Software Requirements

Specification

for

Airline Management System

Version 1.0

Prepared by Arnav Tripathi (230953278)

Aryan Nair (230953294)

Madhavan Komandur (230953298)

Manipal Institute Of Technology
4th April, 2025.

Table of Contents

1. Introduction

1.1. Purpose

The **Airline Management and Booking System** is designed to provide an efficient platform for airline staff and passengers to manage flight operations. The system enables passengers to search for flights, book tickets, and check flight statuses while allowing airline management to update flight details, manage employees, and ensure smooth airline operations.

1.2. Document Conventions

This document follows **IEEE 830-1998 standards** for Software Requirements Specification (SRS), ensuring clarity and completeness.

1.3. Intended Audience and Reading Suggestions

- **Airline Management Staff**: To update flight, employee, and airfare details.
- **Passengers**: To book, cancel, and check flight statuses.
- **Developers**: To implement the system based on specified requirements.
- **Testers**: To verify the system functionalities.

1.4. Product Scope

The system enables:

- **Flight booking and cancellation** for passengers.
- **Real-time flight status tracking.**
- **Employee management** for airline operations.

The system will be developed using:

- **Backend:** Flask (Python)
- **Frontend:** HTML, CSS, Jinja2
- **Database:** MySQL
- **Hosting:** Cloud-based infrastructure, mysql-connector-python

1.5. References

- IEEE 830-1998 Standard for Software Requirements Specification
- Airline industry regulatory guidelines
- UI/UX accessibility standards
- Documentation links:
    - HTML : https://www.w3schools.com/html/
    - BootStrap : https://getbootstrap.com/docs/
    - Flask : https://flask.palletsprojects.com/en/
    - Jinja2 : https://jinja.palletsprojects.com/en/
    - mysql-connector-python : https://dev.mysql.com/doc/connector-python/en/

2. Overall Description

2.1. Product Perspective

The **Airline Management and Booking System** is a **standalone web-based application** that integrates external APIs for real-time flight data.

2.2. Product Functions

- **User Authentication:** Secure login portals for passengers and airline staff.
- **Flight Management:** Add, update, and remove flights.
- **Booking Management:** Allow passengers to book, modify, and cancel flights.
- **Flight Status Tracking:** Provide real-time flight updates.
- **Employee Management:** Manage airline staff details.

2.3. User Classes and Characteristics

- **Passengers –** Users who book, modify, and cancel flights.
- **Airline Staff –** Employees who manage flights and airline data.
- **Administrators –** Users with full access to system functionalities.

2.4. Operating Environment

- **Database:** MySQL
- **Backend:** Flask (Python)
- **Frontend:** HTML, CSS (Bootstrap/Tailwind), JavaScript
- **Hosting:** Cloud-based

2.5. Design and Implementation Constraints

- The system **must be user-friendly and secure**.
- Only available seats must be bookable.
- The system must comply with **airline industry standards**.

2.6. User Documentation

- A user manual will be provided for airline staff and passengers.
- Online help and tooltips will be integrated within the system.
- FAQ section for common user queries.
- Useful Tutorials like : https://www.wikihow.com/Book-a-Flight

2.7. Assumptions and Dependencies

- Users will have internet access to use the system.
- The system will be compatible with modern web browsers.
- External API services for real-time flight data will be available.

3. External Interface Requirements

3.1. User Interfaces

- **Login Page:** Secure login for passengers and staff.
- **Dashboard:** Overview of flights, bookings, and employees.
- **Booking Interface:** User-friendly seat selection and booking system.
- **Flight Status Page:** Displays real-time updates.

3.2. Hardware Interfaces

- Server hosting for backend and database.
- Devices: PC, tablet, mobile (responsive design).

3.3. Software Interfaces

- **Database**: MySQL
- **Backend Framework**: Flask (Python)
- **Frontend**: HTML, CSS, JavaScript

3.4. Communications Interfaces

- HTTPS for secure communication.
- RESTful APIs for third-party integration.

4. System Features

4.1. Flight Management

4.1.1. **Add, update, and remove flight schedules.**

4.1.2. **Assign aircraft to flights.**

4.2. Passenger Booking

4.2.1. **Search flights by route, date, and availability.**

4.2.2. **Select seats and manage bookings.**

4.2.3. **View and cancel bookings.**

4.3. Employee Management

4.3.1. **Add and update employee details.**

4.3.2. **Assign roles and responsibilities.**

4.4. Flight Status Tracking

4.4.1. **Provide real-time flight updates.**

5. Other Nonfunctional Requirements

5.1. Performance Requirements

- System response time should be **less than 3 seconds**.
- Must handle **up to 10,000 concurrent users**.

5.2. Safety Requirements

- Must prevent **unauthorized access to data**.

5.3. Security Requirements

- **Role-based access control** for different user classes.

- **Secure authentication mechanisms** (e.g., OTP, encryption).

5.4. Software Quality Attributes

- **Usability:** Easy to navigate.
- **Reliability:** 99.9% uptime.
- **Scalability:** Supports airline expansion.

5.5. Business Rules

- **Only available seats** should be bookable.
- **Flight schedules** can only be modified by authorized staff.

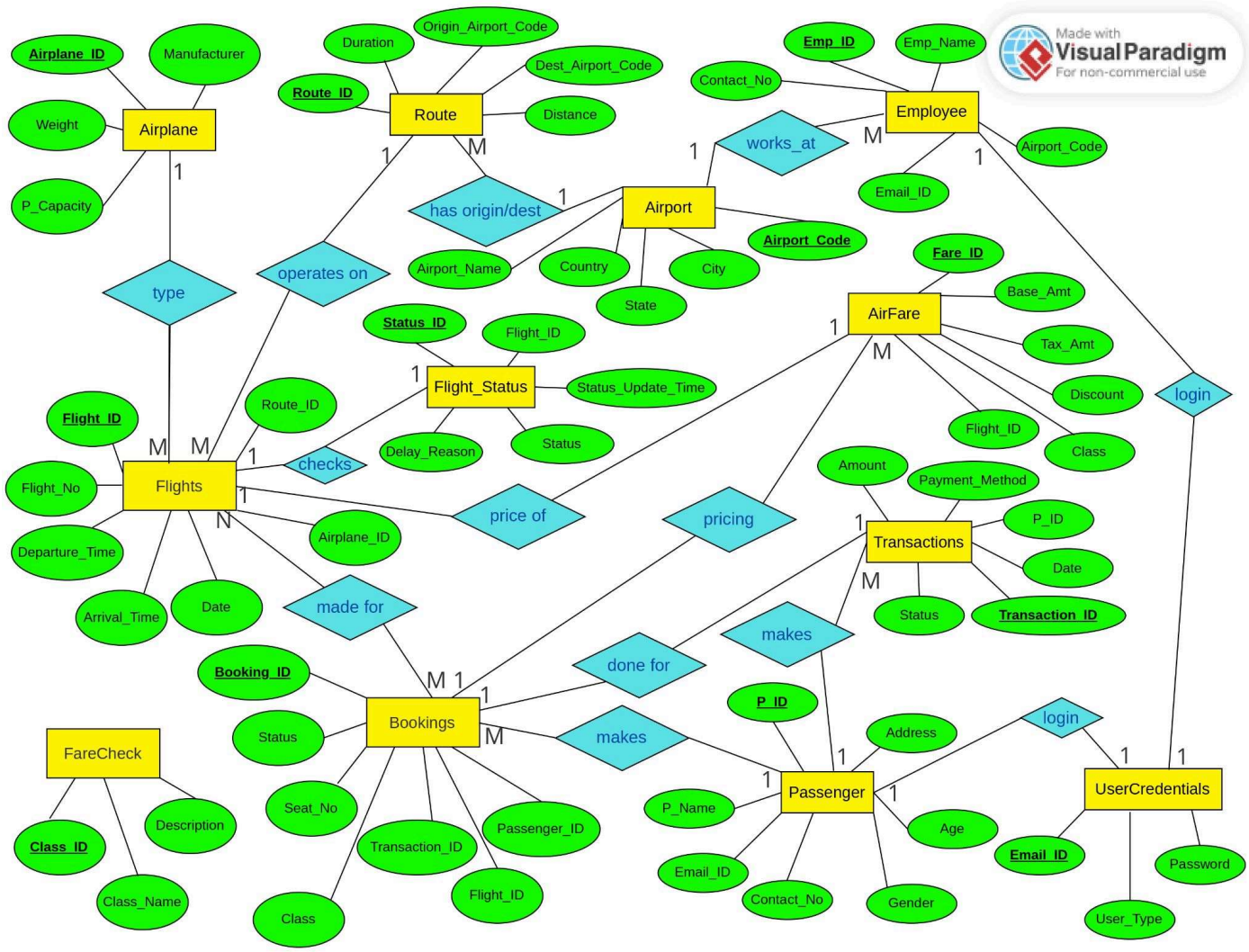
6. Other Requirements

- **Database Design:** Optimized for efficient data retrieval.
- **Internationalization:** Multi-language support for global use.

Appendix A: Glossary

- **Passenger:** A user who books and manages flights.
- **Administrator:** A high-privilege user managing all operations.

Appendix B: Entity Relationship Diagram



Appendix C: Class Diagram