Software Requirements Specification

for

<Project>

Version 1.0 approved

Prepared by <jehad alqurashi>

<flight corporation>

<25/nov>

Table of Contents

Table of Contents ..... **ii**
Revision History ..... **ii**
1. Introduction **3**
1.1 Purpose 3
1.2 Document Conventions 3
1.3 Intended Audience and Reading Suggestions 3
1.4 Product Scope 3
1.5 References ..... **خطأ! الإشارة المرجعية غير معرّفة.**
2. Overall Description **3**
2.1 Product Perspective 3
2.2 Product Functions 4
2.3 User Classes and Characteristics 4
2.4 Operating Environment 5
2.5 Design and Implementation Constraints 5
2.6 User Documentation ..... **خطأ! الإشارة المرجعية غير معرّفة.**
2.7 Assumptions and Dependencies 6
3. External Interface Requirements ..... خطأ! الإشارة المرجعية غير معرّفة.
3.1 User Interfaces ..... **خطأ! الإشارة المرجعية غير معرّفة.**
3.2 Hardware Interfaces ..... **خطأ! الإشارة المرجعية غير معرّفة.**
3.3 Software Interfaces ..... **خطأ! الإشارة المرجعية غير معرّفة.**
3.4 Communications Interfaces ..... **خطأ! الإشارة المرجعية غير معرّفة.**
4. System Features ..... خطأ! الإشارة المرجعية غير معرّفة.
4.1 System Feature 1 ..... **خطأ! الإشارة المرجعية غير معرّفة.**
4.2 System Feature 2 (and so on) ..... **خطأ! الإشارة المرجعية غير معرّفة.**

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to build an online system to manage flights and passengers to ease the flight management.

1.2 Document Conventions

This document uses the following conventions.

DB	Database
DDB	Distributed Database
ER	Intity Relationship

1.3 Intended Audience and Reading Suggestions

This project is a prototype for the flight management system and it is restricted within the college premises. 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 Product Scope

The purpose of the online flight management system is to ease flight management and to create a convenient and easy-to-use application for passengers, trying to buy airline tickets. The system is based on a relational database with its flight management and reservation functions. We will have a database server supporting hundreds of major cities around the world as well as thousands of flights by various airline companies. Above all, we hope to provide a comfortable user experience along with the best pricing available.

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.

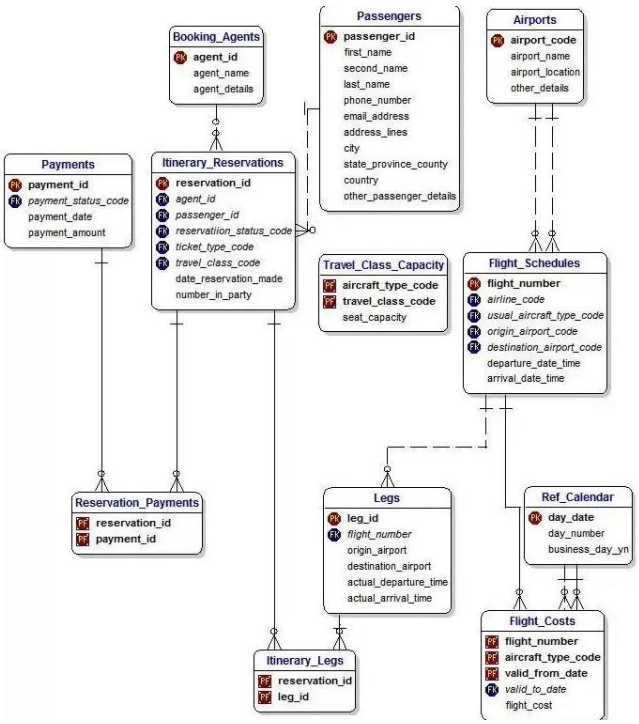
- **Customer description:**
  It includes customer code, name, address and phone number. This

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 Functions

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



2.3 User Classes 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 distributed database

- client/server system
- 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.

2.7 Assumptions and 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 Kolkatta

3.EXTERNAL INTERFACE REQUIREMENTS

3.1 USER INTERFACES

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

3.2 HARDWARE INTERFACES

- Windows.
- A browser which supports CGI, HTML & Javascript.

3.3 SOFTWARE INTERFACES

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 records we have chosen SQL+ database.
VB.Net	To implement the project we have chosen Vb.Net language for its more interactive support.

3.4 COMMUNICATION INTERFACES

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