PES UNIVERSITY

Department of Computer Science and Engineering

UE22CS341A: Software Engineering

PES UNIVERSITY
Department of Computer Science and Engineering
UE22CS341A: Software Engineering

Software Requirements Specification

for

INTERNATIONAL AIRPORT MANAGEMENT SYSTEM

Prepared by:

Nitheesh Pugazhanthi: PES2UG22CS371
Nikhil Srivatsa: PES2UG22CS357

PES UNIVERSITY

10-09-2024

PES UNIVERSITY

BENGALURU

PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Table of Contents

Introduction

Purpose

*The purpose of the **Airport Management System** project is to develop a comprehensive software solution that enhances the overall management of airport operations. This system is designed to improve passenger experiences by reducing wait times and offering dynamic re-routing options, while also increasing operational efficiency for airport management staff and security teams.*

Intended Audience

1. *Developers: To understand the functional and non-functional requirements of the system, guiding them in the design, coding, and testing processes.*

2. *Project Managers: To track the project's progress, ensure that the software meets the defined requirements, and allocate resources effectively.*

3. *Marketing Staff: To gain insight into the key features and benefits of the AMS, enabling them to develop effective strategies for promoting the software to potential clients, including airports and airlines.*

4. *Users: People like the security and management staff need to read to familiarize themselves with the functionality of the Airport Management system, understand how it will enhance their daily tasks, and prepare for eventual training and use.*

5. *Testers: To create test plans, test cases, and testing strategies based on the specified requirements, ensuring the software meets quality standards and functions as intended.*

6. *Documentation writers: To write an efficient and concise documentation for the project the documentation writers need to have a through understanding of the SRS document.*

Product Scope

*The **Airport Management System** is a software solution designed to optimize and streamline airport operations, focusing on enhancing passenger experiences, improving security measures, and increasing operational efficiency. The Airport Management System provides real-time flight tracking, automates passenger check-in, facilitates data analysis for strategic decision-making, and offers dynamic re-routing options in case of flight cancellations.*

The software aligns with corporate goals of improving customer satisfaction, reducing operational costs, and maximizing profitability for airlines. By adhering to international aviation standards, the Airport Management System also supports airports in maintaining compliance with regulatory requirements and enhancing their reputation in the global market.

Overall Description

Product Perspective

*The **Airport Management System (AMS)** is a new, self-contained software solution designed to enhance the management of airport operations and improve the passenger experience. This product is not a follow-on member of an existing product family, nor is it intended as a replacement for any existing system. Rather, it introduces a comprehensive, integrated approach to managing key airport functions, such as flight tracking, passenger check-in, security screening, and data analysis.*

The AMS operates as a standalone system but is designed to interface seamlessly with existing airport infrastructure, including:

- ***Airline Management Systems:** To exchange flight schedules, passenger information, and other critical data required for real-time flight tracking, check-in processes, and dynamic re-routing.*
- ***Security Systems:** To identify flagged passengers during check-in and security checks, integrating with biometric scanning, passport control, and surveillance systems.*
- ***Passenger Information Systems (PIS):** To provide real-time updates on flight statuses, gate changes, delays, and other relevant information to passengers through digital screens, mobile applications, and public address systems.*

Product Functions

1) *Real time tracking*
2) *Passenger management*
3) *Security checks*
4) *Data analysis*
5) *Easy re-routing*
6) *Lower price and future planning*

User Classes and Characteristics

1. ***Airport Management Staff***

- ***Frequency of Use:*** *Daily*
- ***Functions Used:*** *Real-time flight tracking, passenger management, dynamic re-routing, and data analysis.*
- ***Technical Expertise:*** *Moderate; familiarity with airport operations and basic computer skills.*
- ***Security/Privilege Level:*** *High; access to comprehensive system functions, including management dashboards and reporting tools.*
- ***Characteristics:*** *Responsible for monitoring and managing daily airport operations, ensuring efficient passenger flow, handling cancellations and re-routing, and making strategic decisions based on data insights.*
- ***Importance:*** *High; primary users responsible for the day-to-day functioning of the airport.*

2. ***Security Personnel***

- ***Frequency of Use:*** *Daily*
- ***Functions Used:*** *Security management, real-time alerts, flagged passenger identification.*
- ***Technical Expertise:*** *Basic; familiarity with security protocols and standard software interfaces.*
- ***Security/Privilege Level:*** *High; access to sensitive data regarding flagged passengers and real-time security alerts.*

- ***Characteristics:*** *Tasked with identifying and handling flagged passengers, ensuring compliance with security protocols, and responding to potential threats.*
- ***Importance:*** *High; critical for maintaining airport security and safety.*

3. ***Airline Staff***
   - ***Frequency of Use:*** *Daily*
   - ***Functions Used:*** *Real-time flight tracking, passenger check-in, dynamic re-routing, and pricing and future planning.*
   - ***Technical Expertise:*** *Moderate; understanding of airline operations and basic data entry.*
   - ***Security/Privilege Level:*** *Medium; access to flight-related data and passenger management tools.*
   - ***Characteristics:*** *Responsible for managing airline-specific operations, such as check-ins, boarding, and assisting passengers with re-routing and flight changes.*
   - ***Importance:*** *Medium; essential for ensuring smooth airline-specific operations and customer service.*

4. ***IT Support Staff***
   - ***Frequency of Use:*** *Occasional*
   - ***Functions Used:*** *System maintenance, troubleshooting, user support, and security management.*
   - ***Technical Expertise:*** *High; advanced knowledge of software systems, databases, and network infrastructure.*
   - ***Security/Privilege Level:*** *Very High; access to all system components for maintenance and troubleshooting purposes.*
   - ***Characteristics:*** *Provides technical support, ensures system stability, performs updates, and resolves any technical issues that arise.*
   - ***Importance:*** *Medium; critical for ensuring the system remains operational and secure.*

5. ***Passengers***
   - ***Frequency of Use:*** *Occasional*
   - ***Functions Used:*** *Flight status tracking, check-in, dynamic re-routing, and pricing and future planning.*
   - ***Technical Expertise:*** *Low; basic familiarity with digital interfaces (mobile apps or kiosks).*
   - ***Security/Privilege Level:*** *Low; access is restricted to personal data and flight-related information.*
   - ***Characteristics:*** *Users who interact with the system for checking flight status, performing self-check-in, receiving re-routing options, and planning future travel.*
   - ***Importance:*** *Low; indirect users whose experience is influenced by the system's performance.*

Operating Environment

Hardware Platform:

- ***Servers:*** *High-performance servers with at least 16-core CPUs, 32 GB RAM, and 1 TB SSD storage. These servers should be scalable to handle large volumes of data and numerous concurrent users.*

Operating System:

- ***Server OS:***
  - *Linux (Ubuntu Server 20.04 LTS or later, CentOS 8 or later) or*
  - *Windows Server (2016 or later).*

Database Management System:

- ***Database:***
  - *MySQL (8.0 or later)*

Browser Compatibility:

- *The web-based components of the AMS must be compatible with the latest versions of all major web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.*

Design and Implementation Constraints

- ***Data Privacy Laws:*** *Adherence to data protection regulations and other regional privacy laws. These regulations mandate secure data storage, transmission, and handling of passenger information.*

- ***Corporate IT Policies:*** *Conformity with the airport or airline's internal IT policies, which may dictate specific software development standards, security practices, or technology preferences.*

- ***Legacy Systems Integration:*** *The AMS must integrate with existing airport and airline systems, some of which may be outdated or lack modern interfaces. Developers may be limited by the hardware capabilities of these legacy systems, such as low processing power, limited memory, or lack of API support.*

- ***Memory and Storage Requirements:*** *Efficient use of memory and storage is necessary, especially when handling large volumes of data, such as flight schedules, passenger manifests, and security logs. Developers must optimize the software to function within the hardware limits of client machines and servers.*

- ***Access Control and Authentication:*** *The system must implement strict user authentication and role-based access controls to protect sensitive information. This includes integrating with existing Identity and Access Management systems.*

- ***Network Protocols:*** *The software must use secure communication protocols to ensure the confidentiality and integrity of data transmitted over the network. Developers must also ensure compatibility with existing network protocols used by airport systems.*

- ***Maintainability Requirements:*** *The software must be designed for easy maintenance and future updates. This may require modular design, extensive documentation, and adherence to architectural patterns like MVC (Model-View-Controller) or microservices.*

2.6 Assumptions and Dependencies

1. *Continuous Network Availability: It is assumed that the airport's network infrastructure will provide high availability with minimal downtime, ensuring uninterrupted access to the AMS for real-time updates and communication.*

2. *Stable Operating Environment: The operating environment (hardware, network infrastructure, and operating systems) is assumed to remain stable throughout the development and deployment phases. Any major changes or upgrades to the infrastructure could require adjustments to the AMS.*

3. *Adequate User Training and Support: The project assumes that end-users (such as airport staff, security personnel, and airline employees) will receive adequate training on how to use the AMS effectively.*

4. *Third-Party Software and Middleware: The project relies on third-party software components and middleware. Any changes in these components, such as licensing changes, version upgrades, or end-of-life announcements, could affect the project's timeline and requirements.*

5. *High-Level Security Protocols: It is assumed that the existing security protocols, such as firewalls, encryption standards, and intrusion detection systems, are adequate and will support the security requirements of the AMS.*

External Interface Requirements

User Interfaces

1. *The AMS will adhere to industry-standard GUI conventions to ensure consistency and ease of use. It will follow the airport's or airline's internal style guides, including color schemes, fonts, branding elements, and iconography, to maintain a consistent look and feel across all modules.*

2. ***Consistent Layout:*** *Each screen will have a consistent layout with a navigation bar, header, and footer. The navigation bar will allow users to quickly access different modules (e.g., Flight Status, Passenger Management, Security, and Data Analysis).*

3. ***Keyboard Shortcuts and Accessibility:*** *The interface will support keyboard shortcuts for frequent actions, such as accessing the main menu, searching, submitting forms, and navigating between fields. Accessibility features like screen reader compatibility, high contrast mode, and keyboard navigation will be implemented to assist users with disabilities.*

4. ***Passenger Check-in Kiosk:*** *Interface for self-service check-in, allowing passengers to scan their passports or booking codes and receive boarding passes.*
5. ***Airport Staff Dashboard:*** *Interface for airline and airport staff to manage daily operations, flight information, and passenger data.*

6. ***Security Terminal:*** *Interface for security personnel to view flagged passengers, alerts, and incident reports.*

Software Interfaces

*The **Airport Management System** interfaces with several software components:*

1. ***Databases:***
   - ***MySQL:*** *To store the relevant data required to perform all the necessary functions required for the correct functioning of the website.*
2. ***Operating Systems:***
   - ***Linux (Ubuntu 20.04):*** *Hosts server applications.*
   - ***Windows 11 :*** *Manages database operations and backend services.*
3. ***APIs and Libraries:***
   - ***Flight Data API:*** *Provides real-time flight status and schedules.*
   - ***Payment Gateway API:*** *Handles transaction processing for booking fees.*
4. ***Communication Protocols:***

- ***HTTPS/TLS:*** *Secures data transmission between components.*

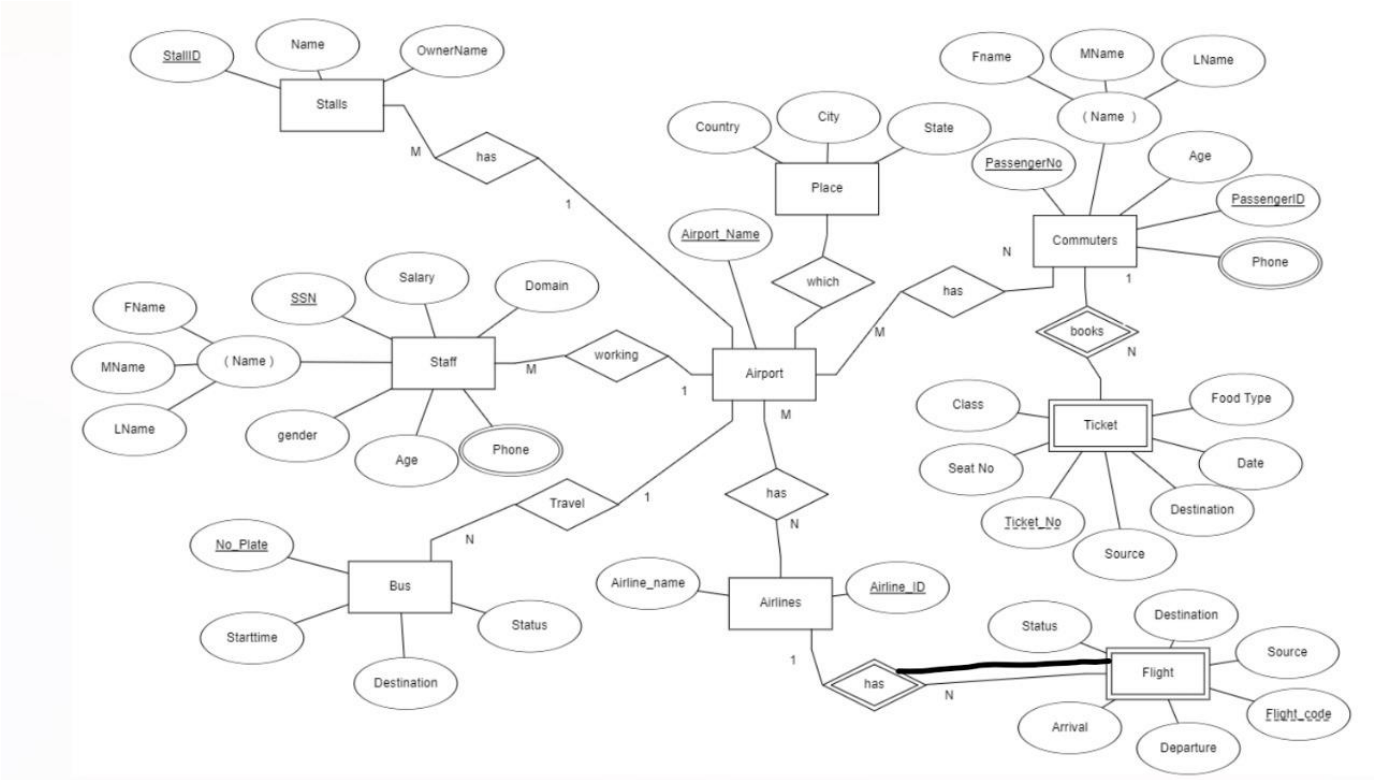
5. ***Data Sharing:***
    - ***Shared Data Items:*** *Flight details, passenger records, and security alerts.*

Communications Interfaces

*The **Airport Management System (AMS)** has several communications requirements:*

1. ***E-Mail:***
    - ***Function:*** *Sends automated notifications (e.g., flight status updates, security alerts) to users and administrators.*
    - ***Format:*** *Standardized email format with HTML and plain text versions for compatibility.*
2. ***Web Browser:***
    - ***Function:*** *Users access the AMS through a web browser.*
    - ***Standards:*** *Compatible with modern browsers (Chrome, Firefox, Edge) using HTML5, CSS3, and JavaScript(React, Node and associated frameworks).*
3. ***Network Server Communications Protocols:***
    - ***HTTP/HTTPS:*** *Used for secure web communication. HTTPS ensures encrypted data transmission.*
    - ***RESTful APIs:*** *For communication between AMS modules and external services (e.g., flight data, payment processing).*
4. ***Data Transfer:***
    - ***Encryption:*** *TLS 1.2 or higher for secure data transmission.*
    - ***Data Rates:*** *Optimized for low latency and high throughput to handle real-time updates.*
    - ***Synchronization:*** *Real-time data synchronization using WebSockets or RESTful API polling*

Analysis Models



System Features

Real Time Tracking:

Description and Priority

Real-time tracking allows the system to use logged data in the database to provide the current status of flights. This feature enables users to query flight data, view live updates on flight positions, delays, and estimated arrival times. Priority: High.

- ***Benefit**: 8*
- ***Penalty**: 7*
- ***Cost**: 6*
- ***Risk**: 5*

Stimulus/Response Sequences

1. ***User Action**: User queries the flight status by entering a flight number or selecting from a list of active flights.*
    - ***System Response**: The system retrieves and displays the real-time flight status, including current location, estimated time of arrival (ETA), and any delays.*
2. ***User Action**: User requests live tracking of a specific flight.*

- ***System Response**: The system initiates real-time tracking, updating the flight's position, speed, and altitude on a map at regular intervals.*

Functional Requirements

- *The system must query the database to retrieve and display flight status updates in real time.*
- *The system must allow users to enter flight numbers or select active flights for tracking.*
- *The system must update flight information at regular intervals to provide real-time status.*
- *In the event of invalid flight numbers or inactive flights, the system must return an error message and prompt the user to try again.*
- *The system must handle database connection errors gracefully, notifying the user and logging the error for system administrators*.

Passenger Management:

Description and Priority

Passenger management streamlines the check-in process by automating key steps, reducing wait times, and enhancing the flow of passengers through the airport. This feature allows passengers to check in efficiently via self-service counters or online platforms. Priority: High.

- ***Benefit**: 9*
- ***Penalty**: 8*
- ***Cost**: 5*
- ***Risk**: 4*

Stimulus/Response Sequences

1. ***User Action**: Passenger selects the check-in option online or at a counter.*
    - ***System Response**: The system prompts the passenger for flight details and personal information (e.g., passport, ticket number), and verifies this information against the database.*
2. ***User Action**: Passenger confirms the check-in and selects seating or baggage options.*
    - ***System Response**: The system updates the database with the passenger's check-in status, assigns a seat, and issues a digital or physical boarding pass.*
3. ***User Action**: Passenger requests changes to seat or baggage options.*
    - ***System Response**: The system processes the request and updates the records accordingly, while ensuring any applicable fees are charged.*

Functional Requirements

- *The system must verify passenger details (e.g., flight number, passport) during check-in.*
- *The system must allow passengers to select seats and add baggage options during check-in.*
- *The system must update the database in real time with check-in status and passenger information.*
- *The system must handle invalid ticket or passenger details by displaying an error message and offering corrective suggestions.*

Security Checks:

Description and Priority

Security checks ensure that flagged passengers, whose passports have been marked for suspicion, are identified during the check-in process. This feature integrates with the database to alert authorities when such individuals attempt to check in, enhancing security and crime prevention. Priority: High.

- ***Benefit**: 9*
- ***Penalty**: 9*
- ***Cost**: 6*
- ***Risk**: 5*

Stimulus/Response Sequences

1. ***User Action**: Passenger attempts to check in, either online or at a counter.*
    - ***System Response**: The system cross-references the passenger's passport information with the flagged records in the database.*
2. ***User Action**: Flagged passenger attempts to proceed with check-in.*
    - ***System Response**: The system alerts security personnel, prevents further check-in actions, and notifies relevant authorities.*
3. ***User Action**: Non-flagged passenger proceeds with check-in.*
    - ***System Response**: The system completes the check-in process without any security interruptions.*

Functional Requirements

- *The system must verify each passenger's passport information against a flagged records database during check-in.*
- *The system must issue an alert and prevent further check-in actions if a flagged passenger is identified.*
- *The system must notify airport security personnel and relevant authorities when a flagged passenger is detected.*
- *The system must handle normal check-ins without interruptions for passengers who are not flagged.*

Easy Re-routing:

Description and Priority

Easy re-routing allows the system to assist passengers when their flight is canceled by analyzing available flights across multiple airlines and suggesting alternative routes for the same day, minimizing delays. This feature enhances customer satisfaction by reducing the inconvenience of cancellations. Priority: Medium.

- ***Benefit**: 7*
- ***Penalty**: 6*
- ***Cost**: 7*
- ***Risk**: 4*

Stimulus/Response Sequences

1. ***User Action**: Passenger receives a notification that their flight has been canceled.*
    - ***System Response**: The system automatically searches for alternative flights departing the same day across all available airlines.*

2. ***User Action****: Passenger reviews the suggested alternative routes and selects a preferred option.*
    - ***System Response****: The system books the passenger on the new flight and updates the reservation details in the database.*
3. ***User Action****: Passenger requests a refund instead of re-routing.*
    - ***System Response****: The system processes the refund request according to airline policies.*

Functional Requirements

- *The system must search for alternative flights across all available airlines when a flight is cancelled.*
- *The system must prioritize flight options that ensure the passenger reaches their destination on the same day.*
- *The system must allow passengers to select from suggested alternative routes or request a refund.*
- *The system must update the booking details with the new flight information once the passenger selects an alternative.*
- *In case of database or search failures, the system must notify the passenger and offer alternative solutions, such as manual re-routing or contacting customer service.*

Lower price and Future Planning:

Description and Priority

Lower price and future planning allows customers to view historical and projected price trends for flights, helping them identify the best time to book. By offering insights into price fluctuations, customers can make more informed decisions, potentially saving money. Priority: Medium.

- ***Benefit****: 7*
- ***Penalty****: 5*
- ***Cost****: 6*
- ***Risk****: 4*

Stimulus/Response Sequences

1. ***User Action****: Customer selects a flight route and searches for ticket prices.*
    - ***System Response****: The system displays price trends over a range of dates, highlighting periods where prices are lower or higher than average.*
2. ***User Action****: Customer reviews the price forecast for a specific flight and selects a date based on the lower price.*
    - ***System Response****: The system confirms the selection, updates the booking details, and provides a summary of the cost-saving benefits.*
3. ***User Action****: Customer requests a notification for future price drops on a specific route.*
    - ***System Response****: The system tracks price changes and notifies the customer when the price drops below a certain threshold or on specific dates.*

Functional Requirements

- *The system must analyze historical pricing data and predict future price fluctuations based on trends and seasonality for specific routes.*

- *The system must allow customers to set alerts for price changes, notifying them when the price drops below a pre-defined threshold or on selected dates.*
- *The system must offer recommendations for alternative travel dates with lower prices based on the customer's selected route and preferences.*
- *The system must ensure that the price trends and predictions are updated frequently to reflect real-time changes in flight pricing.*

Other Non Functional Requirements

1. Performance Requirements

The system must handle appreciable amount of simultaneous users, with decent response times. Real-time flight tracking must refresh every 60 seconds. The system should scale efficiently to accommodate up to 50% more users during unexpected surges, such as during delays or emergencies.

2. Safety Requirements

The system must comply with aviation safety standards and regulations, ensuring that flagged passengers are accurately identified without fail during security checks. There must be fail-safes in place to prevent booking errors that could compromise passenger safety, such as double-booked flights or inaccurate seat assignments. Any system malfunction that impacts flight safety must trigger an alert to the relevant authorities and prompt corrective actions.

3. Security Requirements

The system must enforce multi-factor authentication for all users accessing sensitive data (e.g., passport information, flight manifests). All communication between the system and external databases must be encrypted using industry-standard protocols. The system must comply with passenger data privacy standards.

4. Software Quality Attributes

- ***Availability**: The system must maintain decent uptime escpecially during operational hours.*
- ***Reliability**: Must ensure reliable passenger data handling without crashes or data loss under heavy load.*
- ***Scalability**: Capable of scaling as the airport expands, without major infrastructure changes.*
- ***Usability**: Designed with a user-friendly interface, allowing airport staff and passengers to perform tasks with minimal training.*
- ***Maintainability**: The system should allow for easy updates and bug fixes with minimal disruption.*

5. Business Rules

- *Only authorized personnel can access security-flagged passenger information.*
- *Airlines have the authority to reroute passengers but must offer alternatives that ensure same-day arrival.*
- *Passengers can modify their check-in information up to 2 hours before departure, after which changes require approval by airline staff.*
- *Refunds and re-routing are subject to airline policies, with options presented automatically based on the system's analysis.*

Appendix A: Glossary

1. *AMS: Airport management system*
2. *TCP: Transmission control protocol*
3. *TLS: Transport Layer Security*
4. *API: Application programming interface*
5. *SSD: Solid State Drive*
6. *HTTPS: Hypertext Transfer Protocol Secure*
7. *SQL: Structured Query Language*

Appendix B: Field Layouts

Field	Data Type Length	Data Type	Is Mandatory
Passenger ID	15	Alpha-Numeric	Yes
Passport Number	10	Alpha-numeric	No (not required for domestic flights)
Flight Status	20	String	Yes
Criminal record	1	Boolean	Yes
Food preferences	10	String	Yes
ETA(Expected Time of Arrival)	10	Time	Yes
ETD(Expected Time of Departure)	10	Time	Yes
Destination	30	String	Yes
Date	10	Date	Yes
Delay	1	Boolean	No(Only when there is a delay)

Appendix C: Requirement Traceability Matrix

Req.ID	Req Name	Brief Description of Requirement	Test Case.ID	Status
1.	Login Page	User can login to access the features of the website	TC-01	Ongoing
2.	Schema Initialization	Creation of tables and then uploading the flight data	TC-02	Ongoing
3.	Re-Routing of flights	Must find alternative flight routes	TC-03	Pending
4.	Cost optimization	Must find cheapest flight for the trip	TC-04	Pending
5.	Security Flagging	Must identify flagged commuters and alert security	TC-05	Pending