1. Introduction

1.1. Purpose

This document lists the software requirements specification for **Airplane Ticket Booking System (ATBR).** This document is subject to change as the project progresses.

1.2. Document Conventions

This document follows the IEEE standard for Software Requirements Specification.

1.3. Intended Audience and Reading Suggestions

The intended audience for this document includes software engineers, testers, project managers and documentation writers.

1.4. Project Scope

The software should be able to load user interfaces, search one way and round-trip types, search for multiple reservations, flight reservation, reservation cancellation, online payment, and dynamic flight pricing should be supported.
The user should be able to search flights, add price, destination, passengers, date filters on the flights.
The project is developed as a web-based application which runs its own flights data database and will communicate with airlines for reservation after the reservation price has been made.

1.5. References

1.5.1. Lecture Slides

1.5.2. [How to Write a Software Requirements Specification (SRS Document) | Perforce]

2. Overall Description

2.1. Product Perspective

ATBR is a fully digitalized/automated ticket booking system unlike the traditional ones which relies on airline offices/agents to distribute the tickets.
The problem with manual tickets is that it takes a lot of time, and the agents can hike up the tickets without the customer knowing, which makes the airplane tickets higher and less affordable for people. This software aims to provide a fast and efficient way to search and book flight tickets, while maintaining a competitive price with our competitors.
The system shall allow users to book tickets only if the user has a valid passport and the system allows different airline administrators to set coupons for their flight tickets.

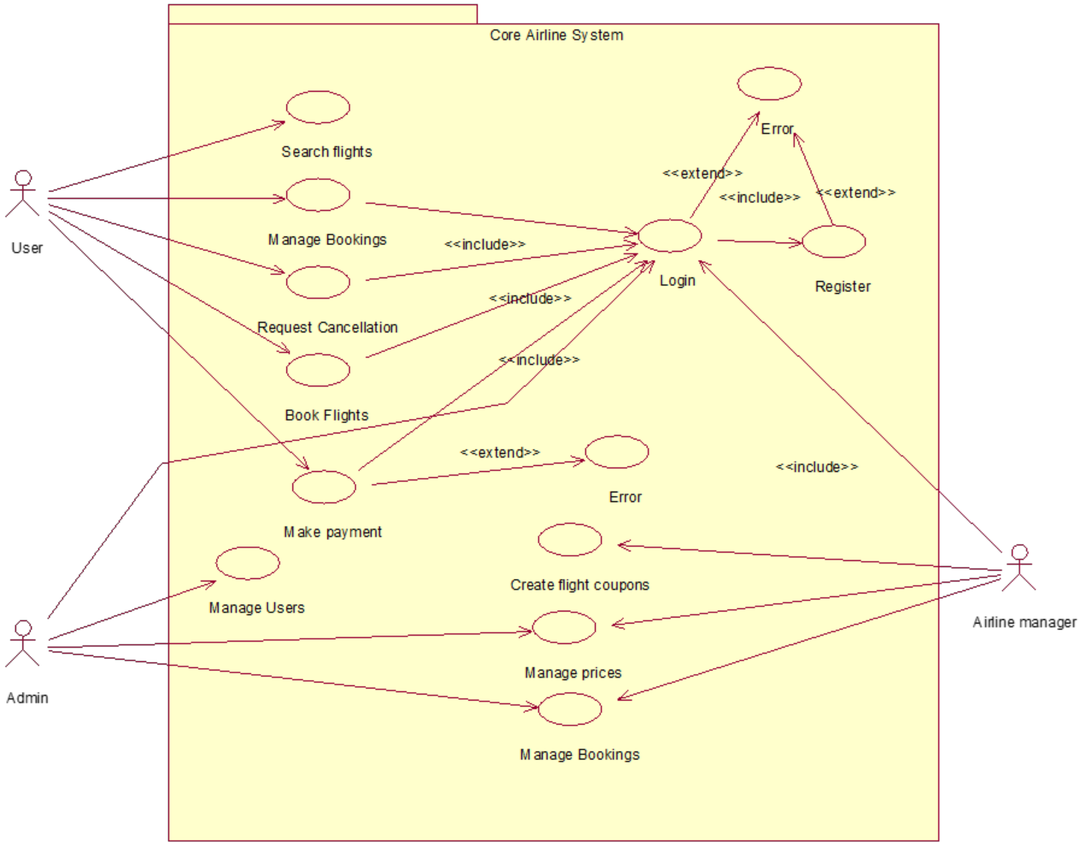
2.2. Product Features

The major product features of this software are

1. View and retrieve flight data from different airlines.
2. Book single, round trip tickets.
3. Search tickets for multiple with multiple passengers.
4. Search users to search for tickets with filters like destination and source, number of hops, transit times, price etc.
5. Provide the user an average of the flight between two stations.
6. User redirection to a third-party provider for secure online payments.
7. Recommend hotels near the stations.
8. Location based nearest airport finder.
9. View cancellation request.
10. Airline administrators can set the minimum ticket price.
11. Airline or Site administrator can set discount coupons and offers.

2.3. User Classes and Characteristics

In the use case diagram given below we have shown how different users interact with our system. Users can use the system to search flights, manage their bookings, request cancellation and book flights and make payments. All user processes except searching for flights require authentication and similarly for admin and airline manager they can manage flight prices and manage bookings. All admin and airline managers must be authenticated before they can perform any action according to the use case given below.



2.3.1 User requirements

1. User properties like Name, Address, Age,
2. Associated with Flight Miles accumulated and Credit Card information.
3. Flight properties like Departing/Arriving City, Departure/Arrival dates and times, Miles, and an identifying Flight Number.
4. Flight Seat properties of identifying seat number, reserved and flight
5. Associated to Flight-by-flight number
6. The users of the system should be able to read and be comfortable with the English Language.
7. Users should be comfortable using general purpose applications on the computer system.

2.4. Operating Environment

The software will be a web-based application, so the operating system will not be an issue. Any node that is connected to the Internet and has a web browser installed in it can access the software free of cost.

2.5. Design and Implementation Constraints

Some design and implementation constraints are listed below:

- **Data Synchronization Issues:** Airlines shared flight data might be uncoordinated if the user takes a lot of time to book.
- .
- Flight date should be displayed according to departure station.
- Flights should be shown according to daylight savings.
- Any information changes should be displayed immediately.

2.6. User Documentation

The flow of the website will be simple and normally the next step would be indicated by large buttons. Also, instructions on how to book flights will be provided on the website.

2.7. Assumptions and Dependencies

The assumptions we have made in this project are:

- Customers can perform online payments.
- The data provided by airlines are up to date.
- Customer does not have malicious intent and is not trying to disrupt our service in any way whatsoever.
- Customer has a reliable internet connection.
- Online transactions will not be rolled back later.

Dependencies in this project are:

- Heavily dependent on airline companies with their flight data.
- Dependent on the political/geographic nature of destinations. (i.e., flights may be canceled due to political strife or any other reasons, airports might be shut down).
- Speed and reliability of the internet connection.

3. System Features

This section provides detailed requirements for the website design, including functional requirements

3.1. General Requirements

3.1.1 Login

Description and Priority:

This function allows a registered user to login his account using his frequent flier number with the airline and password. If a user is not registered, the website should allow the user to enroll first. The System will check both the frequent flight number and password, when a user attempts to login. In most cases, the frequent flight number is convenient for both the user and system performance.
The user easily memorized his or her flight numbers but not a dull string. For the system, when provided the flight number, flight information will be delivered at the same time. Therefore, such operation reduces the second query chance. Theoretically, more than one record can be retrieved by the user's frequent flight number and password. Two or more users may have chosen the same password and same flight number. The

Inputs: Frequent flier number and password

Source: All inputs are provided by the user.

Outputs: Indication that the user is logged in to the system.

Destination: The outputs are displayed on the screen as well as stored in the system.

Requires: The user provides login information including frequent flier number and password.

Post-Conditions: User is logged in to the system, OR user is not logged in because he/she entered unrecognized information.

Side-Effects: None

3.1.2 Book Flights

Description and Priority:

The user can use the Book Flights function to purchase seats for an airplane flight. The system shall present the user with information on all current flights. The user may then select a pair (departure and return) of flights on which to purchase seats. The user can indicate the number of seats and placement of such. Finally, the system shall guide the user completely through the checkout process.

Inputs: User information – the user must already be logged in.

Source: Inputs are from the user except flight information, which is retrieved from the system.

Output: The purchased seats are tied to the user’s account, so he/she can reserve seats later.

Destination: The booked flights will be stored in the user's account information when the user finishes payment. The flight information shall also display on the screen.

Pre-Conditions: The user must have an account with the website and must be logged in.

Post-Conditions: Completion of this function guarantees that the user has seats on a specific flight.

3.1.3 Flight Schedule

Description & Priority:

This section of the system shall allow a user to query flight schedules based upon simple input criteria. The user will provide departure and arrival cities, and a departure/return date. If any flights match the criteria, the system will display the following information:

Flight Number
Departing City & Date/
Time Arriving City & Date/
Time

Inputs: Departing City, Destination City, Departure Date/Time

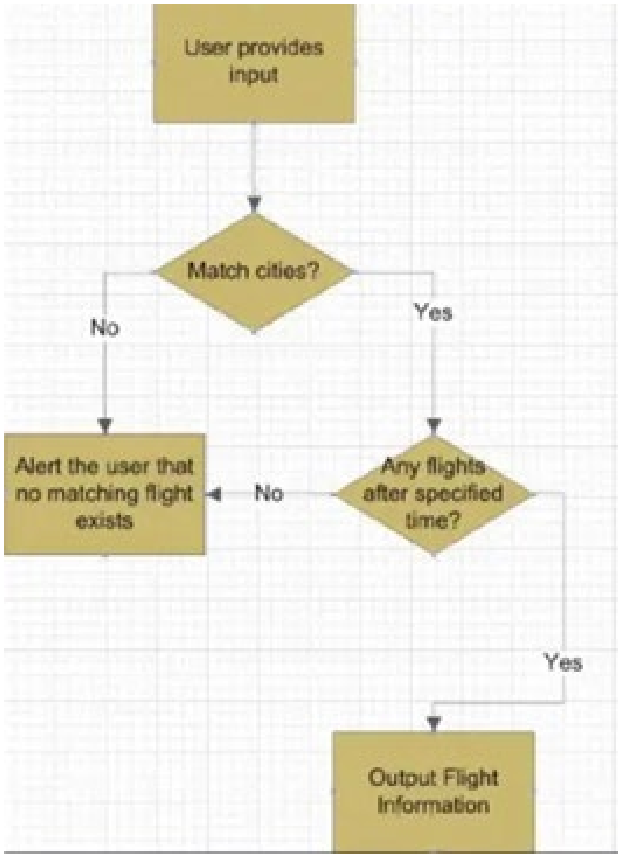
Source: All inputs provided by the user.

Outputs: Flight Information including Flight Number, Departing City & Date/Time, Arriving City and Date/Time, Number of Available Seats.

Destination: All output should display to the screen.

Pre-Conditions: None

Post-Conditions: User has flight information for any matching flight.


User provides input
Match cities?
No
Yes
Alert the user that no matching flight exists
No
Any flights after specified time?
Yes
Output Flight Information

4. External Interface Requirements

4.1. User Interfaces

1. User properties like Name, Address, Age,
2. Associated with Flight Miles accumulated and Credit Card information.
3. Flight properties like Departing/Arriving City, Departure/Arrival dates and times, Miles, and an identifying Flight Number.
4. Flight Seat properties of identifying seat number, reserved and flight
5. Associated to Flight-by-flight number

4.2. Hardware Interfaces

The hardware specification for this application is almost non-existent as all you would need to access the application is a computer that can support web browsers and a computer which is actively connected to the internet.

4.3. Software Interfaces

Any window-based operating system with DOS support are primary requirements for software development. Windows 7 and up are required. The system must be connected via LAN and connection to the internet is mandatory.

5. Other Nonfunctional Requirements

5.1. Performance Requirements

The airline ticket reservation must be able to manage five hundred connections. The system must ensure a connection time of five minutes from the last input for each session; after that time, the connection is regarded to have expired. When it expires, a closing operation will be conducted. The usability and connection quality of each user are the goals of this design.

After receiving a user-submitted form, the system will quickly (within 100ms) send out a verification request.

The system will update all flight status information every 5 minutes.

5.2. Safety Requirements

After a certain amount of inactivity, the system must automatically log out all customers. The system should not save any cookies revealing the user's password on the customer's computer. Only authorized management shall have access to the system's back-end servers.

5.3. Security Requirements

Passwords must be between one and seven digits long and have a minimum of eight characters.

Before a user is given access, the system will validate their email addresses. After enrolling, this verification will be conducted by sending the potential user a confirmation email.
Information specific to completing the enrolling process must be included in this email.

Confidential data exchanges between clients and servers must always use the greatest level of security possible (e.g., https).

5.4. Software Quality Attributes

5.4.1. Usability

The website for the airline must be able to run on both Windows and UNIX (Linux) servers. The plan ought to work with Windows Server 2003, Linux 2.6.x, and versions of V10 UNIX and later.

5.4.2. Robustness

The system design must incorporate recovery scenarios that enable the ability to restore data that is no more than one business day old.

6.Other Requirements

Security:

The system must automatically log out all customers after a period of inactivity. The system should not leave any cookies on the customer's computer containing the user's password. The system's back-end servers shall only be accessible to authenticated management.

Reliability:

The reliability of the overall project depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Also, the system will be functional under a container. Thus, the overall stability of the system depends on the stability of the container and its underlying OS.

Availability

The system should be always available, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. A customer friendly system which is more than people around the world should work 24 hours. In case of a hardware failure or database corruption, a replacement page will be shown. Also, in

In case of hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the organizer. Then the service will be restarted. It means 24x7 availability.

Maintainability

In case of a failure, a reinstall of the system will be done. Also, the software design is being done with modularity in mind so that maintainability can be done efficiently.

Supportability

The code and supporting modules of the system will be well documented and easy to understand. Online user documentation and Help system requirements will be provided.

Appendix A: Glossary

A form of storing information/data, usually this data is repeatedly accessed.

CSS

Cascading Style Sheets is a feature to give users and developers more control on how web site pages are displayed.

Database

Is a structured collection of records or data that is stored in a computer system. In our system, this may pertain to flight records or user information.

Dynamic Links

A pointer to a particular scope called during runtime. Encryption Algorithm a mathematical procedure for performing encryption on data, which is translating data into secret code.

HTTP

Hypertext Transfer Protocol is the underlying protocol used by the World Wide Web. It defines how messages are formatted and transmitted and what actions should be taken in response to various commands.

Hyper Links

Also called link, is a directly followable reference within a hypertext document.

Input Criteria

A defined group of criteria, which defines inputs.

Query or Queries

A form of questioning. In this document, a query pertains to a search entered by a user into a search engine to return results.