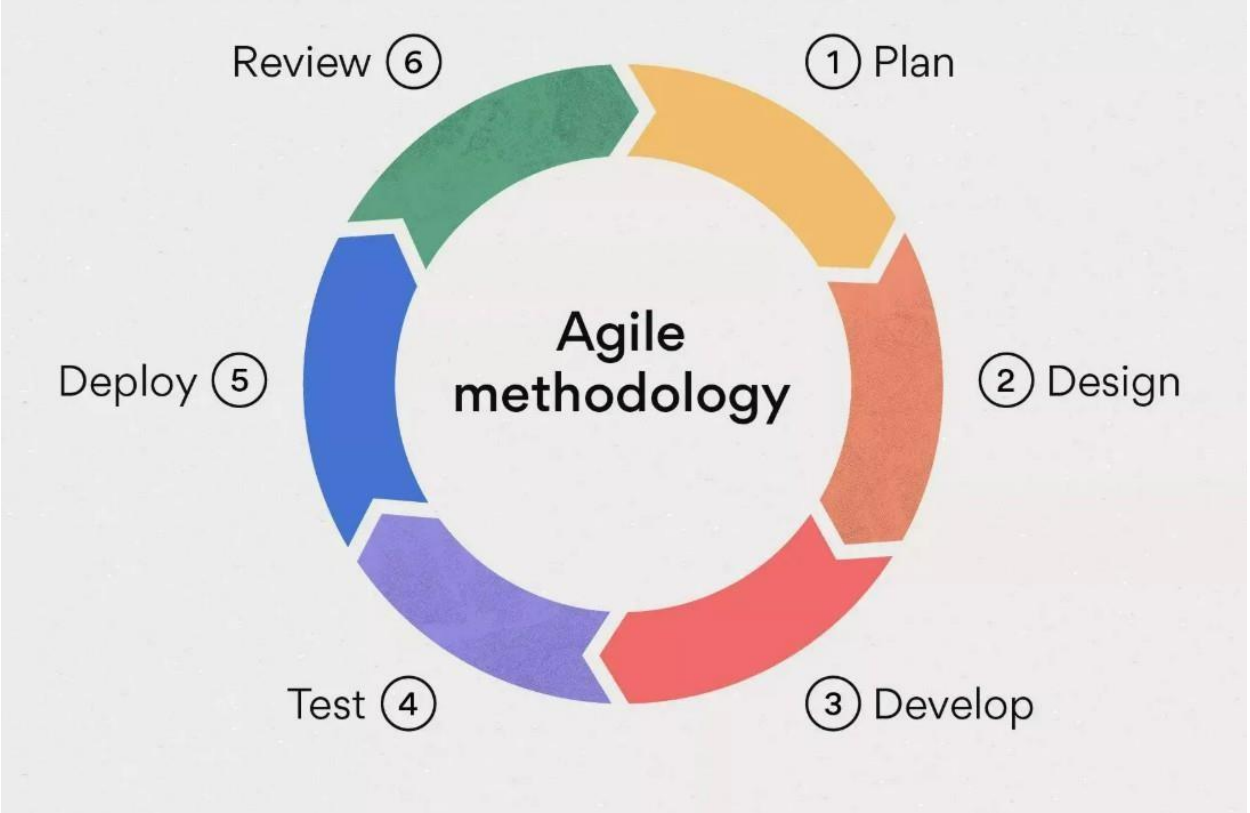

Figure 4.1: Agile Sprint Cycle (adapted diagram)

4.2 Team Members, Roles & Responsibilities

Tanishtha Sharma- Model Development for Facial Emotion

Yashasvi Gupta - Frontend Development

Vivek Shekhawat - Spotify API Integration

Chapter 5
Centering System Testing

The Miles Acquisition System (MAS) was thoroughly tested to ensure its functionality, performance, and usability across multiple roles and scenarios in real-world usage.

5.1 Functionality Testing

The following functional components of MAS were tested for correctness and completeness:

1. Links

   (a) Internal Links: Navigation between dashboards (User, Staff, Admin), and between components (e.g., flight submission, approval panel, redemption requests) was tested. Angular routing was verified for proper redirection and route guards based on authentication status.

   (b) External Links: The system uses Firebase Authentication and backend services. OAuth sign-in (e.g., with Google) and access links to external Firebase console resources were tested and verified.

   (c) Broken Links: All hyperlinks, navigation items, and buttons were manually tested to ensure no broken links exist. No navigation errors or inaccessible routes were found.

2. Forms

   (a) Error Message for Wrong Input: Forms like flight submission and login display accurate error messages for invalid or missing inputs. For instance, submitting a form without uploading a ticket or without required fields triggers user-friendly validation alerts.

   (b) Optional and Mandatory Fields: Mandatory fields (e.g., flight number, date, proof of travel) are enforced via Angular's reactive form validation. Users cannot proceed unless all required fields are correctly filled.

3. Database

   MAS uses Firebase Firestore as a backend. Functionality testing included successful data storage, retrieval, and update operations across collections such as 'Users', 'Flights', 'Miles', and 'Redemptions'. Test cases validated correct CRUD operations and role-based access.

5.2 Performance Testing

The system's performance was assessed across multiple devices and scenarios to ensure consistent operation:

- **Form Responsiveness:** Angular reactive forms responded promptly to user inputs and validation triggers. No noticeable delays occurred on machines with 4–8 GB RAM.

- **Firebase Query Latency:** Read and write operations on Firestore were completed in under 1 second in most cases. Real-time updates on admin and staff dashboards occurred with negligible lag.
- **System Load Handling:** The application was tested under simulated multi-user access. Firebase handled concurrent read/write operations reliably, and no crashes or timeout issues were encountered during extended sessions.

5.3 Usability Testing

Usability testing aimed to confirm that users of all roles (User, Staff, Admin) could interact with MAS easily and intuitively. Feedback was gathered from early testers and interns during system development.

- **User Interface Intuitiveness:** Each dashboard (User/Staff/Admin) was clearly laid out. Testers could submit flight details, approve requests, or manage miles without training. Logical grouping and consistent color coding helped differentiate components.
- **User Feedback:** A small test group of 10 individuals tested various workflows. Users appreciated the structured design, quick form validation, and the clarity of process flows from submission to mile redemption.
- **Learnability and Efficiency:** Most testers were able to operate the system within 5 minutes of exposure. Repeat tasks (e.g., submitting flights, reviewing status) took less than 1 minute, indicating high usability.

- **Accessibility:** The application was designed with cross-device compatibility in mind. All components worked on standard laptops and tablets with Chrome/Firefox. Error prompts and alerts used simple language for better accessibility.
- **Error Recovery and Helpfulness:** Forms gave immediate feedback when input errors occurred. Missing attachments or invalid entries were caught and displayed with clear resolution steps.
- **Aesthetic and Minimal Design:** The application followed a clean UI layout using Angular's material components and card-based structure. Key actions were highlighted, and visual clutter was minimized for a professional, modern look.
- **Role Separation and Clarity:** The interface clearly distinguished between user types (User, Staff, Admin). Each role had access only to its intended features, making navigation easier and reducing confusion.

Chapter 6

Test Execution Summary

The Test Execution Summary Report presents a detailed overview of the results obtained during the testing phase of the Miles Acquisition System (MAS). This report assesses the system's performance against its defined test cases, confirming that each module behaves as expected under diverse conditions. The purpose is to verify that the final product meets all functional, technical, and usability requirements.

Testing followed a structured methodology aligned with the MAS project objectives. Core functionalities such as flight submission, approval workflows, mile crediting, redemption requests, and Firebase integration were rigorously validated. Real-time feedback, authentication flows, and user role separation were especially tested. Manual and tool-assisted testing techniques were used in controlled environments to simulate practical use cases and edge conditions.

The following elements are covered in this Test Summary Report:

1. Test Case Identifiers
2. Detailed objective for each test
3. Execution outcome (Pass/Fail)
4. System/network usage during testing
5. Key observations and insights during execution

S.No	Test Case ID	Test Case Description	Test Status	Resources Consumed
1	TC01	Firebase Authentication: Email/-password and Google login validation	Passed	80 MB RAM, ¡1s latency
2	TC02	Flight submission by user with required fields and file upload	Passed	90 MB RAM, 5% CPU
3	TC03	Admin approval of user-submitted flight and mile assignment workflow	Passed	100 MB RAM, Firestore write/read
4	TC04	Redemption request by user and processing by staff role	Passed	95 MB RAM, 4% CPU
5	TC05	Dashboard access restricted by user role (User/Staff/Admin)	Passed	Role-based route protection validated
6	TC06	Error handling on invalid form submission (missing required fields)	Passed	Form validation alerts shown
7	TC07	Realtime update of dashboard after status change (approved/rejected)	Passed	Firebase real-time sync, ¡500ms delay
8	TC08	Responsive layout test on Chrome, Firefox, and Edge (desktop/mobile)	Passed	No UI distortion, consistent performance
9	TC09	File upload size validation and preview (ticket image/PDF)	Passed	Max 5MB handled, preview displayed
10	TC10	Form reset functionality after successful submission	Passed	State cleared, inputs reset as expected
11	TC11	Verification of miles added per flight using backend logic	Passed	Correct miles reflected in Firestore
12	TC12	Session handling across login/logout flows	Passed	Auto-redirect on logout, session cleared
13	TC13	Email notifications sent for status updates (future enhancement placeholder)	Skipped	Feature not yet implemented
14	TC14	Simultaneous submission by multiple users without collision	Passed	3 concurrent sessions handled smoothly

Table 6.1: Test Case Execution Summary for Miles Acquisition System (MAS)

All implemented test cases executed successfully. No major functional or UI errors were found during evaluation. Firebase integrations were stable, real-time data flow was effective, and the role-based workflows were correctly enforced.

Based on these outcomes, MAS is deemed stable for production deployment. Core components such as authentication, data submis-

sion, approvals, and redemptions were validated for both functionality and performance. The testing results confirm that MAS meets the desired objectives for a secure, scalable, and user-friendly airline mileage management platform.

The successful test cycle reflects the robustness of the application's architecture and its alignment with user expectations. Exception handling, responsive design, and dynamic updates contributed to a reliable and intuitive user experience. Minor layout refinements and error prompts that arose during early tests were resolved in the final build. The system consistently delivered accurate data flow between users, staff, and admin modules, ensuring transparency and traceability across transactions.

Chapter 7

Project Screen Shots

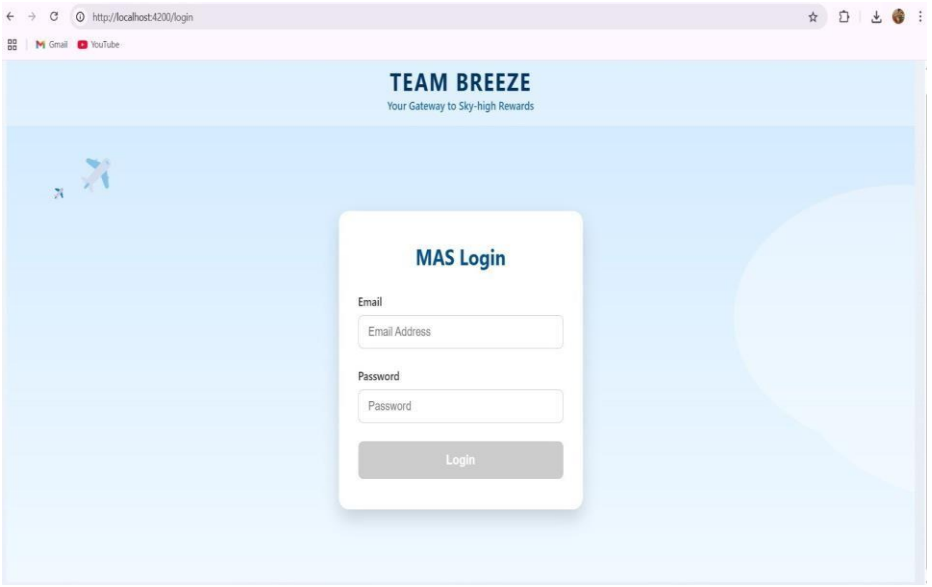

Figure 7.1: login page

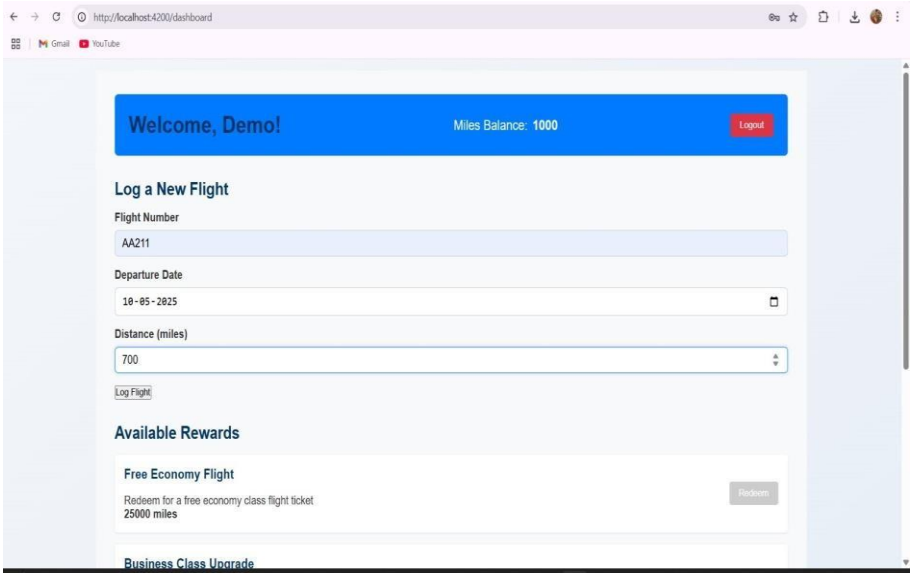

Figure 7.2: User Dashboard

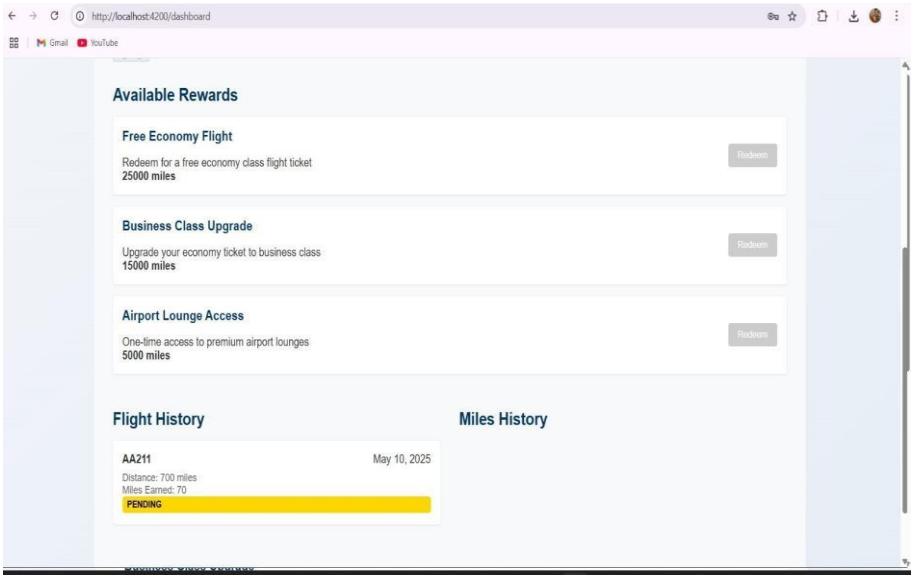

Figure 7.3: User Dashboard

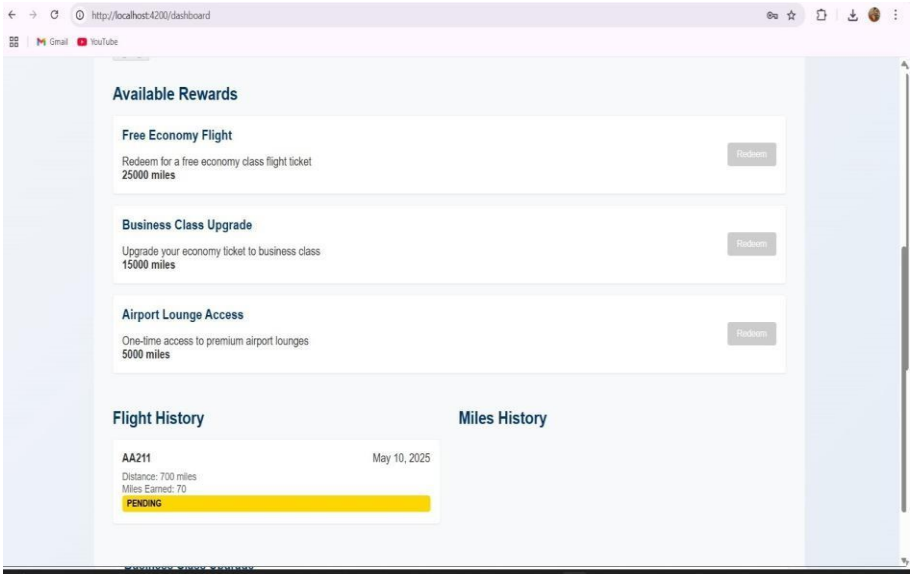

Figure 7.4: User Dashboard

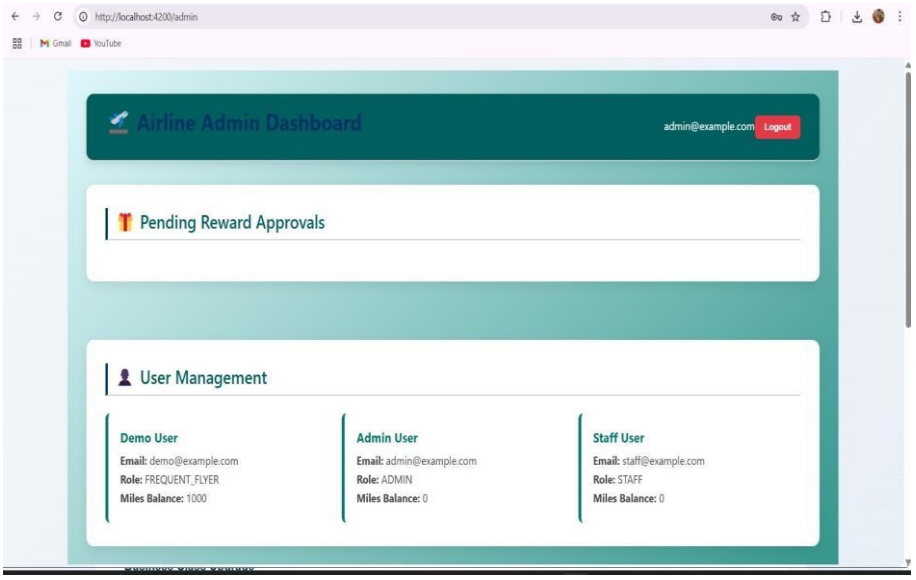

Figure 7.5: Admin Dashboard

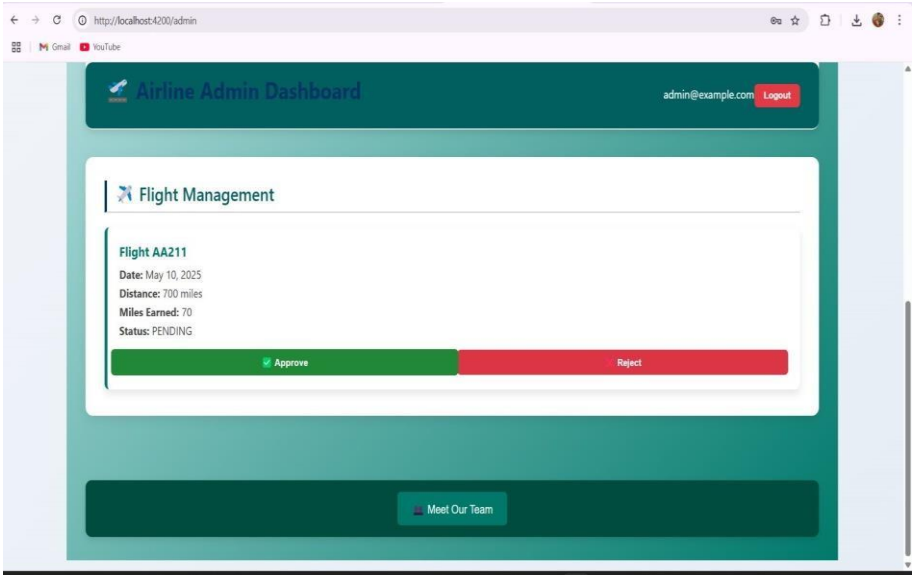

Figure 7.6: Admin Dashboard

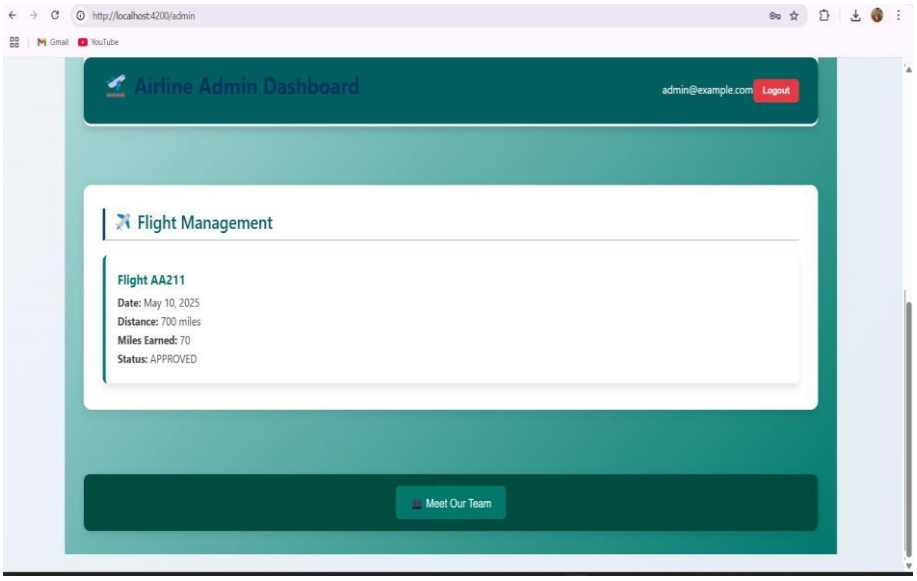

Figure 7.7: Admin Dashboard

Chapter 8
Project Summary and Conclusions

8.1 Project Summary

The Miles Acquisition System (MAS) is a comprehensive web-based platform developed to streamline the process of airline mileage tracking, accumulation, and redemption. Designed for both passengers and administrative staff, MAS automates the submission, approval, and crediting of flight miles based on user activity. The system ensures transparency, efficiency, and ease of use through an intuitive interface and role-based access control.

The application is built using Angular for the front end, Firebase for authentication and real-time database interactions, and Google Cloud services to manage storage and access rights. Users can submit their flight details along with ticket attachments, while staff and admin roles are responsible for reviewing and approving the claims. Once approved, the miles are credited to the user's account, and users can subsequently request redemption of these miles.

Key features of the system include:

- Firebase Authentication supporting both email-password and Google login.
- Flight submission module with ticket upload capability.
- Role-based dashboards for User, Staff, and Admin.

- Real-time update of mile statuses and dynamic rendering of dashboards.
- Mile redemption workflows with status tracking.
- Responsive design suitable for desktop and mobile interfaces.

Throughout the development process, emphasis was placed on ensuring a seamless user experience, robust data handling, and system security. Special attention was given to error handling, form validations, and responsiveness across platforms. The MAS project successfully fulfills its intended goal of digitizing and simplifying the miles acquisition and redemption process.

8.2 Conclusion

In conclusion, the Miles Acquisition System (MAS) effectively meets the core objectives of digitizing the airline miles tracking and management workflow. It offers a reliable, user-friendly interface for passengers to submit their travel details and earn miles, while simultaneously providing administrative staff with tools to process and manage these submissions efficiently. The system eliminates manual paperwork, reduces processing time, and ensures better transparency across the user journey.

The application's architecture leverages Firebase's real-time database and authentication features to ensure fast, secure, and scalable operations. Angular was chosen for its modular design and component-based development approach, which enabled the creation of dynamic and interactive interfaces. Role segregation ensures that data access

is secure and actions are appropriately restricted based on user privileges.

Throughout testing and implementation, the system performed consistently well, demonstrating robustness under both normal and edge-case scenarios. It handled multiple concurrent user sessions, dynamic updates to data, and file uploads without significant performance degradation. The user interface remained responsive and intuitive, even when tested on varying screen sizes and browser environments.

Challenges such as validating complex form inputs, managing asynchronous operations across Firebase, and ensuring smooth user navigation were successfully addressed through effective coding practices and modular design. Furthermore, the system includes fallback mechanisms for interrupted sessions, missing data, or incorrect user roles, thereby ensuring a fail-safe operation.

While the project has achieved all functional milestones, future enhancements can include:

- Integration with third-party airline APIs for automatic ticket validation.
- Notification system for approvals, rejections, and redemption updates.
- Exportable reports for miles history and usage analytics.
- Offline form submission support and progressive web app capabilities.
- Enhanced security via multi-factor authentication and activity log-

ging.

Ultimately, MAS serves as a foundational tool for organizations looking to automate and optimize their frequent flyer management systems. Its scalability, responsiveness, and modular design make it a strong candidate for further enhancement and potential deployment in real-world airline or corporate environments. The successful implementation of MAS underscores the value of full-stack development and cloud integration in building modern web applications that are efficient, reliable, and user-centric.

Chapter 9
Future Scope

The Miles Acquisition System (MAS) has successfully achieved its core objective of streamlining airline mile submissions, approval, and redemption processes for users and staff. However, to further enhance its capabilities and make it more scalable, secure, and user-friendly, there are several areas where the system can be improved and extended. These future enhancements would help MAS cater to a broader user base, increase system automation, and provide a more robust user experience.

- **Integration with Airline APIs:** Future versions of MAS can integrate directly with airline booking APIs to automatically validate ticket submissions and flight details, reducing the need for manual entry and verification.
- **Advanced Notification System:** Implementing real-time notifications (email, SMS, or push notifications) for events such as submission status updates, approval alerts, and redemption confirmations would greatly enhance user communication and engagement.
- **Detailed Analytics and Reports:** Adding reporting features for users and administrators can help track miles earned, redemption trends, flight frequencies, and staff performance. Visual dash-

boards can assist in strategic decision-making.

- **Mobile Application Development:** Developing native mobile applications for Android and iOS platforms would allow users to access MAS on-the-go, improving accessibility and convenience.
- **Offline Submission Capability:** Enabling offline form entry that syncs automatically when internet connectivity is restored would improve usability in low-connectivity regions.
- **Multi-language Support:** Supporting multiple languages would make MAS accessible to a diverse set of users across different regions and backgrounds.
- **Enhanced Security Features:** Future improvements could include multi-factor authentication, access logs, and data encryption at rest to strengthen system security and build user trust.
- **Gamification and Loyalty Enhancements:** Introducing gamification elements like mile-earning challenges, badges, and leaderboards can increase user engagement and loyalty.
- **Expanded Role Management:** Introducing finer-grained roles and permissions (e.g., reviewers, auditors) would support more complex organizational workflows and ensure accountability.
- **Redemption Marketplace:** A future extension could allow users to redeem their miles not just for flights but also for merchandise, vouchers, or services via a dedicated marketplace.

References

[1] *Miles Acquisition System (MAS) Project Overview*; Yashasvi Gupta, Ensylon Pvt. Ltd., April 2025

[2] *Angular and Firebase Integration – Official Documentation*; Angular.io and Firebase.google.com, Accessed February 2025

[3] *"Cloud-Based Airline Customer Loyalty Management System" – [1]. A. Singh and R. Patel, "A Secure and Scalable Cloud System for Airline Reward Programs", IEEE International Conference on Smart Computing, pp. 334–340, 2022*

[4] *"Web Technologies: HTML, CSS, JavaScript and AngularJS"*; Uttam K. Roy, Oxford University Press, 2nd Edition, 2021

[5] *Firebase Authentication and Firestore Security Rules*; Google Firebase Documentation, `https://firebase.google.com/docs/rules`

[6] *Software Engineering: A Practitioner's Approach*; Roger S. Pressman and Bruce R. Maxim, McGraw-Hill Education, 8th Edition, 2019

[7] `https://research-methodology.net`