FACULTY OF SCIENCE, ENGINEERING AND COMPUTING

School of *Computer Science & Mathematics*

FACULTY OF SCIENCE, ENGINEERING AND COMPUTING

School of *Computer Science & Mathematics*

BSc DEGREE

IN

Computer Science (Software Engineering) Top Up

PROJECT INTERIM REPORT

Name: **Mohamed Fariq Abdulla Sabith**

ID Number: **E026503**

Project Title: **Leisure Diary**

Project Type: **Build**

Date: **17.03.2023**

Supervisor: Ms. Sampa Rasanie

Kingston University London

Did you discuss and agree the viability of your project idea with your supervisor? Yes

Did you submit a draft of your proposal to your supervisor? Yes

Did you receive feedback from your supervisor on any submitted draft? Yes or No

Abstract

Leisure Diary is a solution proposed by AFADAMAD (Pvt) Ltd, a travel agent that aims to assist travelers with their inbound and outbound travel needs to ensure their satisfaction by providing a hassle-free, all-in-one trip organizer. The solution offers a wide range of features that reduce the workload of the organizer and provide the required resources at their fingertips. It provides a convenient platform for all the requirements of a traveler and offers an opportunity for related stakeholders to grow their businesses. This report contains all the details about the technologies, methodologies, work plan, constraints, and ways to overcome them.

Contents

List of Figures

List of Tables

Glossary of Terms

HTML: Hyper Text Markup Language

IDE: Integrated Development Environment

CSS: Cascading Style Sheet

RAM: Random Access Memory

JS: JavaScript

API: Application Programming Interface

Introduction & Literature Review

1.1 Introduction

AFADAMD (Pvt) Ltd is a travel agent that assists travelers with their inbound and outbound travel needs. They also act as trip advisors to ensure that travelers are always satisfied. The company's aim is to provide a hassle-free travel experience to all travelers who engage with them. They have a vast network of hotels, resorts, other leisure activities, and related services within their chain.

Tourism has experienced significant growth over the years, with its impact involving all economic aspects of countries, businesses, and individuals. However, countries and related businesses must be involved in improving the tourism industry. The purpose of this project is to achieve the above and mitigate problems faced by travelers and other related business entities through effective communication.

The primary scope of the project is to provide an all-in-one solution for planning and organizing trip-related aspects without any hassle. It will also act as a pillar for effective communication between stakeholders

1.2 Background and Motivation

In today's fast-paced world, people often find themselves struggling with the stresses of daily life. Traveling has become one of the most popular methods for individuals and families to unwind and spend quality time together. However, planning a trip can often be a daunting task due to the multitude of associated entities. Additionally, businesses play a significant role in the tourism industry and are considered stakeholders in trips.

With the majority of internet users accessing the web through mobile devices, reaching travelers through mobile applications has become the most effective method. Proper communication is essential to resolve any potential issues faced by foreign travelers [1]. While there are numerous travel applications available in the market, there is currently no all-in-one solution.

Separate applications for each entity do not effectively connect them together, and travelers and organizers seek convenience in a single application. A successful application ultimately leads to customer satisfaction, which is the ultimate goal. Trip organizations involve multiple tasks that can make it difficult for organizers to engage in the arrangement of the trip while others are enjoying it. The tourism service chain is continually improving, and the involvement of each sector is essential in its overall success.

The primary focus of mobile or any other type of application development is to improve the user experience (UX), as it is crucial to the value of the application. Users always expect a more convenient solution, and thus, solution providers must evaluate user requirements as a critical initial task. A user-friendly solution will attract new users to the application.

1.3 Problem in brief

Planning a trip can be a daunting process, particularly when it comes to accommodating the interests and preferences of multiple individuals. The process typically involves a myriad of tasks, including choosing a destination, arranging transportation, finding suitable accommodation, planning meals, and organizing daily activities throughout the trip. For many people, the purpose of a trip is to unwind and relax from the daily grind, making the planning process an unwelcome chore.

One major challenge that arises during trip planning is the burden that falls on those responsible for organizing the trip. While others get to enjoy the trip, the planners often remain preoccupied with making arrangements and ensuring that everyone's expectations are met. This problem is particularly acute when planning trips involving large groups, such as corporate or school trips, where coordinating transportation, accommodation, and other logistics can be a formidable task.

Currently, there are numerous applications on the market that provide booking services for hotels, transport, and food. However, there is no all-in-one solution that enables users to plan their entire trip and share the plan with service providers without encountering any additional hassles during the trip.

1.4 Aim & Objectives

1.4.1 Aim

The aim of this project is to develop an all-in-one mobile application to plan, organize and review a hassle-free trip to travelers' solution with all the related stakeholder involvement.

1.4.2 Objectives

- To identify and provide all in one solution to the travelers to address all their needs through one application.
- To suggest recommendations for the travelers to get decisions on planning a trip.
- To design an application which can find fulfill all the aspects of needs of all the stakeholders.
- To build a communication and interaction method between travelers, service providers and the organizers.

1.5. Scope

The scope of the project that refers to the specific set of outcomes or results that need to be delivered based on the project requirements. It outlines what is included and excluded from the project and governs what can be added or removed during the implementation phase. In this project the scope implements a mobile application along with a web application to serve the travelers a hassle-free travel experience where all their needs to be found in one place. This it to make the travelling experience more efficient and time saving. Also, which will help the businesses to reach their clients easily. Throughout a SWOT analysis we have identified the identify the project's strengths, weaknesses, opportunities, and threats.

1.1.Project Scope Statement

A project scope statement is a document that defines the objectives, goals, deliverables, and boundaries of a project. It outlines what the project will accomplish, what is included, and what is not included. The scope statement also includes a description of the project's stakeholders, assumptions, constraints, and risks. It serves as a reference for the project team and stakeholders to ensure that the project stays on track and meets its objectives [1].

Project Scope Statement

Project Name	Leisure Diary
Project owner	Sabith Fariq
Date	
Product Scope Description	Scope: • Requirement analyzing and engineering. • Identifying the flow and mapping the Flow charts, Data Flow Diagrams, ER Diagrams and Prototyping • Designing and developing the Leisure Diary Mobile app and Web Application. • Testing the implementation • User Manual • Documenting
Project Deliverables	• Project proposal • Project Gantt Chart and Milestone Schedule • Design document • Develop Mobile app • Develop Web App • Test Document • User manual • Final Report
Constrains	• Time • Resource
Acceptance Criteria	The project scope is agreed and accepted by the management.

Table 1- Project scope statement

1.6 Deliverables

The deliverables describe the primary outcome of the project as the result of the scope. Hence, we have different roadmaps and tracking tools to reach the success without any constrains. In the form of charts the tools will be laid as the path to reach project objectives as mentioned above.

The project delivers followings,

- Project Gantt Chart and Milestone Schedule
- Design document
- A Mobile app for the travelers
- A Web App for the service providers
- Test Document
- User manual
- Final Report

1.5.1. Milestone Schedule

Number	Task	Deliverables	Time period
1	Requirement gathering and analysis	• Functional and non-Functional requirements • Gantt chart • Milestone Schedule • Project proposal	December- January 2023
2	System Design	• Prototyping • UI/UX Designing • Requirement Engineering • Design	January – March 2023

		Diagrams	
3	System Implementation	• Mobile Application • Web Application	March – April 2023
4	Testing	• Testing reports	March – April 2023
5	System deployment and submission	• Final report • Developed product	March – April 2023

Table 2 Milestone Schedule

1.5.2 Gantt Chart

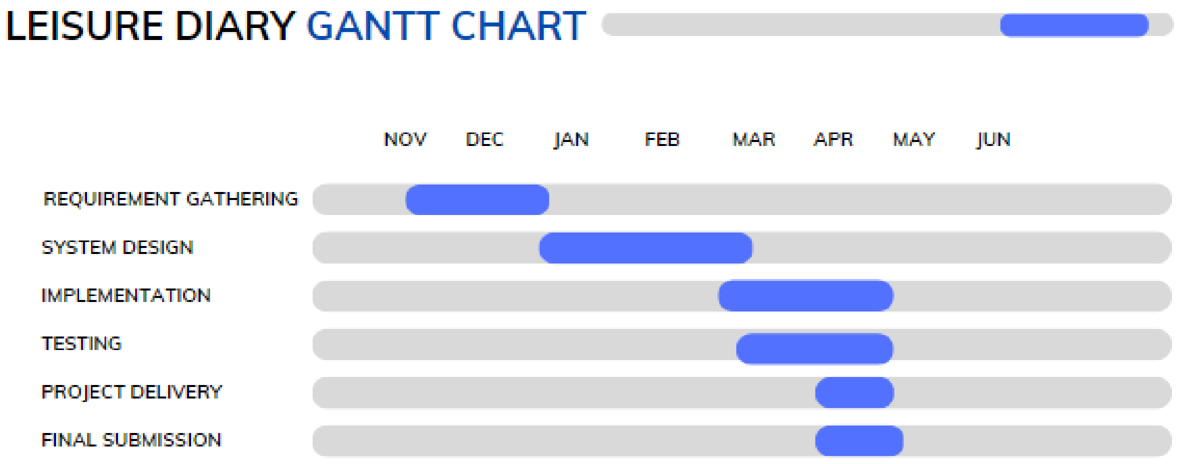


Figure 1 Gantt Chart

1.7 Literature Review

Trip planning is an essential part of tourism and travel. It entails planning, selecting, and analyzing key modules of a journey, including lodging, activities, transportation, and finances. Travelers now have access to a variety of tools that can help them plan their vacations thanks to the development of technology and the ease with which information is available online.

Wang and Chen stated [3] in their study that majority of tourists prefer to use online platforms for travel planning and organizing via websites, social media contents and travel agencies. Also, their study indicates that travelers who have used online platforms to plan and organize their trips are more satisfied.

A study conducted by Lee et al. (2008) indicates the factors which effects the decision-making process of travelers through his study. It also shows how attractive is the destination, convenience and cost are the major factors considered by the travelers when making decisions. Further, the study clearly indicates that travelers more often use online reviews and recommendations posted by other travelers as a key assistant to make decisions [4].

The study which was carried by Deegan et al, which evaluated the planning and organizing element on overall travel experience says the quality of travel planning was positively related to travel satisfaction and loyalty. Also, the study indicated that travel agencies and companies should focus more on creating more personalized packages and travel planning services to enhance and satisfy the travelers [5].

As we aware the tourism is a one of the fastest growing industries in the world. The function of travel planning in encouraging sustainable travel behavior has drawn more attention as the importance of sustainable tourism has grown. The desire of tourists to engage in sustainable travel behavior during the trip preparation phase was examined in a study by Vu and Tasci (2018). Tourism attitudes toward sustainable travel, perceptions of the place, and perceived behavioral control were discovered to be important determinants of travelers' intentions to practice sustainable travel [6].

Mobile Application for travel

Buhalis (2008) has reviewed in his article that there are variety of mobile applications available for the travelers also the author has mentioned there are plenty of features and functionalities on travel applications which bring the convivence for the travelers [7]. In recent years it has proved that travel apps has become increasingly popular among travelers. Studies and researches have explored the factors that influence the use and getting adopted to travel apps.

Wang and Chen (2007) stated travelers are more likely to adopt and use travel apps if they perceive them as useful for their travel needs. This includes features such as booking accommodations and transportation, finding local attractions and activities, and accessing travel guides and maps [3].

A study conducted by Yu, Lio, and Yao (2018) indicates that travelers are more likely to use the travel mobile apps when travelers find the convenience to locate their navigations through it [8]. Moreover the study explored by Xiang, Du, and Ma (2017) stated that adoption of the travel apps are influenced by society. Such as reviews and recommendations influence the decision making of travelers [9].

Security and privacy concerns could be a potential barrier to the adoption of travel apps. People concern more about their privacy and security of data which could reduce consumers to relucted on adopting to travel apps. Kim and Lee (2018) highlighted that developers have to focus more on security to avoid aforementioned types of security and privacy issues [10].

Overall, these factors can have a significant impact on the adoption and use of travel apps by consumers. App developers and marketers should consider these factors when designing and promoting travel apps to ensure that they meet the needs and expectations of users.

2. Analysis

Analysis is a crucial stage in software development that is typically the first step in the software development life cycle. During this phase, the development team collaborates closely with stakeholders to identify and document their requirements. This includes understanding the business objectives, user needs, and technical constraints. Additionally, the team may conduct market research and competitor analysis to better understand the target audience and industry landscape. The analysis phase is critical in ensuring that the software project is feasible, meets the needs of its stakeholders, and identifies potential risks and challenges. It also enables the team to define the project's scope, objectives, and deliverables, which serves as a basis for project planning and estimation. Furthermore, the analysis phase provides a foundation for designing the software architecture and creating a software requirements specification (SRS) that outlines the software's functional and non-functional requirements, performance, security, and usability considerations. Accurately documenting requirements is essential to ensuring that the software meets the stakeholders' needs, is delivered on time, and stays within budget. Overall, the analysis stage is a critical component of software development that sets the foundation for the project's success [2].

2.1 Use case diagram

A use case diagram is a graphical representation commonly used in software development that displays how a system interacts with its users or other systems. Its purpose is to present the system's functional requirements in a simplified way. It provides a broad perspective of the system's functionality and presents the various user types or actors involved in the system. Use case diagrams help to document and convey requirements, and can serve as a blueprint for building software features [3].

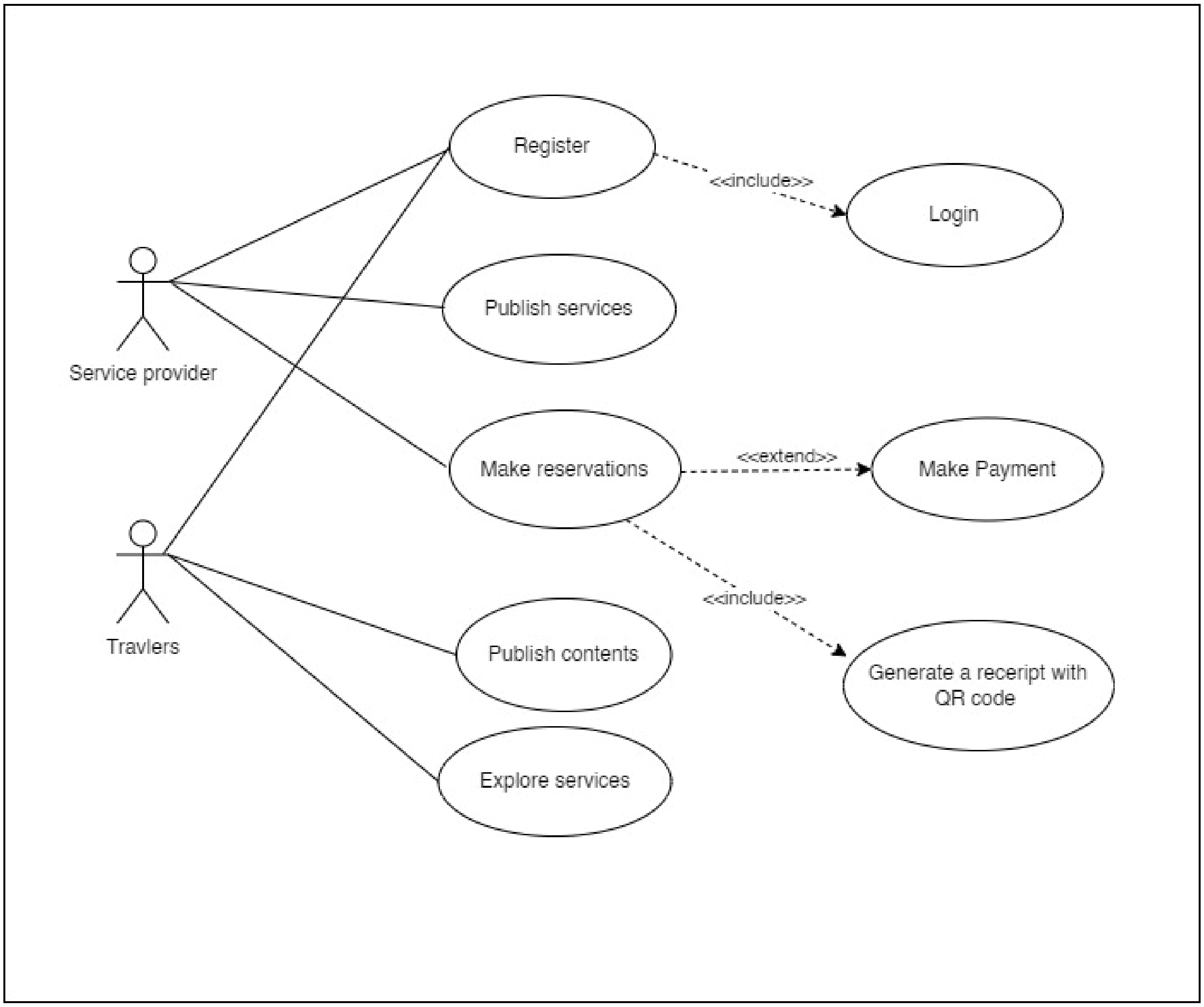


Figure 2Use case diagram

2.2 Clear problem definition

Planning a trip can be a complex process depends on the type people involved and their interests. The process involves numerous tasks, such as searching and selecting a destination, arranging transportation, finding suitable accommodations, organizing food arrangements, and planning day-to-day activities throughout the trip. Many individuals are hesitant to get involved in the planning process because the ultimate goal is to satisfy all co-travelers and their expectations. Those who are often stressed or busy may view a trip as an opportunity to relax and unwind.

2.3 User stories

User stories are a technique used in agile software development to capture and communicate requirements from the perspective of the end-user. Each user story describes a feature or functionality that the user wants to achieve and follows a simple template: "As a [user], I want [feature], so that [benefit]. [4]"

Service provider

- As a service provider, I want to add my services to the system, so that the travelers can explore them and reserve them and that will help to grow my business.
- As a service provider, I can add, edit or remove the published services.
- As a service provider, I want to generate a report of the reservations.

Traveler

- As a traveler, I want to explore the services as per my requirements and find the most suitable one.
- As a traveler, I want to reserve the required services to fulfill my travel needs.

2.4 SWOT Analysis

Strength	Weakness
• Brand recognition establishment for the businesses. • Diverse range of travel offerings and packages • Strong partnerships with hotels, airlines, and other travel-related companies • Convince of travelers on finding related services. • Save time and misleading by the fraud guiders.	• Limited geographic reach or scope of services • Limited differentiation from competitors • Competitor pricing

Opportunities	**Threat**
• Expanding the business to reach more audience from new geographical locations • New and innovative ways to develop and reach the travelers. • Offering personalized and customized travel services to clients • Developing partnership with co-service providers and expand the business and brand recognition	• Competition from online travel booking platforms and price comparison websites • Fluctuating currency exchange rates • Changes in government regulations or travel restrictions • Natural disasters, health crises, or other unforeseen events that could impact travel demand • Increasing costs of travel-related services, such as fuel prices and hotel rates.

Table 3-SWOT Analysis

2.5 Requirement Engineering

Requirement engineering is a process of eliciting, analyzing, specifying, validating, and managing the requirements of a software system. It involves understanding the needs of stakeholders, including end-users, customers, and business owners, and translating those needs into software requirements that can guide the development process.

The goal of requirement engineering is to ensure that the software system being developed meets the needs of its intended users and stakeholders. It involves a range of activities, including gathering and analyzing user needs, creating use cases, and defining functional and non-functional requirements [5].

2.5.1 Functional requirements

Id	Description	Priority

FR1	**Title:** Service provider/ Travelers Registration Requirements Any anonymous user should see a “Register Account” button to register a user account. When the user clicks the “Register Account” button, the application navigates to the Register Account page. The “Register Account” page displays a registration form with the required details: Service providers: • Company details • Service category • Location • Types of service provided • Logo Email address should be validated against proper email structure and existing emails. If the email doesn’t have the format user@domain.com, a validation message should be displayed which says, “Improper email address”, and if the email already exists in the system the error message should be “User with the given email already exists, please login.” The two password fields should appear redacted, and if they do not match a validation message is displayed saying “The password does not match”. The “Submit” button must be disabled until all mandatory fields are filled and validated. The company email and the registration number should be validated to avoid duplicating.	High

	Travelers can create one account with one email address. And Travelers should create their accounts through the mobile app.	
FR2	**Title:** User registration When the user clicks “Submit” on the registration page, the system should create a new user account and the service providers and travelers can login to the system. Service providers can log in to the web portal and publish their services. Travelers can login to the mobile app and explore their needs and services published by the service providers.	High
FR3	**Title:** User Login When an anonymous user visits the website, they can click on the login button to access the login page. On the login page, they will see a form with fields for entering their username and password. They can also see a login button and a cancel button. If the user clicks the cancel button, they will be taken back to the home page. If the user enters an incorrect username or password and tries to log in, they will see an error message that says "Invalid username or password". If they enter the correct login credentials and click the login button, they will be successfully logged into the system and redirected back to the home page.	High
FR4	**Title:** Administrative account privileges Leisure Diary admin account is only accessible to the admins of the	High

	Leisure Diary. The type of the default system account is an "Administrative" account Admin has the access to view details about all the service providers and the services published by them and evaluate them.	
FR5	**Title:** Common Site Layout There are different layouts for the service provider which is a web application and a mobile application for the travelers. Web application: • Login Screen • Profile • Services grid • Create Service form • Reservation list Mobile Application: • Login screen • Profile • Feed • Services feed • Reservation options	

Table 4 Functional Requirements

2.5.2 Nonfunctional requirements

Performance Requirements

Id	**Description**	**Priority**
P.NFR1	All images stored on the server file system must be encoded with WEBP	HIGH

	file format and they must stay within 20KB and 100KB file size for optimal transfer.	
P.NFR2	All CSS and JavaScript assets must be minified and bundled for optimal size.	HIGH

Table 5 Performance requirements

Privacy and Security Requirements

Id	Description	Priority
S.NFR1	All usernames and other sensitive user information must be encrypted.	HIGH

Table 6 Privacy and security requirements

Software Quality Requirements

Id	Description	Priority
Q.NFR 1	Source code should have a unit test coverage of above 90%	MEDIUM
Q.NFR 2	It is advised that developers must implement features with testing in mind, therefore they may follow a suitable test-driven development methodology.	MEDIUM

Table 7 Software Quality Requirements

2.6 Resource requirements

Software requirement	• Visual Studio code editor • Windows 7 or above • MongoDB/Firebase • Android studio
Hardware Requirements	• Personal Computer • 8GB RAM or higher • 500GB HDD

	• 1GB Video Graphics card or Higher
Client requirement	• Mobile phone with internet connection • A PC with internet connection
Technological requirements	• HTML • CSS • JavaScript • Flutter • MongoDB / Firebase • EJS

Table 8 Resource requirements

2.7 Outcome of the analysis

The requirement elicitation is the primary objective achieved by the analysis process. User stories and use case diagram assisted to get a clear picture of the actual scenario and to identify the flow of the system. Clear problem identification phase helped to elaborate the actual problem and align it with the requirement identified in the early phrases. Along with that we have conducted a SWOT analysis which is the best way to identify Strengths, Weakness, Opportunities, and Threats in the related field and forecast the future. As a summary of the requirement elicitation, we have finalized the functional requirements and non- functional requirements to implement the system.

3. Design

3.1 Design Techniques

3.1.1 Flow Chart

A flowchart is a visual representation of a process or algorithm using various symbols and shapes to show the sequence of steps and decisions involved. It is a useful tool for understanding and documenting complex processes or systems. Each symbol in a flowchart represents a specific action or decision point in the process, and arrows connect the symbols to show the flow of the process. Flowcharts are used in a variety of fields, including software development, engineering, and business. They help to simplify complex information and make it easier to understand [6].

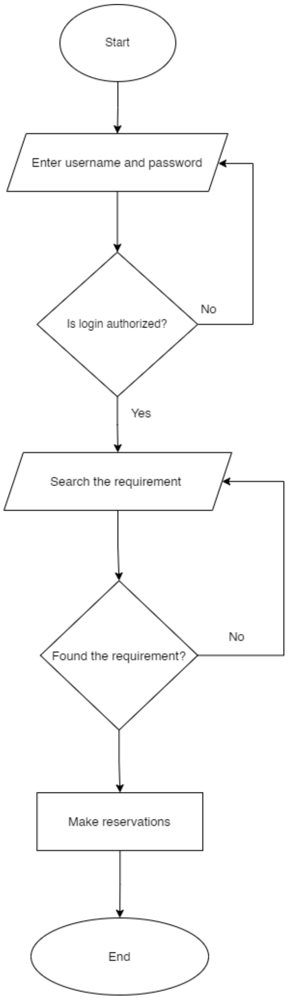


Figure 4 Travelers flow chart

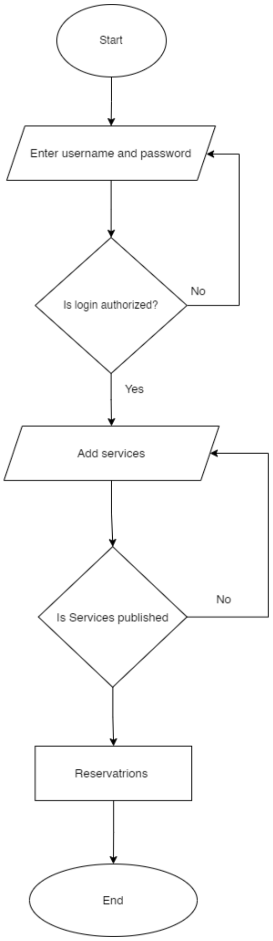


Figure 3 Service provider Flow chart

3.1.2 Use Case Diagram

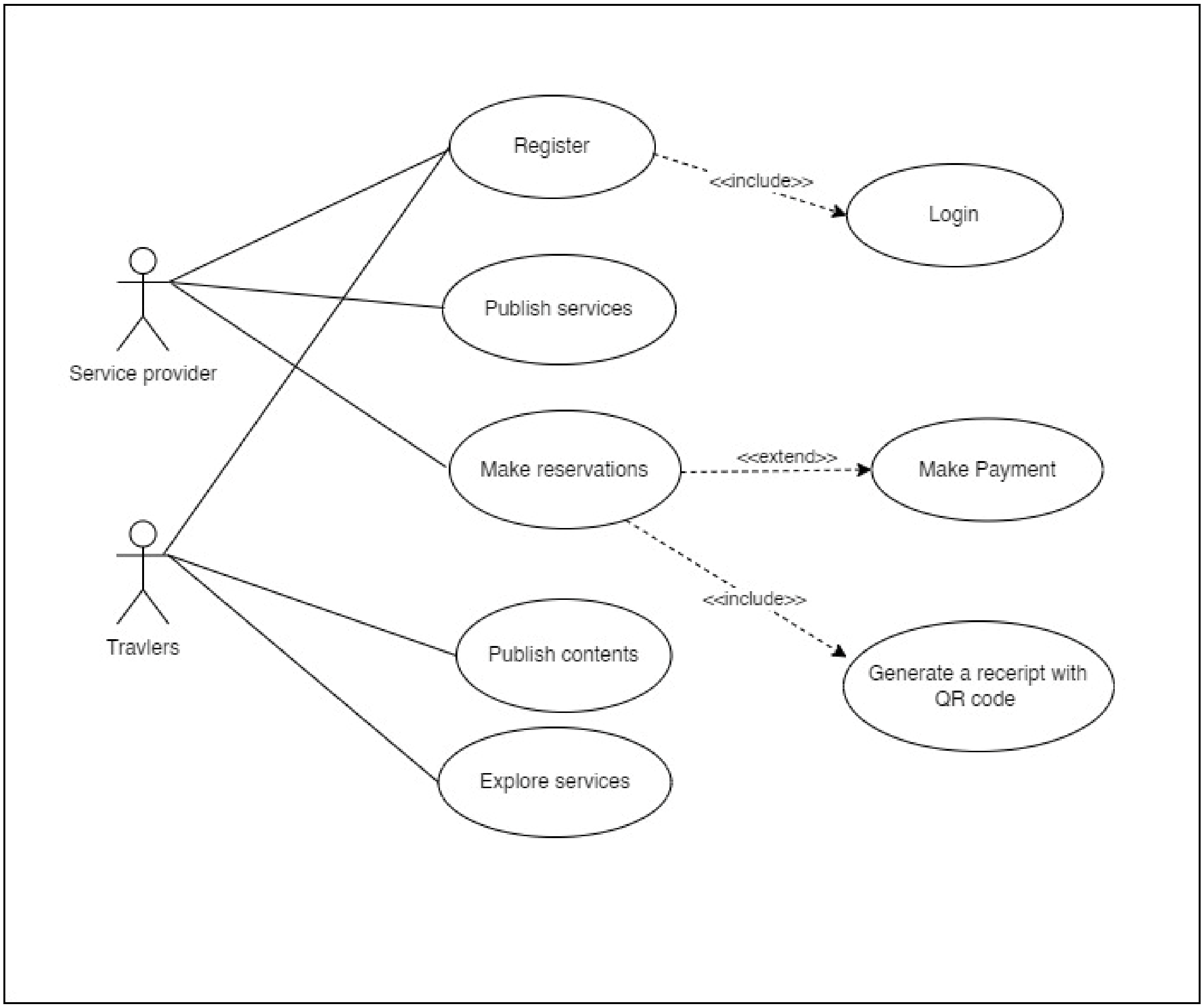


Figure 5- User case diagram

3.1.3 ER Diagram

An Entity Relationship Diagram (ERD) is a type of flowchart that illustrates the relationships between entities such as objects, people, or concepts in a system. ERDs are often used to design and debug relational databases in various fields such as software engineering, business information systems, education, and research. These diagrams use a specific set of symbols, including diamonds, rectangles, ovals, and connecting lines, to represent the interconnectedness of entities, relationships, and their attributes. ERDs follow a grammatical structure where entities are represented as nouns and relationships as verbs. The main purpose of an ERD is to show the infrastructure of the entity framework, as illustrated in the example of the leisure diary project.

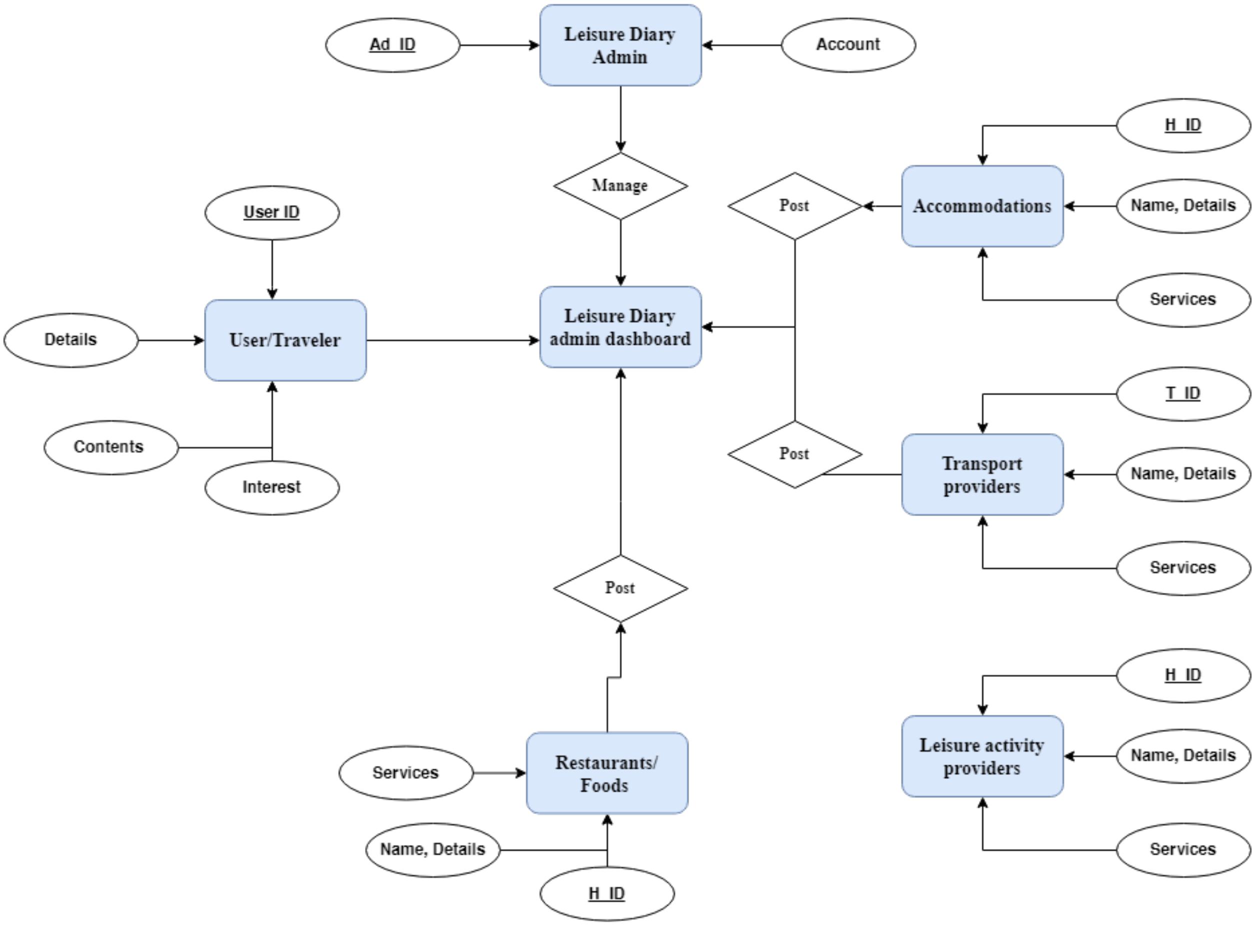


Figure 6 ER Diagram

3.1.4 High Level System Component Diagram

Below is the high-level structure of the application. The backend consists of a REST API coupled with two databases. A HTTP port and a TCP port is exposed for external communication interfaces. WebSocket communication shares the same HTTP port.

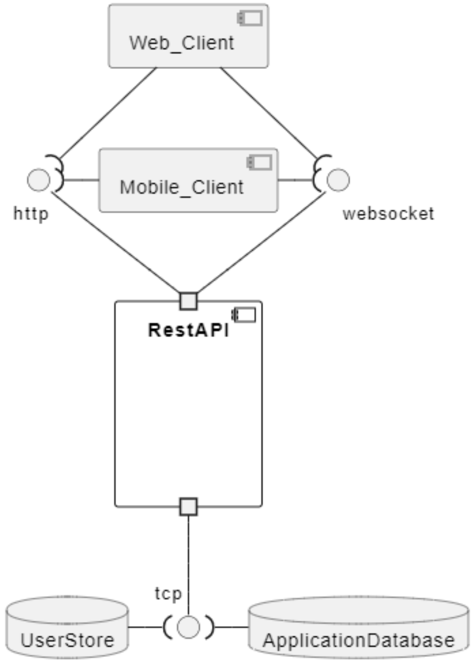


Figure 7 System component diagram

3.2 System Overview

Leisure Diary is the proposed solution for the identified significant problems and requirements addressed by the travelers and other service providers. The solution is a web based mobile application connected to an admin panel through a rest API technology. Application will have many entities such as Location accommodators (Hotels and resorts), transport providers, foods, other restaurants and other leisure activities. The solution is like a story creator which attracts the user's interest with going forward with the application.

A user can register to the application by creating an account. The story begins the user can start to create the travel package. User can base locations, type of foods/restaurants or leisure activities as the initial point of selection. The application shows recommendations based on your interest. With adding the no. of people to the system it filters the available options to proceed. After creating the whole story of the trip, you can share it among your co-travelers and other service providers on their approval of the reservations.

3.2.1 Wireframes

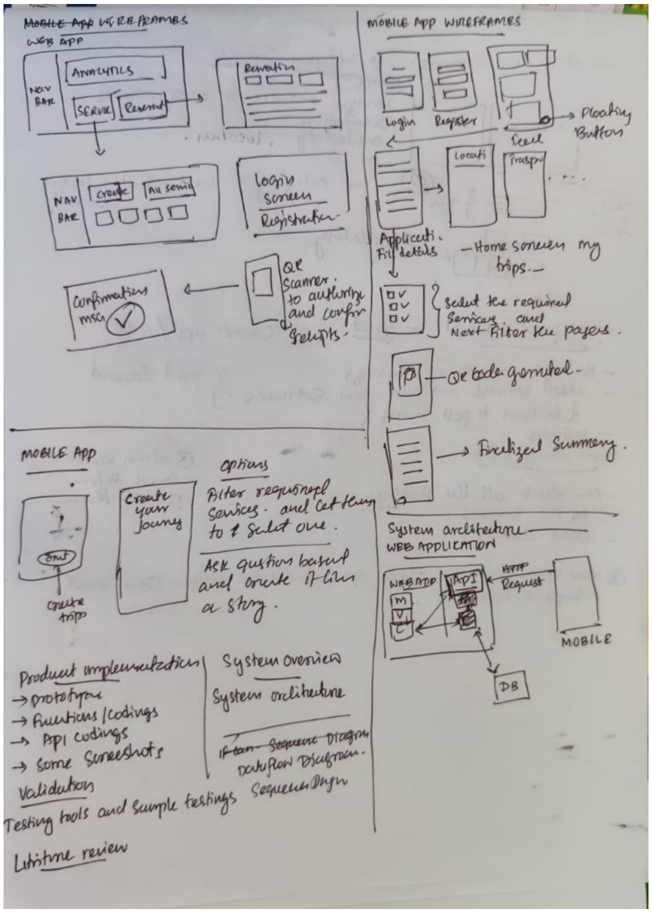
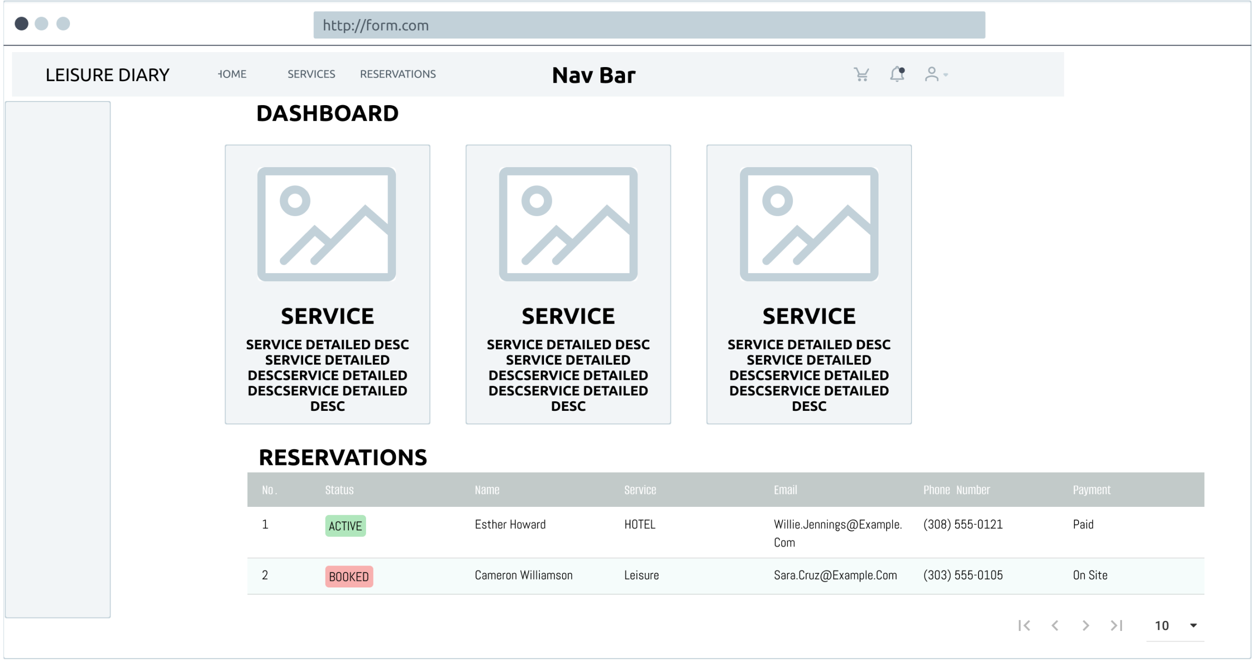
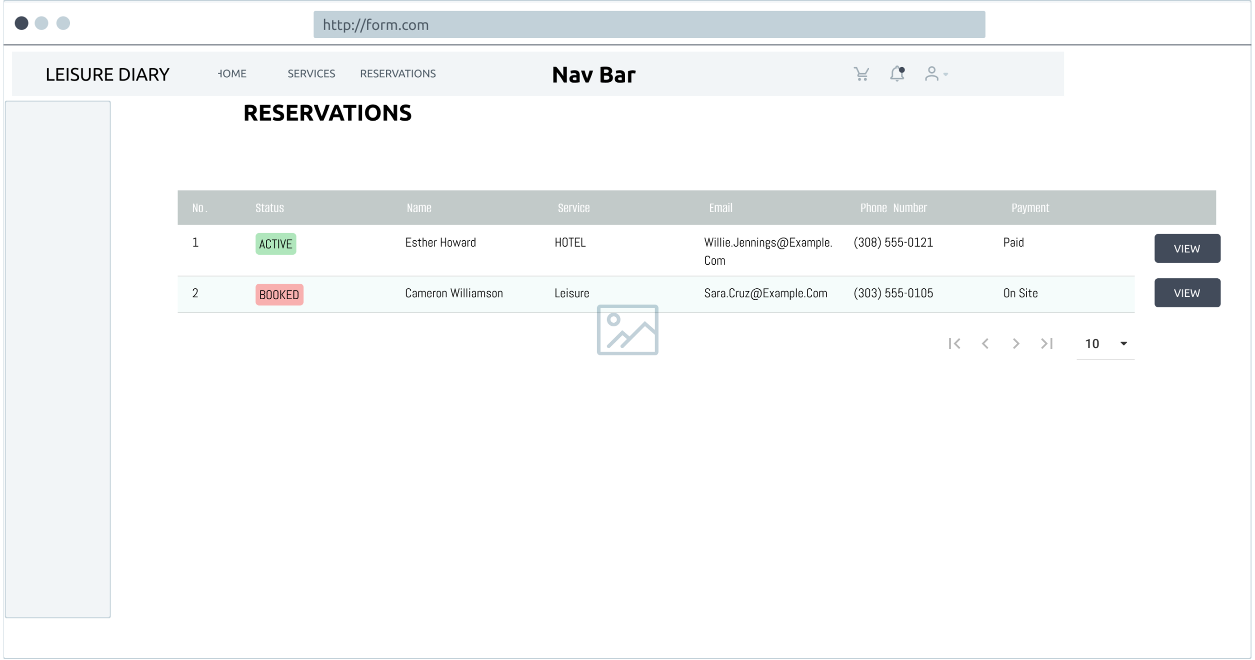


Figure 8 Wireframe Sketch


http://form.com
LEISURE DIARY
HOME
SERVICES
RESERVATIONS
Nav Bar
DASHBOARD
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
RESERVATIONS
10

No.	Status	Name	Service	Email	Phone Number	Payment
1	ACTIVE	Esther Howard	HOTEL	Willie.Jennings@Example.Com	(308) 555-0121	Paid
2	BOOKED	Cameron Williamson	Leisure	Sara.Cruz@Example.Com	(303) 555-0105	On Site

Figure 9 Wireframe



No.	Status	Name	Service	Email	Phone Number	Payment	
1	ACTIVE	Esther Howard	HOTEL	Willie.Jennings@Example.Com	(308) 555-0121	Paid	VIEW
2	BOOKED	Cameron Williamson	Leisure	Sara.Cruz@Example.Com	(303) 555-0105	On Site	VIEW

Figure 10 Wireframe

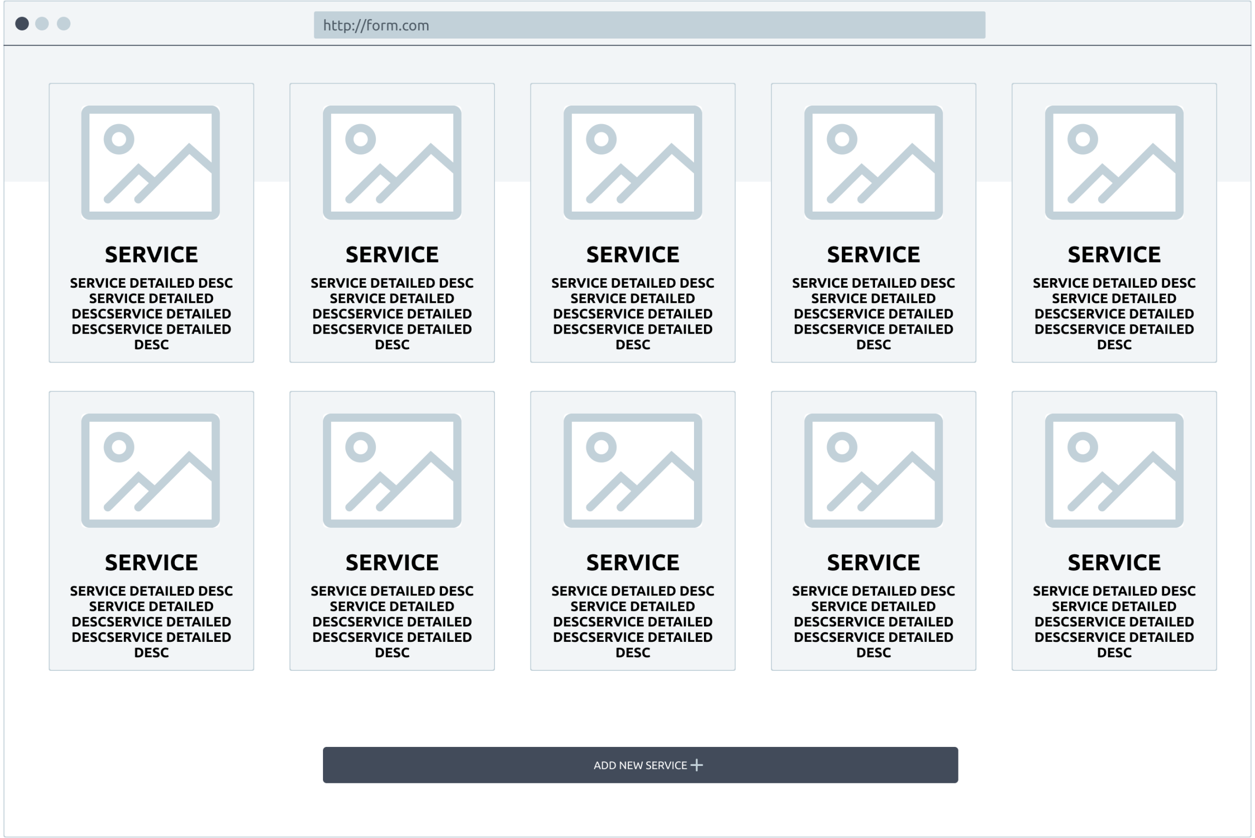

http://form.com
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
SERVICE
SERVICE DETAILED DESC SERVICE DETAILED DESCSERVICE DETAILED DESCSERVICE DETAILED DESC
ADD NEW SERVICE +

Figure 11 Wireframe

Mobile App wireframes

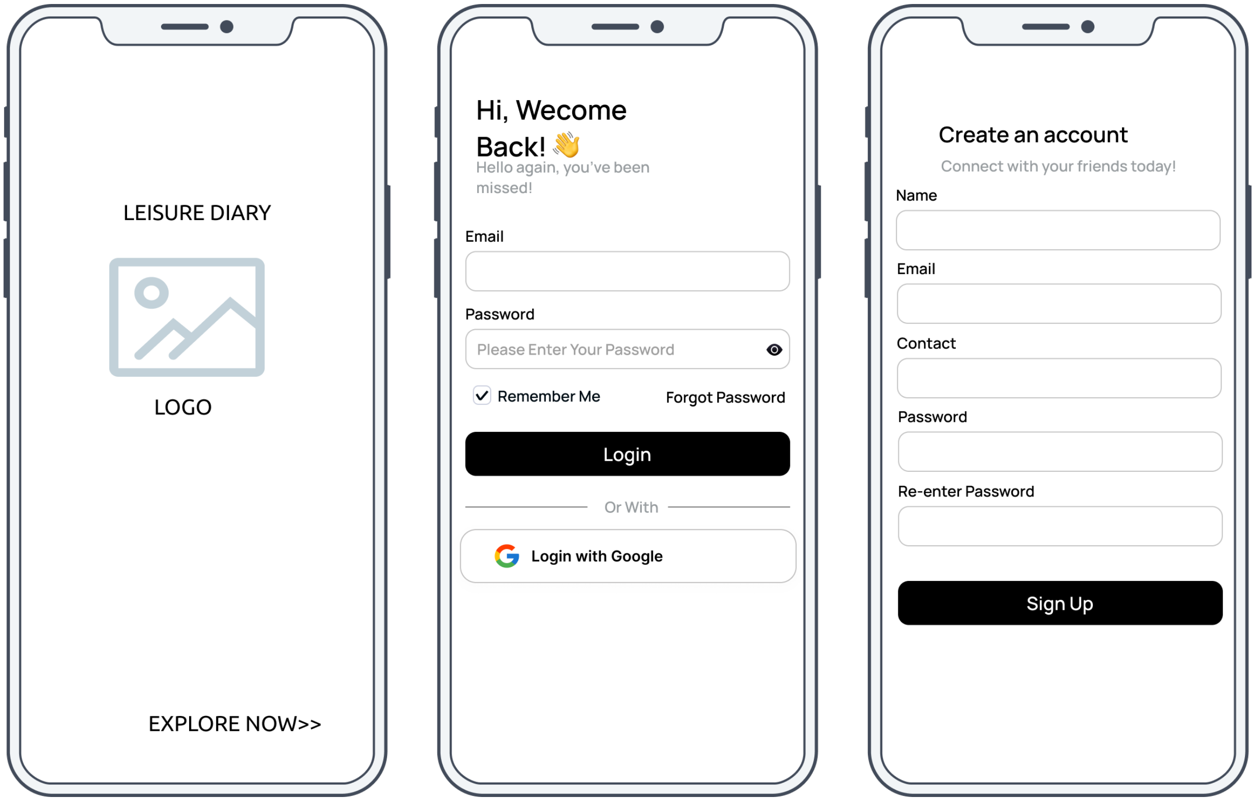
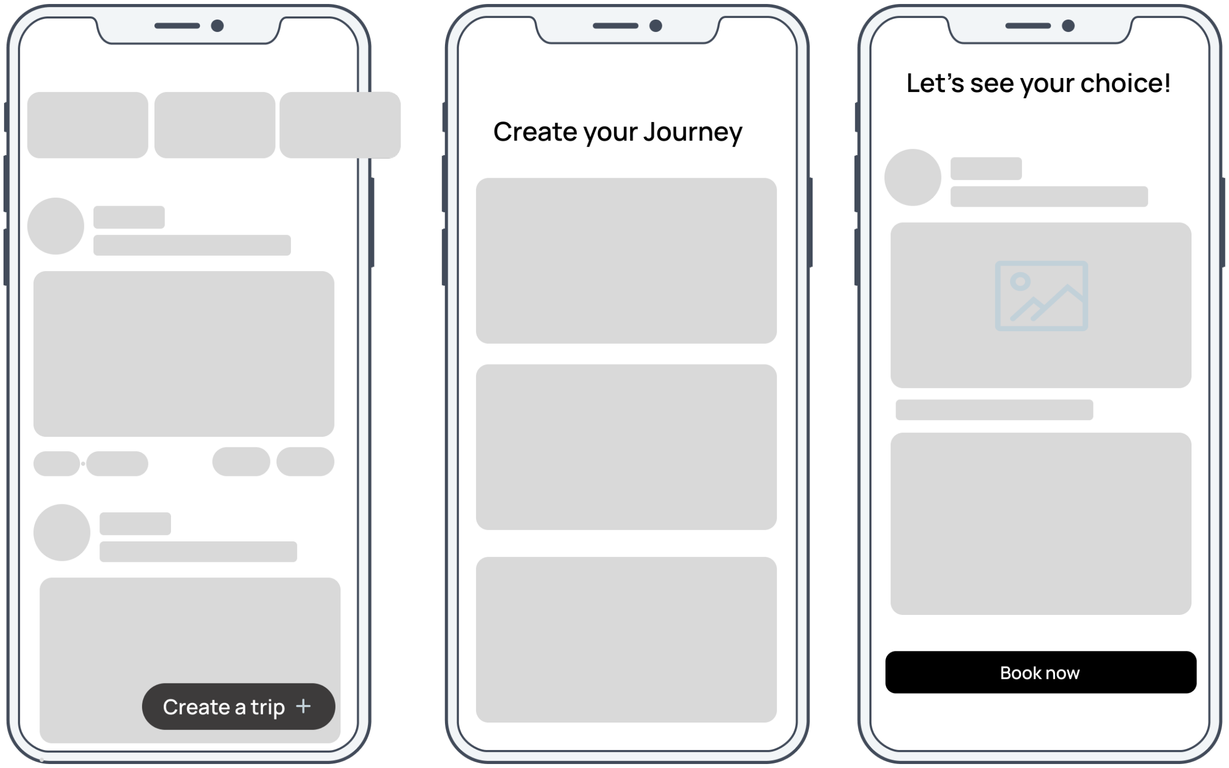
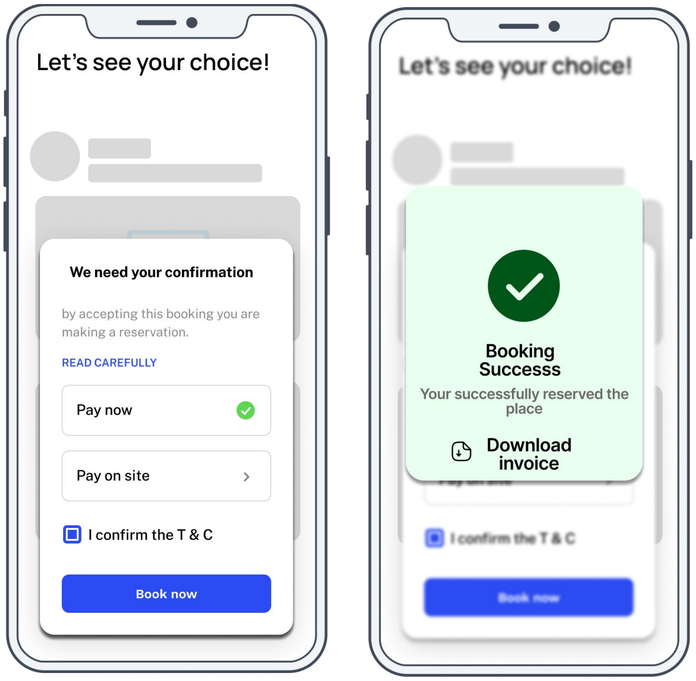


Figure 12 Wireframe





Let's see your choice!

We need your confirmation

by accepting this booking you are making a reservation.

Let's see your choice!

Booking Successs

Your successfully reserved the place

3.2.2 System Architecture

The design of a complicated system that consists of hardware, software, network infrastructure, and other components is referred to as system architecture. It aids in ensuring that all parts perform properly by defining a system's structure, behavior, and functioning.

The system architecture is crucial because it gives stakeholders a high-level picture of the system and makes it possible for them to comprehend the connections and interactions among various components. Making wise decisions about trade-offs between cost, performance, and usefulness is made easier by early identification of potential problems and hazards [16].

As it indicates in the below architecture the web application creates the data and store it to an API which will be saved in the DB. The REST API will be used to communicate with the mobile application. API can directly access the model and the services of the web application and save the changes to cloud database.

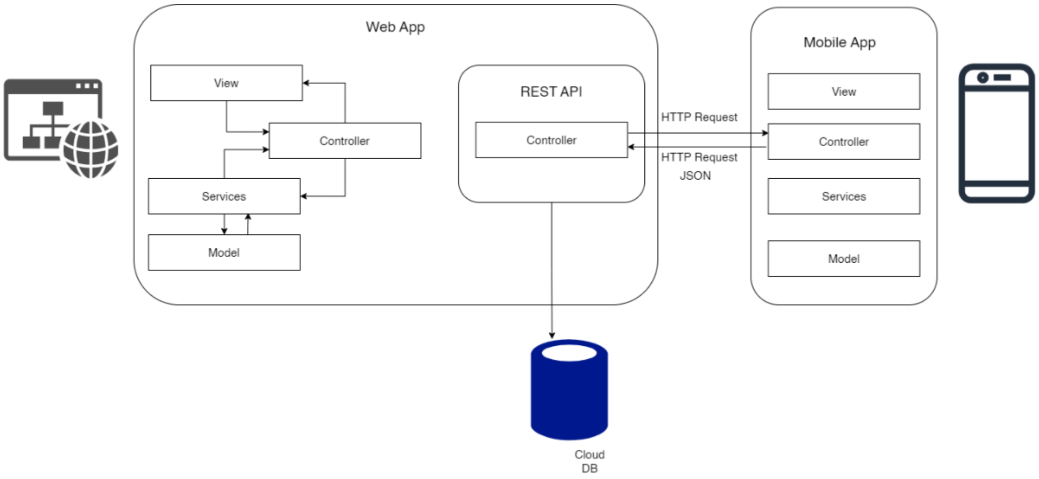
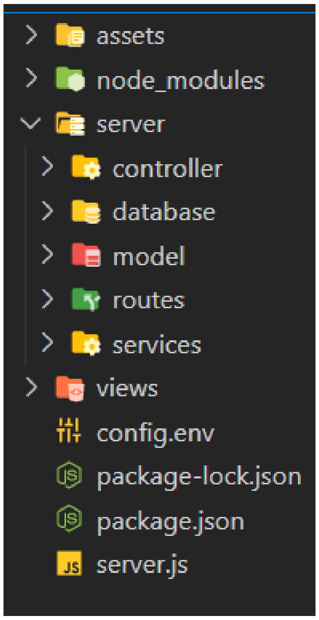


Figure 13 System Architecture

4. Product Implementation

As we identified through requirement elicitation, a web app and a mobile app are to be created and connected through a REST API. The decided technologies that are to be used when implementing the web application are ExpressJs which a frame work of NodeJs to build the back end and HTML, CSS and JavaScript have been used to frontend of the application. To save more time when building the frontend, a templating language EJS (Embedded JavaScript) has been used. It will make the code look simply, reduce the developing time, speedy execution, easy debugging and active development. Also, to avoid the code being more sophisticated to understand we have refactored the code into separate folders as per the MVC Architecture. By using expressJs the REST API endpoints have been created to communicate with the mobile application and being saved in the mongoDb database which is a cloud Database. Also, the system will be utilizing the object-oriented programming (OOP) concept in order to organize and manipulate the objects in the web application and mobile application.

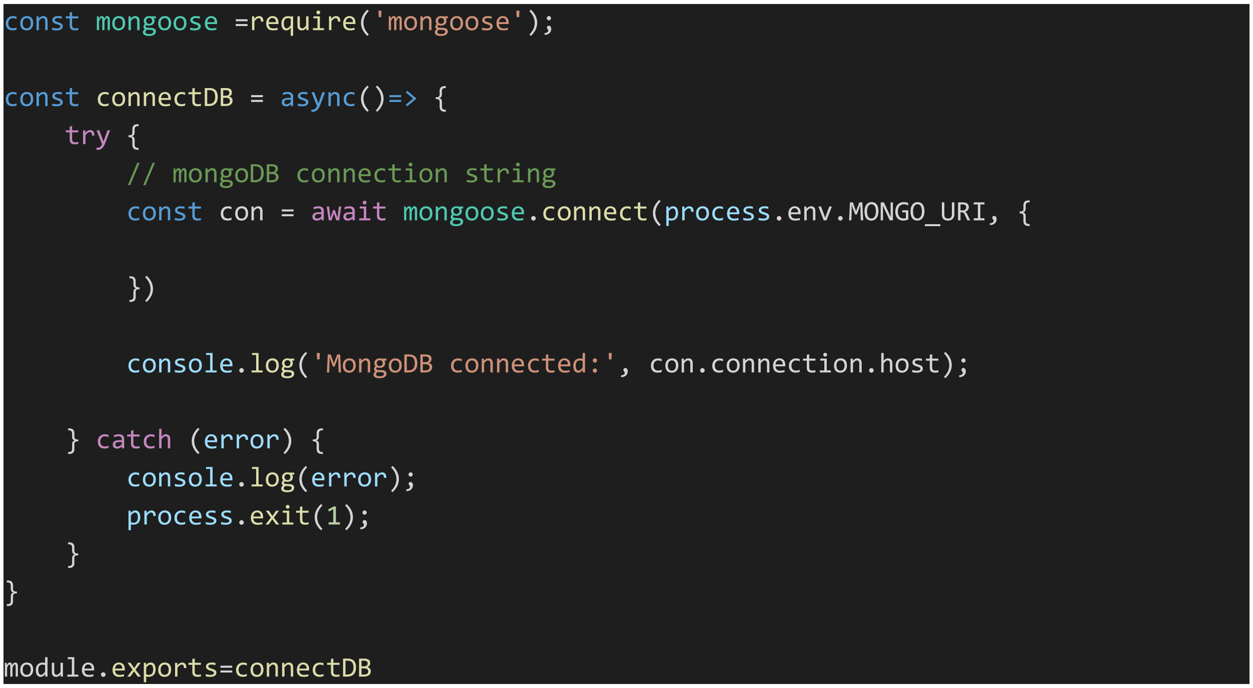
Folder Structure of the web application backend



```
> assets
> node_modules
v server
    > controller
    > database
    > model
    > routes
    > services
> views
  config.env
  package-lock.json
  package.json
  server.js
```

Figure 14- Folder Structure

Creating Database connection.

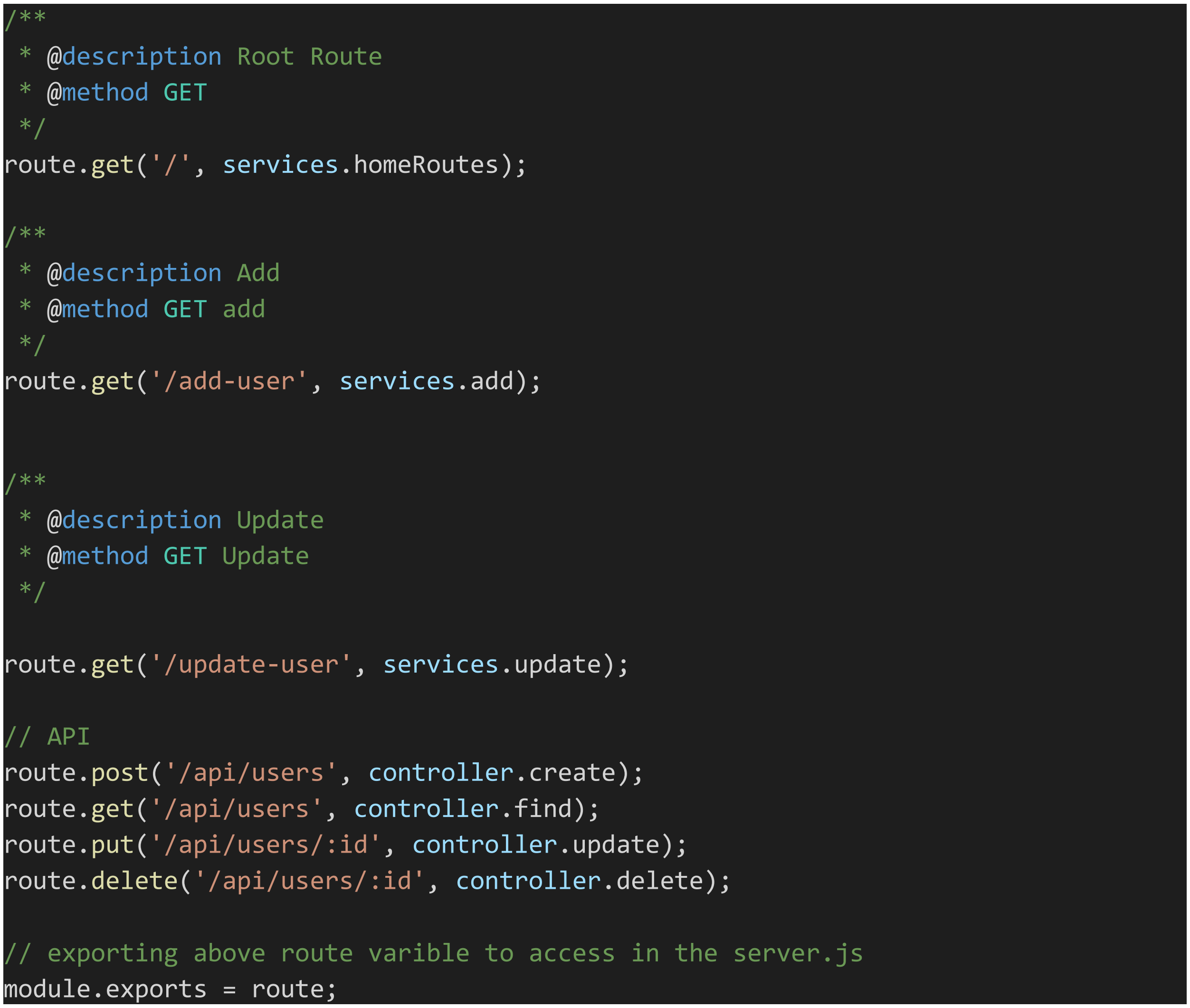


```
const mongoose =require('mongoose');

const connectDB = async()=> {
    try {
        // mongoDB connection string
        const con = await mongoose.connect(process.env.MONGO_URI, {

        })

        console.log('MongoDB connected:', con.connection.host);

    } catch (error) {
        console.log(error);
        process.exit(1);
    }
}

module.exports=connectDB
```

Figure 15- DB connection

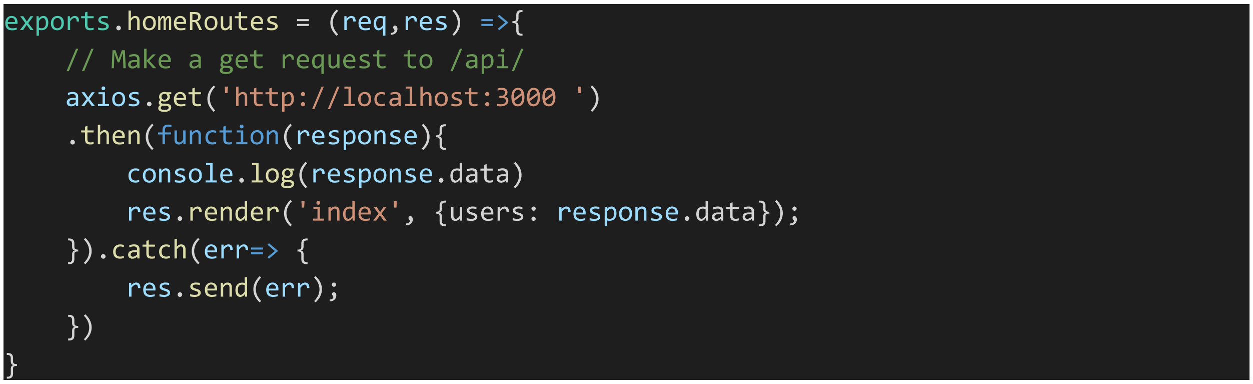
Code refactoring

Routing



```
/**
 * @description Root Route
 * @method GET
 */
route.get('/', services.homeRoutes);

/**
 * @description Add
 * @method GET add
 */
route.get('/add-user', services.add);


/**
 * @description Update
 * @method GET Update
 */

route.get('/update-user', services.update);

// API
route.post('/api/users', controller.create);
route.get('/api/users', controller.find);
route.put('/api/users/:id', controller.update);
route.delete('/api/users/:id', controller.delete);

// exporting above route varible to access in the server.js
module.exports = route;
```

Figure 16 Routing codes

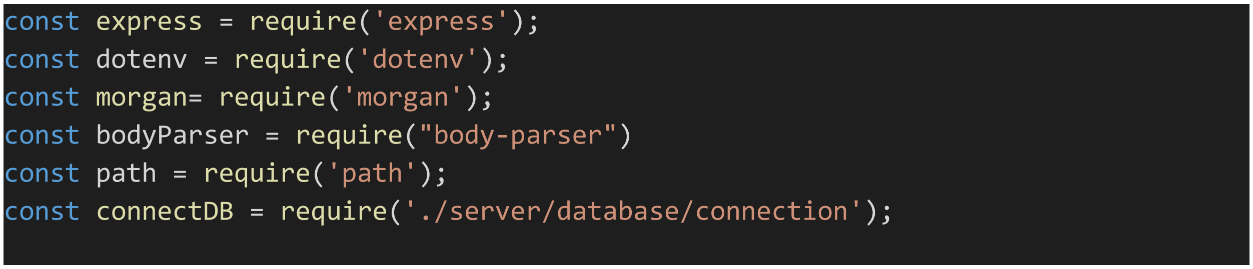


```
exports.homeRoutes = (req,res) =>{
    // Make a get request to /api/
    axios.get('http://localhost:3000 ')
    .then(function(response){
        console.log(response.data)
        res.render('index', {users: response.data});
    }).catch(err=> {
        res.send(err);
    })
}
```

Figure 17 Routing Codes

Libraries used

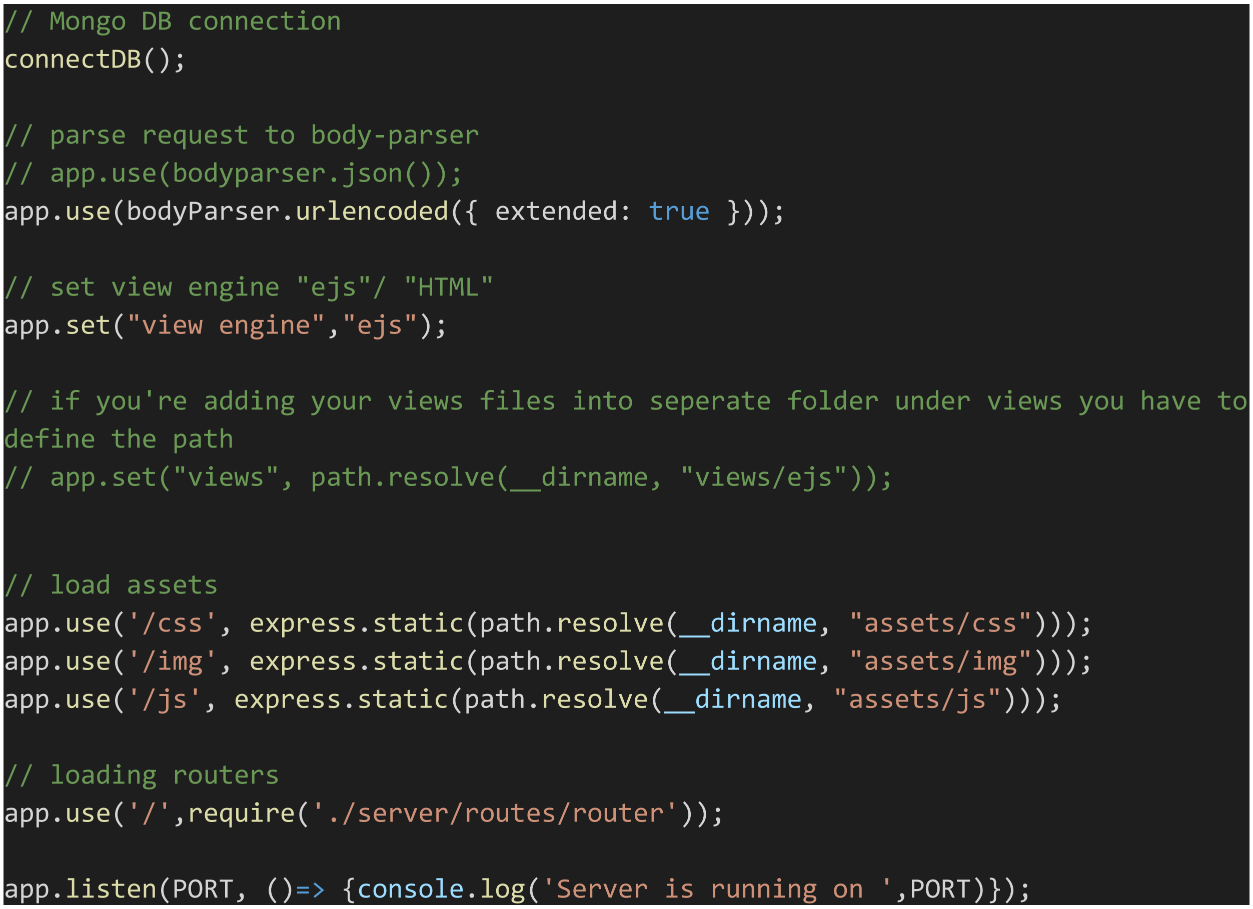
Express is used to connect with the expressJs framework. Env configuration file has been implemented to hide confidential information being shared. Morgan module library is used to print a log whenever we make any request. It will help us to identify bugs in requests. BodyParser encodes and decodes json data communications.



```
const express = require('express');
const dotenv = require('dotenv');
const morgan= require('morgan');
const bodyParser = require("body-parser")
const path = require('path');
const connectDB = require('./server/database/connection');
```

Figure 18 Libraries

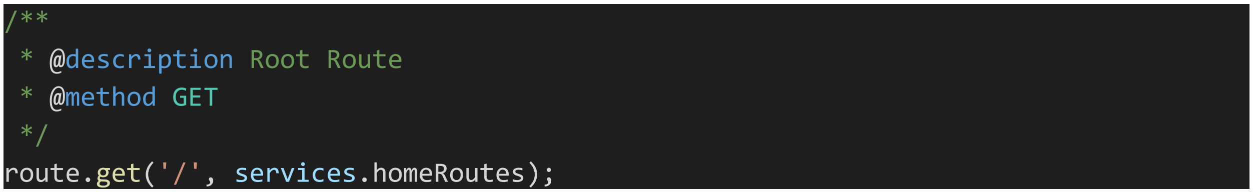
Loading assets to the server



```
// Mongo DB connection
connectDB();

// parse request to body-parser
// app.use(bodyparser.json());
app.use(bodyParser.urlencoded({ extended: true }));

// set view engine "ejs"/ "HTML"
app.set("view engine","ejs");

// if you're adding your views files into seperate folder under views you have to
define the path
// app.set("views", path.resolve(__dirname, "views/ejs"));


// load assets
app.use('/css', express.static(path.resolve(__dirname, "assets/css")));
app.use('/img', express.static(path.resolve(__dirname, "assets/img")));
app.use('/js', express.static(path.resolve(__dirname, "assets/js")));

// loading routers
app.use('/',require('./server/routes/router'));

app.listen(PORT, ()=> {console.log('Server is running on ',PORT)});
```

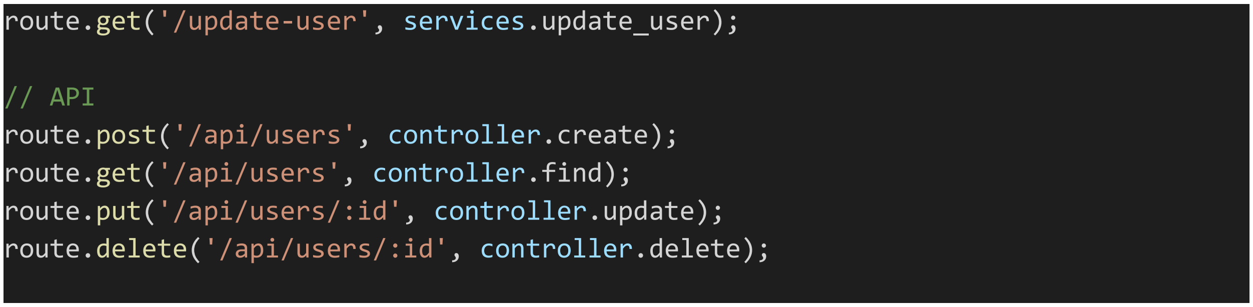
Figure 19 Server implementations

API Endpoints



```
/**
 * @description Root Route
 * @method GET
 */
route.get('/', services.homeRoutes);
```

Figure 20 API Endpoints

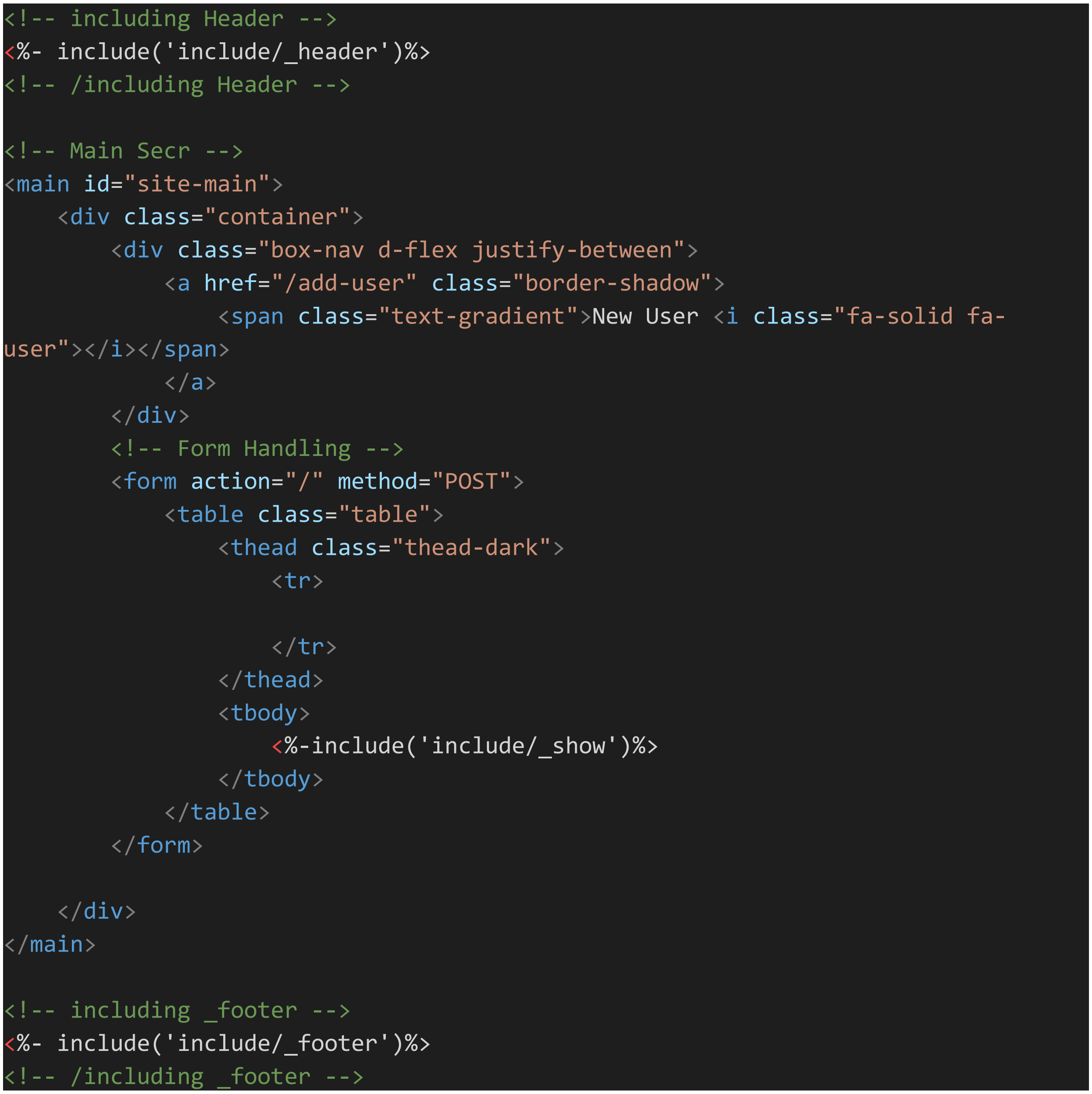


```
route.get('/update-user', services.update_user);

// API
route.post('/api/users', controller.create);
route.get('/api/users', controller.find);
route.put('/api/users/:id', controller.update);
route.delete('/api/users/:id', controller.delete);
```

Figure 21 API Endpoints

Font End views

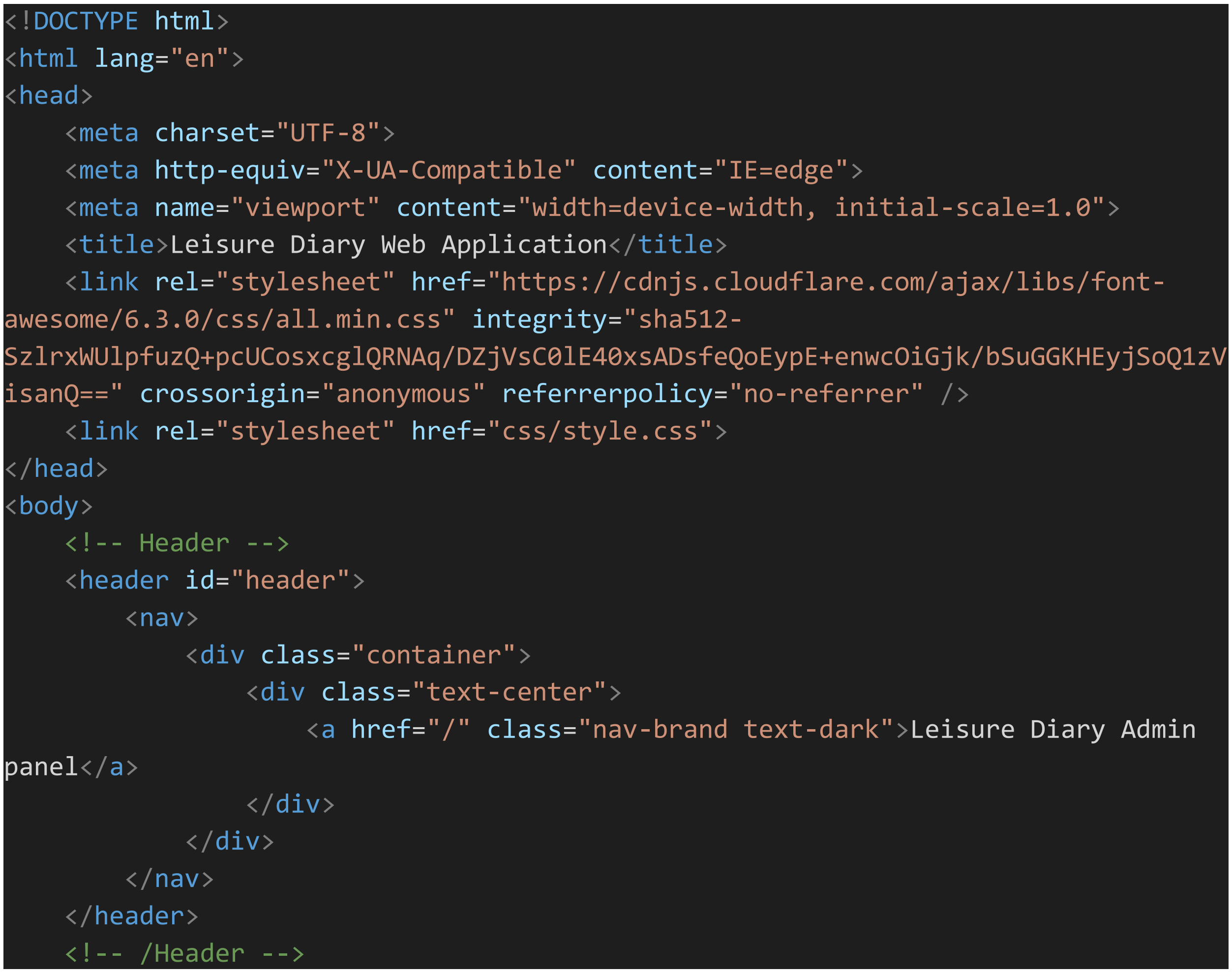
With the help of EJS (Embedded JavaScript) template engine we have refactored out views to make it readable and understandable. It avoids the sophistication of the codes. Each main elements are separated in to files. As in the below example we have separated header and footer into separate files and connected to the “index.html” with “ejs” tags



```
<!-- including Header -->
<%- include('include/_header')%>
<!-- /including Header -->

<!-- Main Secr -->
<main id="site-main">
    <div class="container">
        <div class="box-nav d-flex justify-between">
            <a href="/add-user" class="border-shadow">
                <span class="text-gradient">New User <i class="fa-solid fa-user"></i></span>
            </a>
        </div>
        <!-- Form Handling -->
        <form action="/" method="POST">
            <table class="table">
                <thead class="thead-dark">
                    <tr>

                    </tr>
                </thead>
                <tbody>
                    <%-include('include/_show')%>
                </tbody>
            </table>
        </form>

    </div>
</main>

<!-- including _footer -->
<%- include('include/_footer')%>
<!-- /including _footer -->
```

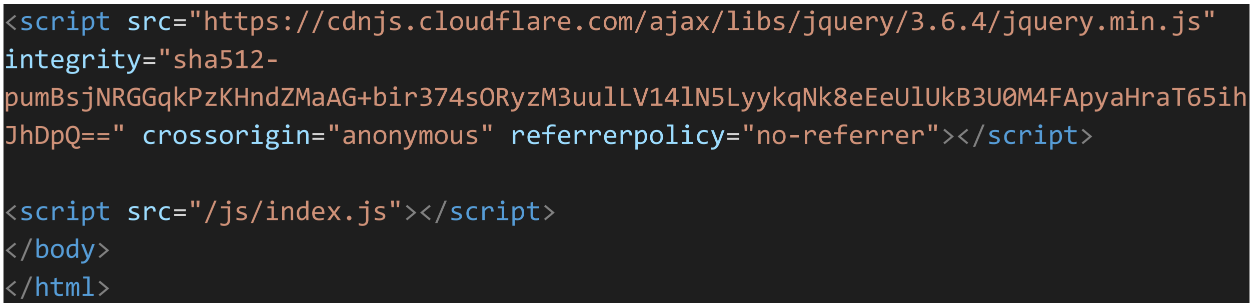
Figure 22 HTML Views

Header & Footer



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Leisure Diary Web Application</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.3.0/css/all.min.css" integrity="sha512-
SzlrxWUlpfuzQ+pcUCosxcglQRNAq/DZjVsC0lE40xsADsfeQoEypE+enwcOiGjk/bSuGGKHEyjSoQ1zV
isanQ==" crossorigin="anonymous" referrerpolicy="no-referrer" />
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <!-- Header -->
    <header id="header">
        <nav>
            <div class="container">
                <div class="text-center">
                    <a href="/" class="nav-brand text-dark">Leisure Diary Admin
panel</a>
                </div>
            </div>
        </nav>
    </header>
    <!-- /Header -->
```

Figure 23 EJS Templating



```html
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.4/jquery.min.js"
integrity="sha512-
pumBsjNRGGqkPzKHndZMaAG+bir374sORyzM3uulLV14lN5LyykqNk8eEeUlUkB3U0M4FApyaHraT65ih
JhDpQ==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>

<script src="/js/index.js"></script>
</body>
</html>
```

Figure 24 EJS Templating

Mobile Application

Primary technology used to build the mobile application Flutter framework which uses dart language. Flutter is an open-source mobile application development framework created by Google. It allows developers to build native apps for iOS, Android, web, and desktop platforms using a single codebase.

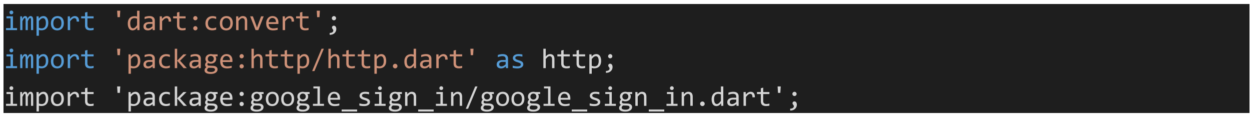
Flutter uses the Dart programming language, which is also developed by Google, and provides a rich set of customizable widgets, tools, and libraries that make it easy to create beautiful and responsive user interfaces. It also has a fast development cycle, thanks to features like hot reload, which allows developers to see changes in the code immediately without restarting the app.

One of the key features of Flutter is its performance. It uses a high-performance rendering engine that allows apps to run at 60 frames per second, resulting in smooth animations and a great user experience. Flutter also provides access to native features and APIs, making it easy to integrate with other native code.

Flutter is also popular among developers because of its strong community support and extensive documentation. It has a large number of packages and plugins available in its ecosystem, making it easy to integrate with other popular libraries and services.

Overall, Flutter is a powerful and flexible framework for building cross-platform apps that offer great performance, native features, and a great developer experience [17].

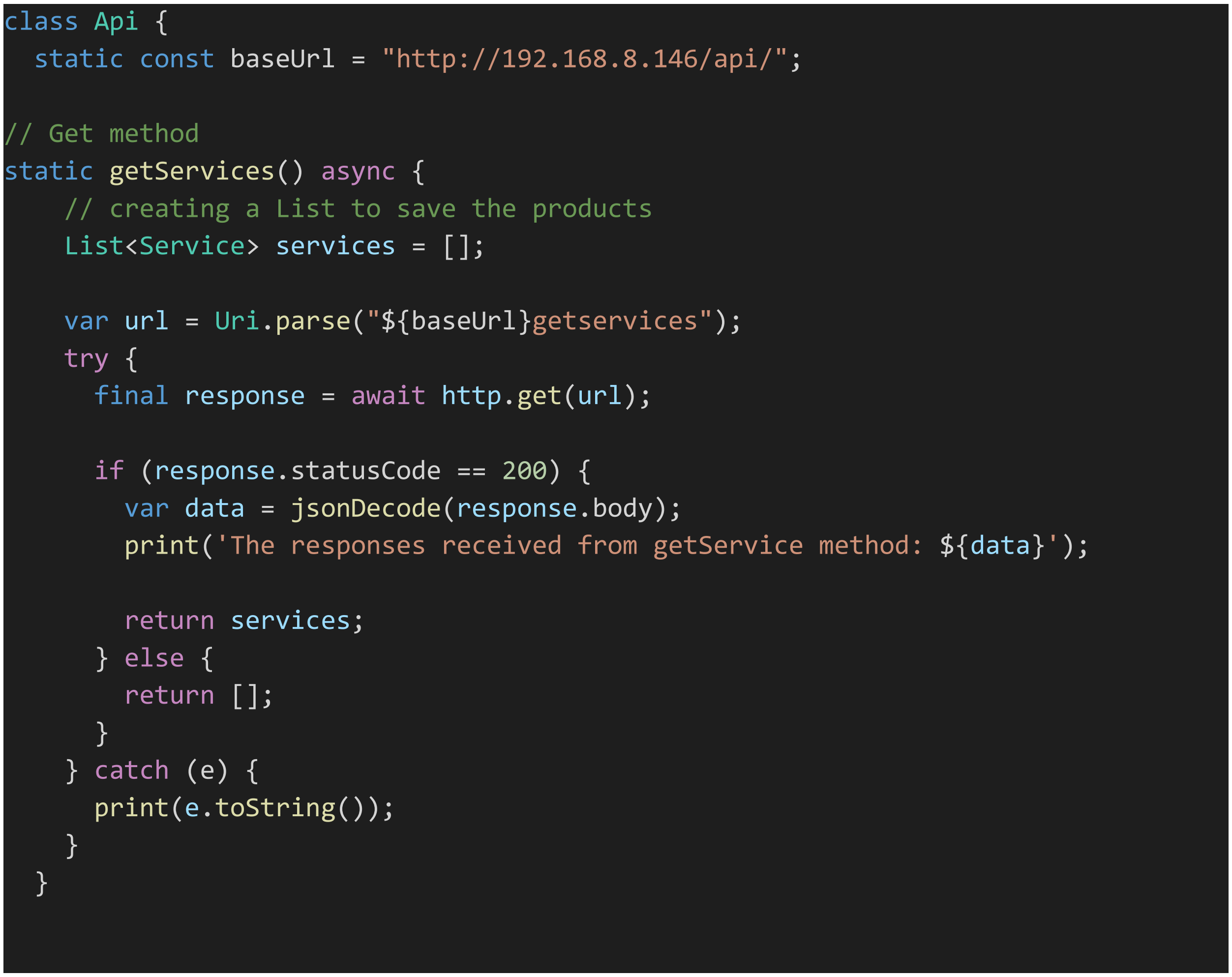
Libraries



```
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:google_sign_in/google_sign_in.dart';
```

Figure 25 Libraries

API Endpoint integration



```
class Api {
  static const baseUrl = "http://192.168.8.146/api/";

// Get method
static getServices() async {
    // creating a List to save the products
    List<Service> services = [];

    var url = Uri.parse("${baseUrl}getservices");
    try {
      final response = await http.get(url);

      if (response.statusCode == 200) {
        var data = jsonDecode(response.body);
        print('The responses received from getService method: ${data}');

        return services;
      } else {
        return [];
      }
    } catch (e) {
      print(e.toString());
    }
  }
```

Figure 26 API Integration

Initially, there were some problems faced during implementation of the system. Such as passing data through API end points (Web application and Mobile application), Connecting the emulators proxy to current PC's IP address, Json data converting error in mongoDB, later solved by a library called bodyPaser.

5. Validation

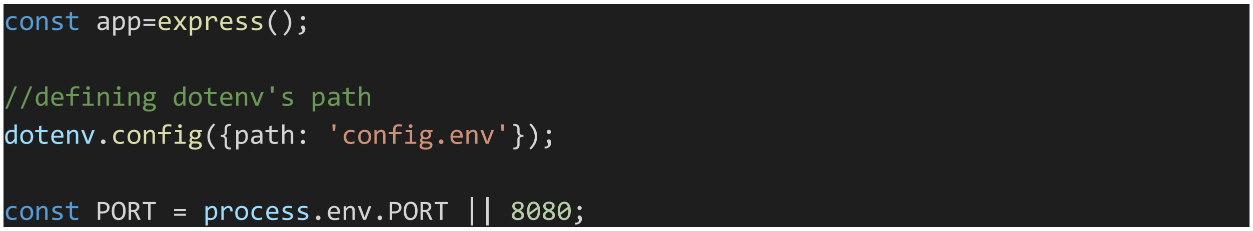
Validation is the process of documenting the systems accuracy which meets the exact requirements. It is documented and reviewed using different tools and techniques. Such process ensures the efficiency of the system, helps to maintain lower cost on system implementation, etc.

There are various tools to automate the testing process such as The Katalon Platform and continuous testing, Selenium and continuous testing, Appium and continuous testing etc. Currently we have used some manual testing methodologies to check our implantation and further automation tool will be used and automate the process.

Validations

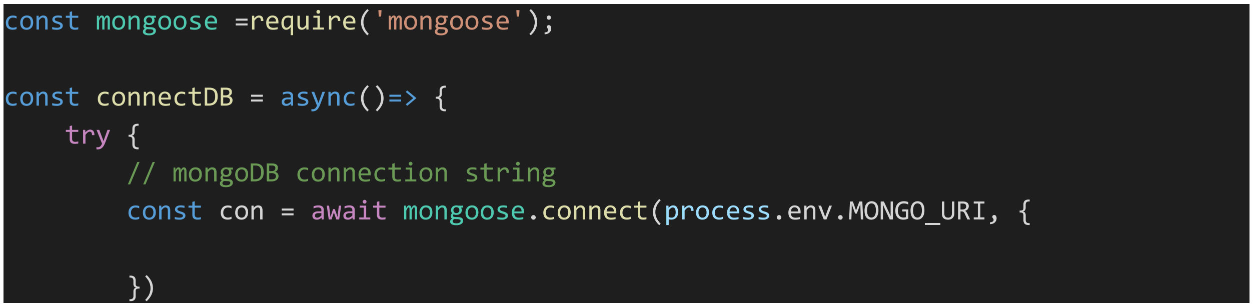
ENV Files- In general scenario the code would be share among the developers and testers. The implementation will consist with confidential credentials. To avoid such conflicts we can abstract our confidential credentials in an ENV file and share the codes without the ENV file. Receiver will have to configure an ENV file as their need.

Hiding the PORT details



```
const app=express();

//defining dotenv's path
dotenv.config({path: 'config.env'});

const PORT = process.env.PORT || 8080;
```

Hiding the Database connection details.



```
const mongoose =require('mongoose');

const connectDB = async()=> {
    try {
        // mongoDB connection string
        const con = await mongoose.connect(process.env.MONGO_URI, {

        })
```

ENV Configuration file

Config.env

```
PORT=3000

MONGO_URI=mongodb+srv://admin:admin123@cluster0.8u6fr2n.mongodb.net/users?retryWrites=true&w=majority
```

Figure 27 Validations

Manual Test Cases

Test case #: 1

Module: Database Connection

Component: Database Connection

Scenario: When the server starts, server should automatically connect to the DB and console should print "Mongo DB is running with its token".

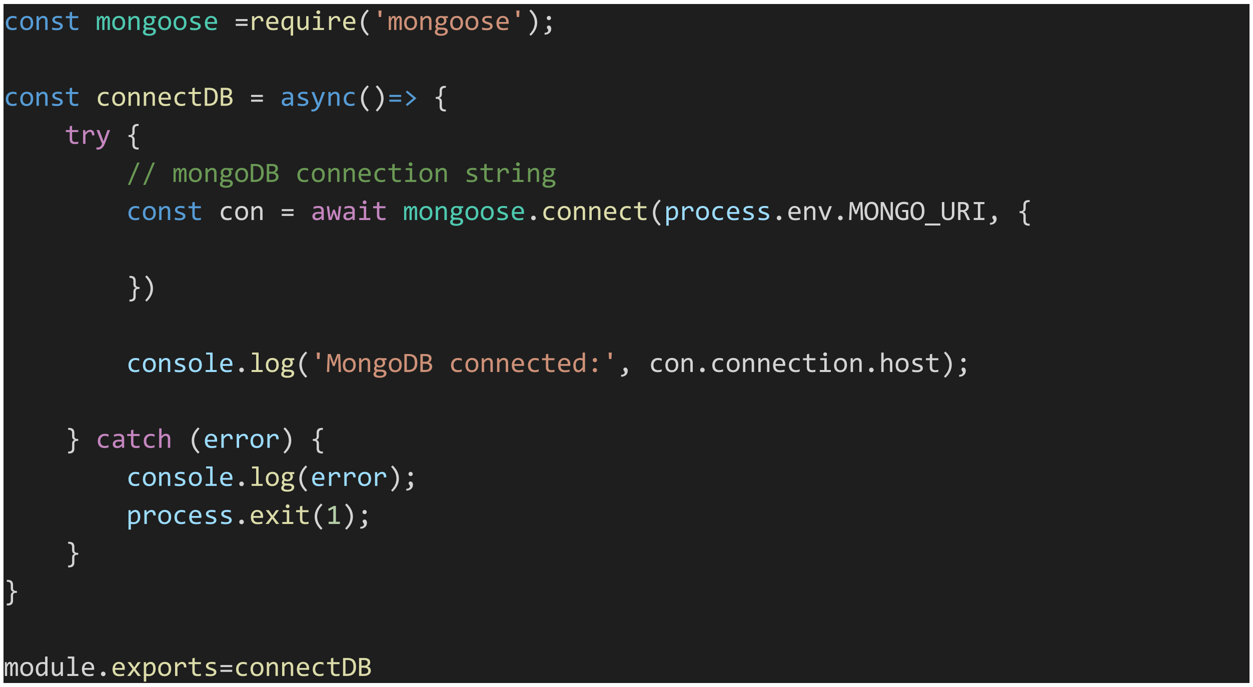
Preconditions:

1. Cloud database connected to the IP.

Expected Results:

1. Database automatically connects to the server.
2. Console should print "Mongo DB connected successfully".

Database connection implementation

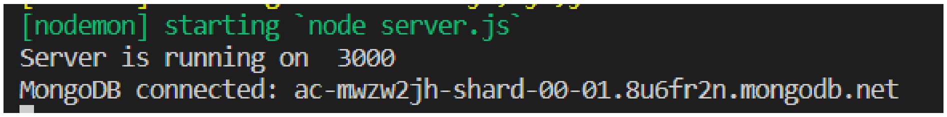


```
const mongoose =require('mongoose');

const connectDB = async()=> {
    try {
        // mongoDB connection string
        const con = await mongoose.connect(process.env.MONGO_URI, {

        })

        console.log('MongoDB connected:', con.connection.host);

    } catch (error) {
        console.log(error);
        process.exit(1);
    }
}

module.exports=connectDB
```

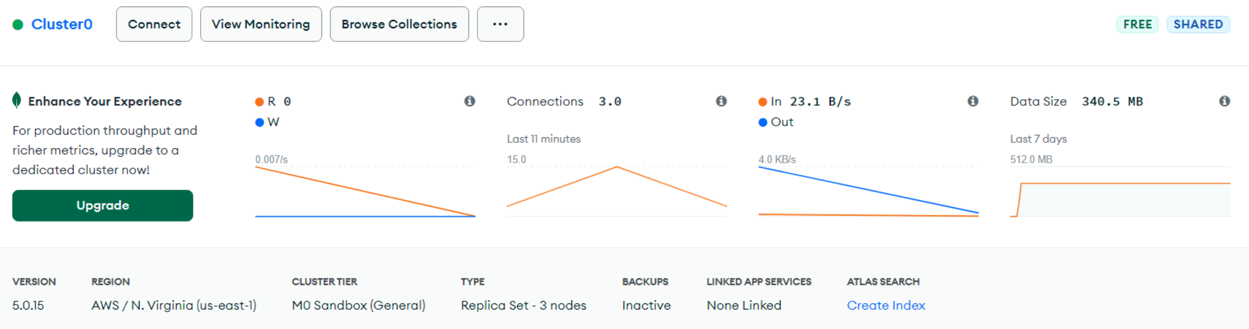
Figure 28 Testing

Figure 29 DB connection code

Figure 30 Test case



```
[nodemon] starting `node server.js`
Server is running on  3000
MongoDB connected: ac-mwzw2jh-shard-00-01.8u6fr2n.mongodb.net
```



VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED APP SERVICES	ATLAS SEARCH
5.0.15	AWS / N. Virginia (us-east-1)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	None Linked	Create Index

Figure 31MongoDB connection

Test case #: 2

Module: Database Connection

Component: Database Connection

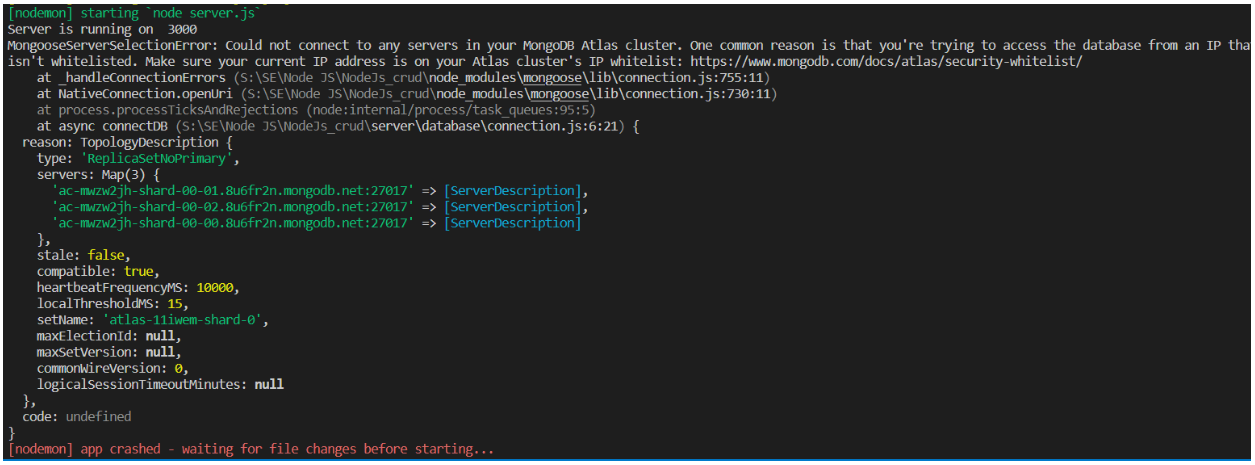
Scenario: When the server failed to connect to the Database.

Preconditions:

1. Cloud database connected to the IP.

Expected Results:

1. Database automatically connects to the server.
2. Console should print error message.



```
[nodemon] starting `node server.js`
Server is running on  3000
MongooseServerSelectionError: Could not connect to any servers in your MongoDB Atlas cluster. One common reason is that you're trying to access the database from an IP tha
isn't whitelisted. Make sure your current IP address is on your Atlas cluster's IP whitelist: https://www.mongodb.com/docs/atlas/security-whitelist/
    at _handleConnectionErrors (S:\SE\Node JS\NodeJs_crud\node_modules\mongoose\lib\connection.js:755:11)
    at NativeConnection.openUri (S:\SE\Node JS\NodeJs_crud\node_modules\mongoose\lib\connection.js:730:11)
    at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
    at async connectDB (S:\SE\Node JS\NodeJs_crud\server\database\connection.js:6:21) {
  reason: TopologyDescription {
    type: 'ReplicaSetNoPrimary',
    servers: Map(3) {
      'ac-mwzw2jh-shard-00-01.8u6fr2n.mongodb.net:27017' => [ServerDescription],
      'ac-mwzw2jh-shard-00-02.8u6fr2n.mongodb.net:27017' => [ServerDescription],
      'ac-mwzw2jh-shard-00-00.8u6fr2n.mongodb.net:27017' => [ServerDescription]
    },
    stale: false,
    compatible: true,
    heartbeatFrequencyMS: 10000,
    localThresholdMS: 15,
    setName: 'atlas-11iwem-shard-0',
    maxElectionId: null,
    maxSetVersion: null,
    commonWireVersion: 0,
    logicalSessionTimeoutMinutes: null
  },
  code: undefined
}
[nodemon] app crashed - waiting for file changes before starting...
```

Figure 32 DB connection 2

Test case #:

Module: API Connection

Component: API Get

Scenario: When we send a Get request to the route "/" it should return the status code "200".

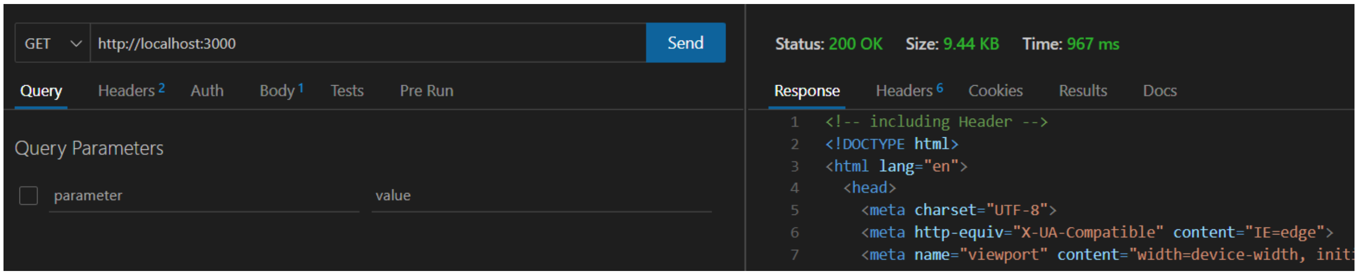
Preconditions:

1. Server and API up and running.

Expected Results:

1. Status code 200.
2. All the data from API body.

Output



```
<!-- including Header -->
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, init
```

Figure 33- API testing

6. Critical Review & Conclusion

6.1 Closing executive summary

In conclusion, the development of our application and the web is in the process of development. Through the system development, it has always made sure to follow the best practices to make the application user friendly. The application includes range of features as elaborated in the functional requirement phase. The application is being tested manually and automatically using various tools to make sure the users will be getting the error free best version of the application. Such testing brings us the confident in launching the application to the market.
Furthermore, application is being implemented under continues monitoring to provide a best version that will satisfy either party. Also, system will be undergone to get user feedbacks and make improvements where ever it is necessary. As well as we will also explore the opportunities to improve the applications features and benefits. In Addition, followings have been identified as potential evolutions to improve the application,

- Create trips and connect their friends
- Interacting option with their Travelers and service providers.
- AI to suggest options
- Language translator.
- Current update about the locations (Weather)
- Accounting system to track the spending and provide reports to others for transparent purposes.

6.2 Conclusion

Traveling is a globally fast-growing industry. Which benefits the travelers by fulfilling their needs and wants on travel planning and organizing such as finding required needs and navigating them to convenient path. Moreover, the related service providing businesses also benefits by interconnecting the travelers to reach their services. People always vote for the convenience.

Majority of tourists prefer to use online platforms for travel planning and organizing via websites, social media contents and travel agencies. Also, their study indicates that travelers who have used online platforms to plan and organize their trips are more satisfied. Service providers can sustain their business if the travelers ended their journey with satisfaction.

References

[1] N. F. P. L. S. K. a. T. S. N. Tavichaiyuth, Covid-19 Travel Planner Mobile Application Design with Lean Product Process Framework, 2022.

[2] M. K. Pratt, "Tech Target," [Online]. Available: https://www.techtarget.com/searchcio/definition/project-scope.

[3] D. &. C. L. Wang, "Online travel review and hotel choice," *Annals of Tourism Research,* pp. 42, 214-216., 2013.

[4] C. K. L. Y. K. &. L. B. Lee, " Understanding the factors influencing tourists' decision-making process. Journal of Travel Research,," Vols. 61-77, p. 57(1), 2018.

[5] C. M. K. &. W. M. Deegan, "The impact of travel planning on travel experience," *Journal of Travel Research,* Vols. 745-757, p. 56(6), 2017.

[6] H. T. &. T. A. D. Vu, " Factors influencing tourists' intention to engage in sustainable travel behavior," *An integrated model. Journal of Sustainable Tourism,* Vols. 1352-1372, p. 26(8), 2018.

[7] D. &. L. R. Buhalis, "Progress in information technology and tourism management," *20 years on and 10 years after the Internet—The state of eTourism research. Tourism Management,,* Vols. 609-623, p. 29(4), 2008.

[8] X. L. L. &. Y. L. Yu, "A study of Chinese outbound tourists," *Journal of Hospitality and Tourism Technology,* vol. 9(1), pp. 68-85, 2018.

[9] Z. D. Q. &. M. Y. Xiang, "Travelers' e-word-of-mouth communication on social networking sites," *An empirical study of Chinese tourists. Journal of Travel Research,* vol. 56(6), pp. 754-767, 2017.

[10] M. &. L. H. Kim, "Determinants of users' continued use of travel apps," *The moderating effect of privacy concern. Journal of Travel Research,* vol. 57(8), pp. 1066-1080, 2018.

[11] P. R. S., "Software engineering: a practitioner's approach. McGraw Hill Education.," 2014.

[Online].

[12] S. W. Ambler, ". The elements of UML 2.0 style. Cambridge University Press.," 2002.

[13] R. Jeffries, "User stories applied. Cigital Incorporated.," 2001. [Online].

[14] G. &. S. I. Kotonya, "Requirements engineering: Processes and techniques. John Wiley & Sons.," 1998. [Online].

[15] Lucidchart, "lucidchart," 2022. [Online]. Available: https://www.lucidchart.com/pages/what-is-a-flowchart.

[16] B. Hyman, "System Architecture," 2012.

[17] Flutter, "flutter.dev," [Online]. Available: https://flutter.dev/.