Software Requirements Specification BlazeDemo

Online flight management System

**Prepared by   
 Abirajan V**

# **1.1 PURPOSE**:

# The Web application helps the user to manage the flight and allow the passengers to book tickets for present and future journey and it allow the user to cancel the tickets. It also allows the users to make their payment on own.

# **1.2 DOCUMENT CONVENTIONS**:

# This document uses the following conventions.

|  |  |
| --- | --- |
| DB | Database |
| DDB | Distributed Database |
| ER | Entity Relationship |

# **1.3 INTENDED AUDIENCE AND READING SUGGESTIONS:**

# This project is a document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. This has been implemented under the guidance of college professors. This project is useful for the flight management team and as well as to the passengers.

# **1.4 PROJECT SCOPE**:

The main scope of the web application that the passengers can manage their own journey and book their own ticket from anywhere in the world. And it allow the user can pay for their ticket using credit, debit, net-banking and UPI methods.

# **1.5 REFERENCES**:

# https://blazedemo.com/

# **2. OVERALL DESCRIPTION:**

### **2.1 PRODUCT PERSPECTIVE**:

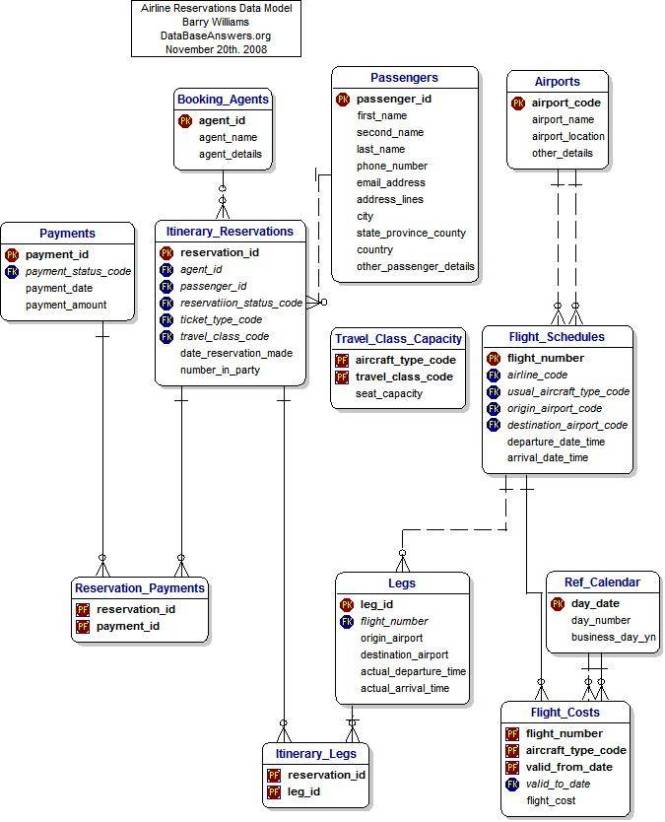
### A distributed airline database system stores the following information.

### Flight details: It includes the originating flight terminal and destination terminal, along with the stops in between, the number of seats booked/available seats between two destinations etc.

### Customer description: It includes customer code, name, address and phone number. This information may be used for keeping the records of the customer for any emergency or for any other kind of information. Reservation description: It includes customer details, code number, flight number, date of booking, date of travel.

### **2.2 PRODUCT FEATURES**:

### The major features of airline database system as shown in below [entity–relationship model](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model) (ER model)

The diagram shows the layout of airline database[](https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database/attachment/ad-layout-of-airline-database-system) system – entity–relationship model

**2.3 USER CLASS and CHARACTERISTICS**:

Users of the system should be able to retrieve flight information between two given cities with the given date/time of travel from the database. A route from city A to city B is a sequence of connecting flights from A to B such that: a) there are at most two connecting stops, excluding the starting city and destination city of the trip, b) the connecting time is between one to two hours. The system will support two types of user privileges, Customer, and Employee. Customers will have access to customer functions, and the employees will have access to both customer and flight management functions. The customer should be able to do the following functions:

* Make a new reservation

• One-way  
• Round-Trip  
• multi-city  
• Flexible Date/time  
• Confirmation

* Cancel an existing reservation
* View his itinerary

The Employee should have following management functionalities:

* CUSTOMER FUNCTIONS.  
  • Get all customers who have seats reserved on a given flight.  
  • Get all flights for a given airport.  
  • View flight schedule.  
  • Get all flights whose arrival and departure times are on time/delayed.  
  • Calculate total sales for a given flight.
* ADMINISTRATIVE  
  • Add/Delete a flight  
  • Add a new airport  
  • Update fare for flights.  
  • Add a new flight leg instance.  
  • Update departure/arrival times for flight leg instances.

Each flight has a limited number of available seats. There are a number of flights which depart from or arrive at different cities on different dates and time.

**2.4 OPERATING ENVIRONMENT**:

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

* Operating system: Windows.
* database: SQL + database
* platform: vb.net/Java/PHP

**2.5 DESIGN and IMPLEMENTATION CONSTRAINTS**:

1. The global schema, fragmentation schema, and allocation schema.
2. SQL commands for above queries/applications
3. How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.
4. Implement the database at least using a centralized database management system.

**2.6 ASSUMPTION DEPENDENCIES**:

Let us assume that this is a distributed airline management system and it is used in the following application:

* A request for booking/cancellation of a flight from any source to any destination, giving connected flights in case no direct flight between the specified Source-Destination pair exist.
* Calculation of high fliers (most frequent fliers) and calculating appropriate reward points for these fliers.

Assuming both the transactions are single transactions, we have designed a distributed database that is geographically dispersed at four cities Delhi, Mumbai, Chennai, and Kolkata

**3. SYSTEM FEATURES**

* **DESCRIPTION and PRIORITY**

The airline reservation system maintains information on flights, classes of seats, personal preferences, prices, and bookings. Of course, this project has a high priority because it is very difficult to travel across countries without prior reservations.

* **STIMULUS/RESPONSE SEQUENCES**
  + Search for Airline Flights for two Travel cities
  + Displays a detailed list of available flights and make a Reservation or book a ticket on a particular flight.
  + Cancel an existing Reservation.
* **FUNCTIONAL REQUIREMENTS**

Other system features include:

**DISTRIBUTED DATABASE:**

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network.

**CLIENT/SERVER SYSTEM**: The term client/server refers primarily to an architecture or logical division of responsibility, the client is the application

**5**.**EXTERNAL INTERFACE REQUIREMENTS**

**4.1 USER INTERFACES**

* Front-end software: Vb.net version
* Back-end software: SQL+

**4.2 HARDWARE INTERFACES**

* Windows.
* A browser which supports CGI, HTML & Javascript.2

**4.3 SOFTWARE INTERFACES**

Following are the software used for the flight management online application.

|  |  |
| --- | --- |
| **Software used** | **Description** |
| Operating system | We have chosen Windows operating system for its best support and user-friendliness. |
| Database | To save the flight records, passengers record we have chosen SQL+ database. |
| VB.Net | To implement the project we have chosen Vb.Net language for its more interactive support. |

**4.4 COMMUNICATION INTERFACES**:

This project supports all types of web browsers. We are using simple electronic forms for the reservation forms, ticket booking etc.  
  
**5.NONFUNCTIONAL REQUIREMENTS**

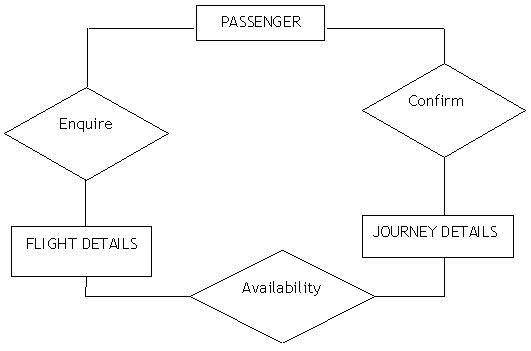
**5.1 PERFORMANCE REQUIREMENTS:**

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

* 1. E-R DIAGRAM

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

* ENTITIES: Which specify distinct real-world items in an application.
* PROPERTIES/ATTRIBUTES: Which specify properties of an entity and relationships.
* RELATIONSHIPS: Which connect entities and represent meaningful dependencies between them.

[](https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database/attachment/ad-er-diagram-of-airline-database)

the diagram shows the ER diagram of airline database

**B.NORMALIZATION:**

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

**5.2 SAFETY REQUIREMENTS:**

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

5.3 SECURITY REQUIREMENTS

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

**5.4 SOFTWARE QUALITY ATTRIBUTES:**

* AVAILABILITY: The flight should be available on the specified date and specified time as many customers are doing advance reservations.
* CORRECTNESS: The flight should reach start from correct start terminal and should reach the correct destination.
* MAINTAINABILITY: The administrators and flight in chargers should maintain correct schedules of flights.
* USABILITY: The flight schedules should satisfy a maximum number of customer’s needs.