

A Project Report On
AIRLINE RESERVATION SYSTEM

Submitted in partial fulfillment of the
MASTER OF COMPUTER APPLICATION

By

Maheswar Saren

Roll no.11

Enrollment No. AJU/210785

Under the esteemed guidance of
Dr. Arvind Kumar Pandey
&

Ms. Alka Kumari

(Internal Guide)



DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

ARKA JAIN UNIVERSITY, JHARKHAND

JAMSHEDPUR

2021-2023



ARKA JAIN UNIVERSITY

GUIDED BY:

Dr. Arvind Kumar Pandey
&
Ms. Alka Singh

PREPARED BY:

Maheswar Saren

SUBMITTED TO

DEPARTMENT OF COMPUTER SCIENCE & IT
ARKA JAIN UNIVERSITY

A Project Report On
AIRLINE RESERVATION SYSTEM

Submitted in partial fulfillment of the
MASTER OF COMPUTER APPLICATION

By
Maheswar Saren
Roll no.11
Enrollment No. AJU/210785

Under the esteemed guidance of
Dr. Arvind Kumar Pandey
&
Ms. Alka Kumari
(Internal Guide)



DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
ARKA JAIN UNIVERSITY, JHARKHAND
JAMSHEDPUR
2021-2023

**ARKA JAIN UNIVERSITY, JHARKHAND
JAMSHEDPUR**



CERTIFICATE

This is to certify that the project entitled, "AIRLINE RESERVATION SYSTEM", is bonafide work of MAHESAWR SAREN bearing Enrollment No. AJU/210785 submitted in partial fulfilment of the requirements for the award of degree of MASTER OF COMPUTER APPLICATION (MCA) from ARKA JAIN UNIVERSITY, JHARKHAND.

Internal Guide

HOD

Date: 23-7-22



University Seal

ACKNOWLEDGEMENT

It is a genuine pleasure to express my profound gratitude and deep regards to my Internal Guide **Ms. Alka Kumari** and our HOD **Dr. Arvind Kumar Pandey** for their exemplary guidance, monitoring and constant encouragement. I would like to express my special thanks to **ARKA JAIN UNIVERSITY** who gave me the golden opportunity to do this wonderful project on the topic **Airline Reservation System**, which helped me in doing a lot of Research and I came to know about so many new things.

With Regards

Maheswar Saren (AJU/210785)

Roll no. 11 (MCA)

DECLARATION

We hereby declare that the work in the project report entitled “**AIRLINE RESERVATION SYSTEM**” done at **Arka Jain University**, Jamshedpur for the award of degree of “**MASTER OF COMPUTER APPLICATIONS**” is an authentic record of my work carried out during the second semester under the supervision of **Ms. Alka Kumari**. The matter embodied in this mini-project report has not been submitted elsewhere by anybody for the award of any other degree/diploma.

Maheswar Saren

Name and Signature of the
Student

Table of Contents

Particulars	Page Nos.
Contents	
1.1. Project Overview	11
1.2. Project Description	11
1.3. Definitions, Acronyms, and Abbreviations	12
1.4. References	13
Problem Definition.....	13
2.1 Existing System	13
2.2. Proposed System:	14
2.3. Product Functions	15
2.4. User Characteristics	15
2.5. Constraints	15
2.6. System Study.....	16
2.7. Assumptions and Dependencies.....	16
Feasibility Study	17
3.1. Technical Feasibility	17
3.2. Operational Feasibility	18
3.3. Economic Feasibility	18
3.4. Interfaces	19
3.5. Hardware Requirements:	19
3.6. Software Requirements:	19
Apache	20
XAMPP	20
System Analysis.....	20
4.1. Software Requirement Specification (SRS).....	20
4.2. Communication Interfaces	21
4.3. Memory Constraints	21
4.4. Operations	22
4.6. Functional Requirements.....	22
DATA FLOW DIAGRAM.....	24

NORMALISATION	29
Normalisation Level 0	29
Normalisation level 1	29
Normalisation Level 2	30
5.1. MODULE DESCRIPTION	32
5.2. Software System Attributes	32
5.3. Organizing the specific Requirements	34
Passenger Table	36
Configuration.php	47
Home.php	47
Register.php	50
About Us	59
Flight Booking	61
Passenger Details	70
Admin Panel	77
9.1. Testing Methods	83
CONCLUSION.....	86
BENEFITS:	87
Assumptions, if any	88

INTRODUCTION

1.1. Project Overview

Airline Reservation System contains the details about flight schedules and its fare tariffs, passenger reservations and ticket records. Air Alliance operates flights to 30 destinations in India namely Allahabad, Bengaluru, Bhopal, Bhuj, Dehradun, Delhi, Diu, Gorakhpur, Guwahati, Hyderabad, Jabalpur, Jaipur, Jammu, Kanpur, Kochi, Kolkata, Kullu, Mumbai, Pant Nagar, Pune, Raipur, Ranchi, Shimla, Surat, Vijayawada, Tezpur and Tirupati.

1.2. Project Description

Airline Reservation System will hold flight schedules and its fare tariffs, passenger reservations and ticket records. It saves time as it allows online procedure as users no longer to wait in a queue to book the flights. It is automatically generated by the server. Admin is the main authority who can do addition, deletion, and modification of flights if required.

The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of Database Mysql and all the user interfaces have been designed using the Adobe Dreamweaver technologies.

The database connectivity is planned using the "SQL Connection" methodology. The standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports, which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The specification has been normalized up to 2NF to eliminate all the anomalies that may arise due to the database transaction that are executed by the general users and the organizational administration. The user interfaces are browser specific

to give distributed accessibility for the overall system. The internal database has been selected as Database Mysql.

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.

1.3. Definitions, Acronyms, and Abbreviations

Personal Details: Details of passengers such as user id, phone number, address, passport no, e-mail address etc.

Contact Details: Details of contact associated with the passenger.

SRS: System Requirement Specification

WWW: World Wide Web

MySQL: is a RDBMS based on SQL which is used for adding, removing, and modifying information in the database.

RDBMS: Relational Database Management System

HTML: Hypertext Markup Language

PHP: Hypertext Preprocessor

CSS: Cascading Style Sheet

HTTP: Hypertext Transfer Protocol

1.4. References

- Software Requirement Specifications from Internet.
- Software Requirement Specifications, Airline Reservation System.
- IEEE (Institute of Electrical and Electronics Engineers) Guide to Software Requirements Specifications.
- www.slideshare.com
- www.google.com
- www.wikipedia.com
- www.project-airline-system-basics.com
- HTML for the World Wide Web with XHTML and CSS Guide, Fifth Edition
 - IEEE Computer Society, 1998.

Problem Definition

2.1 Existing System

In few countries if a person wants to book a flight ticket, he use to follow one of these things:

Disadvantages

- Manually goes to the Airport and book his ticket.
- Downloading the ticket form as paper document, filling it manually and submitting it at Airport.
- Fill the Ticket form on system and get the print out as paper documents to submit it at Airport.
- Booking the Ticket at some particular registered ticket counters in online.
- Even above approaches make a ticket booking online, it was not completely done on online. Passenger may not have much freedom over this approach.

- Hence the Passenger may or may not be satisfied with this approach as it includes manual intervention like travelling to Airport for booking his ticket.
- Cannot Upload and Download the latest updates.
- No use of Web Services and Remoting.
- Risk of mismanagement and of data when the project is under development.
- Less Security.
- No proper coordination between different Applications and Users.
- Fewer Users – Friendly

2.2. Proposed System:

The Proposed system ensures the complete freedom for users, where user at his own system can logon to this website and can book his ticket. Our proposed system allows only registered users to book the tickets, view timings and cancel their tickets.

In this Proposal the entire work is done on online and ticket with id is also provided for passengers as a print document. Here passengers can send their queries and suggestions through a feedback form.

To debug the existing system, remove procedures those cause data redundancy, make navigational sequence proper to build strong password mechanism.

Advantages:

- User friendliness provided in the application with various controls.
- The system makes the overall project management much easier and flexible.
- It provides high level of security with different level of authentication.

2.3. Product Functions

The website will allow access only to authorized users with specific roles (Administrator- maintains the website, Company-Register the passengers, Passenger- Fills the details).

Following are the System Functions:

Passenger role:

On the register form, passenger should enter all their detail such as their name, passport number, Email and contact number.

Administration role

The system administrator must be able to: add, update and modify flights and view the customer details.

2.4. User Characteristics

End Users

All specific knowledge or skills are required from the feeder.

- Educational level: Users should be comfortable with the English language.
- Experience: Users should have prior information regarding the online booking.
- Skills: Users should have basic knowledge and should be comfortable

Administrator

Administrator must be capable to manage user rights.

This system will not take care of any virus problem, which might occur either on the Client or the server system. Avoiding the use of pirated software and ensuring that floppies and other removable media are scanned for viruses before use could minimize the possibility of viral infection.

2.5. Constraints

The Information of all users, subjects and allocations must be stored in a database that is accessible by every connected system. MySQL used for database.

- Users may access from any system connected to the online database.
- Users must have their correct usernames and passwords to enter into their accounts.

2.6. System Study

System Study is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is- what all problems exist in the present system? What must be done to solve the problem? Analysis begins when a user or manager begins a study of the program using existing system.

System study can be categorized into four parts.

- System planning and initial investigation
- Proposed System with objectives

2.7. Assumptions and Dependencies

The Software needs the following third-party products-

- Adobe Dreamweaver for development of project.
- WAMP for database connectivity.

Although basic password authentication and role-based security mechanisms will be used to protect OPMS from unauthorized access; functionality such as email notifications are assumed to be sufficiently protected under the existing security policies applied by the University network team. Redundant Database is setup as the role of backup Database Server when primary database is failure.

Feasibility Study

Preliminary investigation examines project feasibility. The likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- ➤ Economic Feasibility

3.1. Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Does the proposed equipment have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides an easy access to the users.

The database's purpose is to create, establish and maintain a workflow among various entities to facilitate all concerned users in their various capacities or roles. Permission to the

users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

The software and hardware requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing fast feedback to the users irrespective of the number of users using the

3.2. Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system.

That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues.

Beforehand, the management issues and user requirements have been taken into consideration. So, there is no question of resistance from the users that can undermine the possible application benefits.

3.3. Economic Feasibility

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

3.4. Interfaces

In computing, an interface is a shared boundary across which three separate components of computer system exchange information.

User interfaces

The application will have a user friendly and menu-based interface.

3.5. Hardware Requirements:

- Intel I3 2.8 GHz Processor and Above
- RAM 1 GB and Above
- HDD 20 GB Hard Disk Space and Above

3.6. Software Requirements:

- WINDOWS OS (Windows 7, 8 ,10) Or Linux
- Adobe Dreamweaver
- PHP Wamp Server for Windows (Apache)
- Database Mysql for Backend.

Server side An Apache Web server will accept all requests from the client. A development database will be hosted locally (using MySQL); the production database is hosted centrally.

MY-SQL (BACKEND)

MySQL in July 2013, it was the world's second most widely used RDBMS, and the most widely used open-source client server model RDBMS. It is named after cofounder Michael Widenius's. The SQL abbreviation stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements.

Apache

The Apache HTTP Server is web server software notable for playing a key role in the initial growth of the World Wide Web. In 2009 it became the first web server software to surpass the 100 million web site milestone. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

Since April 1996 Apache has been the most popular HTTP server software in use.

XAMPP

XAMPP is a small and light Apache distribution containing the most common web development technologies in a single package. Its contents, small size, and portability make it the ideal tool for students developing and testing applications in PHP and

MySQL. XAMPP is available as a free download in two specific packages: full and lite.

While the full package download provides a wide array of development tools, XAMPP Lite contains the necessary technologies that meet the Ontario Skills Competition standard.

System Analysis

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is- what all problems exist in the present system? What must be done to solve the problem? Analysis begins when a user or manager begins a study of the program using existing system.

4.1. Software Requirement Specification (SRS)

The software, Site Explorer is designed for management of web sites from a remote location.

This section provides software requirements to a level of detail sufficient to enable designers to design the system and testers to test the system.

This section contains all of the functional and quality requirements of the system.

It gives a detailed description of the system and all its features.

Introduction

Purpose: The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system.

Scope: This Document plays a vital role in the development life cycle (SDLC) and it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

Developer's responsibilities overview:

The developer is responsible for:

- Developing the system, which meets the SRS and solving all the requirements of the system?
- Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
- Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
- Conducting any user training that might be needed for using the system.
- Maintaining the system for a period of one year after installation.

4.2. Communication Interfaces

The HTTP protocol will be used to facilitate communications between the client and server. The system supports Google Chrome and Mozilla Firefox web browsers.

4.3. Memory Constraints

Minimum memory of 512MB is required to run the exe file without any lags. This constraint does not possess an issue now a days as the minimum present RAM in a common system is 1GB.

At least 512 MB RAM and 5 MB space on hard disk will be required for running the program.

4.4. Operations

The normal and special operations required by the user such as:

- The various modes of operations in the user organization
- Periods of interactive operations and periods of unattended operations
- Data processing support functions
- Backup and recovery operations

4.5. Site Adaptation Requirements

There should no site adaptation requirement since the Web Application Server was setup.

4.6. Functional Requirements

- It deals with the functionalities required from the system which are as follows:
- The website will help the colleges/organizations/companies to conduct their student registration
- Only authorized person can access related details.
- Organizations can change their information regarding themselves. The students can login through TEST-ID and PASSWORD.
- Administrator will be responsible for updating the site.

4.6.1. Performance Requirements

This subsection specifies numerical requirements placed on the software or on the human interaction with the software, as a whole..Numerical requirements will include:

- 300 terminals will be supported at a time
- Only text information will be supported (HTTP)

4.6.2. Assumptions and Dependencies

The Software needs the following third-party products-

- Adobe Dreamweaver for development of project.
- WAMP for database connectivity.

Although basic password authentication and role-based security mechanisms will be used to protect OPMS from unauthorized access; functionality such as email notifications are assumed to be sufficiently protected under the existing security policies applied by the University network team. Redundant Database is setup as the role of backup Database Server when primary database is failure.

System Design

The purpose of System Design is to create a technical solution. That satisfies the functional requirements for the system. At this Point in the project lifecycle there should be a Functional Specification, written primarily in business terminology, containing a complete description of the operational needs of the various organizational entities that will use the new system.

The Challenge is to translate all of this information into Technical Specifications that accurately describe the design of the system, and that can be used as input to System Construction. The Functional Specification produced during System Requirements Analysis is transformed into a physical architecture. System components are distributed across the physical architecture, usable interfaces are designed and prototyped, and

Technical Specifications are created for the Application Developers, enabling them to build and test the system. Many organizations look at System Design primarily as the Preparation of the system component specifications; however, Constructing the various system components is only one of a setoff major step in successfully building a system.

The preparation of the environment needed to build the system, the testing of the system, and the migration and preparation of the data that will ultimately be used by the system are equally important.

In addition to designing the technical solution, System Design is the time to initiate focused planning efforts for both the testing and data preparation activities.

DATA FLOW DIAGRAM

Four simple notations are used to complete a DFD. These notations are given below: -

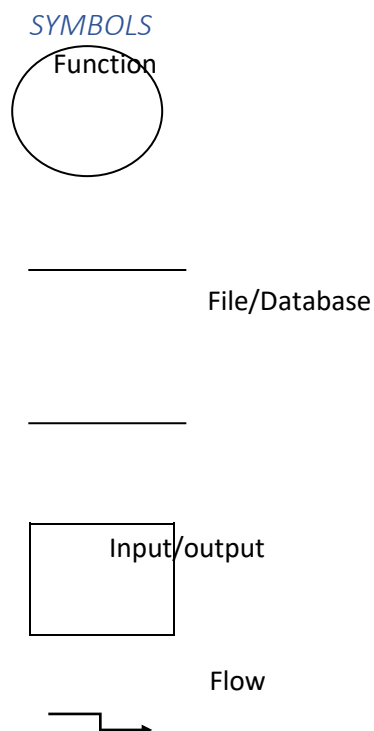
DATA FLOW: - The data flow is used to describe the movement of information from one part of the system to another part. Data flow is represented by an arrow.

PROCESS: - A circle or bubble represents a process that transforms incoming data to outgoing data. Process shows a part of the system that transform inputs to outputs.

EXTERNAL ENTITY: - External entities represent any entity that supplies or receive information from the system but is not a part of the system.

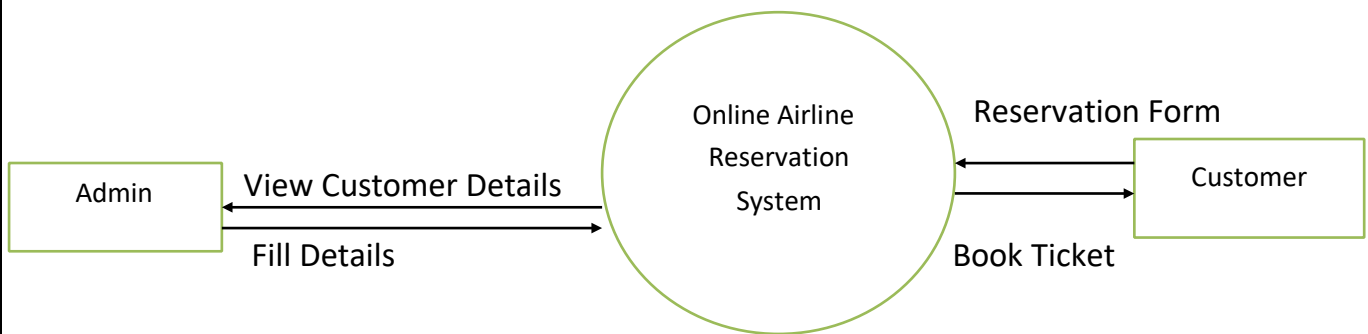
DATA STORE: - The data store represents a logical file. A logical file can represent either a data store symbol which can represent either a data structure or a physical file on disk. The data store is used to collect data at rest or a temporary repository of data. It is represented by open rectangle.

OUTPUT: -The output symbol is used when a hard copy is produced and the user of the copies cannot be clearly specified or there are several users of the output.

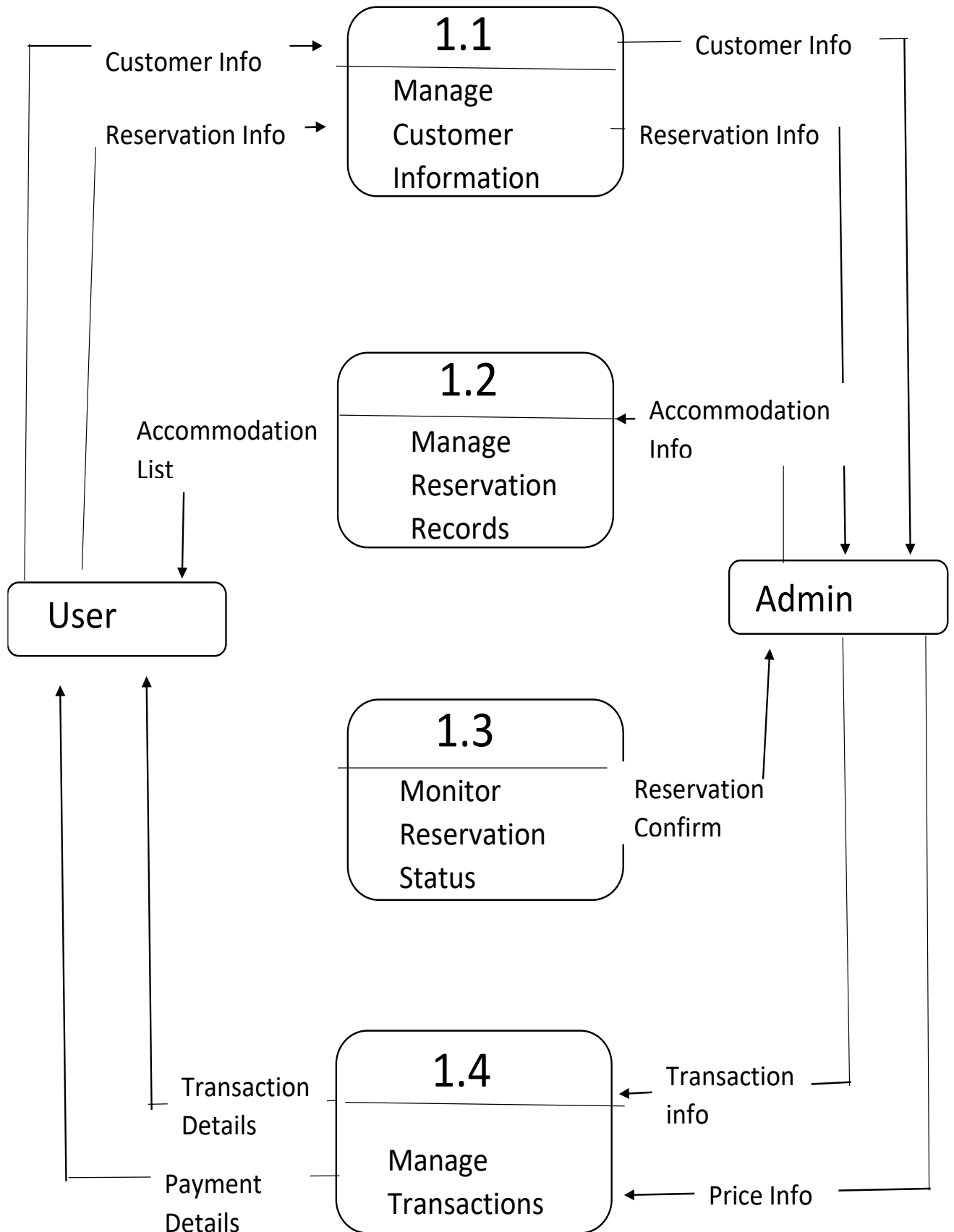


Context Diagram

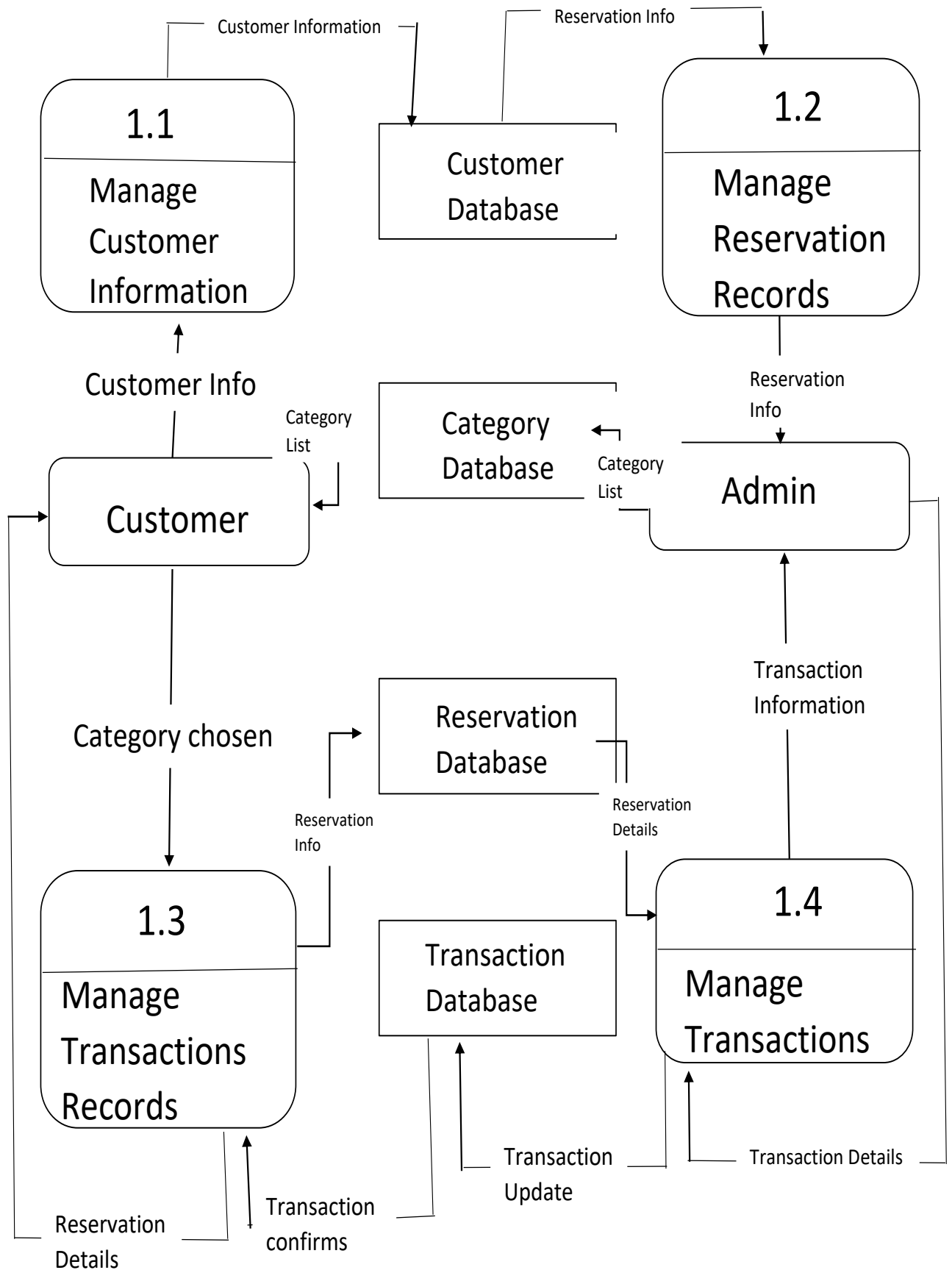
LEVEL-0/Context Diagram :-



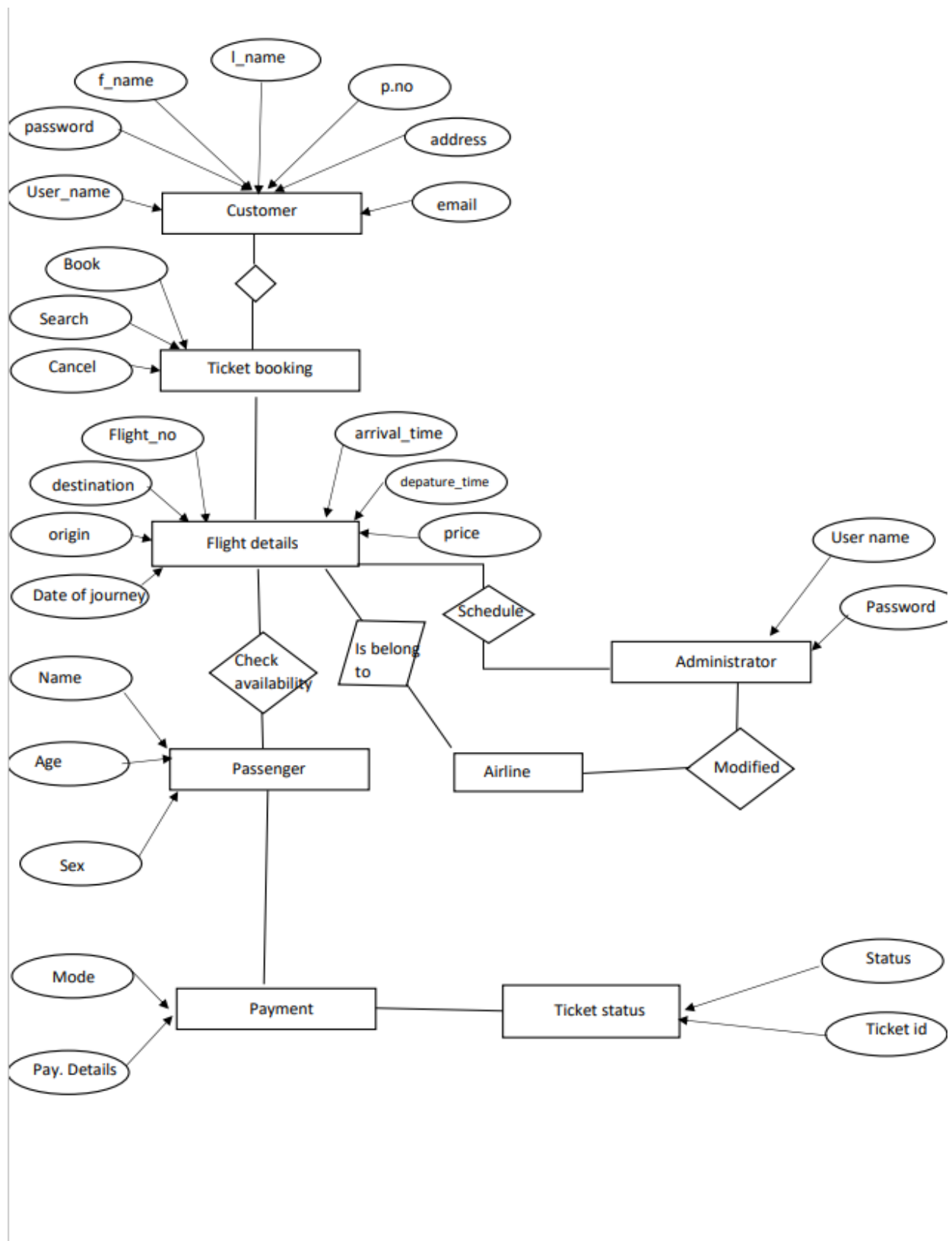
DFD LEVEL -1



DFD LEVEL – 2



ER DIAGRAM



NORMALISATION

Normalisation Level 0

[id]	[int]	NOT NULL
[name]	[varchar](500)	NULL
[phone]	[varchar](500)	NULL
[uid]	[varchar](500)	NULL
[image]	[varchar](500)	NULL
[gender]	[varchar](500)	NULL
[dob]	[varchar](500)	NULL
[contact]	[varchar](500)	NULL
[login]	[varchar](500)	NULL
[book]	[varchar](500)	NULL
[schedule]	[varchar](500)	NULL
[email]	[varchar](500)	NULL
[paymnt]	[varchar](500)	NULL
[username]	[varchar](500)	NULL
[fname]	[varchar](500)	NULL
[lname]	[varchar](500)	NULL
[departure]	[varchar](500)	NULL
[arrival]	[varchar](500)	NULL
[origin]	[varchar](500)	NULL
[destination]	[varchar](500)	NULL

[price]	[varchar](500)	NULL
[ticket id]	[varchar](500)	NULL
[status]	[varchar](500)	NULL
[payment]	[varchar](500)	NULL
[mode]	[varchar](500)	NULL
[pay_detail]	[varchar](500)	NULL
[country]	[varchar](500)	NULL
[contact]	[varchar](500)	NULL

Normalisation level 1

Contact

[id]	[varchar](500)	NOT NULL
[name]	[varchar](500)	NULL
[uid]	[varchar](500)	NULL
[gender]	[varchar](500)	NULL
[age]	[varchar](500)	NULL
[contact]	[varchar](500)	NULL
[mail]	[varchar](500)	NULL
[dob]	[varchar](500)	NULL

flight details

[flight no]	[varchar](500)	NULL
[origin]	[varchar](500)	NULL
[destination]	[varchar](500)	NULL
[arrival time]	[varchar](500)	NULL
[departure time]	[varchar](500)	NULL
[date]	[varchar](500)	NULL
[price]	[varchar](500)	NULL
[schedule]	[varchar](500)	NULL

Adminstrator

[username]	[varchar](500)	NULL
[password]	[varchar](500)	NULL
[modify]	[varchar](500)	NULL
[schdule]	[varchar](500)	NULL
[airline]	[varchar](500)	NULL
[name]	[varchar](500)	NULL
[age]	[varchar](500)	NULL
[sex]	[varchar](500)	NULL

Payment

[username]	[varchar](500)	NULL
[password]	[varchar](500)	NULL
[id]	[varchar](500)	NULL
[mode]	[varchar](500)	NULL
[pay details]	[varchar](500)	NULL
[status]	[varchar](500)	NULL
[conform]	[varchar](500)	NULL
[ticket id]	[varchar](500)	NULL

Normalisation Level 2

Customer

[username]	[varchar](500)	NULL
[password]	[varchar](500)	NULL
[fname]	[varchar](500)	NULL
[lname]	[varchar](500)	NULL
[phone]	[varchar](500)	NULL
[address]	[varchar](500)	NULL
[email]	[varchar](500)	NULL

Login

[username]	[varchar](500)	NULL
[password]	[varchar](500)	NULL
[id]	[varchar](500)	NULL

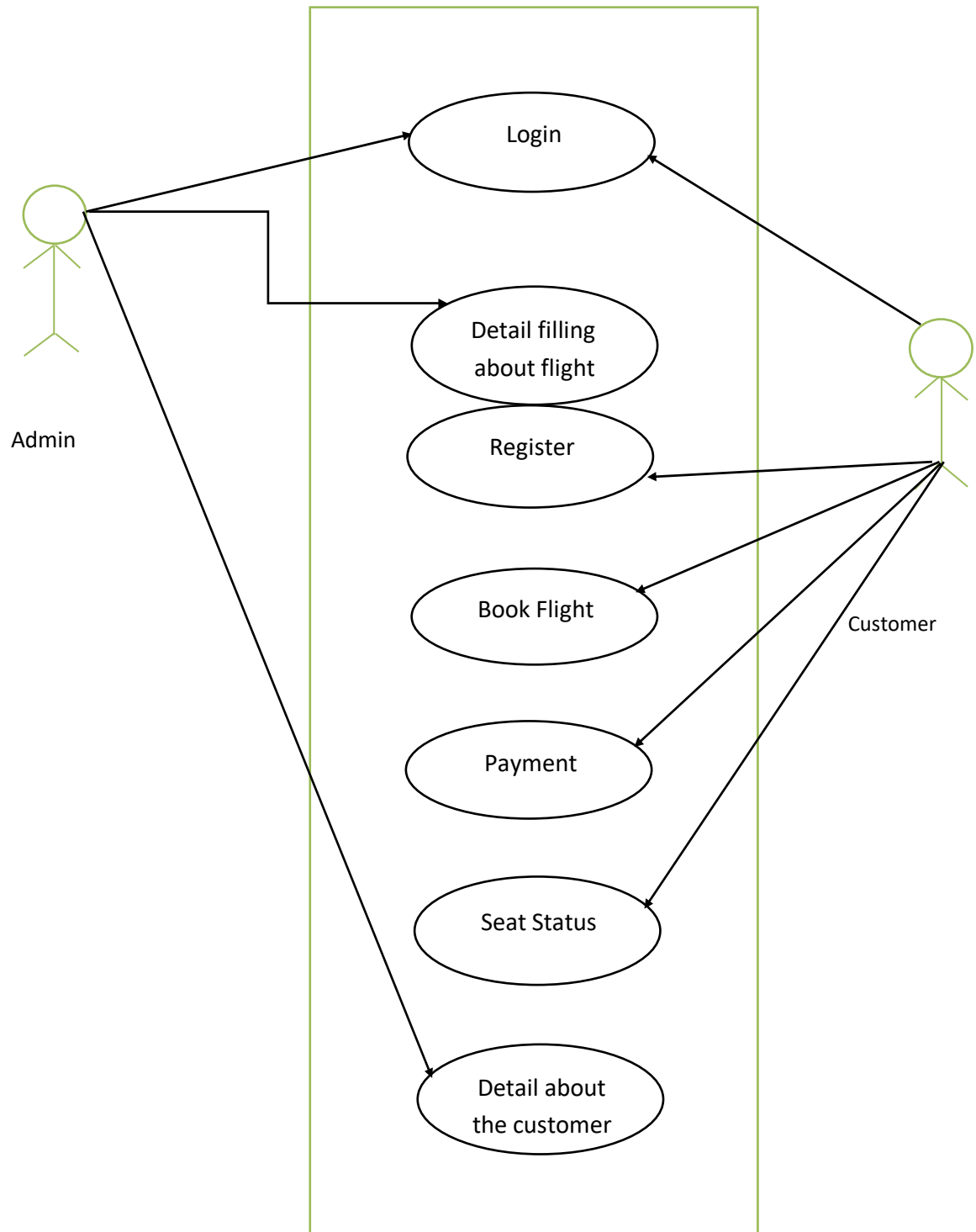
Flight details

[flight no]	[varchar](500)	NULL
[destination]	[varchar](500)	NULL
[orign]	[varchar](500)	NULL
[date]	[varchar](500)	NULL
[arrival]	[varchar](500)	NULL
[departure]	[varchar](500)	NULL
[price]	[varchar](500)	NULL

Payment

[username]	[varchar](500)	NULL
[password]	[varchar](500)	NULL
[mode]	[varchar](500)	NULL
[pay details]	[varchar](500)	NULL
[status]	[varchar](500)	NULL
[conform]	[varchar](500)	NULL
[ticket id]	[varchar](500)	NULL

Use Case Diagram



SYSTEM DESIGN

5.1. MODULE DESCRIPTION

The list of modules incorporated with “AIRLINE RESERVATION SYSTEM “is as follows:

Registration module

After registering with us the passenger can logon to his/her own account and can view all flight details such as Timings, Prices, Availability of seats and can book the ticket with unique ticket id and gives its personal details. Once Passenger registered with us can book any number of tickets.

Administrative module

Administrative module is provided for the sake of administrators to manage the site and update the content at regular intervals, the major operations included in this module are:

- Create and maintain airline schedule, fare and timings of the Flight.
- View the passenger list.
- View the available seats in the flights.
- Updating the flight schedule and timings and fare.

Passenger module

This module is meant for passengers, where a user logging into his/her owns account will view this panel. The major operations included in this module were ➤ View all airline schedules, timings, fare details and seats availability.

- Book for the tickets.
- View the ticket.

5.2. Software System Attributes

There are a number of attributes of software that can serve as requirements. It is important that required attributes by specified so that their achievement can be objectively verified.

The following items provide a partial list of examples. These are also known as non-functional requirements or quality attributes.

These are characteristics the system must possess, but that pervade (or cross-cut) the design. These requirements have to be testable just like the functional requirements.

It's easy to start philosophizing here, but keep it specific.

5.2.1. Reliability

It means the extent to which program performs with required precision. The website developed should be extremely reliable and secure so that information about any questions etc. is not leaked. The system shall not be down more than 2 times in a year.

5.2.2. Availability

Checking that the system always has something to function and always pop up error messages in case of component failure. In that case the error messages appear when something goes wrong so to prevail availability problems.

5.3.3. Security

The security requirements deal with the primarily security. The software should be handled only by the administrator and authorized users. Only the administrator has right to assign permissions like creating new accounts and generating password.

Specific requirements in this area could include the need to:

- Utilize certain cryptographic techniques
- Keep specific log or history data sets
- Assign certain functions to different modules
- Restrict communications between some areas of the program
- Check data integrity for critical variable.

5.2.4. Maintainability

The application is to be designed so that it is easily maintained. Also it should allow incorporating new requirements in any module of system. Backups for database are available.

5.2.5. Portability

The software is a web-based application and is built in PHP and My SQL. So, it is platform independent and is independent of OS. The application will be easily portable on any window-based system.

5.3. Organizing the specific Requirements

For anything but trivial systems the detailed requirements tend to be extensive. For this reason, it is recommended that careful consideration be given to organizing these in a manner optimal for understanding. There is no one optimal organization for all systems. Different classes of systems lend themselves to different organizations of requirements in section 3. Some of these organizations are described in the following subclasses.

5.3.1. System Mode

Some systems behave quite differently depending on the mode of operation. When organizing by mode there are two possible outlines. The choice depends on whether interfaces and performance are dependent on mode.

5.3.2. User Class

Some systems provide different sets of functions to different classes of user

5.3.3. Objects

Objects are real-world entities that have a counterpart within the system. Associated with each object is a set of attributes and functions. These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as class

5.3.4. Feature

A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. Each feature is generally described in as sequence of stimulus-response pairs.

5.3.5. Stimulus

Some systems can be best organized by describing their functions in terms of stimuli.

5.3.6. Response

Some systems can be best organized by describing their functions in support of the generation of a response.

5.3.7. Functional Hierarchy

When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by either common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

DATABASE DESIGN

The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, and flexible for the user. In database design, several objectives are considered.







Control Redundancy:

Redundant data occupies space and therefore, is wasteful. If versions of the data are in different phases of updating the system, it often gives conflicting information. A unique aspect of database design is storing data only once, which controls redundancy and improves system performance.

Table Structure

Registration Table

	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	<u>reg_id</u>	tinyint(11)			No	None	auto_increment
<input type="checkbox"/>	u_name	varchar(20)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	pswd	varchar(20)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	f_name	varchar(20)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	l_name	varchar(20)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	email	varchar(20)	latin1_swedish_ci		No	None	

↑ [Check All / Uncheck All](#) With selected:      

Passenger Table

	Field	Type	Collation	Attributes	Null	Default	Extra
	reg_id	tinyint(11)			No	None	auto_increment
	passport_no	varchar(10)	latin1_swedish_ci		No	None	
	visa_no	varchar(10)	latin1_swedish_ci		No	None	
	f_name	varchar(30)	latin1_swedish_ci		No	None	
	l_name	varchar(30)	latin1_swedish_ci		No	None	
	email	varchar(30)	latin1_swedish_ci		No	None	
	address	varchar(30)	latin1_swedish_ci		No	None	
	contact	varchar(10)	latin1_swedish_ci		No	None	
	pin_no	varchar(10)	latin1_swedish_ci		No	None	

Source Table

	Field	Type	Collation	Attributes	Null
<input type="checkbox"/>	<u>source_id</u>	int(11)			No
<input type="checkbox"/>	source_name	varchar(30)	latin1_swedish_ci		No

Destination Table

	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	<u>destination_id</u>	tinyint(20)			No	None	auto_increment
<input type="checkbox"/>	destination_name	varchar(20)	latin1_swedish_ci		No	None	

Flight Details

Field	Type	Collation	Attributes	Null	Default
<u>flight_id</u>	int(11)			No	None
flight_name	varchar(20)	utf8_general_ci		No	None
source_id	int(11)			No	None
destination_id	int(11)			No	None
arrival_time	int(11)			No	None

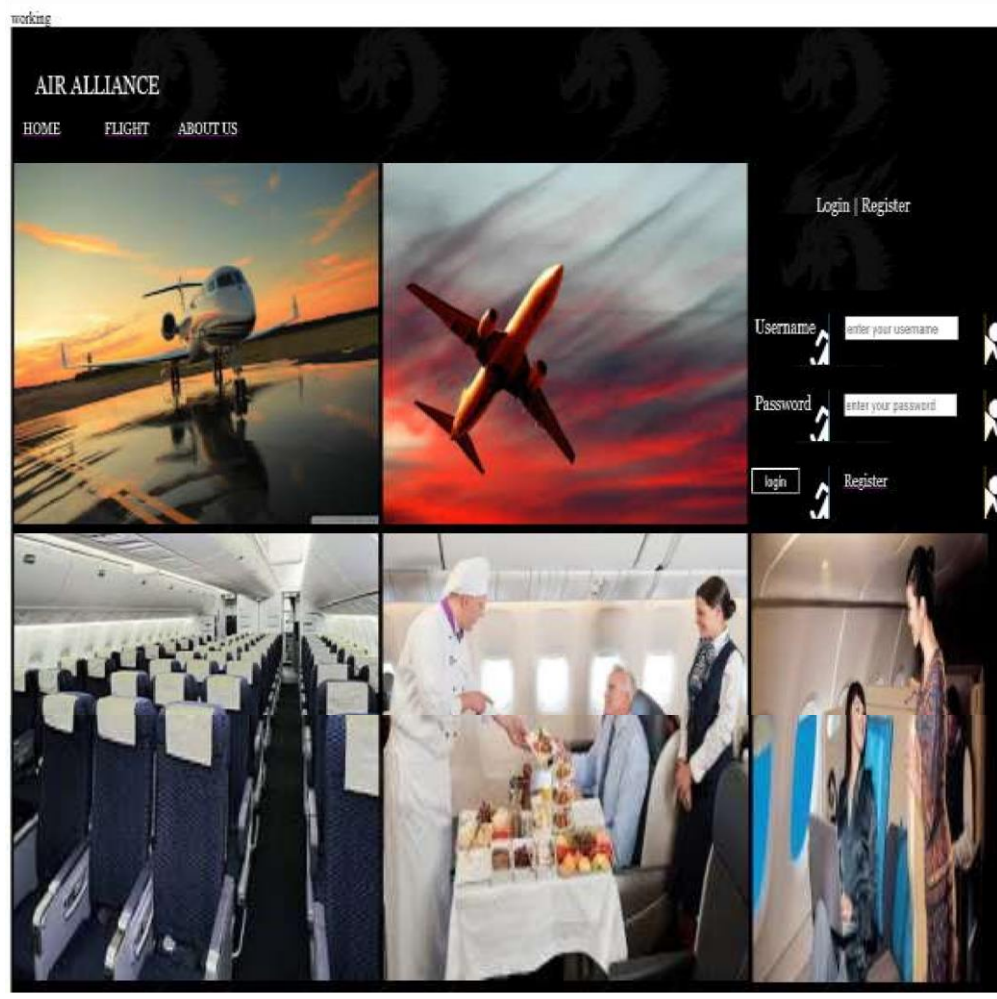
Set Flight

Field	Type	Collation	Attributes	Null	Default	Extra
set_flight_id	int(11)			No	None	
flight_id	int(11)			No	None	
source_id	int(11)			No	None	
destination_id	int(11)			No	None	
arrival_time	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP
departure_time	timestamp			No	0000-00-00 00:00:00	
fare	double			No	None	

Ticket Table

Field	Type	Collation	Attributes	Null	Default
<u>ticket_id</u>	int(11)			No	None
source_id	int(11)			No	None
destination_id	int(11)			No	None
depart_date	date			No	None
return_date	date			No	None
reg_id	tinyint(11)			No	None

Designing Forms : Home Page



Passenger Register Form

working

AIR ALLIANCE

HOME FLIGHT ABOUT US

User Registration

	Username	<input type="text" value="enter your username"/>	
	Password	<input type="text" value="enter your password"/>	
	First Name	<input type="text" value="enter your first name"/>	
	Last Name	<input type="text" value="enter your last name"/>	
	eMail	<input type="text" value="enter your email id"/>	

register

Flight Booking (Round Trip)

working

AIR ALLIANCE
HOME FLIGHT ABOUT US

BOOK YOUR FLIGHT

Round Trip

Leaving from

Depart Date

Return Date

Adult

Children

Detailed description: This is a screenshot of a flight booking interface for a round trip. The background is a dark, abstract geometric pattern with red highlights. At the top left, there is a navigation menu with 'AIR ALLIANCE' in large letters and 'HOME', 'FLIGHT', and 'ABOUT US' in smaller letters. On the right side, there is a section titled 'BOOK YOUR FLIGHT'. Below this title, there are two radio buttons: 'Round Trip' (which is selected) and 'One way'. The 'Leaving from' field is a dropdown menu with 'Delhi' selected. The 'Going to' field is also a dropdown menu with 'Delhi' selected. There are two rows of date pickers: 'Depart Date' and 'Return Date'. Each row has three input fields labeled 'DATE', 'MONTH', and 'YEAR'. Below the date pickers, there are two dropdown menus for the number of passengers: 'Adult' and 'Children', both with '1' selected. At the bottom center, there is a white button with the text 'search flights'.

Flight Booking (One Way)

AIR ALLIANCE
HOME FLIGHT ABOUT US

BOOK YOUR FLIGHT

Round Trip

One way

Leaving from

Going to

Depart Date

Adult

Children

Detailed description: This is a screenshot of a flight booking interface for a one-way trip. The layout is identical to the round trip form, but the 'One way' radio button is selected instead of 'Round Trip'. The 'Depart Date' field is now a single empty text input box instead of three separate fields for date, month, and year. The 'Return Date' field is completely absent. The 'Leaving from' and 'Going to' dropdown menus both show 'Delhi'. The 'Adult' and 'Children' dropdown menus both show '1'. The 'search flights' button is at the bottom center.

Passenger Details

AIR ALLIANCE

[HOME](#) [FLIGHT](#) [ABOUT US](#)

Please provide passenger details

Title ▾

First Name

Last Name

Date Of Birth ▾ ▾ ▾

Address

PIN code

Phone no

eMail

Passport No

Payment Details

AIR ALLIANCE

[HOME](#) [FLIGHT](#) [ABOUT US](#)

Make a payment of Rs **25000**

"Pay the given amount!"

About Us Panel

AIR ALLIANCE

[HOME](#) [FLIGHT](#) [ABOUT US](#)

ABOUT US

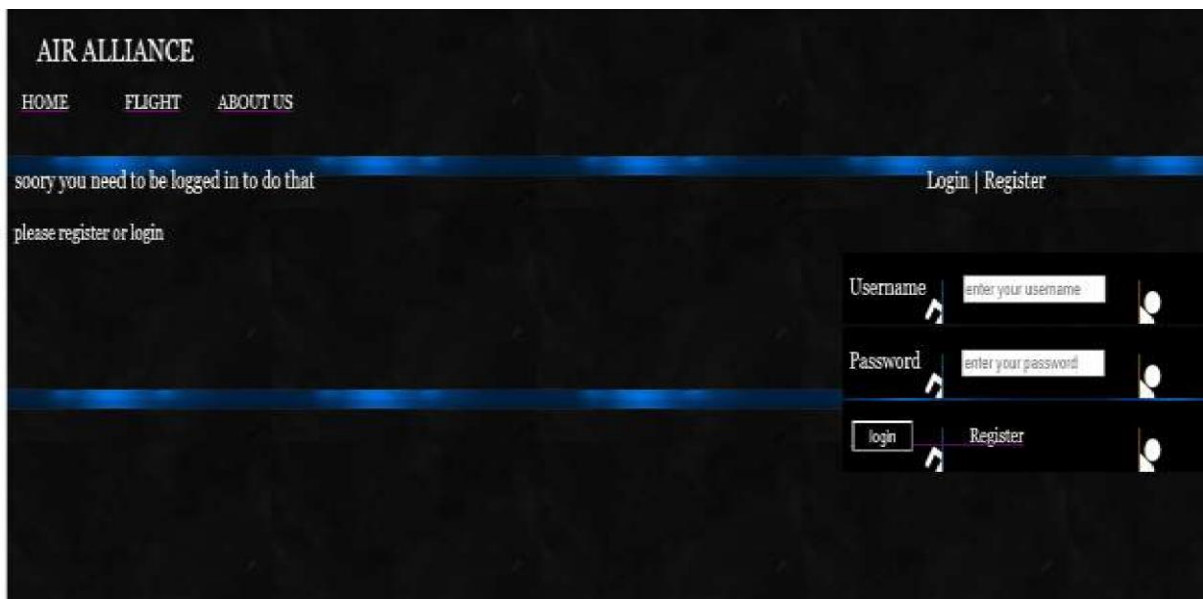
Air Alliance mission is to become India's preferred low cost airline, delivering the lowest air fares with the highest consumer value to price sensitive consumers to fulfil everyone dream of flying. It operates 306 weekly flights to 42 domestic destinations as a subsidiary of Air India. [4] It operates mostly in Tier-2 & Tier-3 Cities in India connecting them to major metro cities. In March 2017 the carrier reverted back to its Alliance Air name and introduced a simplified new livery featuring Air India's former Centaur logo. The flight schedules of Alliance Air are structured in a manner which enables easy connections to / from smaller cities via the hubs of Delhi, Kolkata, Mumbai, & Bengaluru & Hyderabad. Alliance Air provides connections to International stations through its hubs. Air Alliance vision is to address that and ensure that flying is for everyone. The power to fly for everyone with a dynamic fare structure

CARRIERS

Air Alliance, we believe that our people are our greatest asset. We are always on a lookout for a highly motivated individuals who can collaborate with like-minded people. In an environment that embraces individuality and rewards your best work. A career opportunity at air alliance will entail you to unlimited opportunity and a host of benefits. Come, be a part of Air Alliance



ADMIN PANEL























ADD ADMIN

AIR ALLIANCE

HOME ADD FLIGHT MODIFY FLIGHT DELETE FLIGHT

Admin Registration

		Username	<input type="text" value="enter your username"/>		
		Password	<input type="text" value="enter your password"/>		
		First Name	<input type="text" value="enter your first name"/>		
		Last Name	<input type="text" value="enter your last name"/>		
		eMail	<input type="text" value="enter your email"/>		

Modify Flight

HOME ADD FLIGHT MODIFY FLIGHT DELETE FLIGHT

Enter the flight id to be modified

Flight id

Flight name

Leaving from

Going to

Departure date

Depart time

Price

Delete Flight

HOME ADD FLIGHT MODIFY FLIGHT DELETE FLIGHT

Enter the Flight id to be deleted

Flight id	<input type="text"/>
Flight name	<input type="text"/>
Leaving from	<input type="text"/>
Going to	<input type="text"/>
Departure date	<input type="text"/>
Depart time	<input type="text"/>
Price	<input type="text"/>


```
</tr>
</table>
</body>
</html>
Register.php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<?php          require
```

```
"db_connection.php";
```

```
$sql="select * from registration";
```

```
$result=mysql_query($sql);
```

```
?>
```

```
<title>AIRLINE RESERVATION SYSTEM</title>
```

```
</head>
```

```
<body>
```

```
<table width="100%" height="622" border="0" cellpadding="2" align="center"
```

```
background="download3.jpg">
```

```
<form name = "form1" action="insert_new.php" method="post"> <script
```

```
type="text/javascript">
```

```
function check()
```

```
{ if(document.form1.Username.value=="")
```

```
{
```

```
    alert("Plese Enter User name");
```

```
cellspacing="2"
```

```
50
```



```
document.form1.Username.focus();

return false;

}

if(document.form1.Password.value=="")

{

    alert("Plese Enter Your password");    document.form1.Password.focus();

    return false;

}

if(document.form1.Fname.value=="")

{    alert("Plese Enter first name");

document.form1.Fname.focus();

    return false;

}

if(document.form1.lname.value=="")

{    alert("Plese Enter your last name");

document.form1.lname.focus();

    return false;

}

if(document.form1.email.value=="")

{    alert("Plese Enter Your email");

document.form1.gender.focus();

    return false;

}

e=document.form1.email.value;

f1=e.indexOf('@');

f2=e.indexOf('@',f1+1);
```



```

{
return !!filter_var($email,FLITER_VALIDATE_EMAIL); if(valid_email('you@example.com'))

{ echo 'Valid';

} else

{   echo 'Not

Valid';

}

}

</script>

<input name="email" type="text" value="" placeholder="enter your email id" /></td>

</tr>

<tr>

<td height="50">&nbsp;</td>

<td><a href="flightbooking.php">

<input id="register" name="register" type="submit" value="register" style="color:#FFF;

background-color:#000; HEIGHT: 40px; WIDTH: 65px" />

</a></td>

</tr>

</form>

</table>

<p>&nbsp;</p>

```

```
</body>
</html>
Check.php <?php require
"db_connection.php";

$sql="select * from registration where u_name='$_POST[Username]'
pswd='$_POST[Password]";

$result=mysql_query($sql);

$count=mysql_num_rows($result); if($count>0)

{ echo "<script type='text/javascript'> alert('THANK
YOU..! Successfully authenticated.');"

</script>"; include("flightbooking.php");

} else { echo
"error".mysql_error();

include("home.php");

}
?>
```

and

[HOME](home.php) [FLIGHT](flighthide.php) [ABOUT US](aboutus.php)

</tr>

<tr>

<td width="61%" height="201">ABOUT US<p>Air Alliance mission is to become India's preferred low cost airline, delivering the lowest air fares with the highest consumer value to price sensitive consumers to fulfil everyone's dream of flying. It operates 306 weekly flights to 42 domestic destinations as a subsidiary of Air India. [4] It operates mostly in Tier-2 & Tier-3 Cities in India connecting them to major metro cities.

In March 2017 the carrier reverted back to its Alliance Air name and introduced a simplified new livery featuring Air India's former Centaur logo. The flight schedules of Alliance Air are structured in a manner which enables easy connections to / from smaller cities via the hubs of Delhi, Kolkata, Mumbai, & Bengaluru & Hyderabad. Alliance Air provides connections to International stations through its hubs. Air Alliance vision is to address that and ensure that flying is for everyone. The power to fly for everyone with a dynamic fare structure

</td>

</tr>

<tr>


```
<option value="4">Guwahati</option>
<option value="5">Gwalior</option>
<option value="1">Jabalpur</option>
<option value="2">Jaipur</option>
<option value="3">Jammu</option>
<option value="4">Kanpur</option>
<option value="5">Kochi</option>
<option value="1">Kullu</option>
<option value="2">Patnagar</option>
<option value="3">Ranchi</option>
<option value="4">Raipur</option>
<option value="5">Shimla</option>
<option value="1">Surat</option>
<option value="2">Vijayawada</option>
<option value="3">Vizag</option>
<option value="4">Tezpur</option>
<option value="5">Tirupati</option> </select></td>
```

```
<td style="background-color:#000" <font color="#FFFFFF" face="Georgia, Times New Roman, Times, serif" size="+1" style="background-color:#000">Going to</font>
```

```
<label for="Going to"></label>
<select name="Going to" id="Going to">
<option value="Delhi">Delhi</option>
<option value="Mumbai">Mumbai</option>
<option value="Goa">Goa</option>
<option value="4">Hyderabad</option>
<option value="5">Kolkata</option>
<option value="1">Pune</option>
<option value="2">Allahabad</option>
<option value="3">Bengaluru</option>
<option value="4">Bhopal</option>
<option value="5">Bhuj</option>
<option value="1">Dehradun</option>
```



```
<option>DATE</option>
<?php

for($i=1;$i<=31;$i++)
{
echo "<option>".$i."</option>";
}
?>

</select> &nbsp;
<select name="month" style="color:#43484A">
<option>MONTH</option>
<option>JANUARY</option>
<option>FEBURARY</option>
<option>MARCH</option>
<option>APRIL</option>
<option>MAY</option>
<option>JUNE</option>
<option>JULY</option>
<option>AUGUST</option>
<option>SEPTEMBER</option>
<option>OCTOBER</option>
<option>NOVEMBER</option>
<option>DECEMBER</option>
</select> &nbsp;
<select name="year">
<option>YEAR</option>
<?php

for($i=1990;$i<=2016;$i++)
{
echo "<option>".$i."</option>";
}
}
```



```

<tr>
  <td width="295" height="32" align="center"><font color="#FFFFFF" face="Georgia, Times New
Roman, Times, serif">First Name</font></td>

  <td width="771"><input name="fname" type="fname" placeholder="enter your first name"
/></td>
</tr>
<tr>
  <td width="295" height="32" align="center"><font color="#FFFFFF" face="Georgia, Times New
Roman, Times, serif">Last Name</font></td>

  <td width="771"><input name="lName" type="lname" placeholder="enter your last name"
/></td>
</tr>
<tr>
  <td width="295" height="32" align="center"><font color="#FFFFFF"
face="Georgia, Times New Roman, Times, serif">Date Of Birth</font></td>

  <td width="771"><select name="date">
<option>DATE</option>
<?php
for($i=1;$i<=31;$i++)
{
echo "<option>".$i."</option>";
}
?>

</select> &nbsp;
<select name="month" style="color:#43484A; height:30px;">
<option>MONTH</option>
<option>JANUARY</option>

```

```
        <option>FEBURARY</option>
        <option>MARCH</option>
    <option>APRIL</option>

<option>MAY</option>
<option>JUNE</option>
<option>JULY</option>
    <option>AUGUST</option>
    <option>SEPTEMBER</option>

    <option>OCTOBER</option>
    <option>NOVEMBER</option>
    <option>DECEMBER</option>
</select> &nbsp;
<select name="year">

    <option>YEAR</option>
    <?php

        for($i=1990;$i<=2016;$i++)
        {
            echo

            "<option>".$i."</option>";

        }
    ?>

</select>
```

```
</td>
```

```
</tr>
```

```

<tr>
  <td width="295" height="32" align="center"><font color="#FFFFFF" face="Georgia, Times New
Roman, Times, serif">Address</font></td>

  <td width="771"><input name="address" type="address" placeholder="enter
your address" /></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td width="771"><input name="address" type="address" placeholder="enter
your address" /></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td width="771"><input name="address" type="address" placeholder="enter
your address" /></td>
</tr>
<tr>
  <td width="295" height="32" align="center"><font color="#FFFFFF" face="Georgia, Times New
Roman, Times, serif">PIN code</font></td>

  <td width="771"><input name="pin code" type="pin code" placeholder="enter your
PIN code" /></td>
</tr>
<tr>
  <td width="295" height="32" align="center"><font color="#FFFFFF" face="Georgia, Times New
Roman, Times, serif">Phone no</font></td>

  <td width="771"><input name="phone no" type="phone no" placeholder="enter your phone no"
/></td>

```



```

<td width="189"></td>
</tr>
<tr>
<td height="118">&nbsp;</td>
<td width="159"><a href="addadmin.php"><font color="#FFFFFF" face="Georgia,
Times New Roman, Times, serif">ADD ADMIN</font></a></td>
<td width="185"></td>
<td width="508" rowspan="4"></td>
</tr>
<tr>
<td height="110">&nbsp;</td>
<td width="159"><a href="addflight.php"><font color="#FFFFFF" face="Georgia,
Times New Roman, Times, serif">ADD FLIGHT</font></a></td>
<td width="185"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td width="159"><a href="modify_flight.php"><font color="#FFFFFF" face="Georgia, Times New
Roman, Times, serif">MODIFY FLIGHT</font></a></td> <td width="185"></td>
<td width="7"></td>
</tr>
<tr>
<td width="189"></td>
<td width="159"><a href="delete_flight.php"><font color="#FFFFFF" face="Georgia, Times New
Roman, Times, serif">DELETE FLIGHT</font></a></td> <td width="185"></td>
</tr>

```

```

<tr>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td rowspan="3">&nbsp;</td>      <td
rowspan="3">&nbsp;</td>
</tr>
<tr>
  <td>&nbsp;</td>  <td>&nbsp;</td>
</tr>
<tr>
  <td>&nbsp;</td>  <td>&nbsp;</td>
</tr>
</table></td>
</tr>
</table><a href="delete_flight.php"><font color="#FFFFFF"          face="Georgia,
Times New Roman, Times, serif"></font></a></td>
</tr>
</table>      <font color="#FFFFFF"    face="Georgia, Times New Roman, Times,
serif">&nbsp;</font></td>
</tr>
</table>
</body>
</html>
</body>
</html>

```

System Testing

The testing process focuses on the logical intervals of the software ensuring that all statements have been tested and on functional interval is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with the required results. Program level testing, modules level testing integrated and carried out.

9.1. Testing Methods

There are two major type of testing they are

- White Box Testing.
- Black Box Testing.

White Box Testing

White box sometimes called “Glass box testing” is a test case design uses the control structure of the procedural design to drive test case.

Black box Testing

Black box testing focuses on the functional requirements of the software. This is black box testing enables the software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box testing rather it is complementary approach that is likely to uncover a different class of errors that white box methods like..

- Interface errors
- Performance in data structure
- Performance errors
- Initializing and termination errors

9.2. Unit testing

Unit testing is a software verification and validation method in which a programmer tests if individual units of source_code are fit for use.

A unit is the smallest testable part of an application. In procedural programming a unit may be an individual function or procedure.

Ideally, each test case is independent from the others: substitutes like method stubs, objects, fakes and test harnesses can be used to assist testing a module in isolation.

Integration Testing:

This testing is sometimes called Integration and Testing. Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates and delivers as its output the integrated system ready for system testing.

Validation Testing:

Validation Testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that can reasonably expected by a customer. After validation test has been conducted, one of the following two possible conditions exists. The functions or performance characteristics confirm to specification and are accepted.

- In the administrator and login modules, all the fields must be filled.
- In the student registration, contact number should contain exactly 10 numbers.

User Acceptance Testing: User acceptance of a system is a key factor of any system.

The system under consideration is tested for the acceptance by constantly keeping in touch with the prospective system users at the same time of developing and marketing changes whenever required.

This is done in regard to the following points:

- Input Screen Design
- Output Screen Design

9.3. Test Cases

NO	INPUT GIVEN	EXPECTED OUTPUT	ACTUAL OUTPUT OCCURED	TEST PASS	ACTION TAKEN
1	Admin, pass	Admin Home page	Admin Home page	Yes	-
3	Admin, kumar	Admin Home page	Invalid password for user Admin	No	The wrong password ***** is given for user Admin.
4	Login, Sanjana19	Flight Booking	Successfully Authenticated	Yes	-

CONCLUSION

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in Adobe Dreamweaver web-based application and to some extent Windows Application and SQL Server, but also about all handling procedure related with “**Alliances**”. It also proves knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently. The purpose of object-oriented programming is to have code that is structured and organized. For the most part, you’ll find yourself in situations where you’ll be reusing code. Instead of having to recreate that same logic for each project, you can have a class that does most of the work for you.

Another reason for having such strict concepts is because you may end up sharing your class. For example, let’s say you had a database class. A lot of people connect to databases. Some people may be looking for code that can help them connect to a database easier and run certain functions. By creating your database connector as a class, you help prevent collisions in their own program because all code wrapped in a class is unique to the class only.

By defining your classes as public or private, you also prevent developers from ruining the functionality of your class. Of course, they can just go into the class itself and change things around, but then they would be destroying the purpose of object-oriented programming. What if your class updates? They’ll have to do it again and again. You’ll find yourself using other developer’s classes and knowing OOP will make you a better programmer.

You don’t have to use every single concept for your classes. It’s really up to you, but please utilize what you can to help make your code perform better. You don’t have to use classes for all your projects. It really depends on you and how you wish to use it. The basic rule is, if you’re doing the same thing over and over again, then you should probably create a class for it.

BENEFITS:

The project is identified by the merits of the system offered to the user. The merits of this project are as follows: -

- It's a web-enabled project.
- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stage of any new creation, data entry or updating so that the user cannot enter the invalid data, which can create problems at later date.
- Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover, there is restriction for his that he cannot change the primary data field. This keeps the validity of the data to longer extent.
- User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
- From every part of the project the user is provided with the links through framing so that he can go from one option of the project too other as per the requirement. This is bound to be simple and very friendly as per the user is concerned. That is, we can say that the project is user friendly which is one of the primary concerns of any good project.
- Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
- Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time than manual system.

- Allocating of sample results becomes much faster because at a time the user can see the records of last years.
- Easier and faster data transfer through latest technology associated with the computer and communication.
- Through these features, it will increase the efficiency, accuracy and transparency,

Future Scope and further enhancement of the Project

The Online Airline Reservation system is the next generation address book which will provide these two basic services like portability, security.

The future scope includes expand the technologies like HTML and PHP we can also add new technologies like HTML, php many more for improving the efficiency of the software.

The project will be useful for any schools and colleges with slightly modification.

Project is flexible i.e., any change /modification in database may be performing easily.

Also, this project could be made web enabled.

Assumptions, if any

None

Assumptions:

The user is familiar with basic computer components and operations.

Dependencies:

The system should work on all systems.