School of Computer Science and Engineering

Software Engineering Methodologies

Course Code: CBS1005

Class Number: **VL2023240103391**

PROJECT ON

Optimising Transportation

(Ezz Route)

ACKNOWLEDGEMENT

I would like to express my deep appreciation and heartfelt gratitude to my supervisor, Prof.Yoganand S., from VIT University. Without his unwavering motivation and continuous encouragement, this research would not have been completed successfully.

I am sincerely grateful to the Chancellor of VIT University, Dr. G Viswanathan, the Vice Presidents, and the Vice Chancellor for their inspiring guidance and support. Their encouragement and provision of essential infrastructure and resources were instrumental in the progress of my research.

I would also like to extend my sincere thanks to Dr. Ramesh Babu K, the Dean of the School of Computer Science and Engineering (SCOPE) at VIT University, for his kind words of support and encouragement throughout my research journey. Additionally, I am grateful to my classmates for their support in various ways throughout my research work.

Furthermore, I wish to express my profound gratitude to my parents for their enduring sacrifices during my research. Their unwavering support and encouragement have been invaluable to me.

I am truly grateful to all those mentioned above for their contributions to my research, and I feel incredibly fortunate to have had their guidance and support throughout this endeavor.Place – Vellore

Team Members

1. **Divyam vijayvargia 22BBS0005**
2. **Ayush Gupta 22BBS0044**
3. **Sanskar Agrawal 22BBS0052**
4. **Khushi Seth 22BBS0228**

Signature of Supervisor

Table of Contents:

INTRODUCTION:

Ezz route aims to ease your travel by taking the input of start and end location from the user and then providing with a best route for it with directions as well.. we used here mapbox for taking the live location.
Also, we tried providing the distance and toll calculator.

Objective: The website's primary goal is to provide users with easy access to geographic information, such as maps, directions, and points of interest.

Interactive Geographic Information: A map website is an online space that provides users with access to a wealth of interactive geographic information. It allows users to explore the world, from local neighborhoods to global landscapes, through digital maps.

Real-Time Data: Many map websites incorporate real-time data, enabling users to stay updated on traffic conditions, weather, and more. This real-time information enhances the user's ability to make informed decisions.

Storytelling: Map websites are not just about navigation but also storytelling. They allow users to convey information and stories through maps, making them valuable for educational, journalistic, and business purposes.

User-Friendly Interface: These websites prioritize a user-friendly interface, ensuring that individuals of all technical levels can effectively navigate and utilize the features.

Data Accuracy: Maps on these websites are drawn accurately on a flat surface, ensuring that users can rely on the data for navigation and decision-making.

ABSTRACT:

The concrete details involves the different modes of travel included in our project along with the live coordinates provided by the mapbox and then the starting and final destination names provided by the user.
Coming further to our next part of it we included as told to us the toll calculator and distance calculator. It takes the live coordinates provided by the map box and then converts it to the different names of the cities or states respective to that of the coordinates. Based on that it furthers calculates the set distance and toll for it.

Our main aim is to provide the best route but we also took a call on the toll calculation for further optimizing it.

- It helps the user to discover and navigate our land with ease and precision.
- Our users can seamlessly plan trips, explore new destination within a user friendly interface. It helps to simplify our navigation.

LOGO :

Software Requirements Specification (SRS)

l'kis doc"mc→ıt is a moíc dctailcd s"mmaíQ or tkc sortwaíc ícq"iícmc→ıts spcciricatio→ı (SRS) roí a tíarric optimizatio→ı sortwaíc callcd EZZROUl'E. l'kc doc"mc→ıt co:cís tkc rollowi→ıg aspccts or tkc sortwaíc:

- **I→ıtíoductio→ı**: l'kc doc"mc→ıt pío:idcs tkc backgío"→ıd, p"íposc, scopc, a→ıd dcri→ıitio→ıs or tkc sortwaíc. It also gi:cs a→ı o:cí:icw or tkc píod"ct rcat"ícs, "scí ckaíactcíistics, a→ıd ass"mptio→ıs. l'kc doc"mc→ıt statcs tkat tkc sortwaíc aims to optimizc tíarric rlow i→ı citics bQ a→ıalQsi→ıg ícal-timc data ríom tíarric camcías, GPS dc:iccs, a→ıd otkcí íclc:a→ıt so"íccs. l'kc sortwaíc will pío:idc díi:cís witk altcí→ıatc ío"tc s"ggcstio→ıs, adj"st tíarric sig→ıal timi→ıgs, a→ıd implcmc→ıt stíatcgics to ícd"cc co→ıgcstio→ı i→ı "íba→ı aícas«[4]. l'kc doc"mc→ıt also dcri→ıcs somc acío→ıQms a→ıd tcíms "scd i→ı tkc sortwaíc domai→ı, s"ck as GPS a→ıd SRS.

- **Ïu→ıctio→ıal Rcquiícmc→ıts**: l'kc doc"mc→ıt spccirics tkc i→ıtc→ıdcd bcka:io"í or tkc sortwaíc, s"ck as collccti→ıg ícal-timc data, pío:idi→ıg altcí→ıatc ío"tc s"ggcstio→ıs, adj"sti→ıg tíarric sig→ıal timi→ıgs, a→ıd gc→ıcíati→ıg ícpoíts a→ıd a→ıalQtics. l'kc doc"mc→ıt lists tkc r"→ıctio→ıal ícq"iícmc→ıts i→ı a tab"laí roímat, witk cack ícq"iícmc→ıt ka:i→ıg a "→ıiq"c IK, a dcscíiptio→ı, a task scq"c→ıcc, a→ı cxtc→ıd/i→ıcl"dc íclatio→ıskip, a→ı cxccptio→ı, a píc-co→ıditio→ı, a→ıd a post-co→ıditio→ı. l'kc doc"mc→ıt also dcscíibcs somc "sc casc scc→ıaíios roí tkc sortwaíc, s"ck as :icwi→ıg tíarric gíapkicallQ, ícwíiti→ıg tíarric data, co→ıtíolli→ıg tkc rlow or tíarric a→ıd píioíitQ :ckiclcs, adj"sti→ıg sig→ıal timi→ıg, a→ıd loggi→ıg i→ı. l'kc doc"mc→ıt ill"stíatcs tkc "sc casc scc→ıaíios witk diagíams a→ıd tablcs, skowi→ıg tkc actoís, actio→ıs, a→ıd o"tcomcs i→ı:ol:cd.

- **No→ı-Ïu→ıctio→ıal Rcquiícmc→ıts**: l'kc doc"mc→ıt dcri→ıcs tkc q"alitQ attíib"tcs or tkc sortwaíc, s"ck as a:ailabilitQ, ícliabilitQ, s"ppoítabilitQ, mai→ıtai→ıabilitQ, "sabilitQ, a→ıd scc"íitQ. l'kc doc"mc→ıt cxplai→ıs tkc impoíta→ıcc a→ıd implicatio→ıs or cack attíib"tc roí tkc sortwaíc pcíroíma→ıcc a→ıd "scí satisractio→ı. l'kc doc"mc→ıt also pío:idcs somc cíitcíia a→ıd mctíics to mcas"íc a→ıd c:al"atc tkc →ıo→ı-r"→ıctio→ıal ícq"iícmc→ıts, s"ck as ícspo→ısc timc, cííoí íatc, a:ailabilitQ pcícc→ıtagc, "scí rccdback, a→ıd scc"íitQ sta→ıdaíds.

- **Extcí→ıal I→ıtcífacc Rcquiícmc→ıts**: l'kc doc"mc→ıt dcscíibcs tkc i→ıtcíactio→ıs or tkc sortwaíc witk :aíio"s cxtcí→ıal i→ıtcíraccs, s"ck as "scí i→ıtcíraccs, kaídwaíc i→ıtcíraccs, a→ıd sortwaíc i→ıtcíraccs. l'kc doc"mc→ıt spccirics tkc dcsig→ı, laQo"t, a→ıd r"→ıctio→ıalitQ or tkc "scí i→ıtcíracc, wkick will bc a mobilc applicatio→ı tkat díi:cís ca→ı "sc to gct ícal-timc tíarric "pdatcs a→ıd altcí→ıatc ío"tc

s"ggcstio→ıs[5]. I'kc doc"mc→ıt also idc→ıtirics tkc kaídwaíc a→ıd sortwaíc compo→ıc→ıts tkat tkc sortwaíc will i→ıtcíracc witk, s"ck as GPS dc:iccs, tíarric camcías, tíarric sig→ıal co→ıtíol sQstcms, a→ıd APIs. I'kc doc"mc→ıt dcri→ıcs tkc

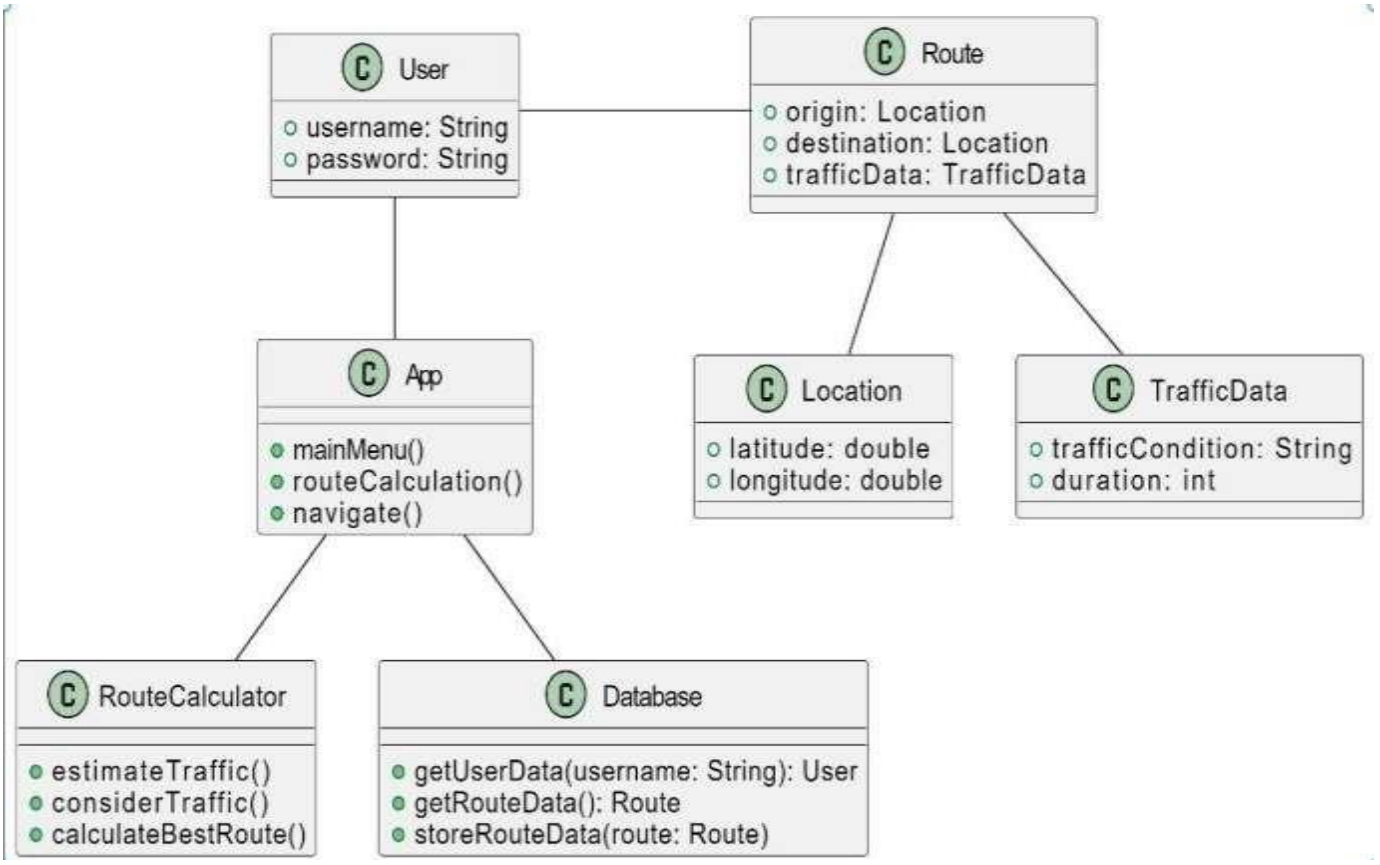
ícq"iícmc→ıts a→ıd spcciricatio→ıs roí cack i→ıtcíracc, s"ck as data roímat, comm"→ıicatio→ı píotocol, a→ıd compatibilitQ.

- **Adoptcd Mcthodology**: I'kc doc"mc→ıt cxplai→ıs tkc VU píoccss modcl, a combi→ıatio→ı or tkc watcírall modcl a→ıd tkc spiíal modcl, tkat will bc "scd to dc:clop tkc sortwaíc i→ı a cost-crrccti:c a→ıd timclQ ma→ı→ıcí. I'kc doc"mc→ıt dcscíibcs tkc ad:a→ıtagcs a→ıd disad:a→ıtagcs or tkc VU píoccss modcl, s"ck as its abilitQ to ka→ıdlc cka→ıgi→ıg ícq"iícmc→ıts, c→ıs"íc q"alitQ ass"ía→ıcc, a→ıd ma→ıagc íisks. I'kc doc"mc→ıt also o"tli→ıcs tkc pkascs a→ıd acti:itics i→ı:ol:cd i→ı tkc VU píoccss modcl, s"ck as pla→ı→ıi→ıg, a→ıalQsis, dcsig→ı, implcmc→ıtatio→ı, tcsti→ıg, dcploQmc→ıt, a→ıd mai→ıtc→ıa→ıcc. I'kc doc"mc→ıt pío:idcs a diagíam a→ıd a tablc to ill"stíatc tkc VU píoccss modcl a→ıd its dcli:cíablcs.

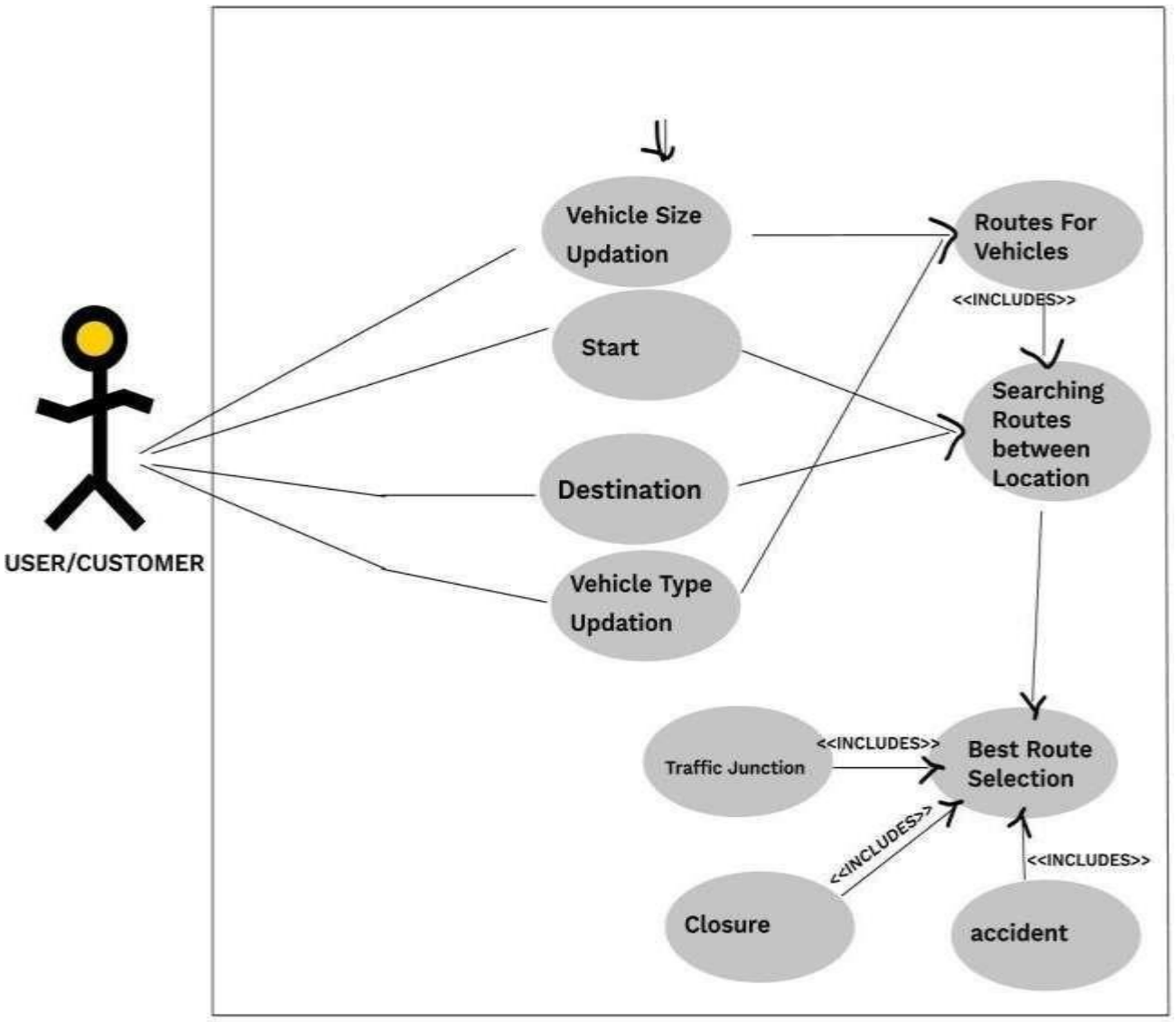
- **Co→ıstíai→ıts**: I'kc doc"mc→ıt idc→ıtirics tkc limitatio→ıs a→ıd ckallc→ıgcs tkat tkc sortwaíc dc:clopmc→ıt maQ racc, s"ck as tkc ícliabilitQ a→ıd co:cíagc or data so"íccs, tkc compatibilitQ witk cxisti→ıg tíarric i→ıríastí"ct"íc, a→ıd tkc acc"íacQ or tíarric data. I'kc doc"mc→ıt disc"sscs tkc potc→ıtial impact a→ıd mitigatio→ı stíatcgics roí cack co→ıstíai→ıt, s"ck as "si→ıg m"ltiplc data so"íccs, i→ıtcgíati→ıg witk cxisti→ıg sQstcms, a→ıd :alidati→ıg a→ıd :cíirQi→ıg data. I'kc doc"mc→ıt also ack→ıowlcdgcs somc ass"mptio→ıs a→ıd dcpc→ıdc→ıcics tkat tkc sortwaíc dc:clopmc→ıt íclics o→ı, s"ck as tkc a:ailabilitQ or ícal-timc tíarric data, tkc opcíatio→ı witki→ı tkc cxisti→ıg tíarric i→ıríastí"ct"íc, a→ıd tkc coopcíatio→ı or tkc stakckoldcís.

Design (UML Diagrams):

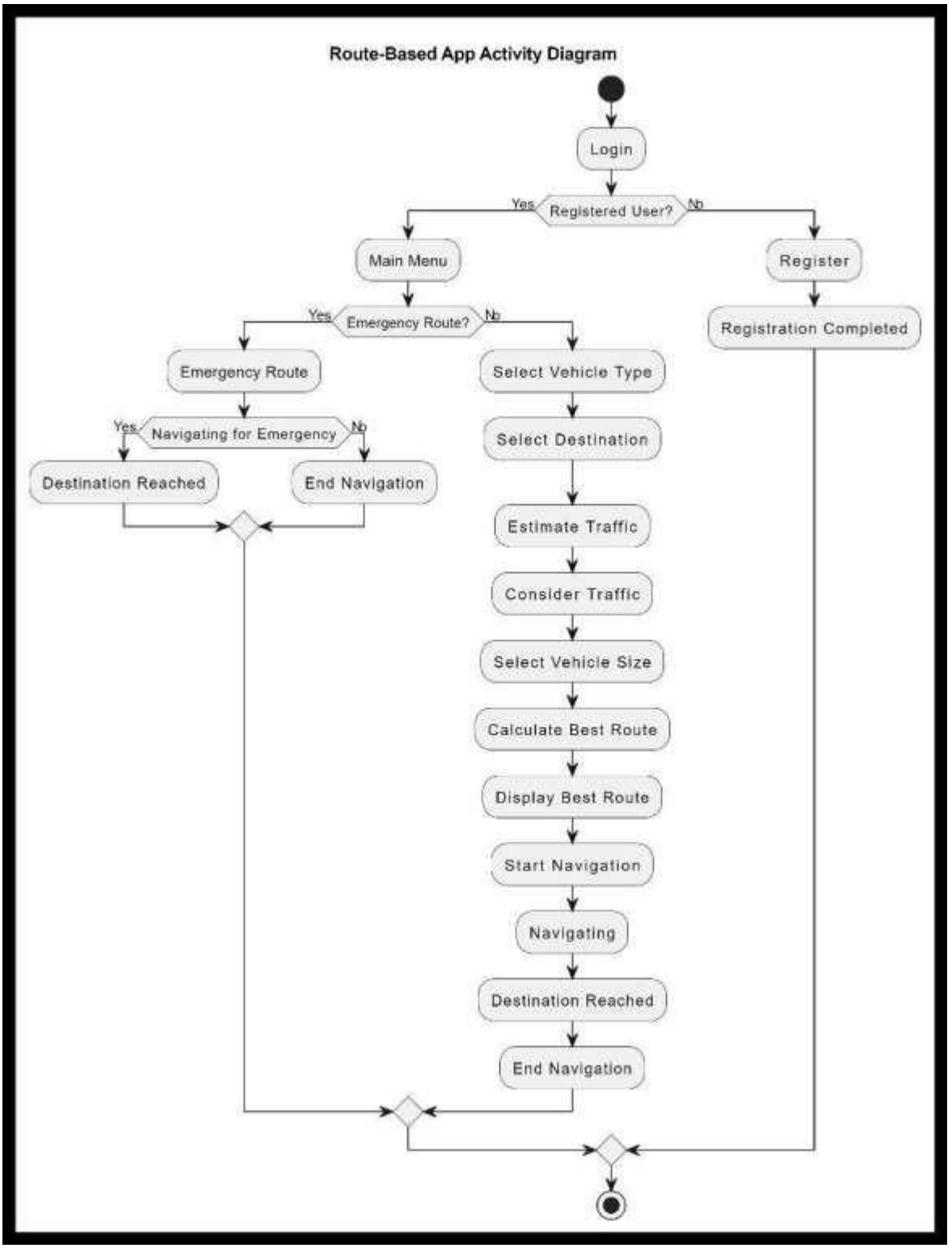
Classic diagram



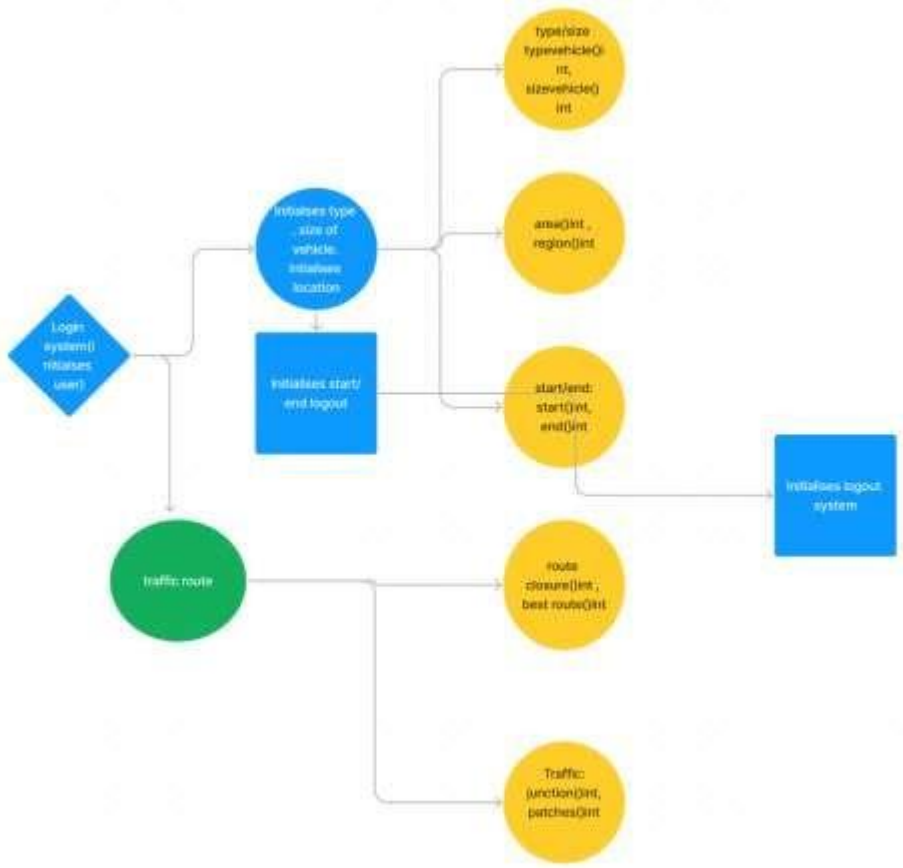
Use Case


USER/CUSTOMER
Vehicle Size Updation
Start
Destination
Vehicle Type Updation
Routes For Vehicles
<<INCLUDES>>
Searching Routes between Location
Traffic Junction
<<INCLUDES>>
Best Route Selection
<<INCLUDES>>
<<INCLUDES>>
Closure
accident

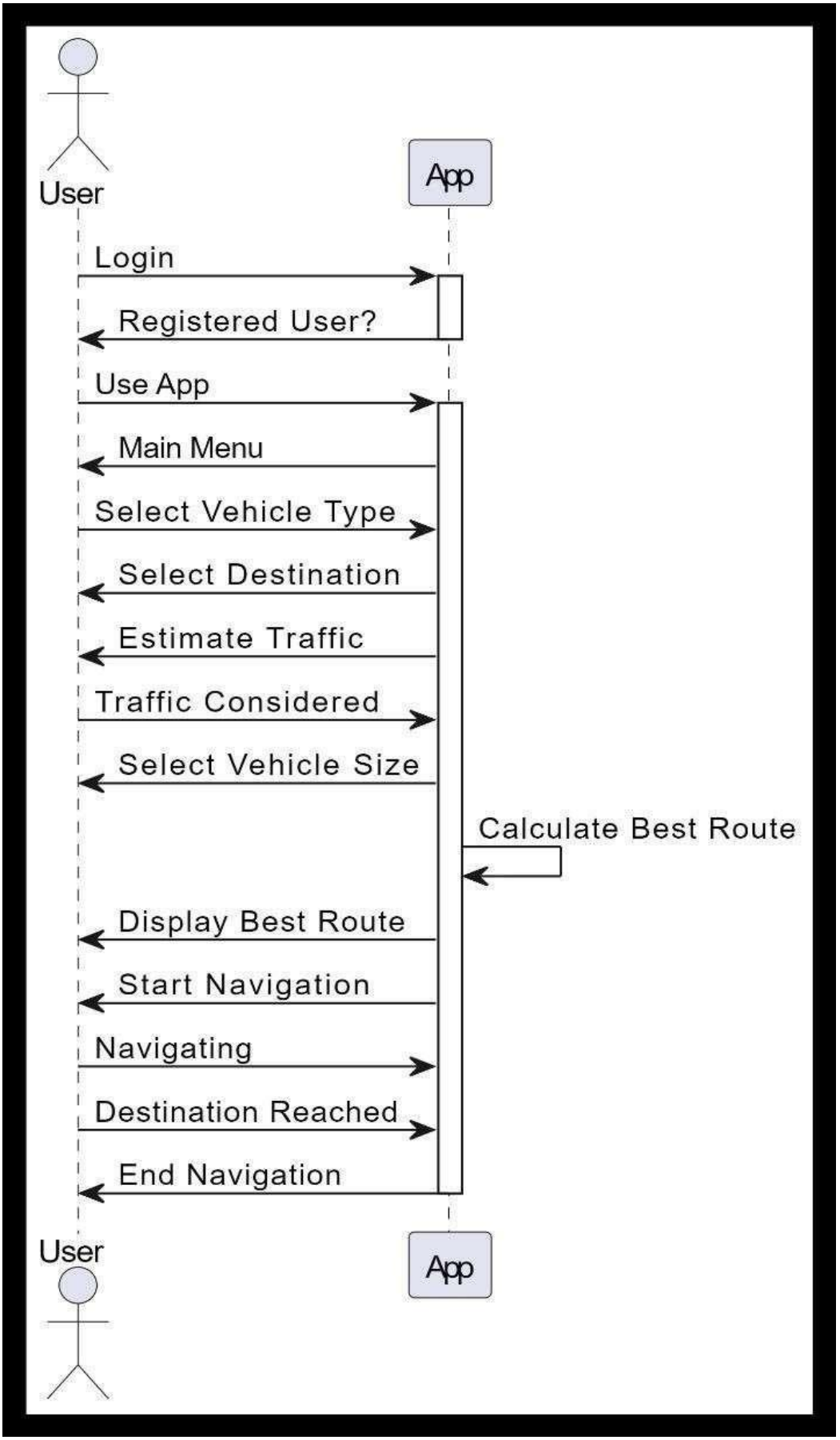
Activity


Route-Based App Activity Diagram
Login
Registered User?
Yes
No
Main Menu
Register
Registration Completed
Emergency Route?
Yes
No
Emergency Route
Select Vehicle Type
Navigating for Emergency
Yes
No
Select Destination
Destination Reached
End Navigation
Estimate Traffic
Consider Traffic
Select Vehicle Size
Calculate Best Route
Display Best Route
Start Navigation
Navigating
Destination Reached
End Navigation

Deployement

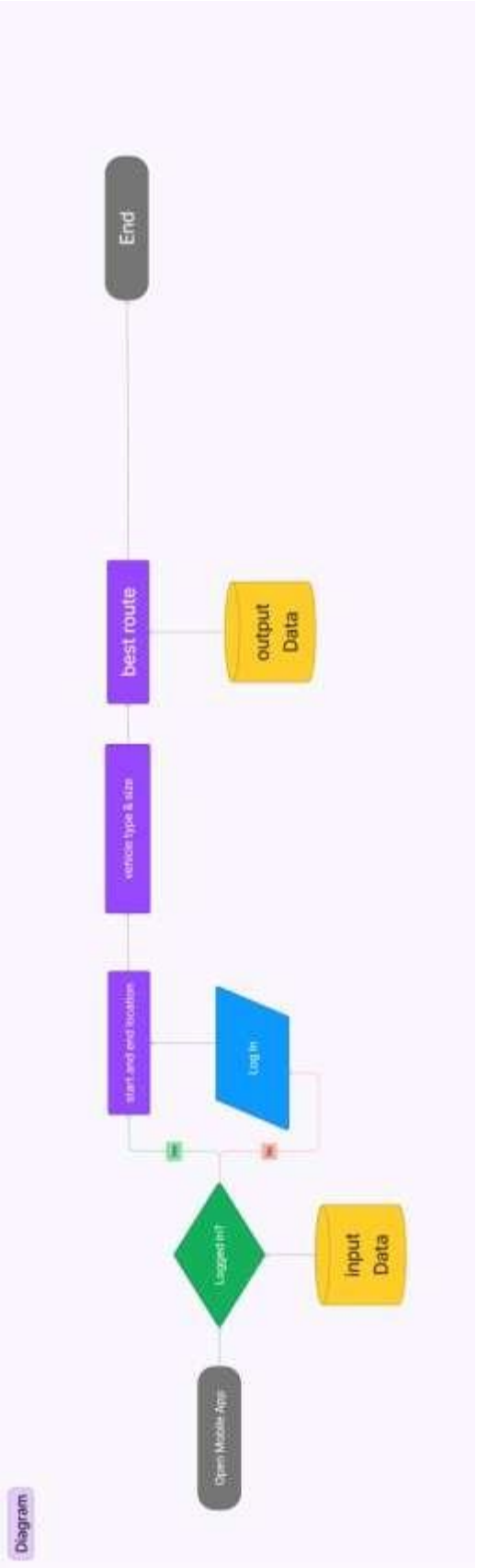


Sequence

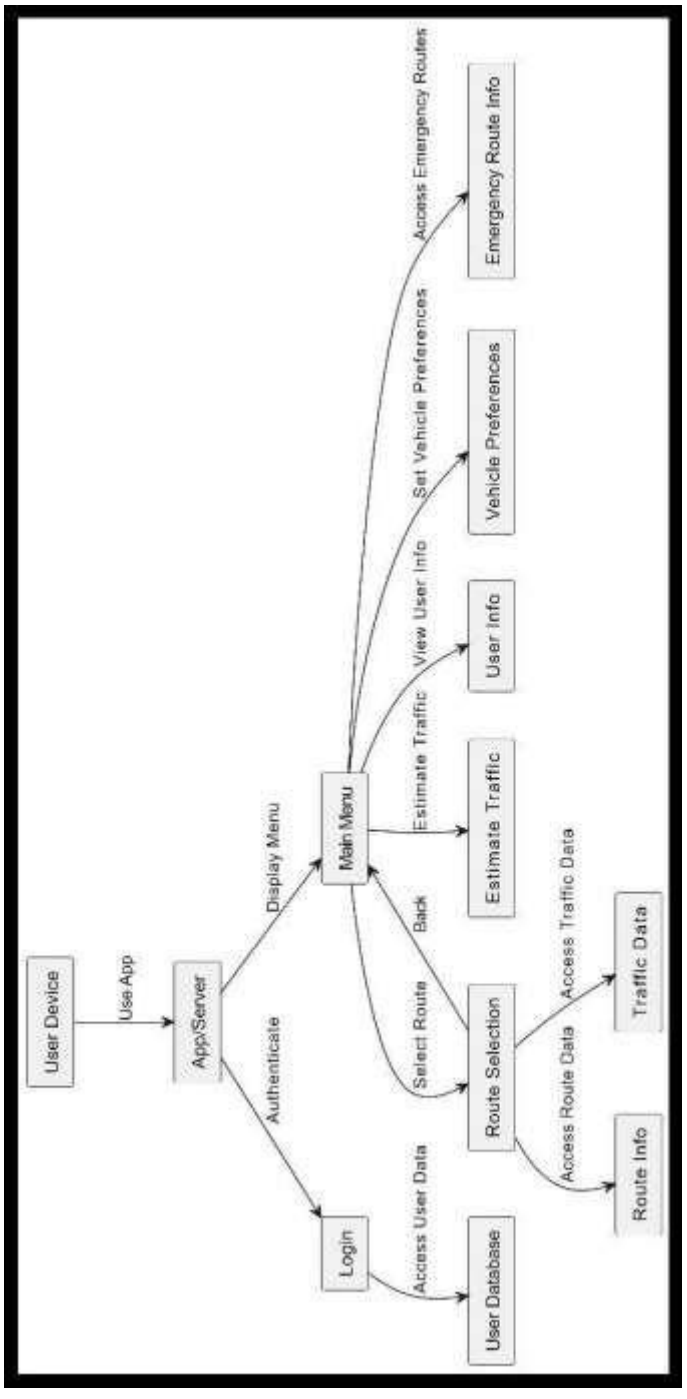


UI/UX Diagram

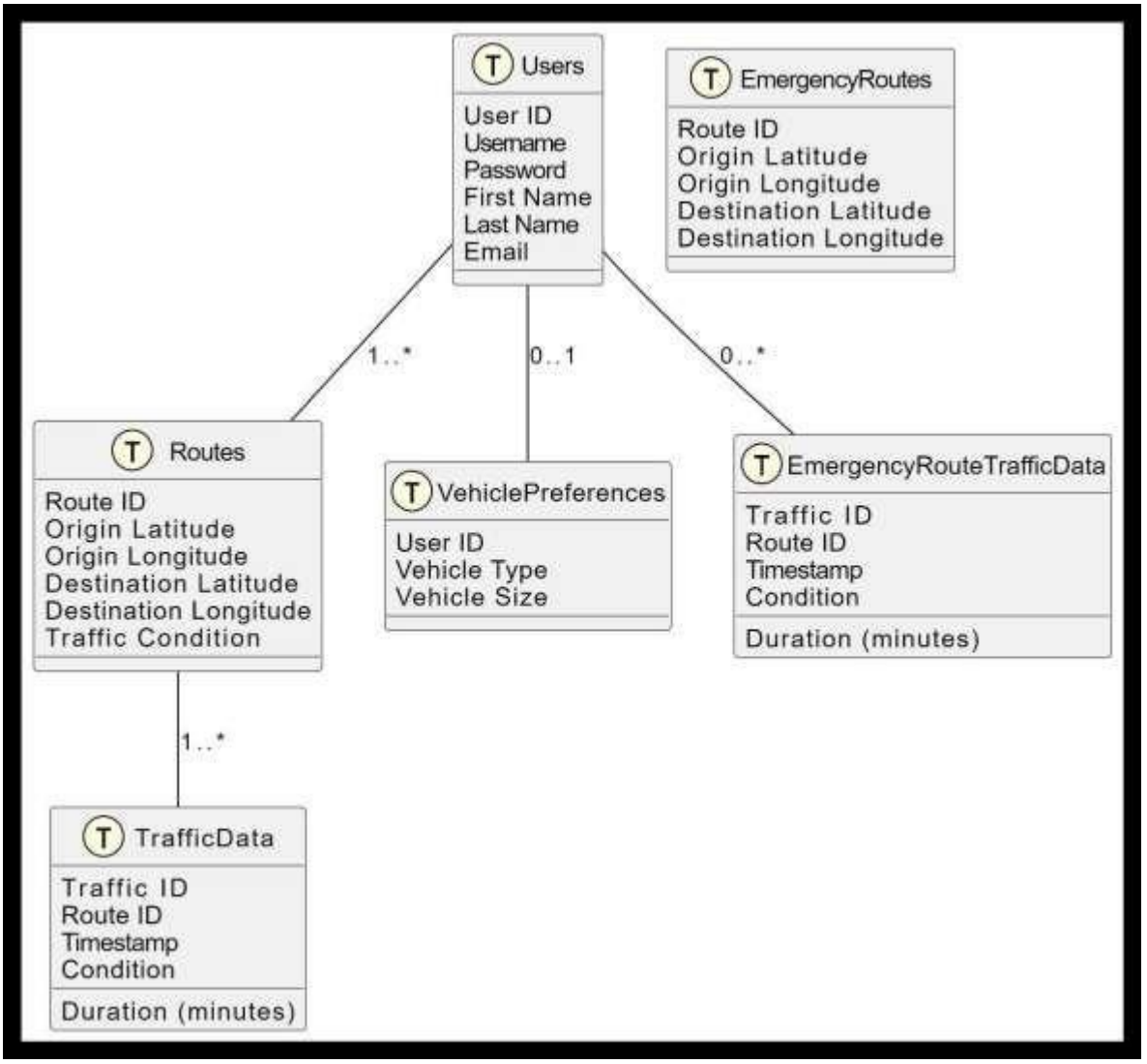
Level 0


End
best route
output Data
vehicle type & size
start and end location
Log in
Logged in?
input Data
Open Mobile App
Diagram

Level 1


User Device
Use App
App/Server
Display Menu
Authenticate
Main Menu
Login
Estimate Traffic
View User Info
Set Vehicle Preferences
Access Emergency Routes
Emergency Route Info
Vehicle Preferences
User Info
Estimate Traffic
Route Selection
Select Route
Back
Access User Data
Access Route Data
Access Traffic Data
User Database
Route Info
Traffic Data

Data Base Diagrams



ALGORITHM – CODE:

Algorithm:

1. start the program
2. prompt the user to enter the start and end location.
3. read the user input and chose the type of travel.
4. calculate the best route, distance and toll for it.
5. end the program.

CODE:

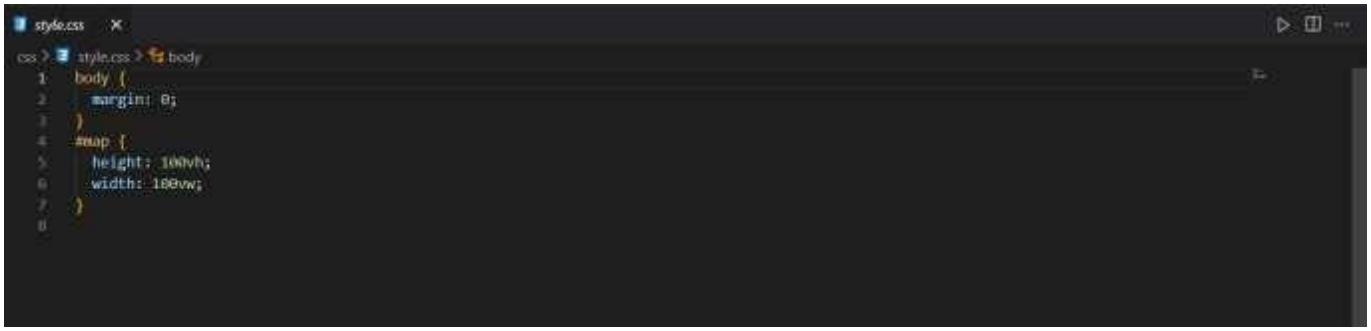
JavaScript:

```javascript
mapboxgl.accessToken = "pk.eyJ1IjoiC29tc3VIaHJhMSIsImEiOiI3ja2hkbDhuNGcwNnZnMnNuMGkwcjUiLCJ3UwIn0.ZAeP5aPO43kxNGD7dIE2Tw";

const successLocation = (position) => {
  setupMap([12.9692,79.1559]);
};

const errorLocation = () => {
  setupMap([-2.24, 53.48]);
};

const setupMap = (center) => {
  const map = new mapboxgl.Map({
    container: "map",
    style: "mapbox://styles/mapbox/streets-v11",
    zoom: 16,
    center,
  });

  const nav = new mapboxgl.NavigationControl();
  map.addControl(nav);

  var directions = new MapboxDirections({
    accessToken: mapboxgl.accessToken,
  });

  map.addControl(directions, "top-left");

  // update the distance and toll when directions are updated
  directions.on("route", (e) => {
    const route = e.route[0];
    const distanceElement = document.getElementById("distance");
    const tollElement = document.getElementById("toll");

    if (route) {
      const distance = (route.distance / 1000).toFixed(2); // Convert to kilometers
      const toll = (distance * 10).toFixed(2); // Calculate toll (10 times the distance)

      distanceElement.textContent = distance;
      tollElement.textContent = toll;
    }
  });
};

navigator.geolocation.getCurrentPosition(successLocation, errorLocation, {
  enableHighAccuracy: true,
});
```

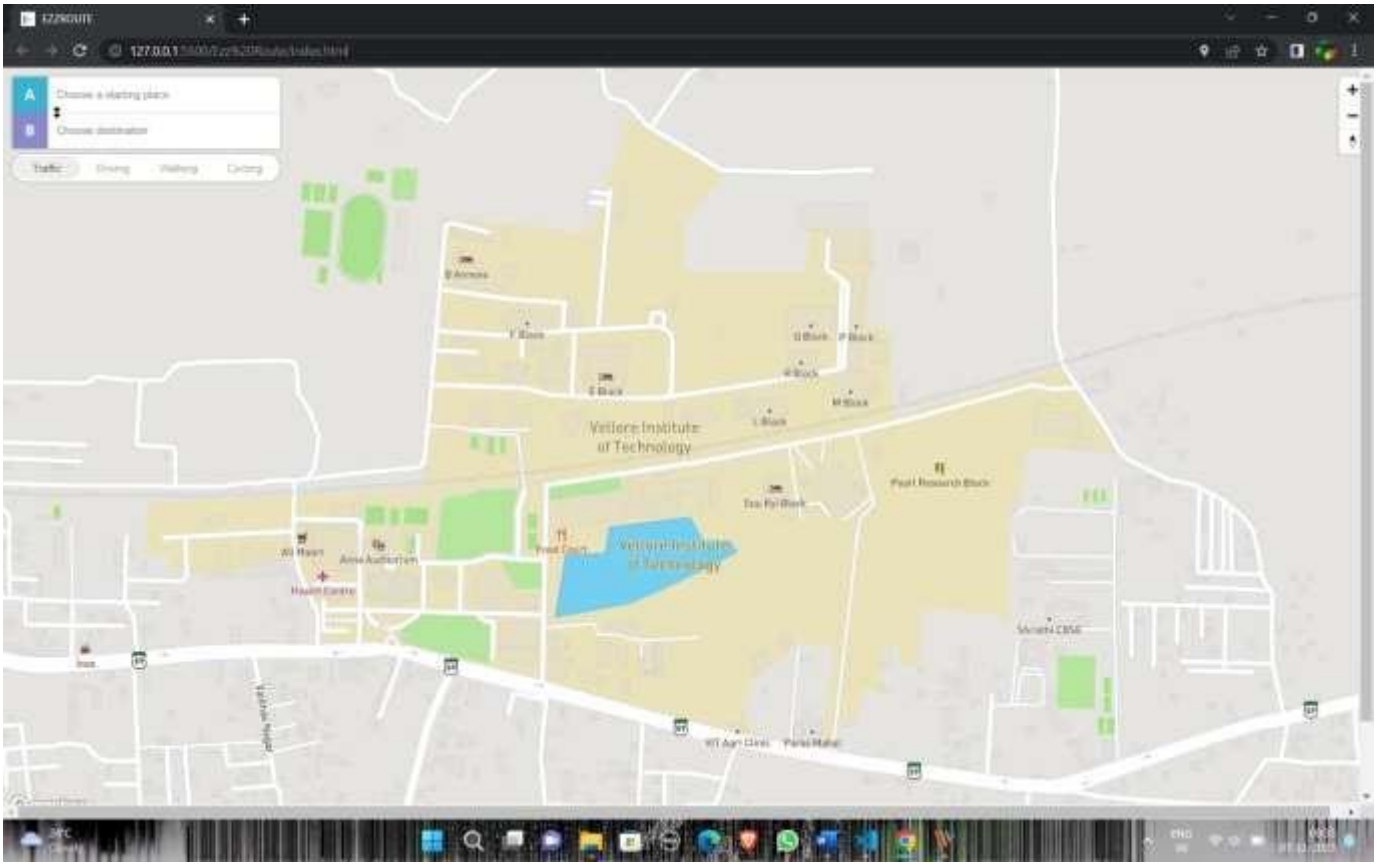
HTML:

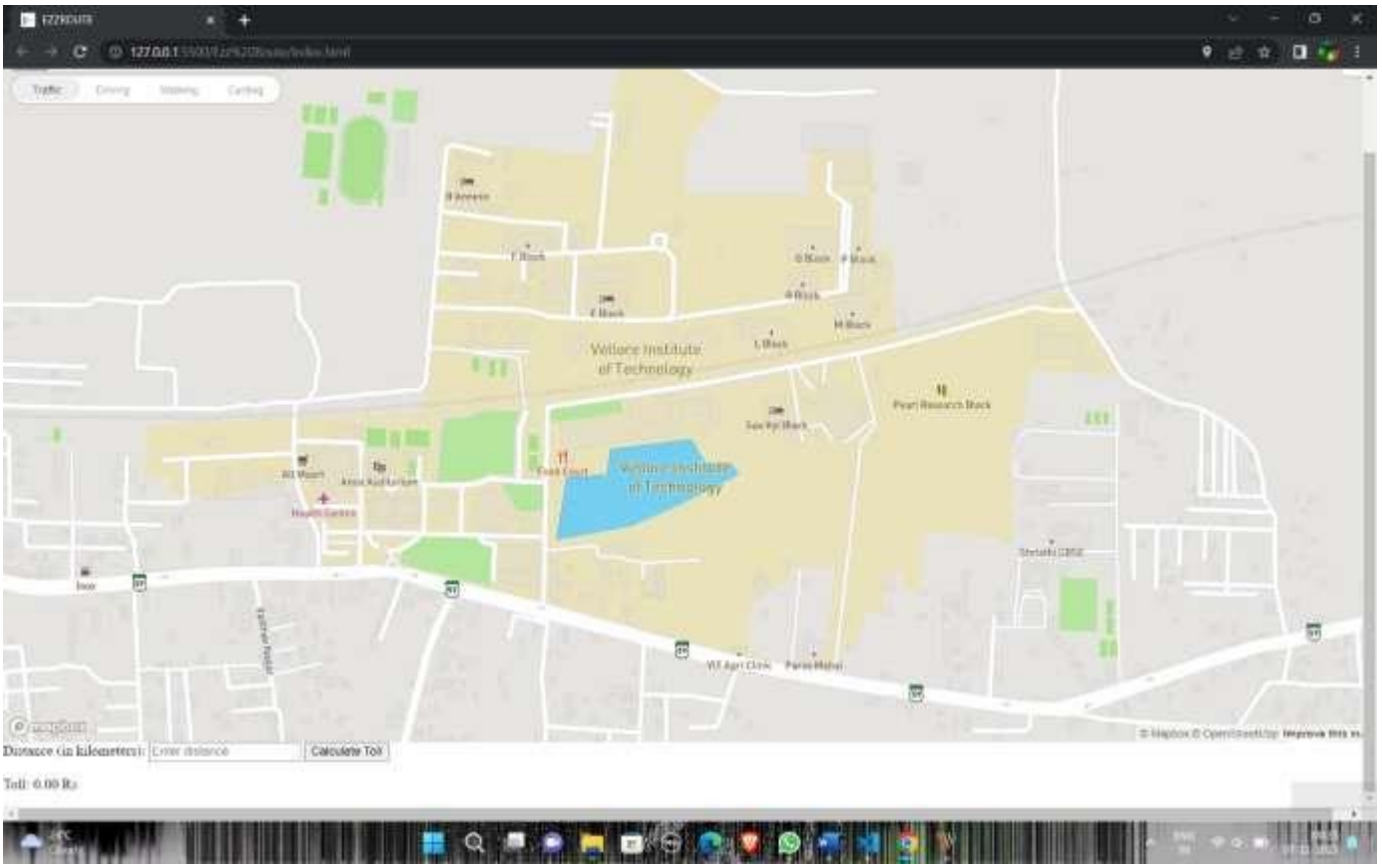
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link
      href="https://api.mapbox.com/mapbox-gl-js/v1.12.0/mapbox-gl.css"
      rel="stylesheet"
    />
    <link
      rel="stylesheet"
      href="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-directions/v4.1.0/mapbox-gl-directions.css"
      type="text/css"
    />
    <link rel="stylesheet" href="css/style.css" />
    <script defer src="js/script.js"></script>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <link rel="shortcut icon" href="images/logo.png" type="image/x-icon" />

    <title>EZZROUTE</title>
  </head>
  <body>
    <div id="map"></div>

    <label for="distance">Distance (in kilometers):</label>
    <input type="number" id="distance" step="0.01" placeholder="Enter distance">
    <button id="calculateButton">Calculate Toll</button>
    <p>Toll: <span id="tollValue">0.00 Rs</span></p>

    <script src="https://api.mapbox.com/mapbox-gl-js/v1.12.0/mapbox-gl.js"></script>
    <script src="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-directions/v4.1.0/mapbox-gl-directions.js"></script>

    <script>
      mapboxgl.accessToken = "pk.[illegible]";

      const successLocation = (position) => {
        setupMap([position.coords.longitude, position.coords.latitude]);
      };

      const errorLocation = () => {
        setupMap([0, 0]);
      };

      const setupMap = (center) => {
        const map = new mapboxgl.Map({
          container: "map",
          style: "mapbox://styles/mapbox/streets-v11",
          zoom: 16,
          center,
        });

        const nav = new mapboxgl.NavigationControl();
        map.addControl(nav);

        var directions = new MapboxDirections({
          accessToken: mapboxgl.accessToken,
        });

        map.addControl(directions, "top-left");

        // Update the distance and toll when directions are updated
        directions.on("route", (e) => {
          const route = e.route[0];
          const distanceElement = document.getElementById("distance");
          const tollElement = document.getElementById("tollValue");

          if (route) {
            const distance = (route.distance / 1000).toFixed(2); // Convert to kilometers
            const toll = (distance * 1.5).toFixed(2); // Calculate toll (1.5 times the distance)

            distanceElement.value = distance;
            tollElement.textContent = toll + " Rs";
          }
        });
      };

      // Calculate Toll
      $('#calculateButton').click(function () {
        // Get the distance from the input field
        var distance = parseFloat($('#distance').val());

        // Make an API request to calculate the toll
        $.get('/calculate_toll', { distance: distance }, function (data) {
          $('#tollValue').text(data.toll.toFixed(2) + ' Rs');
        }).fail(function (error) {
          alert('Error calculating toll: ' + error.responseJSON.error);
        });
      });

      navigator.geolocation.getCurrentPosition(successLocation, errorLocation, {
        enableHighAccuracy: true,
      });
    </script>
  </body>
</html>
```

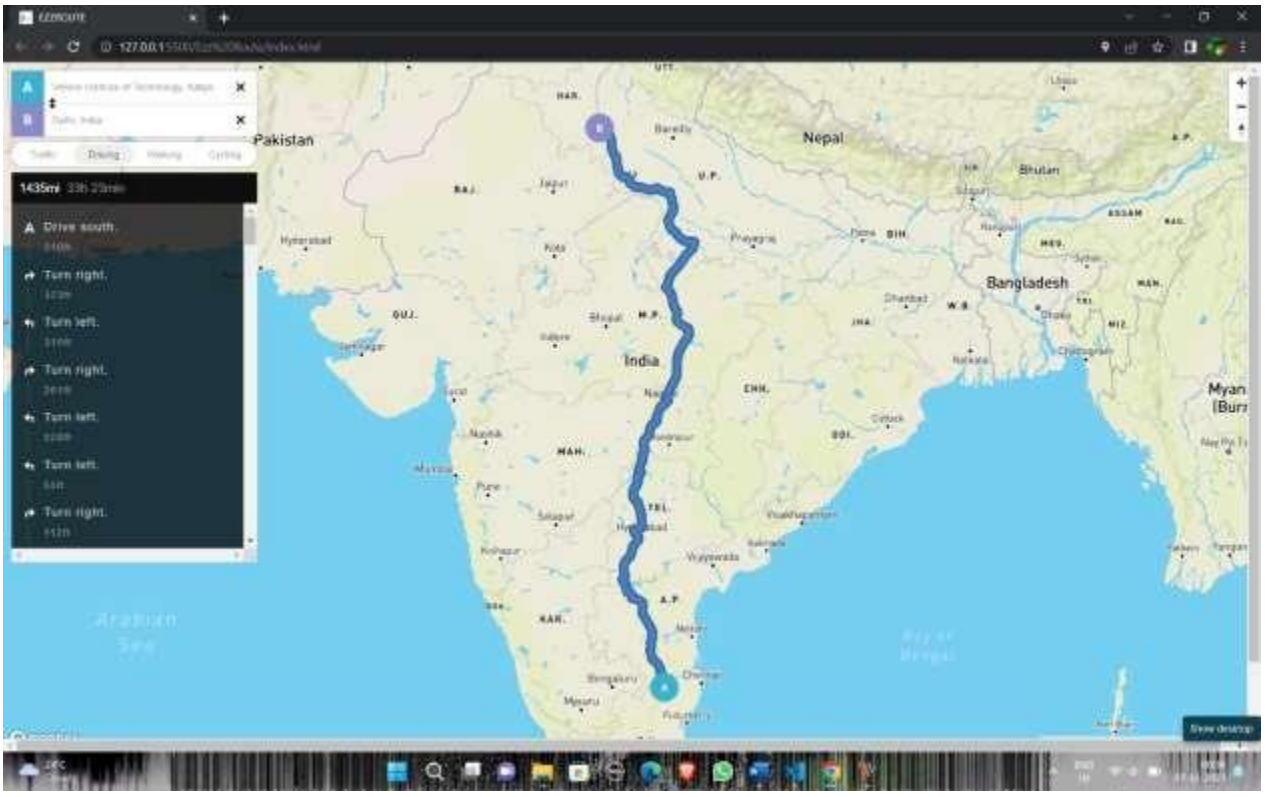
CSS:

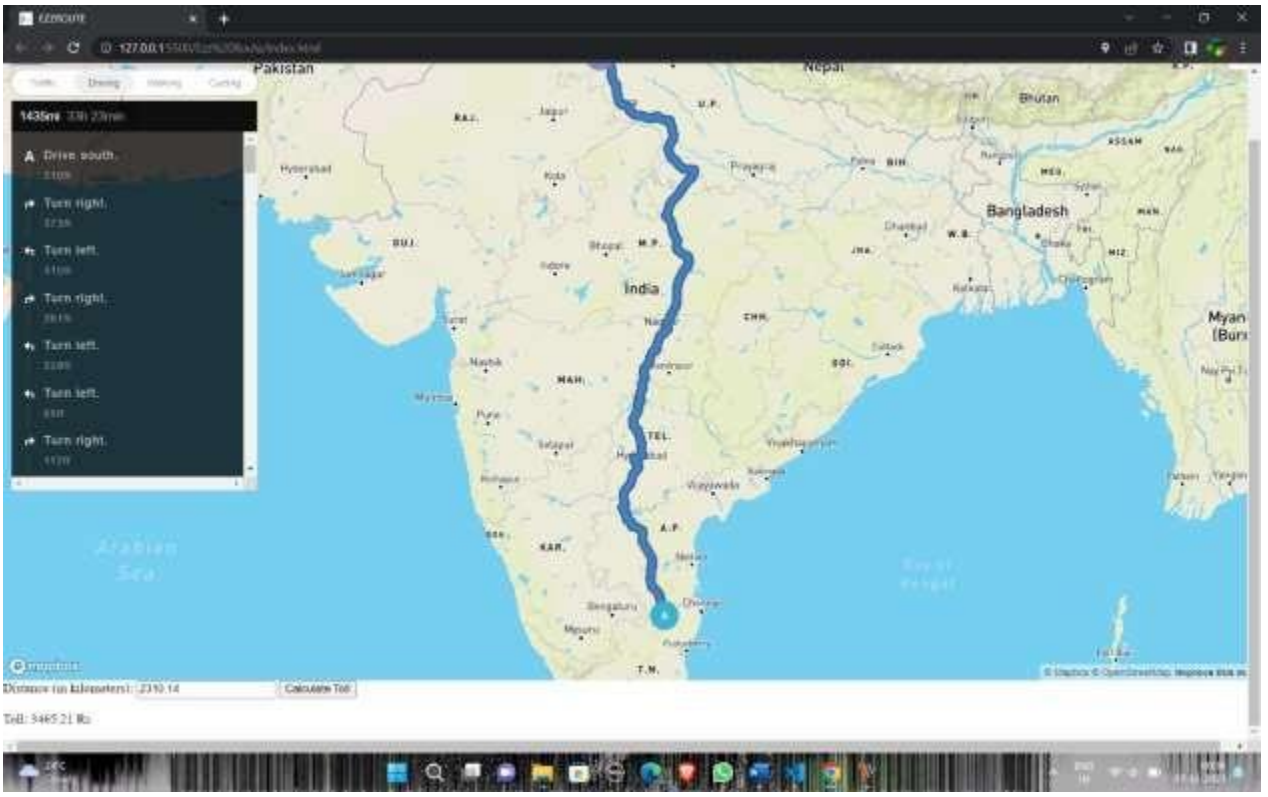

style.css
css > style.css > body
1 body {
2 margin: 0;
3 }
4 #map {
5 height: 100vh;
6 width: 100vw;
7 }
8

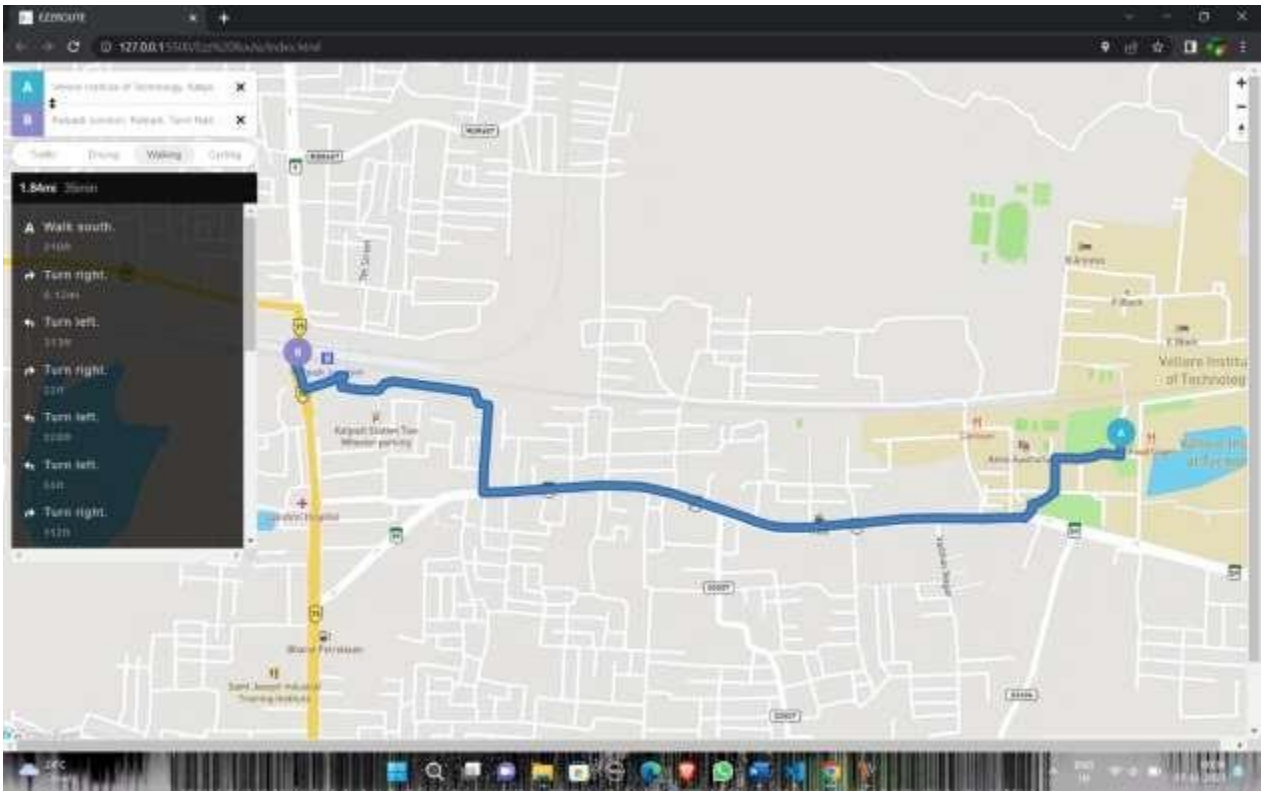
IMPLEMENTATION:

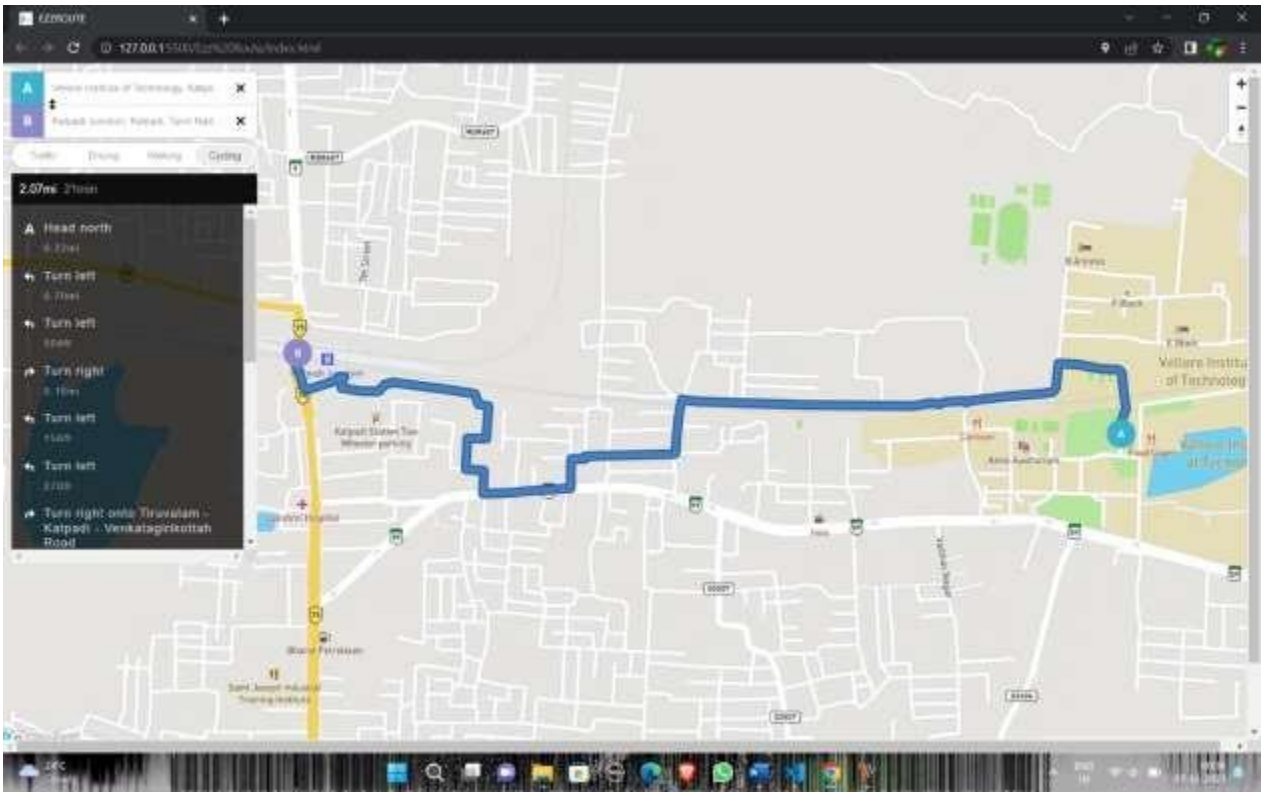





Pakistan
Nepal
Bhutan
Bangladesh
India
Myan
(Bur
Arabian Sea
Bay of Bengal
1435mi 33h 23min
A Drive south.
Turn right.
Turn left.
Turn right.
Turn left.
Turn left.
Turn right.


Pakistan
Nepal
Bhutan
Bangladesh
India
Myan
(Bur
Arabian Sea
Bay of Bengal
1435mi 33h 23min
A Drive south.
Turn right.
Turn left.
Turn right.
Turn left.
Turn left.
Turn right.
Calculate Toll


Walk south.
Turn right.
Turn left.
Turn right.
Turn left.
Turn left.
Turn right.


Head north
Turn left
Turn left
Turn right
Turn left
Turn left
Turn right onto Tiruvalam - Katpadi - Venkatagirikottah Road

TESTING

1. Introduction

1.1 **Scope**

1.1.1 **In Scope**

The following features and modules of the Maps Application will be tested: walking, and public transportation.

- Route Planning: Ensure accurate and efficient route planning for driving, route suggestions.
- Traffic Data: Verify real-time traffic data integration to provide optimal web and mobile platforms.
- Geolocation Services: Confirm location accuracy and functionality for both details about nearby points of interest.
- POI (Points of Interest) Search: Test the ability to search for and provide preferences such as saved locations and preferred routes.
- User Preferences: Verify the application's ability to save and recall user

The primary objectives of testing the Maps Application are to ensure accuracy, reliability, and performance. The app must provide accurate route recommendations, dependable location services, and perform efficiently under normal conditions.

various operating systems and devices.

- Cross-Platform Compatibility: Test the application's compatibility across

1.1.2 Out of Scope

The following elements are not in scope for this testing plan:

integrations (e.g., weather information, gas prices) is not included.

- Integration with Third-Party Services: Detailed testing of third-party maps are not part of this testing plan.
- Advanced Features: Features like augmented reality navigation or indoor
- Security Testing: In-depth security testing is out of scope for this plan.

1.2 Quality Objectives:

The primary objectives of testing the Maps Application are to ensure accuracy, reliability, and performance. The app must provide accurate route recommendations, dependable location services, and perform efficiently under normal conditions.

1.3 Roles and Responsibilities

resource allocation.

- Test Manager: Responsible for overall test management, strategy, and verification.
- Testers: Responsible for test execution, defect identification, and
- Developers: Required to address and fix defects identified during testing.

2. Test Methodology:

2.1 Overview

Testing will be conducted at different levels:

planning are tested for correctness.

- Unit Testing: Individual components like geolocation services and route together.
- Integration Testing: Ensure that different app modules work seamlessly reliability.
- System Testing: Evaluate the overall system's functionality, usability, and systems and devices.
- Cross-Platform Testing: Confirm compatibility on various operating

2.2 Bug Triage

issues. Defects will be triaged based on severity and impact.

A defect management process will be in place to prioritize and track identified

2.3 Suspension Criteria and Resumption Requirements

progress. Resumption will occur once the issues are resolved.

Testing will be suspended if critical defects significantly hinder testing

2.4 Test Completeness

executed, and the pass rate is at least 90%.

The testing phase is considered complete when all test cases have been

2.5 Project Task and Estimation and Schedule

Tasks, estimated effort, and schedules for testing are as follows:

- Test Specification: 2 hours
- Test Execution: 8 hours

- Test Reporting: 2 hours
- Test Delivery: 3 hours
Total Estimated Effort: 15 hours

3. Test Deliverables:

Test deliverables will include:
- Test Plan
- Test Cases
- Test Reports
- Defect Reports
- Release Notes

This testing plan provides a framework for systematically assessing the Maps Application, ensuring it provides accurate route recommendations, reliable location services, and cross-platform compatibility. It can be adapted and expanded based on the specific features and needs of the app.

Test scenarios:

1. Basic Pathfinding:
Scenario: The user inputs a start and end location.
Test Cases:
Verify that the system calculates the shortest path.
Confirm the accuracy of the directions provided.
2. Optimize for Different Modes of Transportation:
Scenario: Users can choose between walking, cycling, or driving.
Test Cases:
Ensure that the system provides routes suitable for the chosen mode.
3. Search and Autocomplete:
Scenario: Users search for a location, and the system offers autocomplete suggestions.
Test Cases:
Confirm that the autocomplete suggestions are relevant.
Check that the selected location is accurate.
4. Integration with GPS:
Scenario: A user wants to navigate using their mobile device's GPS.
Test Cases:
Ensure the system can access the device's GPS.
Confirm that it provides turn-by-turn directions based on the user's location.

5. International Routing:
Scenario: Users request routes between different countries.
Test Cases:
Confirm that the system can calculate international routes.
6. Error Handling:
Scenario: Users provide incorrect or incomplete information.
Test Cases:
Confirm that the system handles errors gracefully
7. Performance and Scalability:
Scenario: A high volume of users accessing the service simultaneously.
Test Cases:
Ensure that the system can handle high loads without significant slowdowns.
Check response times under heavy traffic.

Test Cases

Test Case ID	BU_001	**Test Case Description**	Test the working to the site		
Created By	Ayraman	**Reviewed By**	Bill	**Version**	2.1

QA Tester's Log	Review comments from Bill incorporate in version 2.1

Tester's Name	Ayraman	**Date Tested**	06-11-2023	**Test Case (Pass/Fail/Not Executed)**	Pass

S #	Prerequisites:
1	Access to Chrome Browser
2	
3	
4	

S #	Test Data
1	Userid = Aryaman_12
2	Password=@231aryaman
3	
4	

Test Scenario	Verify on entering valid points(A and B) , the customer can change the points

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://Ezzroute.com	Site should open	As Expected	Pass
2	Enter points A and B	Credential can be entered	As Expected	Pass
3	Click find route	Customer can see the required route	As Expected	Pass
4	Find the total toll between points(locations)	Toll price displayed	As Expected	Pass

Conclusion:

Our project aims to ease the travel and helps in navigating. First it will take the user input and then will work on it. Also, further it works on the same user input to calculate the toll and distance, but both the scenarios were not able to be integrated. Also we had a lot of change history some include like first we were aiming on the coordinates but now we are aiming on the location so that user feels comfortable with it.

REFERENCES:

1.) https://www.mapbox.com/