
AIRLINE RESERVATION SYSTEM PROJECT

SOFTWARE REQUIREMENTS SPECIFICATION

Group 24

Prepared by

210057R	BALASOORIYA K.B.R.D.
210161F	FERNANDO K.J.R.K.
210413G	NAWAGAMUWA N.P.V.N.
210483T	PRABASHWARA D.G.H.
210676M	VITHURSHAN P.

Department of Computer Science and Engineering
University of Moratuwa

September 8, 2023

Submitted in partial fulfilment of the requirements of *CS3043 - Database Systems*

Contents

1. Introduction	3
1.1 Purpose	3
1.2 Scope	3
1.3 Glossary	4
1.4 References	4
1.5 Overview of Document	4
1.5.1 Overall Description	4
1.5.2 External Interfaces	5
1.5.3 System Features	5
1.5.4 Other Non-Functional Requirements	5
1.5.5 Analysis Models	5
2. Overall Description	6
2.1 Product Perspective	6
2.2 Product Functions	6
2.3 User Classes & Characteristics	6
2.4 Operating Environment	7
2.5 Design & Implementation Constraints	7
2.6 Assumptions & Dependencies	8
3. External Interface Requirements	9
3.1 User Interfaces	9
3.2 Hardware Interfaces	9
3.3 Software Interfaces	9
3.4 Communication Interfaces	9
4. System Features	10
4.1. Display flight schedules	10
4.2. Allow booking of flights	11
4.3. Login	11
4.4. Allow admin accounts to perform Update and Delete operations on certain database information.	12
4.5. Generate and display reports	12
5. Other Non-Functional Requirements	14
5.1 Performance	14
5.2 Security	14
5.3 Business rules	14
5.4 Accuracy	15
5.5 Scalability	15
5.6 Maintainability	14
5.7 Portability	15
5.8 User-friendliness	15
6. Analysis Models	16
6.1. ER Diagram	16

6.2. <i>User Hierarchy Diagram</i>	17
6.3 <i>State Diagram</i>	18

1. Introduction

This section serves as a gateway into the comprehensive exploration of the airline reservation system project which is undertaken as a part of module *CS3043 – Database Systems*. This section will provide basic details of the project as well as will provide details to navigate through this document.

1.1 Purpose

This document is prepared with the purpose of providing a detailed description of the functional airline reservation system that we are required to implement as part of module *CS3043 – Database Systems*. In this document, we will discuss the features of the system, the interfaces we plan to implement, its core functionality, and the operating environment it is expected to function in. This document is written with the intended audience being the project team members, project evaluators, as well as all other stakeholders who might become involved in this project.

1.2 Scope

The software system we are required to implement is an airline reservation system where travellers should be able to view flight schedules and book flights to destinations of their choice. The system is intended to be used by a small-scale airline in possession of a small number of aircrafts, which caters only to a few destinations and does not offer transfer flights.

This system is designed to allow any user on the platform to view the flight schedule. When a user wants to make a booking, the user can either sign into the account or proceed as a guest. Even though the system does not facilitate a payment mechanism it allocates a ticket for a booking after the payment is done. The system also generates necessary reports only for the administrators. The system also contains a relational database containing lists of registered users, guests, flights, and bookings.

1.3 Glossary

Term	Description
<i>B Airways</i>	The name of the company by which the system will be used
<i>Virgin Airlines</i>	The parent company of <i>B Airways</i>
User	Anyone who accesses the system
Traveller	Someone who accesses the system and has personal information in the system. Could be a registered user or guest who enters the information
Administrator	Special user accounts with elevated permissions. Can view generate/view reports
Frequent Flyer Membership	A registered user who has booked over “X” number of flights with the system. (Get a 5% discount)
Gold Flyer Membership	Registered user who has booked over “Y” number of flights with the system. (Get a 9% discount)
Economy, Business, Platinum	Standard names for flight-passenger classes.
BIA	Bandaranaike International Airport (Sri Lanka)
BKK	Suvarnabhumi International Airport (Bangkok)
BOM	Mumbai International Airport (India)
CGK	Soekarno–Hatta International Airport (Indonesian)
DEL	Indira Gandhi International Airport (India)
DMK	Don Mueang International Airport (Thailand)
DPS	Denpasar International Airport (Bangkok)
HRI	Mattala International Airport (Sri Lanka)
MAA	Madras International Meenambakkam Airport (India)
SIN	Changi International Airport (Singapore)

1.4 References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.

1.5 Overview of Document

To guide the reader through the next sections of this document a brief overview of what each section will contain is presented here.

1.5.1 Overall Description

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It provides details about the product perspectives and product functions. user class, the product’s operating environment, its limitations, and its dependencies.

1.5.2 External Interfaces

In the third chapter, External Interfaces section this document provides a detailed description of external interfaces that this system is involved with. It includes the user interface, software interface, hardware interface, and communication interface.

1.5.3 System Features

The system feature section which is the fourth chapter provides detailed information about functionalities of the functional requirements.

1.5.4 Other Non-Functional Requirements

In the fifth chapter this document describes the non-functional requirements and their importance in this project.

1.5.5 Analysis Models

In the last chapter, the Analysis Models section of the document provides models that are helpful in analysing this system. The Entity Relation Diagram and User Hierarchy diagram are included in this section.

2. Overall Description

This section provides a comprehensive overview of the airline reservation system. Subsequent subsections provide an idea about how the system is behaving and functioning, its constraints and dependencies and the project's operating environment.

2.1 Product Perspective

This system is designed as an independent entity capable of managing the seat booking details of a small-scale airline. It does not handle details such as visas of passengers, payments, keeping track of in-flight services, flight-crew details, aircraft maintenance, etc. The system is designed in lieu of decisions taken by the management of *B Airways* to expand its operations and cater to more destinations than simply internal Indonesian flights.

While the current version of the system does not support payment handling, it is important to note that it may need to integrate with a payment manager in future to facilitate payment processing. However, in the current version, the payment status can be updated according to an external payment manager.

2.2 Product Functions

- Viewing flight schedules
- Booking a flight with or without logging in to the platform
- Logging into a user account using a username and a password
- Viewing and Updating flight schedules, aircraft details, routes, and airports
- Generating and Viewing reports

2.3 User Classes & Characteristics

Any entity that accesses the system will be henceforth referred to in this document as a "user". This designation will change if they choose to sign in and verify their identity as a traveller or administrator, or if they decide to book a flight as a guest and enter the required information. Once a guest user enters their information they too will be considered as a traveller.

The system will be used by two major groups:

- *Travellers* – General passengers wishing to fly with *B Airways*.
- *Administrators* – System administrators who are duly authorised employees of *B Airways* or its parent company, *Virgin Airlines*.

"Travellers" on the system will themselves comprise of two groups:

- *Registered users* – Users who have entered a valid username and password and have entered correct personal information.
- *Guest users* – Users who wish to make a one-time booking and do not wish to maintain an account. They will still be required to enter certain information prior to making a booking.

Travellers will be able to view flight schedules and book flights as they require. Registered users will be tracked, and membership status will be maintained according to the number of times they have booked with the system. They will be eligible for certain concessions depending on their membership status.

There is currently two membership status:

- *Frequent membership* – Registered users who have made bookings exceeding a predefined threshold, that is specified by the company management, will be eligible for the Frequent membership. Frequent members are entitled to have a discount, which can be configured by the management, on their total cost for the booking.
- *Gold membership* – Gold membership is an elevated tier, reserved for users who have booked well above frequent users. Gold members will also enjoy a discount on the total price which is typically higher than the frequent membership .

A diagram explaining the user class and the change of state between users is included in the [Analysis Model chapter](#).

2.4 [Operating Environment](#)

The system should be a web-application and should be accessible to any user who wishes to book a flight with *B Airlines*. The backend of the application is made using NodeJS while the frontend is made using ReactJS. The database is a relational SQL database. For demonstration purposes, the database will be stored on team members' personal computers, and the personal computers will also function as servers for the web application.

2.5 [Design & Implementation Constraints](#)

The current system version does not have any payment feature. Therefore to make the system properly function there is a need for an external payments management system. It is possible to update the payment status according to the information from the external payment manager.

Since the project is hosted on the team members' personal computers there are several constraints. Since the web application is demonstrated in localhost, users can only visit the web application through the personal computer that is hosting the web application. Performance wise the application might not be able to handle large amounts of requests due to limited processing power. Also storing large amounts of data in databases is not possible since personal computers only have limited storage. Furthermore, scaling up the application is not possible due to limited resources.

The current version of the system does not offer transit/transfer flights since the business has only a small range of destinations. If the business is about to scale up then the system can be changed to handle the transit/transfer flights.

Since the system has to deal with passwords and authentications as well there is a constraint with regards to the communication protocol the system uses. HTTPS protocol should be used for the secure connection although the protocol system has is HTTP.

2.6 Assumptions & Dependencies

This project will depend on the dependencies of the software runtimes, libraries, and software frameworks that are used to design this project. Once deployed, the performance of this system will also depend on the user having a reliable connection. And it is assumed that users have a basic understanding of how to navigate through a web application.

3. External Interface Requirements

In this chapter we specify the interface requirements that the system will have, under the titles presented below.

3.1 User Interfaces

- Users will primarily access the system through a web based interface provided by the user application. This web Application serves as the primary software interface for all users.
- Web application is compatible with any web-browser such as Mozilla, Chrome, Edge etc.

3.2 Hardware Interfaces

- A network-accessible device capable of running a modern web-browser.
- A sufficiently updated version of any modern web-browser installed.

3.3 Software Interfaces

- Database implementation was done with SQL, with backend NodeJS.
- GUI components were implemented using React.

3.4 Communication Interfaces

- Ideally, the system should use HTTPS protocol since it requires transferring username, password information as well as user-specific personal information. However, for demonstration purposes HTTP protocol was considered adequate.

4. System Features

In this section we intend to provide a clear and comprehensive overview of the major functions that the system is expected to perform. The high-level objectives are provided as headings with more specific expectations, restrictions, and rules being discussed and noted under each. In a nutshell, the system is expected to behave as an inventory management system, where the inventory is the seats on the particular flights.

4.1. Display flight schedules

For each day and for each flight, the Airline will create a “flight schedule”. This will include the aircraft that is assigned to make the journey as well as its departure and expected arrival times. The flight schedule will be central to the system, as all users who first access the system (regardless of their verification status) should be able to view the flight schedule.

The following information must be prominently displayed when viewing the flight schedule, or at least be easily and intuitively accessible:

- **Origin airport** – The airport from which the flight will depart (i.e. take-off)
- **Destination airport** – The airport at which the flight is expected to land. Since *B Airlines* only offers direct flights at the moment, a flight will only be associated with a single pair of origin and destination airports.
- **Departure time** – The time at which the flight is expected to depart. Additionally, the system should specify which time-zone it is using.
- **Expected arrival time** – The time at which the aircraft is expected to arrive at its destination. Flight delays should be updated on the database, which will then be reflected on the schedule.
- **Route number** – The unique number assigned to the route between the origin and destination airport
- **Number of free seats** – The number of free seats belonging to each class should be displayed per flight.

4.2. Allow booking of flights

Depending on the information they obtain from the flight schedules we expect our users to make the choice of whether there is a suitable flight offered for them by *B Airlines*. If there is, and they wish to reserve a seat on that flight, the system should allow booking of flights. However, this action is restricted as only users who have submitted certain essential information will be able to do so.

In submitting information users should be presented with a choice:

- Continue as a guest – They will enter their information with the intent of making a one-time only booking. Any subsequent time that they access the system, they will be unable to reuse the information they have already provided, and will have to instead refill the required forms.
- Register – Users should also be able to register with the system or use a login they have already configured. In this case, they will not have to refill any already-provided information unless they want to, and will be able to proceed straight to making the booking.

In the process of booking a flight the traveller will be presented with a choice of passenger class. This will include a choice between:

- Economy
- Business
- Platinum

Different flights and aircraft may offer different services/amenities according to the above passenger classes, and these will not be included/tracked by the system. Moreover, travel rates and tariffs charged will vary according to passenger class in addition to factors such as type of aircraft and route and will be decided upon at the discretion of the Airline. Therefore the system is not required to know or contain logic of how rates are calculated, but should store the value as specified by the Airline.

Once a traveller decides to reserve a seat, the system will create a “booking” on their behalf. Each seat in each flight can only have a single booking, as *B Airlines* has a policy of not overbooking their flights. A booking will be considered “complete” once the system receives verification of a payment being made. It is at this stage that a ticket will be issued to the traveller. The system itself will not handle payments, which will likely be done by an external entity.

4.3. [Login](#)

As stated above, users will be offered a choice of either registering or continuing as a guest. If a user chooses on the former, the system should prompt the user for a username and a password.

If they are signing up, the username and password will be stored on the database and the new user account will be assigned a unique user ID. Afterwards, the user will be able to provide their personal information which will be stored against the user’s account. This will enable the user to make bookings in the future without going through the hassle of providing the same information yet again.

If the user is attempting to login, the provided username and password will be authenticated

against those stored on the database. Once verified, they can then proceed to making the booking, opting to edit their provided information only if required. Logged-in users will additionally be eligible for certain concessions, depending on the number of times they have booked flights with *B Airlines*. In this regard, the system will maintain two membership types:

- Frequent
- Gold

Users with the “Frequent” membership type will be eligible for a 5% discount, whereas users under the “Gold” membership type will be eligible for a 9% discount.

In the event that a registered user requires to make a booking on behalf of another (for instance, on behalf of their underage child), they will be able to make the booking using their own account, making the discount valid for that booking as well. They will however have to add the other individual’s information to the system where this user information will be stored as a “guest” who is affiliated with a registered user.

Furthermore, users should have the ability to delete their user account. In this case the account will be marked for deletion and should be deleted from the database while adhering to the parent company’s policies as well as any policies in place regarding passenger-data retention.

The system will define a single login-point for both travellers as well as admin users. Once an admin user logs in to the system and is verified, they should have access to an additional range of operations than a typical traveller account. These operations should include:

- Updating flight information in case of a delay
- Generating and viewing reports

4.4. Allow admin accounts to perform Update and Delete operations on certain database information.

The system should enable authorised and verified admin accounts to perform certain update and delete operations on the information stored within the database. The system need not support all CRUD operations on all database information via interfaces, however, actions such as updating expected flight times to account for delays should be possible via the system itself.

4.5. Generate and display reports

The system should be capable of generating and displaying reports to verified admin accounts. These actions should not be accessible to unverified users or conventional travellers’ accounts. The reports which the system is required to generate include:

- Display all passengers in a given flight (specified by flight no.) and whether they are above/under the age of eighteen.

- The number of passengers travelling to a certain destination within a given date-range.
- The number of bookings by each passenger type within a given date-range.
- Given origin and destination, all past flights, states, passenger counts data
- Total revenue generated by aircraft type

5. Other Non-Functional Requirements

The airline reservation system incorporates several non-functional requirements that are integral to its performance, security, business rules, accuracy, scalability, maintainability, portability and user-friendliness. These requirements are outlined below.

5.1 Performance

- The performance of the system is important to ensure a seamless and responsive user experience. The system must demonstrate robust performance capabilities including the ability to handle the simultaneous users without degradation in system responsiveness or throughput. The chosen SQL database system will provide the above capability. The system is working with a local host, the response and data transfer delays also will be minimal.

5.2 Security

- The security of user data is a top priority. The system will employ advanced encryption techniques to securely store user passwords and safeguard personal information.
- All data transmission, particularly the transfer of passwords and sensitive user information, should be conducted over secure HTTPS connections. This will ensure safeguarding data integrity and confidentiality during transit.

5.3 Business rules

- The system will establish distinct user roles and assign well-defined privileges to each role. For instance, passengers and administrators will have separate roles with specific functions. Users who have not provided complete personal information will be restricted from booking flights to maintain data accuracy and business logic
- Travellers will not be granted the ability to modify flight details or schedules. This limitation is in place to ensure consistency in flight operations and prevent disruptions. The ability is only offered to the administrators
- Access to report generation and viewing will be reserved exclusively for admin accounts. This access control measure guarantees data confidentiality and prevents unauthorised access to sensitive information.

5.4 Accuracy

- The system will be meticulously designed to prevent the overbooking of aircraft. It will maintain a real-time synchronisation of booked seats and available seating capacity to eliminate any inconsistencies.
- To uphold fairness and data accuracy, the system will strictly enforce the rule that only one individual can be booked per seat, eliminating the possibility of multiple bookings for the same seat

5.5 Scalability

- The architectural design of the system will proactively consider scalability to accommodate future growth and expansion. This strategic planning aligns with B Airways' vision to broaden its destinations and services, ensuring that the system can seamlessly adapt to increased demands.

5.6 Maintainability

- The system's choice of a widely adopted and trusted technology stack, including nodeJS, reactJS, and SQL, is deliberate. These technologies facilitate easy system maintenance, updates, and troubleshooting. Their broad community support and documentation ensure long-term system maintainability and reliability

5.7 Portability

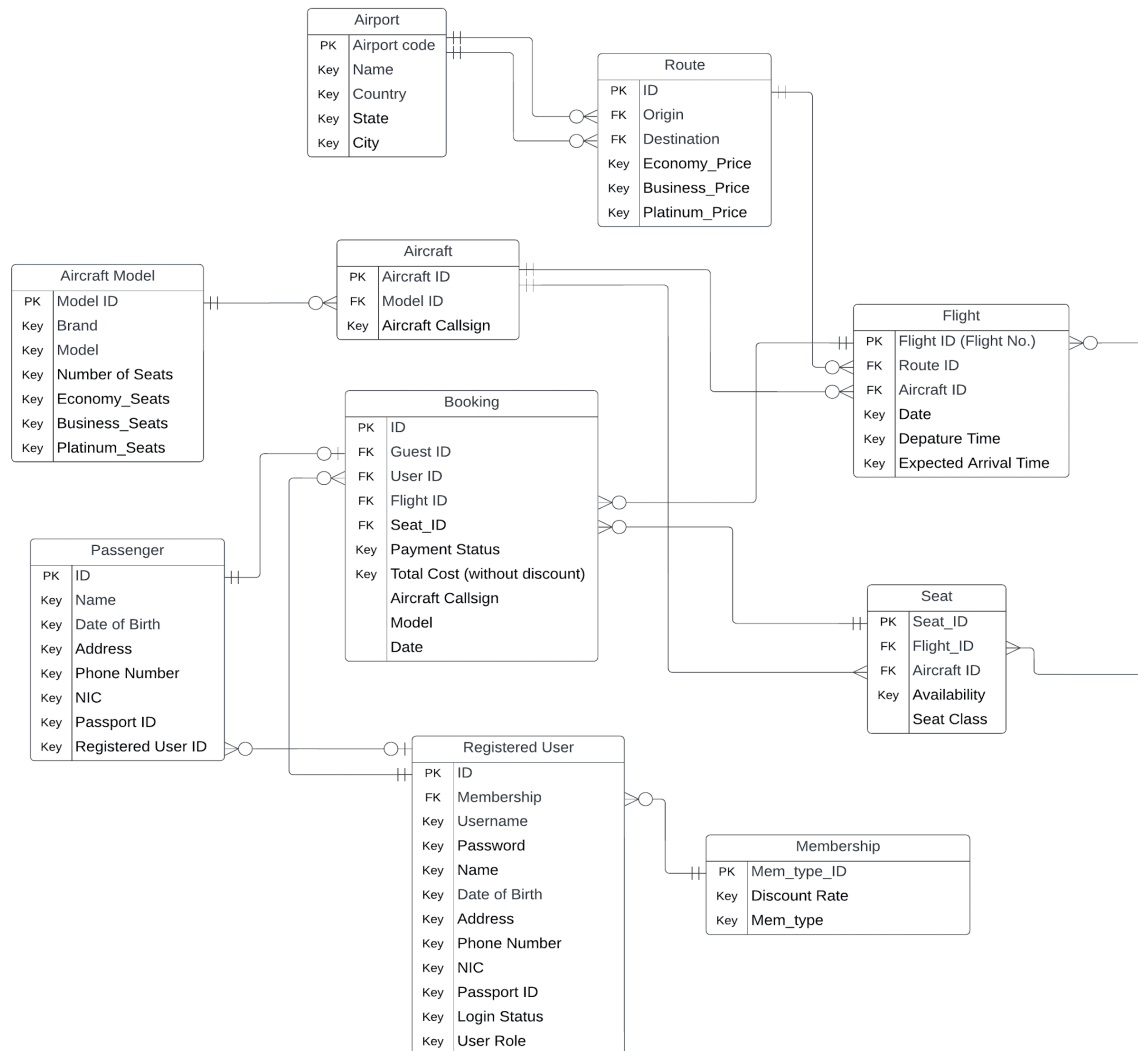
- In anticipation of potential integration with Virgin Airlines' systems, the system will be architected to enable data transfer to its parent company's databases and systems. The specifics of this data transfer will be determined based on the compatibility of systems used by Virgin Airlines.

5.8 User-friendliness

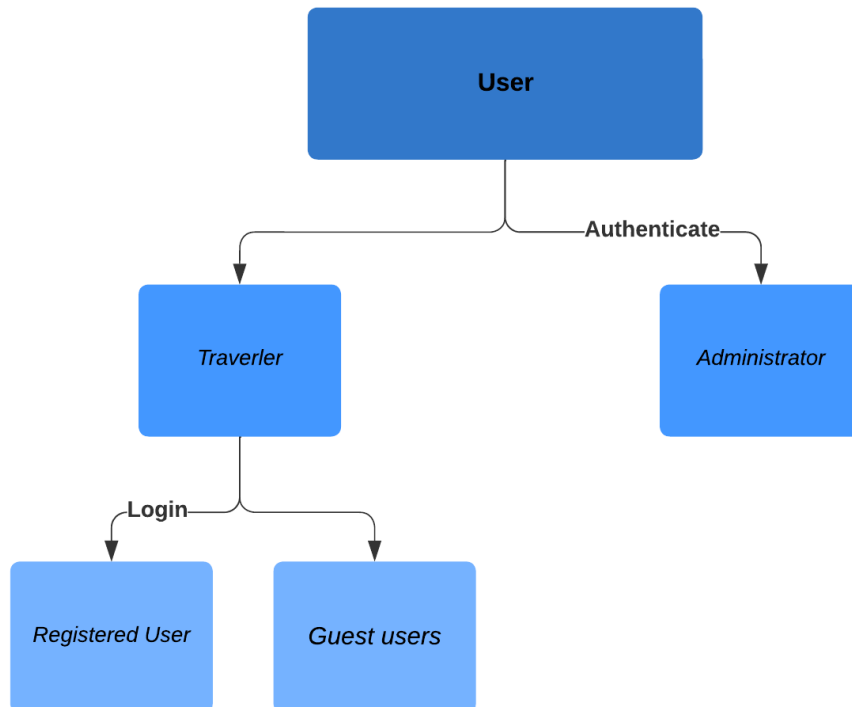
- User experience is a critical aspect of the system's design. The user interface will prioritise user-friendliness and intuitiveness, ensuring that passengers and administrators can interact with the system efficiently and with minimal learning curve. The design will focus on clear navigation, straightforward processes, and user-friendly features to enhance user satisfaction.

6. Analysis Models

6.1. ER Diagram



6.2. User Hierarchy Diagram



6.3 User State Diagram

