CHAPTER 1 : INTRODUCTION

1 Introduction

The economic status of a nation is measured by several parameters and transport sector is one main parameter among all.

The Airline Reservation System project is an implementation of a general Airline Ticketing website like Orbitz and MakeMyTrip, which helps the customers to search the availability and prices of various airline tickets, along with the different packages available with the reservations. This project also covers various features like online registration of the users, by adding, deleting or modifying the customer details, flights or packages information. In general, this website would be designed to perform like any other airline ticketing website available online.

The website is named as **ReplicaDB** and is the database to access and book the domestic flights whose cost of service extremely reasonable.

1.1 Aim

To simulate an airline reservation database using which people can find flight schedule and enjoy all sorts of services provided by a particular airline company.

1.2 Objective

The main objectives of our project are :

- To explore the features of Database Management System.
- Experimenting basic, intermediate and advanced SQL.
- Entity Relationship Modeling (ER diagram).
- Database design and normalization.
- Accessing database using suitable front-end tool(s).
- Query processing to use databases for applications.
- Gaining knowledge about the internals of the database system.

1.3 Scope

The scope of the project is as follows :

- The project gives the provision to book one-way journey ticket to the users.
- The database is exclusively for domestic flights with few destinations.
- The front-end is designed only for the client/user end and not the admin end as linking admin end is not a secure plan because of threats of SQL injections and similar malicious activities.

CHAPTER 2 : SYSTEM REQUIREMENTS AND ANALYSIS

2 System Requirements

2.1 Software Requirements

- Operating System : Windows 10/ Ubuntu 16.04 or equivalent.
- Software : Sublime Text editor, MySQL server-client, any stable browser.
- Front-end : Python-Flask web framework
- Back-end : MySQL server-client version 5.7
- Middleware : Python MySQLdb module.

2.2 Hardware Requirements

- Processor : 2.2 GHz
- RAM : 4 GB
- Input Device : Standard keyboard and mouse
- Output Device : Monitors with decent resolution.

2.3 Functional Requirements

- Collection of flight schedule :
  The schedule of the flight services from different airline companies is to be collected and populated into the database.
- Redundancy check:
  The schedule must be verified for redundancy and clashes in the runway timings.
- Graphical User Interface :
  A graphical user interface has been designed to pass the control to the built system and display the results.

2.4 Non-Functional requirements

- Usability :
  The graphical user interface (GUI must be user-friendly.
- Accuracy and perfection :
  The results of the queried operation must generate the required results and make the expected changes as well properly with highest accuracy.

2.5 Input Requirements

- Input of city names as source and destinations.
- Input of date of journey and number of seats to be booked.

2.6 Output Requirements

- Acknowledge of proper ticket generation and updation of the database.

CHAPTER 3 : SYSTEM DESIGN AND ANALYSIS

3 System Design

3.1 Database Implementation

3.1.1 Tools used

MySQL server-client was employed for the implementation of the database which has the entities and relations as shown in the ER-diagram in figure 1. MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

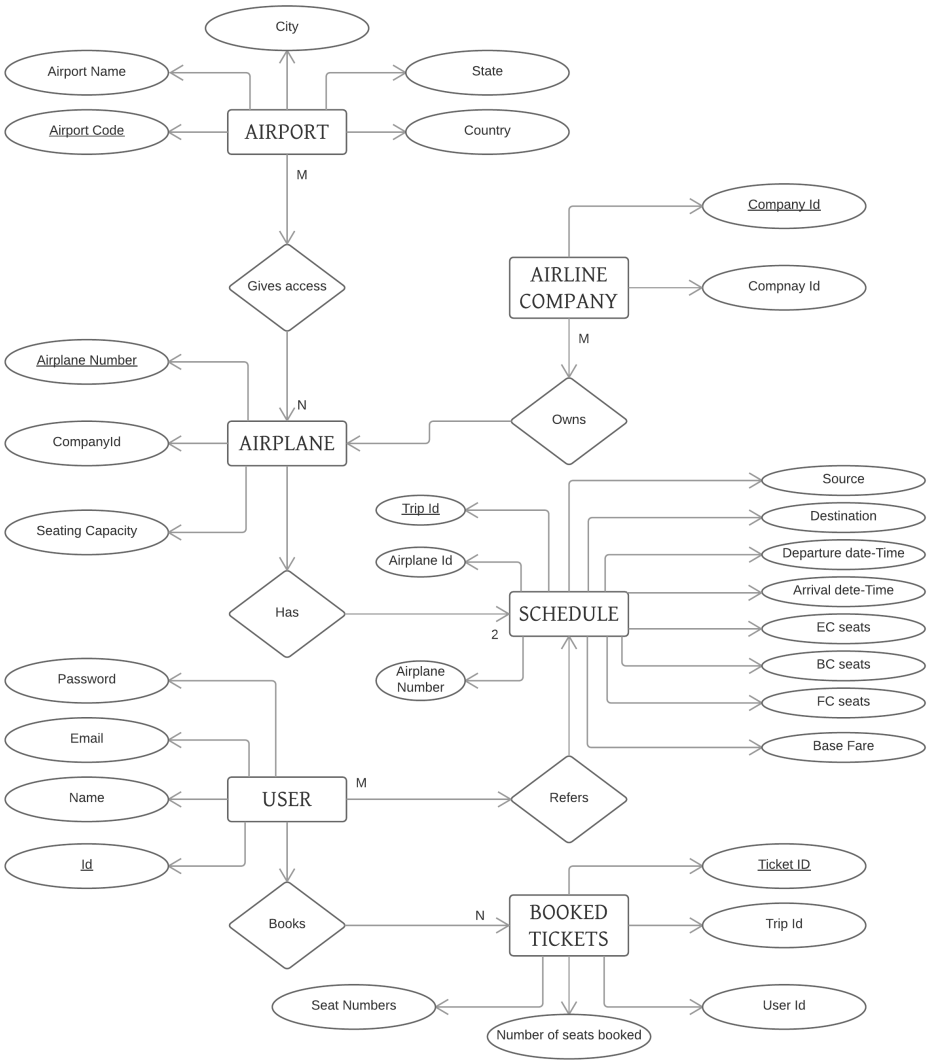

Figure 1: Typical ER Diagram of the proposed database

The replicaDB has six entities as shown in the ER diagram which are Airline Company, Airplane, Airport, Booked Tickets, Schedule and Users whose attributes go like this:

- Airport : Airport Code, Airport Name, City, State, Country
- Airline Company : Company Id, Company Name
- Airplane : Airplane Number, Seating Capacity, Company Id
- Schedule: Trip Id, Airline Id, Airplane Id, Source, Destination, Depart, Arrival, Remaining seats in Economy Class, Remaining seats in Business Class, Remaining seats in First Class, Base Fare
- Booked Tickets : Ticket Id, user Id, trip Id, Count, Class, Allotted Seat Numbers

The MySQL queries to create the above tables are as follows (in the sequence as above):

```
CREATE airport (airportCode INT PRIMARY KEY AUTO_INCREMENT,
airportName VARCHAR(100) NOT NULL, city VARCHAR(50) NOT NULL,
state VARCHAR(50) NOT NULL, country VARCHAR(50) NOT NULL);

CREATE airlineCompany (companyId INT PRIMARY KEY AUTO_INCREMENT,
companyName VARCHAR(100) NOT NULL);

CREATE airplane (airplaneNumber INT PRIMARY KEY AUTO_INCREMENT,
seatingCapacity INT NOT NULL, companyId INT NOT NULL);

CREATE schedule (tripId INT PRIMARY KEY AUTO_INCREMENT,
airlineId INT NOT NULL, airplaneId INT NOT NULL, source VARCHAR(100)
NOT NULL, destination VARCHAR(100) NOT NULL,
depart DATE, arrival DATE, economyClass INT NOT NULL,
businessClass INT NOT NULL, fisrtClass INT NOT NULL,
baseFare INT NOT NULL);

CREATE bookedTickets (ticketId INT PRIMARY KEY AUTO_INCREMENT,
userId INT NOT NULL, tripId INT NOT NULL,count INT NOT NULL,
class VARCHAR(50) NOT NULL, seatNumbers VARCHAR(100));
```