**Software Requirements Specification**

Version 1.0

May 29, 2022

Aviation Data Display Application

Nilay Purayar (20112021)

Naveen N S (20112045)

Submitted in fulfilment

of the requirements of Software Engineering

**Table of Contents**

1.     Introduction

1.1  Purpose

1.2  Document Conventions

1.3  Intended Audience and Reading Suggestions

1.4  Project Scope

1.5  References

2.     Overall Description

2.1  Product Perspective

2.2  Product Features

2.3  User Classes and Characteristics

2.4  Operating Environment

2.5  Design and Implementation Constraints

2.6  Assumption and Dependencies

3.     System Features

3.1  Functional Requirements

4.     External Interface Requirements

4.1  User Interfaces

4.2  Hardware Interfaces

4.3  Software Interfaces

4.4  Communication Interfaces

5.     Non-Functional Requirements

5.1  Performance Requirements

5.2  Safety Requirements

5.3  Security Requirements

5.4  Software Quality Attributes

**1. INTRODUCTION**

**1.1 PURPOSE**

The objective of this work is to create a JavaFX-based desktop application that will offer data on aircraft operations by airport, route, and carrier. By delivering precise data, this application intends to reduce uncertainty amongst travellers and aviation management staff.

**1.2 DOCUMENT CONVENTIONS**

This document uses the following conventions:

* DB – Database
* ER – Entity Relationship
* GUI – Graphical User Interface

**1.3 INTENDED AUDIENCE AND READING SUGGESTIONS**

This is a prototype for a scheduled flight information display system that is only available on college grounds. This has been put in place with the help of college instructors. This application is primarily intended to aid in the effective operation of airports.

**1.4 PROJECT SCOPE**

By providing flight schedules, the objective of the flight information display system is to simplify flight management and provide a simple and easy-to-use application for travellers as well as airport management personnel for effective operations. With its schedule flight management and displaying a feature, the system is built on JavaFx and a relational database. We'll have a MySQL database that will store all of the planned flight information from multiple airports, routes, and carriers. Above all, we strive to give a pleasant customer experience at the most competitive price.

**1.5 REFERENCES**

* <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-databaseCSAC>
* <https://www.youtube.com/watch?v=hwCbXOM4_Qc>
* <https://www.javaguides.net/2019/07/javafx-jdbc-mysql-tutorial.html>
* <https://www.youtube.com/watch?v=whhSR0wlWQY>
* https://studylib.net/doc/25306078/flight-managment-system

**2. OVERALL DESCRIPTION**

**2.1 PRODUCT PERSPECTIVE**

The flight data database system stores the following information.

* **Scheduled flight data with respect to airports:**

It includes departure time, route, type of travel, aircraft model, and the airline of the scheduled flight.

* **Scheduled flight data with respect to routes:**

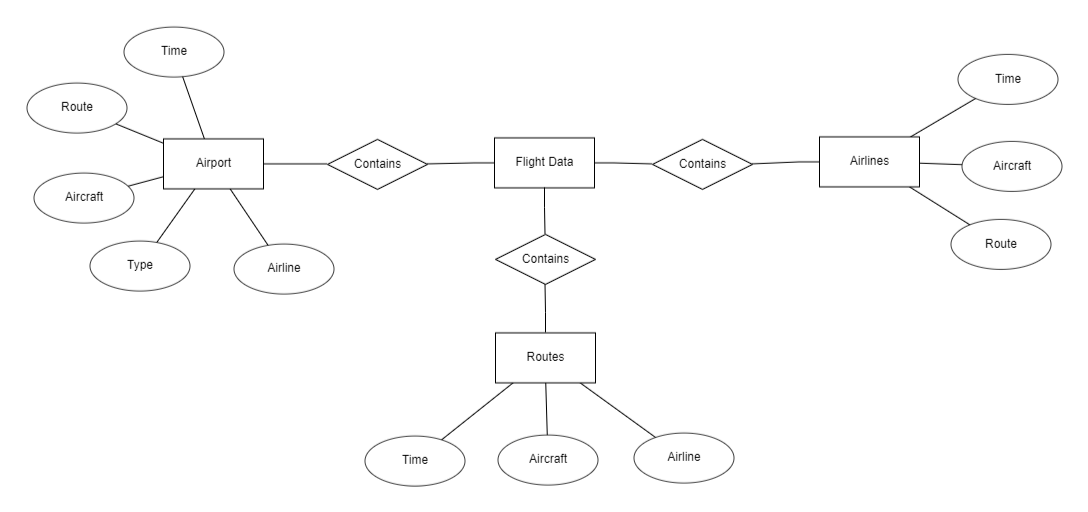
It includes departure time, aircraft model, and the airline of the scheduled flight.

* **Scheduled flight data with respect to airlines:**

It includes departure time, route, and aircraft model of the scheduled flight.

**2.2 PRODUCT FEATURES**

The major features of the airline database system as shown in the below **entity-relationship model** (**ER model**)



**ER Diagram of Flight Data DB**

**2.3 USER CLASS and CHARACTERISTICS**

Users of the system should be able to search the database for flight information based on airport name, airline or a specific route. A route from city A to city B is a set of connecting flights from A to B. Customer and Employee user privileges will be supported by the system. Employees will have access to both customer and flight management operations, while customers will have access to customer functions. The consumer must be able to do the following tasks:

* **Search for scheduled flights based on airport/IATA code:**

Upon pressing the button to search for scheduled flights based on airports, the user will get the privilege to choose the airport whose information he wishes to get. Based on this response, the application will fetch a specific table from the DB and display it on GUI.

* **Search for scheduled flights based on Route:**

Upon pressing the button to search for scheduled flights based on airports, the user will get the option to choose departure and arrival airports. Based on this response, the application will fetch a specific table from the DB and display all the flights scheduled for that particular route and display it on GUI.

* **Search for scheduled flights based on Airline:**

Upon pressing the button to search for scheduled flights based on airlines, the user will get the option to select from airlines whose flight data are available DB. Based on this response, the application will fetch a specific table from the DB and display all the flights scheduled for that particular route and display it on GUI.

**2.4 OPERATING ENVIRONMENT**

Operating environment for the airline management system is as listed below.  distributed database

* Client/server system
* Operating system: Windows, macOS, Linux
* Database: MySQL DB
* Platform: JavaFx

**2.5 DESIGN and IMPLEMENTATION CONSTRAINTS**

1. Our DB includes 3 tables which are independent to each other with *Time* being the primary key constraint.

2. SQL commands for queries:

* SELECT \* FROM *BLR*
* SELECT \* FROM *IXR*
* SELECT \* FROM *BOM*
* SELECT \* FROM *CCU*

* SELECT \* FROM *INDIGO*
* SELECT \* FROM *AIRASIA*
* SELECT \* FROM *GOFIRST*
* SELECT \* FROM *AIRINDIA*

* SELECT \* FROM *BLR\_PNQ*
* SELECT \* FROM *IXR\_BLR*
* SELECT \* FROM *BLR\_DXB*
* SELECT \* FROM *BOM\_DEL*

3. Response for the queries will be generated as per the button click from user

4. Our DB is currently working in the localhost of the admin’s system. We are trying to push the DB to the cloud-based DB management system.

**2.6 ASSUMPTION DEPENDENCIES**

         We have assumed that our application will be used by passengers and airport management staff as per their requirements and necessities. Thus, it will be used for the following applications.

* Which is the airline mostly tracked by travellers
* Which airport is the busiest based on the number of scheduled flights
* Which route has the most/least number of scheduled flights.

**3. SYSTEM FEATURES**

**3.1 FUNCTIONAL REQUIREMENTS**

§  **DESCRIPTION and PRIORITY**

The aviation data management application maintains information on flights, airports and routes. This project has a high priority because it is very difficult to track information regarding scheduled flights due to regular change in schedules and operational complexity.

* **STIMULUS/RESPONSE SEQUENCES**
  + Search for Airline Flights for two Travel cities
  + Search for Airline Flight of specific airlines
  + Search for Airline Flight from specific airport.
  + Displays a detailed list of available flights.

* **FUNCTIONAL REQUIREMENTS**
  + Centralised DBMS
  + Client/Server System

**4. EXTERNAL INTERFACE REQUIREMENTS**

**4.1 USER INTERFACES**

* Front-end software: Fxml
* Back-end software: MySQL, Java

**4.2 HARDWARE INTERFACES**

* Windows/ macOS/ Linux

**4.3 SOFTWARE INTERFACES**

|  |  |
| --- | --- |
| **Software Used** | **Description** |
| Operating System | Windows is preferable |
| Database | MySQL DB |
| JavaFx | Platform used to build the software |

**4.4 COMMUNICATION INTERFACES**

         This project supports every personal computer or equivalent. We are also in the process to deploy the same on the web.

**5.**    **NON-FUNCTIONAL REQUIREMENTS**

**5.1 PERFORMANCE REQUIREMENTS**

The steps involved to perform the implementation of the airline database are as listed below.

**A) E-R DIAGRAM**

The E-R Diagram is a method for visually expressing the logical structure of a database. After that, this analysis is utilized to arrange data as a relation, normalize the relation, and create a relational database.

* ENTITIES: These are the different real-world things in an application that are specified.

* PROPERTIES/ATTRIBUTES: These define an entity's properties and relationships.

* RELATIONSHIPS: These are the connections between items that indicate meaningful interdependence.

**B) NORMALIZATION:**

Normalization's main goal is to eliminate redundancy, which implies that data should only be saved once. Several times storing information wastes storage space and increases the total size of data stored.

Modification anomalies can occur if a database is not correctly constructed. When data is added to, altered, or removed from a database table, modification anomalies occur. Data redundancy may also be a problem with conventional databases and poorly constructed relational databases. A database can be normalized to remove these issues.

The act of dissolving a table into smaller tables is known as normalization. As a result, each table focuses on a specific subject. There are three types of anomaly modifications, and the first, second, and third normal forms (3NF) are deemed sufficient for the majority of practical needs. It should only be considered after a comprehensive examination and comprehension of its ramifications.

**5.2 REQUIREMENTS FOR SAFETY**

  In case of a catastrophic failure, such as a disk crash, causes extensive damage to a large portion of the database since we are currently having a centralised DB, hosted in our localhost, our entire data will be lost. Thus, we need to shift our DB to distributed cloud DB hosting sites as soon as possible. On the other hand, we need to have a copy of our DB stored elsewhere as a backup.

**5.3 REQUIREMENTS FOR SECURITY**

Security systems, like many other applications, require database storage. However, due to the unique needs of the security sector, suppliers must carefully select their database partners.

**5.4 ATTRIBUTES OF SOFTWARE QUALITY**

* AVAILABILITY:  The flight should be obtainable based on the choice of the user.

* ACCURATENESS: The flight should take off from the proper terminal and

 arrive at the correct destination, airport-wise and airline-wise data must also be accurate.

* MAINTENANCE: Flight schedules should be kept up to date by administrators

and flight attendants.

* USABILITY: The flight schedules should be able to meet the demands of as

many users as possible.