SEPM
REPORT

DONE BY:

AKSHAT RAWAT (RA1911030010088)
YUKTA GOEL (RA1911030010089)
RAHUL GOEL (RA1911030010094)

PROBLEM STATEMENT:

Aim: To Frame a project team, analyze and identify a Software project

Project Title: **Airline Booking Management System**

Project Description:

Large Airlines have to deal with thousands of airline reservations every day which makes it challenging to manage these reservations and cancellation requests made by users. Thus arises the need for an application based on a powerful database management system.

The aim of this project is to create an Airline Booking Management System which will hold flight schedules and its fare tariffs and passenger reservations. 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 perform addition, deletion, and modification of flights if required.
The admin will also be able to view passenger reservations of the flights.
The users will be able to view all flight related information online and will be able to make cancellation requests effortlessly.

Result:

Thus, the project team formed and the project is described

BUSINESS CASE STUDY:

Strategic Business Context

Business Need

The main purpose of an airline site cannot be fulfilled without an effective flight booking system. Users need the proper airline reservation system to search and book flights so that they can travel to various destinations around the World.

Business Outcomes

An online flight reservation system will enable users to book flight tickets online without having to physically go somewhere to book tickets. This system will also enable the airline to handle these booking requests seamlessly and thus reduce costs to handle such requests.

Detailed Business Problem

Problem/Opportunity Statement

For a user to book an airline ticket he/she is required to go to the airline site, look at fares then contact an agent who then book the ticket via a system accessible to them. This is a tedious process for both the airline and the user. Also such a transaction is costly for the airline. This can be avoided by cutting the middleman and letting the user directly book flights using the airline website.

High Level Requirements

This would require a server which would be owned by the airline to receive requests, manage the client database and the flight database.

Assumptions

S.No	Assumptions
1.	There are no assumptions as of now. To be updated in later versions of the vision document.

Constraints

S.No	Category	Constraints
1.	Deadline	March 2021

Dependencies

Dependency Description	Critical Date	Contact
Na	Na	Na

Stakeholder Analysis

<Describe the stakeholder environment by considering the types of stakeholders, their specific roles, and their contributions to the realization of the investment>

Name	Designation	Role in Project
Mr. Jay Patel	Corporate Head for Sales & Marketing	Executive Sponsor
Mr. Nishant Raghuvanshi	Chief Information Officer Or Regional Head of Sales & Marketing	Project Sponsor
Mr. Suryam Juneja	Finance Head	Cost Approver
Mr. Akshat Uniyal	Department Head(s)	Scope / Requirement Approver
Mr. Akshit Jain	Business User(s)	Validate the functionalities

REQUIREMENTS ELICITATIONS & USER STORIES

Requirements Elicitations:

Airline Booking Management system is a project which aims in developing a computerized system to maintain all the reservations records, boom reservation and other online features in it. This project has many features which are generally not available in booking in person or from a third party, we added features like facility of user login. It also has a facility of admin login through which the admin can monitor the whole system. It has also a facility where clients after logging in their accounts can see list of reservation booked and its reserved. The client after logging into his account can generate various options

other than booking like give a feedback etc. Overall, this project of ours is being developed to help the users as well as staffs to maintain the records in a discrete and the best way possible and also reduce the human efforts.

1. Project Scope

Expand on the scope definition and outline the major activities required to successfully complete the project (for example, develop module ABC, develop requirements document, etc.). Out of scope activities are identified to reduce ambiguity.

In Scope

Out of Scope

S.No	Activities In Scope	Activities Out of Scope
1.	Designing Project Layout	Customer service in person
2.	Planning methods of project work	Funding for standard uplift
4.	Develop and test modules	Promoting chain marketing and branding
5.	Performing quality checks	Reading environment
6.	Constraining software size	Marginal tariff setting for services
7.	Public Deployment/Hosting	Proper customer communication

Requirements

Functional Requirements

Functional Requirements can also be expressed in the form of “user story” which is the smallest unit of work in an agile framework. It’s an end goal, not a feature, expressed from the software user’s perspective.

Requirement (#)	Requirement Specification	Department	Name of Business User	Status
E1FR1	The system must be able to allocate proper request we should be able to search based on history	Department of Media	Mr. Singhla	Journalism
E1FR2	System must be able to enter issue information in the database. System must be able to update when required	Management	Mr. Bahl	Resort Manager
E1FR3	System should be able to recommend popular search available from users search history	Technical Team, XYZ	XYZ Software Solutions and Consultancy.	SDE 1

Non-Functional Requirements

Requirement (#)	Category of NFR	Requirement Specification	Department	Name of Business User	Status
NFR1	Performance	All pages should load within 3 seconds	Dept. Of Media	Mr. AK Singh	Editor in Chief
NFR2	Performance	Search should bring the results less than 7 seconds	Business Development	Vivek Bindra	Business Coach
	Availability	Application should be available for 24x7	Medical Consultancy	G.N.S Krishna Murthi	Medical Representative
	Scalability	Registration Service should scale to serve 1000 request per second over 5 minutes timespan	Education	Dr. Murthy Roy	Professor

	Confidentiality	All records are being held I privacy which only the user can access.	Communication	Mr. Obroy	Senior Manager
	Compliance	Agreeing terms and condition before using the application.	Income Tax	Mr. Kiran	Inspector
E1NFR2	Usability	Easily Adaptable to every user	Revenue	Mrs. Priya	H.O.D
	Security	Have personal password interface, with upgraded features.	Administration	Mrs. Dyna Chanu	Collector
	Traceability	(according to user's preference)	Student	Mr. Raj	Student
	Flexibility	Multi-Language and Help support.	Software Dept.	Mr. Swami Reddy	Team Leader
E1NFR1	Extensibility	Ordering Service should be extensible to other countries within 2 days	Bussiness	Mr. Rajiv Kumar	Manager
	Interoperability	Can interact via multiple devices.	Tourist	Mr. Jack Johnson	Vlogger, Photographer
	Reliability	Full secure and trustworthy.	N/A	Mrs. Sharma	Housewife
	Rapidity	Integrate New Payment Integrator	Management	Mr. J.K Singhania	CA

Infrastructure Requirements

Requirement (#)	Requirement Specification	Department	Name of Business User / Project Team Member
IR1	Development Machine with atleast 12GB RAM and intel i5 or equivalent processor.	Software Dept.	Akshat Rawat
IR2	Code Repository	Software Dept.	Yukta Goel
IR3	AWS S3 Bucket	Software Dept.	Akshat Rawat
IR4	IDE – VS Code	Software Dept.	Rahul Goel

Requirement definition in Agile

User story is the smallest unit of work in an agile framework. It's an end goal, not a feature, expressed from the software user's perspective.

- As a developer, our team wants to bring change in airline reservation in an easy way to benefit all users, and enjoy this service together.
- As a team, we want to organize our work, so we can feel more in control.
- As a manager, I want to be able to understand my colleagues progress, so I can better report our success and failures.

COST AND RISK ANALYSIS:

Executive Summary

A project estimate is prediction of how much time and money is needed to complete a project. A well proper planning has been set up before hand, in a sequence wise.

The very first, Project Management Plan being one of the key elements to follow on further. The content in project planning contains key issues driving the project.

The very next provides Estimation regarding Effort, Cost Estimation and Infrastructure/Resource Cost. Here, we would deal with the cost and expenditure which are circulating throughout and an approximate cost when project comes in active, i.e., maintenance and support cost. This also includes the marketing expenditure to attract clients. Later, the project team formation with the Responsibility Assignment Matrix.

Estimation

Project Team Formation

A	Accountable
R	Responsible
C	Consult
I	Inform

Identification Team members

Name	Role	Responsibilities
RAY	Key Business User (Product Owner)	Provide clear business and user requirements
Mr. Akshat.	Project Manager	Manage the project
Mr. Rahul	Business Analyst	Discuss and Document Requirements
Ms. Yukta	Technical Lead	Design the end-to-end architecture
Mr. Rahul	UX Designer	Design the user experience
Ms. Yukta	Frontend Developer	Develop user interface
Ms. Akshat	Backend Developer	Design, Develop and Unit Test Services/ API/DB
Mr. Akshat	Cloud Architect	Design the cost effective, highly available and scalable architecture
Ms. Yukta	Cloud Operations	Provision required Services
Mr. Rahul	Tester	Define Test Cases and Perform Testing

PROJECT PLANNING:

Effort and Cost Estimation

WBS	Activity	Activity Description	Sub-Task	Sub-Task Description	Effort (in hours)	Cost in INR
E1FR 1	E1R1A 1	Website Development	E1R1A1T 1	Create desktop level website using HTML	4-5	60-70K
			E1R1A1T 2	Design of application using CPP, or other.	4-5	60-70K
			E1R1A1T 3	Using and MySQL	4-5	60-70K
E1FR 2	E1R1A 2	Research and Development	E1R1A2T 1	Evaluation of software technology trends and incorporating them with updates and patches	4-5	For Research - 80K For Rolling out patches/ updates – 50K
E1FR 3	E1R1A 3	Data Analytics	E1R1A3T 1	Administrator of Data, Database Management, Evaluating trends in data	8-10 With Shifts	Data Analyst-100K Data Scientist-90K Database Manager-80K
E1FR 4	E1R1A 4	Website Maintenance	E1R1A4T 1	Weekly website maintenance check	6 hours a week	Web developer-10 K
E1FR 5	E1R1A 5	Licensing/ Trademarks/ Copyrighting	E1R1A5T 1	Database IDE Server Terms Trademarking Copyrighting	(one time)	1.5L (one time)
E1FR 6	E1R1A 6	Marketing	E1R1A6T 1	Advertisements, Public Relations	8-10 On requirement	80K-1L

E1FR7	E1R1A7	Customer Care	E1R1A7T1	Customer care	24 With shifts	Customer Care-12K to 20K

Effort (hr)	Cost (INR)
1	500

Infrastructure/Resource Cost [CapEx]

Infrastructure Requirement	Qty	Cost per Qty	Cost per Item
Domain Rights	1	2 - 5k	2 - 5k
Server firm and other hardware	1	150K	150K
Legal and other government documents	1	200K	200K

Maintenance and Support Cost [OpEx]

Category	Details	Qty	Cost per qty per annum	Cost per item
People	Network, DB Admin, Customer support, Developer	4	20 L	80 L
License	Database, server, Middleware, Government License	3	10,000	30,000
Infrastructures	Server Storage and Network	2	20,000	40,000

Project Management Plan

Focus Area	Details
Integration Management	Governance Framework Project Team Structure Roles & Responsibilities of Team Change Management (Change Control, Issue Management) Project Closure
Scope Management	Scope Statement Requirement Management (Gathering, Control, Assumption, Constraint Stakeholder) Define Deliverable Requirement Change Control Activities and Sub-Tasks
Schedule Management	Define Milestones Schedule Control
Cost Management	Estimate Effort Assign Team Budget Control
Quality Management	Quality Assurance: Quality assurance will be managed including governance, roles and responsibilities, tools and techniques and reporting Quality Control: Specify the mechanisms to be used to measure and control the quality of the work products
Resource Management	Estimate and Manage the need People: People & Skills Required Finance: Budget Required Physical: Facilities, IT Infrastructure
Stakeholder	Identifying, Analysing, Engaging Stakeholders
Communication Management	Determine communication requirements, roles and responsibilities, tools and techniques. [Type of Communication, Schedule, Mechanism Recipient]
Risk Management	Identifying, analysing, and prioritizing project risks
Procurement Management	Adhering to organization procurement process

Responsibility Assignment Matrix

RACI Matrix	Team Members			
Activity	**Name (BA)**	**Name (Developer)**	**Name (Project Manager)**	**Key Business User**
User Requirement Documentation	A	C/I	R	C
Code – I	R	A	C/I	A
Code -II	A	C/I	R	A
Test	A	C/I	R	A
Feedback Collection	A	C/I	R	A
Deployment	A	C/I	R	C

WORK BREAKDOWN STRUCTURE(WBS) With Project Schedule

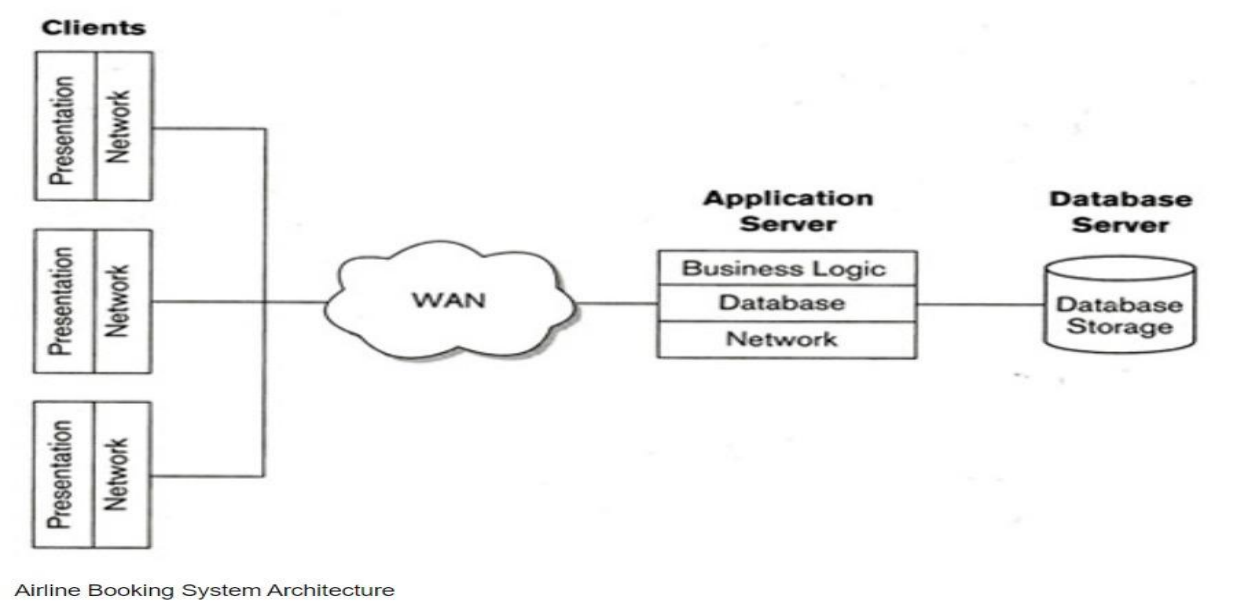
Module (#)	Activity (#)	Sub-Task(#)	Assignee(s)	Planned Start Date	Planned End Date	Actual Start Date	Actual End Date	Status
Client-side development.	UI/UX	-	Mr Akshat	09-03-2021	09-06-2021	10-03-2021	09-06-2021	In Progress
Server-side development	Data Base Management	-	Ms Yukta	09-03-2021	09-06-2021	10-03-2021	09-06-2021	In Progress
Refactoring the code	Code validation	-	Mr Akshat	24-05-2021	27-05-2021	25-05-2021	28-05-2021	In Progress
Testing for security flaw	Code verification	-	Ms Yukta	28-05-2021	2-06-2021	29-05-2021	3-06-2021	In Progress
Final Testing	Testing	-	Mr Rahul	04-06-2021	08-06-2021	5-06-2021	8-06-2021	In Progress

SYSTEM DIAGRAMS:

SYSTEM ARCHITECTURE:

An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between **components**. It is an important tool as it provides an overall view of the physical deployment of the software system and its evolution roadmap.

For an airline booking system the architecture diagram will demonstrate the relationship between the frontend user web page to the payment gateway.



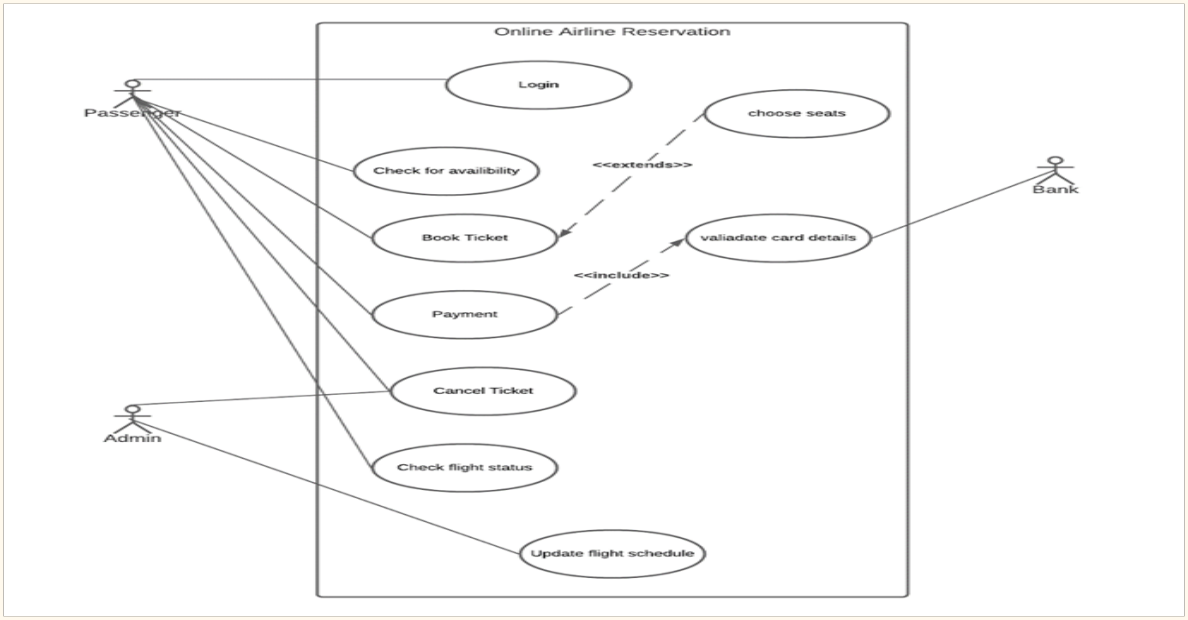
Airline Booking System Architecture

USE CASE DIAGRAM:

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

It represents the graphic depiction of the interactions among the

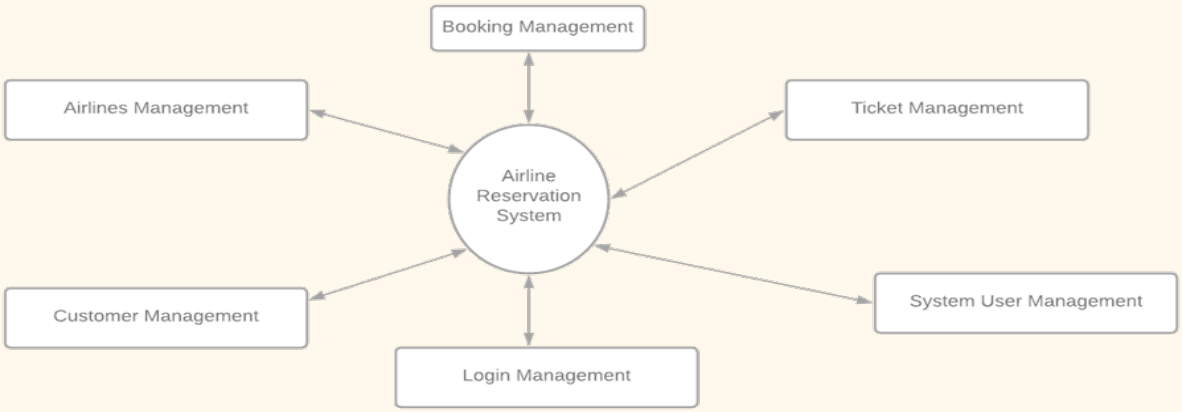
elements of the airline reservation system.



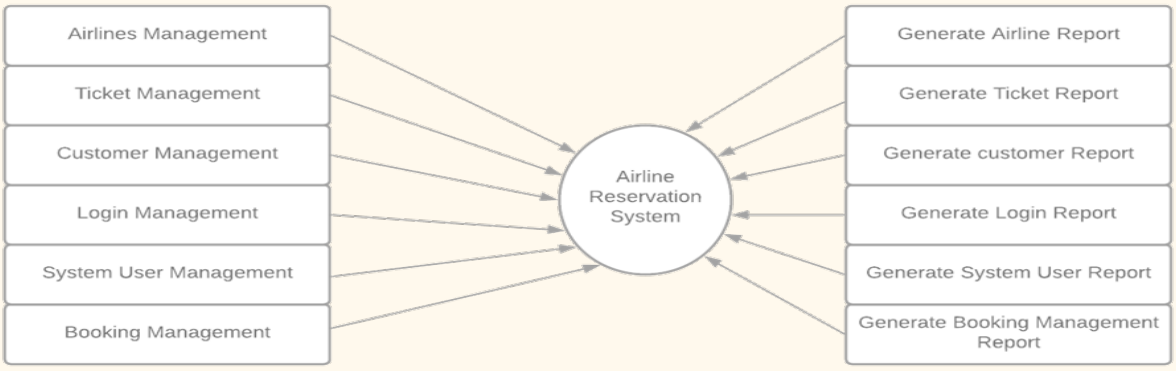
DFD DIAGRAM:

Also known as **DFD**, **Data flow diagrams** are used to graphically represent the flow of data in a business information system. **DFD** describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. **Data flow diagrams** can be divided into logical and physical.

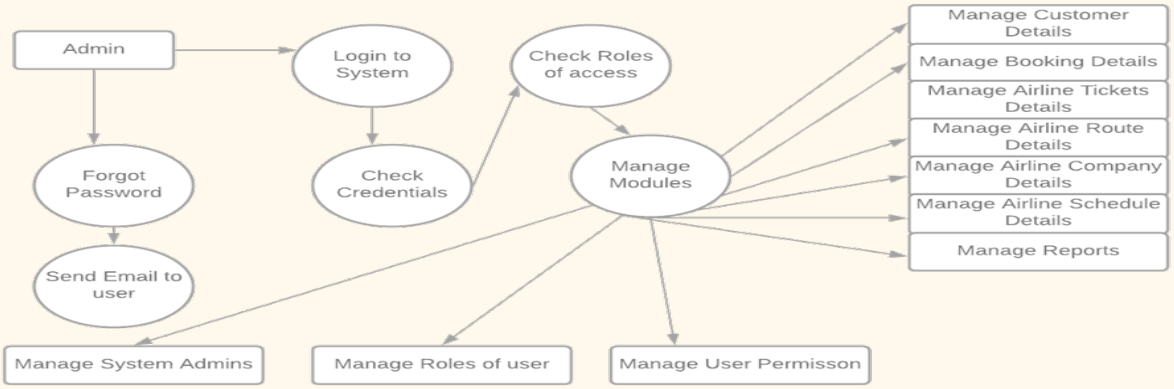
ZERO LEVEL:



FIRST LEVEL:



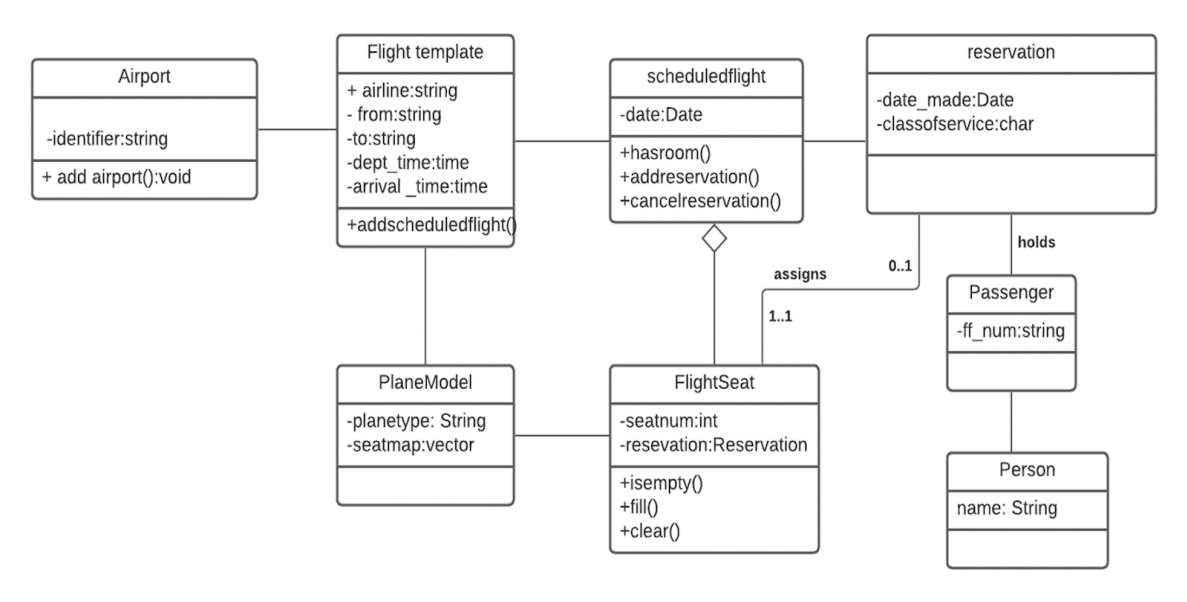
SECOND LEVEL:



CLASS DIAGRAM:

class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. **Class diagrams** can also be used for data modeling.

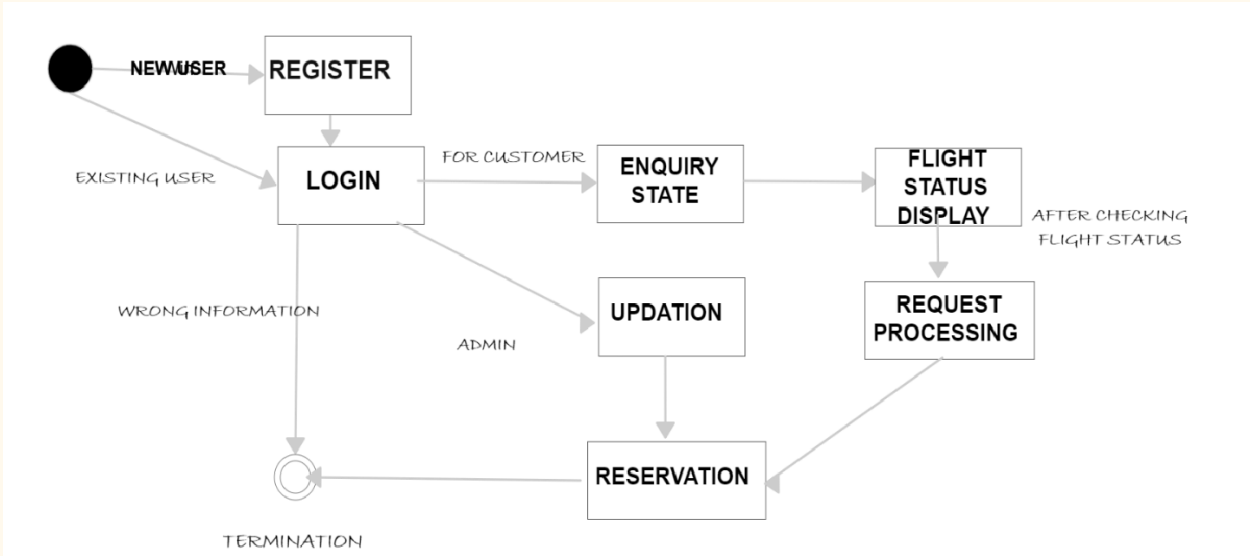
In the airline booking management system it describes the airline booking system classes, their attributes , operations and the relationships between the objects

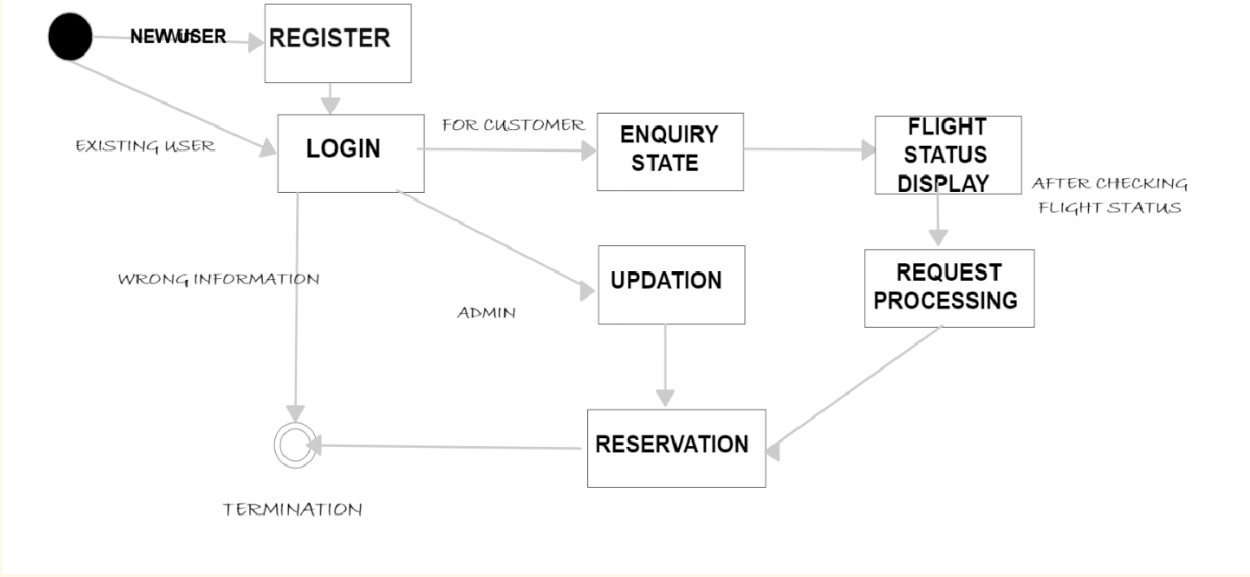


DESIGN STATE:

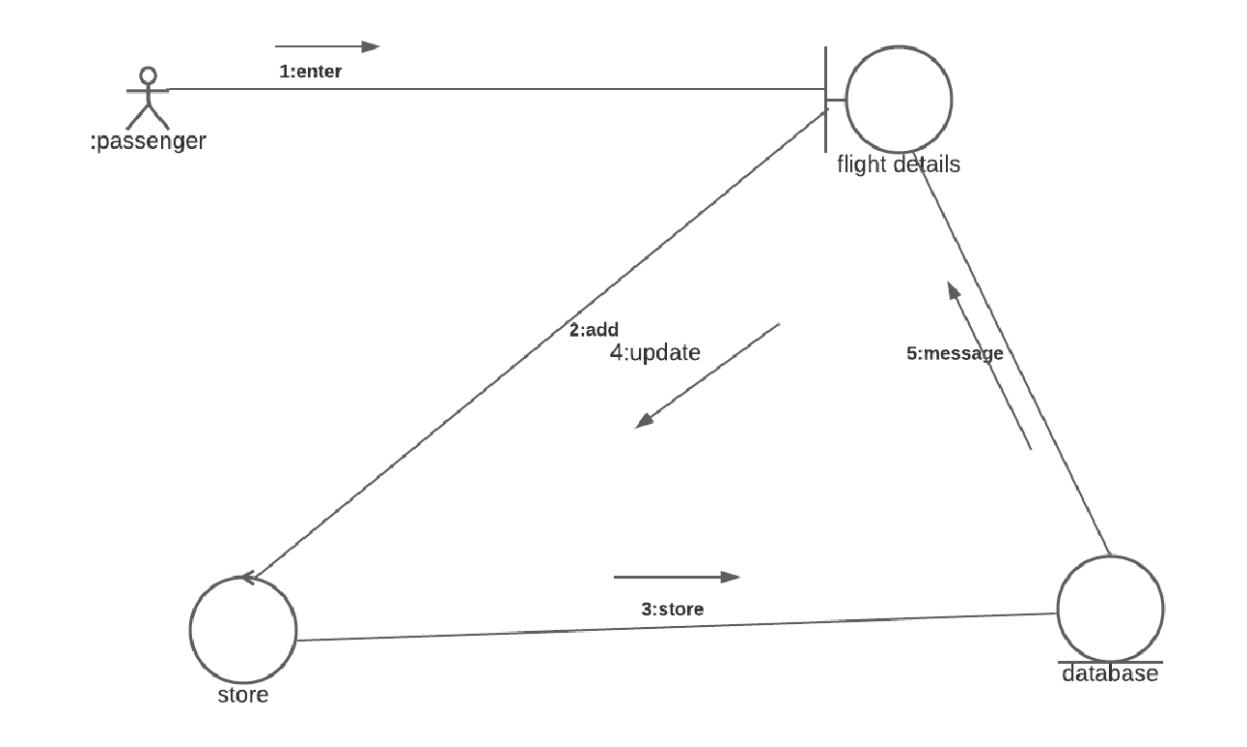
State Chart Diagram

State-chart diagrams are used to describe the states of different objects in its life cycle. Emphasis is placed on the state changes upon some internal or external events. These states of objects are important to analyze and implement them accurately.


NEW USER
REGISTER
EXISTING USER
LOGIN
FOR CUSTOMER
ENQUIRY STATE
FLIGHT STATUS DISPLAY
AFTER CHECKING FLIGHT STATUS
WRONG INFORMATION
ADMIN
UPDATION
REQUEST PROCESSING
RESERVATION
TERMINATION


NEW USER
REGISTER
EXISTING USER
LOGIN
FOR CUSTOMER
ENQUIRY STATE
FLIGHT STATUS DISPLAY
AFTER CHECKING FLIGHT STATUS
WRONG INFORMATION
ADMIN
UPDATION
REQUEST PROCESSING
RESERVATION
TERMINATION

Collaboration Diagram

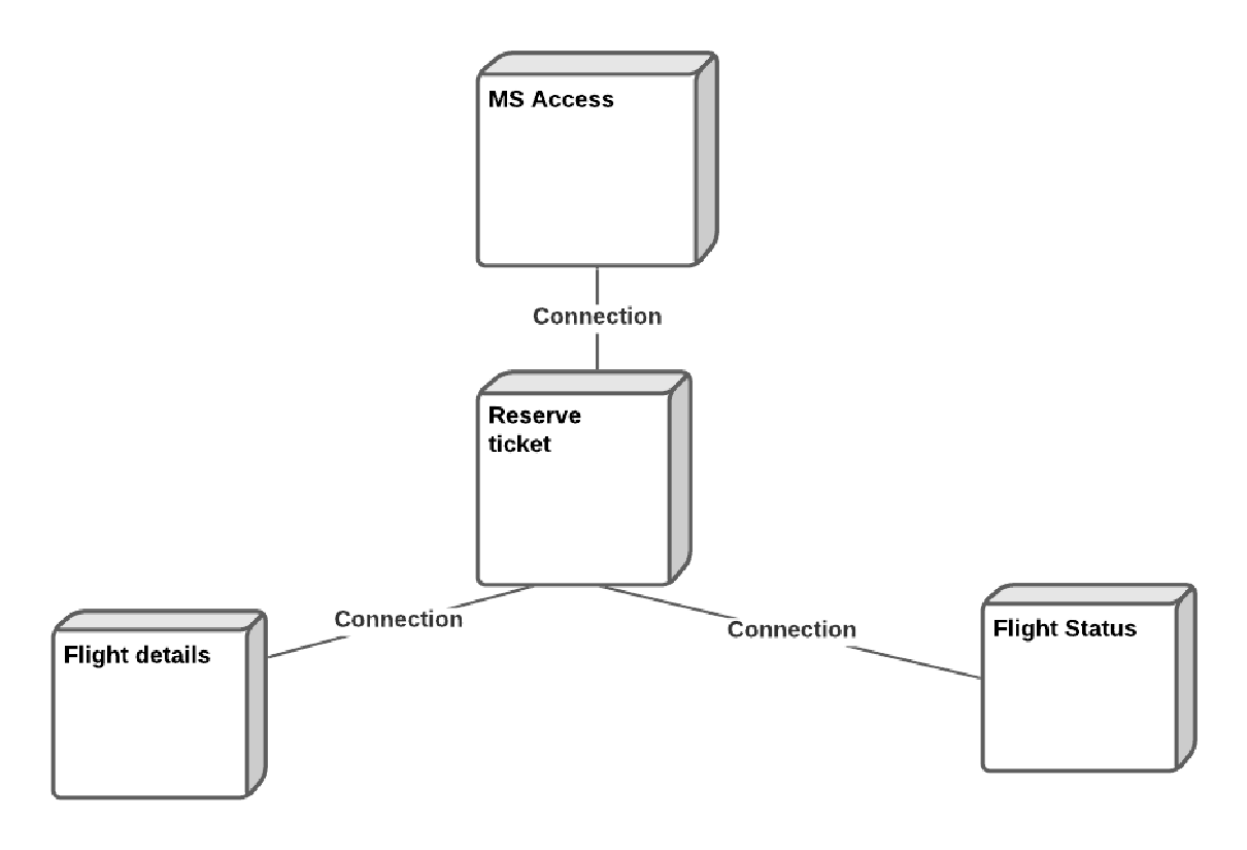


Collaboration diagrams are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaboration diagrams are used to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces.

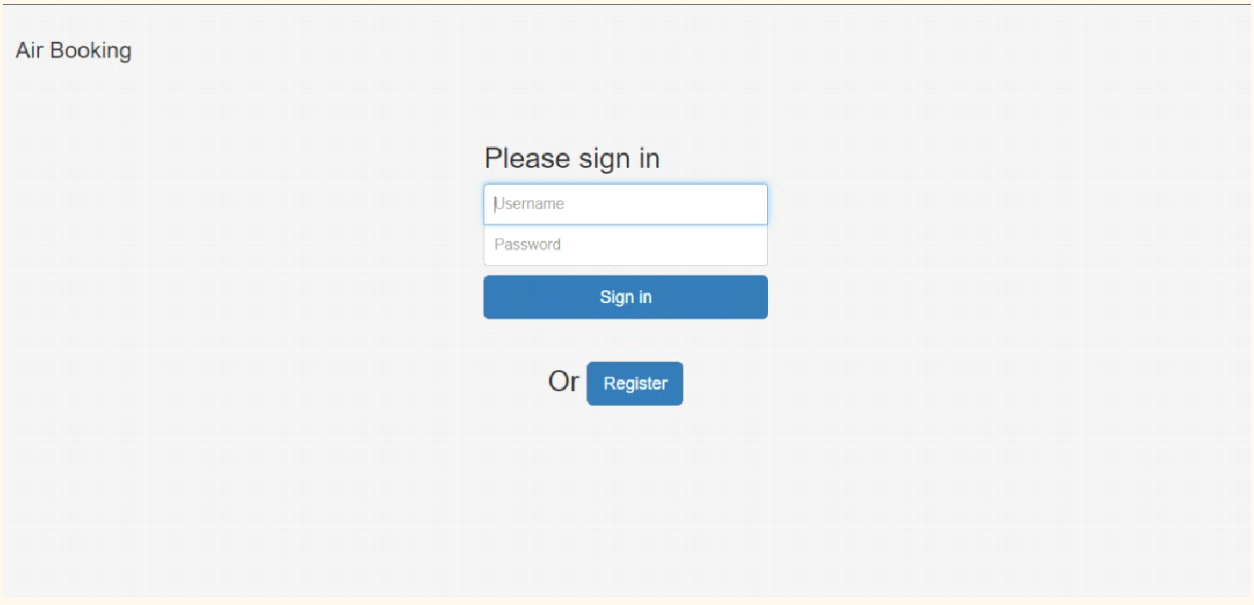
SAMPLES

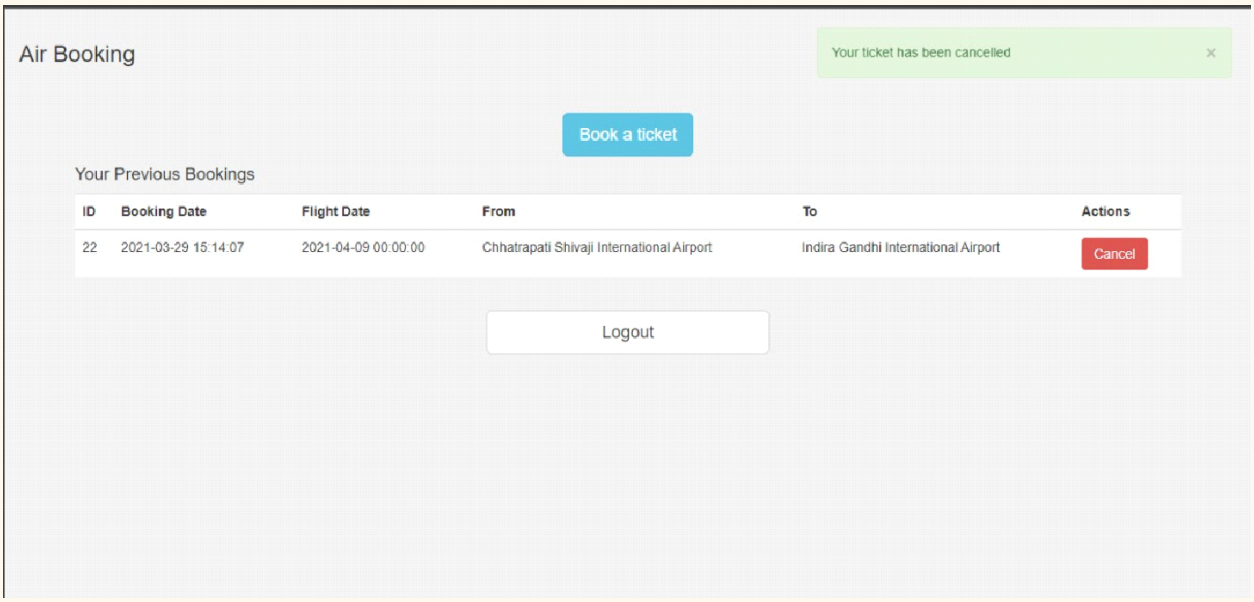
Deployment Diagram

Deployment diagrams are important for visualizing, specifying, and documenting embedded, client/server, and distributed systems and also for managing executable systems through forward and reverse engineering. It’s a special kind of class diagram, which focuses on a system's nodes.



Sample Frontend Design (UI/UX)





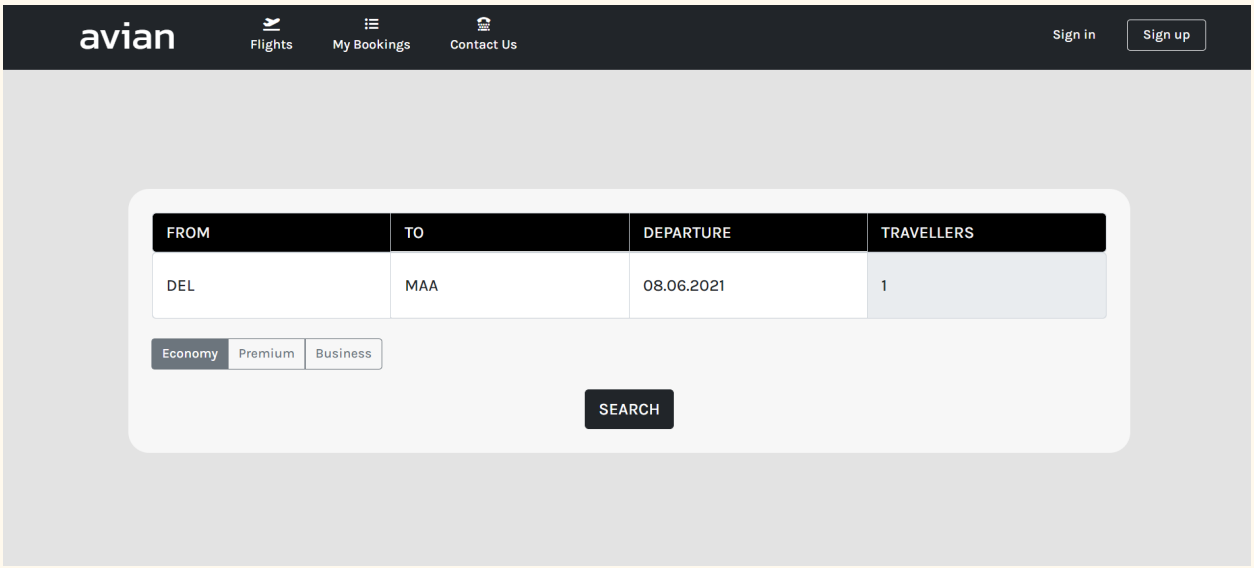
CODING MODULE 1:

```
<% layout("layouts/boilerplate") %>


<%
    let date = new Date();
    date.setDate(date.getDate() + 1);
    date = date.toLocaleDateString("en-GB", {month: "2-digit",day: "2-digit",year: "numeric"}).split("/").join(".")
%>


<form action="/" method="post">
    <div>
        <div class="center">
            <div class="input-group input-group-lg">
                <label for="from" class="form-control heading-label">FROM</label>
                <label for="to" class="form-control heading-label">TO</label>
                <label for="date" class="form-control heading-label">DEPARTURE</label>
                <label for="passengerCount" class="form-control heading-label">TRAVELLERS</label>
            </div>
            <div class="input-group input-group-lg">
                <input list="airports" type="text" aria-label="from" class="form-control index-input" value="DEL" name="from" autocomplete="false" required>
                <input list="airports" type="text" aria-label="to" class="form-control index-input" value="MAA" name="to" autocomplete="false" required>
```

```
                <datalist id="airports">
                    <% for (let airport of airports) { %>
                        <option value=<%= airport.IATA_code %>><%=
airport.city_name %></option>
                    <% } %>
                </datalist>
                <input type="text" id="datepicker" aria-label="date"
class="form-control index-input"  name="date" value="<%= date %>"
autocomplete="false" required>
                <select class="form-control" name="passengerCount"
required>
                    <option value=1>1</option>
                    <option value=2>2</option>
                    <option value=3>3</option>
                    <option value=4>4</option>
                    <option value=5>5</option>
                    <option value=6>6</option>
                </select>
            </div>
            <div class="bt">
                <div class="btn-group" role="group" aria-label="Basic
radio toggle button group">
                    <input type="radio" class="btn-check" name="group"
id="group1" autocomplete="off" value="Economy" checked>
                    <label class="btn btn-outline-secondary"
for="group1">Economy</label>

                    <input type="radio" class="btn-check" name="group"
id="group2" autocomplete="off" value="Premium">
```

```
                    <label class="btn btn-outline-secondary"
for="group2">Premium</label>



                    <input type="radio" class="btn-check" name="group"
id="group3" autocomplete="off" value="Business">

                    <label class="btn btn-outline-secondary"
for="group3">Business</label>

                </div>

                <button type="submit" class="btn btn-dark btn-lg
button">SEARCH</button>

            </div>

        </div>

    </div>

</form>


<script src="/scripts/datePicker.js"></script>
```

OUTPUT:



CODING MODULE 2:

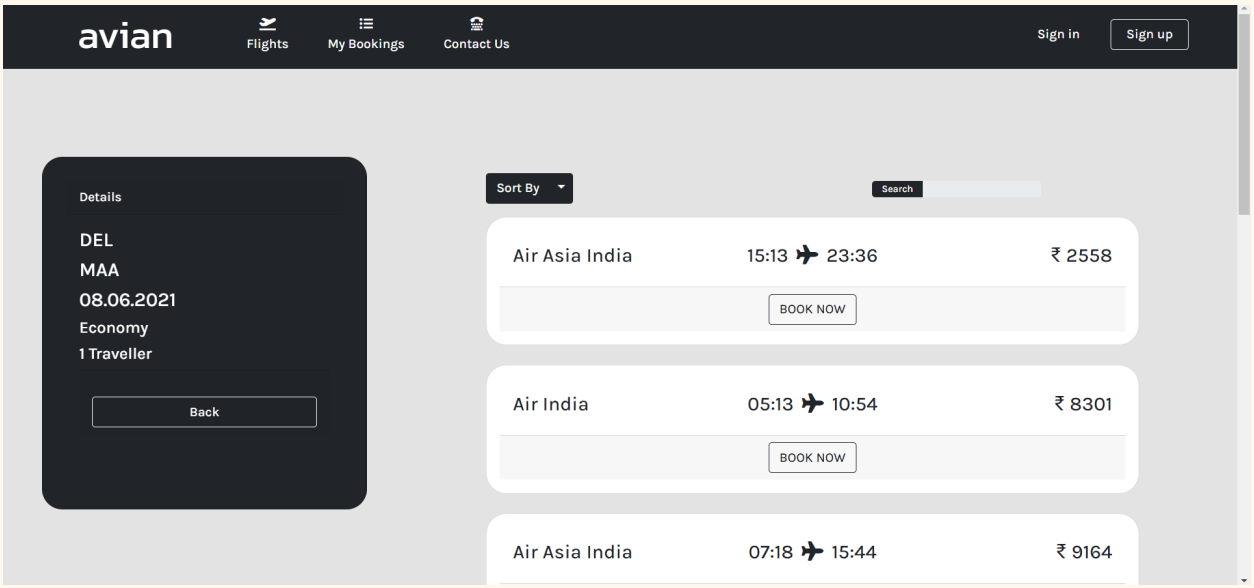
```
<% layout("layouts/boilerplate") %>


<div>

    <div class="row">

        <div class="col-6 col-md-4">

            <div class="card text-white bg-dark left-card" style="max-
width: 25rem;">

                <div class="card-header">Details</div>

                <div class="card-body">

                    <h4 class="card-title"><%= details[0].from %></h4>

                    <h4 class="card-title"><%= details[0].to %></h4>

                    <h4 class="card-title"><%= details[0].date %></h4>

                    <h5 class="card-title"><%= details[0].group %></h5>

                    <% if (details[0].passengerCount > 1) { %>

                        <h5 class="card-title"><%=
details[0].passengerCount %> Travellers</h5>

                    <% } else { %>

                        <h5 class="card-title"><%=
details[0].passengerCount %> Traveller</h5>

                    <% } %>

                    <div class="card-footer text-muted text-center">

                        <a href="/" class="btn btn-outline-light
button">Back</a>

                    </div>

                </div>
```

```
            </div>

        </div>



        <div class="col-12 col-md-8">



            <div class="above-search">

                <div class="btn-group">

                    <button type="button" class="btn btn-dark">Sort By</
button>

                    <button type="button" class="btn btn-dark dropdown-
toggle dropdown-toggle-split" data-bs-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">

                    <span class="sr-only">Toggle Dropdown</span>

                    </button>

                    <ul class="dropdown-menu">

                    <li><a class="dropdown-item" href="/search?
sortby=ao">Alphabetical Order</a></li>

                    <li><a class="dropdown-item" href="/search?
sortby=duration">Duration</a></li>

                    <li><a class="dropdown-item" href="/search?
sortby=price">Price</a></li>

                    </ul>

                </div>

                <div class="progress">

                    <div class="progress-bar bg-dark" role="progressbar"
style="width: 30%;" aria-valuenow="25" aria-valuemin="0" aria-
valuemax="100">Search</div>

                </div>

            </div>
```

```
            <% for (let detail of details) { %>
                <form action="/search" method="post">
                    <div class="card right-card">
                        <div class="card-body">
                            <h4 class="card-title">
                                <div class="row">
                                    <div class="col">
                                        <%= detail.airline %>
                                    </div>
                                    <div class="col text-center">
                                        <%= detail.fromTime %><span
class="fas fa-plane plane"></span><%= detail.toTime %>
                                    </div>
                                    <div class="col text-right">
                                        ₹ <%= detail.price - 489 %>
                                    </div>
                                </div>
                            </h4>
                        </div>
                        <input type="hidden" name="detail" value="<%=
JSON.stringify(detail) %>" />
                        <div class="card-footer text-muted text-center">
                            <button type="submit" class="btn btn-outline-
dark">BOOK NOW</button>
                        </div>
                    </div>
```

```
                </form>

                <br>

            <% } %>



        </div>

    </div>

</div>


<script src="/scripts/stable.js"></script>
```

OUTPUT:



CODING MODULE 3:

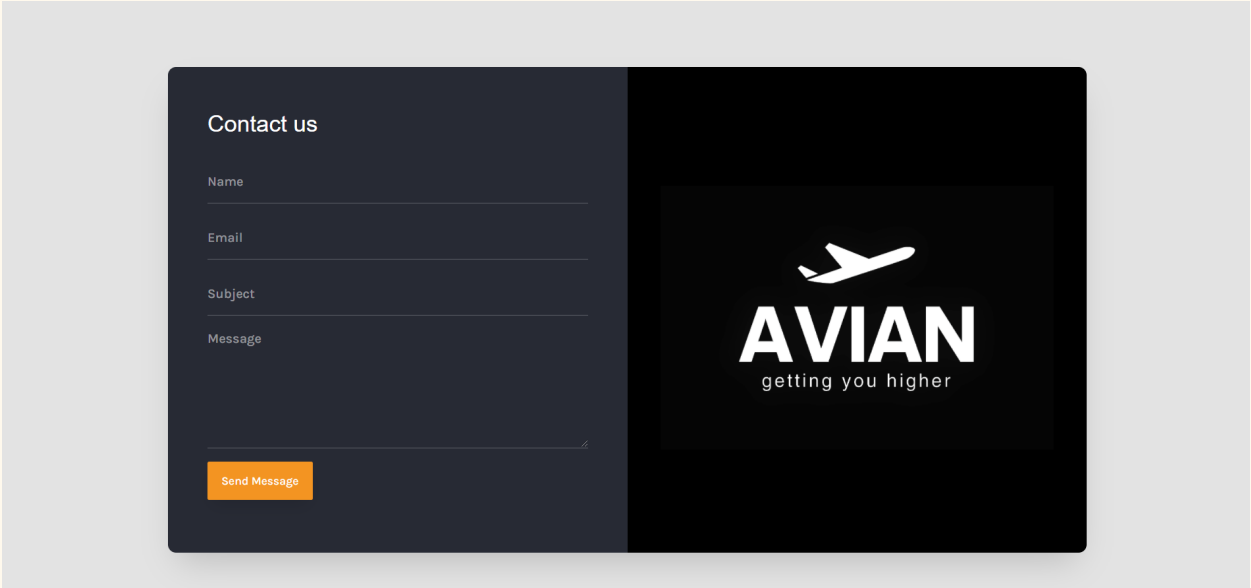
```
<!DOCTYPE html>

<html>

<head>

    <!-- Required meta tags -->

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">



    <!-- Bootstrap CSS -->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/
bootstrap.min.css" rel="stylesheet" integrity="sha384-
wEmeIV1mKuiNpC+IOBjI7aAzPcEZeedi5yW5f2yOq55WWLwNGmvvx4Um1vskeMj0"
crossorigin="anonymous">



    <!-- Google Fonts -->

    <link rel="preconnect" href="https://fonts.gstatic.com">

    <link href="https://fonts.googleapis.com/css2?
family=Karla:wght@300;500&family=Ubuntu&family=Kanit&display=swap"
rel="stylesheet">



    <!-- Icons (Font Awesome) -->

    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/
5.15.3/css/all.min.css" rel="stylesheet">

    <link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/
font-awesome.min.css" rel="stylesheet">
```

```
    <!-- Favicon -->
    <link rel="icon" type="image/png" sizes="180x180" href="/images/
plane.png">

    <!-- CSS -->
    <link rel="stylesheet" type="text/css" href="/stylesheets/contact-
styles.css">

    <title>Avian</title>
</head>
<body>
    <%- include("../partials/navbar")%>

    <%- include("../partials/flash") %>

    <section class="ftco-section">
        <div class="container">
            <div class="row justify-content-center">
                <div class="col-lg-16">
                    <div class="wrapper">
                        <div class="row no-gutters">
                            <div class="col-md-6 d-flex align-items-
stretch">
                                <div class="contact-wrap w-100 p-md-5 p-4
py-5">
                                    <h3 class="mb-4">Contact us</h3>
                                    <div id="form-message-warning"
class="mb-4"></div>
```

```
                                    <form action="/contact" method="post"
id="contactForm" name="contactForm" class="contactForm">

                                        <div class="row">

                                            <div class="col-md-12">

                                                <div class="form-group">

                                                    <input type="text"
class="form-control" name="name" id="name" placeholder="Name" required>

                                                </div>

                                            </div>

                                            <div class="col-md-12">

                                                <div class="form-group">

                                                    <input type="email"
class="form-control" name="email" id="email" placeholder="Email" required>

                                                </div>

                                            </div>

                                            <div class="col-md-12">

                                                <div class="form-group">

                                                    <input type="text"
class="form-control" name="subject" id="subject" placeholder="Subject"
required>

                                                </div>

                                            </div>

                                            <div class="col-md-12">

                                                <div class="form-group">

                                                    <textarea
name="message" class="form-control" id="message" cols="30" rows="6"
placeholder="Message" required></textarea>

                                                </div>

                                            </div>
```

```
                                        <div class="col-md-12">
                                            <div class="form-group">
                                                <input type="submit"
value="Send Message" class="btn btn-primary">
                                                <div
class="submitting"></div>
                                            </div>
                                        </div>
                                    </form>
                                </div>
                            </div>
                            <div class="col-md-6 d-flex pic"
style="background-color: #000000; justify-content: center;">
                                <img src="/images/avian-white.png" alt=""
style="max-width: 30rem; max-height: 20rem; margin: 9rem 0 0 0;">
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>


    <%- include("../partials/footer")%>
</body>
</html>
```

OUTPUT:



NEW REQUIREMENTS/NEW CHANGES:

LOGIN PAGE:

S.NO	TEST CASE	RESULT	TEST
1.	On the click of login button	The user should be logged into the site if the username and password match. Else the user should be denied access.	successful

2.	**On the click of register button**	**The user should be taken to the registration page.**	**successful**

REGISTRAION PAGE:

S.NO	TEST CASE	RESULT	TEST
1.	**On the click of register button**	**The user along with all the other user information should be added to the database.**	**successful**

MAIN PAGE:

S.NO	TEST CASE	RESULT	TEST

1.	**On the click of book tickets button.**	**The user should be taken to ticket booking page.**	**successful**
2.	**On the click of cancel ticket button**	**The ticket booked by the user should be cancelled.**	**successful**
3.	**On the click of logout button**	**The user should be logged out of the site and the session should be destroyed.**	**successful**

BOOKING PAGE:

S.NO	TEST CASE	RESULT	TEST

1.	**On the click of book button**	**The user's ticket should be booked.**	**successful**
2.	**On the click of cancel button**	**The user should be taken back to the main page.**	**successful**

CFG, CYCLOMATIC COMPLEXITY & MTR:

1. Executive Summary

This program describes the Airline booking management Test Cases. Testing provides a system that uses a variety of test results. Testing helps in predicting the any future errors or the debugs in the program. Then the errors are detected and corrected using test steps and corrections are made for future reference. Thus, a sequence of checks is carried out on the software before being applied.

2. Test Plan

2.1. Scope of Testing

The aim of the system testing process was to determine all defects in our project .The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing

2. Integration testing

2.2. Types of Testing , Methodology , Tools

TESTING	METHODOLOGY	TOOLS REQUIRED
UNIT TESTING		
Login	User can login using his or her credentials.	SQL,PHP/ NODE.JS,JAVASCRIPT,HTML
Simplifies Integration	Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.	SQL
INTEGRATION TESTING		

Integration testing	In this type of testing we test various integration of the project module by providing the input .The primary objective is to test the module interfaces in order to ensure that no errors are occurring when one module invokes the other module.	PHP
Booking	Check if the booking page works properly and books the ticket as requested by the user.	SQL,PHP/ NODE.JS,JAVASCRIPT,HTML

2.3. Test Deliverables

Test Deliverables are the test artifacts which are given to the stakeholders of a software project during the SDLC (Software Development Life Cycle). A software project which follows SDLC undergoes the different phases before delivering to the customer. In this process, there will be some deliverables in every phase. Some of the deliverables are provided before the testing phase commences and some are provided during the testing phase and rest after the testing phase is completed. Airline booking management system is a comprehensive internet system to manage the airline bookings made by users. Users can search for flights and book tickets. These bookings will be managed using SQL.

Test Case

Functional Test Cases

LOGIN PAGE:

S.NO	TEST CASE	RESULT	TEST
1.	On the click of login button	The user should be logged into the site if the username and password match. Else the user should be denied access.	successful
2.	On the click of register button	The user should be taken to the registration page.	successful

REGISTRAION PAGE:

S.NO	TEST CASE	RESULT	TEST
1.	On the click of register button	The user along with all the other user information should be added to the database.	successful

MAIN PAGE:

S.NO	TEST CASE	RESULT	TEST
1.	On the click of book tickets button.	The user should be taken to ticket booking page.	successful
2.	On the click of cancel ticket button	The ticket booked by the user should be cancelled.	successful
3.	On the click of logout button	The user should be logged out of the site and session should be destroyed.	successful

BOOKING PAGE:

S.NO	TEST CASE	RESULT	TEST
1.	On the click of book button	The user's ticket should be booked.	successful
2.	On the click of cancel button	The user should be taken back to the main page.	successful

4. Defect Log

Requirement	Defect Description	RESULT	Status
To successfully login user if the credentials are correct	TC04	User successfully logged in	PASS
To register a user by adding the user to the database along with user info	TC02	New user registered and added to the database.	PASS
To book user requested ticket	TC03	The user requested ticket was booked	PASS

CONCLUSION:

Airline booking system allows the user to view and book flights from a given source to destination. This software package allows storing the details of all the data related to the users and their bookings. The system is strong enough to withstand regressive yearly operations under conditions where the database is maintained and cleared over a certain time of span. The system has been scaled to withstand heavy client side traffic and is a scalable application if further scaling is required.