KARNATAKA LAW SOCIETY'S

# GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELGAUM-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belgaum)

**(APPROVED BY AICTE, NEW DELHI)**



*Course Activity Report*

**TOPIC: A Travel & Tourism Management System**

*Submitted in the partial fulfillment for the academic requirement of*

***5th Semester B.E.***

***in***

***Software Design & Modelling Lab***

***Submitted by:***

| NAME | USN |
|---|---|
| CHIDAMBAR INAMDAR | 2GI19CS035 |
| DEEPAK KUMBLE | 2GI19CS038 |
| GOURAV SHANBHAG | 2GI19CS046 |
| IRFAN KAMATE | 2GI19CS052 |

## GUIDE:

**Prof. Gajendra Deshpande**

**2021-2022**

Karnataka Law Society's

## GOGTE INSTITUTE OF TECHNOLOGY
Udyambag, Belagavi -590008 Karnataka, India.

**Department of Computer Science and Engineering**

# Certificate

This is to certify that the Course Project work titled **"A Travel & Tourism Management System "** carried out by students **Chidambar Inamdar, Deepak Kumble, Gourav Shanbhag, Irfan Kamate** bearing **USNs: 2GI19CS035 , 2GI19CS038 , 2GI19CS046 , 2GI19CS052** is submitted in partial fulfilment of the requirements for 5$^{th}$ semester B.E. in **COMPUTER SCIENCE AND ENGINEERING,** Visvesvaraya Technological University, Belagavi. It is certified that all corrections/ suggestions indicated have been incorporated in the report. The course project report has been approved as it satisfies the academic requirements prescribed for the said degree.

Date:  29/12/2021                                                  Signature of Guide

Place: Belagavi                                                  Mr. Gajendra Deshpande
                                                                        Asst. Prof., Dept. of CSE,

                                                                    KLS Gogte Institute of Technology,
                                                                                    Belagavi

Name of the Examiners                                    Signature of the Examiners

1._____                          1. _____

2._____                          2. _____

# PROBLEM STATEMENT

To create a website that is an automated travel agent system for processing bus reservations. The website will intermediate between the traveller/customer and the travel agency computing system, which then interfaces with commercial bus reservation services. The proposed system will also assist in booking and changing bus reservations.

# OBJECTIVES

- The objective of the project is to develop a system that automates the processes and activities of a travel and tourism agency.

- The purpose is to design a system using which one can perform all operations related to traveling and sight-seeing.

- With the increasing trend of online activities, businesses must shift their business operations to online platforms.

# INTRODUCTION

In today's day and age people often need a break from their busy lives to spend some time with family and friends or simply travel for their own purpose but never get satisfactory transport facilities and destination brochure. This motivated us to create a tourism management system which helps the travellers experience a better and satisfactory journey from the start to end. The Tours and Travel Management System is a web based application. The main purpose of "Tours and travels management system" is to provide a convenient way for a customer to book train and bus for tour purposes.

The objective of this project is to develop a system that automates the processes and activities of a travel agency. In this project, we will make an easier task of searching places and for booking train or bus. In the present system a customer has to approach various agencies to find details of places and to book tickets.

This often requires a lot of time and effort. We provide approach skills to critically examine how a tourist visits and its ability to operate in an appropriate way when dealing with the consequences of tourism, locally, regionally, and nationally including visitor security and ecological influences.

It is tedious for a customer to plan a particular journey and have it executed properly. The project 'Tours and Travels Management System' is developed to replace the currently existing system, which helps in keeping records of the customer details of destination as well as payment received.

**Existing System:**

- In the present system a customer has to approach various agencies to find details of places and to book tickets.
- This often requires a lot of time and effort.
- A customer may not get the desired information from these offices and often the customer may be misguided.
- It is tedious for a customer to plan a particular journey and have it executed properly.

**Drawbacks of the Existing System:**

- Increased transaction leads to increased source document and hence maintenance becomes difficult.
- If any admin, user entry is wrongly made then the maintenance becomes very difficult.

**Proposed System:**

- The proposed system is a web based application and maintains a centralized repository of all related information.
- The system allows one to easily access the relevant information and make necessary travel arrangements.
- Users can decide about places they want to visit and make bookings online for travel and accommodation.

**Advantages of the Proposed System:**

- Gives accurate information at all times

- Simplifies the manual work

- It minimizes the documentation related work

- Provides up to date information

- Provides a user-Friendly Environment

- All details can be provided in the single page to the traveller at the end

- Booking confirmation is also provided.

# FUNCTIONAL REQUIREMENTS

A Functional Requirement is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behaviour, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements in Software Engineering are also called Functional Specification. The functional requirements for our software are:

1. To register new users

2. Login for existing users

3. Information about bus and it's fare

4. Making, changing and cancelling traveler's booking

5. Selection of payment mode and generation of tickets

6. Generating travel itineraries.

7. The client must register for create the account and login using username and password to use function in the application

8. Collecting client's personal information (name, address, number, age)

9. The client can view or find the information of tourist attraction.

10. The client can search for booking bus after select date time to go.

# NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are necessary for products and directly affect acceptance of the product. They make products attractive, easy to use, fast, reliable and secure. The existence of such property is not because they are the basic functions of the product, but because customers want these functional activities to achieve a particular quality. Non-functional requirements describe product features such as user experience, products appearance or usability.

## Performance Requirements:

As for web-based information systems, performance requirements can help designers and developers to define system behaviour under different loads. Performance is often an important criterion to measure usability of the system. The most important thing of user experience is often relevant to performance, including waiting time, throughput, and stability and so on. Travel management system should meet the following performance requirements:

## 1. Response Time:

There has always been a variety of definitions for response time of the performance requirements. A more common definition counts from when the user sends a request to when the last byte reaches the user's end. Page response time means the time required to load a page. The time unit is normally second or millisecond. Average page response time of the system should be within 3 seconds.

## 2. System Capacity

Sufficient database space is required for storing information related to all the users of the application.

## 3. Stability

Stability needs a long time to reflect and is used to measure declining rate of the figure of performance after long-running. System with poor stability may result in memory leak or the database transaction log overflowing. Stability requirements of Travel Management system are moderate with respect to other systems

### Usability Requirements

Usability requirements make the products meet the needs of the user's ability and expectation to use. Usability will influence efficiency and error rate of using the product as well as acceptance of new products. Sources of usability requirements result from the following two factors: one is the required usability level of the product, and the other is user's expectations

### 1. Easy to use

Travel system should be easy enough for any ordinary user who, if has no professional background, can achieve necessary function through the information provided on page. Users can complete an objective without excessive page jump actions. Because too many clicks and jumps easily lead to confusion, which makes the user tend to forget the initial purpose. Users should clearly know what can be accomplished through this system, such as browsing the travel notes, rating scenic spots and so on. All functions are located in the appropriate place of each page to ensure that users can see and hold an overview of the whole function.

### 2. Easy to learn

The biggest difference between easy to learn and easy to use lies in that learning to use the system is difficult in the beginning, but once master it, using the system will become very easy. In the initial adaptation phase, users can learn how to use all functions according to suggestion and help in a short time.

### 3. Low Error Rate

Travel management system is required to help users minimize the number of errors. Important factors involve providing users interface and context which are consistent with daily habits and clear navigation, etc. Help information is also very critical and should be provided in appropriate place. When error occurs, the system can correct the error or return to the previous step, which also means good performance of the system robustness.

### 4. Efficiency

Efficiency requirement of the system can be measured by the time taken to complete a task or the average number of tasks completed in unit time by users. Efficient user experience will directly increase customer satisfaction. Efficiency of the system can be divided into two parts: efficiency of the user-controllable and the efficiency of the system itself. Reasonable interface layout, buttons, and logical hierarchy enable high operating efficiency for users; background processing and connectivity speed of database will affect the efficiency of the system itself.

### 5. High User Satisfaction

User satisfaction, a more or less abstract measure, means that users enjoy using the system and fell satisfied with the whole process. As for Travel-SYS system, the number of registered users can be regarded as a measure of satisfaction. If users are willing to use the system as a travel information search platform and share their experiences with others on this platform, user satisfaction will prove to be high.

### 6. Security Requirement

Security requirement is a critical requirement, for big risks may be brought by products that fail to meet security requirement. Security of system information tends to be even more important nowadays when the Internet wins wide popularity.

# SOFTWARE REQUIREMENTS

The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view.

The requirement for a system are the description of what the system should do, the service or services that it provides and the constraints on its operation.

- Operating System: Windows

- Frontend/Web-Technologies: HTML, CSS, JavaScript

- Backend/Database: Firebase

# HARDWARE REQUIREMENTS

Computer hardware specifications are technical descriptions of the computer's components and capabilities. Processor speed, model and manufacturer. Processor speed is typically indicated in gigahertz (GHz). The higher the number, the faster the computer.

Random Access Memory (RAM), this is typically indicated in gigabytes (GB). The more RAM in a computer the more it can do simultaneously.

Hard disk (sometimes called ROM) space. This is typically indicated in gigabytes (GB) and refers generally to the amount of information (like documents, music and other data) your computer can hold.

Other specifications might include network (Ethernet or Wi-Fi) adapters or audio and video capabilities.

Hardware requirements of our project are as follows-

- Processor: Pentium Minimum 1 GHz; Recommended 2GHz or more
- Memory (RAM): Minimum 1 GB; Recommended 2 GB or above
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more
- Display type: Standard VGA or SVGA card

# USE-CASE DIAGRAM
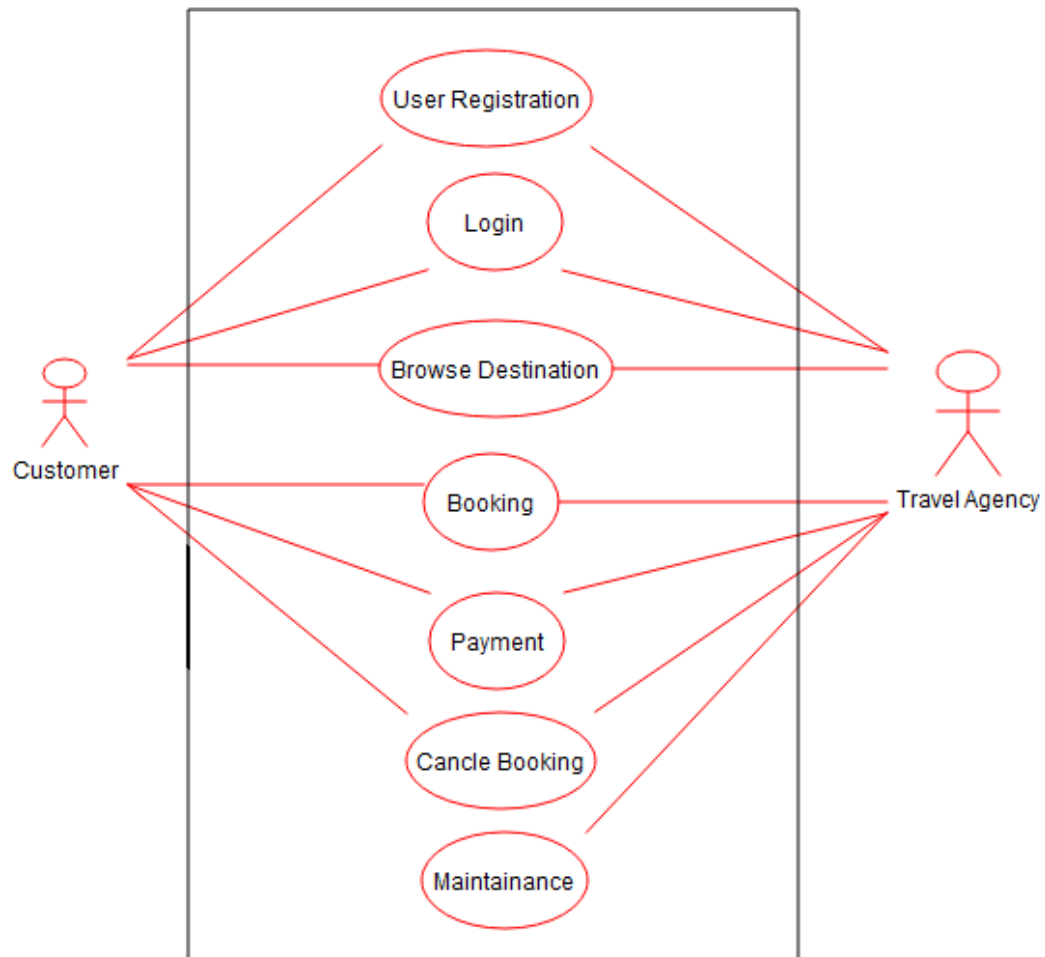
# USE CASE DIAGRAM DESCRIPTION

## Registration

| Use Case Name: | User Registration | |
|---|---|---|
| Scenario: | User (Admin, Customer) registers to the system. | |
| Brief Description | Allow a customer to create a new account if they don't have one already | |
| Actor: | User (Admin, Customer) | |
| Pre-Condition: | Customer must provide valid/legit details | |
| Post-Condition: | Popup message saying Successfully Registered. | |
| Flow of events: | User (Admin, Customer) | System |
| | <ul><li>Must provide a valid email.</li><li>Must provide a valid username & password.</li></ul> | <ul><li>Confirm email.</li><li>Confirm username.</li><li>Confirm password.</li></ul> |
| Exception | If invalid details are provided, popup message saying invalid details. | |

**Login**

| Use Case Name: | Login | |
|---|---|---|
| Scenario: | User (Admin, Customer) can login to the system. | |
| Brief Description | Customer and Admin should login with valid username and password. | |
| Actor: | User (Admin, Customer) | |
| Pre-Condition: | Must be valid username and password type. | |
| Post-Condition: | Popup message saying Login Successful! | |
| Flow of events: | User (Admin, Customer) | System |
| | <ul><li>Must provide registered username.</li><li>Must provide registered password.</li></ul> | <ul><li>Confirm username.</li><li>Confirm password.</li></ul> |
| Exception | If invalid username or password, popup message saying invalid credentials. | |

**Browse Destinations**

| Use Case Name: | Browse Destination | |
|---|---|---|
| Scenario: | Logged User (Customer) can browse/view various destinations. | |
| Brief Description | User can decide which place to visit based on the various destinations provided by the agency. | |
| Actor: | User (Customer) | |
| Pre-Condition: | User must be logged in. | |
| Post-Condition: | Proceed to payment. | |
| Flow of events: | User (Admin) | System |
| | • User must log in.<br>• User may/may not select a destination after looking at them. | • Confirm whether user has logged in.<br><br>• Confirm the selected destination and proceed to payment page. |
| Exception | Nil. | |

**Booking**

| Use Case Name: | Booking | |
|---|---|---|
| Scenario: | Customers can book their tickets | |
| Brief Description | Users can view the destination and book their tickets | |
| Actor: | User (Customer) | |
| Pre-Condition: | A destination must be selected along with a date | |
| Post-Condition: | Booking confirmation | |
| Flow of events: | User (Customer) | System |
| | <ul><li>Login</li><li>Select details</li><li>Book the tickets</li></ul> | <ul><li>Check whether user is logged in</li><li>Check all details and availability of tickets</li></ul> |
| Exception | If any error occurs during entering information, an appropriate error message is displayed. | |

**Payment**

| | |
|---|---|
| Use Case Name: | Payment |
| Scenario: | User (Customer) should pay for their tickets and should handle their transactions |
| Brief Description | Customers should make the payment and the payment service should confirm and generate transaction id and send it to the server for confirmation. |
| Actor: | User (Customer), Payment service |
| Pre-Condition: | Confirm the destination and date of departure and arrival |
| Post-Condition: | Generate Bill after successful payment |
| Flow of events: | **User (Customer), Payment Service**<br><br>• Confirm destination and date<br>• Make Payment<br>• Generate the bill<br><br>**System**<br><br>• Check transaction id<br>• Send the Bill to customer |
| Exception | Destination(s) fully booked. |

## Cancel Booking

| Use Case Name: | Cancel Booking | |
|---|---|---|
| Scenario: | Users (Customer) can cancel the tickets that they had booked previously | |
| Brief Description | Customers can cancel their tickets due to personal reasons or a change of mind. | |
| Actor: | User (Customer) | |
| Pre-Condition: | A ticket must have been booked. | |
| Post-Condition: | Ticket Cancelled. | |
| Flow of events: | Users (Customer), PayPal | System |
| | • Login<br>• Provide transaction id<br>• Cancel ticket | • Check whether user is logged in<br>• Check the transaction id.<br>• If all details are correct, cancel the booking and send confirmation to the customer |
| Exception | If transaction is doesn't match failed, display message "Error" | |

**Maintenance**

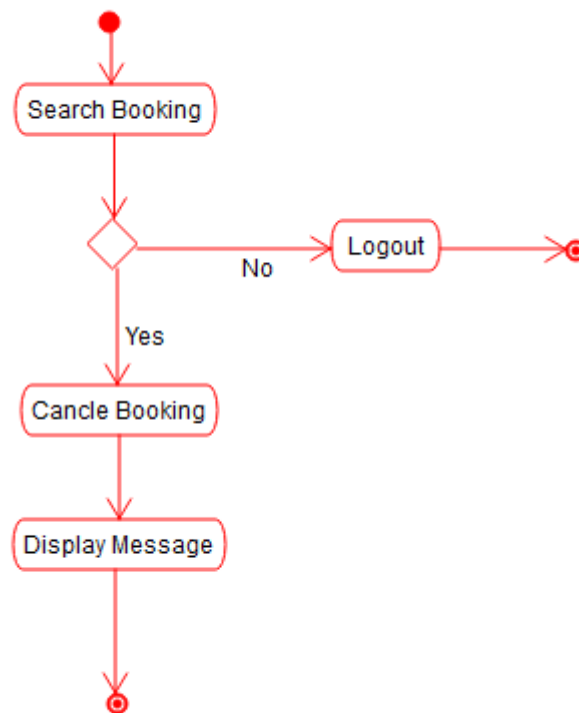| Use Case Name: | Maintenance | |
| --- | --- | --- |
| Scenario: | Users (Admin) can maintain the website and can also make minor changes if necessary. | |
| Brief Description | Admin can add, remove and edit travel destinations provided by the agency | |
| Actor: | User (Admin) | |
| Pre-Condition: | Admin must be logged in | |
| Post-Condition: | Confirm the changes made by the admin. | |
| Flow of events: | Users (Admin) | System |
| | <ul><li>User must login to manage the website</li><li>User can add, remove and edit travel destinations</li></ul> | <ul><li>Confirm whether user has logged in</li><li>Store the modified products in the database.</li></ul> |
| Exception | Trying to add/remove an already existing destination. | |

# SEQUENCE DIAGRAM FOR LOGIN
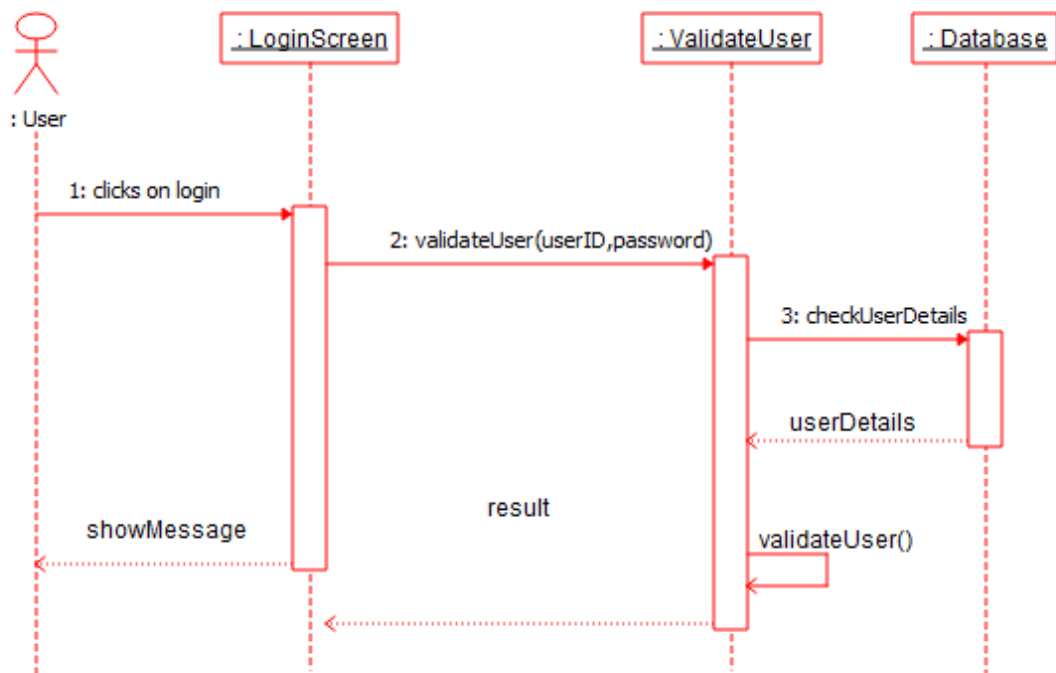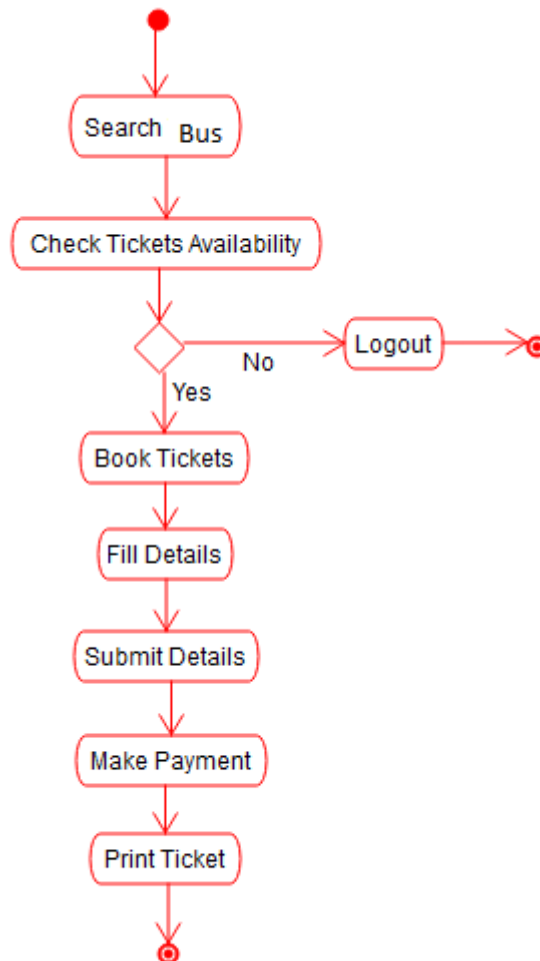
# SEQUENCE DIAGRAM FOR REGISTER
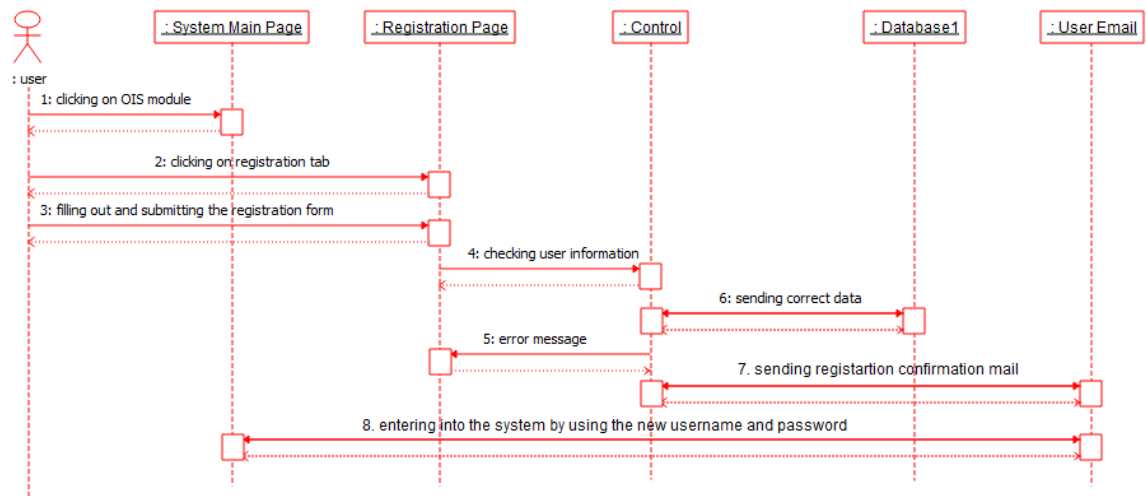
# SEQUENCE DIAGRAM FOR BROWSING DESTINATIONS
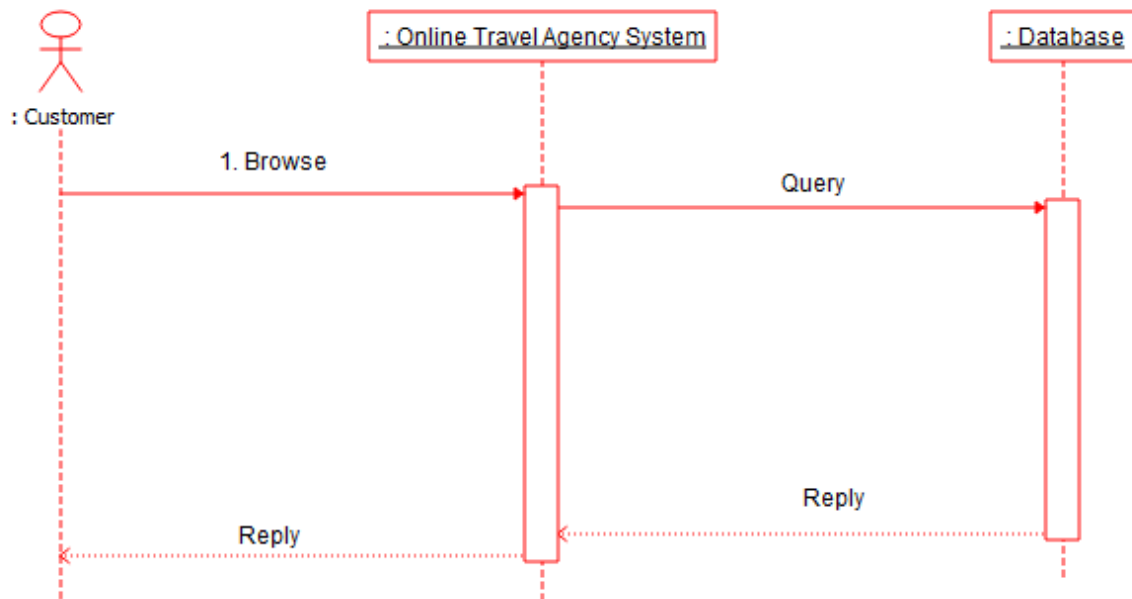
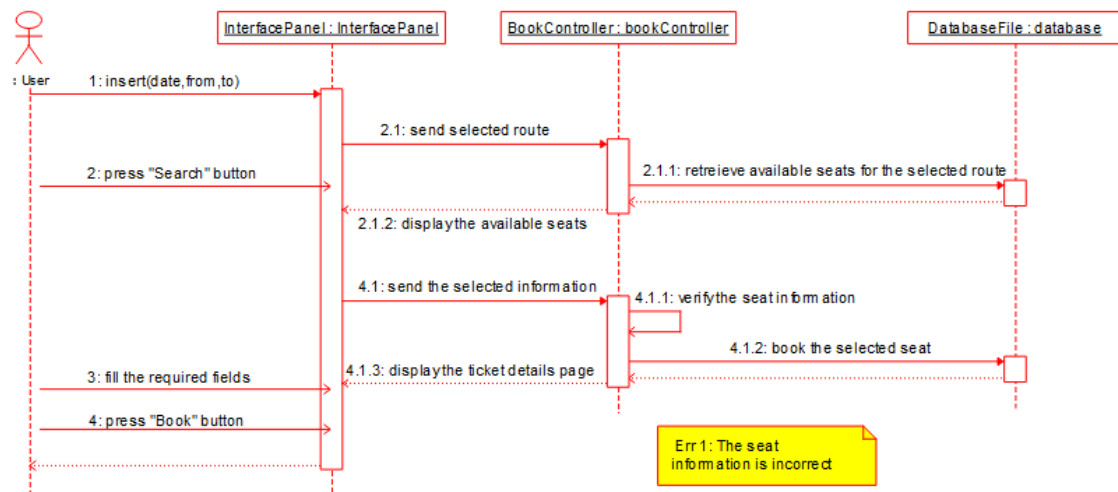# ACTIVITY DIAGRAM FOR CANCEL BOOKING

# ACTIVITY DIAGRAM FOR PAYMENT

# SEQUENCE DIAGRAM FOR BOOKING

# Class Diagram

**Permission Class**

+permission_id: int
+permission_role_id: int
+permission_title: string
+permission_module: string
+permission_description: string

+addPermission()
+editPermission()
+deletePermission()
+searchPermission()

**Customer Class**

+customer_id: int
+customer_name: string
+customer_mobile: string
+customer_email: string
+customer_address: string
+customer_username: string
+customer_password: string

+addCustomer()
+editCustomer()
+deleteCustomer()
+searchCustomer()

**User Class**

+user_id: int
+user_role_id: int
+user_name: string
+user_email: string
+user_dob: date
+user_address: string

+addUser()
+editUser()
+deleteUser()
+searchUser()

**Booking Class**

+booking_id: int
+booking_type: string
+booking_description: string
+booking_title: string
+booking_hotel_id: int
+booking_date: date

+addBooking()
+editBooking()
+deleteBooking()
+searchBooking()

**Payment Class**

+payment_id: int
+payment_amount: string
+payment_date: date
+payment_description: string
+payment_customer_id: int

+addPayment()
+editPayment()
+deletePayment()
+searchPayment()

**Bus Class**

+bus_id: int
+bus_name: string
+bus_description: string
+bus_type: string

+addBus()
+editBus()
+deleteBus()
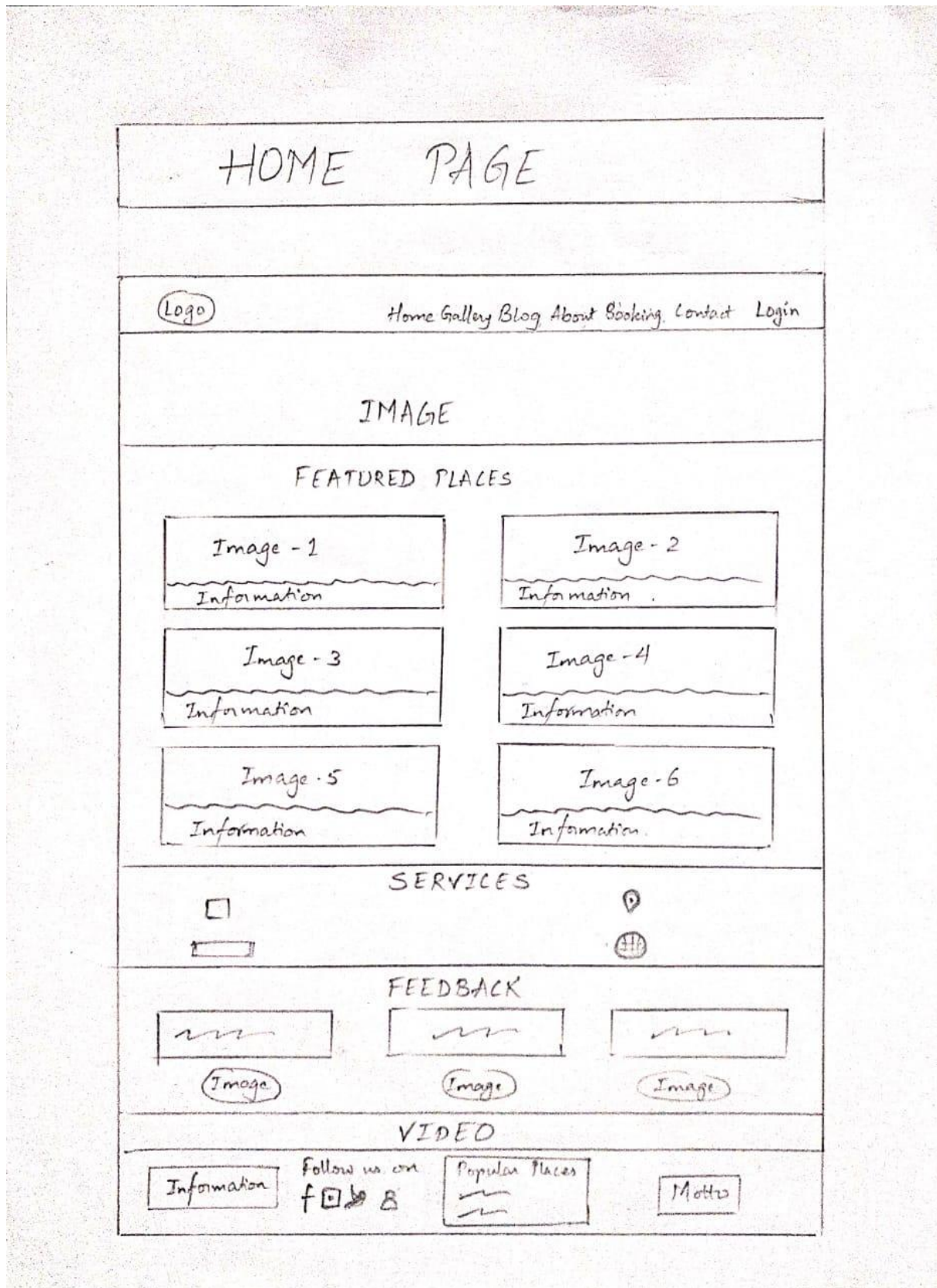+searchBus()

# ER - DIAGRAM

# UI MODEL

User interface modelling is a development technique used by computer application programmers. Today's user interfaces (UIs) are complex software components, which play an essential role in the usability of an application. The development of UIs requires therefore, not only guidelines and best practice reports, but also a development process including the elaboration of visual models and a standardized notation for this visualization.
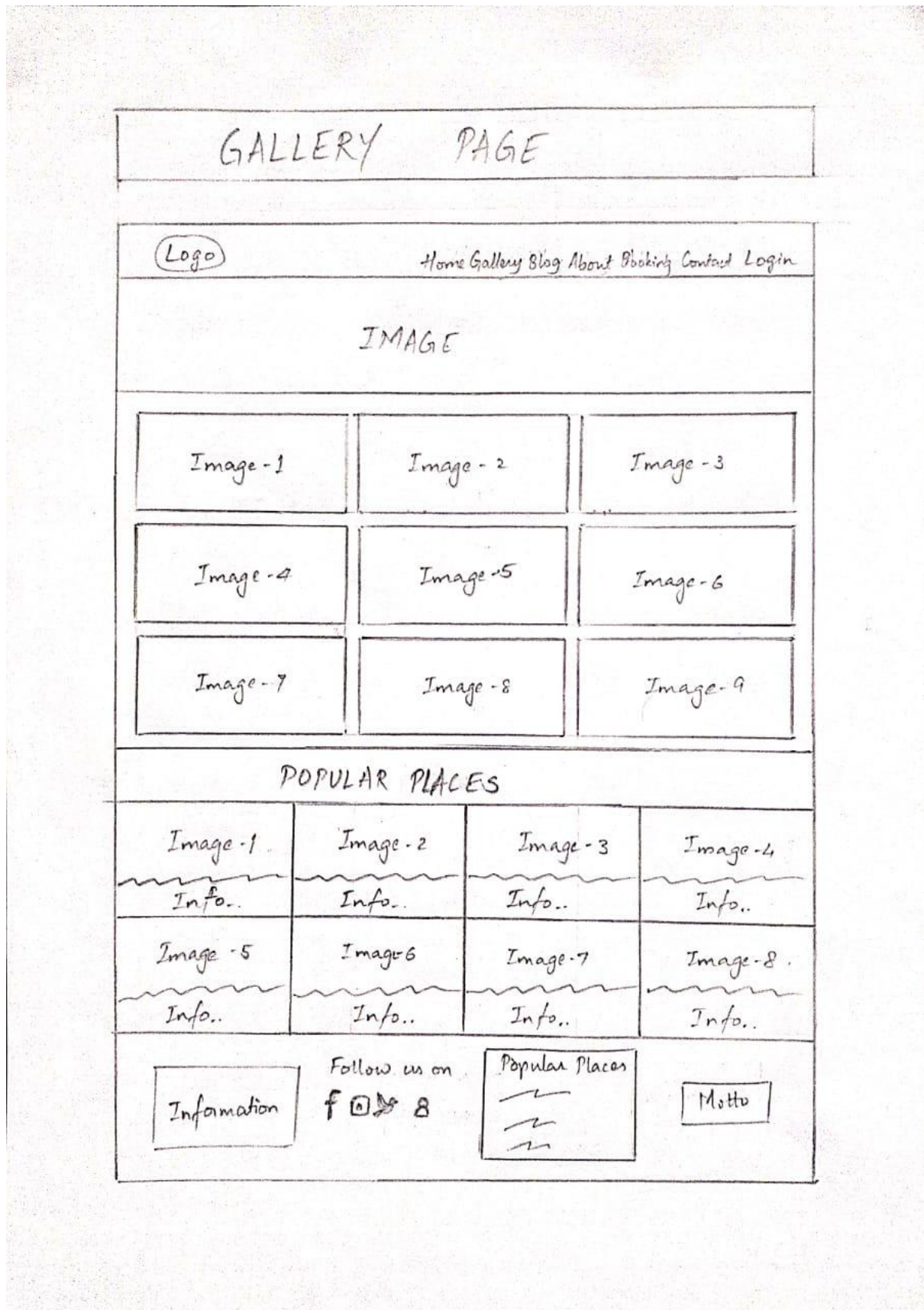
We have used React JS and Bootstrap for frontend UI which has following web pages:
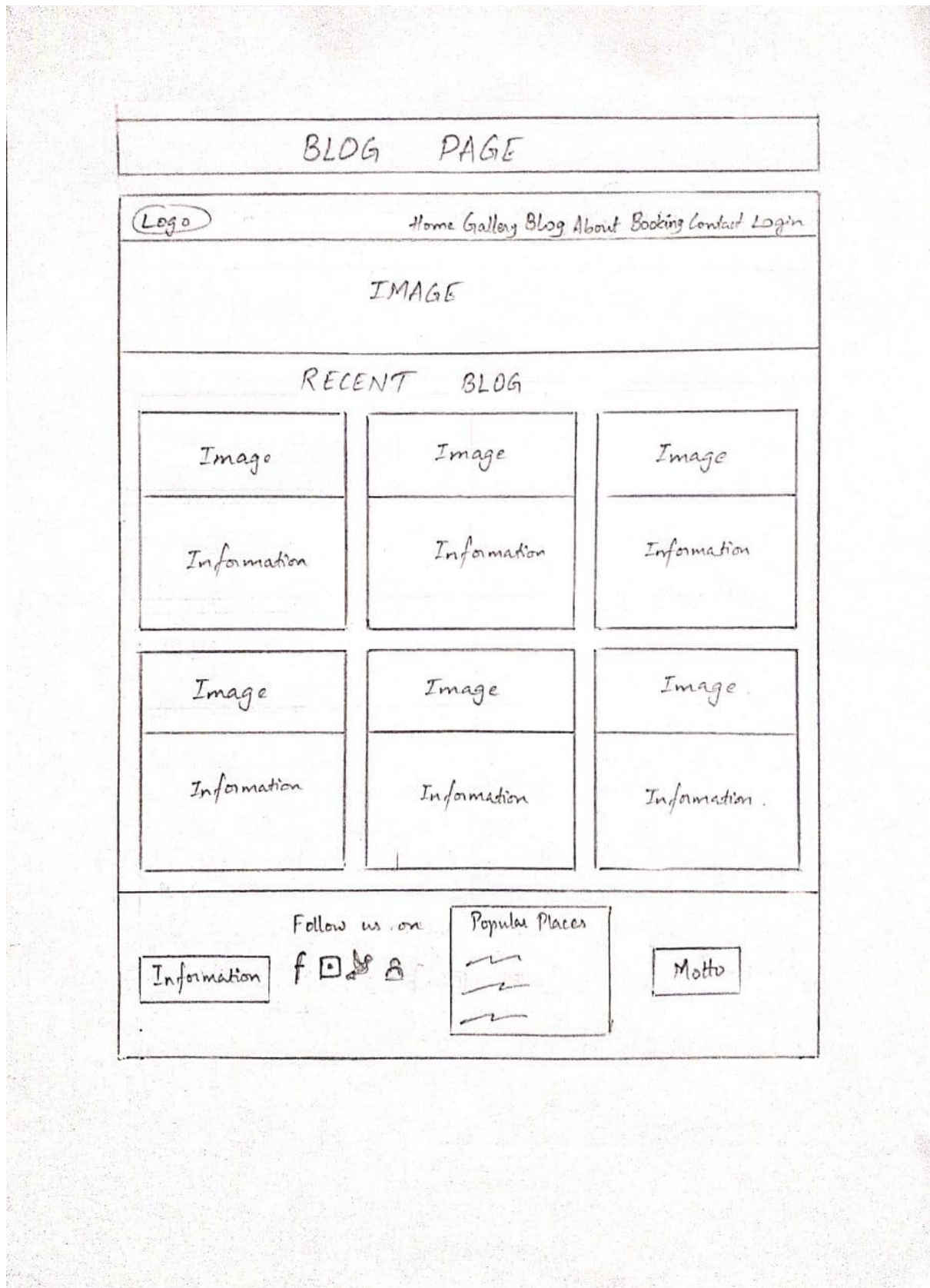
Every Page in common contains a Header and footer as we see in a typical Single Page Web Applications.
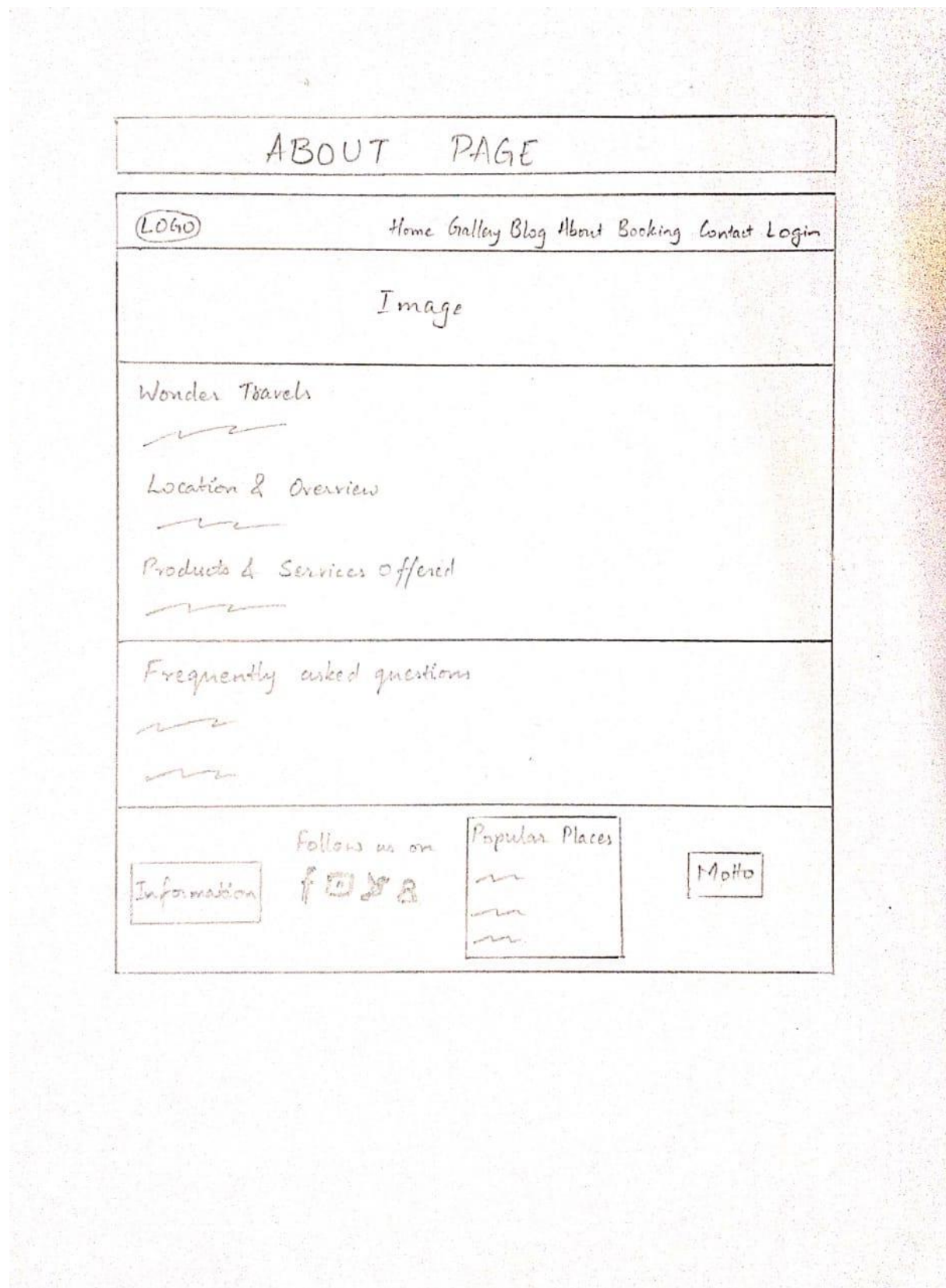
1. **Home Page:** It consists of the list of travel destinations provided by the agency along with the various services it provides followed by a feedback section from the customers.

2. **Gallery Page:** It displays the pictures of all the major landmarks present in various destinations which are provided by the agency along with various popular places one can visit.

3. **Booking Page:** It allows to customer to select the date, time and destination of the location where they want to travel.

4. **Individual Booking Page:** It provides detailed information about the various tourist attractions and activities present at the selected location.

5. **Blog Page:** It displays all the latest blogs about the tourist destinations which give an idea as to where the customers of the agency can visit.

6. **About Page:** It displays information regarding the agency

7. **Contact Page:** It has owner's information such as phone number, email and various social media handles. It also has a Send message field where users can submit their queries (if any).

8. **Login Page:** It has a form with inputs for email and password and an option to create new user if the user is new (register)
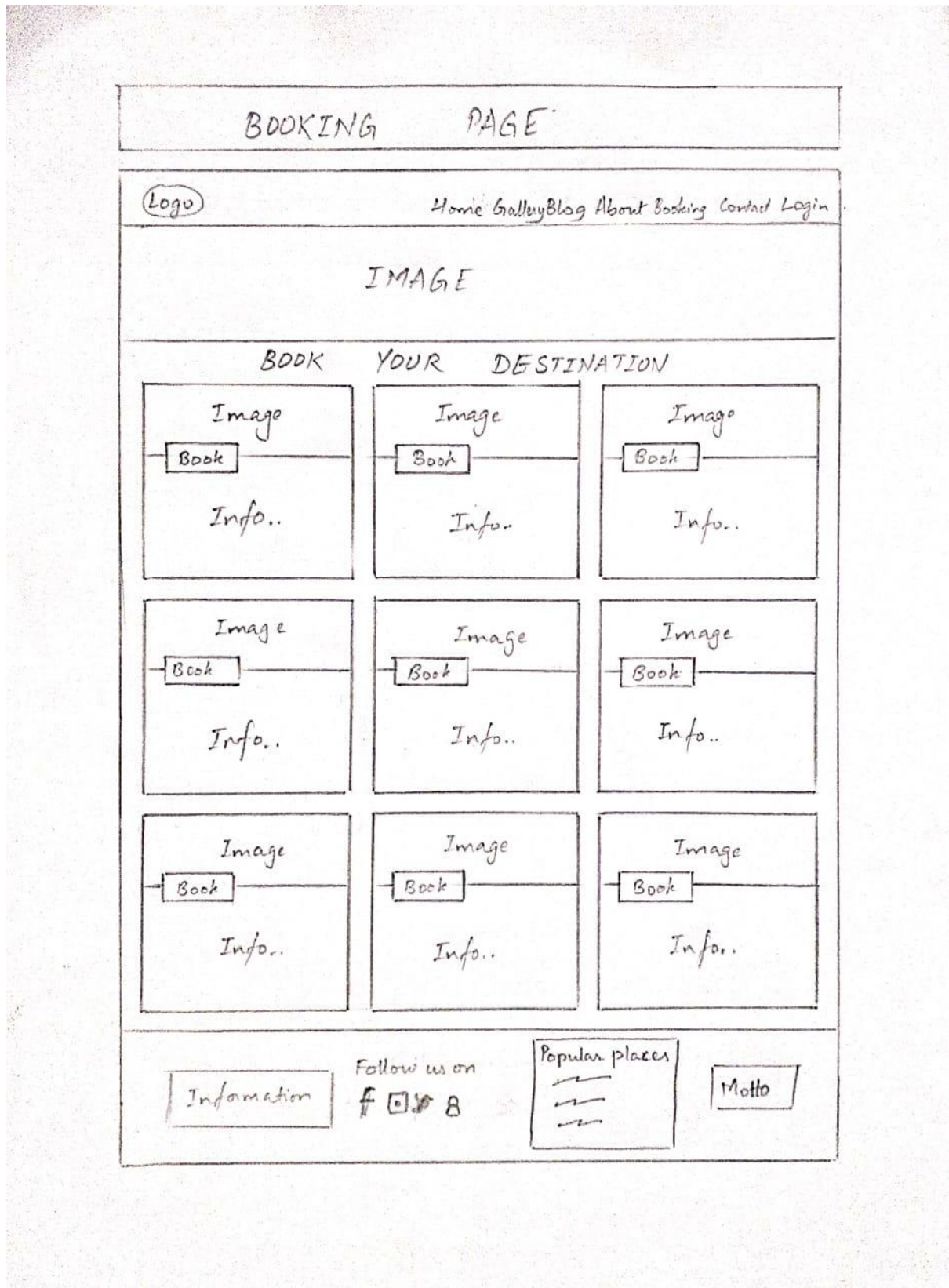
# UI MODEL

BLOG PAGE

Logo                                   Home Gallery Blog About Booking Contact Login

IMAGE

RECENT BLOG

| Image | Image | Image |
|-------|-------|-------|
| Information | Information | Information |

| Image | Image | Image |
|-------|-------|-------|
| Information | Information | Information |

Follow us on          Popular Places

Information    f ⬛ 🕊 🔒                        Motto

## ABOUT PAGE

(LOGO)     Home Gallery Blog About Booking Contact Login

Image

Wonder Travels

Location & Overview

Products & Services Offered

Frequently asked questions

Follow us on    Popular Places

Information    f 🔲 🐦 &             Motto

BOOKING     PAGE

Logo                    Home GalleyBlog About Booking Contact Login

IMAGE

BOOK     YOUR     DESTINATION

Image          Image          Image
Book           Book           Book
Info..          Info..          Info..

Image          Image          Image
Book           Book           Book
Info..          Info..          Info..

Image          Image          Image
Book           Book           Book
Info..          Info..          Info..

Information    Follow us on    Popular places    Motto

## INDIVIDUAL BOOKING PAGE

(Logo)                    Home Gallery Blog About Booking Contact Login

Image

From _____                Weather in ～～～

To _____

(Submit)

| Image - 1 | Image - 2 | Image - 3 |
| Image - 4 | Image - 5 | Image - 6 |

BUS SCHEDULE

| Bus Name | Arrival Time | Reaching Time | Fare | Seat Availability |
|----------|--------------|---------------|------|-------------------|
|          |              |               |      |                   |
|          |              |               |      |                   |

| Number of SEATS |        | BOOK |

| Information | Follow us on   Popular Places | Motto |
|             | f 🔲 🕊 &    ～              |       |

## CONTACT PAGE

(LOGO)    Home Gallery Blog About Booking Contact Login

### Image

### CONTACT US

Your name

Your Email

Your Message

Send Message

📞 Phone

🔵 Address

✉ Message

Follow us on
f 📷 🕊 8

Information

Popular Places

Motto

LOGIN PAGE

Logo           Home Galley Blog About Booking Contact Login

( Log In )   ( Register )

User ID

Email ID

Enter Password

( Register )
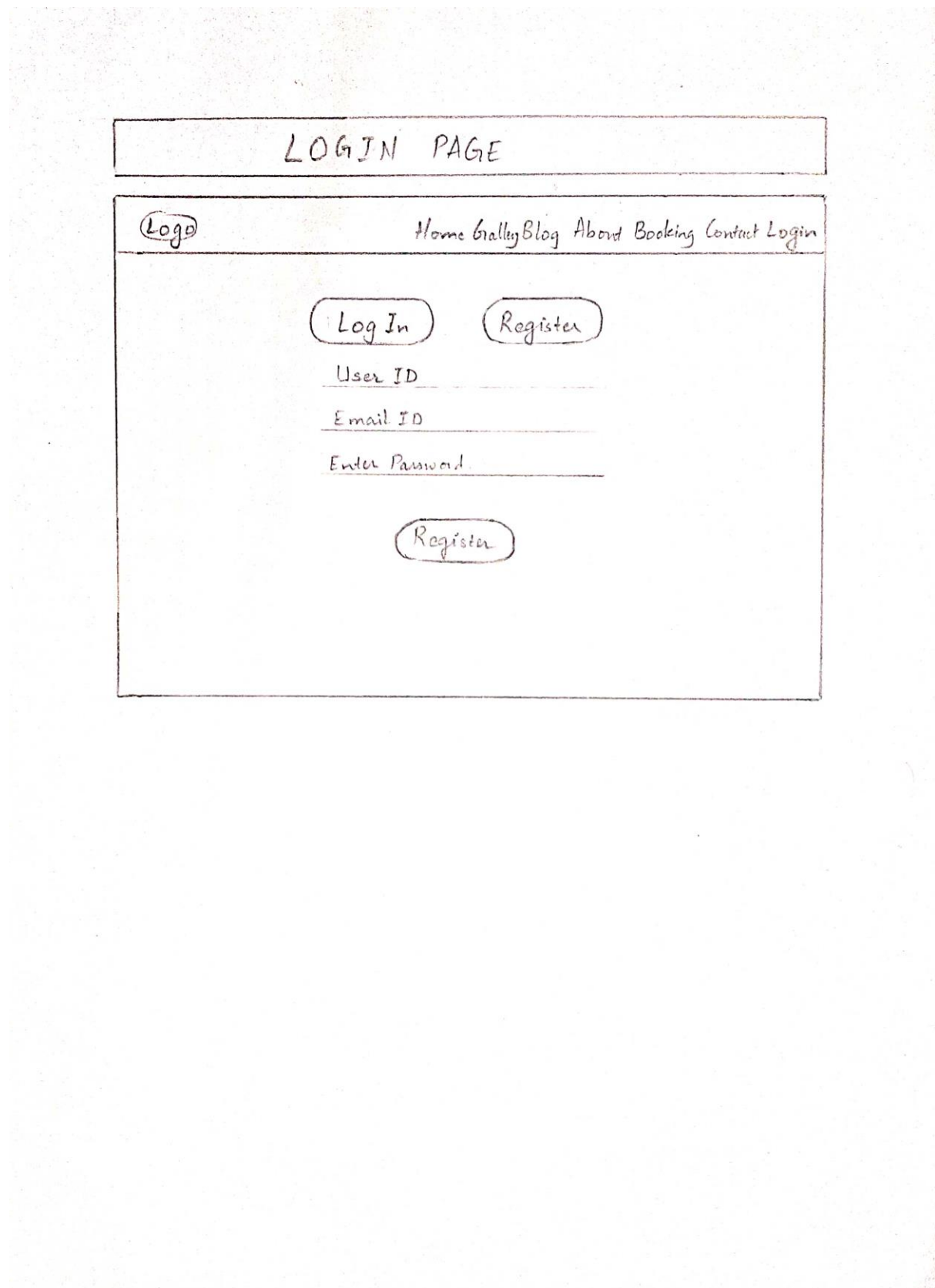
# TEST CASES

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works properly. The process of developing test case can help find problems in the requirements or design of an application.

- Ensure that logical decisions on their true and false side.
- Practice all the logical decisions on their true and false side.
- Check equivalent partitions and boundary value within their operations bounds
- Exercise internal data structure to assure their validity

## Test Case of Login

| Test Case #01 | Test Case Name: Testing the login page. |
|---|---|
| System: Windows | Subsystem: Login |
| Short Description: This field handle's the login functionality of the website. ||
| Precondition: Go to login. ||

**Table: Test case of login**

| Steps | Action | Action Result | Expected System Response | Pass/Fail |
|---|---|---|---|---|
| 01 | Enter valid username and valid password | Get logged in. | Logged in into the system. | Pass |
| 02 | Valid username and invalid password | Not logged in and error Message. | Not logged in and error message. | Pass |
| 03 | Click login without any Data | Required message | Required message | Pass |

## Test Case of Register

| Test Case #02 | Test Case Name: Testing the registration page. |
|---|---|
| System: Windows | Subsystem: Register |
| Short Description: This field handle's the registration functionality of the website. ||
| Precondition: Go to register. ||

**Table: Test case of register**

| Steps | Action | Action Result | Expected System Response | Pass/Fail |
|---|---|---|---|---|
| 01 | Click register without any data | Required message | Required message | Pass |
| 02 | Click register partially filling with data | Not register and required messages. | Not register and required messages. | Pass |
| 03 | Click register with valid data and password more than 6 characters. | Registered in and error message. | No Required message | Fail |
| 04 | Click register with valid data and password not matching with confirm Password. | Not register in and error message. | Not registered in and error message. | Pass |
| 05 | Click register with valid data and password. | Registered and redirected to main landing Page. | Registered and redirected to main landing page. | Pass |

## Test Case of Browse Destination

| Test Case #03 | Test Case Name: Testing the browse functionality |
|---|---|
| System: Windows | Subsystem: Browse |
| Short Description: This field handle's browsing of the destination of the website. | |
| Precondition: Go to Browse field | |

**Table: Test case of Browse Destination**

| Steps | Action | Action Result | Expected System Response | Pass/Fail |
|---|---|---|---|---|
| 01 | Enter valid Destination name | Get the Destination | Destination is found in the website | Pass |
| 02 | Enter invalid Destination name | Destination should not be found | Message indicating that the Destination is not found | Pass |
| 03 | Submit empty field | Required message | Required message | Pass |

## Test Case of Booking

| Test Case #04 | Test Case Name: Testing the Booking page. |
|---|---|
| System: Windows | Subsystem: Booking |
| Short Description: This field handle's the Booking functionality of the website. ||
| Precondition: Press book option. ||

**Table: Test case of Booking**

| Steps | Action | Action Result | Expected System Response | Pass/Fail |
|---|---|---|---|---|
| 01 | Click book without any data | Required message | Required message | Pass |
| 02 | Click book partially filling with data | Not booked and required messages. | Not booked and required messages. | Pass |
| 03 | Click book with valid data | register in and error message. | No Required message | Fail |
| 04 | Click book with valid data and customer not registered | Not register in and error message. | Not registered in and error message. | Pass |
| 05 | Click book with valid data and customer registered | Ticket is booked | Message indicating ticket is booked | Pass |

## Test Case of Payment

| Test Case #05 | Test Case Name: Testing the Payment page. |
|---|---|
| System: Windows | Subsystem: payment |
| Short Description: This handle's the payment functionality of the website. | |
| Precondition: Press pay option. | |

**Table: Test case of payment**

| Steps | Action | Action Result | Expected System Response | Pass/Fail |
|---|---|---|---|---|
| 01 | Click pay without any data | Required message | Required message | Pass |
| 02 | Click pay partially filling with data | Not booked and required messages. | Not booked and required messages. | Pass |
| 03 | Click pay with valid data | Payment successful | Message indicating Payment Successful | Pass |
| 04 | Click pay with valid data and user account has less amount | Payment unsuccessful | Message indicating payment unsuccessful | Pass |

## Test Case of cancelling the booking

| Test Case #06 | Test Case Name: Testing the cancel booking functionality |
|---|---|
| System: Windows | Subsystem: cancel booking |
| Short Description: This handle's the cancel booking functionality of the website. ||
| Precondition: Should have a booking ||

**Table : Test case of payment**

| Steps | Action | Action Result | Expected System Response | Pass/Fail |
|---|---|---|---|---|
| 01 | Click cancel booking without any previous booking | Required message | Required message | Pass |
| 02 | Click cancel booking with a booking done by the customer previously | Booking cancelled successfully | Booking is cancelled successfully | Pass |

**S**

# CONCLUSION

This web application was successfully created and stored all the travel admin tourism packages booking, creation managing and tour details into the database using this application. The application was tested very well and the errors were properly debugged. Testing also concluded that the performance of the system is satisfactory. All the necessary output is generated. This system thus provides an easy way to automate all the functionalities of consumption. If this application is implemented in few consumption, it will be helpful. Further enhancements can be made to the project, so that the website functions in a very attractive and useful manner than the present one. It is concluded that the application works well and satisfy the needs of the user.

SRS (Software Requirements Specification) document helps the customers to define their needs with accuracy, while it helps the development team understand what the customers need in terms of development. Investing time in writing the SRS document will lead to the successful development of the software the customers need. We would like to conclude that the SRS document for our Project Titled "A Travel & Tourism Management System" has been created successfully including all the client requirements, UML diagrams, test cases and User Interface (UI) model.

# REFERENCES

1. All the requirements are taken from the client "Wonder Travels"

2. https://stackoverflow.com/

3. https://www.lucidchart.com/

4. https://www.youtube.com/