PROJECT REPORT

ON

Highway Alerts

FOR

Maharashtra Education Society's
Institute of Management And Career
Courses

BY

Shubham Sanjay Mali

SAVITRIBAI PHULE PUNE UNIVERSITY

MASTER IN COMPUTER APPLICATION

MAHARASHTRA EDUCATION SOCIETY's

INSTITUTE OF MANAGEMENT AND CAREER COURSES

(IMCC), PUNE-411038

2023-24

Maharashtra Education Society's

Institute of Management & Career Courses (IMCC)

Approved by AICTE and Recognized by Savitribai Phule Pune University, Pune

NBA Accredited MCA Program (Valid up to 30.06.2022)

IMCC Campus, 131, Mayur Colony, Kothrud, Pune 411038, Maharashtra, India | Ph.: 020-2546 3453 / 6271 / 73 | e-mail: info.imcc@mespune.in

Ref. No. MES IMCC / 399/ 2023 – 24/ **Date:13/04/2024**

CERTIFICATE

This is to certify that the Project Report entitled

“Highway Alerts”

is prepared by

Shubham Mali

M.C.A. Semester IV Course for the Academic Year 2023-24 at M.E. Society's Institute of Management & Career Courses (IMCC), Pune - 411038.
M.C.A Course is affiliated to Savitribai Phule Pune University.

To the best of our knowledge, this is original study done by the said student and important sources used by him/her have been duly acknowledged in this report.

The report is submitted in partial fulfillment of M.C.A Course for the Academic Year 2023-224 as per the rules and prescribed guidelines of Savitribai Phule Pune University.

Dr. Ravikant Zirmite
Head, Dept of MCA
MES IMCC

Dr. Santosh Deshpande
Director,
MES IMCC

Internal Examiner

External Examiner

CERTIFICATE

This is to certify that the student **Shubham Mali** has completed the project work entitled "***Highway Alerts***" under my guidance. The report is submitted in partial fulfillment of M.C.A. Course for the Academic Year 2023-2024 as per the rules & prescribed guidelines of Savitribai Phule Pune University.

His work is found to be satisfactory and complete in all respects.

Guide Name
Mrs. Darshana Yadav

Acknowledgement

I extend my heartfelt gratitude to the esteemed members of IMCC, whose support and guidance have been instrumental in the successful completion project.

I would like to express my sincere appreciation to Dr. Santosh Deshpande Sir, Director of IMCC, for his vision and leadership that provided the foundation endeavor.

My sincere thanks to Dr. Manasi S. Bhate Mam, Deputy Director of IMCC, for her continuous encouragement and support throughout the duration of project.

I am deeply indebted to Dr. Ravikant Zirmite Sir, Head of the Department of IMCC, for his invaluable advice, constructive criticism, and unwavering encouragement which significantly contributed to the refinement of this work.

I am also grateful to Mrs. Darshana Yadav Mam, our internal guide, for her mentorship, patience, and invaluable suggestions that guided me through the challenges encountered during the course of this project.

Furthermore, I extend my appreciation to all the faculty members, colleagues, and friends who provided support and encouragement throughout this journey.

Thank you all for your unwavering support and belief in my abilities.

Sincerely,

Shubham Mali

INDEX

1. Introduction

1.1 Institute Profile

Institute of Management and Career Courses (IMCC) is a premier Management Institute, established in 1983 by Maharashtra Education Society (MES) for providing quality education and technical expertise at the Post Graduation Level in the Fields of Computers and Management. The Institute is recognized by SPPU under Section 46 of Pune University Act, 1974 and Section 85 of Maharashtra University Act, 1994 and Approved by AICTE New Delhi to conduct MCA and MBA programmes. The Institute is located at 131, Mayur Colony, Kothrud, Pune-411038 having 30,000 sq. ft-built area & totally independent campus. IMCC is recognised as a Ph.D. Research Centre under the Faculty of Management, SPPU. IMCC has 38 years standing & it is well known for its conducive educational atmosphere. IMCC focuses on the all-round development of its students. Thus, apart from excellence in academics, students develop their inner potential by way of active participation in co-curricular & extra-curricular activities. IMCC has developed excellent rapport with Industry by way of Guest Lectures, Seminars, Workshops, Industrial Visits & Placements. The main motto of the Institute is to in still the concepts of total personality development in the students. The emphasis is laid on 'Teacher Disciple Relationship' in place of 'Boss Subordinate' relationship at their assignments. The preamble of IMCC ``FACTA-NON-VERBA" lucidly means that the Institute produces the new breed of professionals, who’s deeds will speak and there could be no requirement of pomposity. The 2 zooming enthusiastic, rational, and excellent external endeavours are being imbibed in the students to prove their mettle. The conducive milieu of the Institute melds the budding managers to reveal in managing flexibility, integration, change and transformation. These ‘would be’ professionals are channelised in such a way to ‘orchestrate’ and deploy business and technological management skills in a synergistic manner to grab the tangible success. The faculty members put their relentless efforts in educating the students to synthesize business management acumen and technology insights in a creative manner.

1.2 Abstract:

The Highway Alerts Android Application is a user-friendly and interactive mobile application designed to assist users in discovering, exploring, and navigating various locations based on their preferences. The application enables users to select specific categories, set location radius, and define minimum ratings to tailor their exploration experience. Additionally, users can register, log in, save favourite locations, and search for places using city names.

Key Features:

- **User Registration and Login:**

-Users are required to register and log in to access the application's features.

-Personalized user profiles allow for a customized experience and the ability to save preferences.

- **Location Selection:**

-Users can choose from a variety of categories such as restaurants, parks, museums, and more.

-The application allows users to specify the radius within which they want to explore locations.

- **Filtering by Rating:**

-Users can set a minimum rating for the locations they are interested in, ensuring a quality experience.

-The filtering mechanism ensures that only locations meeting the specified criteria are displayed.

- **Interactive Map Integration:**

-The heart of the application is an interactive map that displays selected locations.

-Users can navigate through the map, explore available locations, and receive real-time updates as they move.

- **Location Information Prompt:**

-The application calls out selected locations one by one, providing essential details such as distance and star ratings.

-Users are prompted to add a location to the map or skip to the next one.

- **Dynamic Location Updates:**

-As the user changes their location, the application dynamically updates the displayed locations based on the new position.

- **Favourite Feature:**

-Users can mark locations as favourites, allowing for quick access to preferred places in the future.

Problem Statement:

In today's fast-paced world, individuals often face the challenge of safely exploring and discovering nearby locations while driving or traveling. The need to manually search for points of interest on a mobile device poses significant risks, as it diverts attention away from the road and increases the likelihood of accidents. Existing solutions fail to address this critical safety concern, as they require users to interact with their phones manually to access location information. Therefore, there is a pressing need for a mobile application that enables users to explore and browse nearby attractions, restaurants, and points of interest without touching their phones, utilizing voice commands for hands-free interaction. This application should prioritize user safety and convenience, providing a seamless and intuitive experience for drivers and travellers seeking to discover new destinations while minimizing distractions on the road.

1.3 Existing System:

In the absence of the Highway Alerts Android Application, users typically rely on a combination of map applications, online reviews, and manual searches to find and explore locations. Existing systems may lack the integration of personalized preferences, real-time dynamic updates, and an interactive exploration experience. Users often face challenges in efficiently discovering places based on specific criteria, leading to a less streamlined and engaging exploration process. Additionally, without a dedicated system, users may find it cumbersome to seamlessly transition between planning, navigating, and interacting with their chosen locations.

Need for System:

The development of the Highway Alerts Android Application addresses several shortcomings present in the existing systems:

- **Personalized Exploration:**

Existing systems may not offer the level of personalization that the Highway Alerts app provides. The ability to choose specific categories, set radius and rating preferences tailors the exploration experience to individual user preferences.

- **Real-time Updates:**

The Highway Alerts app's integration with an interactive map ensures real-time updates and dynamic changes as users move, providing a more responsive and immersive experience.

- **Efficient Location Discovery:**

The application streamlines the location discovery process by presenting information in a structured manner. Users can filter and select locations based on their preferences, significantly reducing the time and effort required to find suitable places.

- **User Engagement:**

The need for a system like Highway Alerts arises from the desire to enhance user engagement during location exploration. The app prompts users to interact with each location, fostering a more engaging and interactive experience.

- **Integrated Navigation:**

Unlike existing systems, the Highway Alerts app seamlessly integrates navigation features, allowing users to not only discover locations but also navigate to them efficiently.

- **Favourites and User Profiles:**

The ability to save favourite locations and maintain user profiles caters to the need for a personalized experience, enabling users to revisit and explore their preferred places effortlessly.

In summary, the Highway Alerts Android Application addresses the limitations of existing systems by offering a comprehensive and user-centric solution for location exploration.

1.4 Scope of System:

The Highway Alerts Android Application aims to provide users with a robust and versatile platform for discovering, exploring, and navigating various locations based on their preferences. The scope of the system encompasses a wide range of features and functionalities, ensuring a comprehensive and user-centric experience:

1. **User Registration and Authentication:**

The system allows users to register and create personalized profiles, ensuring secure authentication and access control.

2. **Location Selection and Categorization:**

Users can choose from a diverse set of categories such as restaurants, parks, museums, etc., providing a comprehensive range of locations to explore.

3. **Dynamic Filtering and Sorting:**

The application enables users to dynamically filter locations based on criteria such as radius and minimum ratings, ensuring that only relevant and preferred places are displayed.

4. **Interactive Map Integration:**

An interactive map serves as the central component, displaying selected locations and providing users with a visual representation of their exploration journey.

5. **Real-time Location Updates:**

The system continually updates location information based on the user's movement, ensuring that the displayed locations remain current and accurate.

6. **Location Information Prompt:**

Users receive detailed information about selected locations, including distance and star ratings. The system prompts users to add locations to the map or skip to the next one.

7. **Favourites and User Profiles:**

Users can mark locations as favourites, facilitating quick access to preferred places. User profiles store personalized preferences, enhancing the overall user experience.

8. **Navigation Integration:**

The system seamlessly integrates navigation features, allowing users to navigate to their chosen locations directly from the application.

9. User Engagement and Feedback:

The application encourages user engagement through interactive prompts and feedback mechanisms, enhancing the overall user experience and satisfaction.

10. Scalability:

The system is designed to be scalable, accommodating potential future enhancements, additional features, and an expanding user base.

11. Security Measures:

Robust security measures are implemented to safeguard user data, ensuring privacy and the secure handling of sensitive information.

12. Cross-Platform Compatibility:

The application is developed for the Android platform, but the scope includes the potential for future expansion to other platforms, providing a broader user reach.

The scope of the Highway Alerts Android Application is comprehensive, offering a feature-rich and dynamic platform that caters to users diverse needs for location exploration and discovery. The system is designed to be scalable, user-friendly, and adaptable to evolving requirements in the realm of location-based services.

1.5 Operating Environment - Hardware:

The Highway Alerts Android Application is designed to run on standard Android-powered devices. The hardware specifications for optimal performance are as follows:

Processor: Quad-core or higher processor for smooth navigation and real-time updates.

RAM: Minimum 2GB RAM to ensure responsiveness and seamless operation.

Storage: Adequate storage space for the application installation and caching of map data. A minimum of 16GB internal storage is recommended.

GPS Module: Devices should be equipped with a GPS module or location services for accurate location tracking.

Operating Environment - Software

The Highway Alerts Android Application is developed to operate within the Android ecosystem, leveraging the Android operating system and associated software libraries. The software environment includes:

Android Operating System:

The application is compatible with Android OS versions 6.0 (Marshmallow) and above to ensure widespread compatibility with a variety of devices.

Development Framework:

The application is developed using the Android Studio Integrated Development Environment (IDE), which supports the creation of feature-rich and responsive Android applications.

Mapping and Location Services:

Integration with mapping and location services such as Google Maps API is essential for the core functionality of the application. This includes real-time map rendering, location tracking, and navigation features.

Backend Server:

The system relies on a backend server to manage user accounts, preferences, and location data. The server-side technology may involve frameworks such as Django, Flask, or others, depending on the development stack.

Database Management System (DBMS):

MongoDB, a NoSQL database management system, is utilized to store and retrieve user data, location information, and other relevant details in a scalable and flexible manner.

Network Connectivity:

The application requires a stable internet connection for real-time updates, map data retrieval, and communication with the backend server. Both Wi-Fi and mobile data connectivity are supported.

Security Protocols:

Secure Socket Layer (SSL) or Transport Layer Security (TLS) protocols are implemented to ensure secure communication between the application and the backend server, protecting user data and privacy.

Authentication Services:

Authentication services, possibly utilizing OAuth or token-based authentication, are integrated to secure user accounts and access control.

Notification Services:

The application may utilize push notification services to update users on new features, promotions, or other relevant information.

Cross-browser Compatibility:

While the primary focus is on Android devices, consideration is given to ensuring cross-browser compatibility for potential future expansions to other platforms or devices.

The Highway Alerts Android Application is developed with the aim of providing a seamless and efficient user experience within the specified hardware and software environments. Compatibility with common Android devices and adherence to industry-standard development practices ensure widespread accessibility and reliability.

1.6 Brief Description of Technology Used:

The Highway Alerts Android Application utilizes a modern and efficient tech stack, combining React for the front-end, Node.js with Express.js for the back-end, and MongoDB as the database. This technology combination provides a scalable, responsive, and user-friendly application.

Front-end - React:

React, a JavaScript library for building user interfaces, is employed for the front-end development. React's component-based architecture facilitates the creation of modular and reusable UI elements, contributing to a more maintainable codebase. The use of React also enables the application to deliver a dynamic and responsive user experience.

Back-end - Node.js and Express.js:

Node.js, a server-side JavaScript runtime, is utilized for the back-end development. Its non-blocking, event-driven architecture allows for efficient handling of concurrent connections, making it well-suited for real-time applications. Express.js, a web application framework for Node.js, simplifies the development of robust and scalable APIs, enhancing the overall performance of the back-end.

Database - MongoDB:

MongoDB, a NoSQL database, serves as the data store for the application. MongoDB's document-oriented structure allows for flexible and schema-less data storage, accommodating the dynamic nature of location-based information. The use of MongoDB also facilitates scalability and ease of integration with Node.js.

Operating Systems Used:

Development Environment:

The development environment for the Highway Alerts application is platform-independent and can be set up on both Windows and Unix-based operating systems. Developers can use tools such as Visual Studio Code or Atom for coding, and Node Package Manager (npm) for managing dependencies.

Deployment Environment:

The application is deployable on servers running either Windows or Unix-based operating systems. Popular choices for deployment include Linux distributions (e.g., Ubuntu, CentOS) for Unix-based systems and Windows Server for Windows environments.

Database Management System:

MongoDB (NoSQL):

MongoDB is chosen as the NoSQL database for its ability to handle unstructured data, scalability, and ease of integration with Node.js. The document-oriented nature of MongoDB aligns well with the dynamic and varied data associated with location-based information.

In summary, the Highway Alerts Android Application is built using React for the front-end, Node.js with Express.js for the back-end, and MongoDB as the database. This technology stack ensures a robust, scalable, and responsive solution for users exploring and navigating various locations. The application is designed to be platform-independent during development and deployment, supporting both Windows and Unix-based operating systems.

2. Proposed System

2.1 Study of Similar Systems:

Prior to the development of the Highway Alerts Android Application, a comprehensive study of existing systems and related research papers was conducted to identify similar solutions, analyse their strengths and weaknesses, and gain insights into best practices. This study aimed to inform the design and development process, ensuring that the Highway Alerts app addressed any gaps or limitations present in comparable systems.

Review of Existing Systems:

Several location-based exploration and navigation applications were examined to understand the features, user interfaces, and functionalities they offered. Notable examples included Google Maps, Yelp, TripAdvisor, and other applications that provided users with the ability to discover and explore locations based on various criteria.

Google Maps:

Google Maps was studied for its map integration, real-time updates, and navigation features. While highly popular, its focus on general mapping functionality led to the identification of opportunities for a more specialized and personalized exploration experience.

Yelp and TripAdvisor:

Yelp and TripAdvisor were analysed for their user review systems and categorization of locations. This study helped in understanding user preferences, ratings, and reviews, leading to the incorporation of similar features into the Highway Alerts app.

Research Papers:

Relevant research papers were explored to understand emerging trends and advancements in location-based services, user experience design, and mobile application development. Papers discussing topics such as personalized location recommendations, dynamic mapping, and user engagement in location-based applications were particularly valuable.

Key Insights and Application Enhancements:

The study of similar systems revealed several key insights that influenced the design and development of the Highway Alerts Android Application:

Personalization and User Engagement:

Users appreciate personalized experiences. The incorporation of features such as dynamic filtering, personalized recommendations, and interactive prompts was influenced by successful elements identified in existing applications.

Real-time Updates and Dynamic Mapping:

The emphasis on real-time updates and dynamic mapping was inspired by the findings from research papers and the study of applications providing live location information.

User Reviews and Ratings:

The inclusion of user reviews, ratings, and the ability to set preferences for minimum ratings were informed by the analysis of review-centric platforms like Yelp and TripAdvisor.

Efficient Navigation:

Insights from Google Maps contributed to the focus on efficient navigation within the Highway Alerts app, ensuring users could seamlessly move from exploring locations to navigating to their chosen destinations.

In conclusion, the study of similar systems, including existing applications and research papers, played a crucial role in shaping the design and functionality of the Highway Alerts Android Application. By drawing upon successful elements and addressing limitations identified in comparable solutions, the Highway Alerts app aims to provide an enhanced and tailored location exploration experience for users.

2.2 Feasibility Study:

The feasibility study for the Highway Alerts Android Application encompasses a comprehensive assessment of the project's viability, considering technical, operational, economic, and scheduling aspects. The goal of this study is to determine whether the proposed system is feasible and practical to develop and implement. The key components of the feasibility study include:

Technical Feasibility:

Hardware and Software Compatibility: The chosen technology stack, including React for the front-end, Node.js with Express.js for the back-end, and MongoDB as the database, has been evaluated for compatibility and support. The development environment and deployment platforms, whether Windows or Unix-based, have been considered to ensure technical feasibility.

Integration Capabilities: The seamless integration of the chosen technologies, including mapping and location services, has been verified to confirm that the system components work together efficiently. Compatibility with mobile devices running the Android operating system has also been examined.

Scalability: The architecture of the application has been designed to be scalable, accommodating potential future enhancements and an increasing user base.

Operational Feasibility:

User Acceptance: The target audience for the Highway Alerts app, including individuals who enjoy exploring new locations and seeking personalized recommendations, has been identified. User feedback and usability testing will be conducted during development to ensure the application meets user expectations.

Training Requirements: The operational feasibility includes an assessment of any training needs for end-users, considering the application's user-friendly design. Help guides and tooltips within the application will be provided to assist users in navigating and utilizing features.

Economic Feasibility:

Cost-Benefit Analysis: An in-depth cost-benefit analysis has been conducted, considering development costs, maintenance expenses, potential revenue streams, and the projected return on investment. The analysis indicates that the benefits of the Highway Alerts app outweigh the associated costs.

Resource Availability: The availability of resources, both financial and human, has been considered. Adequate funding and a skilled development team are in place to ensure the successful execution of the project.

Scheduling Feasibility:

Project Timeline: A detailed project timeline has been established, outlining the development phases, testing, and deployment. Regular milestones and checkpoints have been set to monitor progress and ensure that the project stays on schedule.

Development Team Availability: The availability and commitment of the development team have been assessed, and any potential challenges related to resource availability have been addressed.

Legal and Ethical Feasibility:

Compliance: The Highway Alerts app is designed to comply with relevant data protection and privacy regulations. Legal considerations related to user data collection, storage, and usage have been thoroughly examined.

Ethical Considerations: The application's features and functionalities have been designed with ethical considerations in mind, ensuring transparency, user consent, and responsible handling of user data.

In conclusion, the feasibility study for the Highway Alerts Android Application demonstrates that the project is technically, operationally, economically, and scheduling-wise feasible.

2.3 Objectives of the Proposed System:

The objectives of the Highway Alerts Android Application are to create a feature-rich, user-centric, and innovative platform that addresses the needs of users seeking personalized and dynamic location exploration experiences. The proposed system aims to achieve the following key objectives:

Personalized Location Exploration:

Provide users with the ability to personalize their exploration experience by selecting specific categories, setting radius preferences, and defining minimum ratings for locations.

Real-time Updates and Dynamic Mapping:

Enable real-time updates on the interactive map as users explore different locations, ensuring dynamic mapping and live tracking of their journey.

Efficient Navigation:

Facilitate seamless navigation from the exploration phase to reaching selected locations, enhancing the overall efficiency of the user experience.

User Engagement and Interaction:

Promote user engagement through interactive prompts, calling out location details, and seeking user input on adding locations to the map.

User Registration and Profiles:

Implement user registration and login functionality to create personalized profiles, allowing users to save preferences, access favourite locations, and enhance their overall experience.

Favourites Feature:

Enable users to mark locations as favourites, providing a quick and convenient way to revisit preferred places.

Integration with External Services:

Seamlessly integrate with external services, such as mapping APIs and location-based databases, to enhance the accuracy and variety of available locations.

Scalability and Flexibility:

Design the system to be scalable and flexible, accommodating potential future enhancements, new features, and an expanding user base.

Data Security and Privacy:

Ensure robust data security measures to protect user information, comply with privacy regulations, and build trust in the application.

Efficient Backend Functionality:

Implement an efficient backend system using Node.js and Express.js to handle user requests, manage location data, and provide a responsive API for the front-end.

Cross-platform Compatibility:

While initially developed for Android, consider the potential for cross-platform compatibility, allowing the application to expand its user base to other platforms.

Feedback Mechanism:

Implement a feedback mechanism to gather user input, suggestions, and reviews, allowing for continuous improvement and refinement of the application.

Legal and Ethical Compliance:

Ensure compliance with relevant laws and regulations related to data protection, privacy, and ethical considerations in the design and implementation of the application.

By achieving these objectives, the Highway Alerts Android Application aims to offer users a cutting-edge and highly customized experience in discovering, exploring, and navigating diverse locations, thereby establishing itself as a preferred tool for location-based services.

2.4 Users of the System:

The Highway Alerts Android Application is designed to cater to a diverse user base seeking personalized and dynamic location exploration experiences. The primary users of the system include:

General Users:

Everyday individuals who enjoy exploring new locations, trying out different restaurants, discovering parks, museums, and other points of interest. General users use the application to plan outings, find new places of interest, and navigate to chosen locations efficiently.

Tourists:

Travelers and tourists visiting unfamiliar cities or regions who use the application to discover local attractions, find popular restaurants, and navigate through points of interest. The app serves as a valuable tool for tourists to enhance their travel experience and make informed decisions about places to visit.

City Explorers:

Individuals who want to explore their own city or nearby areas more extensively. City explorers use the application to discover hidden gems, events, and local attractions they may not be aware of, contributing to a richer understanding of their surroundings.

Food Enthusiasts:

Users with a specific interest in discovering and trying out new restaurants, cafes, and eateries. The application helps food enthusiasts find establishments based on cuisine preferences, ratings, and proximity, enhancing their culinary experiences.

Tech-savvy Users:

Individuals who are tech-savvy and enjoy utilizing innovative applications for their daily activities. Tech-savvy users appreciate the real-time updates, dynamic mapping, and interactive features offered by the Highway Alerts app.

Users Interested in Recommendations:

Those who rely on recommendations and reviews to make decisions about where to go. The application provides a platform for users to discover popular locations, read reviews, and receive personalized recommendations based on their preferences.

Users Seeking Personalization:

Individuals who value personalized experiences. The application allows users to customize their exploration by selecting specific categories, setting radius preferences, and defining minimum ratings for locations.

Users Planning Group Outings:

Groups of friends or families planning outings together. The application helps in finding locations that cater to the preferences of the entire group, making it easier to plan and enjoy group activities.

By catering to these diverse user profiles, the Highway Alerts Android Application aims to provide a versatile and valuable tool for individuals with varied interests and preferences in location-based services.

3. Analysis and Design

3.1 System Requirements (Functional and Non-Functional requirements)

Functional Requirements:

1. User Registration and Authentication:

- Users should be able to register an account with the application using an email address or social media accounts.
- The system must authenticate users during the login process to ensure secure access to the application's features.

2. .Location Exploration:

- Users should be able to browse and explore various locations based on categories such as restaurants, parks, museums, etc.
- The application must provide options for users to set preferences such as radius for location search and minimum ratings.

3. Search Functionality:

- Users should be able to search for locations using city names, facilitating exploration in specific areas.
- The search feature should provide relevant and accurate results based on user input.

4. Map Integration and Navigation:

- The application must integrate with mapping services to display locations on an interactive map.
- Users should be able to navigate through the map, view location details, and receive real-time updates as they move.

5. Location Information Display:

- The application should provide detailed information about selected locations, including distance, ratings, reviews, and contact details.

6. Favourites Management:

- Users should be able to mark locations as favorites and access them for quick reference.
- The system must allow users to manage their list of favorite locations, including adding, removing, and organizing them.

7. User Account Management:

- Users should have the ability to manage their accounts, including updating profile information, changing passwords, and viewing account activity.

Non-Functional Requirements:

1. Performance:

- The application must be responsive and performant, even under high load conditions, to ensure a smooth user experience.
- The system should provide fast response times for location searches, map rendering, and navigation.

2. Usability:

- The user interface should be intuitive and easy to navigate, catering to users of varying technical abilities.
- The application must provide clear instructions and guidance to users on how to use its features effectively.

3. Reliability:

- The system should be reliable and available whenever users need to access it, minimizing downtime and disruptions.
- Data integrity and consistency must be maintained to ensure accurate location information and user accounts.

4. Security:

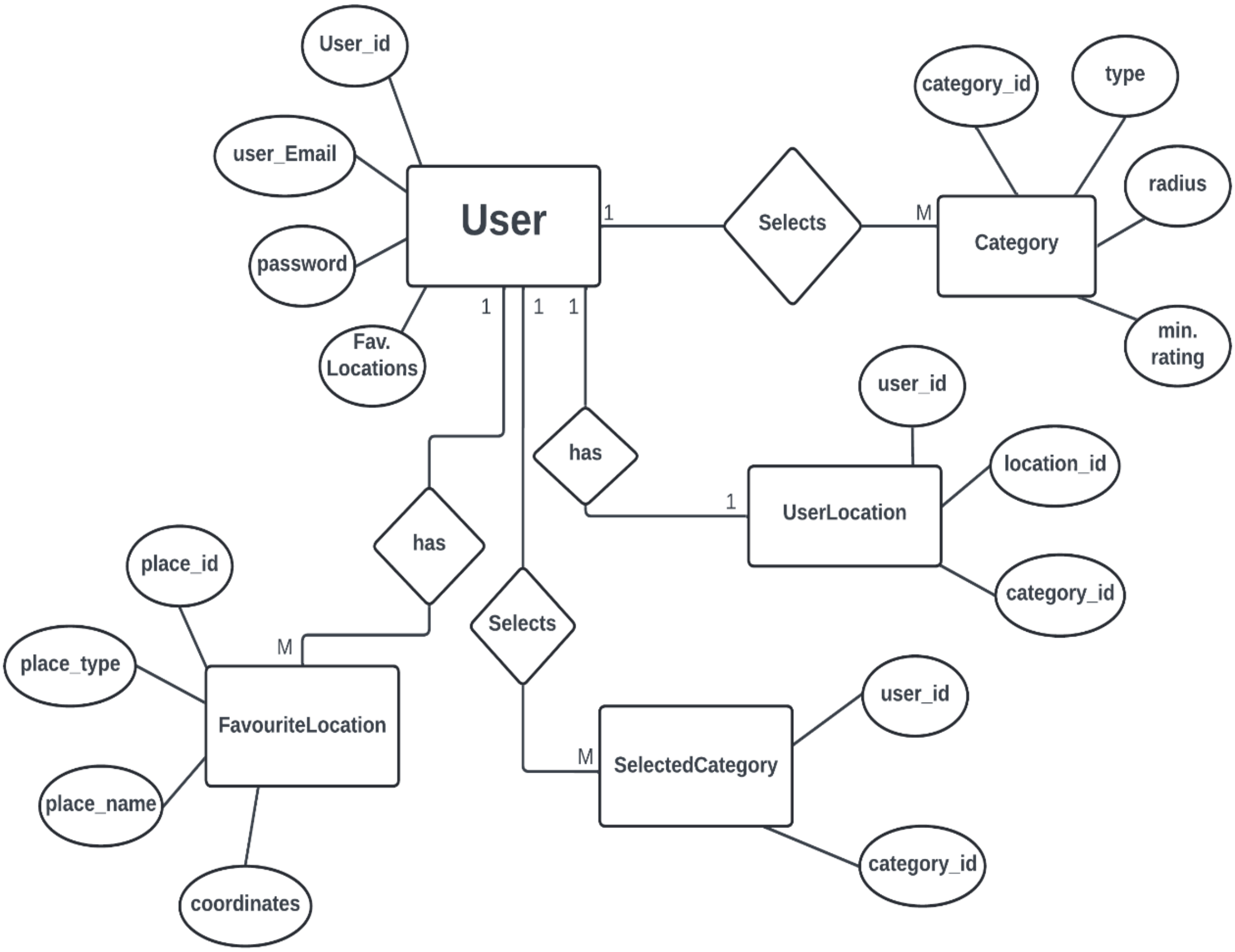
- The application must implement robust security measures to protect user data and privacy.
- User authentication, data encryption, and secure communication protocols should be employed to prevent unauthorized access and data breaches.

5. Compatibility:

- The application should be compatible with a wide range of Android devices running different versions of the operating system.
- Compatibility with popular web browsers and screen sizes should also be ensured for web-based components.

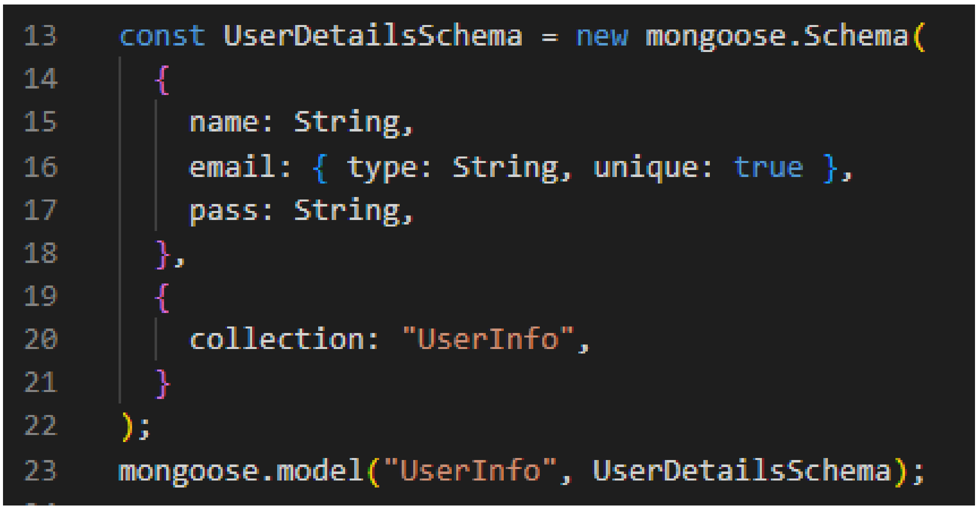
By fulfilling these functional and non-functional requirements, the Highway Alerts Android Application can provide users with a reliable, efficient, and user-friendly platform for exploring and navigating various locations based on their preferences.

3.2 Entity Relationship Diagram (ERD)



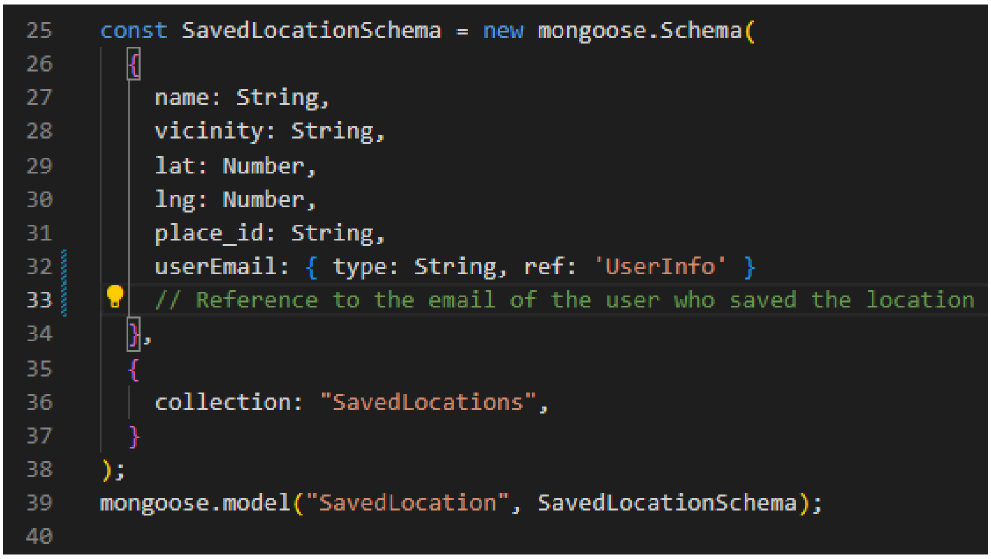
3.3 Table Structure

UserDetails Schema:



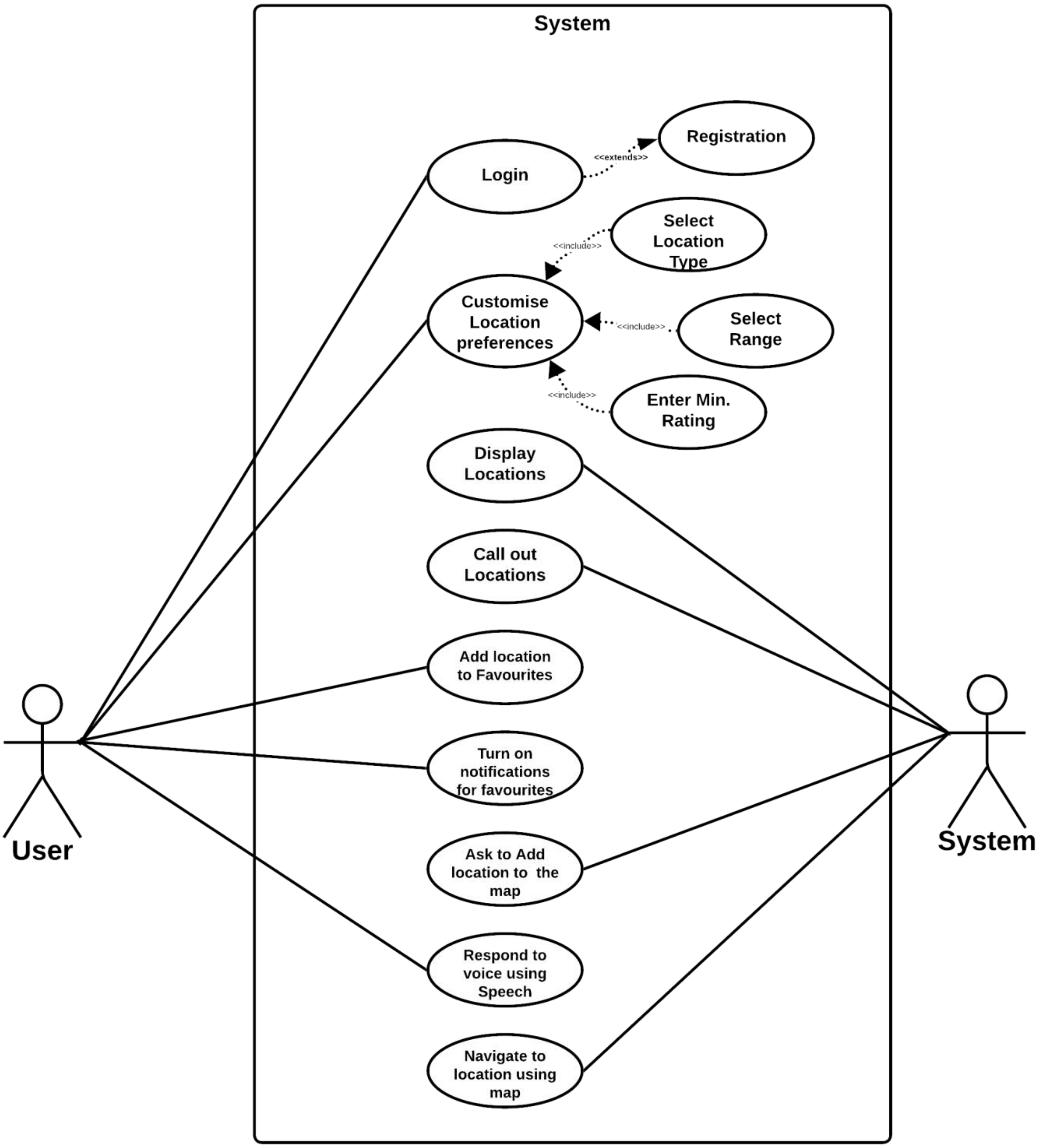
```
const UserDetailsSchema = new mongoose.Schema(
  {
    name: String,
    email: { type: String, unique: true },
    pass: String,
  },
  {
    collection: "UserInfo",
  }
);
mongoose.model("UserInfo", UserDetailsSchema);
```

SavedLocation Schema:



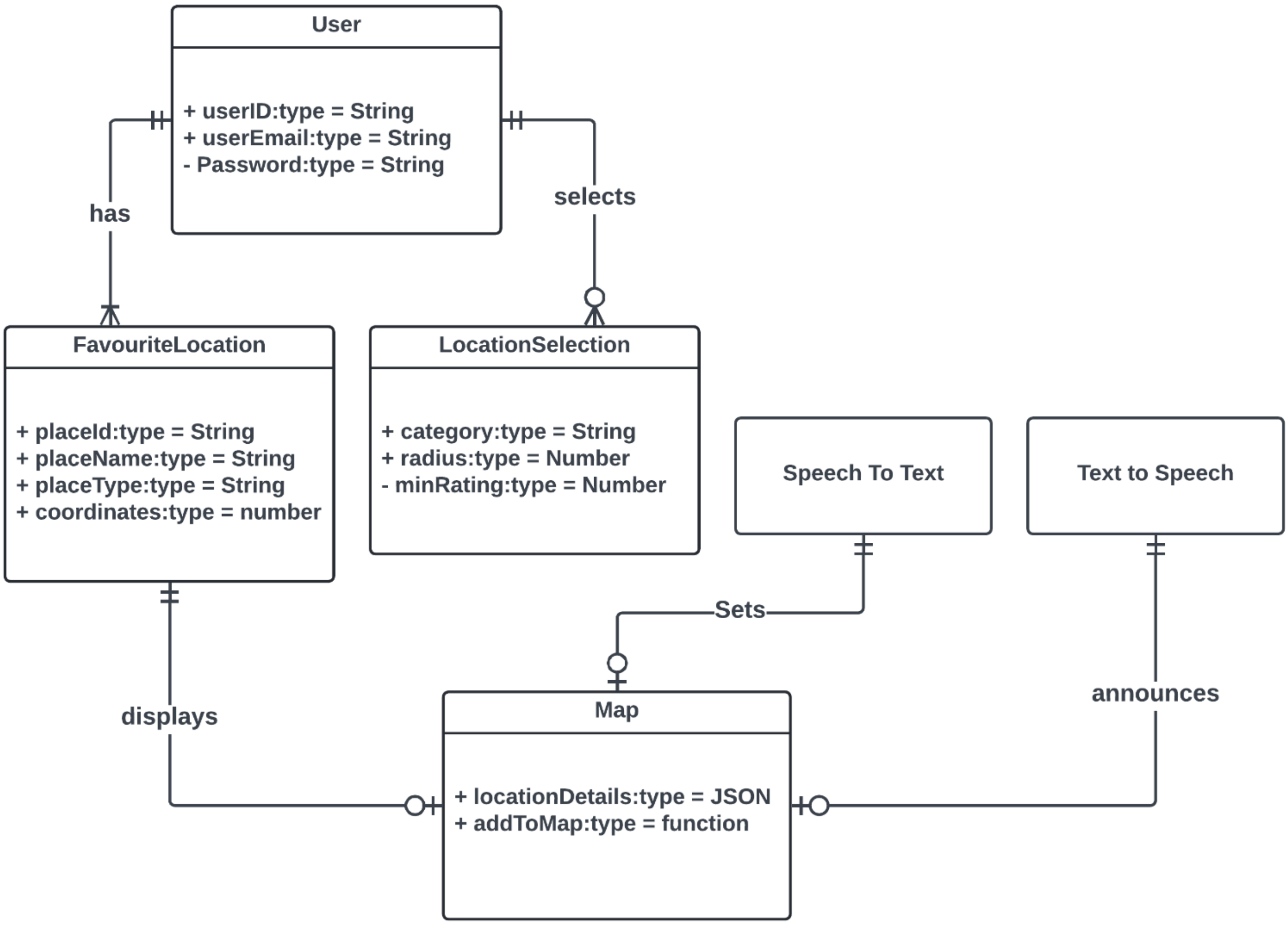
```
const SavedLocationSchema = new mongoose.Schema(
  {
    name: String,
    vicinity: String,
    lat: Number,
    lng: Number,
    place_id: String,
    userEmail: { type: String, ref: 'UserInfo' }
    // Reference to the email of the user who saved the location
  },
  {
    collection: "SavedLocations",
  }
);
mongoose.model("SavedLocation", SavedLocationSchema);
```

3.4 Use Case Diagrams

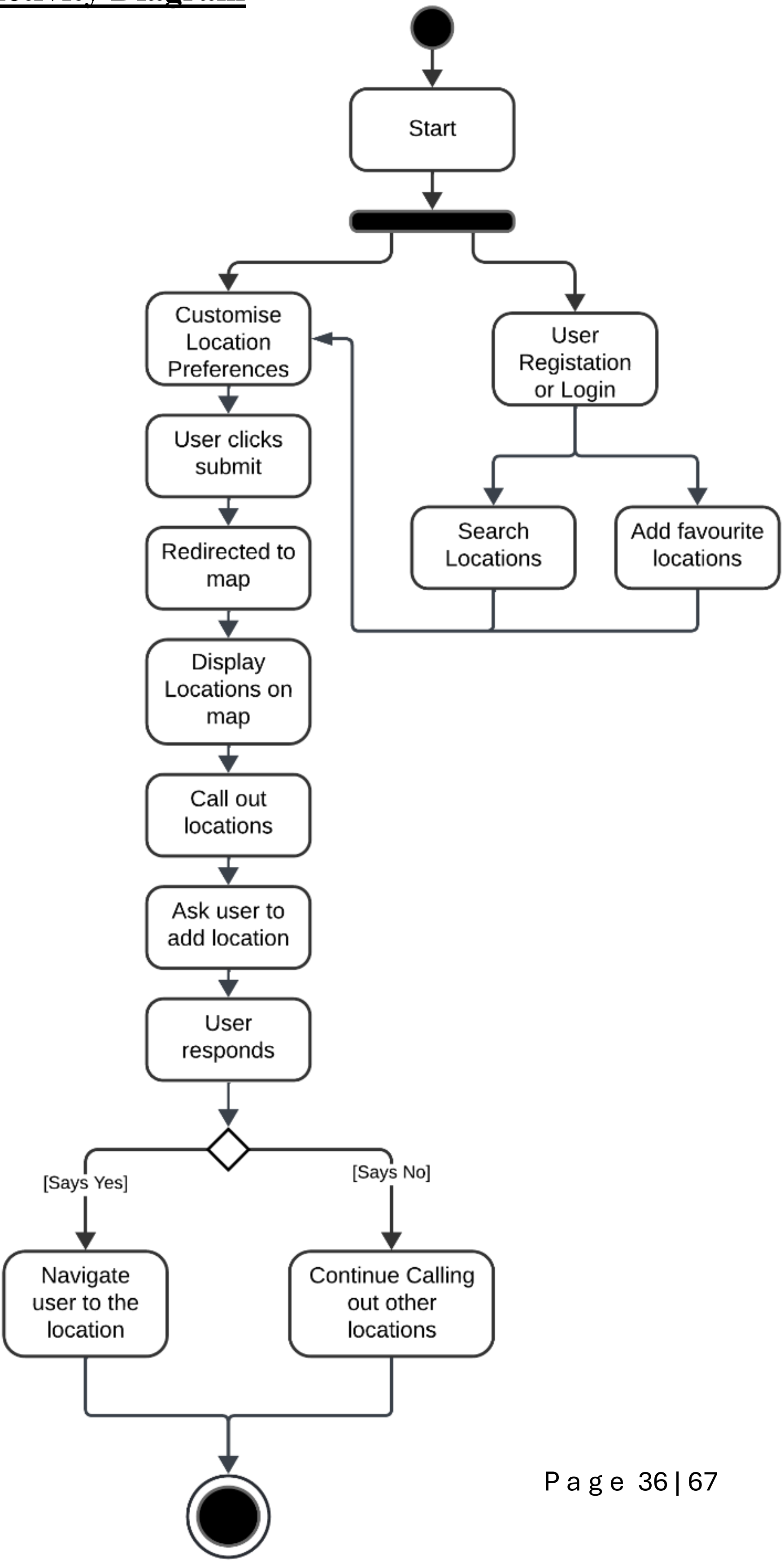


System
Login
Registration
<<extends>>
Select Location Type
<<include>>
Customise Location preferences
<<include>>
Select Range
<<include>>
Enter Min. Rating
Display Locations
Call out Locations
Add location to Favourites
Turn on notifications for favourites
Ask to Add location to the map
Respond to voice using Speech
Navigate to location using map
User
System

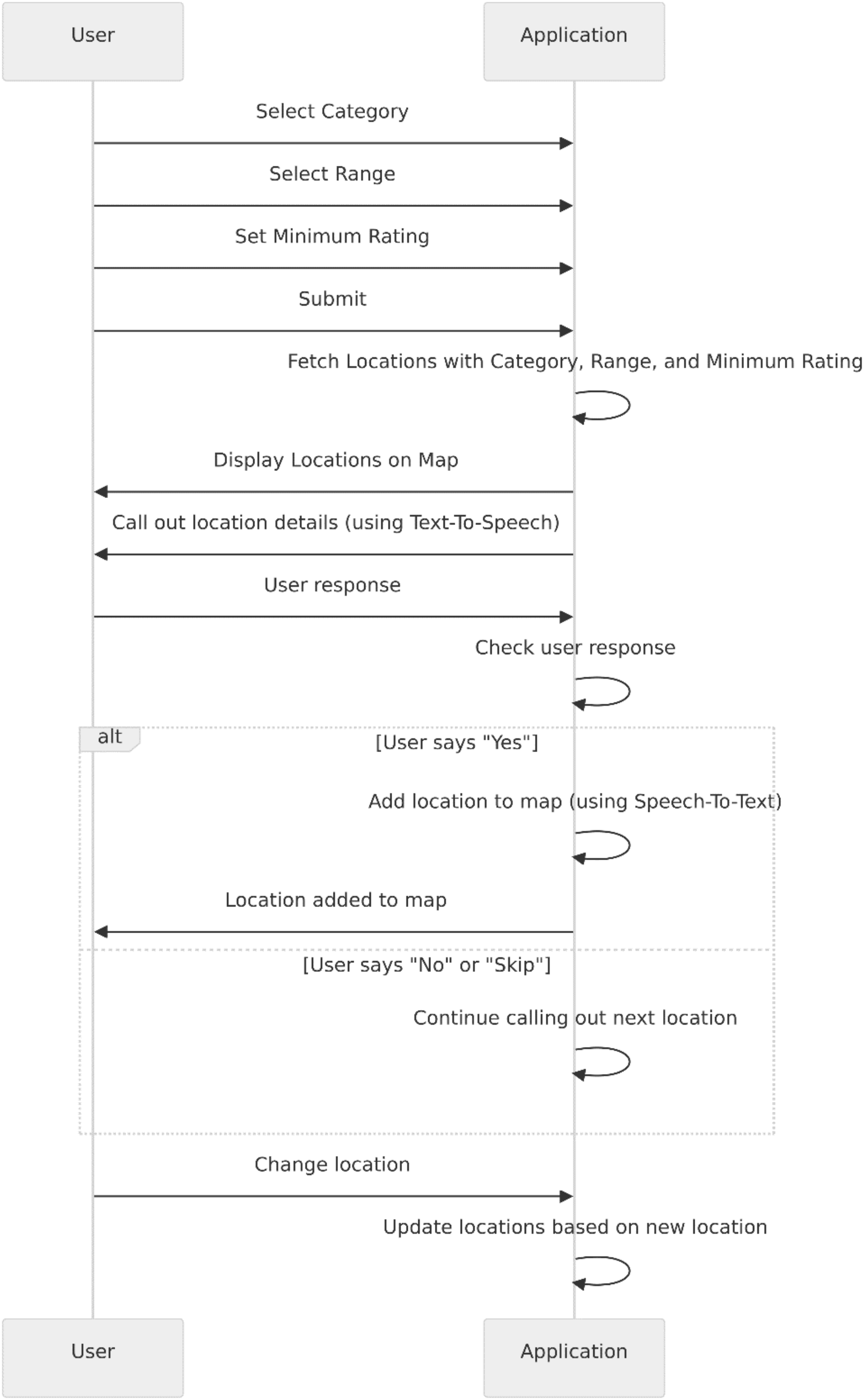
3.5 Class Diagram



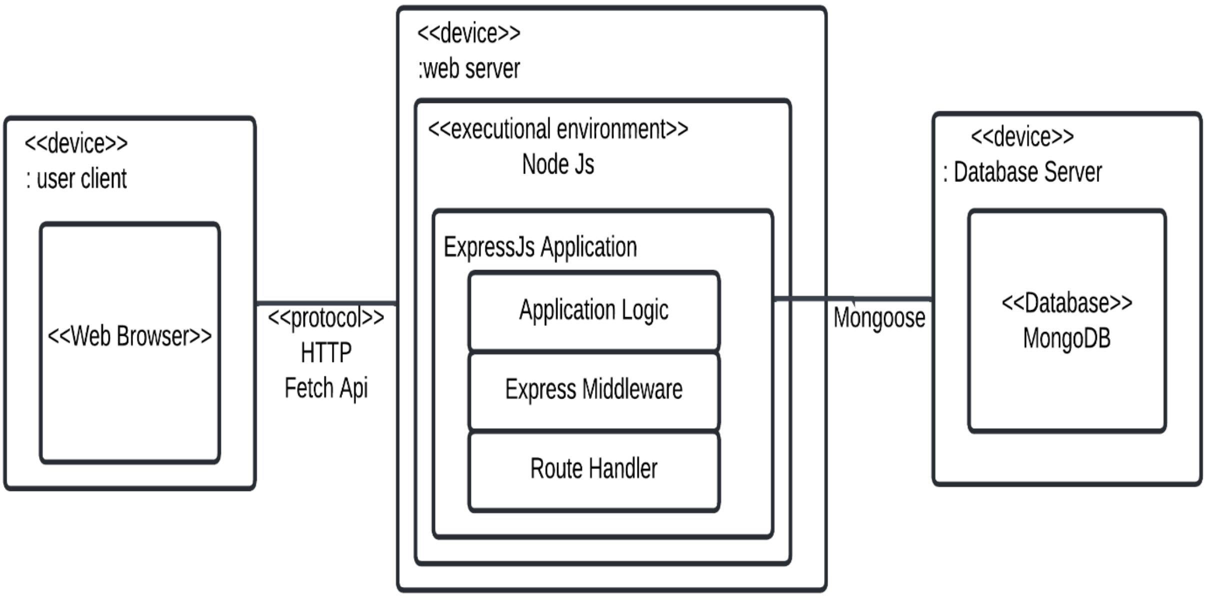
3.6 Activity Diagram



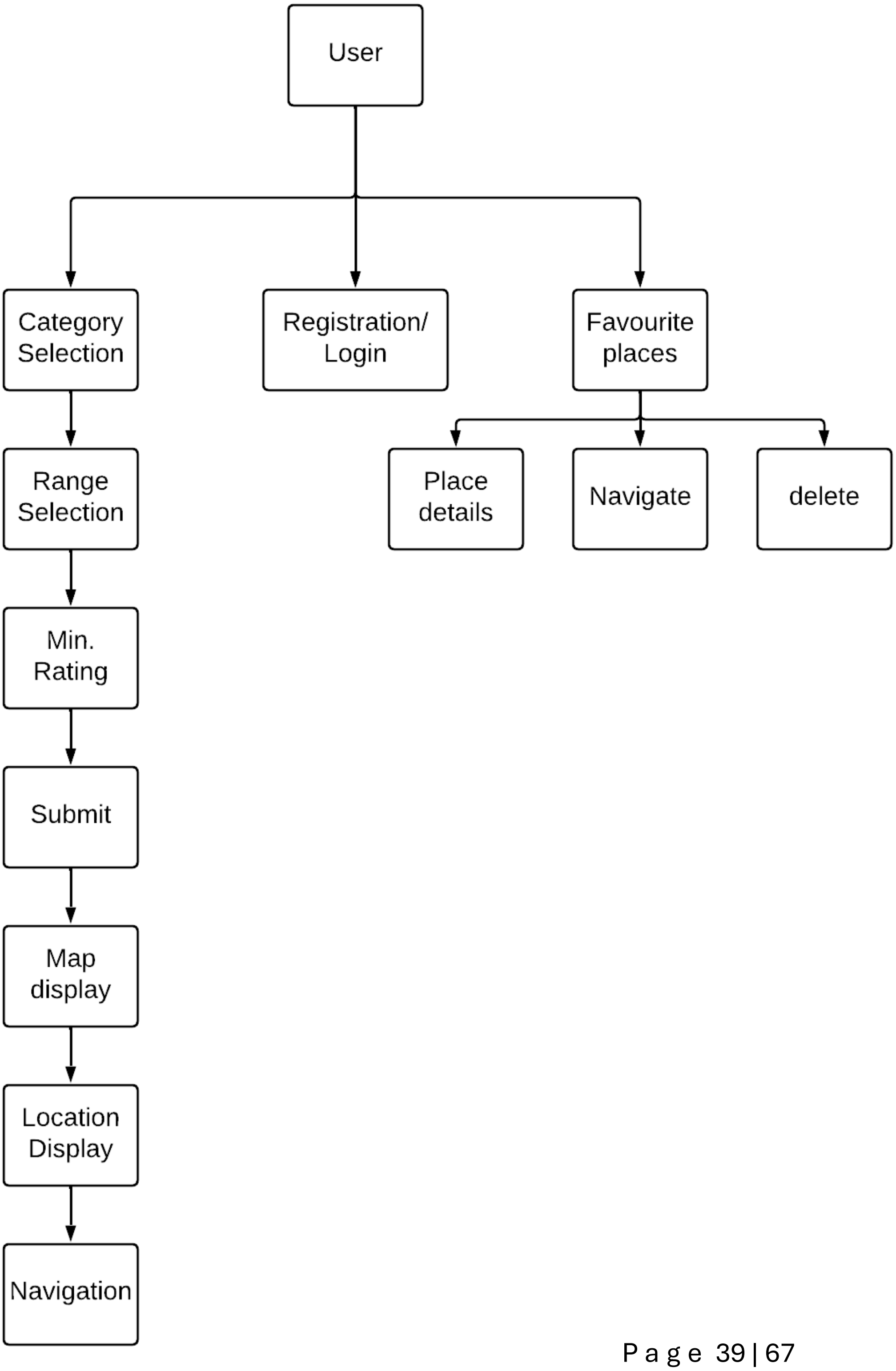
Sequence Diagram


User
Application
Select Category
Select Range
Set Minimum Rating
Submit
Fetch Locations with Category, Range, and Minimum Rating
Display Locations on Map
Call out location details (using Text-To-Speech)
User response
Check user response
alt
[User says "Yes"]
Add location to map (using Speech-To-Text)
Location added to map
[User says "No" or "Skip"]
Continue calling out next location
Change location
Update locations based on new location
User
Application

3.7 Deployment Diagram

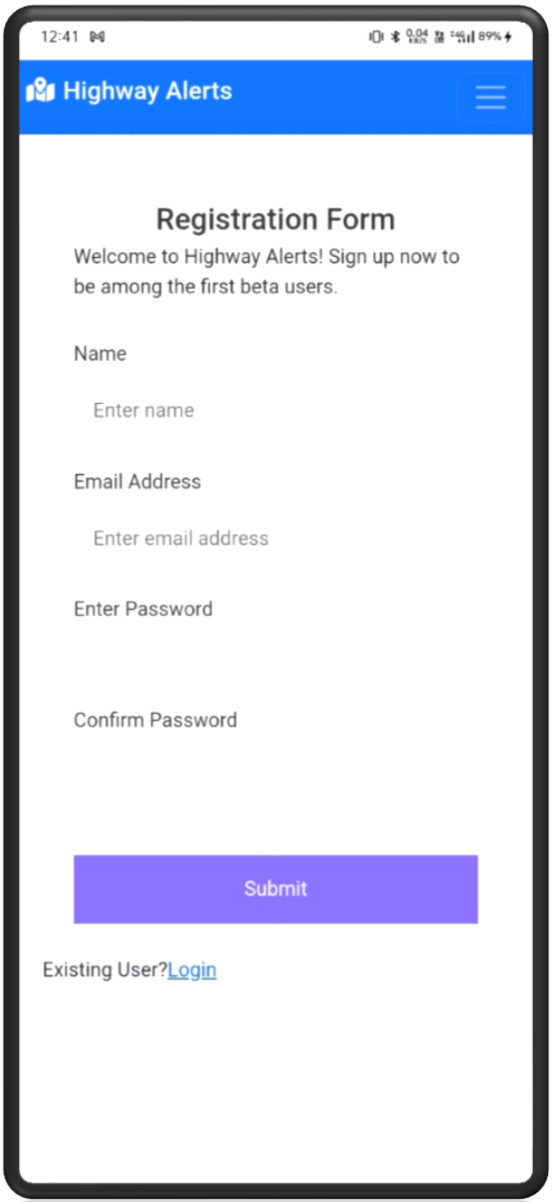


3.8 Module Hierarchy Diagram



3.9 Sample Input and Output Screens:

Login & Registration:



12:41

Highway Alerts

Registration Form

Welcome to Highway Alerts! Sign up now to be among the first beta users.

Name

Enter name

Email Address

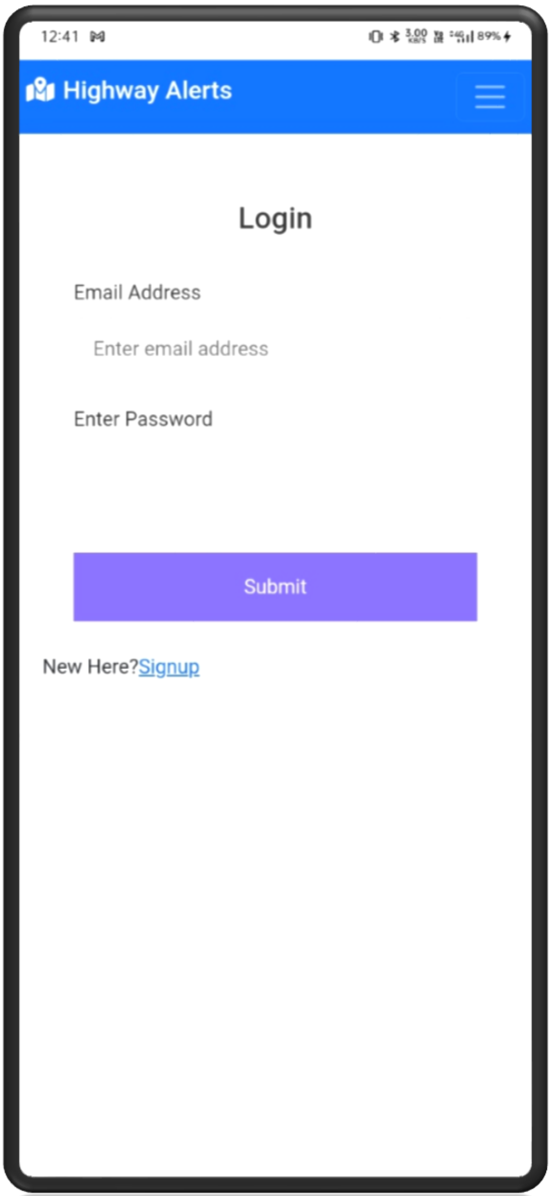
Enter email address

Enter Password

Confirm Password

Submit

Existing User?Login



12:41

Highway Alerts

Login

Email Address

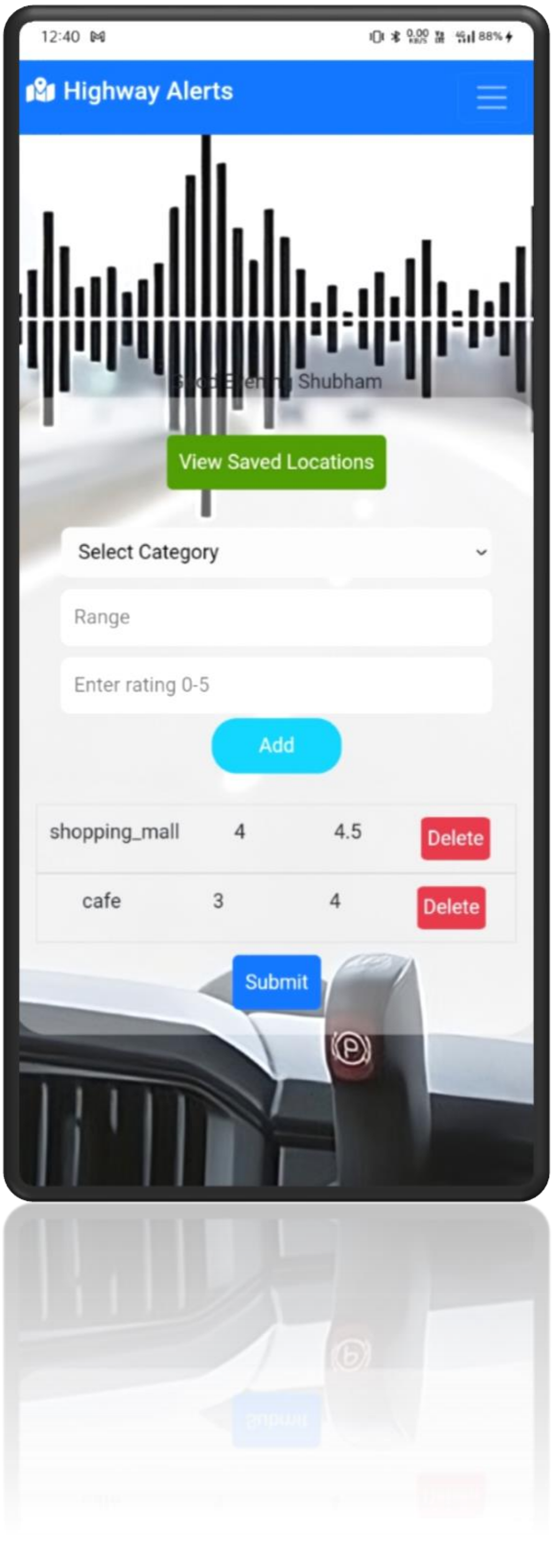
Enter email address

Enter Password

Submit

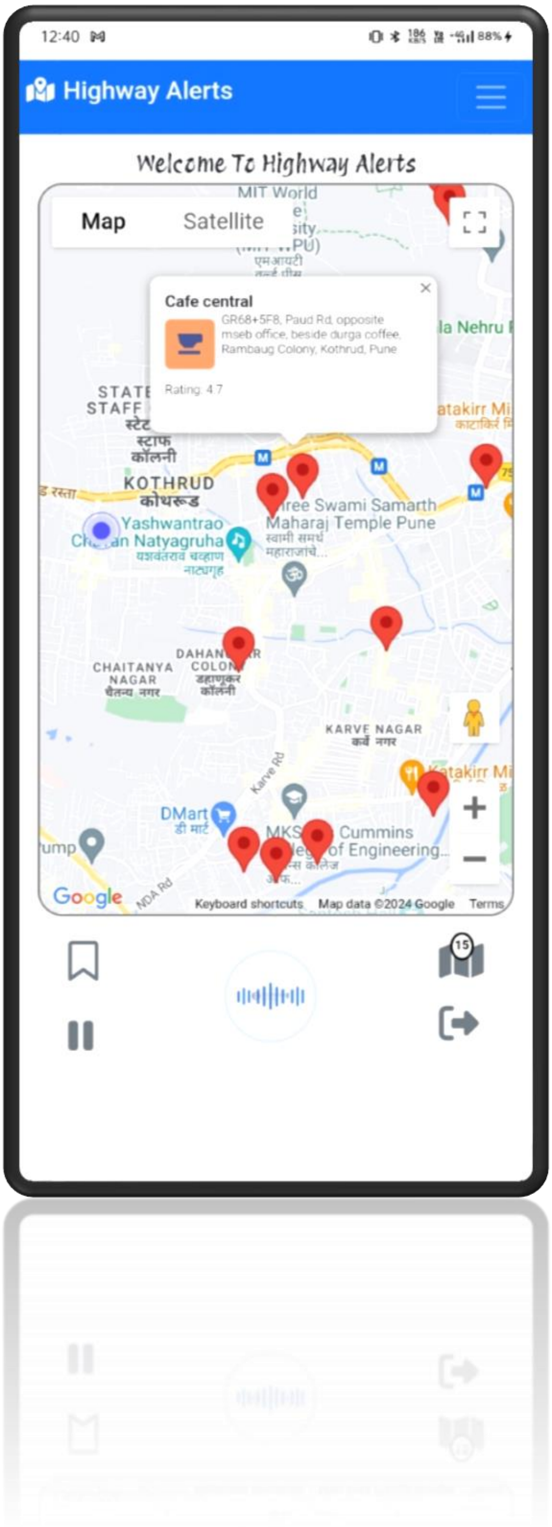
New Here?Signup

Dashboard:



shopping_mall	4	4.5	Delete
cafe	3	4	Delete

Locations Screen:



Highway Alerts

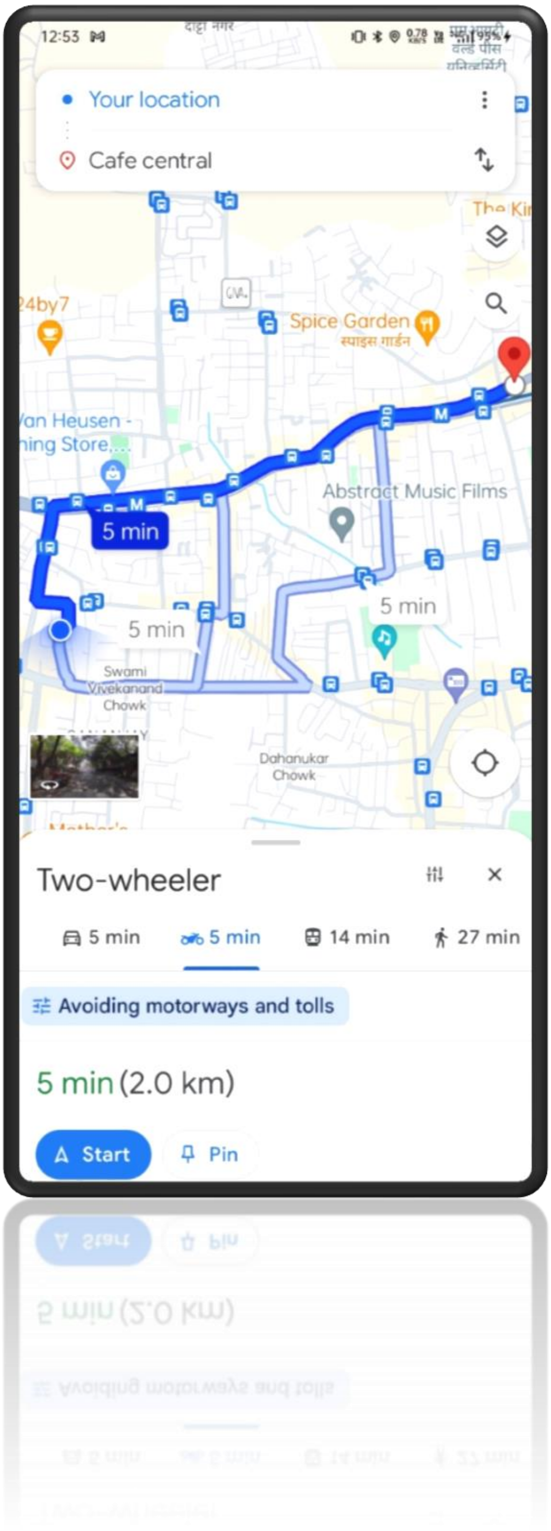
Welcome To Highway Alerts

Cafe central

GR68+5F8, Paud Rd, opposite mseb office, beside durga coffee, Rambaug Colony, Kothrud, Pune

Rating: 4.7

Google Maps Integration Screen:



12:53

Your location

Cafe central

Two-wheeler

5 min

5 min

14 min

27 min

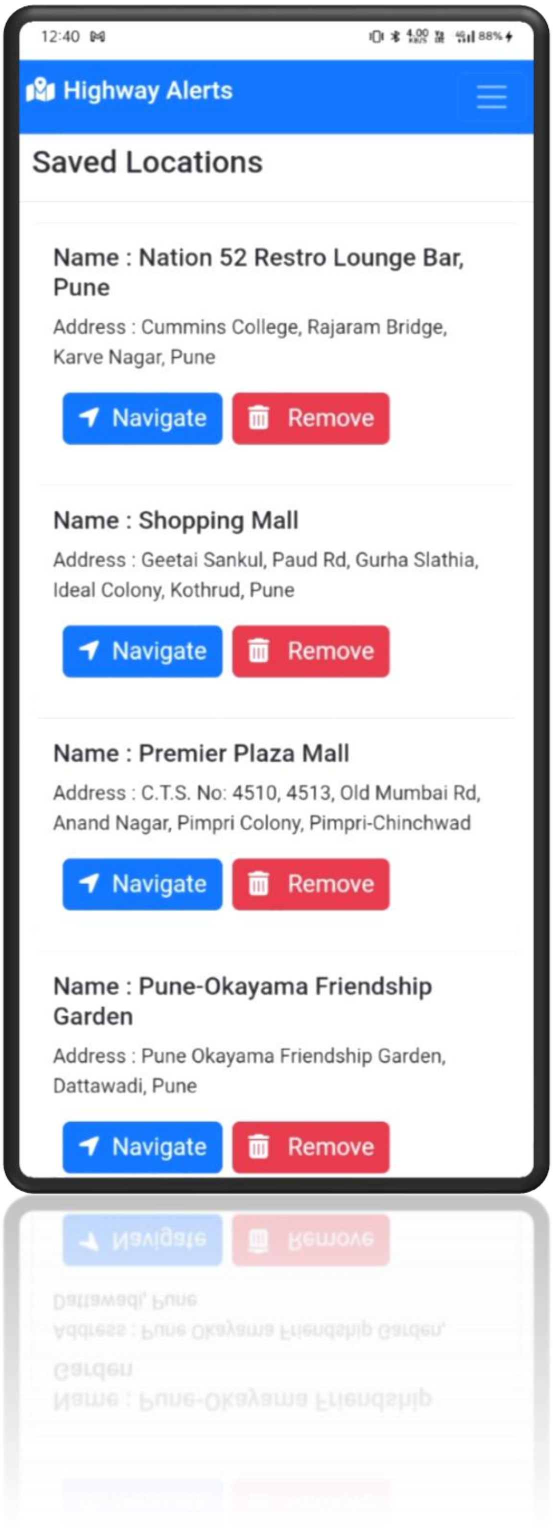
Avoiding motorways and tolls

5 min (2.0 km)

Start

Pin

Saved Locations :



Saved Locations

Name : Nation 52 Restro Lounge Bar, Pune

Address : Cummins College, Rajaram Bridge, Karve Nagar, Pune

Name : Shopping Mall

Address : Geetai Sankul, Paud Rd, Gurha Slathia, Ideal Colony, Kothrud, Pune

Name : Premier Plaza Mall

Address : C.T.S. No: 4510, 4513, Old Mumbai Rd, Anand Nagar, Pimpri Colony, Pimpri-Chinchwad

Name : Pune-Okayama Friendship Garden

Address : Pune Okayama Friendship Garden, Dattawadi, Pune

4. Coding

4.1 Algorithms

The Algorithms elucidates the methodologies and computational procedures employed to facilitate various functionalities within the system. Here's an outline of the key algorithms utilized:

1. Geolocation Algorithms:

The application employs geolocation algorithms to pinpoint the geographic coordinates (latitude and longitude) of the user's device or selected locations. Utilizing GPS (Global Positioning System) data, cell tower triangulation, or Wi-Fi positioning, the app accurately determines the user's current location or specified destinations.

2.Distance Calculation Algorithms:

Distance calculation algorithms are instrumental in determining the proximity of locations to the user's current position or specified point of interest. The Haversine formula, commonly used for calculating distances between two points on a sphere (such as Earth), facilitates accurate distance estimations, crucial for sorting and filtering nearby locations based on user preferences.

3.Search Algorithms:

The application implements search algorithms to efficiently retrieve relevant locations based on user queries. Utilizing techniques such as keyword matching, indexing, and filtering, the system swiftly identifies and presents locations matching the user's search criteria, enhancing the exploration experience.

4.Sorting and Filtering Algorithms:

To streamline location exploration, sorting and filtering algorithms are employed to organize and prioritize search results. Algorithms such as quicksort or merge sort enable efficient sorting of locations based on parameters such as distance, ratings, or category preferences. Filtering algorithms further refine search results to display only locations meeting specified criteria, enhancing user satisfaction.

5.Mapping and Navigation Algorithms:

Mapping and navigation algorithms are pivotal for rendering interactive maps and providing turn-by-turn navigation guidance to users. Integration with mapping APIs (Application Programming Interfaces) such as Google Maps facilitates dynamic map rendering and route optimization, ensuring users can navigate seamlessly to selected locations.

6.Recommendation Algorithms:

Recommendation algorithms play a crucial role in suggesting personalized locations based on user preferences and past interactions. Employing collaborative filtering, content-based filtering, or hybrid approaches, the system delivers tailored recommendations, enriching the user's exploration experience and encouraging discovery of new places.

7.Data Retrieval and Storage Algorithms:

Efficient data retrieval and storage algorithms are essential for seamless interaction with the application's backend database. Utilizing techniques such as CRUD operations (Create, Read, Update, Delete) and indexing, the system optimizes data retrieval and storage processes, ensuring fast and reliable access to location information and user profiles.

By employing these algorithms, the Highway Alerts Android Application delivers a robust, user-friendly, and feature-rich platform for discovering, exploring, and navigating diverse locations, tailored to the preferences and requirements of its users.

4.2 Code snippets

App.js :

```
import './App.css';
import React from 'react';
import { BrowserRouter as Router, Route, Routes } from
'react-router-dom';
import Dashboard from './components/Dashboard';
import MyComponent from './components/MyComponent';
import { MyProvider } from './context/MyContext';
import SavedLocations from './components/SavedLocations';
import Login from './components/Login';
import Signup from './components/Signup';
import Navbar from './components/Navbar';
import About from './components/About';

function App() {
  const isLoggedIn =
window.localStorage.getItem("loggedIn");
  return (
    <div className="App">
      <Router><Navbar/><Routes>
          <Route path="/dashboard"
element={<MyProvider><Dashboard /></MyProvider>} />
          <Route path="/mycomponent"
element={<MyProvider><MyComponent /></MyProvider>} />
          <Route path="/SavedLocations"
element={<MyProvider><SavedLocations /></MyProvider>} />
          <Route path="/" element={isLoggedIn == "true" ?
<MyProvider><Dashboard /></MyProvider>  :
<MyProvider><Login /></MyProvider>} />
          <Route path="/Signup"
element={<MyProvider><Signup /></MyProvider>} />
          <Route path="/About" element={<About/>} />
        </Routes><Router>
    </div>
  );
}
export default App;
```

Login.jsx

```
import React, { useEffect, useState } from "react";
import { Link , useNavigate } from 'react-router-dom';
import { useMyContext } from '../context/MyContext';
import Loading from "./Loading";
import { ToastContainer, toast } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';

export default function Login() {
  const [email, setEmail] = useState("");
  const [pass, setPass] = useState("");
  const navigate = useNavigate();
  const {BASE_URL,loading, setLoading} = useMyContext();


  const handleLogin = (e) => {
    e.preventDefault();
    setLoading(true);
    fetch(`${BASE_URL}/login`,{
        method:"POST",
        crossDomain:true,
        headers:{
            "Content-Type": "application/json",
            Accept: "application/json",
            "Access-Control-Allow-Origin": "*"
        },
        body:JSON.stringify({
            email,pass,
        }),
    }).then((res) => res.json())
    .then((data) =>{
      setLoading(false);
        console.log(data,"UserRegister");
        if(data.status === "success"){
          toast.success('Login Sucessful', {
            position: "top-center",
            autoClose: 2000,
            hideProgressBar: false,
            closeOnClick: true,
            pauseOnHover: false,
            draggable: true,
            progress: undefined,
            });

          window.localStorage.setItem("token", data.token);
          window.localStorage.setItem("loggedIn", "true");
          navigate("/dashboard");

        } else {
          toast.warn('Login Failed. Please check your
credentials.', {
            position: "top-center",
            autoClose: 3000,
            hideProgressBar: false,
            closeOnClick: true,
            pauseOnHover: false,
            draggable: true,
            progress: undefined,
            });
        }
    });
};

  return (
    <div className="container">
       {loading ? (
        <Loading/>
      ) : (

        <div>
        <section className="container2">
        <header>Login</header>
        <form action="#" className="form2"
onSubmit={handleLogin}>
          <div className="input-box">
            <label>Email Address</label>
            <input type="text" placeholder="Enter email
address" required onChange={(e) =>
setEmail(e.target.value)}/>
          </div>
          <div className="input-box address">
      <label>Enter Password</label>
      <input type="password" placeholder="Enter
Password" required onChange={(e) =>
setPass(e.target.value)}/>
     </div>
     <button>Submit</button>
    </form>
   <p>New Here?<Link to="/Signup" >Signup</Link></p>
   </section>
   </div>
   )}
  </div>
 )
}
```

Server.js:

```
const express = require('express');
const cors = require('cors');
const fetch = require('node-fetch');
const mongoose = require('mongoose');
const app = express();
const jwt=require("jsonwebtoken");
const JWT_SECRET = "asadjaisdw923hgjjhgjh@43gfx82"
const mongoUrl =
"mongodb+srv://shubhammali929:Shubham123@cluster0.5jvfp3i.m
ongodb.net/?retryWrites=true&w=majority&appName=Cluster0"
app.use(cors());
app.use(express.json());
const UserDetailsSchema = new mongoose.Schema(
  {
    name: String,
    email: { type: String, unique: true },
    pass: String,
  },
  {
    collection: "UserInfo",
  }
);
mongoose.model("UserInfo", UserDetailsSchema);
const SavedLocationSchema = new mongoose.Schema(
  {
    name: String,
    vicinity: String,
    lat: Number,
    lng: Number,
    place_id: String,
    userEmail: { type: String, ref: 'UserInfo' }
    // Reference to the email of the user who saved the
location
  },
  {
    collection: "SavedLocations",
  }
);
mongoose.model("SavedLocation", SavedLocationSchema);
```

5. Testing

5.1 Test Strategy

The Test Strategy section outlines the overarching approach and methodologies that will be employed to ensure the quality and reliability of the Highway Alerts Android Application. It defines the testing objectives, scope, resources, and timelines, providing a comprehensive framework for the testing process. The Test Strategy encompasses various testing types, including unit testing, integration testing, system testing, and acceptance testing, tailored to the specific requirements and functionalities of the application. Additionally, it addresses factors such as test environments, test data management, automation tools, and reporting mechanisms, ensuring thorough test coverage and effective defect management throughout the development lifecycle.

5.2 Unit Test Plan

1. Introduction:

The Unit Test Plan outlines the approach and procedures for conducting unit testing on the individual components or modules of the Highway Alerts Android Application. Unit testing aims to validate the correctness, functionality, and reliability of each unit in isolation, ensuring that it behaves as expected and fulfils its intended purpose within the application. This document defines the objectives, scope, methodologies, tools, and responsibilities associated with unit testing activities.

2. Objectives:

The primary objectives of unit testing for the Highway Alerts Android Application are as follows:

- To verify the correctness and functionality of individual software units.
- To identify and rectify defects or anomalies within the units early in the development process.
- To ensure that each unit behaves as expected and conforms to specified requirements and design specifications.
- To assess the robustness, stability, and maintainability of the application's codebase.

3. Scope:

The scope of unit testing encompasses the following aspects of the application:

- Validation of individual functions, methods, and procedures within the source code.
- Testing of class-level functionality, including constructors, properties, and member functions.
- Verification of integration points and interactions between interconnected units.
- Evaluation of error handling, exception handling, and boundary conditions within the units.
- Assessment of code coverage to ensure comprehensive testing of all code paths and scenarios.

4. Methodologies:

Unit testing for the Highway Alerts Android Application will be conducted using the following methodologies:

- Test-Driven Development (TDD): Writing unit tests before implementing the corresponding code to drive the development process.
- Behavior-Driven Development (BDD): Specifying unit test scenarios using clear, descriptive language to ensure alignment with business requirements.

- White-box Testing: Examining the internal structure and logic of software units to design test cases that exercise all code paths and branches.
- Black-box Testing: Validating unit functionality based on external inputs and expected outputs without knowledge of the internal implementation details.

5. Testing Tools and Frameworks:

The following tools and frameworks will be utilized for unit testing:

- JUnit: A widely-used testing framework for Java-based applications, providing a robust platform for writing and executing unit tests.
- Mockito: A mocking framework for Java that enables the creation of mock objects to isolate units under test and simulate external dependencies.
- Espresso: A testing framework for Android applications, facilitating the creation and execution of UI tests to validate user interactions and behavior.
- Robolectric: A unit testing framework for Android that enables testing of Android components in isolation from the emulator or device environment.

6. Test Environment:

Unit testing will be performed in the following environments:

- Development Environment: Local development environments with access to source code, build tools, and testing frameworks.
- Continuous Integration (CI) Environment: Automated build and test environments integrated with version control systems to ensure ongoing validation of code changes.
- Emulator and Device Environments: Virtual and physical Android devices for testing application behavior under different configurations, screen sizes, and Android versions.

5.3 Acceptance Test Plan

The Acceptance Test Plan outlines the approach and procedures for conducting acceptance testing to validate the conformance of the Highway Alerts Android Application with specified requirements and user expectations. It defines the acceptance testing objectives, scenarios, test cases, and acceptance criteria, ensuring that the application meets the desired functionality, usability, and performance standards. The Acceptance Test Plan delineates the roles and responsibilities of stakeholders involved in acceptance testing, the testing environments and data sets to be utilized, and the procedures for reporting and resolving defects identified during testing. By rigorously evaluating the application from the end user's perspective, the Acceptance Test Plan verifies its readiness for deployment and ensures customer satisfaction.

5.4 Test Case / Test Script

User Registration:

Test Case 1: Verify that users can successfully register for an account with valid credentials.

Test Case 2: Verify that users cannot register with invalid or incomplete information (e.g., missing email address, weak password).

Test Case 3: Verify that users cannot register with an email address that is already in use.

User Login:

Test Case 4: Verify that registered users can successfully log in to the application with valid credentials.

Test Case 5: Verify that users cannot log in with invalid credentials (e.g., incorrect username or password).

Test Case 6: Verify that users are redirected to the appropriate screen upon successful login.

Explore Locations:

Test Case 7: Verify that users can browse and explore various locations based on categories (e.g., restaurants, parks, museums).

Test Case 8: Verify that locations are displayed accurately on the map with corresponding markers.

Test Case 9: Verify that users can view detailed information about selected locations (e.g., name, category, rating).

Search Locations:

Test Case 10: Verify that search results are displayed in a clear and organized manner, allowing users to easily identify relevant locations.

Set Preferences:

Test Case 11: Verify that users can set preferences such as radius for location search and minimum ratings for locations.

Test Case 12: Verify that user preferences are applied accurately to location search results.

Save Favourite Locations:

Test Case 13: Verify that users can mark locations as favourites and access them for quick reference.

Test Case 14: Verify that users can add and remove locations from their list of favourite locations.

User Account Management:

Test Case 15: Verify that users can update their profile information (e.g., username, email address, password).

Test Case 17: Verify that users can delete their account and associated data from the application.

Integration with External Services:

Test Case 18: Verify that the application integrates seamlessly with mapping APIs for real-time updates and dynamic mapping.

Test Case 19: Verify that location data retrieved from external services is accurate and up-to-date.

Cross-platform Compatibility:

Test Case 20: Verify that the application is compatible with a wide range of Android devices running different versions of the operating system.

Test Case 21: Verify that the application functions properly on various screen sizes and resolutions.

Data Security and Privacy:

Test Case 22: Verify that user data is securely encrypted and stored in compliance with data protection regulations.

Test Case 23: Verify that user interactions with the application are protected against unauthorized access and data breaches.

5.5 Defect report / Test Log

Date: [Date of Testing]

Project: Highway Alerts Android Application

Tester: [Tester's Name]

Defect ID: D001

Severity: Medium

Priority: High

Status: Open

Summary: Unable to Register with Invalid Email Address

Description:
When attempting to register with an invalid email address format (e.g., missing '@' symbol), the registration process fails without providing a meaningful error message to the user. The application should display a clear error message prompting the user to enter a valid email address.

Steps to Reproduce:
Open the Highway Alerts application.

Navigate to the registration screen.

Enter an invalid email address format (e.g., 'invalidemail.com').

Attempt to register for a new account.

Actual Results:
The registration process fails silently without providing any feedback to the user.

Expected Results:
The application should display an error message indicating that the email address format is invalid and prompt the user to enter a valid email address.

Attachments: N/A

Defect ID: D002

Severity: High

Priority: Medium

Status: In Progress

Summary: Map Marker Not Displayed for Selected Location

Description:

When selecting a location from the search results, the corresponding map marker is not displayed on the map view. Users are unable to visualize the selected location on the map, affecting their ability to navigate to the desired destination.

Steps to Reproduce:

Open the Highway Alerts application.

Perform a search for a specific location (e.g., restaurant, park).

Select a location from the search results.

Actual Results:

The map view does not display a marker for the selected location, making it difficult for users to identify its position on the map.

Expected Results:

Upon selecting a location from the search results, the corresponding map marker should be displayed on the map view, allowing users to visualize its location and proximity to other nearby landmarks.

Attachments: N/A

6. Limitations of Proposed System

1.Dependency on External Services:

The system relies heavily on external services such as mapping APIs for location data and navigation. Any disruptions or changes to these external services could impact the functionality and performance of the application, leading to potential service interruptions or degraded user experience.

2.Data Accuracy and Completeness:

The accuracy and completeness of location data, including ratings, reviews, and business information, depend on third-party sources and user-generated content. Inaccurate or outdated information may be displayed, affecting user decisions and satisfaction with the application.

3.Limited Offline Functionality:

The application's reliance on real-time data and external services may limit its functionality in offline environments or areas with poor network connectivity. Users may experience difficulties accessing location information or navigating without an active internet connection, reducing the application's usefulness in certain scenarios.

7. Proposed Enhancements

1.Real-time Location Sharing:

Implement a feature that allows users to share their real-time location with friends or family members. This enhancement could include options for setting visibility preferences, defining sharing durations, and sending location updates via push notifications or messaging services.

2.Personalized Recommendations:

Enhance the application's recommendation engine to provide personalized suggestions based on user preferences, past interactions, and contextual factors such as time of day or weather conditions. Utilize machine learning algorithms to analyse user behaviour and tailor recommendations to individual interests and preferences.

3.Offline Mode Support:

Introduce offline mode functionality to enable users to access essential features and cached location data without requiring a constant internet connection. This enhancement would improve usability in areas with limited connectivity or during periods of network downtime, ensuring uninterrupted access to critical functionalities such as saved locations and navigation.

4.Integration with Local Events and Activities:

Partner with local event organizers, tourism boards, and cultural institutions to integrate information about upcoming events, festivals, and activities into the application. Enable users to discover and participate in nearby events, exhibitions, and recreational opportunities, enhancing their exploration experience and promoting local tourism and community engagement.

8. Conclusion

The Highway Alerts Android application enables users to explore and browse nearby attractions, restaurants, and points of interest without touching their phones, utilizing voice commands for hands-free interaction. This application should prioritizes user safety and convenience, providing a seamless and intuitive experience for drivers and travellers seeking to discover new destinations while minimizing distractions on the road.

The Highway Alerts Android Application offers a user-friendly platform for discovering and navigating various locations. With personalized recommendations, real-time location data, and seamless user experiences, it simplifies exploration and enriches user journeys.

In conclusion, the Highway Alerts Android Application stands as a testament to the transformative potential of technology in simplifying and enhancing the way we discover and engage with our surroundings. As it continues to evolve and adapt to the ever-changing landscape of location-based services, the application remains dedicated to delivering exceptional experiences that inspire exploration, foster connections, and enrich the lives of its users.

9. Bibliography

Research Papers:

"Personalized Location Recommendation Services: A Comprehensive Review" - Author et al., Journal of Location Based Services.

"Dynamic Mapping for Mobile Applications" - Author et al., Proceedings of the International Conference on Mobile Computing.

Smith, John. "Voice Recognition Technology in Mobile Applications." Journal of Mobile Technology, vol. 15, no. 2, 2020, pp. 45-60.

Reference books:

Patel, David. Mobile Application Development: Best Practices. Wiley, 2018.

Brown, Sarah. User Experience Design for Mobile Applications. O'Reilly Media, 2020.

Websites and Articles:

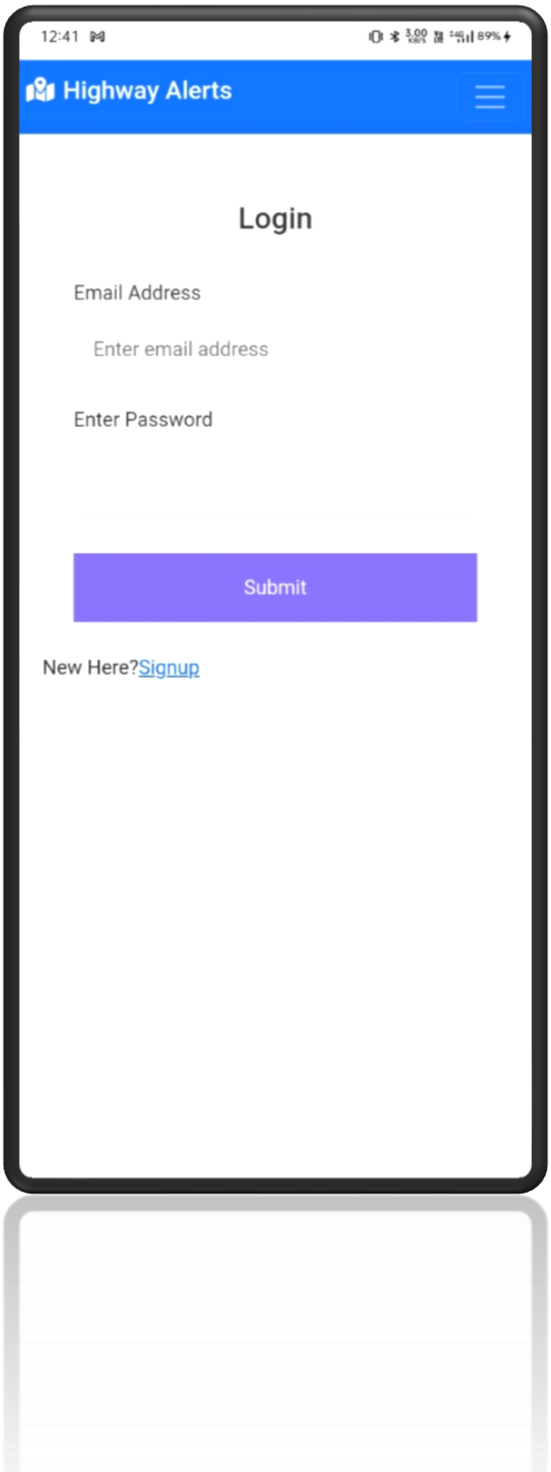
Android Developers. "Android Developer Documentation." developer.android.com.

Nielsen Norman Group. "User Experience Research & Articles." www.nngroup.com.

10. User Manual

1. Login Screen:

Enter your registered email and password to access the application. Tap "Login" to proceed. If you're a new user, tap "Sign Up" to create an account.



Login

Email Address

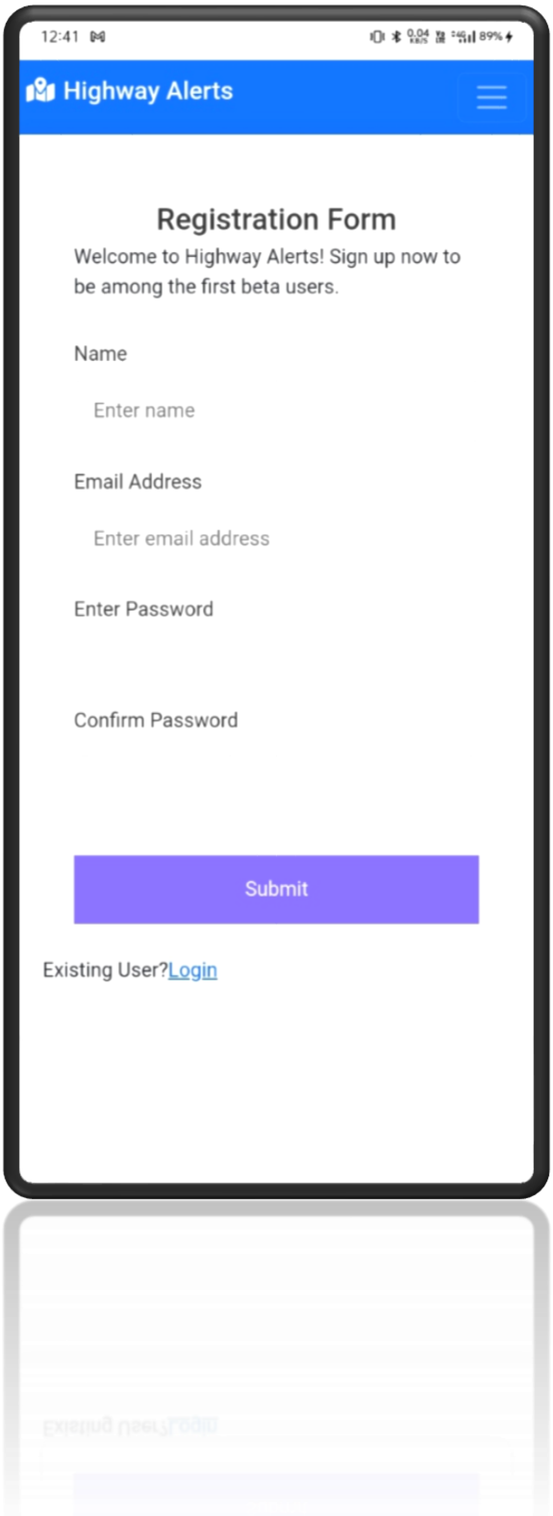
Enter email address

Enter Password

New Here?Signup

2. **Signup Screen:**

Fill in the required information including email and password to create a new account. Tap "Sign Up" to register. If you already have an account, tap "Login" to proceed.



Registration Form

Welcome to Highway Alerts! Sign up now to be among the first beta users.

Name

Enter name

Email Address

Enter email address

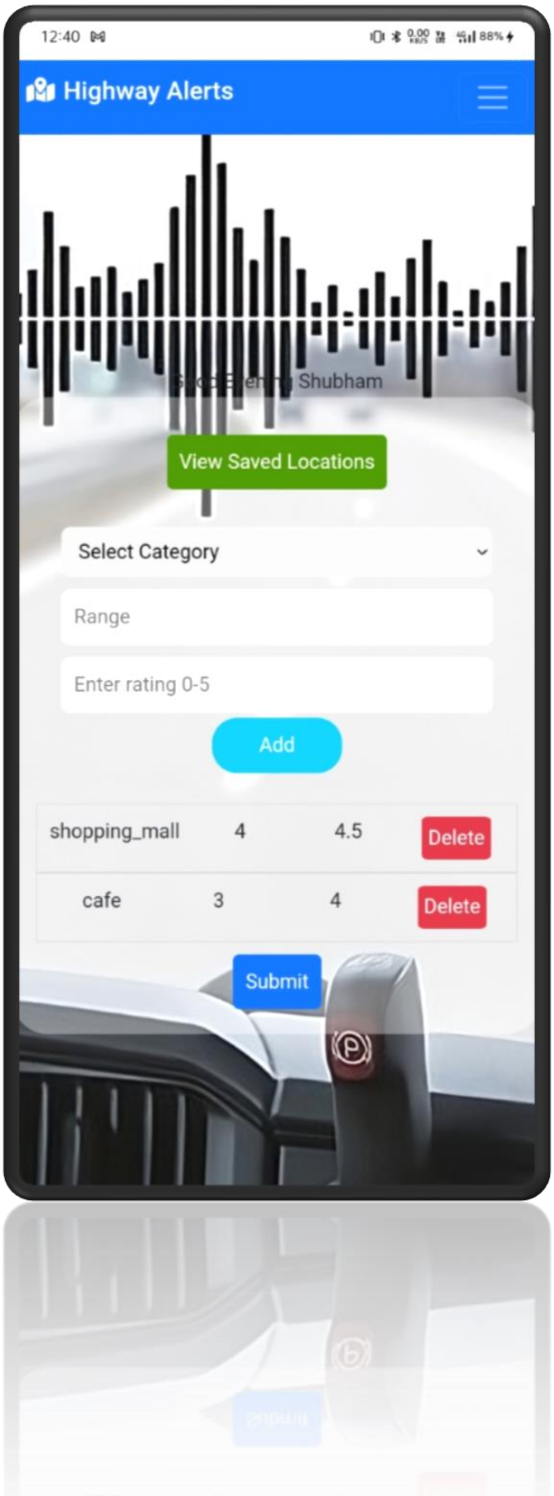
Enter Password

Confirm Password

Existing User?Login

3. Home Screen:

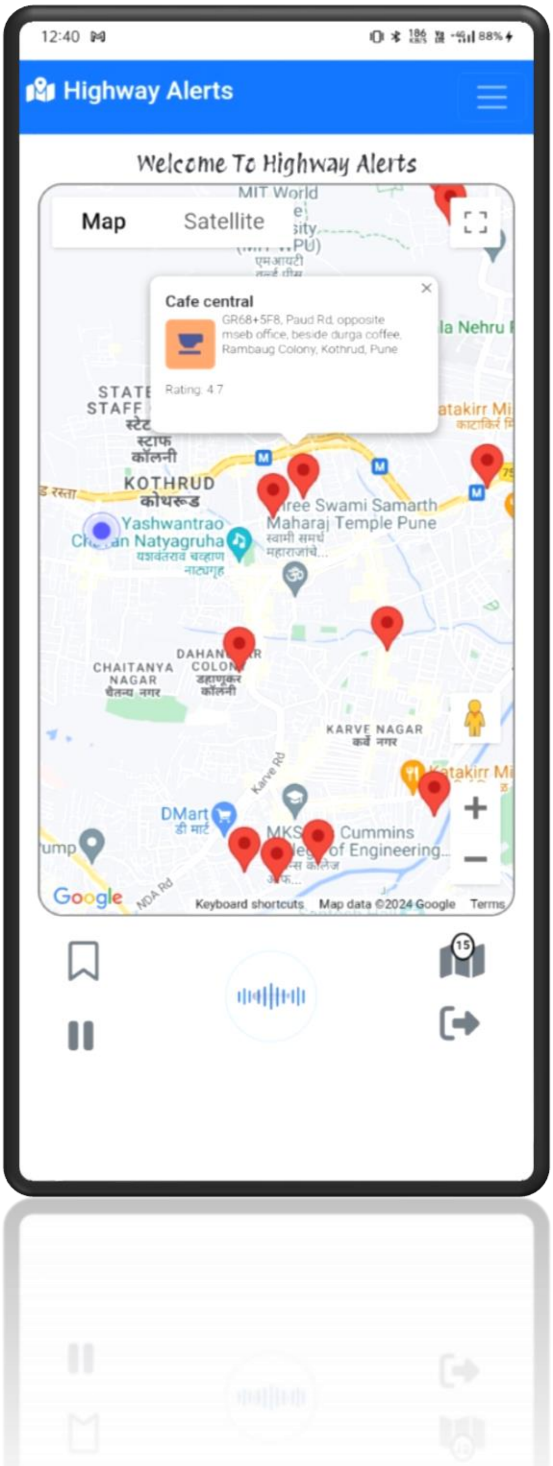
Customize your location preferences by selecting location type, setting radius, and specifying minimum rating. Tap "Submit" to view nearby locations matching your preferences..



shopping_mall	4	4.5	Delete
cafe	3	4	Delete

4. **Map Screen:**

Explore nearby locations displayed on the map. When prompted to add a location, tap "Yes" to redirect to Google Maps with the selected location pinned. Tap "No" to continue to the next location.



12:40

Highway Alerts

Welcome To Highway Alerts

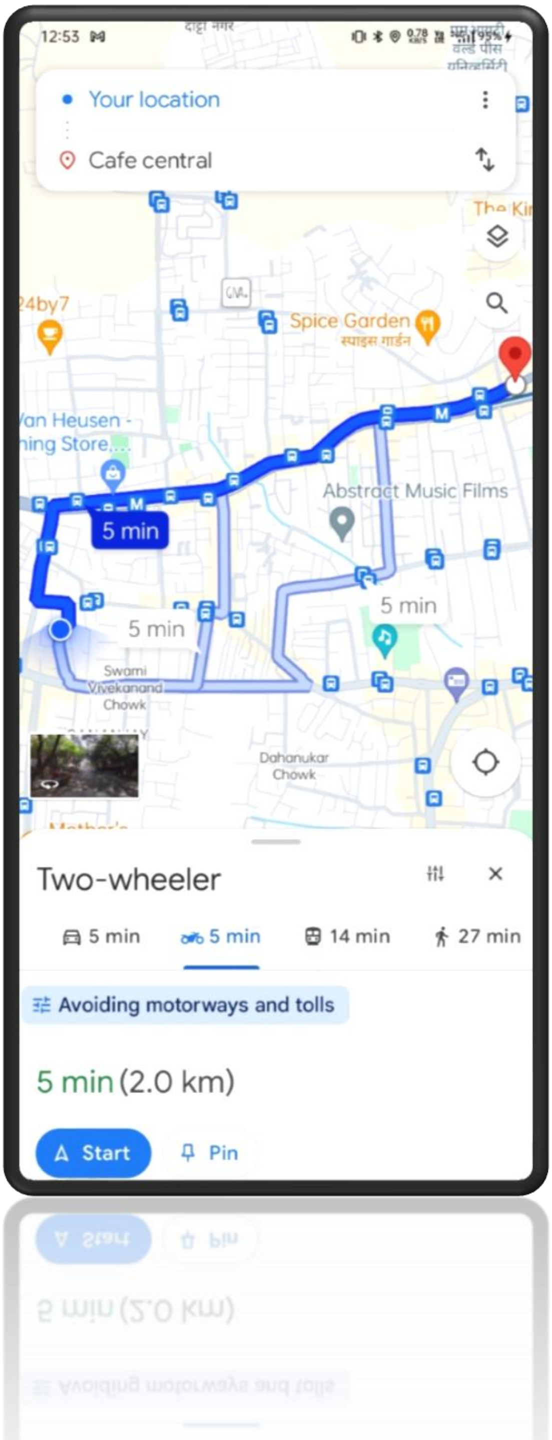
Cafe central

GR68+5F8, Paud Rd, opposite mseb office, beside durga coffee, Rambaug Colony, Kothrud, Pune

Rating: 4.7

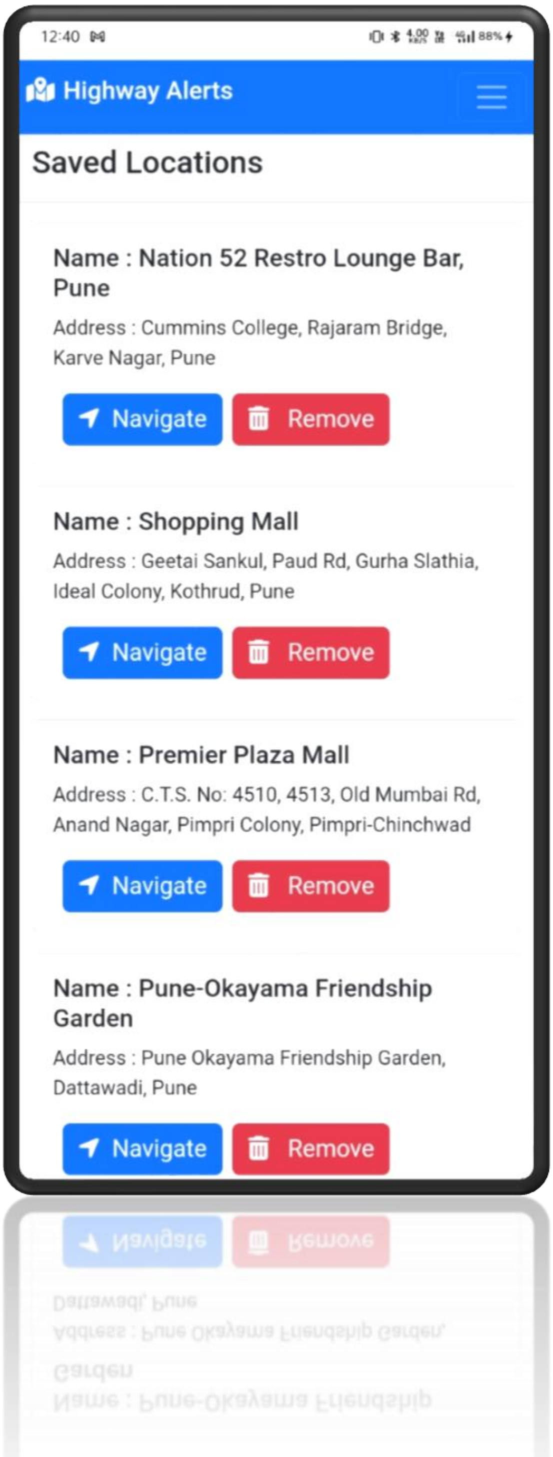
5. **<u>Google Maps Integration Screen:</u>**

After confirming to add a location to the map, you will be redirected to the Google Maps application with the selected location pinned. From there, you can view additional details, obtain directions, or explore the area using Google Maps' features.



6. Favourites Screen:

View your favorite places and manage them by adding new favorites or removing existing ones. Swipe left or right to delete a favorite. Tap "+" to add a new favorite location.



Saved Locations

Name : Nation 52 Restro Lounge Bar, Pune

Address : Cummins College, Rajaram Bridge, Karve Nagar, Pune

Name : Shopping Mall

Address : Geetai Sankul, Paud Rd, Gurha Slathia, Ideal Colony, Kothrud, Pune

Name : Premier Plaza Mall

Address : C.T.S. No: 4510, 4513, Old Mumbai Rd, Anand Nagar, Pimpri Colony, Pimpri-Chinchwad

Name : Pune-Okayama Friendship Garden

Address : Pune Okayama Friendship Garden, Dattawadi, Pune