Software Requirements Specification

for

Airline Reservation System for Bon Voyage Airlines

Version 1.0

Prepared by Tushaar Gangarapu (156307 15IT117)

National Institute of Technology Karnataka

December 29, 2017

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Tushaar Gangarapu	12.29.2017	Creation of the SRS for *Bon Voyage* Airlines	1.0

1. Introduction

1.1 Purpose

This Software Requirement Specification (SRS) outlines the steps taken in the deployment of an Airline Reservation System (ARS); describes the requirements involved in the development of ARS for Bon Voyage Airlines. The main purpose of this software would be to make it convenient for the customers (end users) to book flights as per their requirement so as to check availability, make, modify and/or cancel reservations.

1.2 Document Conventions

Main topics are bolded in heading (whole numbered) followed by subtopics (decimal numbered) and bullets. All acronyms have been introduced with their full names, followed by the acronym in the parenthesis. In the glossary of Appendix A, all the acronyms used within this document are outlined in the alphabetical order.

1.3 Intended Audience and Reading Suggestions

This document is written in response to outlining first phase of the ARS and is intended for use by all the developers, designers, programmers, testers and documentation writers involved in the development of the ARS. This document is best read from beginning to the end to fully grasp the implementation and development details. However, the table of contents can be used to hone in on specific areas of interest to specific users. This document however, is not intended for the use by marketing staff.

1.4 Product Scope

This software (ARS) being developed will provide the following functionality:

- The ARS can be used to check the availability of flight tickets for a specified *Bon Voyage* flight, destination and date of journey.
- If the availability of tickets to customer's needs and specifications is confirmed, then the ARS provides a facility to book the tickets.
- If the passenger wants to cancel his/her tickets, he/she can use the cancellation module of the ARS for the same.

The ARS will be accessible by users with the access to authenticated computers with the working application installed.

1.5 References

[1] Lewis, D. (2016, November 11). Software Requirement Specification for First Class Airline Reservation System - version 1.7. Retrieved from https://www.slideshare.net/DeborahKronk/airline-reservation-software-requirement-specification

[2] Faqeem, A. Airline Reservation System SRS. Retrieved 2017, December 29 from https://www.scribd.com/doc/130966364/Airline-Reservation-System-SRS

[3] SRM University. (2012 - 2013). CS0411 - Software Engineering Lab - Laboratory Manual. Retrieved from http://www.srmuniv.ac.in/sites/default/files/files/SOFTWARE%20Engineering%20LAB-CS0411.pdf

[4] Wiegers K. E. (1999). IEEE Requirements Specification Template. Retrieved from https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc

2. Overall Description

2.1 Product Perspective

The ARS is an independent application and is a self-contained project; eliminating Site Adaptation Requirements, i.e, no modifications are required to adapt to a particular installation.

The ARS will work with the following hardware interfaces:

- *Hard disk*: The database connectivity requires a hardware configuration with a fast database system running on high rpm hard disk permitting complete data redundancy and backup systems to support the primary goal of reliability.
- The system must interface with the standard output device, keyboard and mouse to interact with this software.

The ARS will work with the following software interfaces (both system and user interfaces (UI)):

- *Back End*: mysql Ver 14.14 Distrib 5.7.20, for Linux (x86_64) using EditLine wrapper.
- *Front End*: Netbeans 8.2.

The ARS will stage the following operations:

- The user mode enables the end users to do the end user operations like checking the availability, reserving and canceling of flight tickets.

The ARS must be user friendly and interactiveness must be ensured; functionality provided by the system like displaying error messages should adapt itself to the different users of the software.

2.2 Product Functions

The ARS allows users to perform a series of actions to aid in the *Bon Voyage* airline reservation process. These functions include the following:

- *Viewing Flight Details*: The user must have the access to the most recently updated information about the flights including- Flight number, Flight name, Flight route (Start and Destination stations), Flight timings, Seat availability.
- *Reserving Tickets*: The user must be able to reserve tickets after selecting the following- Flight number, Flight route.
- *Canceling Tickets*: The user must be able to cancel tickets that he has earlier reserved by quoting the ticket number, credit card number and bank name.

Fig. 1. depicts the data flow diagram for *Bon Voyage* airline reservation application.

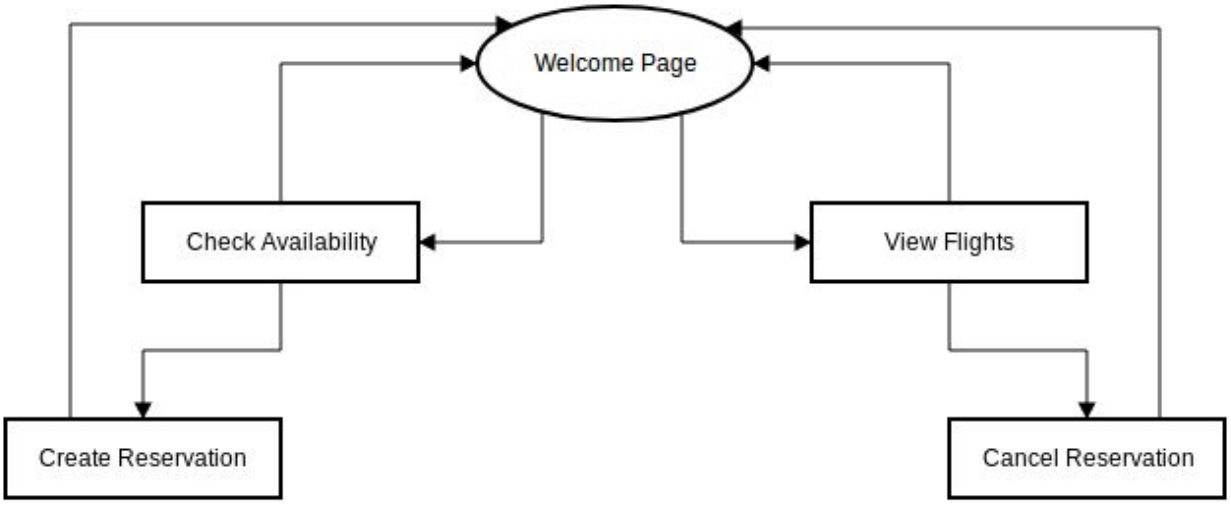

Fig. 1. UML Data flow diagram for *Bon Voyage* ARS

2.3 User Classes and Characteristics

Passengers with the access to the installed application are the target users of the ARS and can utilize the benefits of the ARS application. The characteristics of these users can be described as follows:

- The product is absolutely user friendly, so the intended users can be naive users.
- Need not have specific knowledge as to what the internal operation of the system is. Thus the end user is at a high level of abstraction that allows easier, faster operation and reduces the knowledge requirement of end user.
- May or may not be technically proficient; any person who knows to use the mouse and the keyboard can successfully use this product.
- May or may not have prior experience with travel applications.
- May or may not be educated and trained in the domain.

2.4 Operating Environment

This ARS application has chosen the following elements of the environment for the *Bon Voyage* reservation application:

- Linux or Windows based Operating System (OS) will be used to house the application.
- The language chosen for the deployment of the application would be Java (javac 1.8.0_151).
- The UI will be handled by Netbeans, a Java Integrated Development Environment (IDE).
- The database which holds customer information and flight details is deployed in MySQL.

2.5 Design and Implementation Constraints

The following are the constraints laid before utilizing the ARS application in full scale:

- It is the responsibility of the potential traveler to ensure that he books tickets from authenticated computers or servers hosting the application.
- *Reliability requirements*: Data redundancy and use of special characters must be avoided.
- *Safety and Security considerations*: The user must always exit the application normally.
- *Programming language requirements*: The developers and programmers must have experience with Java and MySQL.
- At the time of reservation, each user is provided a unique ticket number that must be used for further operation like cancellation. Hence the user is required to remember or store this number carefully.
- The necessity of providing options to customer to choose their seat or to choose the economic or business class can be delayed until future versions of the software are developed.

2.6 User Documentation

The *Bon Voyage* reservation application will use very intuitive design, leaving the following considerations:

- A Portable Document Format (PDF) User manual will be provided by the documentation writers to the *Bon Voyage* airlines with an overview of application and its functionalities.
- It is the responsibility of the *Bon Voyage* airlines to provide their users with instructions on the use of the application from a users' viewpoint.
- The application will hold a *help* option, which can be accessed on the main page of the application. When the user clicks on this option, a help center article provided by *Bon Voyage* airlines will be displayed.

2.7 Assumptions and Dependencies

The following assumptions have been made while deploying the ARS:

- The application is used only by confirmed passengers and potential passengers of *Bon Voyage*.
- Users of the application are knowledgeable about using applications on a computer system.
- It is the responsibility of *Bon Voyage* airlines to keep the application updated with the most current content.

3. External Interface Requirements

3.1 User Interfaces

The ARS application has four main interfaces with the help page excluded, that the users will view. Initial configurations of these interfaces are provided below. The interfaces must be easy to understand and the user interface typically includes the following:

- *Screen Formats*: The introductory screen (welcome page) will be the first to be displayed which will allow the users to choose one of the five available options (help and exit options included).
- *Window Formats*: When the user chooses one of the available options, the corresponding page associated with the choice will be displayed in a new window, which ensures multiple windows (single instance of a window) to be visible on the screen.
- *Data Formats*: The data and information presented by and to the user will be alphanumeric.
- *End Messages*: Appropriate error messages will be displayed, guiding the user to get out of such scenarios.

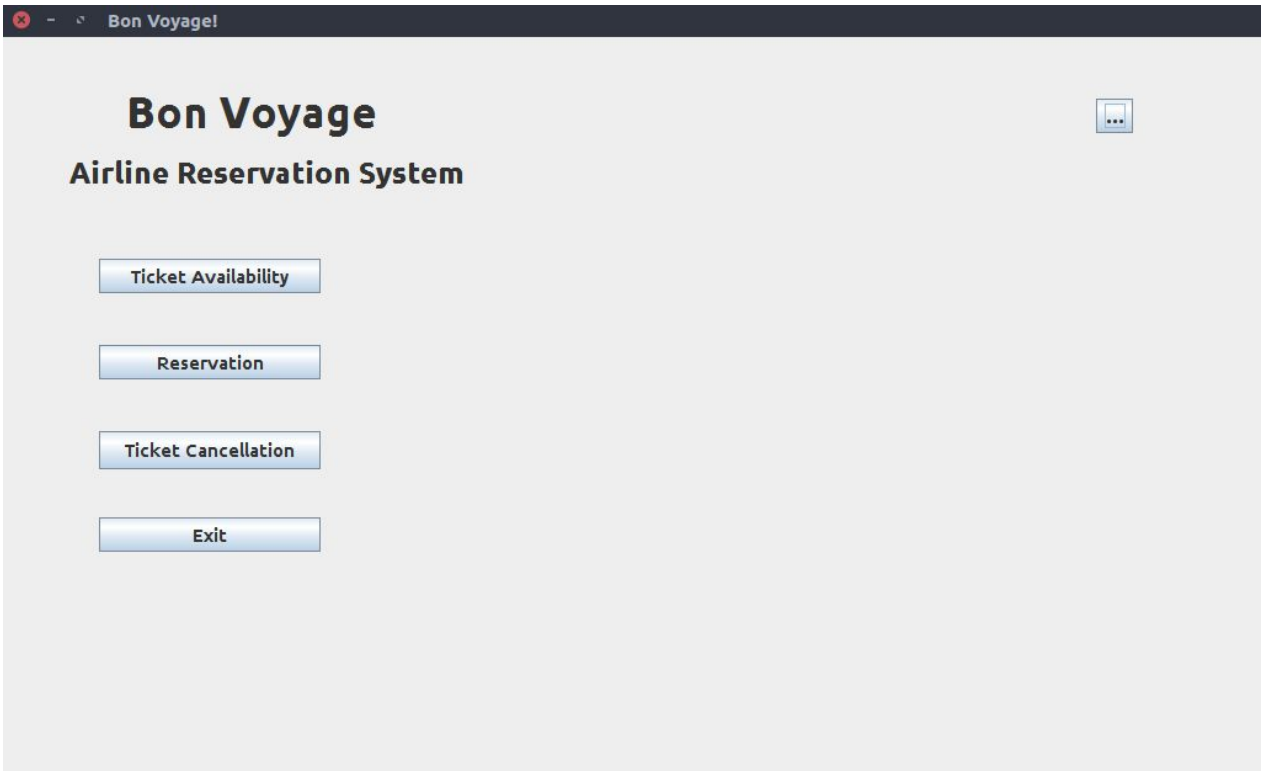

Fig. 2. Welcome Interface

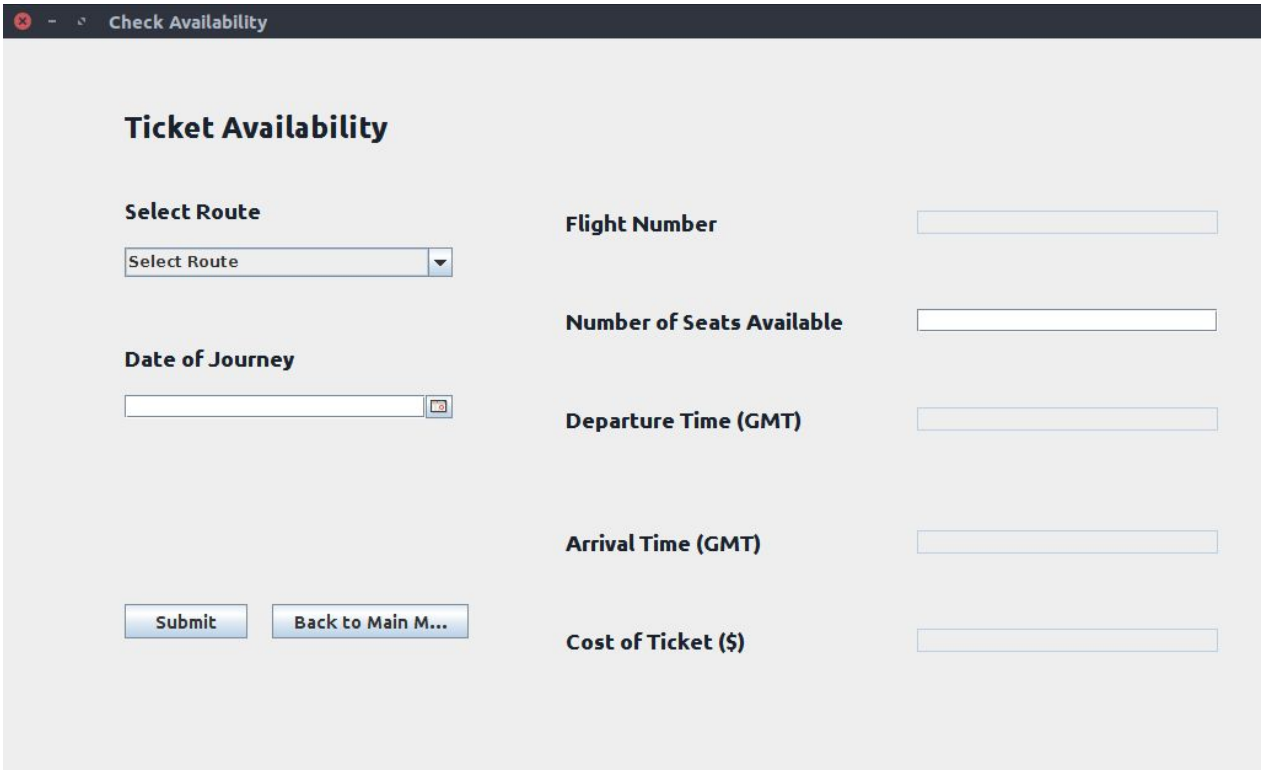

Fig. 3. Availability Checking Interface

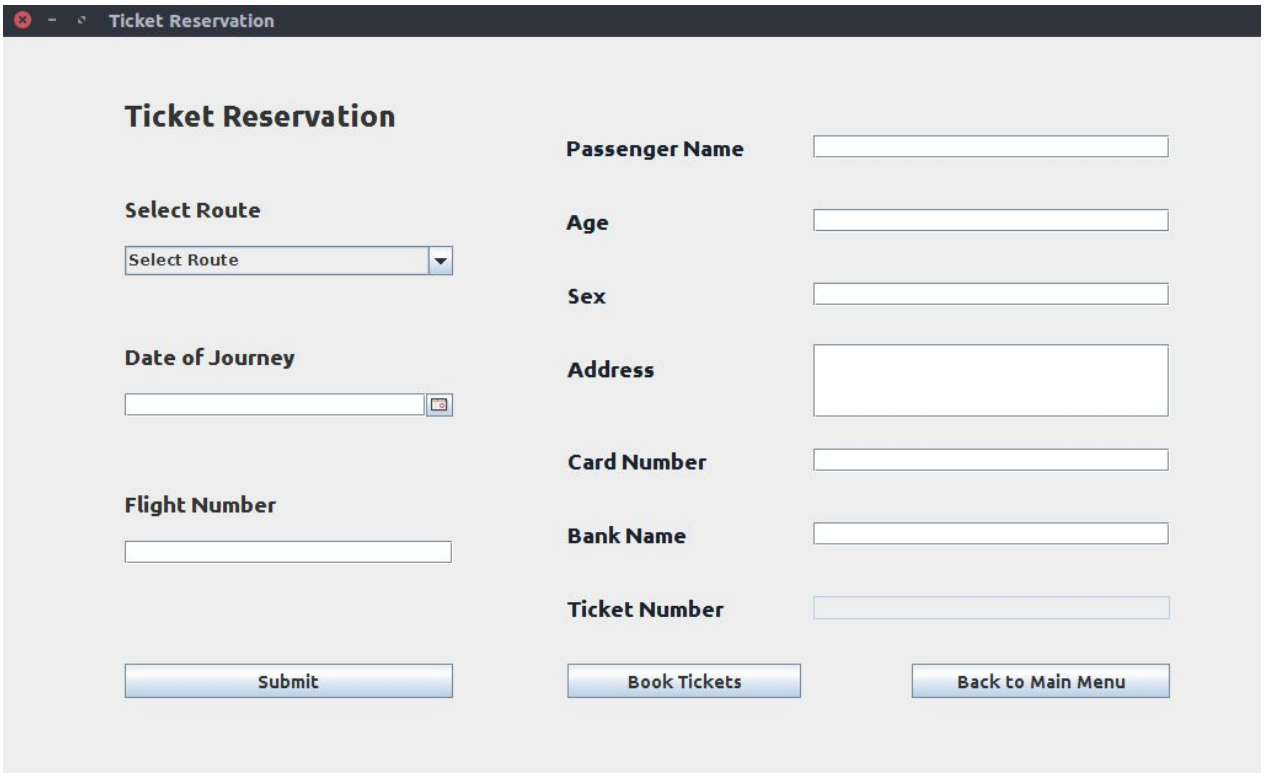

Fig. 4. Ticket Reservation Interface

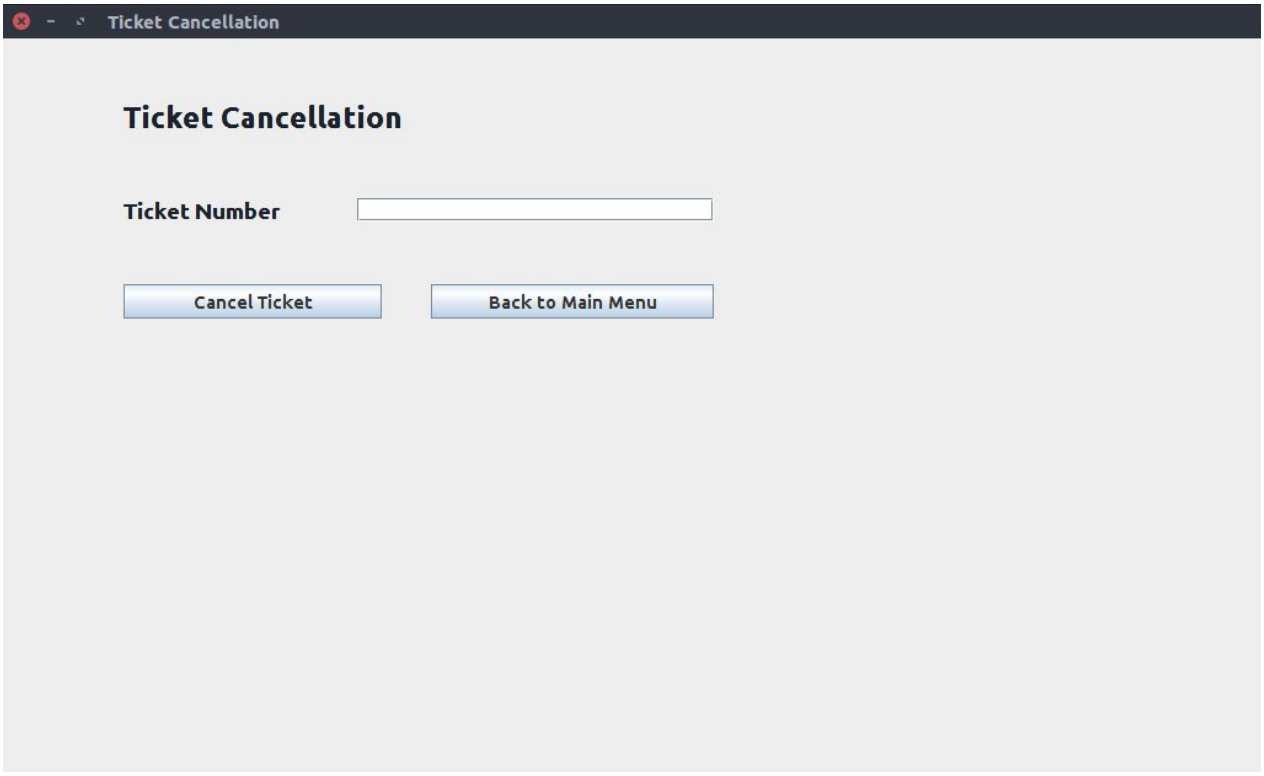

Fig. 5. Ticket Cancellation Interface

Figures 1 through 4, depict visual mockups (prototypes) of the four main interfaces of the ARS application including the welcome page, availability checking page, booking page and cancellation page.

3.2 Hardware Interfaces

The system must basically support certain input and output devices, whose descriptions are described as in Table 1.

Table 1. List of Hardware Interfaces

Name of the Interface	Description or Purpose
Keyboard and Mouse	To accept data from the user like name, address etc.
Processor Number	1 or 2 per node
Processor Model	i5 6200U
Processor Speed	2.5 GHz

Processor Cores	8, 6 or 4
Hard disk	Fast and reliable database connectivity
Memory	8GB DDR3 @ 2.8GHz

3.3 Software Interfaces

The following software interfaces are used in the deployment of the ARS:

MySQL Database Management System (DBMS):

Table 2. List of Software Interfaces with respect to MySQL DBMS

Name of the Interface	Description or Purpose
Type System	Dynamic, Static
Architecture	Relational Model
Software License	Proprietary
Operating System	Windows, Mac OS X, Linux, Unix
Max DB Size	Unlimited
Max Table Size	4 GB
Max Row Size	8 KB
Max Column Name Size	30
Max Columns per Row	1000

The front end of the application is developed using Netbeans 8.2 IDE with Java as the development language and Ubuntu 16.04 is used as the base operating system.

3.4 Communications Interfaces

Not applicable as the application being developed is a stand-alone application that can be accessed only via authorized computers. In the future, we may maintain a server that hosts the application, where in which, every client system would require a secure Local Area Network (LAN) connection via Secure Sockets Layer (SSL) for security, to communicate with the server.

4. System Features

This section outlines the low level details (implementation) of every system function of the ARS application system. It sets the priorities of each function so that the developers and the investors would have a lucid picture of what will be accomplished first and what functions will be available to review first.

4.1 Search for Flights

4.1.1 Description and Priority

Upon successful entry into the *Bon Voyage* Reservation Application, user can select *Check Availability* from the application's welcome page. This allows users to search for available flights based on route and date of travel. This function is one of the main aspect to flight ticket booking, only not so crucial, so its feature is of medium priority.

4.1.2 Stimulus/Response Sequences

After a user selects *Check Availability* from the *Bon Voyage* Reservation welcome page, a new interface opens. The application accepts an entry of the travel route and travel date from the user and submits this request to the existing Database. Then an available *Bon Voyage* flight that meets the user's selection is presented to the user. If no flights are available, the user will be provided with an appropriate message.

4.1.3 Functional Requirements

SF-1: A search form will be provided to the user that requests the travel route and travel dates.

SF-2: A submit button submits the entered data.

SF-3: Data is validated for completeness, accurate dates, and valid route options.

SF-4: If the data is complete and valid, the FCA Reservation application will request a matching flight from the deployed database.

SF-5: If a matching flight exist, the *Bon Voyage* Reservation application will display a matching *Bon Voyage* flight to the user.

SF-6: If no matching flight exists, an appropriate error message will be presented to the user.

4.2 Create New Reservation

4.2.1 Description and Priority

After checking the availability of a *Bon Voyage* flight, users can create a reservation for the selected flight. Users will then submit their personal data to secure their seat for the flight. This function is one of the main aspect to flight ticket booking, only not so crucial, so its feature is of medium priority.

4.2.2 Stimulus/Response Sequences

After an available flight is selected, the *Bon Voyage* Reservation application displays a reservation form with travel route, flight number and travel date fields. Upon submission, the *Bon Voyage* Reservation application provides another form to fill the passenger details, with a unique ticket number (UTN). If the seats are not available on the flight selected an appropriate error message is displayed to the user.

4.2.3 Functional Requirements

CNR-1: A search form will be provided to the user that requests the travel route and travel dates along with the selected flight number.

CNR-2: Display a reservation form to users to enter their personal and banking information to book tickets.

CNR-3: A *book tickets* button that submits the entered data.

CNR-4: Data is validated for completeness.

CNR-5: If data is missing, an error message is displayed to the user.

CNR-6: If data is complete, the *Bon Voyage* Reservation application will submit the user's personal and banking to the Database for storage upon selection of the submit button.

CNR-7: If the seats are not available on the flight selected an appropriate error message is displayed to the user.

4.3 Cancel Reservation

4.1.1 Description and Priority

Users who have previously made a reservation can cancel a reservation for flights scheduled to depart more than a day (24 hours) in the future. This functionality is not core to the vision of the application as a whole, so its priority is low.

4.1.2 Stimulus/Response Sequences

A user who has a confirmed reservation selects *Ticket Cancellation* option from the welcome page. For flights scheduled to depart within a day's time, the user is given an error message. For flights scheduled to depart more than 24 hours in the future, the cancellation using the UTN can be done, which updates the existing Database.

4.1.3 Functional Requirements

CR-1: For flights scheduled to depart within the next 24 hours, an error message will be displayed to the user.

CR-2: For flights scheduled to depart 24 hours or more in the future, the *Bon Voyage* Reservation application will cancel the reserved ticket using the USN and update the existing Database.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The following are the performance requirements of the *Bon Voyage* reservation application:

Table 3. Performance Requirements

Requirement	Description
System Availability	Unscheduled downtime and maintenance must be addressed and resolved in as little time as possible.
System Responsiveness	The system must be responsive and interactive.
System Response	The system must take lesser time to respond to any request and return data from the Database.

5.2 Safety Requirements

The following are the safety requirements of the *Bon Voyage* reservation application:

Table 4. Safety Requirements

Requirement	Description
System Accessibility	The application responds to the Netbeans settings for font styles and sizes which includes default settings and special accommodations.
Soft Color Scheme	The application uses color scheme that does not use high contrast colors for reduced eye strain.
Lost or Low Battery	The application closes when the computer's battery is depreciated, which requires user to restart the application on the battery recovery.

5.3 Security Requirements

The following are the security requirements of the *Bon Voyage* reservation application:

Table 5. Security Requirements

Requirement	Description
Cache	The application will not store any personal data on the device.

Memory Protection	The application performs garbage collection on closing it.

5.4 Software Quality Attributes

The following are the software quality attributes for the *Bon Voyage* reservation application:

Table 6. Software Quality Attributes

Requirement	Description
Font Family	The application uses Netbeans default fonts.
Font Size	The application uses default 12-point to 24-point font sizes.
Usability	The application's usability should be high; ease of use must be ensured.
Help Icon	Help icon on the welcome page can be used to access the instruction manual for the application.
Updates	Each update of the application includes versioning for troubleshooting and traceability purposes.
Reusability	The application is deployed using Object Oriented Programming (OOP), which allows the design features to be reused.

5.5 Business Rules

The following business rules can be implied in the deployment of *Bon Voyage* ARS:

Table 7. Business Rules

Individual Role	Access or Circumstance
Database Administrators	Access to the database used in the application.
Java Programmers	Access to the back-end of the application.
End Users	Access to the UI or front-end of the application.

Appendix A: Glossary

Terms and acronyms which may not be familiar to all readers are provided in this section. Definitions provided are limited to their scope within this document. All terms are listed in alphabetical order for easy reference.

Table 8. Glossary of Terms used

Term	Definition
ARS- Airline Reservation System	An application that allows users to search for available flights, make a reservation, view reservations, and cancel reservations.
DBMS- Database Management System	Management of a database which is an organized collection of data.
IDE- Integrated Development Environment	It is a software application that provides comprehensive facilities to computer programmers for software development.
LAN- Local Area Network	A local area network is a computer network that interconnects computers within a limited area such as a residence, school, laboratory, university campus or office building.
OOP- Object Oriented Programming	Object-oriented programming is a programming paradigm based on the concept of *objects*.
OS- Operating System	The software that supports the basic functions of the device on which it is installed.
PDF- Portable Document Format	A file format that provides an electronic image of text and graphics that can be transmitted.
RTM- Requirements Traceability Matrix	A table used to track the requirements of the *Bon Voyage* reservation system.
SRS- Software Requirements Specification	A description of software to be developed.
SSL- Secure Sockets Layer	It is a standard security protocol for establishing encrypted links in an online communication.
UI- User Interface	The space where interactions between humans and machines occur.
UTN- Unique Ticket Number	Unique number presented while reserving tickets.

Appendix B: Analysis Models

UC-1: Check Availability

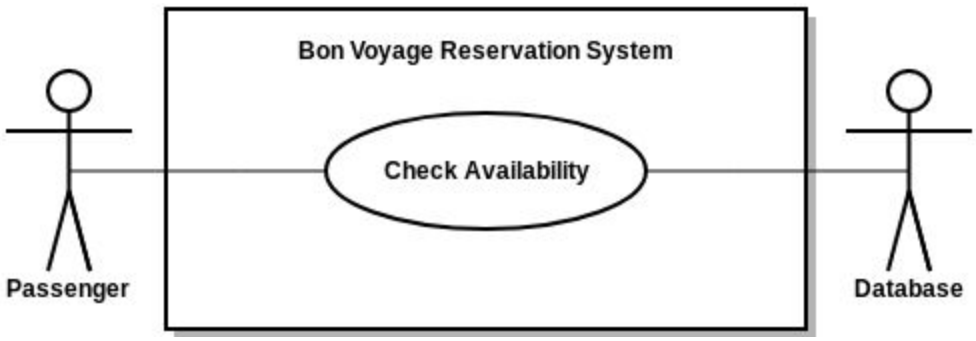

Fig. 6. Use-case diagram to check availability

The above use case diagram has been elucidated in Table 9.

Table 9. UC-1 to check availability

Term	Explanation
Actors	Passenger, *Bon Voyage* Database.
Description	The passenger searches for available flights based on route and travel date.
Trigger	The passenger selected *Check Availability* option from welcome page.
Flow	The user selects the welcome page option to look for available flights. The user enters travel route and travel date. The most relevant *Bon Voyage* flight is presented to the user. If no flights are available or the seats are not available, then an appropriate error message is displayed to the user.
Business Rules	User must enter the travel route (departure and arrival locations) and travel date
Exceptions	None.
Priority	Medium.
Frequency of Use	Each time the *Check Availability* option is selected to search for available and relevant *Bon Voyage* flights.
Assumptions	*Bon Voyage* maintains a current and most accurate database on the authenticated system(s).

UC-2: Create New Reservation

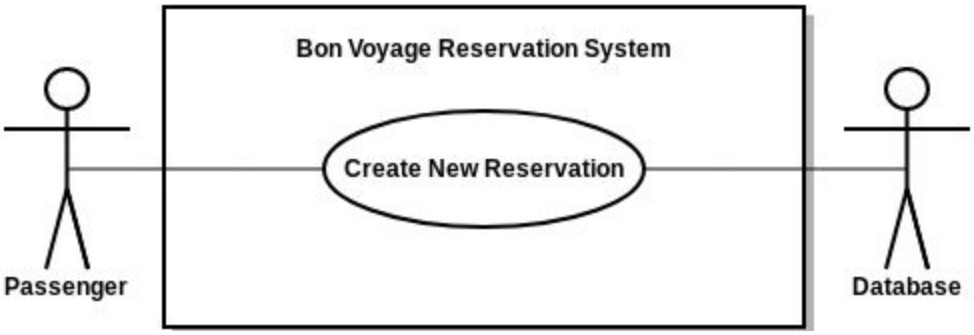

Fig. 7. Use-case diagram to create new reservation

The above use case diagram has been elucidated in Table 10.

Table 10. UC-2 to create new reservation

Term	Explanation
Actors	Passenger, *Bon Voyage* Database.
Description	After the best suitable or available flight has been presented to the user, the user creates a reservation to secure his/her spot on the desired flight.
Trigger	The user chooses *Book Ticket* option on the welcome page.
Flow	The user selects the *Book Ticket* option, from the welcome page and enters the travel route, travel date and the desired flight number. Then the user is prompted to fill a form with pre-determined UTN to secure a reservation for the flight provided. User submits his/her valid personal and banking data. If no seats are available at the time of submission, an appropriate error message is displayed.
Business Rules	All flights must be paid in full for the reservation to be successfully created.
Exceptions	A dummy card number (of type 4111 1111 1111 1111) will allow administrators to make a fake reservation. Also parallelism when servers are deployed is a serious concern (synchronization must be ensured).
Priority	Medium.
Frequency of Use	Any time an available flight is selected.
Assumptions	*Bon Voyage* takes care of the banking and money transactions; also *Bon Voyage* provides developers with the reservation form and data that is to be requested from the end user.

UC-3: Cancel Reservation

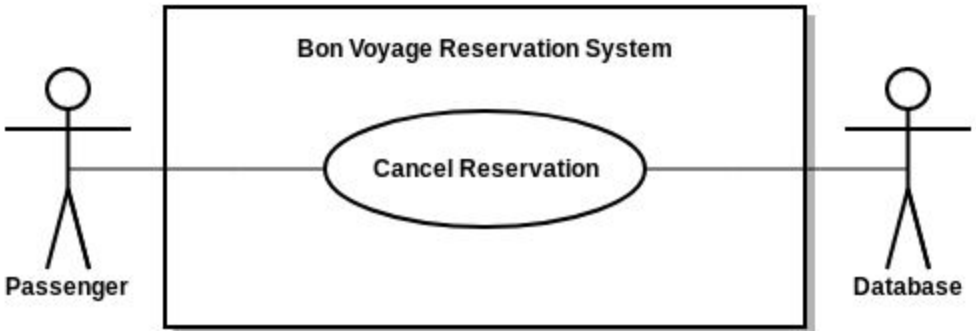

Fig. 8. Use-case diagram to cancel reservation

The above use case diagram has been elucidated in Table 11.

Table 11. UC-3 to cancel reservation

Term	Explanation
Actors	Passenger, *Bon Voyage* Database.
Description	A passenger cancels their reservation for a flight that is scheduled for at least a day in the future.
Trigger	The passenger selects *Cancel Reservation* option from the welcome page.
Flow	For flights scheduled to depart 24 hours or more in the future, the *Bon Voyage* Reservation application will cancel the reserved ticket using the USN and update the existing Database. For flights scheduled to depart within the next 24 hours, an error message will be displayed to the user.
Business Rules	If the flight is scheduled for departure within the next 24 hours, customers must contact *Bon Voyage* for both cancellation and refund (if applicable).
Exceptions	None.
Priority	Low
Frequency of Use	Any time *Cancel Reservation* is selected from the welcome page.
Assumptions	The user has a reservation.

Appendix C: Traceability Matrix

The Requirements Traceability Matrix (RTM) tracks system features throughout the software validation process. It ensures that each of the requirements defined for the ARS application are included for full functionality.

Table 7. RTM

System Features	Requirement Specification	Use Case	Specific Requirement
SF-1	Checking Availability	UC-1, UC-2	SF-1, SF-2, SF-3, SF-4, SF-5, SF-6
SF-2	Create New Reservation	UC-2	CNR-1, CNR-2, CNR-3, CNR-4, CNR-5, CNR-6, CNR-7
SF-3	Cancel Reservation	UC-3	CR-1, CR-2