1. INTRODUCTION

1.1 PURPOSE AND MOTIVATION The main purpose of this vision document is to list the requirements of the Airline Reservation System project. This document also helps us to collect and analyze the ideas gathered for the project. This vision document will be subject to change, if more requirements are added to the project. This document is mainly prepared to set stage for the design phase of the project. The document being prepared is the first version of vision document for the Airline Reservation System project. C#.NET is a new technology which is being used a lot in the IT field. My interest to learn this new technology has prompted me to take up this project, which would set the stage for the applications I would be developing in the future.

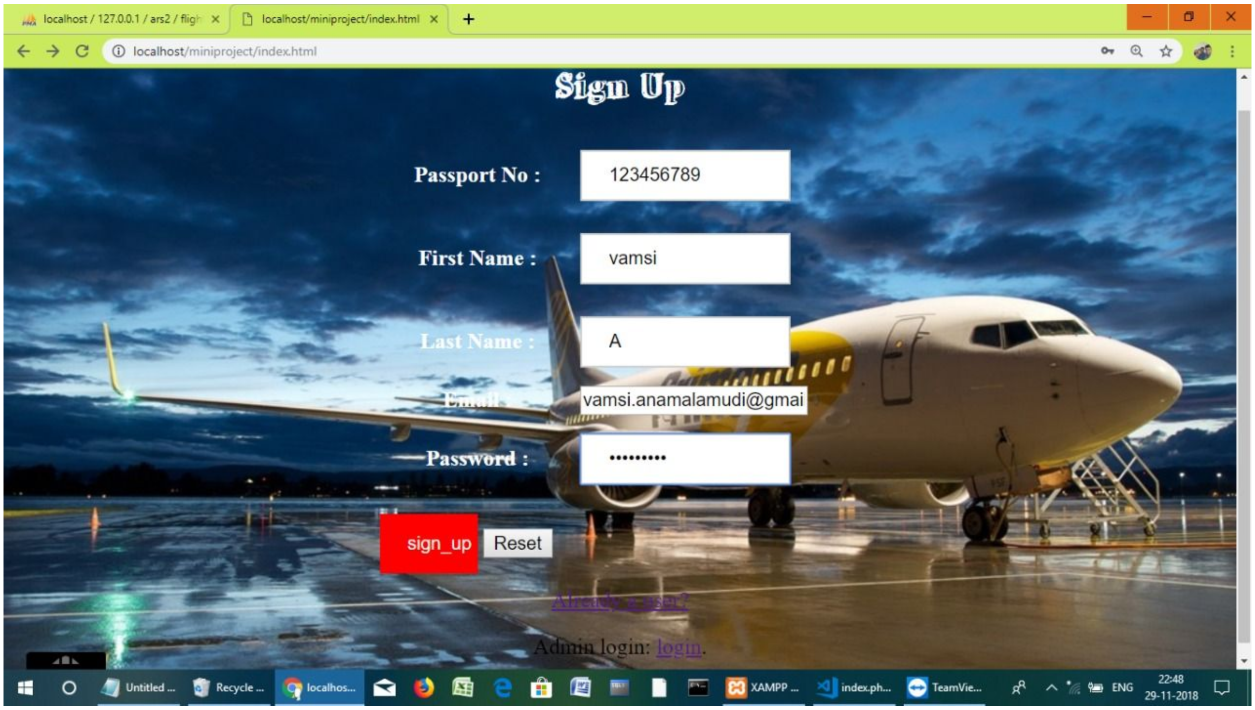
1.2 PROJECT OVERVIEW The Airline Reservation System project is an implementation of a general Airline Ticketing website like Orbitz, which helps the customers to search the availability and prices of various airline tickets, along with the different packages available with the reservations. This project also covers various features like online registration of the users, modifying the details of the website by the management staff or administrator of the website, 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.

2. OVERALL PRODUCT DESCRIPTION

2.1 PRODUCT PERSPECTIVE The Airline Reservation System project uses the .NET framework 1.1 and is completely independent. The project itself is a bigger product and does not need to be introduced into a larger system. The application would be running on a Windows XP/2000 Operating system

2.2 PRODUCT FEATURES: The Airline Reservation System has the following features: This project is mainly intended for two types of audiences. One is the customer or the end user and the other is the administrator of the website. Some of the major functions of the product can be categorized under two different categories that are for the administrator and the user.

2.2.1 Customer / End user activities Figure 1 - Customer use case diagram 16 The above use case diagram depicts all the functions or activities that a user or a customer can perform on the application. They can be discussed in detail as follows: Home Page: Like all the other airline websites available online, the user can access the user home page of the Airline Reservation System website, after he logs into the system.Here, he can look up information regarding flights, packages and motels. Login and Register: The Airline Reservation System also comes with the customer registration details page, where the customer can enter his details and register. He can also create a username and password. Moreover, he will also be able to modify the registration information in case of a change in his e-mail address or any other information. Booking Flights: The customer can also search for the flights available and reserve his place on the flight by purchasing a ticket. Book Motels: Different Airline websites like Orbitz also offer various deals for booking Motels along with the airline tickets. So, the customer will also be able to view this functionality on the website. Book Packages: This functionality is similar to the motel booking function, except the fact that the customer can look up various touring packages available at the person's destination. Contact the


localhost / 127.0.0.1 / ars2 / fligh
localhost/miniproject/index.html
localhost/miniproject/index.html
sign_up
Reset
Untitled ...
Recycle ...
localhos...
XAMPP ...
index.ph...
TeamVie...
ENG
22:48
29-11-2018

Sign Up

Passport No : 123456789

First Name : vamsi

Last Name : A

Email : vamsi.anamalamudi@gmai

Password : •••••••••

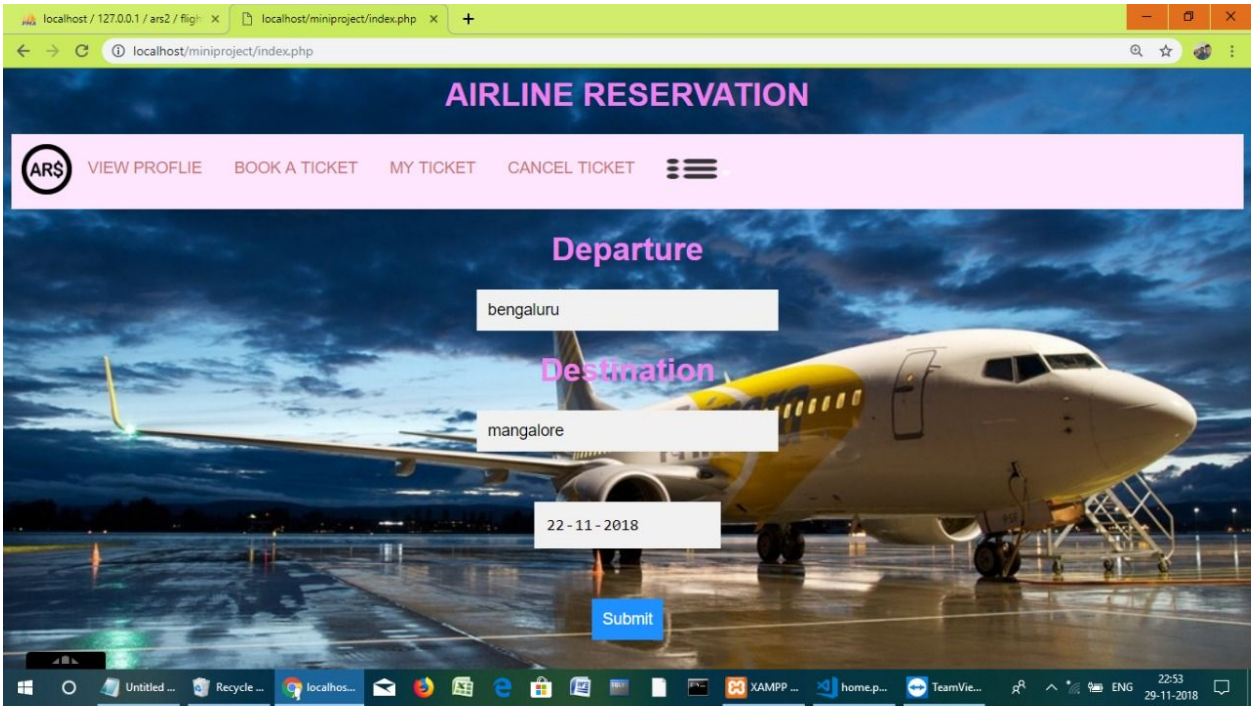
Already a user?

Admin login: login.

Company: The Customer can also call the company if he has any concerns or questions related to the bookings he has made online. Booking Instructions: The website also provides instructions to the customers on how to book airline tickets or motels along with the different packages. Some of the functions of the Airline Reservation System, such as creating, maintaining and updating the database are available only to the administrator. The functions of the administrator, explained in detail are as follows: 17

2.2.2 Administrator Activities

Login/Logout: The administrator has to login first in order to be able to make changes to the Airline Reservation System, by adding, deleting or modifying the data in the Airline Reservation System database. After making the necessary changes, he then has to logout of the system, in order to prevent misuse of the data.


localhost / 127.0.0.1 / ars2 / fligh
localhost/miniproject/index.php
localhost/miniproject/index.php
AIRLINE RESERVATION
AR$
VIEW PROFLIE
BOOK A TICKET
MY TICKET
CANCEL TICKET
Submit
22:53
29-11-2018

Departure

bengaluru

Destination

mangalore

22-11-2018

Add/Modify Customer Information: Daily the Airline Reservation System will have many customers registering with the website and many of them unsubscribing. Only the administrator will have the sole rights to modify the database accordingly.

Add/Modify Flight Information: The Administrator also has the sole rights to add, delete or modify the flight information. Sometimes, flights get cancelled for some reason, so such flights would be removed from the list of flights available to the customer. Similarly whenever any flight information has to be modified or if any new flights need to be added to the database, these operations are performed by the administrator.

Cancellation of Reservations: Sometimes, after making a reservation, a customer might cancel the reservation he has made. So, the administrator also handles such special situations and sends the customer an e-mail confirmation after deleting the specific transaction.

Email confirmations: Whenever a customer makes or cancels a reservation, the administrator is responsible for sending confirmation emails to the customer, confirming the transaction.

2.3 USER CHARACTERISTICS

There are two kinds of users for the Airline Reservation System. One is the customer and the other is the administrator. The customers do not need to have any prior training to use the application. However, instructions for making flight and motel reservations would be provided to them on the airline website. The administrators would however need to be trained in order to use the application.

2.4 CONSTRAINTS

In Case of changes made to the database, the application should be able to show the updated information on the website, without much delay. The database for the project is designed to be of moderate size. Currently, the application is designed to be able to run in Internet 19 Explorer. The Airline Reservation system will be designed in such a way that, it can be run on a Windows XP/2000 and IIS server. The .NET technology will be used to code the project and SQL server 2000 will act as the database for the project. The project will run on Internet Explorer and it should be installed on User's system.

2.5 ASSUMPTIONS AND DEPENDENCIES There are no assumptions as of now. To be updated in later versions of the vision document.

3. SPECIFIC REQUIREMENTS

3.1 EXTERNAL INTERFACES The different types of interfaces that we would come across while developing the Airline Reservation System application are as follows: • User Interface • Hardware Interface • Software Interface

3.1.1 USER INTERFACE There are two types of users for the Airline Reservation System project. One is the customer and the other is the administrator. Both the customer and administrator user interface would be a graphical user interface The Graphical User Interface would mainly consist of Hyperlinks, Data entry fields like the E-mail Id field, push down buttons like the Login button etc. The administrator of the website would also have a similar Graphical User Interface. After an administrator logs onto the system.

3.1.3 SOFTWARE INTERFACE The application should run on a Windows XP/2000 Operating System. Since the application needs a database to store all the customer details, airline, motel and package information, SQL server 2000 would be used. Visual Studio.NET 2003 would be used for creating the application. All the coding will be done in C#.

4. FUNCTIONAL REQUIREMENTS

The functional requirements of the Airline Reservation System are divided among the customer and the administrator of the application. These functional requirements can be explained in detail as follows: IMAGE Customer Details Flight details Hotel Details Package Details Cancellations 21

4.1 Use Case name: User Registration • Description: This use case describes the scenario where the user registers with the application by providing all the necessary details, in order to make reservations or bookings for flights, motels, special packages. • Actor: User or the Customer • Input: The user or the customer will have to provide all the necessary details present in the customer registration form of the application. • Output: All the details entered in the customer registration page will be verified and accepted by the system into the database.

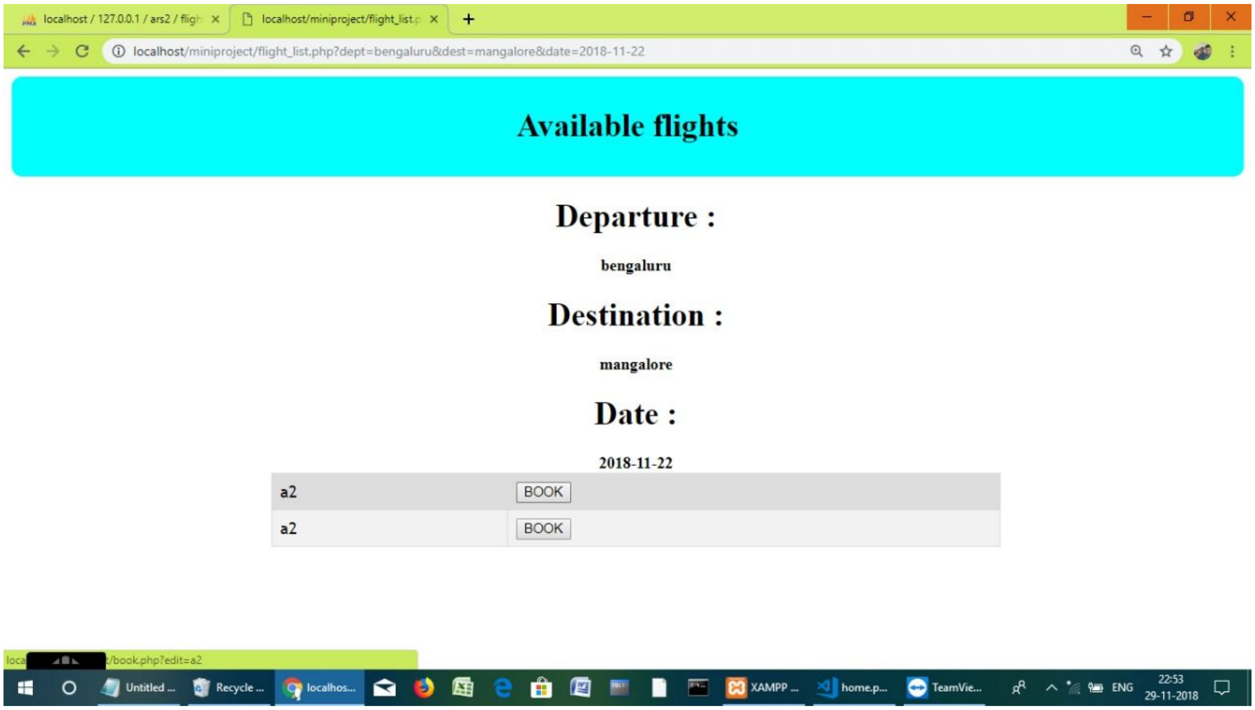
4.2 Use Case name: User Login • Description: This use case describes the scenario where the user logs into the application, with the username and password he has provided while registering with the system. • Actor: User or the Customer • Input: The user or the customer creates a username and password at the time of registering with the system. He then uses them to logon to the system and make reservations or view any information. • Output: The application then verifies the authenticity of the username and password that the customer has provided and allows the user to view the information available on the system, if the username and password are valid.

4.3 Use Case name: Contact the company • Description: This use case describes the scenario where the user contacts the company for any information. • Actor: User or the Customer • Input: The customer can contact the airline company, requesting them for any information he needs. • Output: The application verifies the authenticity of the username and password that the customer has provided and allows the user to view the contact information for the company. 22

4.4 User Case name: Booking Instructions • Description: This use case describes the scenario where the user views the instructions for booking flights, packages, or motels. • Actor: User or the Customer • Input: After the customer logs onto the application with his username and password, he can look up the instructions posted on the website for booking flights, packages or motels. • Output: The application verifies the authenticity of the username and password and displays the how to book instructions page.

4.5 Use Case name: Book Flights • Description: This use case describes the scenario where the user books airline tickets. • Actor: User or the Customer • Input: After logging into the application, the customer looks up the information related to various airlines and checks the availability of seats on flights. If he finds that there are any available tickets, he then purchases them. • Output: The application verifies the authenticity of the username and password and then displays information related to various flights to the customer.

4.6 Use Case name: Booking Packages • Description: This use case describes the scenario where the user books different touring packages at the airline ticket reservation • Actor: Customer or the user • Input: The customer looks up information regarding various touring packages available at his destination at the time of airline ticket reservation. • Output: The application verifies the authenticity of the username and password of the customer and then displays information of various touring packages available at customer's choice of place. The administrator activities use cases will be described here:



localhost/miniproject/flight_list.php?dept=bengaluru&dest=mangalore&date=2018-11-22

Available flights

Departure :

bengaluru

Destination :

mangalore

Date :

2018-11-22

a2	BOOK
a2	BOOK

loca[redacted]/book.php?edit=a2

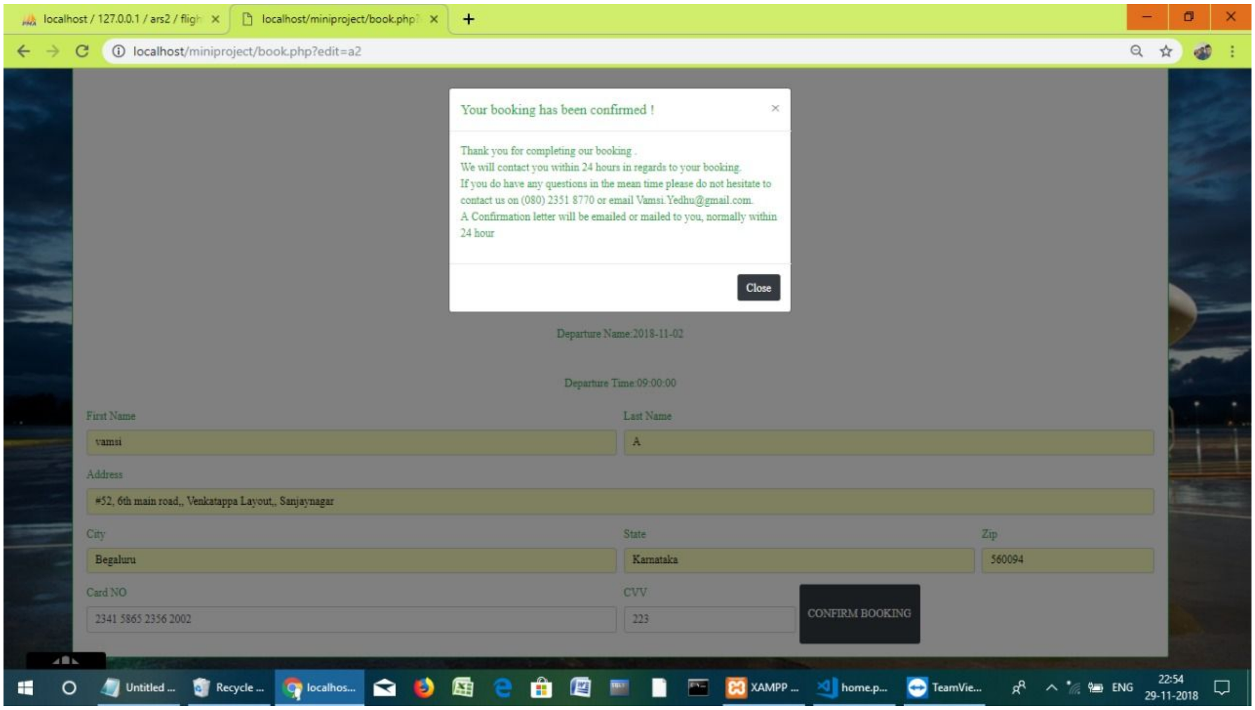
4.7 Use Case name: Login/Logout • Description: This use case describes the scenario where the administrator of the application, logs into the system and logs out after the work is done. • Actor: Administrator • Input: The administrator of the website logs into the application with the username and password provided to him. • Output: The application verifies the authenticity and displays the home page of the administrator.

4.8 Use Case name: Add/Delete/Modify Customer information • Description: This use case describes the scenario where the administrator adds, deletes or modifies customer information in the system database • Actor: Administrator • Input: The administrator of the applications logs onto the system with his username and password. • Output: The application authenticates the administrator, and then displays the page where the administrator can add new customers to the database, or delete existing customers or modify details of customers in the database. 24

4.9 Use Case name: Add/Delete/ Modify flight information • Description: This use case describes the scenario where the administrator adds, deletes or modifies flight information in the application database • Actor: Administrator • Input: The administrator logs onto the system with the username and password provided to him. • Output: The application authenticates the administrator, by verifying the username and password. Then the application displays the page where the administrator can add new flights to the database, delete the flights that have been cancelled or modify information for the flights.

4.10 Use Case name: Cancellation of Reservations • Description: This use case describes the scenario where the administrator handles the cancellation of reservations by the customers. • Actor: Administrator • Input: The administrator logs onto the system with the given username and password. • Output: The application authenticates the administrator and then displays the page where the administrator looks up the id of the customer who has requested cancellation of reservation. After canceling the reservation, the administrator then sends a confirmation e-mail to the customer.

4.11 Use Case name: Email confirmations • Description: This use case describes the scenario where the administrator sends email confirmations to the customers of the application. • Actor: Administrator • Input: The administrator logs onto the application with the username and password provided. • Output: The application then authenticates the administrator and displays the page where the administrator can send email confirmations to the customer. These e-mail 25 confirmations may be sent in cases where the customer has cancelled a reservation or changed the personal information available on the website.



localhost / 127.0.0.1 / ars2 / fligh
localhost/miniproject/book.php?
localhost/miniproject/book.php?edit=a2
Close
CONFIRM BOOKING
Untitled ...
Recycle ...
localhos...
XAMPP ...
home.p...
TeamVie...
ENG
22:54
29-11-2018

Your booking has been confirmed !

Thank you for completing our booking .
We will contact you within 24 hours in regards to your booking.
If you do have any questions in the mean time please do not hesitate to contact us on (080) 2351 8770 or email Vamsi.Yedhu@gmail.com.
A Confirmation letter will be emailed or mailed to you, normally within 24 hour

Departure Name:2018-11-02

Departure Time:09:00:00

First Name: vamsi
Last Name: A
Address: #52, 6th main road,, Venkatappa Layout,, Sanjaynagar
City: Begaluru
State: Karnataka
Zip: 560094
Card NO: 2341 5865 2356 2002
CVV: 223

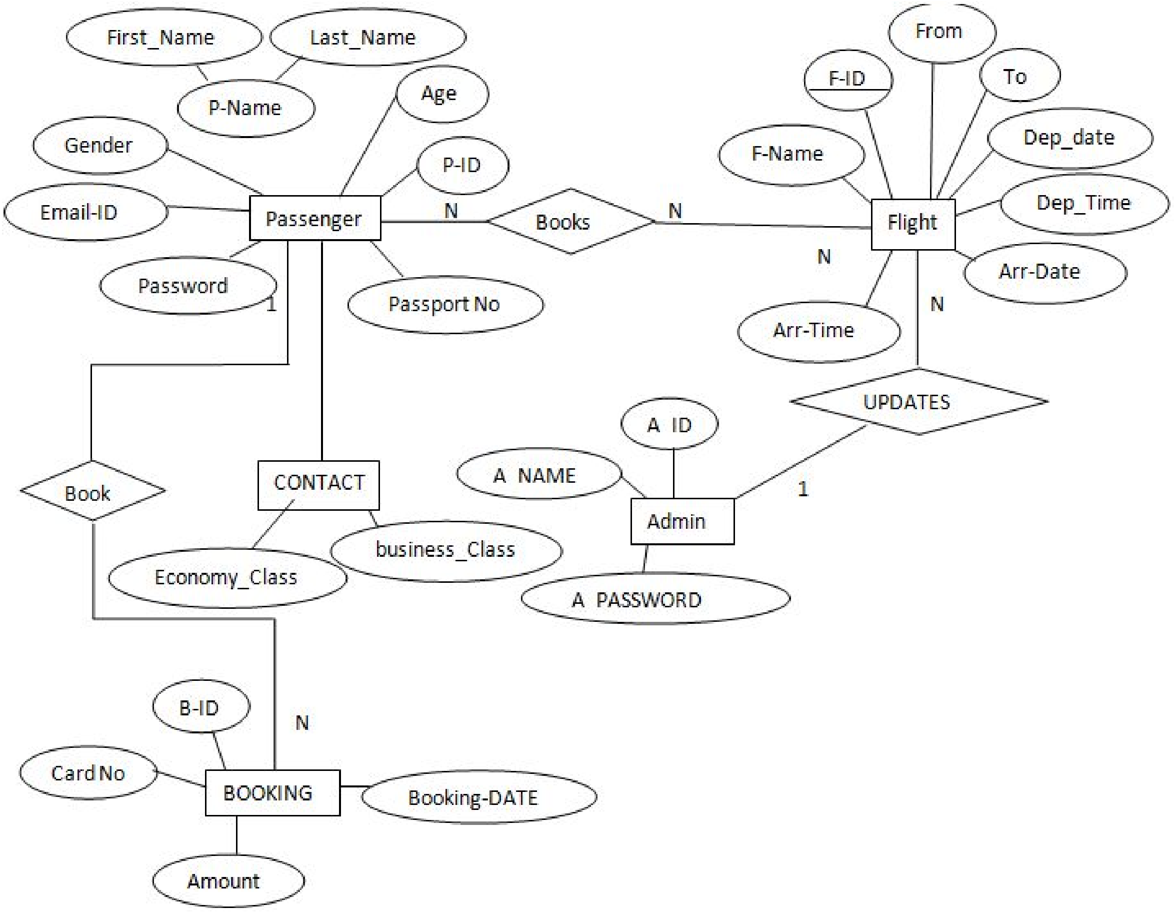
4.12 Use Case name: Modifying details of webpage • Description: This use case describes the scenario where the administrator logs onto the application to modify the details of the airline website • Actor: Administrator • Input: The administrator logs onto the application with the username and password provided to him • Output: After verifying the username and password of the administrator, the application then allows the administrator to login. The administrators can then browse through the website and change the details of any webpage in the Airline Reservation system application.

4.13 Use Case name: Add/Delete or Modify package information • Description: This use case describes the scenario where the administrator adds, deletes or modifies package information in the application database • Actor: Administrator 26 • Input: The administrator logs onto the system with the username and password provided to him. • Output: The application authenticates the administrator, by verifying the username and password. Then the application displays the page where the administrator can add new packages to the database, delete the packages that are no longer available or modify information for any particular package.

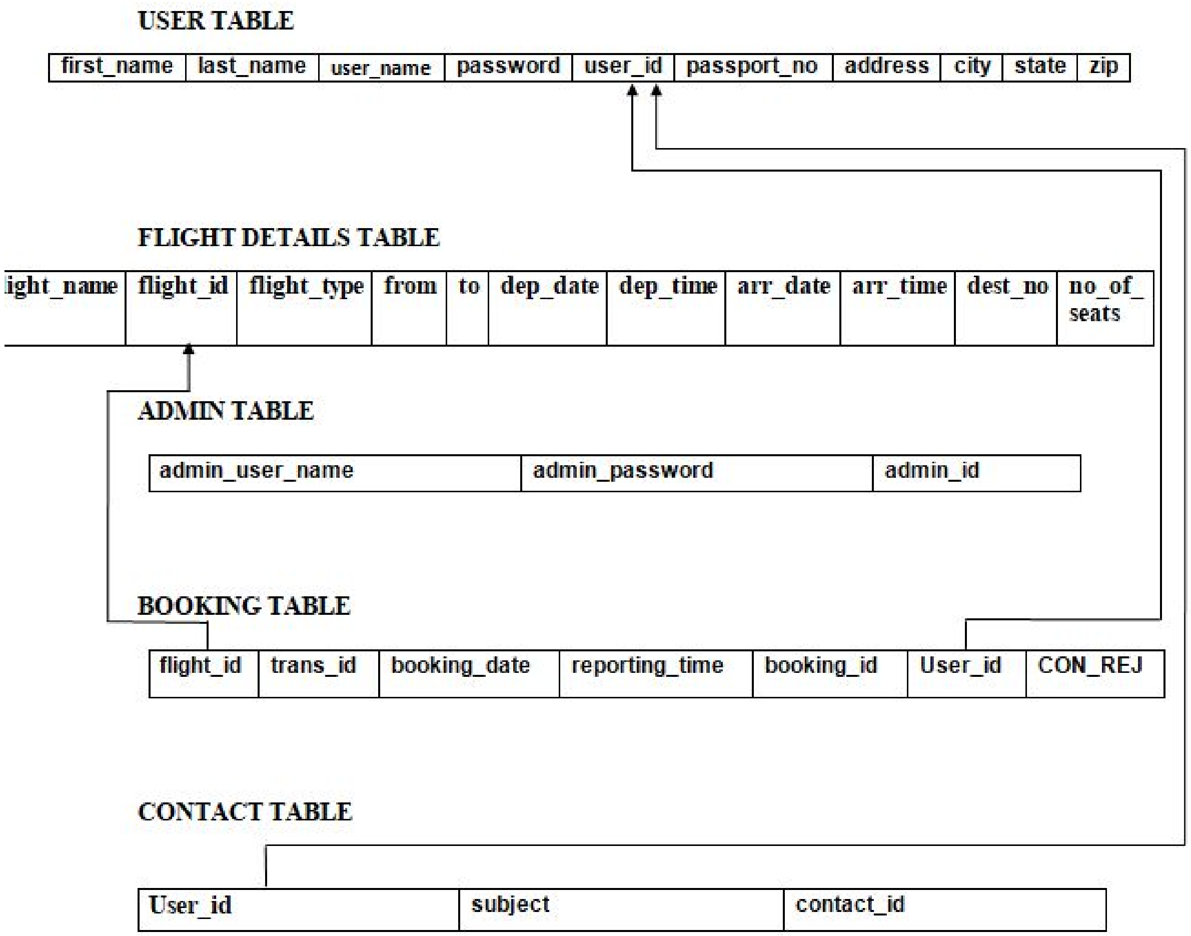
5. PERFORMANCE REQUIREMENTS

The Airline Reservation System application should be able to respond to the queries submitted by the customer without much delay. When a user searches for a flight leaving from a particular place to another place, the application should not take much time to return the results, similarly for the motel and package information. Considering that the application is of moderate size, it should be able to display 10 results at a time on each page, when the customer looks up for any particular data. Since the Airline Reservation websites have much traffic, the user should also be able to logon to the system using high speed internet. Most of the requests sent to the application should be answered in less than 5 seconds.

6. ENTITY RELATION DIAGRAM (E-R DIAGRAM)



7. SCHEMA DIAGRAM



USER TABLE

first_name	last_name	user_name	password	user_id	passport_no	address	city	state	zip

FLIGHT DETAILS TABLE

ight_name	flight_id	flight_type	from	to	dep_date	dep_time	arr_date	arr_time	dest_no	no_of_ seats

ADMIN TABLE

admin_user_name	admin_password	admin_id

BOOKING TABLE

flight_id	trans_id	booking_date	reporting_time	booking_id	User_id	CON_REJ

CONTACT TABLE

User_id	subject	contact_id

8. TOOLS, TECHNIQUES AND METHODOLOGIES

The following are the tools that will be used for coding, testing and documentation:

- Microsoft Visual Studio .NET 2003 – for coding
- HTML – for coding
- MS WORD 2003 – for documentation
- User Testing

9. APPENDIX

9.1 **Appendix-I**

9.1.1 **Bibliography**

https://en.wikipedia.org/wiki/Stored_procedure

https://www.w3schools.com/

https://www.tutorialspoint.com/

https://stackoverflow.com/

https://www.wikipedia.org/

https://www.youtube.com/

http://www.inmotionhosting.com/support/edu/website-design/using-php-and-mysql/grab-all-comments-from-database

https://www.edureka.co

https://www.udemy.com

https://www.google.co.in/

9.2 **Appendix-II**:

9.2.1 **Development Tools:**

1. MYSQL Workbench:

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OS X

MySQL Workbench simplifies database design and maintenance, automates time-consuming and error-prone tasks, and improves communication among DBA and developer teams. It enables data architects to visualize requirements, communicate with stakeholders, and resolve design issues before a major investment of time and resources is made. It enables model-driven database design, which is the most efficient methodology for creating valid and well-performing databases, while providing the flexibility to respond to evolving business requirements. Model and Schema Validation utilities enforce best practice standards for data modeling, also enforce MySQL-specific physical design standards so no mistakes are made when building new ER diagrams or generating physical MySQL databases..

MySQL Workbench delivers visual tools for creating, executing, and optimizing SQL queries.

2. Sublime Text 3:

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

Version 3 entered beta on 29 January 2013. At first available only for registered users who have purchased Sublime Text 2, on 28 June 2013 it became available to the general public. However, the very latest development builds still required a registration code.[12] Sublime Text 3 was officially released on 13 September 2017.[13]

Two of the main features that Sublime Text 3 adds include symbol indexing and pane management. Symbol Indexing allows Sublime Text to scan files and build an index to facilitate the features Goto Definition and Goto Symbol in Project. Pane Management allows users to move between panes via hotkeys.[14]

9.2.2 **Software Environment**

1. Html

2. Php

3. Mysql

4. Java Script

5. Css

9.2.3 Hardware Environment:

System

Manufacturer DELL,,Mac

Model DELL Pavilion g6 Notebook PC

Total amount of system memory 4.00 GB RAM

System type 64-bit Operating System

Processor Intel® Core i5-3230 CPU 4 x 2.60

GHz

Storage

Total size of hard disk 500 GB

Disk partition (C:) 40.8 GB Free (371 GB Total)

Disk partition (D:) 2.10 GB Free (16.7 GB Total)

Disk partition (H:) 5 GB Free (34 GB Total)