

---

# **Software Requirements Specification**

**for**

## **PEEWEE**

**Version 2.0 approved**

**Prepared by:**  
**Eugenia Poon (U2222896D)**  
**Ho Zi Qin (U2221837D)**  
**Hamka Bin Dzulkarnain (U2220636E)**  
**Lim Zheng Guang (U2221246G)**  
**Lee Cheng Yao (U2221387J)**  
**Long Jiwei Aaron (U2021740K)**

**<Nanyang Technological University, SDAA, Team Peewee>**

**12 November 2023**

# Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
<b>1.1 Purpose.....</b>	<b>1</b>
<b>1.2 Document Conventions.....</b>	<b>1</b>
<b>1.3 Intended Audience and Reading Suggestions.....</b>	<b>1</b>
<b>1.4 Product Scope.....</b>	<b>2</b>
<b>2. Overall Description.....</b>	<b>3</b>
<b>2.1 Product Perspective.....</b>	<b>3</b>
<b>2.2 Product Functions.....</b>	<b>4</b>
<b>2.3 User Classes and Characteristics.....</b>	<b>5</b>
<b>2.4 Operating Environment.....</b>	<b>6</b>
<b>2.5 Design and Implementation Constraints.....</b>	<b>7</b>
<b>2.6 User Documentation.....</b>	<b>8</b>
<b>2.7 Assumptions and Dependencies.....</b>	<b>9</b>
<b>3. External Interface Requirements.....</b>	<b>10</b>
<b>3.1 User Interfaces.....</b>	<b>10</b>
<b>3.2 Hardware Interfaces.....</b>	<b>13</b>
<b>3.3 Software Interfaces.....</b>	<b>14</b>
<b>3.4 Communications Interfaces.....</b>	<b>15</b>
<b>4. System Features.....</b>	<b>16</b>
<b>4.1 Registration.....</b>	<b>16</b>
<b>4.2 Sign In.....</b>	<b>16</b>
<b>4.3 View Traffic Conditions.....</b>	<b>17</b>
<b>4.4 View Traffic Incidents.....</b>	<b>17</b>
<b>4.5 Report Traffic Incidents.....</b>	<b>18</b>
<b>4.6 View Map.....</b>	<b>18</b>
<b>4.7 Save and View Routes.....</b>	<b>19</b>
<b>5. Nonfunctional Requirements.....</b>	<b>20</b>
<b>5.1 Performance Requirements.....</b>	<b>20</b>
<b>5.2 Safety Requirements.....</b>	<b>20</b>
<b>5.3 Security Requirements.....</b>	<b>20</b>
<b>5.4 Software Quality Attributes.....</b>	<b>21</b>
<b>5.5 Business Rules.....</b>	<b>21</b>
<b>6. Use Cases.....</b>	<b>22</b>
<b>6.1 Use Case Model.....</b>	<b>22</b>
<b>6.2 Use Case Notations.....</b>	<b>23</b>
<b>6.3 Use Case Specifications.....</b>	<b>23</b>
<b>7. Other Requirements.....</b>	<b>45</b>
<b>8. System Testing.....</b>	<b>47</b>
<b>8.1 Black Box Testing.....</b>	<b>47</b>

8.2 White Box Testing.....	50
9. Appendices.....	57
Appendix A: Data Dictionary.....	57
Appendix B: Analysis Models.....	58
Appendix C: User Interface Screenshots.....	83

## Revision History

Name	Date	Reason For Changes	Version
Everyone	30 Oct 2023	Initial Draft	1.0
Everyone	12 Nov 2023	Final Documentation	2.0

# 1. Introduction

## 1.1 Purpose

The purpose of this SRS is to provide a detailed description of the requirements for the development of PEEWEE (Predictive Environment for Easy Wayfinding with Enhanced Efficiency), a web application. This SRS will outline the functional and non-functional requirements of the application including, use case models, class diagrams, sequence diagrams, dialog maps, user interface design, system architecture and other requirements. This document will also serve as a reference for the development team, stakeholders and clients. This is to ensure everyone involved in this project is clear and understands its objectives.

## 1.2 Document Conventions

- A. Format: .pdf file
- B. Font: Times New Roman
- C. Font-style: Headings will be in font size 18, subheadings will be in font size 14, and font size 11 for body content.
- D. Line height: 1.15

## 1.3 Intended Audience and Reading Suggestions

This SRS document is intended for all stakeholders involved in the making and implementation of PEEWEE software system, including

- A. Software developers: To understand the technical details and constraints of the system.
- B. Project Managers: To ensure the project is on track within deadline, budget and caters to the client's requirements.
- C. End Users: To reference how the PEEWEE software functions, its features and their responsibilities when utilising the software.

Users are not required to follow this document in a sequential manner; they are encouraged to navigate through the contents page to any section they deem pertinent. A brief overview of each section of our document is shown below.

- A. Introduction: Introduces a comprehensive understanding of the project's background, objectives, and scope.
- B. Overall Description: This section grasps a broader context of the PEEWEE system, including its functionalities and constraints.
- C. External Interface Requirements: This section seeks to explain how the system's external interface interacts with other systems or users.
- D. System Features: This section provides detailed insights into the specific functionalities and capabilities of the system.

- E. Non-functional Requirements: This section contains the system's performance, security, and other aspects that are not directly related to specific functionalities.
- F. Use cases: The use cases provide a practical understanding of how different types of users would interact with the system and the specific scenarios they are likely to encounter.
- G. Other requirements: This section presents a holistic view of any additional aspects or constraints that might impact the system's development or usage.
- H. System testing: Testing strategies and methodologies that were employed to ensure the system's functionality, performance, and reliability.
- I. Appendices: Any supplementary information which could provide a more comprehensive understanding of the system.

## **1.4 Product Scope**

PEEWEE is a specialised web application crafted to enhance the daily lives of frequent drivers in Singapore. Its primary goals involve alleviating the challenges associated with regular commuting and fostering the widespread integration of technology in streamlining this convenience. By leveraging PEEWEE, users can access information regarding the least congested routes to their destinations and stay updated on incidents or road blockages through shared updates from other users within the app.

## 2. Overall Description

### 2.1 Product Perspective

#### Context and Origin

PEEWEE is envisioned as an innovative step forward in the domain of real-time traffic navigation and wayfinding solutions. It is not merely an iteration of existing mapping technologies but a new, standalone application focused on delivering unparalleled accuracy and immediacy in traffic condition reporting. The concept of PEEWEE originated from the identified gap in current navigation solutions, which do not fully capture or relay traffic conditions in real-time, resulting in less optimal route planning and increased travel times.

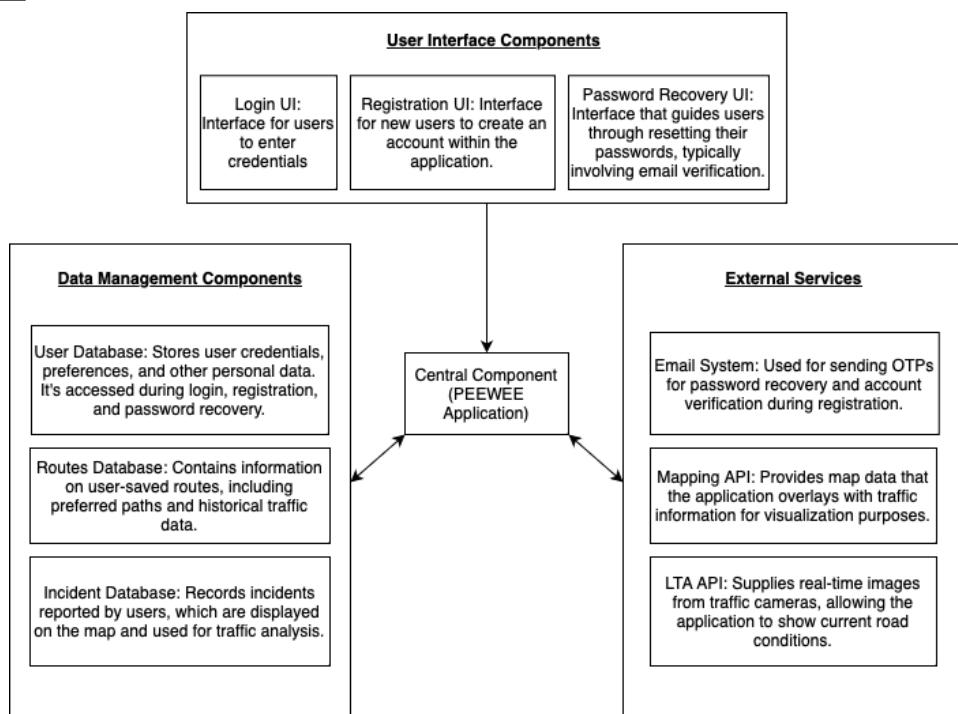
#### Relationship to Existing Systems

PEEWEE aims to interface with various data streams, including congestion levels, live traffic cameras images, through employing the use of advanced predictive algorithms to inform drivers of potential delays. The application features a user-friendly interface for commuters and communicates with Land Transport Authority(LTA) API to fetch and update traffic data simultaneously. PEEWEE aims to interact with smartphone hardware such as GPS and mobile data to ensure a seamless user experience.

#### Project Completion Criteria

Peewee aims to reach its completion when the application undergoes real-world trials, and once successful, it will then receive endorsement from the LTA. The approval will mark the readiness of PEEWEE for public release and its contribution to the Smart Nation initiative.

#### Brief Diagram



## 2.2 Product Functions

### 1. User Account Management

- A. Login: Allows the user to securely access their account
- B. Register: Enables new user to create new account
- C. Forget Password: Assists user in resetting password if forgotten

### 2. Traffic Analysis and Display

- A. View Traffic Camera Images: Users can view live images from various traffic cameras to assess current road conditions.
- B. Analyse Traffic Images: The system automatically analyses traffic camera images to detect congestion trends.
- C. Show Traffic Trends: The application provides insights into traffic trends based on historical and real-time data, including plotting out live traffic heatmap based on the camera's peakedness out for each hexagon covering an specific area.

### 3. Route Management and Visualization

- A. Visualise on Map: Users can view their routes on a map, complete with real-time traffic conditions.
- B. Create Route: Users can define custom routes.
- C. Favourite Route: Users can save custom routes.
- D. Unfavourite Route: Allows users to remove previously favorited routes.
- E. View Favourites: Displays a list of all saved routes for the user.

### 4. Incident Reporting and Viewing

- A. Report Incident: Users can manually report traffic incidents not detected by the system.
- B. View Reported Incidents: Allows users to view incidents reported by other users.

## 2.3 User Classes and Characteristics

### Casual Commuters:

1. **Frequency of Use:** Sporadic, typically during daily commutes.
2. **Functions Used:** Basic route searching, traffic updates.
3. **Technical Expertise:** Basic. Prefer user-friendly interfaces.
4. **Characteristics:** These users primarily want quick traffic updates and the best routes for their daily commutes. They might not be interested in advanced features or analysis.

### Frequent Travellers:

1. **Frequency of Use:** Regular, often multiple times a week.
2. **Functions Used:** Route searching, saving routes, real-time traffic analysis, incident reporting.
3. **Technical Expertise:** Moderate. Comfortable with more advanced features.
4. **Characteristics:** Users who often travel for work or leisure and need more in-depth information and flexibility from the system.

### System Administrators:

1. **Frequency of Use:** Daily.
2. **Functions Used:** All functionalities, with special focus on system monitoring, user management, and incident verification.
3. **Technical Expertise:** High. Need to understand both the user interface and backend processes.
4. **Characteristics:** Responsible for the smooth operation of the PEEWEE system, ensuring data accuracy, and addressing user concerns.

### Emergency Responders:

1. **Frequency of Use:** As required based on incidents.
2. **Functions Used:** Incident reports, traffic analysis, route blockage features.
3. **Technical Expertise:** Moderate.
4. **Characteristics:** Users who need to respond to traffic emergencies. They require accurate and immediate data about incidents and route blockages.

### Most Important User Classes for Satisfaction:

- (1) System Administrators: Ensure System Overall Functionality and service to users.
- (2) Emergency Responders: Address Immediate and critical Situations.

### Least Critical User Classes for Satisfaction:

- (1) Casual Commuters: Uses Limited subset of functions and have alternatives for basics needs.

## 2.4 Operating Environment

### Hardware Platform:

Mobile Devices: Compatible with smartphones and tablets having at least 2GB RAM and 16GB storage. GPS functionality is required for real-time location tracking.

Desktop/Laptop Computers: Requires a minimum of 4GB RAM, 2GHz dual-core processor, and 20GB of free disk space.

Server Infrastructure: Optimised for cloud server platforms, with scalability options for increased user loads. Requires SSD storage for quick data access, a minimum of 8GB RAM, and multi-core processors.

### Operating System and Versions:

Mobile: Compatible with iOS (version 13 and above) and Android (version 10 and above).

Desktop: Supports Windows (10 and 11), macOS (Catalina, Big Sur, and newer), and Linux distributions (Ubuntu 18.04 and newer).

Server: Optimised for Linux server distributions, especially Ubuntu 20.04 and Rocky Linux 8.

Software Components:

Database: Uses MongoDB (version 7.0 and above) for storing user data, routes, and traffic information. It can coexist with other database management systems but requires dedicated resources.

Web App: Built on the React framework. Uses Axios to perform HTTPS requests with the backend.

Backend: Developed using Node.js (version 14 and above), mongoDB, and can run alongside other backend services using different ports.

### Coexisting Software:

The PEEWEE system can coexist with antivirus software, firewalls, and other security software. However, it's essential to whitelist the PEEWEE app to ensure uninterrupted data flow.

Integration capabilities with third-party mapping software such as Google Maps, allowing for enhanced route suggestions and data sharing.

## 2.5 Design and Implementation Constraints

**Corporate or Regulatory Policies:** The PEEWEE system must adhere to PDPA data protection and privacy regulations, which may limit certain data processing capabilities.

**Hardware Limitations:** The software's performance may be constrained on devices with RAM below 2GB, affecting responsiveness and overall user experience.

**Parallel Operations:** Implementing multitasking with lower processing power may lead to performance degradation and slower response times for each task.

**Language Requirements:** The primary development language will be TypeScript. If the development team lacks expertise in Typescript, the development process could be slowed down.

**Communication Protocols:** Data transfer will use the HTTPS protocol (CRUD), ensuring secure communication between the client and server. It may introduce a slight overhead in terms of data transfer speed, especially on devices with limited processing power.

**Security Considerations:** Two-factor authentication must be implemented for user logins. Users without access to a secondary authentication method may find difficulties in the login process. All user data must be encrypted both at rest and in transit. This may limit and prolong development.

**Design Conventions or Programming Standards:** Development follows the MVC (Model-View-Controller) design pattern, this may increase development time and complexity if the development team is not familiar with the architectural approach.

## **2.6 User Documentation**

This section details the envisioned documentation to guide users in the usage of PEEWEE.

**User Manuals:** We will be providing a comprehensive guide detailing all the features and functionalities of the PEEWEE system.

**Tutorials:** We will be expanding our repository of tutorials with video and written guides that focus on key functionalities like including how to navigate around the application, use some of its basic functionalities.

**FAQs:** The FAQ segment, both within the application and on the website, will be updated continually to address the evolving questions and challenges that consumers face through feedback or test-run.

**Release Notes:** We will be informing users of documented updates for each new version of the PEEWEE system, including new features and bug fixes.

## **2.7 Assumptions and Dependencies**

### **Assumptions:**

**Development Environment:** It is assumed that developers will use the latest version of Visual Studio or any other form of compatible IDE, and have access to a test server with similar specifications to the production environment.

**Third-party Tools:** The project assumes that all licences for third-party tools such as MongoDB will be procured and available for integration.

**Operating Conditions:** It's assumed that end-users will have a consistent internet connection to access cloud-based features.

### **Dependencies:**

**External Libraries:** The software will integrate with the MongoDB database system for storing of information.

**Cloud Services:** The project will rely on GitHub for version control and data synchronisation across devices.

## 3. External Interface Requirements

### 3.1 User Interfaces

#### UI Standards

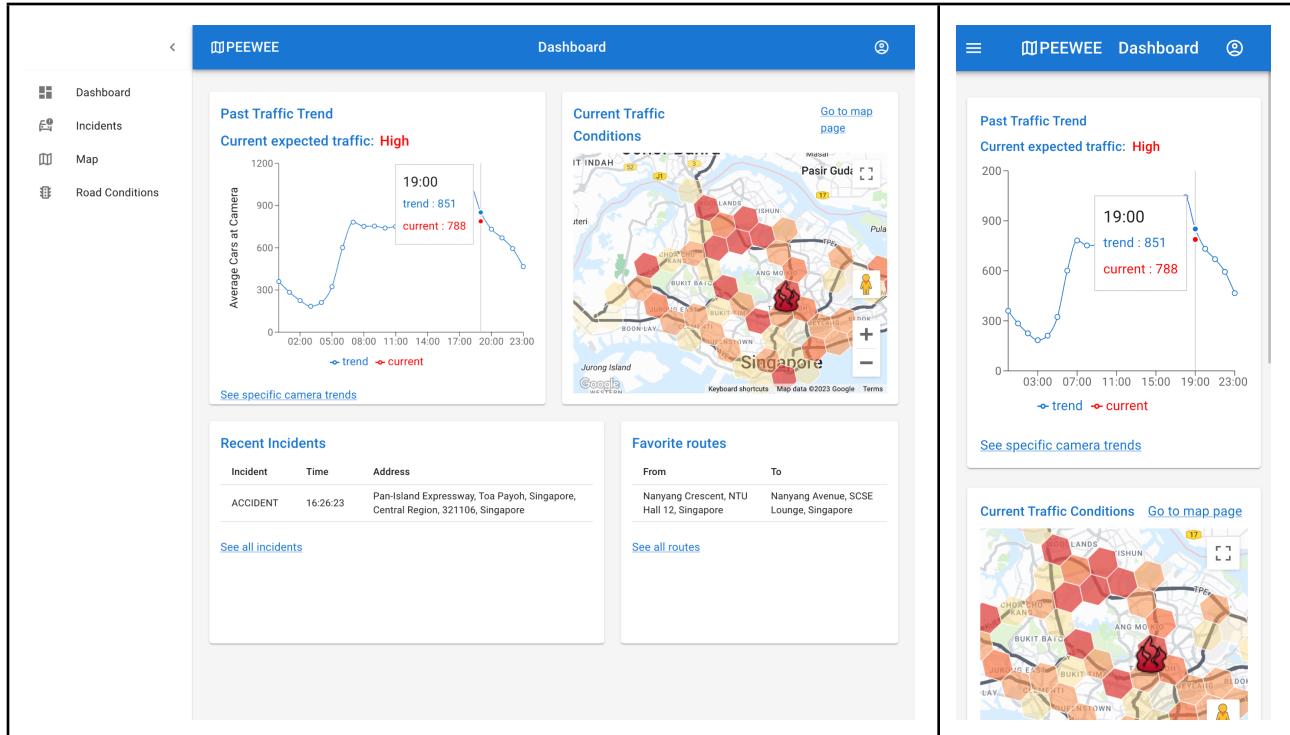
Peewee utilises the Material Design System. Created by Google, it helps us build high quality, familiar, and intuitive user experiences. The design extends across all platforms, from mobile devices such as iOS and Android to the Web. In Material Design there is a great emphasis on a responsive layout, allowing the interface to dynamically adapt to different screen sizes and orientations, which will be elaborated further in later sections.

Our team worked closely with the Material Design Guidelines to introduce interface components such as the App Bar, Typography, Cards, and Snackbars in our application. By following these design guidelines, we are conforming to the best practices of UI design, providing the best experience to our users.

In our implementation, Material Design is used in our React front-end through a package called Material-UI, allowing our team to easily implement its design principles into our web application.

## Responsive Design

A responsive design is integral to our potential users, who will be using our web application while on the go. Hence there's an emphasis to optimise our application for mobile clients, while still making use of the large screens on desktop clients. We applied the responsive layout grid detailed in the Material Design Guidelines to create a dynamic, responsive user interface for our users, adapting to various screen sizes.



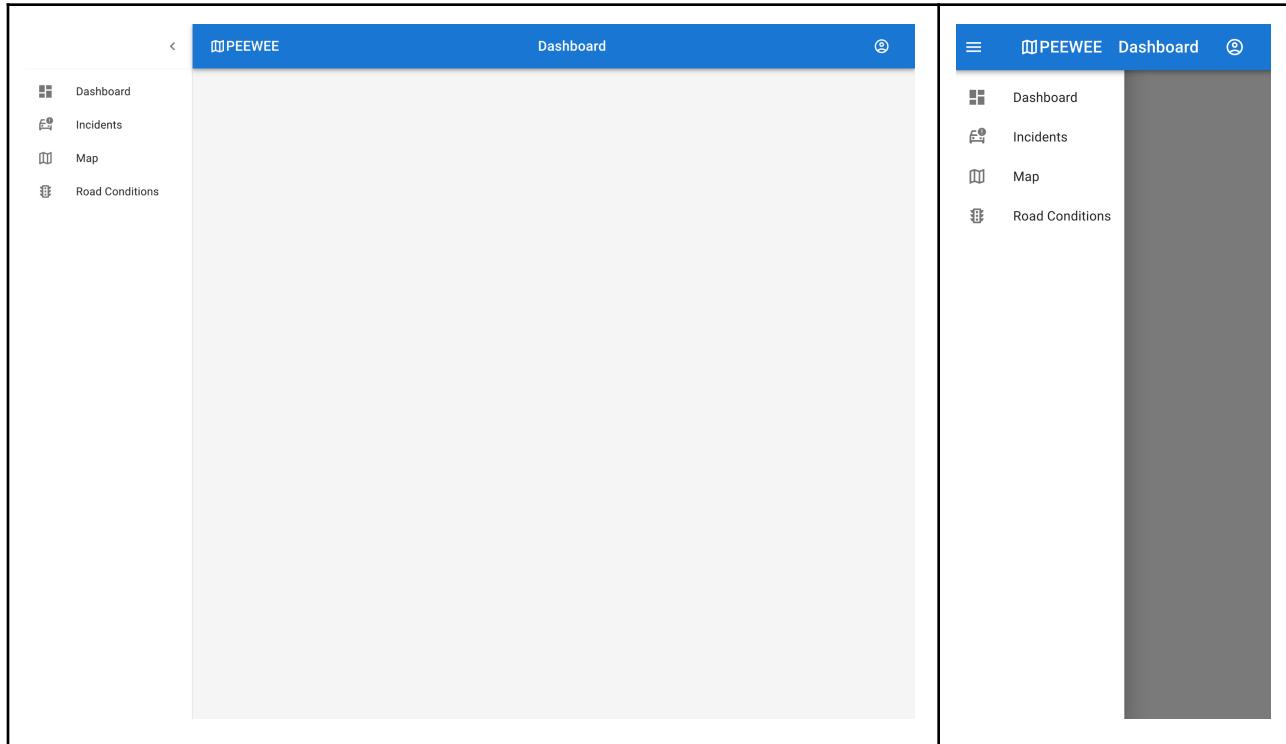
**Figure 1:** Dashboard page rendered on the desktop and mobile respectively

The rest of our UI documentation will follow the same convention presented by Figure 1, with the desktop screenshot on the left and a mobile one on the right.

Referring to the same figure, our UI dynamically scales according to the device's screen sizes. On a mobile, the Menu component is hidden and the Typography components are scaled down comfortably. To achieve this, we used breakpoints as described in the Material Design Guidelines to detect smaller screens and optimise the UI accordingly. Similar responsive optimisations have been made to the current page, and throughout all pages in our web application to ensure a seamless, on the go mobile experience.

## Standard Functions

An App Bar and a collapsible Drawer is visible on all screens in our application to facilitate a seamless navigation experience. It displays the current page the user is viewing, and allows the user to open the drawer by clicking on the menu button.



**Figure 2:** The static App Bar component and its collapsible Drawer component

Alert components are used to display any errors that users may encounter from day to day use. Although they are not described in the Material Design Guidelines, we used them here in our UI to capture the user's attention without interrupting their task. It is displayed inline with the rest of our UI, as compared to Snackbar components which may appear too briefly to properly convey the error as well as corrective steps to the user.

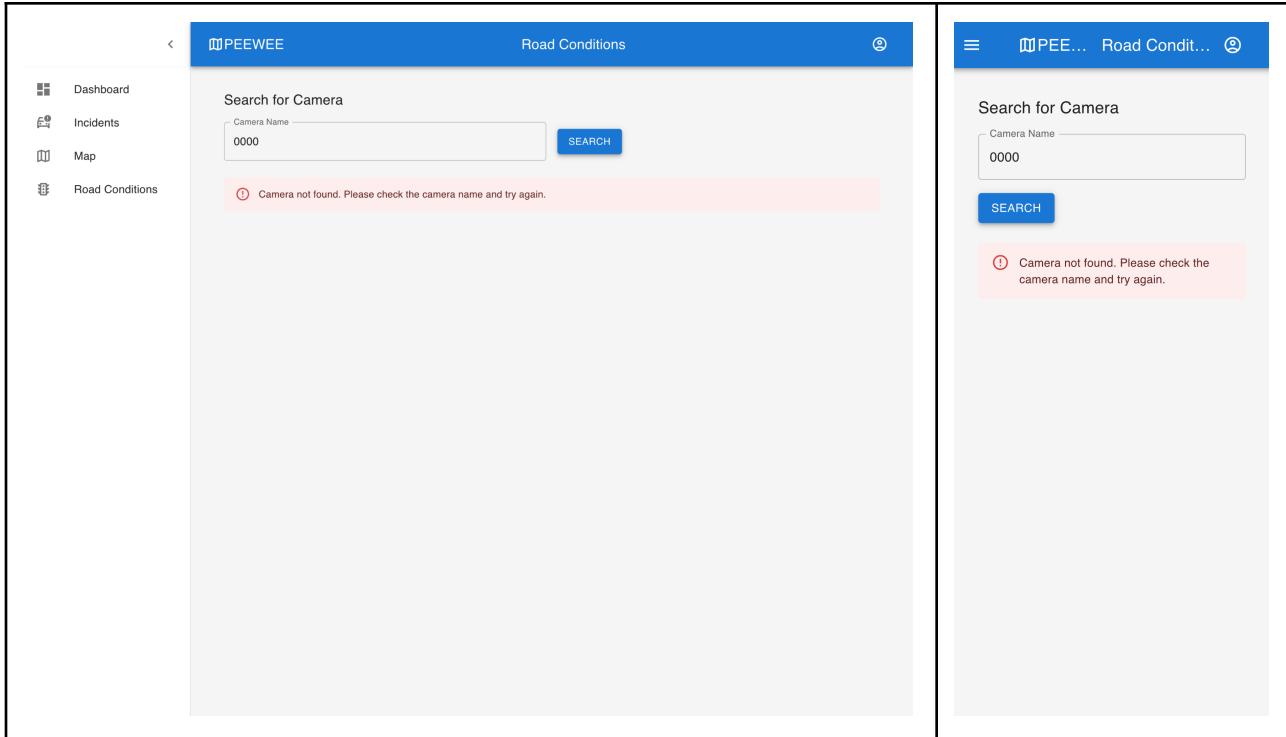


Figure 3: The Alert component displaying errors

## User Interface Screenshots and Demonstration

Screenshots of Peewee's User Interface will be detailed in the appendix under the title: [Appendix C: User Interface Screenshots](#). A video demonstrating our application can be accessed via this link: <https://www.youtube.com/watch?v=777DNkQB9Pc>.

### 3.2 Hardware Interfaces

Device types: PEEWEE is a Web-based application, can be accessed through multiple devices, across different browsers.

Hardware components: The browser will request the user's current location data via their device's GPS.

### 3.3 Software Interfaces

#### Server

##### **MongoDB - Version 7.0 or higher**

MongoDB is a NoSQL document database used to store and manage user, routes and traffic data. The app will communicate with the MongoDB database using the Mongoose ODM library.

##### **Node.js - Version 12 or higher**

Node.js is an open-source, cross-platform JavaScript runtime environment that allows developers to build scalable, high-performance web applications. The app will use Node.js as the server-side runtime environment.

##### **Express.js - Version 4.18.2 or higher**

Express.js is a fast, minimalist web framework for Node.js that provides a set of robust features for web and mobile applications. The app will use Express.js to build the RESTful API endpoints for data communication between the server and client.

##### **brcrypt - Version 5.1.1 or higher**

brcrypt is a library used to perform salted hashing on passwords so that they can be stored in our database securely.

##### **JSON Web Token - Version 9.0.2 or higher**

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

#### App

##### **React.js - Version 18.2 or higher**

React.js is a javascript library used to build the user interface as well as the logic for the app.

##### **Axios - Version 1.5.1 or higher**

Axios is a library that handles HTTPS requests to the backend API for data communication between the server and client.

#### **Data items entering and exiting the system**

1. User authentication and authorization data
2. Traffic Information in the forms of traffic images, current vehicle count, past vehicle counts and peakedness of traffic
3. Details of User Routes in terms of coordinates and name of location for both source and destination

4. Details of Incident Reports such as location in the form of coordinates and name as well as type of incident

Data that will be shared across software components includes user authentication data, traffic data, user routes data and reports. The data sharing mechanism will be implemented using MongoDB as the database management system and REST APIs for data communication.

## Services Needed

### LTA Traffic Images API

This API is used to generate the links to the latest images from traffic cameras all around Singapore. LTA's Traffic Images are updated every 20 seconds for all 90 traffic cameras islandwide. The PEEWEE app uses this API in its core function, along with our own traffic AI, to generate traffic trends.

### PEEWEE Traffic AI

Our Traffic AI is trained on a large set of traffic images built on top of the YoloV8 model. The AI would return the number of cars detected in a given traffic image. The model has also been exported as an .onnx file to be used without the need of external resource-intensive modules or libraries. Please refer to the Other Requirements section for further elaboration on the model.

### Google Maps API

The Google Maps API is used to display the map, along with all the map functions such as plotting of routes, traffic cameras and traffic heatmap.

### Brevo API

Brevo is used to send two-factor authentication emails to the user to be used for registration.

### LocationIQ API

LocationIQ is used to convert coordinates of users' current location to an address for display.

## 3.4 Communications Interfaces

The communication between the front end app and the back end server would be done with HTTPS protocols via RESTful API calls.

The user password undergoes hashing via the bcrypt library so that it can be securely stored in our MongoDB database as a hash.

## 4. System Features

### 4.1 Registration

#### 4.1.1 Description

The registration page allows a user to register for an account with a valid email and password.

#### 4.1.2 Stimulus/Response Sequences

The user will be redirected to the registration page after selecting to register for an account from the landing page.

#### 4.1.3 Related Use Case

UC-02

#### 4.1.4 Functional Requirements

1. System must allow the user to create an account
  - 1.1. User must enter an email address and password to sign up for an account
  - 1.2. System must verify that the email address exists and is not registered under other users
  - 1.3. System must implement restrictions on the password complexity
    - 1.3.1. Passwords must have at least 8 characters, with at least 1 uppercase, 1 lowercase and 1 special character

### 4.2 Sign In

#### 4.2.1 Description

The Sign In page allows a user to log into his or her account by entering a valid email and password.

#### 4.2.2 Stimulus/Response Sequences

The user will be redirected to the sign in page after selecting to sign in to his or her account on the landing page.

#### 4.2.3 Related Use Cases

UC-01 and UC-03

#### 4.2.4 Functional Requirements

1. System must allow the user to log in to an existing account
  - 1.1. User must enter a valid email address and password
  - 1.2. System must verify that the email address and password entered are valid
2. System must allow the user to reset the password if the user forgets the password.
  - 2.1. System must send an 6 digit OTP to the user via the registered email address to reset their password
  - 2.2. User must enter a valid OTP to reset password

- 2.3. User must enter a new password that complies with the restrictions on the password complexity outlined in 4.1.4.1.3.1

## 4.3 View Traffic Conditions

### 4.3.1 Description

The Traffic Conditions page allows the user to search for information on the traffic conditions at specific locations.

### 4.3.2 Stimulus/Response Sequences

The user will be redirected to the traffic condition page after selecting to view traffic conditions on the navigation bar.

### 4.3.3 Related Use Cases

[UC-10](#) and [UC-11](#)

### 4.3.4 Functional Requirements

1. System must allow the user to view all real-time traffic images retrievable from the Traffic Image API
2. System must allow the user to search and view real-time traffic images from specific traffic camera locations
  - 2.1. User must enter a valid location to search for a traffic image
3. System must allow the user to view hourly and daily trends of the number of vehicles across all traffic camera locations and at specific traffic camera locations
4. System must analyse and update traffic conditions every 5 minutes.

## 4.4 View Traffic Incidents

### 4.4.1 Description

The Traffic Incidents page allows the user to view traffic incidents reported by other users.

### 4.4.2 Stimulus/Response Sequences

The user will be redirected to the traffic incident page after selecting to view traffic incidents on the navigation bar.

### 4.4.3 Related Use Case

[UC-09](#)

### 4.4.4 Functional Requirement

1. System must allow the user to view information of all the incidents reported in the past 24 hours

## 4.5 Report Traffic Incidents

### 4.5.1 Description

The Traffic Incidents page allows the user to view and report traffic incidents.

### 4.5.2 Stimulus/Response Sequences

The user will be redirected to the report incident page after selecting to report incident on the traffic incident page.

### 4.5.3 Related Use Case

#### UC-08

### 4.5.4 Functional Requirements

1. System must allow the user to report incidents that occur on driving roads
  - 1.1. User must enter the incident type and incident description to report an incident
  - 1.2. Incident type is restricted to accidents, roadworks and closures
  - 1.3. System must request to access and use the user's current location as the incident location

## 4.6 View Map

### 4.6.1 Description

The Map page allows the user to view traffic incidents, view traffic conditions, search driving routes on a map.

### 4.6.2 Stimulus/Response Sequences

The user will be redirected to the map page after selecting to view the map on the navigation bar.

### 4.6.3 Related Use Cases

#### UC-04, UC-05 and UC-06

### 4.6.4 Functional Requirements

1. System must allow the user to view real-time traffic congestion levels along driving roads, traffic camera locations and reported incident locations on google map
  - 1.1. System must assess real-time traffic congestion level by analysing real-time traffic images from the Traffic Image API
    - 1.1.1. The congestion level must be updated every 5 minutes.
  - 1.2. System must allow the user to filter the map based on traffic camera types and incident types
2. System must allow the user to create driving routes

2.1. User must enter a starting point and a destination to create driving routes

## 4.7 Save and View Routes

### 4.7.1 Description

The Map and Favourite Routes page allows the user to save and view their driving routes on the map.

### 4.7.2 Stimulus/Response Sequences

The user can save their routes in the map page and will be redirected to the favourite routes page after selecting to view their favourite route list on the map page.

### 4.7.3 Related Use Case

#### UC-07

### 4.7.4 Functional Requirements

1. System must allow the user to save their driving routes
2. System must allow the user to view the number of traffic cameras and real-time congestion levels along a selected driving route on the map

## 5. Nonfunctional Requirements

### 5.1 Performance Requirements

#### a. Real-Time Monitoring

- i. The system must load pages in under 5 seconds.
- ii. The system must detect vehicles in traffic images for congestion analysis with an accuracy of above 70%.

#### b. Scalability

- i. The system needs to be able to support at least 100 simultaneous users without decrease in performance.

#### c. System Availability

- i. The system must be compatible with latest versions of web browsers such as Chrome, Firefox, Safari, and Edge.
- ii. The system should have an uptime of at least 99.9%.
- iii. The system must remain responsive even when external APIs are unresponsive or unavailable.

### 5.2 Safety Requirements

#### a. Incident Alerts

- i. The system should have the capability to generate alerts for users in the case of unexpected situations, such as accidents or road closures, to enhance user safety and awareness.

### 5.3 Security Requirements

#### a. Encryption

- i. Passwords of users must be encrypted using industry-standard encryption protocols and stored in a secure storage.

#### b. User Authentication

- i. Users must undergo a secure authentication process before accessing the System.

## **5.4 Software Quality Attributes**

### **a. Usability**

- i. The user interface should use clear labels and prompts, ensuring that users can easily navigate the system and interpret traffic information.

### **b. Maintainability**

- i. The system's codebase should be well-documented to facilitate readability.
- ii. The system should be modular, facilitating easy maintenance and updates.

### **c. Reliability**

- i. The system should operate consistently without unexpected failures. Automated tests should be in place to detect and resolve potential issues proactively.

## **5.5 Business Rules**

### **a. Data Retention Policy**

- i. The system should adhere to a data retention policy, storing historical traffic data for a minimum of six months to support long-term analysis.

## 6. Use Cases

### 6.1 Use Case Model

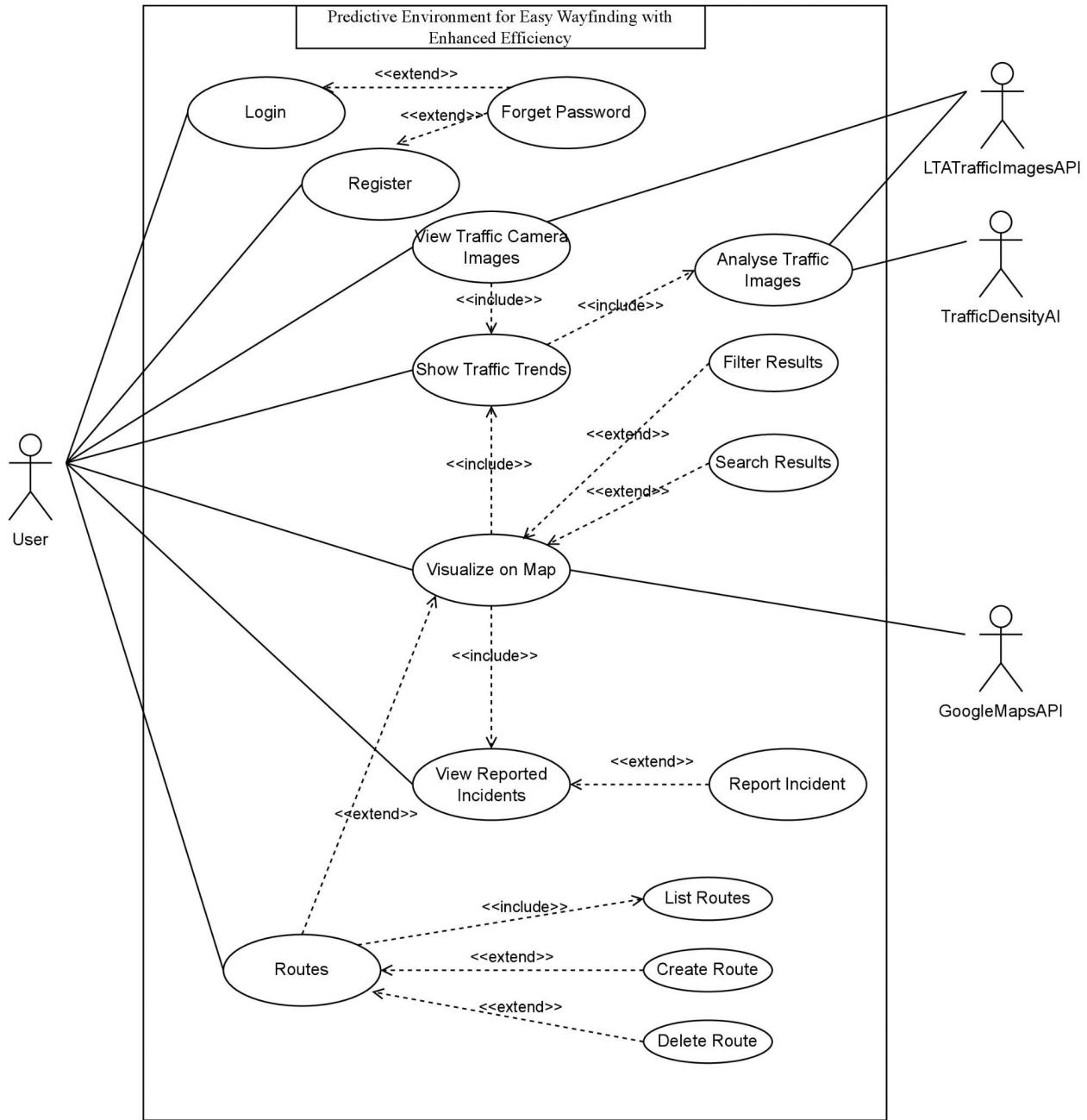


Figure 4: Use Case Diagram for PEEWEE

## 6.2 Use Case Notations

### 5.2.1. Alternative Flow Origin

We will indicate the exact step of the Normal Flow from which the Alternative Flow originated from, using the following notation: AF-Sx, where **AF** denotes Alternative Flow, and **Sx** denotes that it came from **Step X** of the Normal Flow. For instance, **AF-S1** will denote that the Alternative Flow originated from **Step 1** of the Normal Flow.

### 5.2.2 Use Case ID

We will label each use case using the following notation: UC-xx, where UC denotes Use Case and xx denotes the use case number

## 6.3 Use Case Specifications

### UC-01: Login

- a. [Class Diagram](#)
- b. [Sequence Diagram](#)
- c. Specification:

Use Case ID:	UC-01		
Use Case Name:	Login		
Created By:	Guang	Last Updated By:	Zi Qin
Date Created:	05/09/2023	Date Last Updated:	12/11/2023

Actor:	User
Description:	Authenticate the User
Preconditions:	1. The User must have an account registered in the system
Postconditions:	The User will be authenticated
Priority:	High

Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. User provides authentication information such as email address and password</li> <li>2. User clicks on login</li> <li>3. The system authenticates User</li> <li>4. User is brought to the dashboard</li> </ol>
Alternative Flows:	<p>AF-S1: User clicks on forgot password</p> <ol style="list-style-type: none"> <li>1. The system redirects User to UC-03 Forgot Password, and User changes to a new password</li> <li>2. The system brings User back to current flow</li> <li>3. Flow resets to Step 1 of normal flow</li> </ol> <p>AF-S3: Credentials are incorrect</p> <ol style="list-style-type: none"> <li>1. The system displays an error message indicating an incorrect credentials</li> <li>2. Flow resets to Step 1 of normal flow</li> </ol> <p>AF-S3: The system fails to authenticate User</p> <ol style="list-style-type: none"> <li>1. The system displays an error message indicating the reason of failure</li> <li>2. Flow resets to Step 1 of normal flow</li> </ol>
Exceptions:	<p>EX1: Error connecting to server</p> <ol style="list-style-type: none"> <li>1. System displays an error message indicating connection is lost</li> <li>2. Use case ends</li> </ol>
Includes:	(Extends) UC-03 Forgot Password
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

**UC-02: Register**

- a. [Class Diagram](#)
- b. [Sequence Diagram](#)
- c. Specification

Use Case ID:	UC-02		
Use Case Name:	Register		
Created By:	Guang	Last Updated By:	Cheng Yao
Date Created:	05/09/2023	Date Last Updated:	25/10//2023

Actor:	User
Description:	Register a User account in the system
Preconditions:	1. The User must not have an account already registered in the system with the same credentials
Postconditions:	A new User account will be created in the system
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. User provides a valid email address</li> <li>2. User provides a password and confirms the password in the confirm password field</li> <li>3. The system validates the provided information and registers a new account for User</li> <li>4. User is redirected to the Dashboard</li> </ol>
Alternative Flows:	<p>AF-S1: User provides invalid email address</p> <ol style="list-style-type: none"> <li>1. The system displays an error alert</li> <li>2. Flow resets to Step 1 of normal flow</li> </ol>

	<p>AF-S2: Password and confirm password fields do not match</p> <ol style="list-style-type: none"> <li>1. The system displays an error and prompts User to re-enter password</li> <li>2. Flow resets to Step 1</li> </ol> <p>AF-S2: Entered password does not conform to password policy</p> <ol style="list-style-type: none"> <li>1. The system will prompt the User to re-enter a new password, specifying which condition policy has not been satisfied</li> </ol> <p>Flow resets to Step 2</p> <p>AF-S3: User provides an email address that is already registered</p> <ol style="list-style-type: none"> <li>1. The system will alert User</li> <li>2. The system will prompt User to register with another email address</li> <li>3. Flow resets to Step 1</li> </ol>
Exceptions:	<p>EX1: System is offline and is unable to register a new account</p> <ol style="list-style-type: none"> <li>1. The system displays an error message with an error code</li> <li>2. Use case ends</li> </ol>
Includes:	(Extends) UC-03 Forgot Password
Special Requirements:	<p>For security, passwords must conform to a specific password policy:</p> <ol style="list-style-type: none"> <li>1. Minimum length of 8 characters</li> <li>2. At least 1 uppercase character</li> <li>3. At least 1 lowercase character</li> <li>4. At least 1 special character</li> </ol>
Assumptions:	-
Notes and Issues:	-

**UC-03: Forgot Password**

- a. [Class Diagram](#)
- b. [Sequence Diagram](#)
- c. Specification:

Use Case ID:	UC-03		
Use Case Name:	Forgot Password		
Created By:	Guang	Last Updated By:	Cheng Yao
Date Created:	05/09/2023	Date Last Updated:	25/10/2023

Actor:	User
Description:	Reset the password of the User
Preconditions:	1. The User must have an account registered in the system
Postconditions:	The User account's password will be changed to a new password
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. User provides their email address</li> <li>2. The system sends an OTP to the User's email</li> <li>3. User enters the OTP into the page</li> <li>4. The system prompts the User for a new password with a password and confirm password field</li> <li>5. User clicks on change password</li> <li>6. The system successfully changes User's password and redirects to the Login use case</li> </ol>
Alternative Flows:	AF-S1: User provides an invalid email address <ol style="list-style-type: none"> <li>1. The system shows an error message and prompts User for a</li> </ol>

	<p>valid email address</p> <p>2. Flow returns to Step 1 of normal flow</p> <p><b>AF-S3: OTP is incorrect or expired</b></p> <ol style="list-style-type: none"> <li>1. The system will prevent the User from changing their password.</li> <li>2. User can request for a new OTP after a cooldown of 60 seconds, up to 3 times total</li> <li>3. Flow returns to Step 3 of normal flow</li> </ol> <p><b>AF-S4: Password and confirm password fields do not match</b></p> <ol style="list-style-type: none"> <li>1. The system displays an error and prompts User to re-enter password</li> <li>2. User enters the password correctly</li> <li>3. Flow moves to Step 5 of normal flow</li> </ol> <p><b>AF-S4: Entered password does not conform to password policy</b></p> <ol style="list-style-type: none"> <li>1. The system will prompt the User to re-enter a new password, specifying which condition policy has not been satisfied</li> <li>2. Flow returns to Step 4 of normal flow</li> </ol>
Exceptions:	<p>EX1: OTP entered incorrectly for 3 times</p> <ol style="list-style-type: none"> <li>1. The system displays an error message</li> <li>2. Use case ends</li> </ol>
Includes:	-
Special Requirements:	<p>For security, passwords must conform to a specific requirement:</p> <ol style="list-style-type: none"> <li>1. Minimum length of 8 characters</li> <li>2. At least 1 uppercase character</li> <li>3. At least 1 lowercase character</li> <li>4. At least 1 special character</li> </ol> <p>OTPs will expire after 5 minutes</p>
Assumptions:	-
Notes and Issues:	-

**UC-04: Visualise on Map**

- a. [Class Diagram](#)
- b. [Sequence Diagram](#)
- c. Specification:

Use Case ID:	UC-04		
Use Case Name:	Visualise on Map		
Created By:	Aaron	Last Updated By:	Zi Qin
Date Created:	05/09/2023	Date Last Updated:	12/11/2023

Actor:	User
Description:	<ul style="list-style-type: none"> <li>• Facilities Users to visually interact with a dynamic map, spotting real-time traffic conditions, traffic cameras, and road incidents.</li> <li>• The map offers an intuitive overlay, giving users a holistic view of the traffic ecosystem, ensuring a seamless navigation experience.</li> </ul>
Preconditions:	<ol style="list-style-type: none"> <li>1. User is authenticated and has access rights.</li> <li>2. Stable and consistent Internet connection.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. Users get a graphical representation of the traffic landscape with key highlights.</li> <li>2. Relevant traffic markers, including cameras and incidents, are visible to users.</li> </ol>
Priority:	Essential
Frequency of Use:	Regular (before and during commuting)
Flow of Events:	<ol style="list-style-type: none"> <li>1. User selects the “Visualise on Map” feature.</li> <li>2. System does step 1 to 3 in UC-09 normal flow to gather incident reports.</li> </ol>

	<ol style="list-style-type: none"> <li>3. System does steps 1 to 3 in UC-11 normal flow to gather the general traffic trends.</li> <li>4. Application loads a real-time map of the user's current location or specified area.           <ul style="list-style-type: none"> <li>• Traffic conditions are colour-coded: green for light, orange for medium, and red for heavy.</li> <li>• Markers for traffic cameras and incidents are plotted on the map.</li> </ul> </li> <li>5. User can zoom, pan, and click on individual markers for more detailed information.</li> <li>6. System plots all incidents when selected.</li> </ol>
Alternative Flows:	<p>AF-S2: If the map fails to load due to weak connection or API issue:</p> <ol style="list-style-type: none"> <li>1. The application advises the user: "Unable to load the map. Please check your connection or try again later".</li> <li>2. Return to Step 2 in normal flow.</li> </ol> <p>AF-S2: If the user seeks a region with restricted data:</p> <ol style="list-style-type: none"> <li>1. The map displays, but with limited or no data, and notifies: "Information for this area is limited or restricted."</li> <li>2. Return to Step 2 in normal flow.</li> </ol>
Exceptions:	-
Includes:	<p>UC-11 View Traffic Trends        UC-09 View Reported Incidents</p>
Special Requirements:	<ul style="list-style-type: none"> <li>• Data must be as real-time as possible with refresh intervals not exceeding a few minutes.</li> <li>• The application must handle high-resolution camera feeds without lag.</li> </ul>
Assumptions:	<ul style="list-style-type: none"> <li>• The database is constantly updated with fresh traffic, camera, and incident data.</li> <li>• Users have basic knowledge of map interfaces and can understand common traffic symbols.</li> </ul>
Notes and Issues:	<ul style="list-style-type: none"> <li>• Ensure consistent uptime for the service to be useful during critical times.</li> <li>• Periodically review and update the map interface to include new roads, landmarks, and traffic management tools.</li> </ul>

**UC-05: Search Results**

- a. [Class Diagram](#)
- b. [Sequence Diagram](#)
- c. Specification:

Use Case ID:	UC-05		
Use Case Name:	Search Results		
Created By:	Aaron	Last Updated By:	Guang
Date Created:	05/09/2023	Date Last Updated:	25/10/2023

Actor:	User
Description:	<p>Allows users to search traffic conditions.</p> <ul style="list-style-type: none"> <li>• The feature allows users to search for traffic cameras and traffic incidents within the website.</li> <li>• Data aims to give users a comprehensive understanding of the traffic conditions, easing their efforts in making informed decisions about their desired routes.</li> </ul>
Preconditions:	<ul style="list-style-type: none"> <li>• User must be registered and logged into the system.</li> <li>• Internet connectivity is available.</li> </ul>
Postconditions:	<ul style="list-style-type: none"> <li>• User is presented with accurate traffic conditions and related statistics for the chosen location or route.</li> <li>• User's search queries are saved in their search history for future reference.</li> </ul>
Priority:	High
Frequency of Use:	High (High usage during peak commuting hours.)
Flow of Events:	<p><b>Traffic Search:</b></p> <ol style="list-style-type: none"> <li>1. User accessing "Traffic Search" feature.</li> </ol>

	<ol style="list-style-type: none"> <li>2. System prompts the user to put a specific road, landmark, or search for a traffic camera or reported incident.</li> <li>3. User input the desired location or criteria.</li> <li>4. System fetches data and provides information on traffic camera or reported incidents</li> </ol>
Alternative Flows:	<p>AF-S2: If the user's input does not match any existing roads or landmarks:</p> <ol style="list-style-type: none"> <li>1. The system suggests similar or nearby locations.</li> <li>2. User can select a suggested location or re-enter their search.</li> <li>3. User Return to Step 2 in Normal Flow.</li> </ol> <p>AF-S2: If the system cannot retrieve live data to any external issues:</p> <ol style="list-style-type: none"> <li>1. The user is notified of the temporary unavailability of data.</li> <li>2. The system may provide the most recent cached data with a timestamp, if available.</li> <li>3. User Return to Step 2 in Normal Flow.</li> </ol>
Exceptions:	<p>EX1: If the system faces an unexpected internal error:</p> <ol style="list-style-type: none"> <li>1. The user is alerted: "An unexpected error occurred. Please try again later."</li> </ol>
Includes:	-
Special Requirements:	<ul style="list-style-type: none"> <li>● System ensures data accuracy and updates traffic conditions in real-time.</li> <li>● Ensure privacy and security of user search history.</li> </ul>
Assumptions:	<ul style="list-style-type: none"> <li>● Traffic data provided is sourced from reliable channels or databases.</li> <li>● Registered users understand basic traffic terms and classifications.</li> </ul>
Notes and Issues:	<ul style="list-style-type: none"> <li>● Regularly update the system's database to include new roads or landmarks.</li> <li>● Monitor for any discrepancies or inconsistencies in traffic data sourced from multiple channels.</li> </ul>

**UC-06: Filter Results**

- a. [Class Diagram](#)
- b. [Sequence Diagram](#)
- c. Specification:

Use Case ID:	UC-06		
Use Case Name:	Filter Results		
Created By:	Aaron	Last Updated By:	Zi Qin
Date Created:	05/09/2023	Date Last Updated:	12/11/2023

Actor:	User
Description:	<p>Allows users to filter map information.</p> <ul style="list-style-type: none"> <li>• By eliminating unwanted incident types and traffic cameras, users can make informed decisions within the system more effectively.</li> </ul>
Preconditions:	<ul style="list-style-type: none"> <li>• User must be registered and logged into the system.</li> <li>• Internet connectivity is available.</li> </ul>
Postconditions:	<p>Traffic conditions on the map are updated to reflect the user's selected filters.</p> <p>Users can save filter preferences for future use.</p>
Priority:	High
Frequency of Use:	High(High usage during peak commuting hours.)
Flow of Events:	<ol style="list-style-type: none"> <li>1. Accessing Filter Options: <ul style="list-style-type: none"> <li>- Once the search results are displayed on the map, a filter icon or button appears on the top or side of the screen.</li> <li>- The user can click on this icon to access filter options.</li> </ul> </li> <li>2. Display of Filter Options:</li> </ol>

	<ul style="list-style-type: none"> <li>- A dropdown or slide-out panel appears displaying various filter criteria.</li> <li>- Filters are primarily categorised as "Traffic Cameras" and "Road Incidents."</li> <li>- Under "Road Incidents," options like "Accidents," "Roadwork," and "Closures" are presented.</li> </ul> <p>3. Selecting Filters:</p> <ul style="list-style-type: none"> <li>- Users can choose to select multiple filters or deselect them based on their preferences. For example, they might want to see only "Accidents" and "RoadWorks" but not other incidents.</li> </ul> <p>4. Applying Filters to Map View:</p> <ul style="list-style-type: none"> <li>- Map updates to display the traffic cameras and incidents that match the user's selected filters.</li> <li>- All irrelevant data points are removed from the map.</li> </ul> <p>5. Close the Filter Panel:</p> <ul style="list-style-type: none"> <li>- Once user set preferences, they can click outside the filter panel or click on the “close or X” icon to return to the map view with their filter settings applied.</li> </ul>
Alternative Flows:	<p>AF-S1: If system faces a delay in updating the map view:</p> <ul style="list-style-type: none"> <li>- User is shown a loading indicator with a message “Updating your view, please wait”</li> <li>- System returns to Step 1 in the normal flow.</li> </ul>
Exceptions:	-
Includes:	-
Special Requirements:	<ul style="list-style-type: none"> <li>● System must have an updated traffic database and camera locations.</li> <li>● System should be responsive and adaptive to different screen sizes and electronics gadgets.</li> </ul>
Assumptions:	<ul style="list-style-type: none"> <li>● Traffic data is frequently updated in the background for the most accurate representation.</li> </ul>
Notes and Issues:	-

**UC-07: Routes**

- a. [Class Diagram](#)
- b. [Sequence Diagram](#)
- c. Specification:

Use Case ID:	UC-07		
Use Case Name:	Routes		
Created By:	Cheng Yao	Last Updated By:	Zi Qin
Date Created:	05/09/2023	Date Last Updated:	12/11/23

Actor:	User
Description:	Allows users to check traffic conditions.
Preconditions:	<ul style="list-style-type: none"> <li>User is logged in</li> <li>User clicks on “Routes” under Map</li> </ul>
Postconditions:	<ol style="list-style-type: none"> <li>1. User exits the app OR</li> <li>2. User moves to another screen/page</li> </ol>
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. User clicks on routes under Map screen</li> <li>2. System displays the user’s favourite routes.</li> <li>3. System prompts the user to select a route from their favorite route list or search for a route using Origin and Destination</li> <li>4. User selects a route from their favorite list</li> <li>5. System will retrieve route information from Maps API through the Map feature. System will display the traffic conditions along the route.</li> </ol>

	<ol style="list-style-type: none"> <li>6. If the route has not been favorited, user can click on the unlit favourite icon to save the route</li> <li>7. If user favourites the route, the system will store it into the user's favourites list</li> <li>8. Users can search for another route or click on previously searched routes</li> </ol>
Alternative Flows:	<p>AF-S6: If the user's route is already favorited</p> <ol style="list-style-type: none"> <li>1. System will add the route to the favourite routes list</li> <li>2. Clicking on the unfavourite button will unfavourite the route, removing it from their favourites list</li> <li>3. The System returns to Step 5 in Normal Flow</li> </ol> <p>AF-S3: If the user searches for another route</p> <ol style="list-style-type: none"> <li>1. User enters the origin and destination in the map</li> <li>2. The system verifies the validity of the origin and destination</li> <li>3. If the origin and destination are valid, the system returns the user to Step 5 in the normal flow</li> </ol>
Exceptions:	<p>EX1: If there temporary issue or outage with the Maps API:</p> <ol style="list-style-type: none"> <li>1. System notifies the user: "There seems to be a temporary issue retrieving data. Please try again in a moment."</li> <li>2. Use case ends and the user is returned to UC-04 Visualise on Map</li> </ol>
Includes:	(Extends) UC-04 Visualise on Map
Special Requirements:	Requires real-time syncing with the Maps API
Assumptions:	Traffic conditions will change frequently, requiring continuous updates. Users could input multiple favourite routes.
Notes and Issues:	<p>Possibility in minor delays upon fetching real-time data during peak server loads.</p> <p>Camera data only covers specific locations in a fixed angle, might not be available for all locations.</p>

**UC-08: Report Incidents**

- a. [Class Diagram](#)
- b. [Sequence Diagram](#)
- c. Specification:

Use Case ID:	UC-08		
Use Case Name:	Report Incidents		
Created By:	Zi Qin	Last Updated By:	Zi Qin
Date Created:	05/09/23	Date Last Updated:	12/11/23

Actor:	User
Description:	This use case allows the user to alert other users of incidents that occur on the driving roads.
Preconditions:	User has logged into the system and is viewing the Traffic Incident page
Postconditions:	User is returned to the Traffic Incident page
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user selects “Report Incident” in the Traffic Incident page.</li> <li>2. The system displays the report form and location access message.</li> <li>3. The user selects the incident type and enters the incident description.</li> <li>4. The user accepts the system's request to access the user's current location.</li> <li>5. The system detects and displays the current location detected.</li> <li>6. The system validates user input and displays a “Submit” button.</li> </ol>

	<ol style="list-style-type: none"> <li>7. The user clicks on the “Submit” button.</li> <li>8. The system saves a record of the time of report submission, incident type, incident location and incident description.</li> <li>9. The system displays a message on the submission status and the option to submit another form.</li> <li>10. The user selects to return to the incident page on the navigation bar.</li> </ol>
Alternative Flows:	<p>AF-S4: If the user rejects system’s request to access current location</p> <ol style="list-style-type: none"> <li>1. The system goes back to Step 4 of normal flow</li> </ol> <p>AF-S5: If the system fails to detect the user’s location</p> <ol style="list-style-type: none"> <li>1. The system displays a message “Failed to detect location. Redetect current location?”</li> <li>2. The user selects to redetect the current location.</li> <li>3. The system goes back to Step 5 of normal flow.</li> </ol> <p>AF-S9: If the system fails to save the incident details</p> <ol style="list-style-type: none"> <li>1. The system displays an error message and the option to resubmit form</li> <li>2. The user selects to resubmit form</li> <li>3. The system goes back to Step 8 of normal flow</li> </ol>
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

**UC-09: View Reported Incidents**

- a. [Class Diagram](#)
- b. [Sequence Diagram](#)
- c. Specification:

Use Case ID:	UC-09		
Use Case Name:	View Reported Incidents		
Created By:	Zi Qin	Last Updated By:	Zi Qin
Date Created:	05/09/23	Date Last Updated:	21/10/23

Actor:	User
Description:	This use case informs users about the incidents reported by other users.
Preconditions:	User has logged into the system
Postconditions:	User is viewing the Traffic Incident page
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user selects “Incidents” in the main menu of the user interface.</li> <li>2. The system retrieves records of all incidents reported on the day.</li> <li>3. The system displays all the information on the reported incidents:           <ul style="list-style-type: none"> <li>• Incident Type</li> <li>• Reported Time</li> <li>• Incident Location</li> <li>• Incident Description</li> </ul> </li> </ol>
Alternative Flows:	<p>AF-S2: If there are no incidents reported on the day</p> <ol style="list-style-type: none"> <li>1. The system displays the message “There are no incidents reported today”.</li> </ol>

	<ol style="list-style-type: none"><li>2. The user refreshes the user interface.</li><li>3. The system goes back to Step 2 of normal flow</li></ol> <p>AF-S2: If the system fails to retrieve incident records</p> <ol style="list-style-type: none"><li>1. The system displays a message “Error in loading. Please refresh the page again.”</li><li>2. The user refreshes the user interface.</li><li>3. The system goes back to Step 2 of normal flow.</li></ol>
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

### UC-10: View Traffic Camera Images

- a. [Class Diagram](#)
- b. [Sequence Diagram](#)
- c. Specification

Use Case ID:	UC-10		
Use Case Name:	View Traffic Camera Images		
Created By:	Hamka	Last Updated By:	Eugenia
Date Created:	05/09/2023	Date Last Updated:	25/10/2023

Actor:	User
Description:	This use case allows users to look at the live image from a specific camera
Preconditions:	The user has selected a specific camera location
Postconditions:	System allows users to view traffic photos of the specific camera location.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. User clicks on View Traffic Camera Images in the specific camera page</li> <li>2. System displays the following information of the camera <ul style="list-style-type: none"> <li>o Location</li> <li>o Traffic Image</li> <li>o Camera ID</li> <li>o Timestamp</li> <li>o Specific Reported Incidents</li> <li>o Traffic Peakness Level</li> </ul> </li> </ol>

	i. Low ii. Medium iii. High
Alternative Flows:	-
Exceptions:	-
Includes:	UC-11 View Traffic Trends
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

**UC-11: Show Traffic Trends**

- a. [Class Diagram](#)
- b. [Sequence Diagram](#)
- c. Specification:

Use Case ID:	UC-11		
Use Case Name:	Show Traffic Trends		
Created By:	Hamka	Last Updated By:	Eugenia
Date Created:	05/09/2023	Date Last Updated:	25/10/2023

Actor:	User
Description:	This use case allows users to view overall traffic trends.
Preconditions:	The user is either on the specific camera page or at the overall Road Conditions page
Postconditions:	System allows users to view traffic trends of either the specific camera or overall trends of the whole of Singapore
Priority:	High
Frequency of Use:	High
Flow of Events:	<ul style="list-style-type: none"> <li>○ The system shows a graph of the hourly traffic trends for Singapore</li> <li>○ The graph displays</li> <li>○ Count(Number of Vehicles on the rRoad)</li> <li>○ Peakness(Count compared to count at other timings)</li> </ul>
Alternative Flows:	<p>AF-S1: If the user chooses to view the traffic trend of a specific camera on the traffic camera page</p> <ol style="list-style-type: none"> <li>1. The system prompts a slide-out panel that shows a overview of the traffic trend of the specific camera</li> </ol>

	2. Flow returns to Step 2 of normal flow
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

## 7. Other Requirements

### PEEWEE Traffic AI

Our Traffic AI is an object detection and image segmentation model being run only using a CPU, to save on server resources. The AI uses the pre-existing YoloV8 Small model and trains it further on 400 manually annotated images through 100 epochs. We split our data into a train test validation split of 70%, 10% and 20% respectively to prevent overtraining on the train dataset. The model was also trained on night images as the untrained YoloV8 model was able to detect cars in clear daytime images but not any of the night time images or the smaller cars in the background.

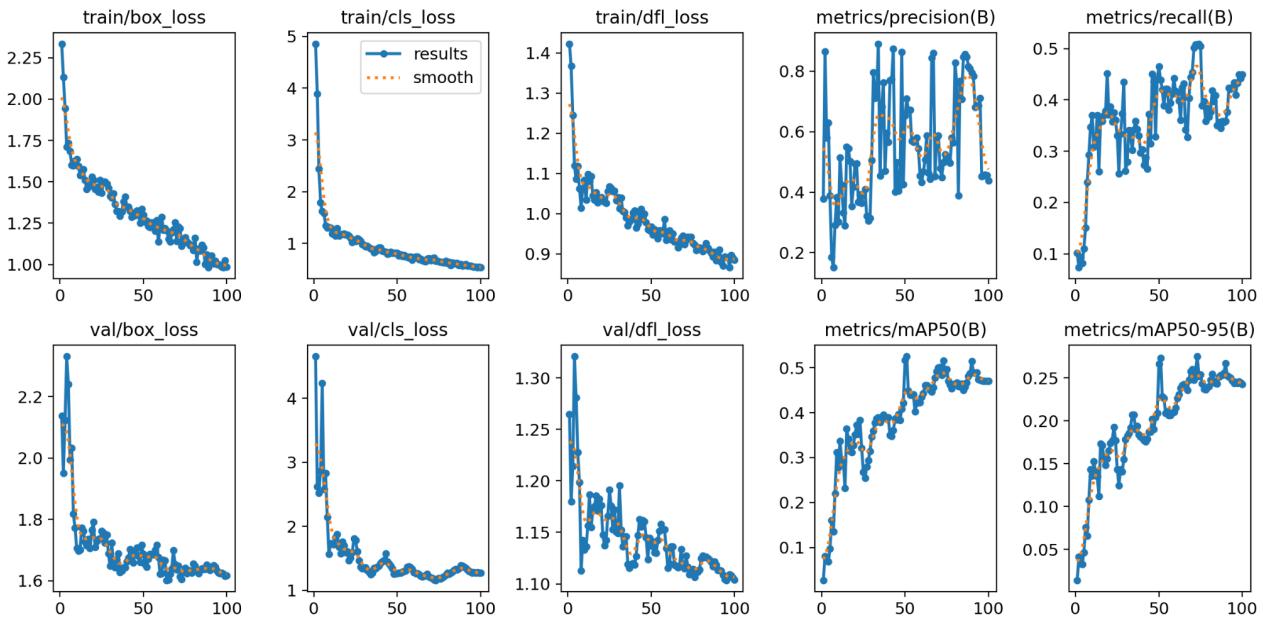
### Trained vs Untrained YoloV8 Model

In our examples below, we compare the trained model (left) to the untrained YoloV8 Model (right). Not only is it able to detect the vehicles at night and vehicles in the background, it also generally has a higher confidence in determining the object type.



**Figure 5:** Trained vs Untrained AI

### Training Metrics

**Figure 6:** Training statistics for PEEWEE AI

**Training losses:** The training losses for bounding box loss (*box\_loss*), classification loss (*cls\_loss*), and distance focal loss (*dfl\_loss*) all decrease over time, indicating that the model is learning to detect objects more accurately.

**Precision and recall:** The precision and recall metrics on the validation set also improve over time, indicating that the model is becoming better at both detecting real objects and avoiding false positives.

**mAP:** The mean average precision (*mAP*) metric on the validation set also improves over time, reaching a peak of 0.5 at epoch 100. This indicates that the model is able to detect objects accurately and consistently.

### Validation Examples

In our validation dataset of 80 images (20% of 400), our fine-tuned AI achieved an accuracy of 82% in vehicle detection.

### Further training

Given more resources such as in the use of a GPU on the server as well as more time and manpower dedicated to annotating the images, a higher accuracy and precision between object classes could be achieved.

## 8. System Testing

One important controller module, UserController, and two other modules that implement complex application logic, Map and ReportIncident, have been tested to ensure that they meet the performance requirements.

### 8.1 Black Box Testing

#### 8.1.1 Register

Test #	Description	Input	Expected Output	Actual Output	Status
1	Valid registration	<b>Email:</b> test@example.com  <b>Password:</b> correctPassword!23  <b>Retyped Password:</b> correctPassword!23	The user should be successfully registered and redirected to the Dashboard	User successfully registered and redirected to the Dashboard.	Pass
2	Invalid Email Format	<b>Email:</b> user@.com  <b>Password:</b> correctPassword!23  <b>Retyped Password:</b> correctPassword!23	The system should display an error alert	System displayed an error message "The email address format is invalid."	Pass
3	Mismatch Password and Retyped Password	<b>Email:</b> test1@example.com  <b>Password:</b> correctPassword!23  <b>Retyped Password:</b> correctPassword!234	The system should display an error alert	System displayed an error message "Password and retyped password do not match"	Pass
4	Password does not meet policy	<b>Email:</b> test2@example.com  <b>Password:</b> correctpassword!23	The system should display an error alert	System displayed an error message "Password does not meet the	Pass

		<b>Retyped Password:</b> correctpassword!23		requirements"	
5	Register with Already Existing Email	<b>Email:</b> test@example.com  <b>Password:</b> correctPassword!23  <b>Retyped Password:</b> correctPassword!23	The system should display an error alert	System displayed an error message "User already exists"	Pass

### 8.1.2 Login

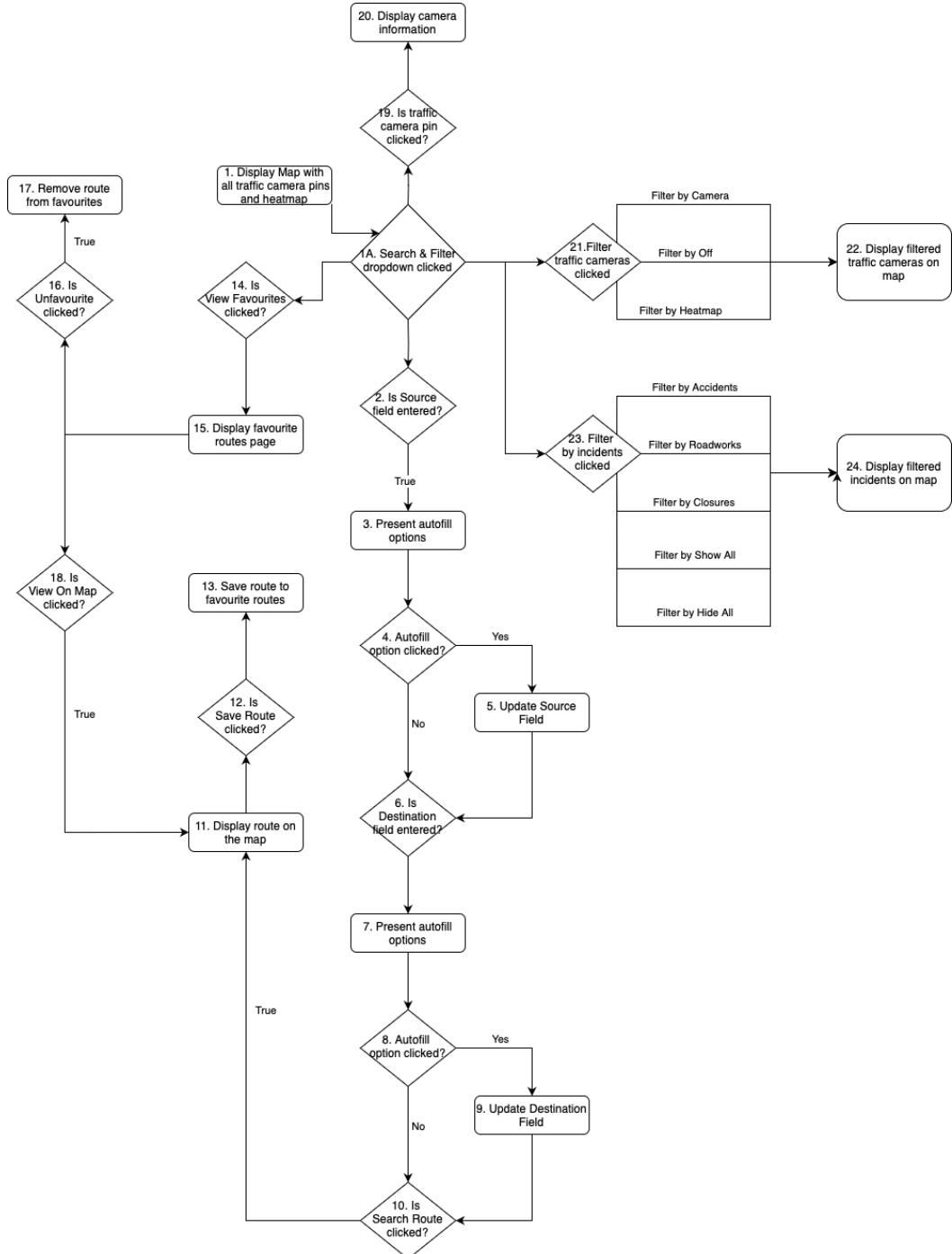
Test #	Description	Input	Expected Output	Actual Output	Status
1	Valid Login	<b>Email:</b> test@example.com  <b>Password:</b> correctPassword!23	User Logged in and redirected to dashboard.	User successfully authenticated and redirected to the dashboard.	Pass
2	Invalid Email	<b>Email:</b> test@  <b>Password:</b> correctPassword!23	System display error message about invalid email format.	System threw out an error message "login failed. Double check the email again"	Pass
3	Invalid Password	<b>Email:</b> test@example.com  <b>Password:</b> wrongPassword!23	System displays an error message about incorrect credentials.	System unable to match correct password with database and threw out an error message "Login failed. Double check password again"	Pass

4	Empty	<b>Email:</b> ``  <b>Password:</b> ``	System displays an error message indicating fields cannot be empty or a similar message.	System will not proceed with empty login credentials. Prompt the user to enter both email and password first.	Pass
5	Special Characters in Email	<b>Email:</b> us.er.test@example.com  <b>Password:</b> correctPassword123	System accepts the email with special characters and logs in.	System is able to handle characters with “.” , @, etc, which is a must, and is able to proceed on with the login.	Pass
6	Special Characters in Password	<b>Email:</b> test@example.com  <b>Password:</b> P@ssw0rd!#\$\$	System accepts the password with special characters and logs in.	System doesn't check special characters, which is ideal for additional password security	Pass
7	Email Case Insensitivity	<b>Email:</b> TEST@EXAMPLE.COM  <b>Password:</b> testPassword123	System treats email addresses as case-insensitive and logs in.	System is not case sensitive, which is ideal for email account logins.	Pass
8	Incorrect Password Case Sensitivity	<b>Email:</b> test@example.com <b>Password:</b> TestPassword123 (assuming the correct password is all lowercase)	System treats the password as case sensitive and does not log in.	System double check case sensitivity for additional security verification.	Pass

## 8.2 White Box Testing

### 8.2.1 Search Routes

Control Flow Graph



**Figure 7:** Control Flow Graph for Search Routes

Test Case 1: Valid source and destination user inputs with autofilled fields clicked and save route		
Location: "app/src/pages/Map.tsx"		
Inputs	Expected Flow and Outputs	Actual Flow and Outputs
Click Search and Filter dropdown Source: "Bed" Click on Bedok autofill Destination: "Tamp" Click on Tampines autofill Click Search Route Click Save Route	Flow: 1,1A,2,3,4,5,6,7,8,9,10,11,12,13  "Bedok" autofill presented "Tampines" autofill presented Source: "Bedok" Destination: "Tampines" Route plotted on map	Flow: 1,1A,2,3,4,5,6,7,8,9,10,11,12,13  "Bedok" autofill presented "Tampines" autofill presented Source: "Bedok" Destination: "Tampines" Route plotted on map

Test Case 2: Valid source and destination without clicking autofill		
Location: "app/src/pages/Map.tsx"		
Inputs	Expected Flow and Outputs	Actual Flow and Outputs
Click Search and Filter dropdown Source: "Bedok" Destination: "Tampines" Click Search Route Click Save Route	Flow: 1,1A,2,3,4,5,6,7,8,9,10,11,12,13  "Bedok" autofill presented "Tampines" autofill presented Source: "Bedok" Destination: "Tampines" Route plotted on map	Flow: 1,1A,2,3,4,5,6,7,8,9,10,11,12,13  "Bedok" autofill presented "Tampines" autofill presented Source: "Bedok" Destination: "Tampines" Route plotted on map

Test Case 3: Invalid source and destination without clicking autofill		
Location: "app/src/pages/Map.tsx"		
Inputs	Expected Flow and Outputs	Actual Flow and Outputs
Click Search and Filter	Flow:	Flow:

dropdown Source: "Bed" Destination: "Tamp" Click Search Route	1,1A,2,3,4,5,6,7,8,9,10,11,12,13  Source: "Bedok" Destination: "Tampines" Route not plotted on map	1,1A,2,3,4,5,6,7,8,9,10,11,12,13  Source: "Bed" Destination: "Tamp" Route not plotted on map
--	--	--

Test Case 4: View favourite routes and plot on map		
Location: "app/src/pages/Map.tsx"		
Inputs	Expected Flow and Outputs	Actual Flow and Outputs
Click Search and Filter dropdown Click View Favourites Click View on Map	Flow: 1,1A,14,15,18,11  Display Favourite Routes page Route plotted on map	Flow: 1,1A,14,15,18,11  Display Favourite Routes page Route plotted on map

Test Case 5: View favourite routes and unfavourite		
Location: "app/src/pages/Map.tsx"		
Inputs	Expected Flow and Outputs	Actual Flow and Outputs
Click Search and Filter dropdown Click View Favourites Click Unfavourite	Flow: 1,1A,14,15,16,17  Display Favourite Routes page Favourite Routes removed	Flow: 1,1A,14,15,16,17  Display Favourite Routes page Favourite Routes removed

Test Case 6: View Traffic Camera Pin		
Location: "app/src/pages/Map.tsx"		
Inputs	Expected Flow and Outputs	Actual Flow and Outputs

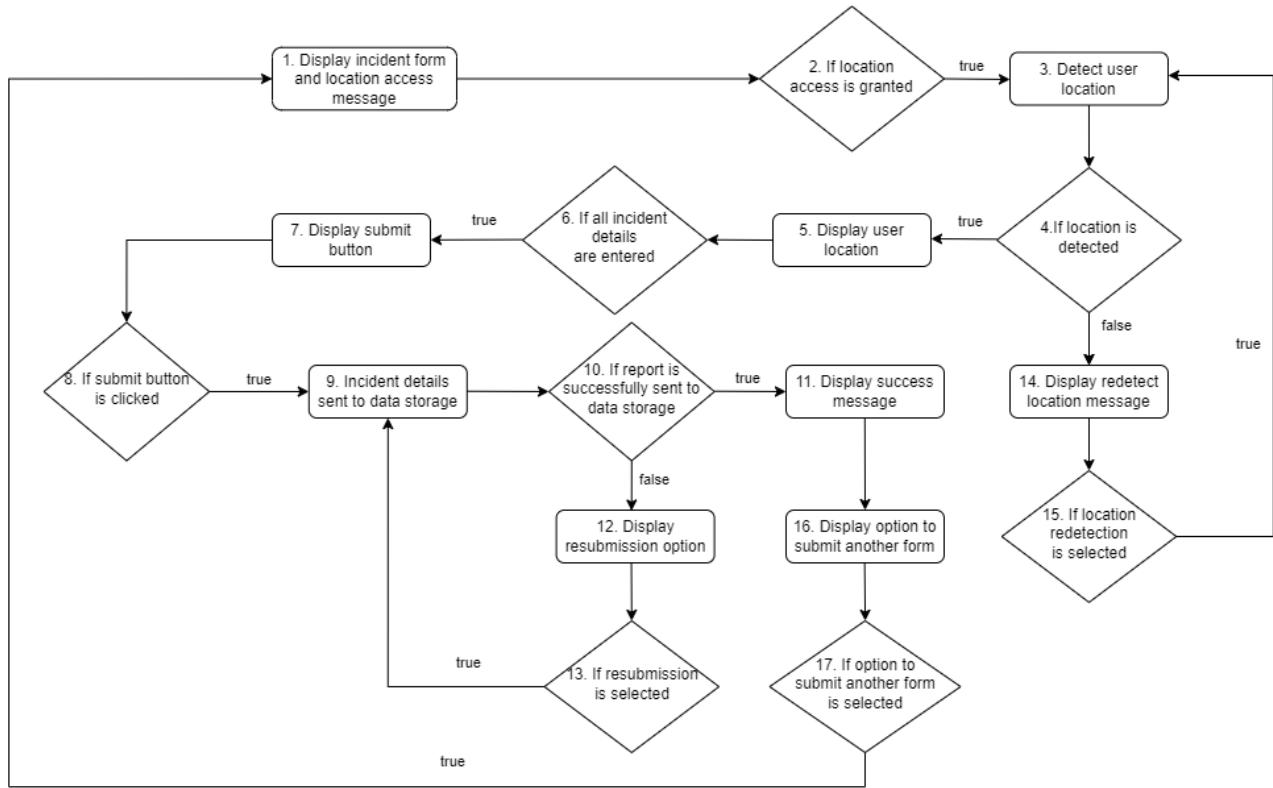
Click Search and Filter dropdown Click on Pin of Camera 6705	Flow: 1,1A,19,20  Display Traffic Camera Information “Camera 6705 Vehicle Count: xx Peakedness: xx% View Camera”	Flow: 1,1A,19,20  Display Traffic Camera Information “Camera 6705 Vehicle Count: 10 Peakedness: 47.6% View Camera”
---	--	--

Test Case 7: Filter by incidents		
Location: “app/src/pages/Map.tsx”		
Inputs	Expected Flow and Outputs	Actual Flow and Outputs
Click Search and Filter dropdown Click Hide All Click Show All Click Accidents Click Roadworks Click Closures	Flow: 1,1A,23,24  Hide all incidents on map. Show all incidents on map. Hide accidents on map, showing only roadworks and closures. Hide roadworks on map, showing only closures. Hide closures on map, hiding all.	Flow: 1,1A,23,24  Hide all incidents on map. Show all incidents on map. Hide accidents on map, showing only roadworks and closures. Hide roadworks on map, showing only closures. Hide closures on map, hiding all.

Test Case 8: Filter by traffic cameras		
Location: “app/src/pages/Map.tsx”		
Inputs	Expected Flow and Outputs	Actual Flow and Outputs
Click Search and Filter dropdown Click Off Click Heatmap Click Camera	Flow: 1,1A,21,22  Hide all traffic cameras and heatmap Show Heatmap Show Traffic Cameras	Flow: 1,1A,21,22  Hide all traffic cameras and heatmap Show Heatmap Show Traffic Cameras

### 8.2.2 Report Incident

Control Flow Graph



**Figure 8:** Control Flow Graph for Report Incident

Test Case 1: Valid user inputs, valid location and valid form submission		
Location: "app/src/pages/ReportIncident.tsx"		
Inputs	Expected Flow and Outputs	Actual Flow and Outputs
Incident Type = 'Accidents' Incident Description = 'Car collision on the right lane' Incident Location = '82 Bukit Batok Road'  Select to submit another form	Flow: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 16, 17, 1  "Incident Location: 82 Bukit Batok Rd, Singapore" (coordinates: 1.3732, 103.7522)  "Incident is successfully reported."	Flow: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 16, 17, 1  "Incident Location = Bukit Batok Road, Choa Chu Kang, West Region, 658517, Singapore" (coordinates: 1.3732, 103.7523)  "Incident is successfully reported"

Test Case 2: Valid user inputs, invalid location and valid form submission		
Location: "app/src/pages/ReportIncident.tsx"		
Inputs	Expected Flow and Outputs	Actual Flow and Outputs
Incident Type = 'Roadworks' Incident Description = 'Construction works on left lane' Location access denied	Flow: 1, 2  No location displayed	Flow: 1, 2  No location displayed
Incident Type = 'Roadworks' Incident Description = 'Construction works on left lane' Incident Location: '82 Bukit Batok Road'  Fails to fetch data from LocationIQ API due to network errors	Flow: 1, 2, 3, 4, 14, 15  "Failed to detect address. Redetect current location?"	Flow: 1, 2, 3, 4, 14, 15  "Failed to detect address. Redetect current location?"

Test Case 3: Invalid user inputs, valid location and valid form submission		
Location: "app/src/pages/ReportIncident.tsx"		
Inputs	Expected Flow and Outputs	Actual Flow and Outputs
Incident Type = '' Incident Description = 'Closure on right lane' Incident Location = '81 AYE'	Flow: 1, 2, 3, 4, 5, 6  "Incident Location: 81 AYE" (coordinates: 1.3233, 103.746)  Submit button not displayed.	Flow: 1, 2, 3, 4, 5, 6  "Incident Location: Teban Flyover, Ayer Rajah Expressway, Teban Gardens, Jurong East, Jurong East, West Region, 609338, Singapore" (coordinates: 1.3247, 103.742)  Submit button not displayed.
Incident Type = 'Closure' Incident Description = 'Closure on right lane' Incident Location = '81 AYE'		

Test Case 4: Valid user inputs, valid location and invalid form submission		
Location: "app/src/pages/ReportIncident.tsx"		
Inputs	Expected Flow and Outputs	Actual Flow and Outputs
Incident Type = 'Accidents' Incident Description = 'Tree has fallen onto main road.' Incident Location = 'Rail Corridor' Error in posting data using Reports API Selection to resubmit	Flow: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 9, 10, 12  "Incident Location: Rail Corridor" (coordinates: 1.3078, 103.7915)  "Error in report submission"  "Error in report submission"	Flow: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 9, 10, 12  "Incident Location: Rail Corridor, Ghim Moh, Queenstown, Central Region, 139350, Singapore" (coordinates: 1.3074, 103.7906)  "Error in report submission"  "Error in report submission"

## 9. Appendices

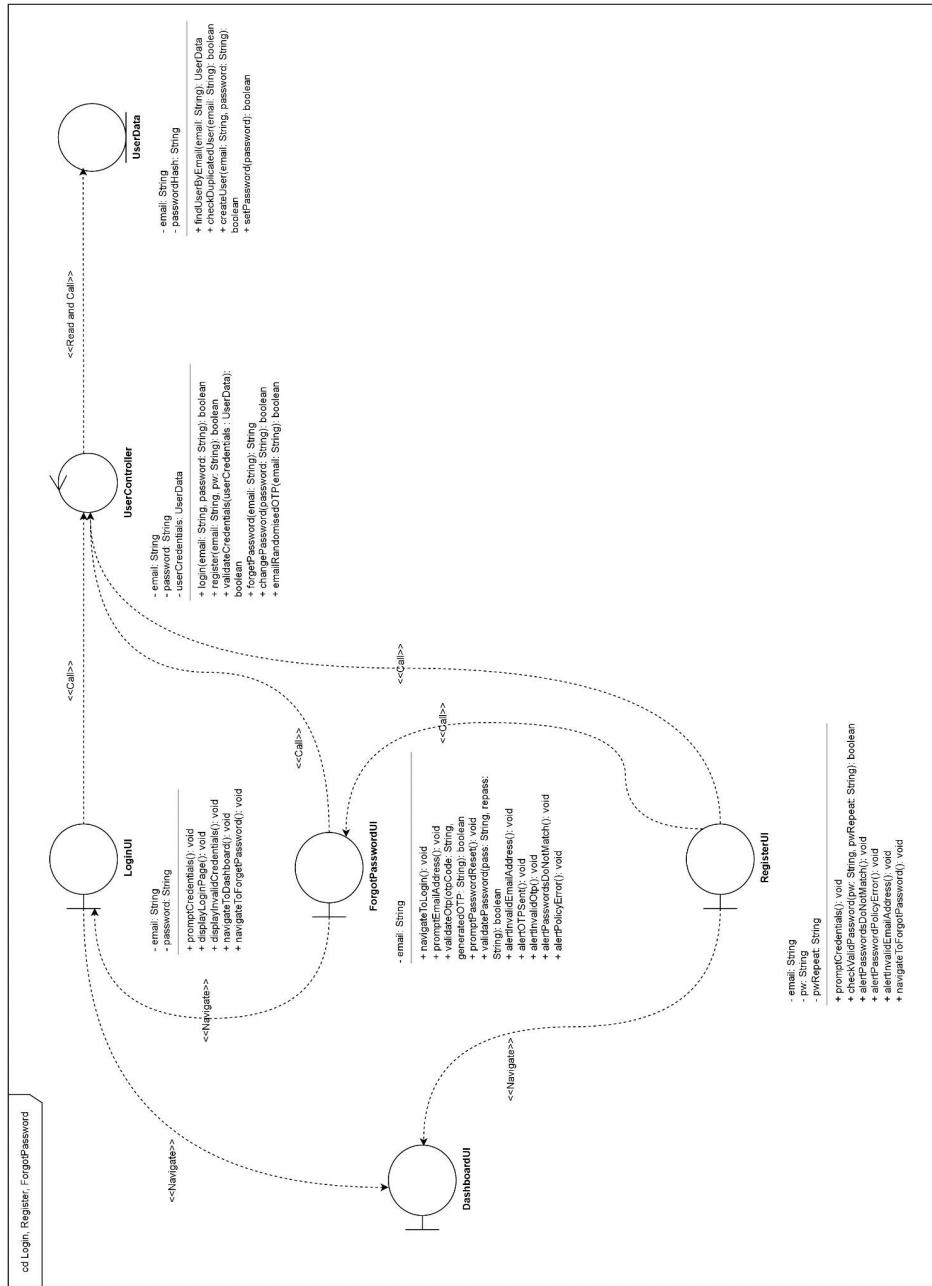
### Appendix A: Data Dictionary

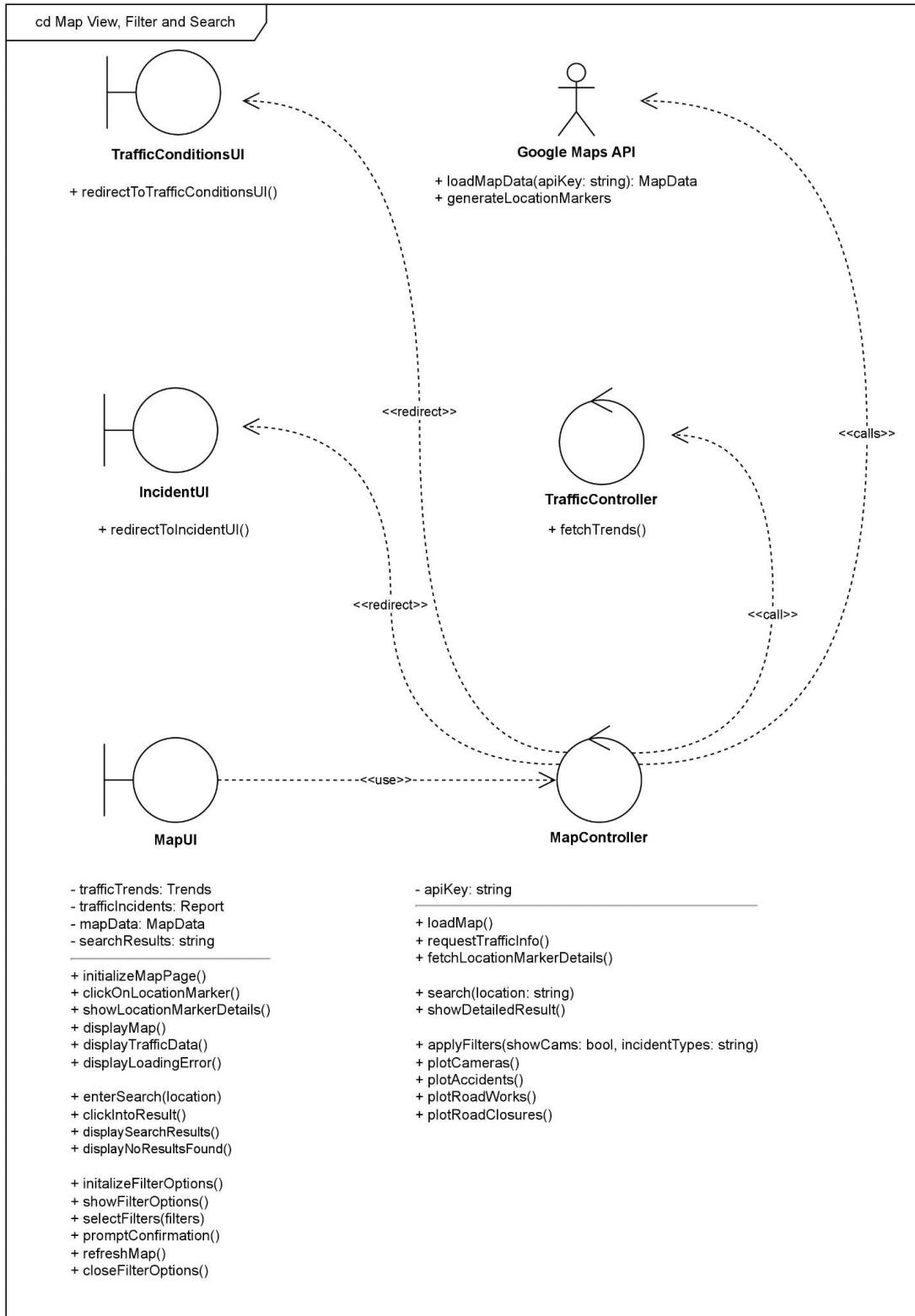
Term	Definition
User	A driver who utilises the application to gain real-time information on traffic congestion levels, trends of traffic congestion levels and locations of roadblocks to plan out their routes.
Account	An identity created for a user. Every user account has a unique email address and password.
Traffic congestion level	The level of vehicle occupancy in an area, which can span multiple roads. The system's AI model will classify roads as high, moderate or low congestion.
Traffic camera	A camera currently operated to monitor vehicular traffic and its image data is retrievable from the Traffic Image API.
Traffic image	An image taken by a traffic camera that is retrievable from the Traffic Image API.
Driving road	A road that is mainly utilised by vehicles and which people are usually prohibited from travelling on except when crossing roads.
Accident	An incident that may result in injuries or damage that occurs on driving roads.
Roadwork	A driving road involved in construction works may be barricaded and barriers may be set up on the driving road.
Closure	A driving road closed by authorities cannot be used by the public.
Slow traffic	Traffic flow that is slower than what the user typically experiences. The speed of traffic flow is subjective to each user.
Real-time	Current time or not more than 20 seconds ago as the Traffic Image API retrieves images from traffic cameras every 20 seconds.
Driving route	A path on driving roads from a starting point to a destination.

## Appendix B: Analysis Models

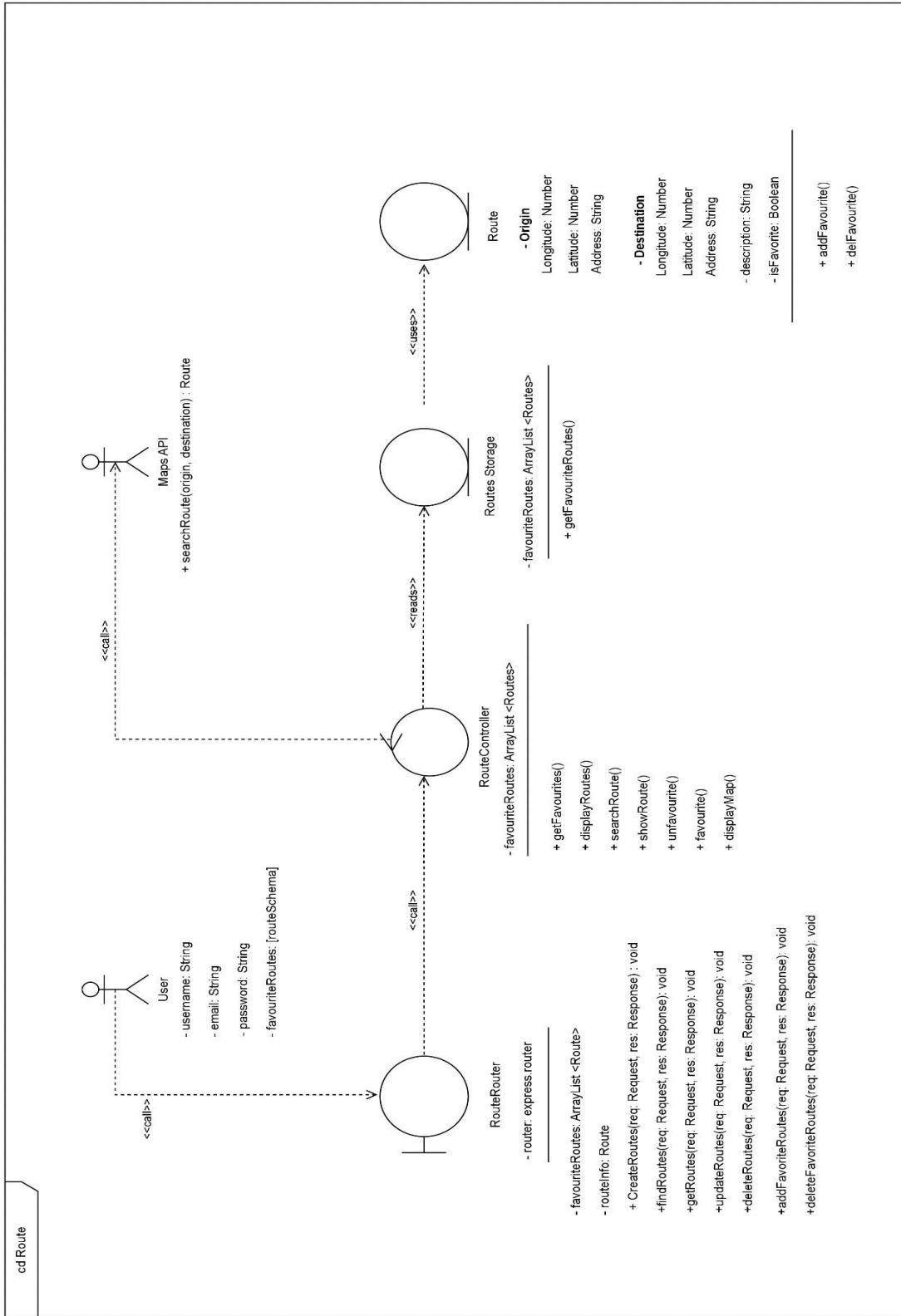
### Class Diagrams

#### Login, Register, Forgot Password

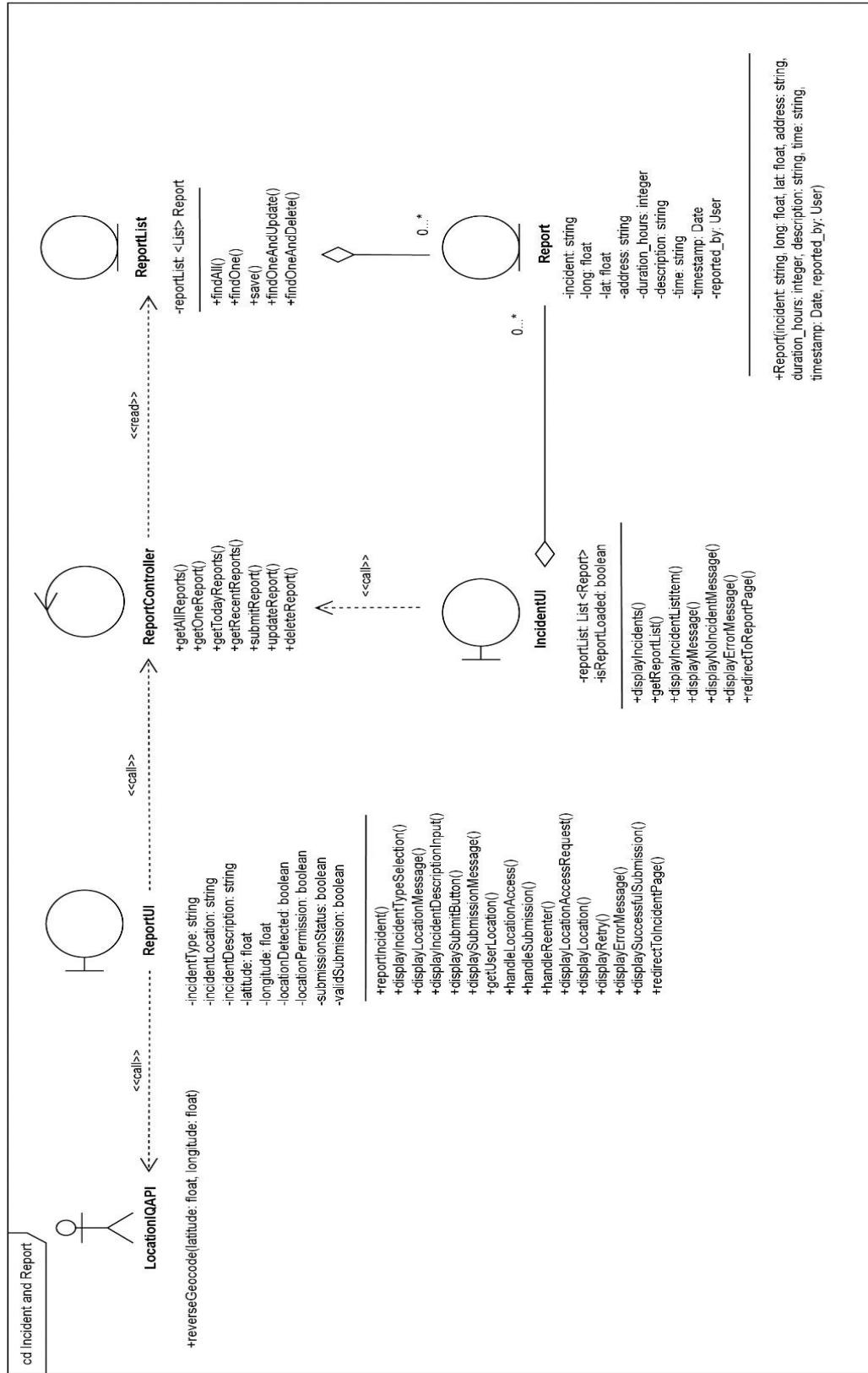


**Map View, Filter, Search**

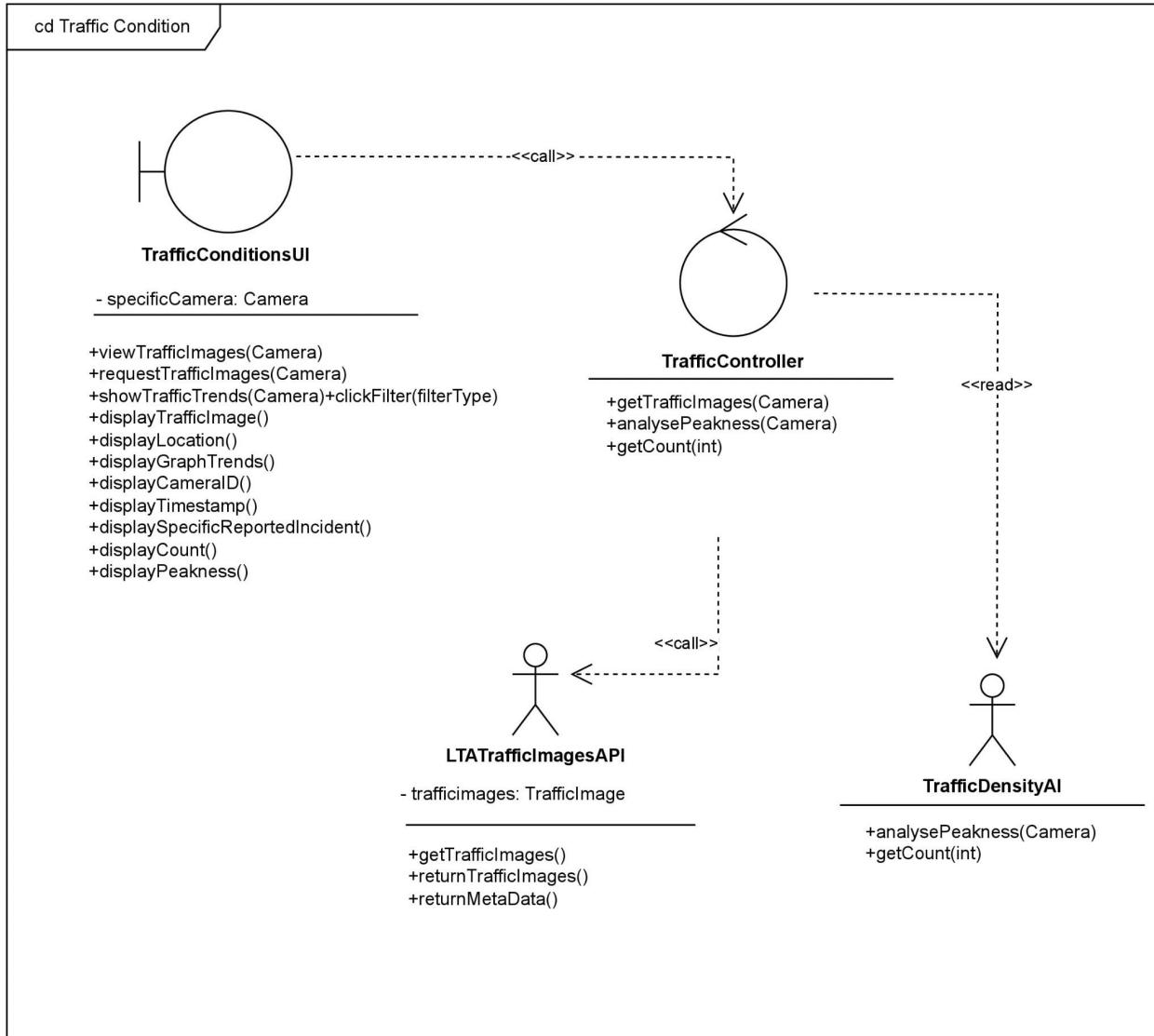
## Routes



## Incident and Report

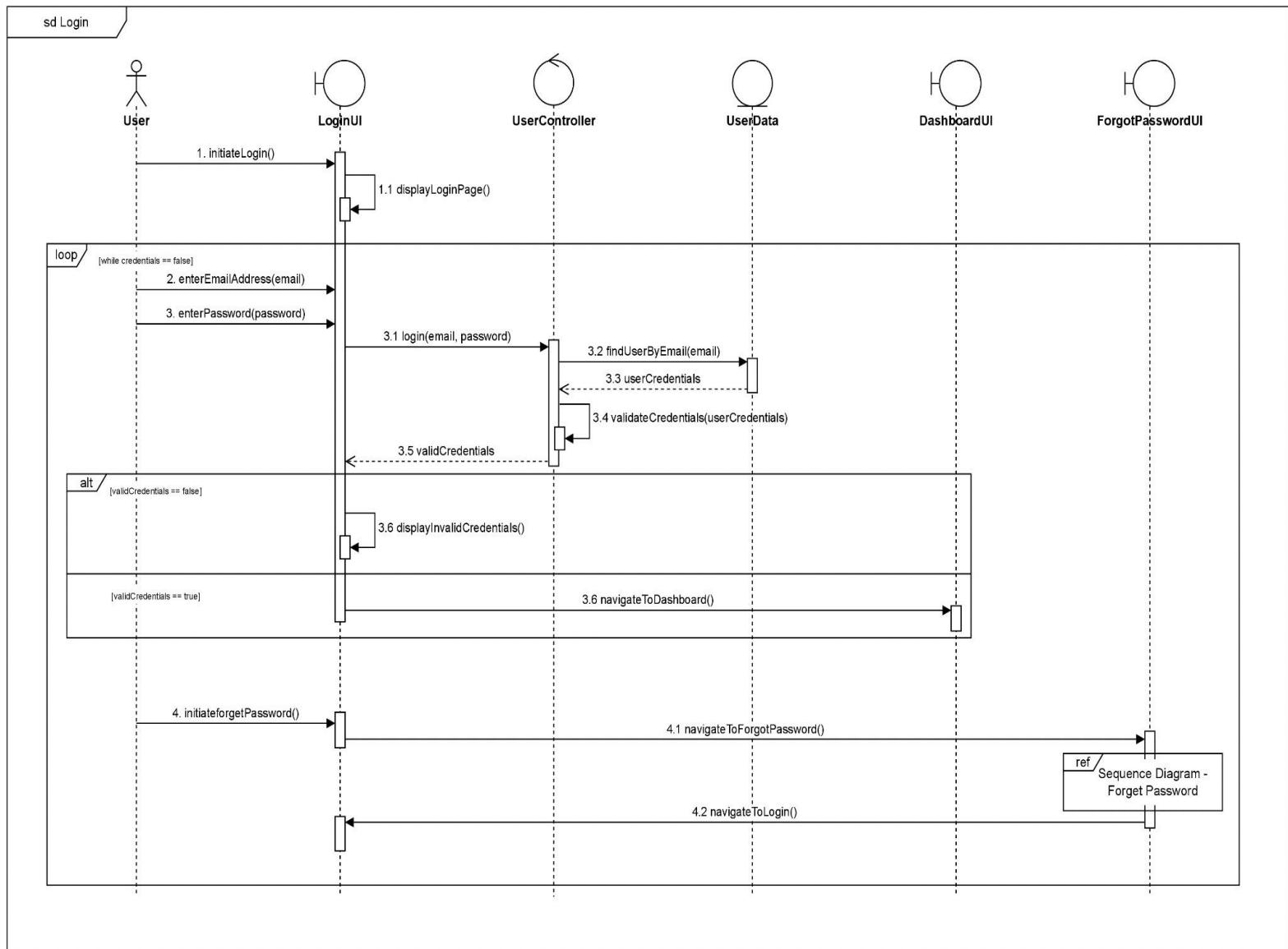


### Traffic Condition

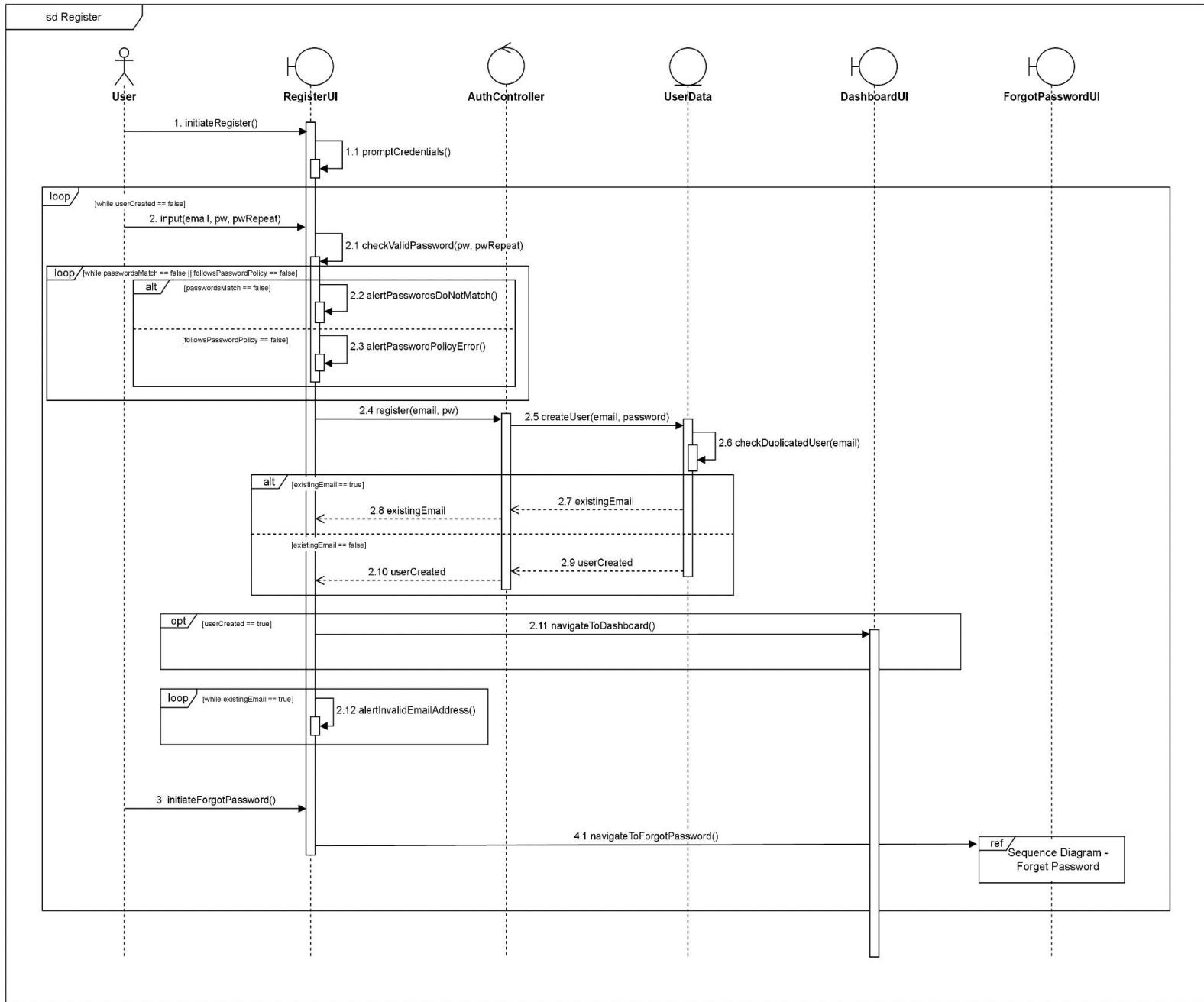


## Sequence Diagrams

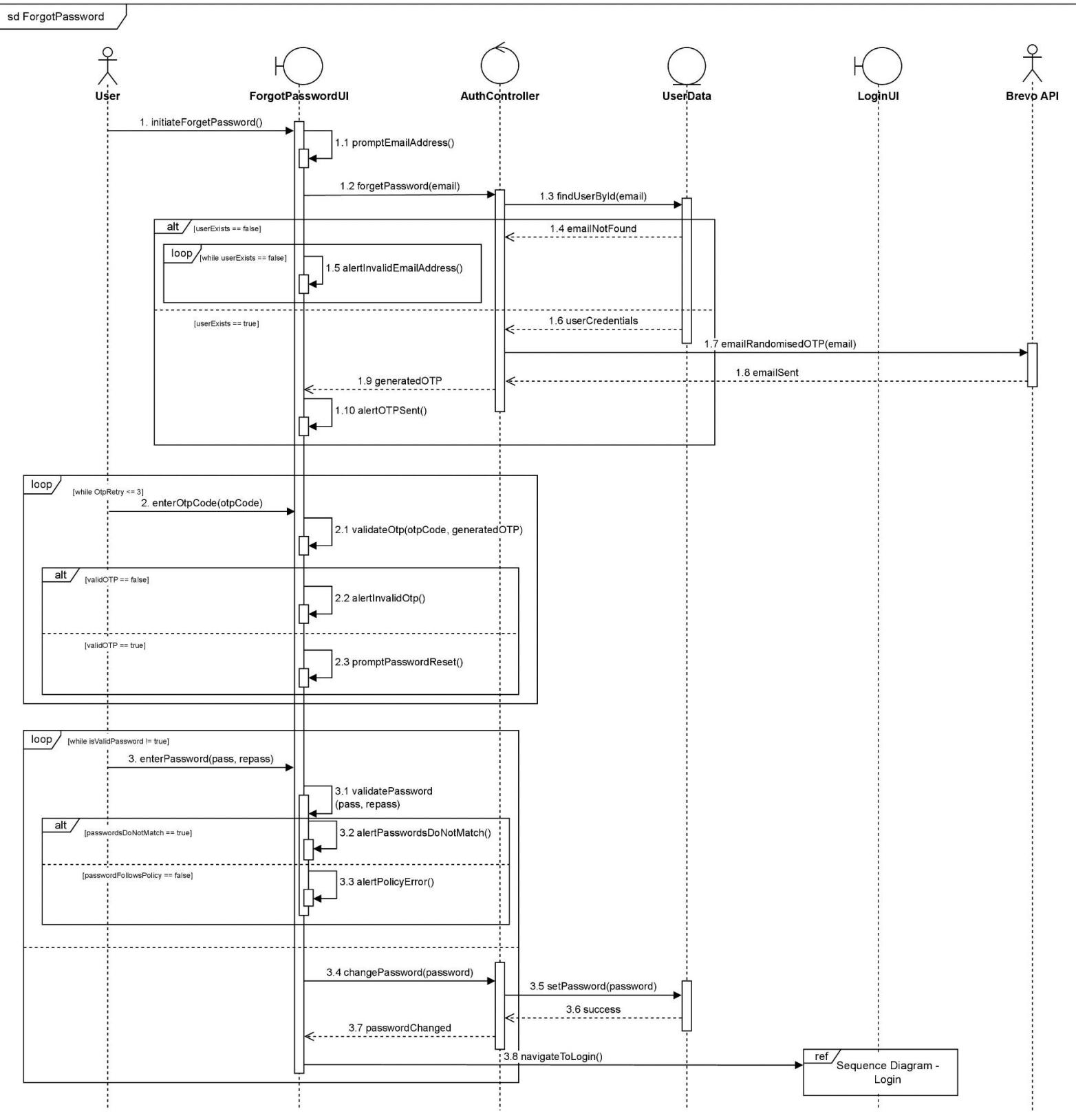
### Login



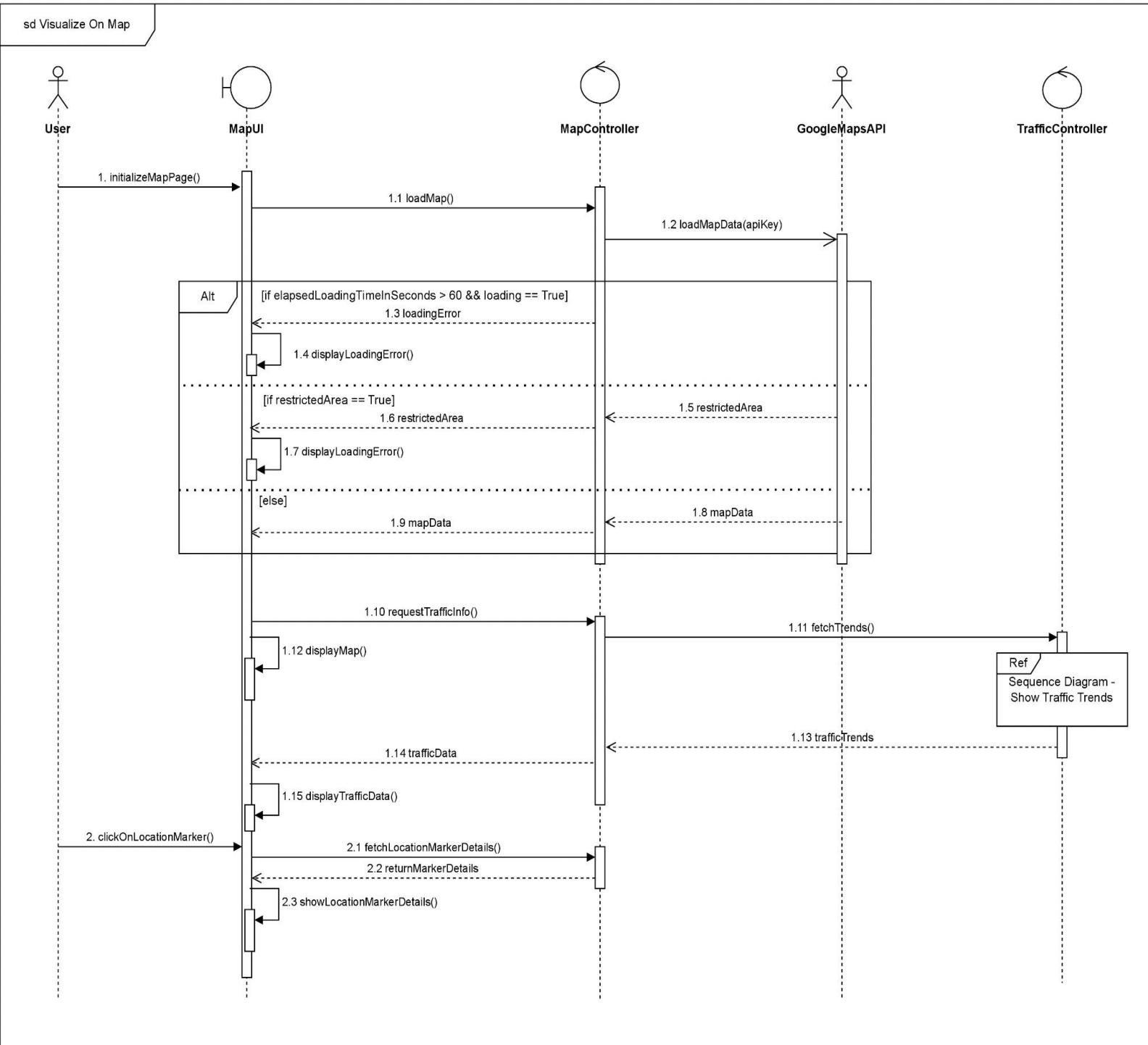
## Register



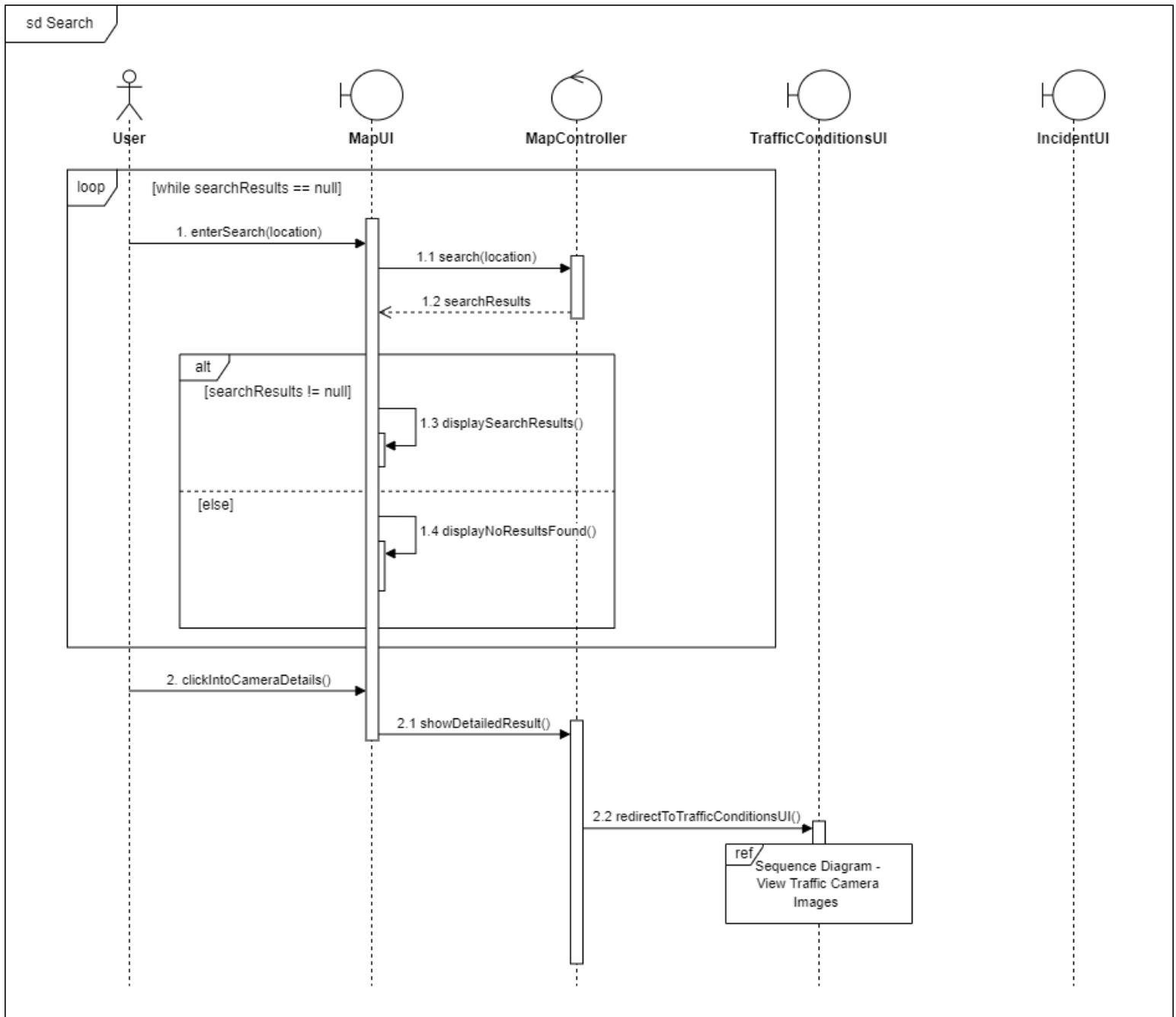
## Forgot Password

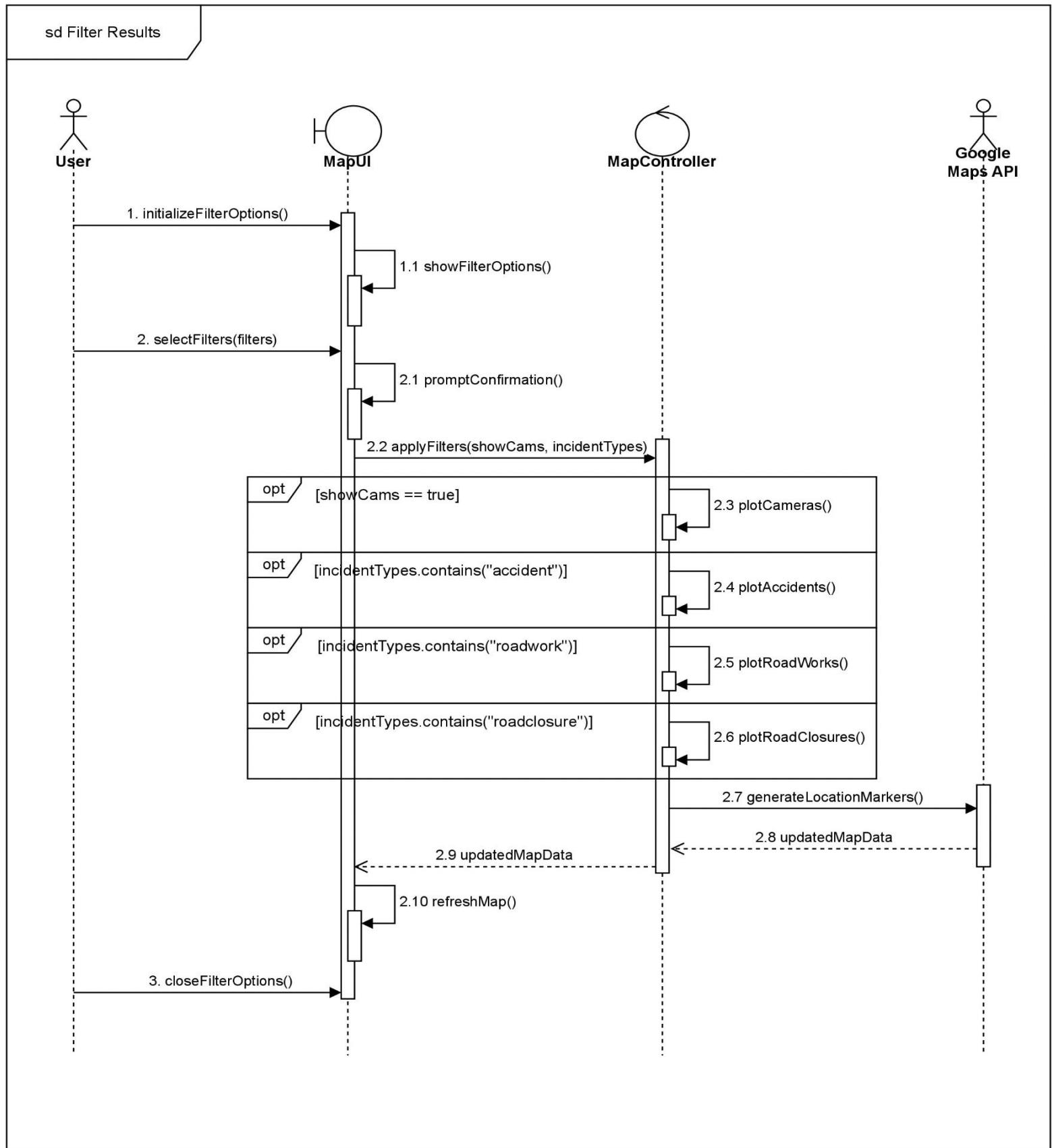


### Visualise on Map

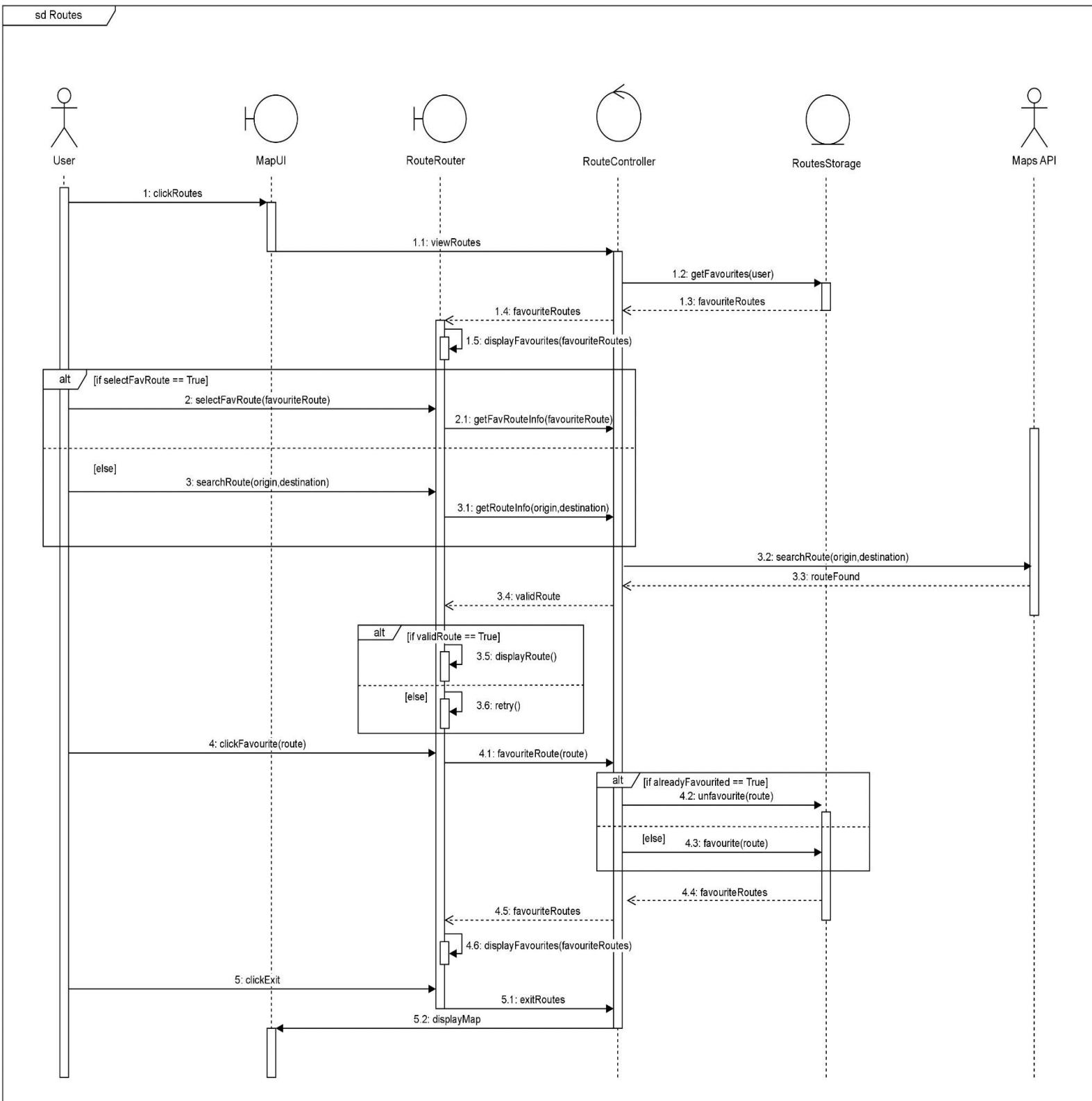


## Search Results

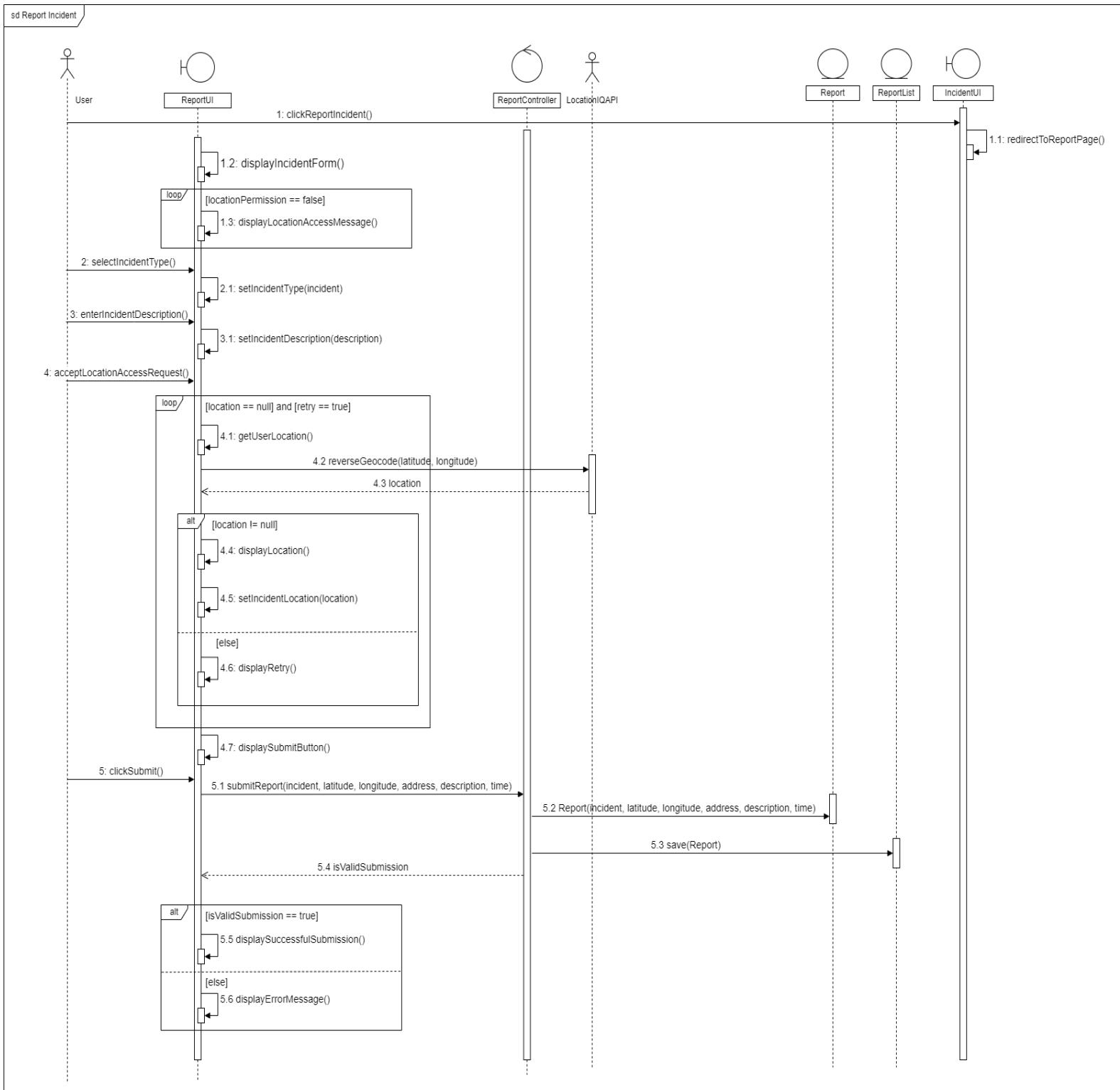


**Filter Results**

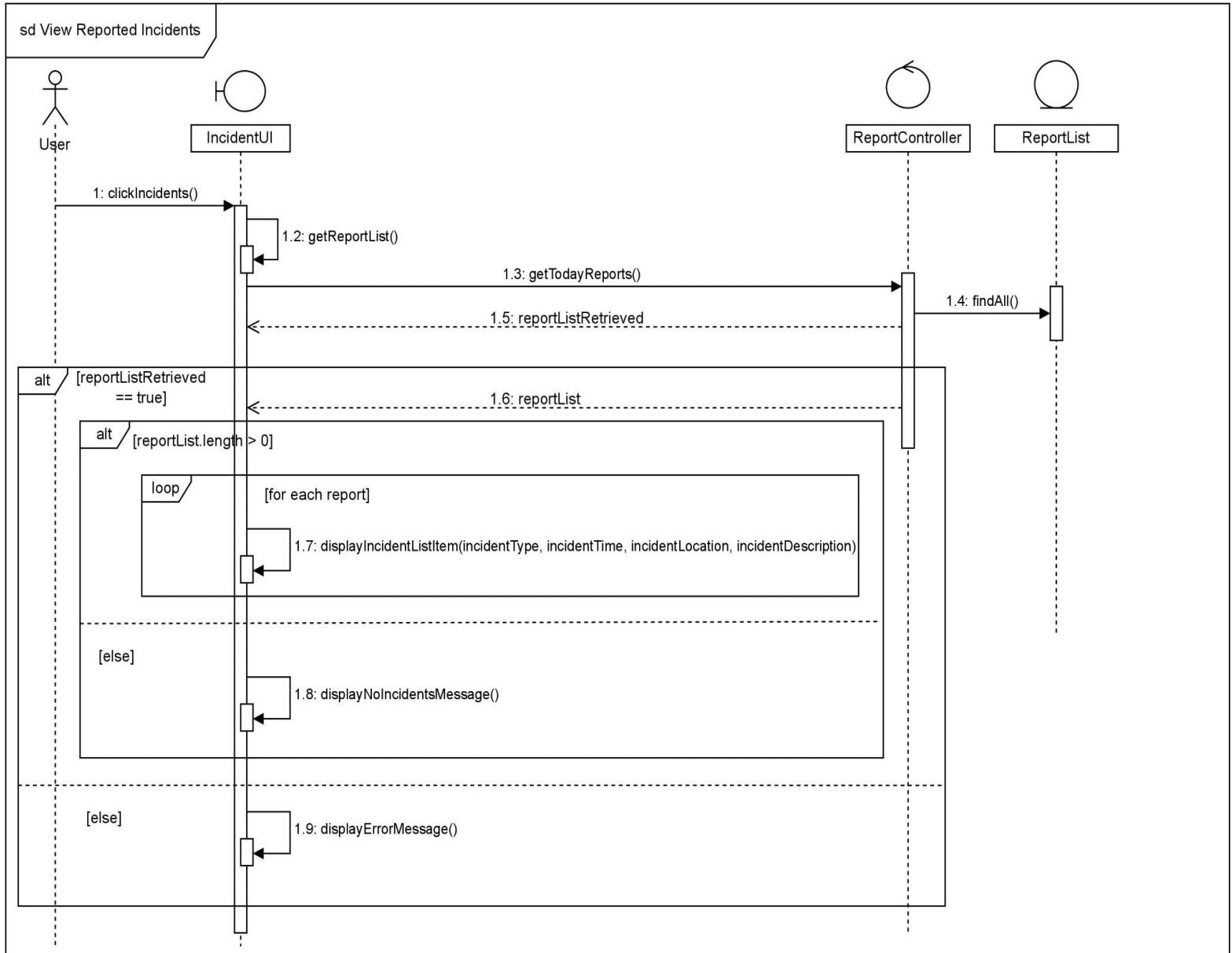
## Routes



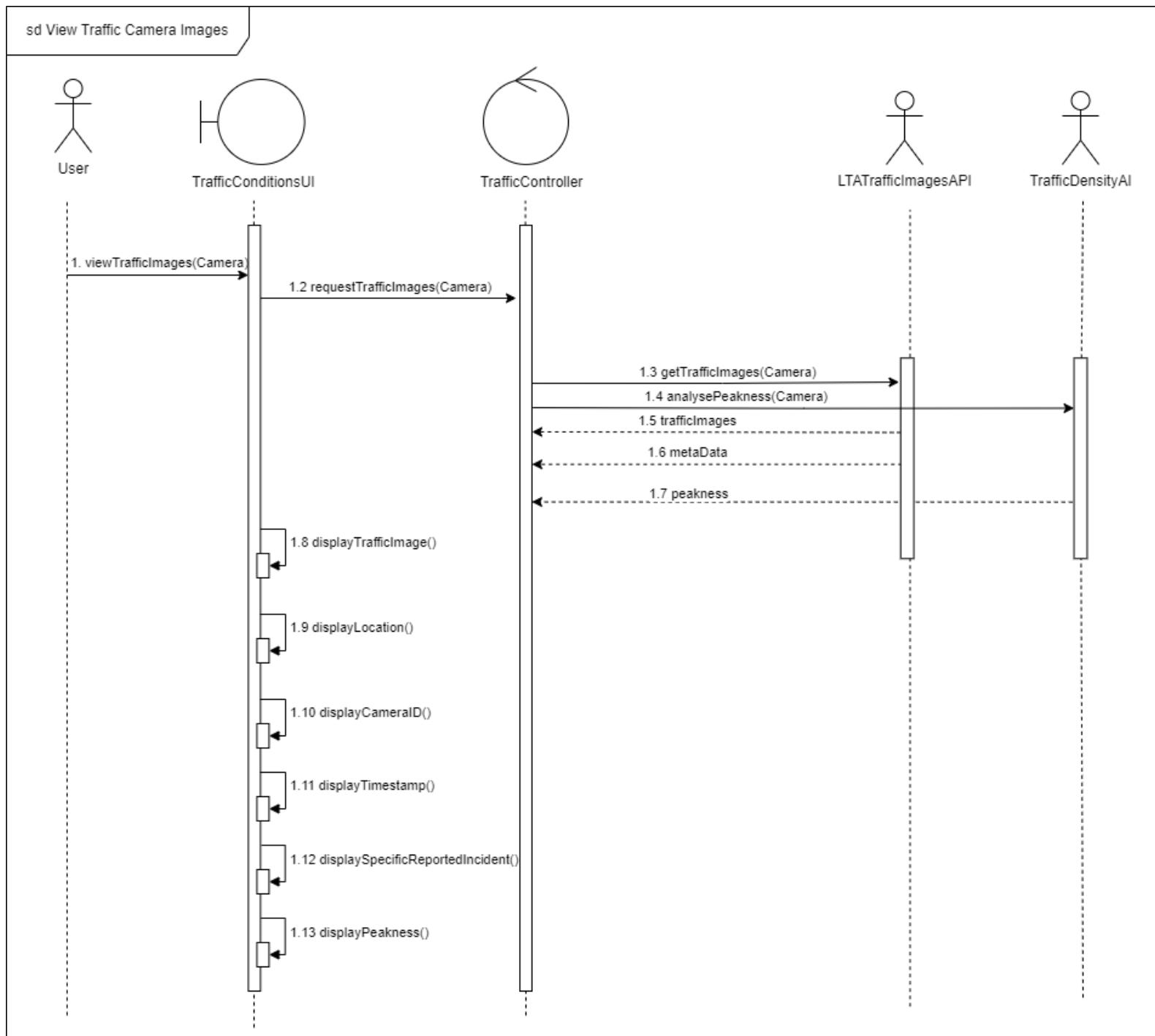
## Report Incidents

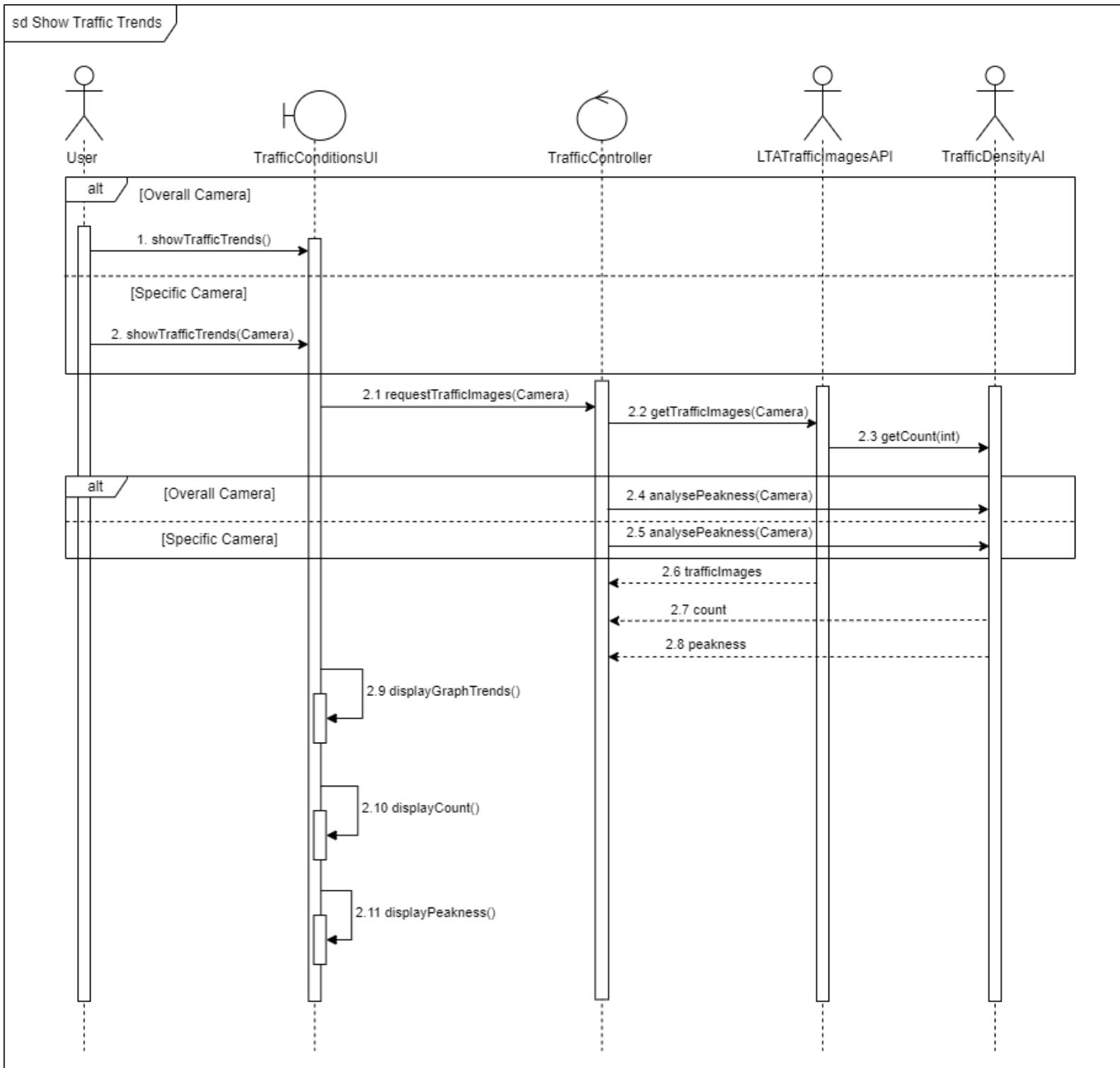


### View Reported Incidents

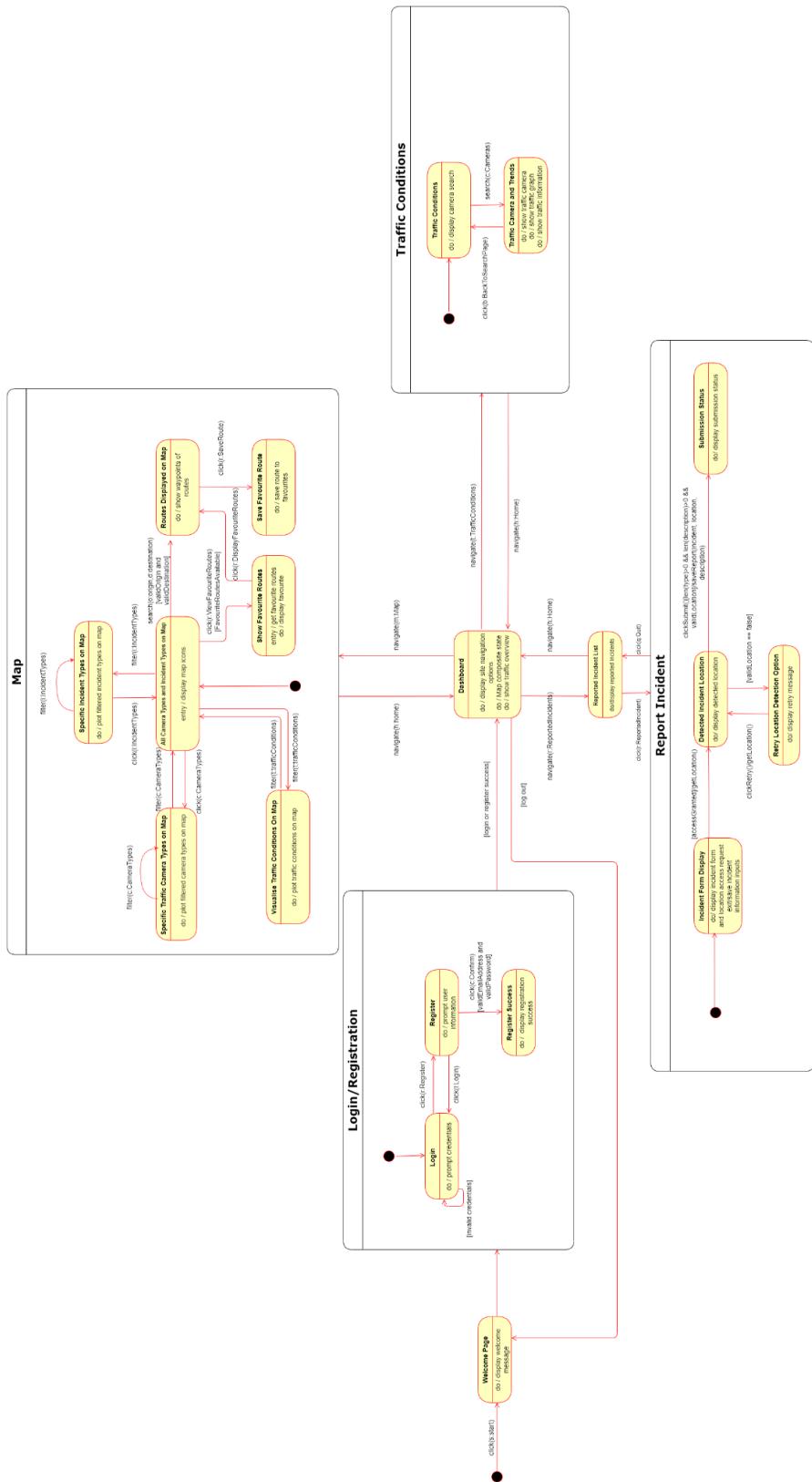


### View Traffic Camera Images

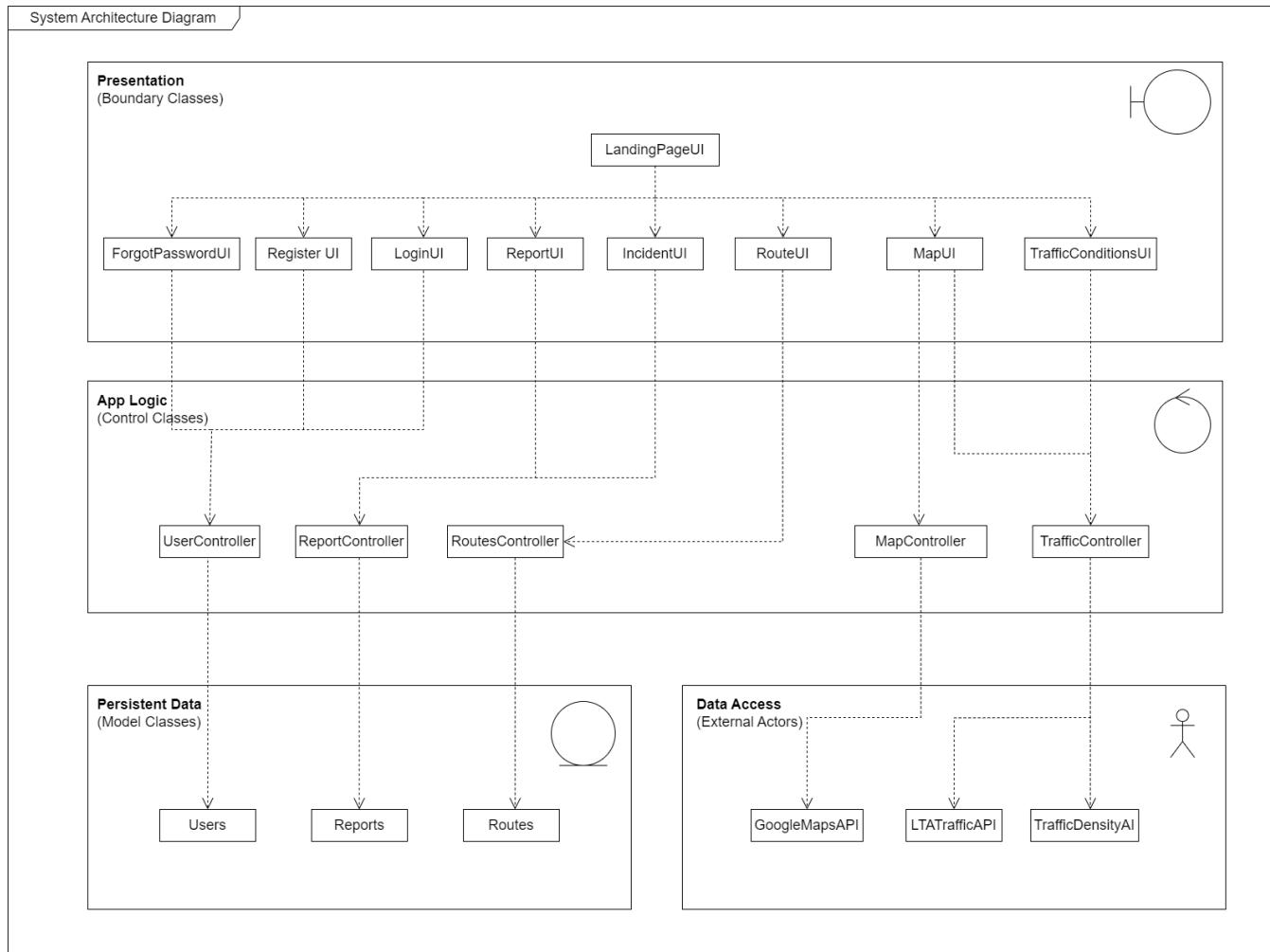


**Show Traffic Trends**

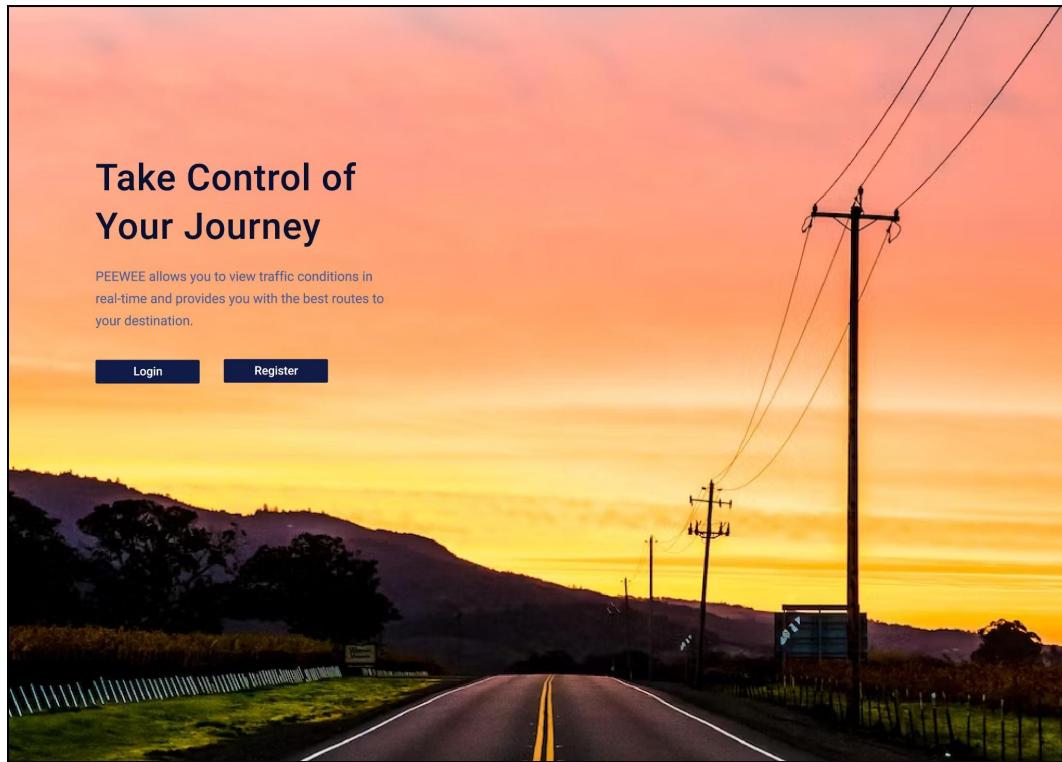
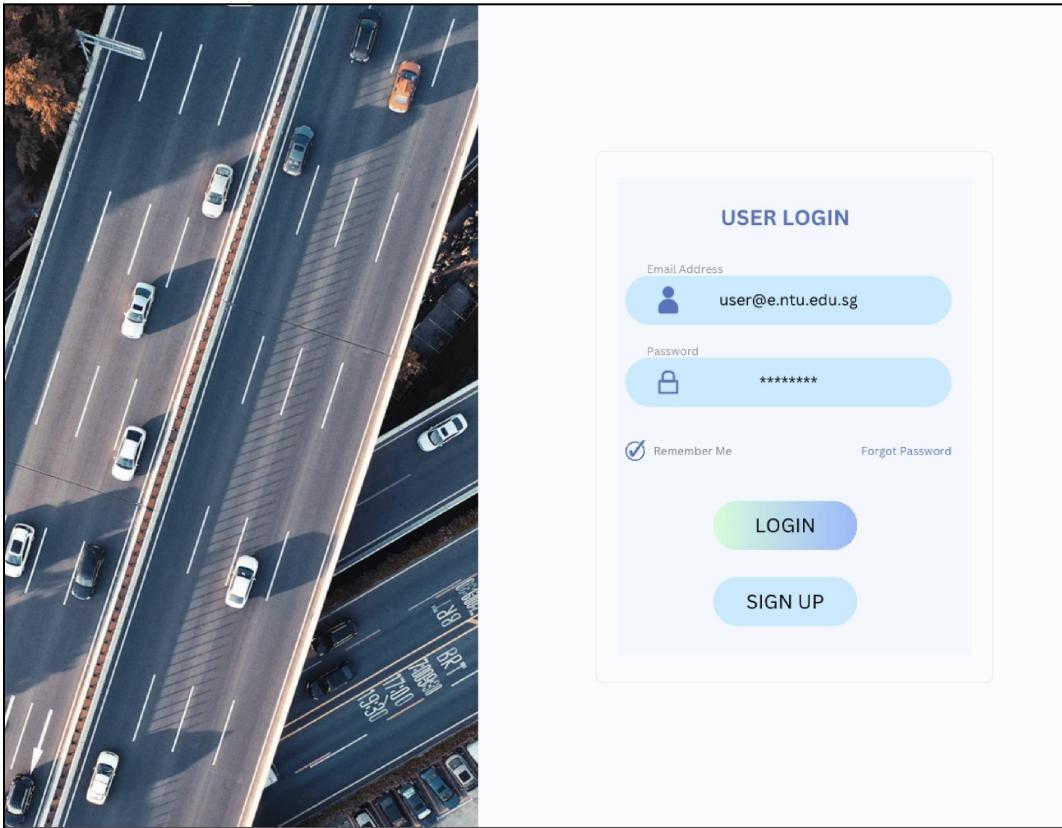
# Dialog Map



## System Architecture



## UI Mockups



### PEEWEE

## Dashboard

Dashboard Incidents Map Road Conditions Log Out

### TRAFFIC TREND

DAY HOUR MIN

### TRAFFIC OVERVIEW

DROP PIN SEARCH SAVE

### RECENTS

### FAVOURITES

### PEEWEE

## Road Conditions

Dashboard Incidents Map Road Conditions Log Out

### Traffic Trends

### Traffic Images

Land Transport Authority

ECP(CITY)

KPE(ECP)

PIE(CHANGI)

PIE(TUAS)

KPE(TPE)

ECP(CHANGI)

Land Transport Authority

Land Transport Authority

PIE(CHANGI)

PIE(CHANGI)

KPE(ECP)

KPE(TPE)

Land Transport Authority

KPE(ECP)

PIE(TUAS)

Minute Hour Day

### Road Conditions

Search a location

Traffic Trends

Location: Causeway  
Camera ID: 2701  
Timestamp: 2023-09-04 23:15:00  
Traffic Level: HIGH

### Favourite Routes

Start: Blk 366A Yio Chu Kang Ave 8 [View on Map](#)  
End: Nanyang Technological University [Unfavourite](#)

Start: Nanyang Technological University [View on Map](#)  
End: Blk 366A Yio Chu Kang Ave 8 [Unfavourite](#)

[Back to Map](#)

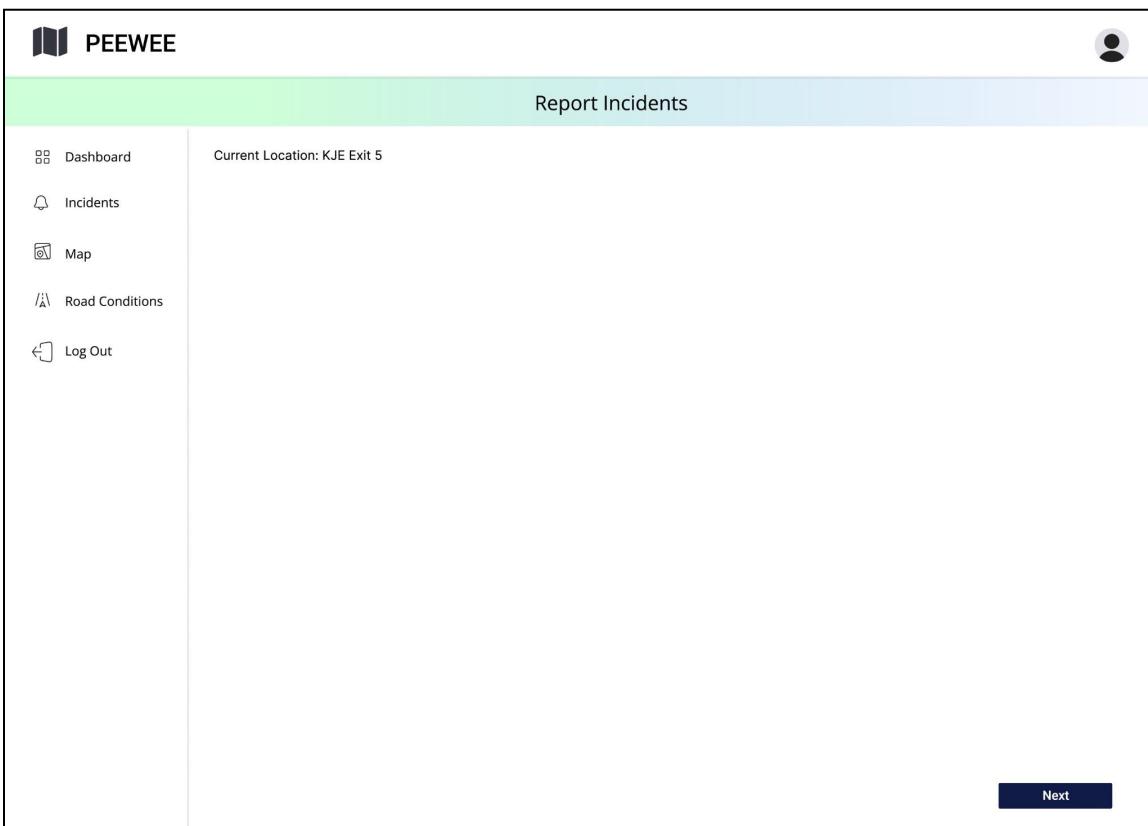
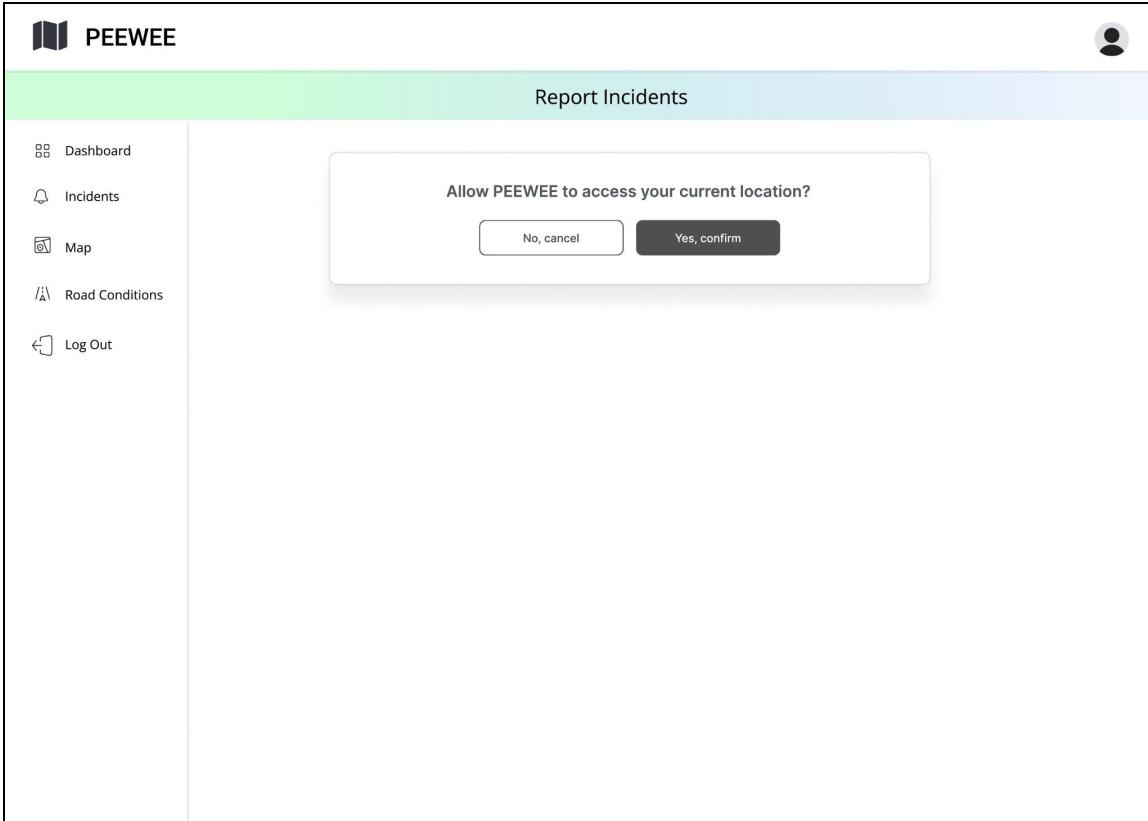
The screenshot shows the PEEWEE application interface. At the top left is the logo and the word "PEEWEE". At the top right is a user profile icon. The main header is "Incidents". On the left sidebar, there are links for Dashboard, Incidents (which is selected), Map, Road Conditions, and Log Out. The main content area contains two sections: "ACCIDENT" and "ROADWORK".  
**ACCIDENT:**  
- Time: 12:45  
- Location: KJE Exit 5  
- Description: A car collided with a bicycle at the leftmost lane, slight congestion due to traffic redirection.  
**ROADWORK:**  
- Time: 18:30  
- Location: PIE Exit 26A  
- Description: Construction works on the rightmost lane, lane is partially closed, traffic is heavy near the area.

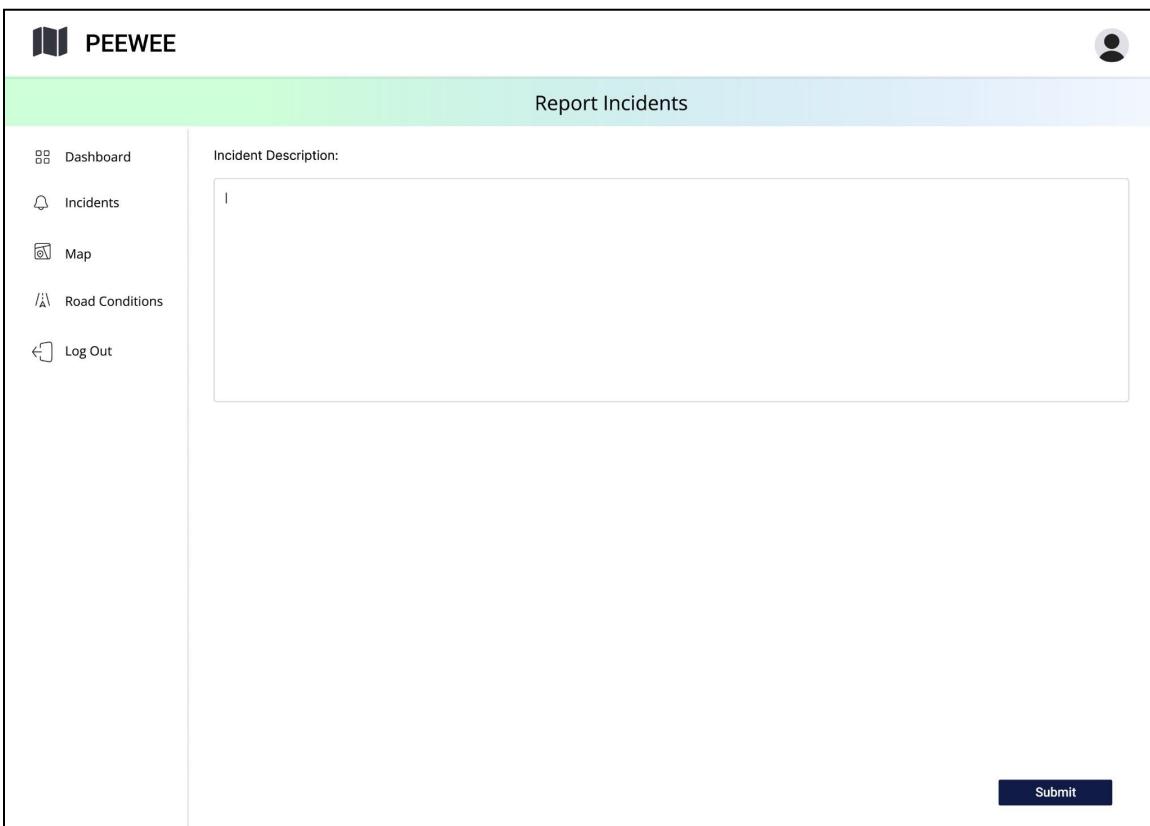
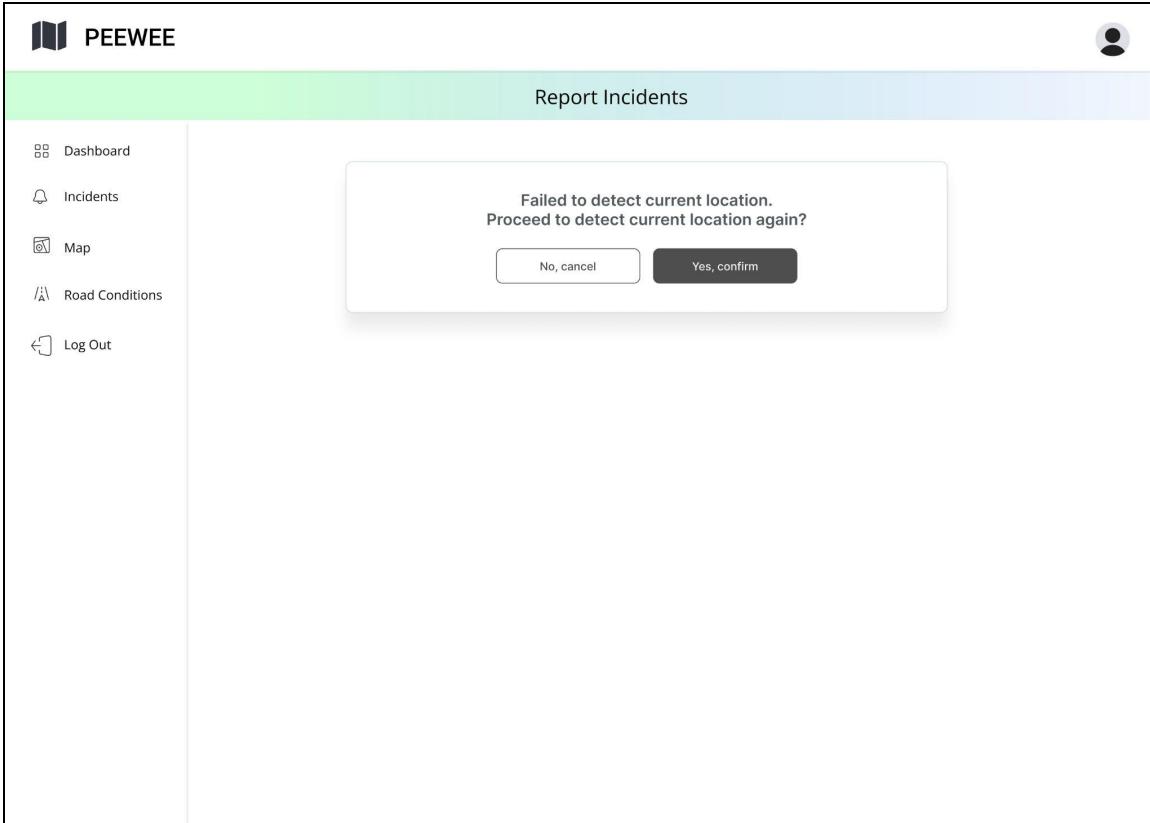
Report

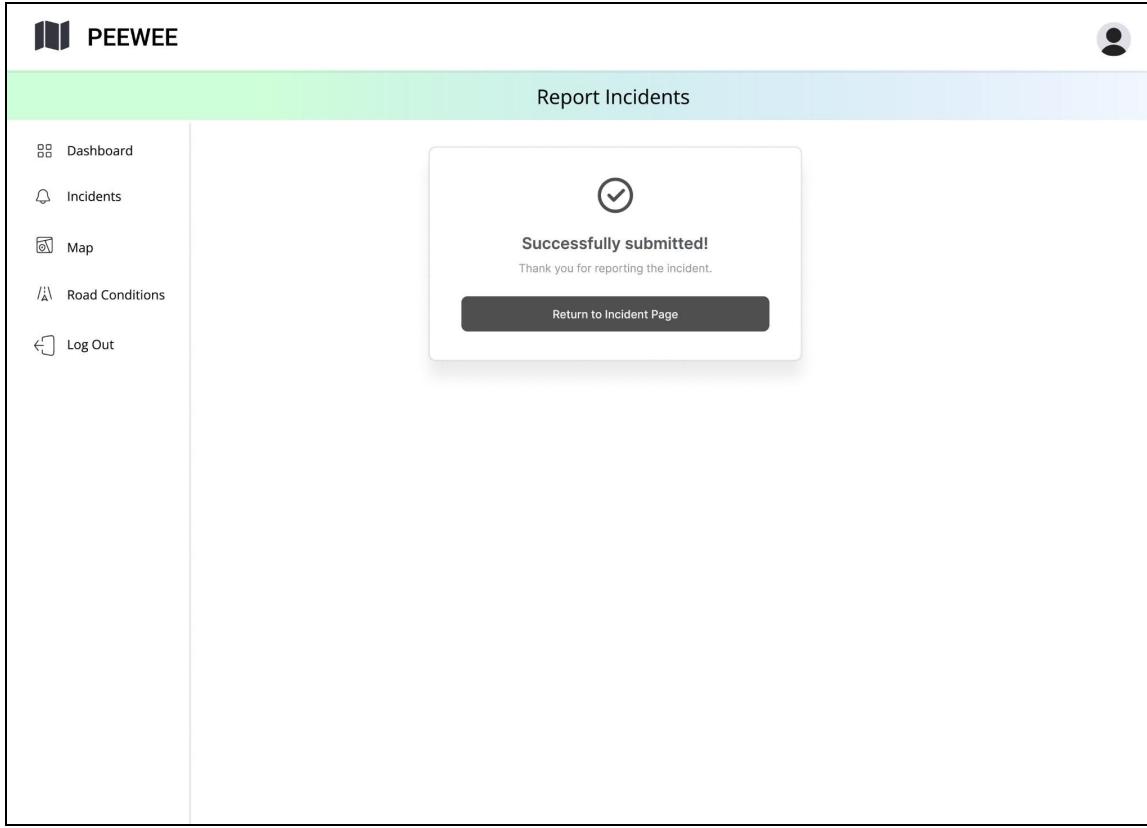
The screenshot shows the PEEWEE application interface for reporting incidents. At the top left is the logo and the word "PEEWEE". At the top right is a user profile icon. The main header is "Report Incidents". On the left sidebar, there are links for Dashboard, Incidents, Map, Road Conditions, and Log Out. The main content area has a dropdown menu titled "Incident Type" with the following options:

- Accident
- Roadworks
- Closure
- Slow Traffic

Next







These UI Mockups were designed in Figma. Alternatively, please click into this link to [visit our Figma page](#) and view our design files instead of the screenshots shown here.

## Appendix C: User Interface Screenshots

Please access our [demonstration video](#) for a more cohesive overview of the Peewee User Interface.

### Dashboard

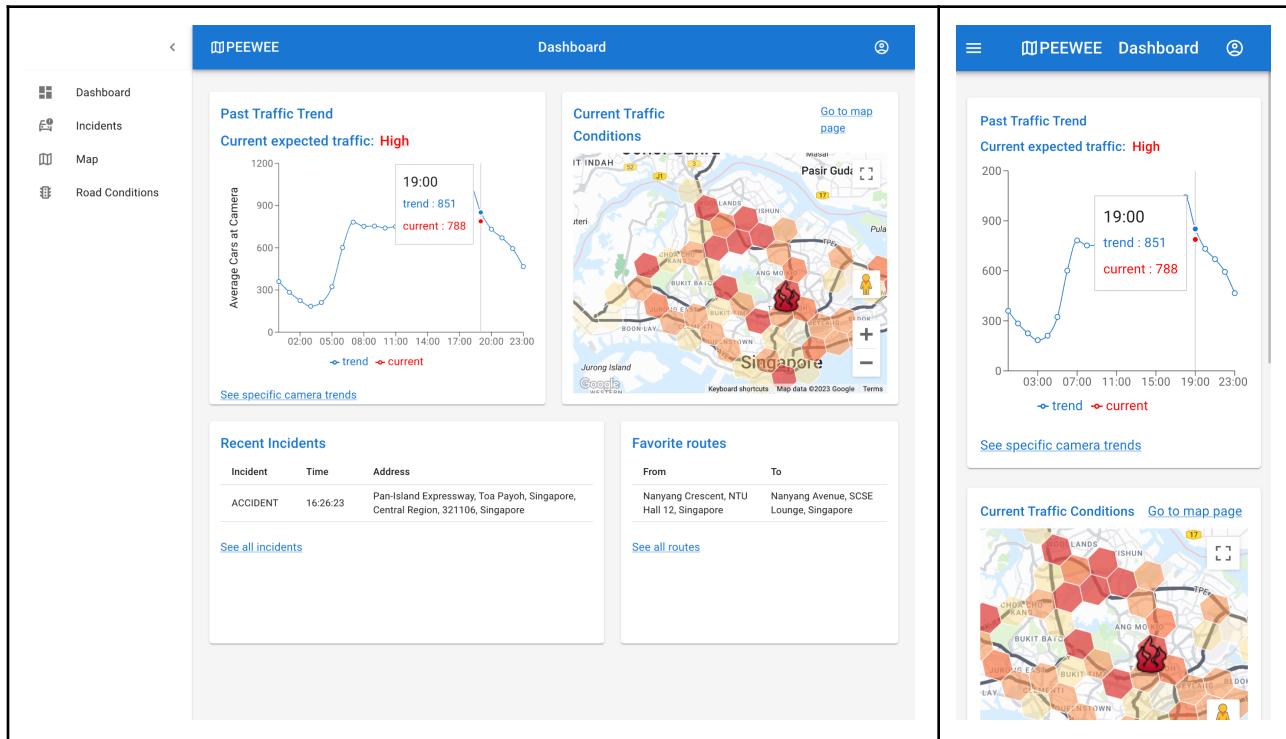
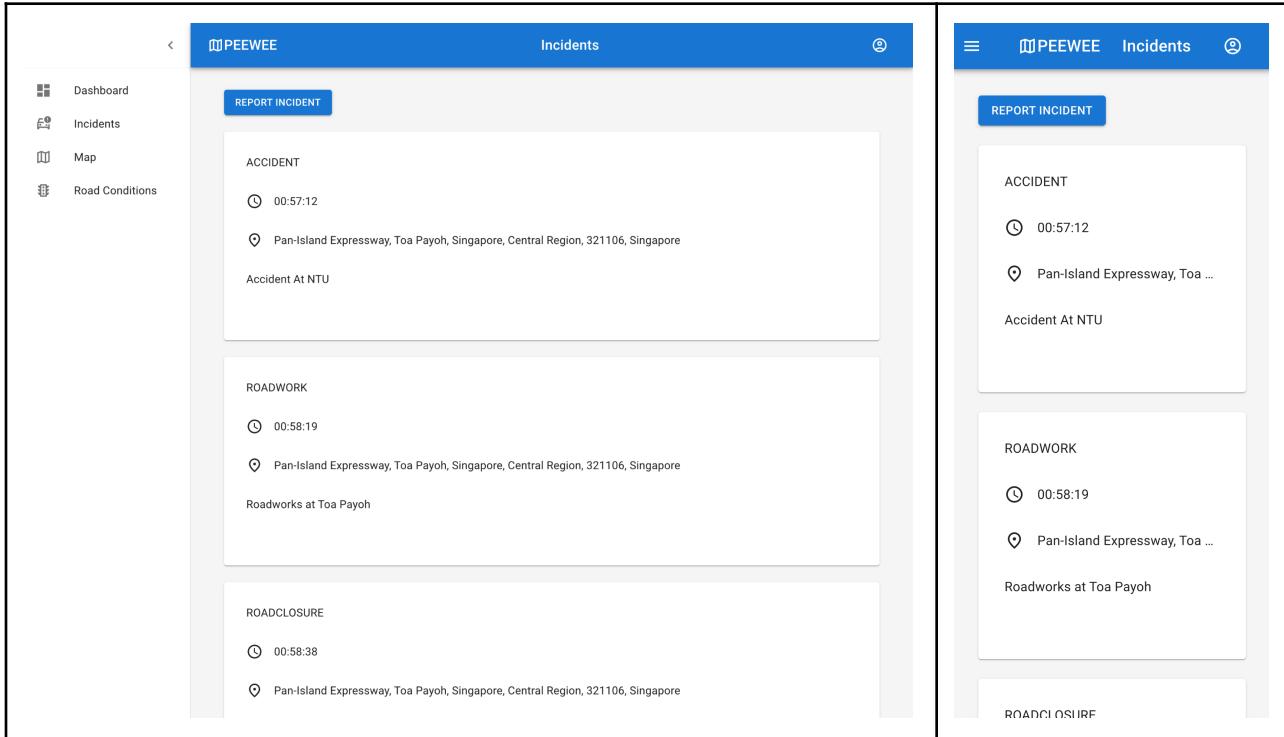


Figure C1: Dashboard page

The Dashboard contains useful information at a glance. Miniature, simplified components such as Traffic Trends, the Map, Recent Incidents, and Favourite Routes are shown in the dashboard for convenience. Based on the data the user can then select into the many links provided to view in depth information in their respective specialised pages. On mobile, interface elements shrink to display content comfortably.

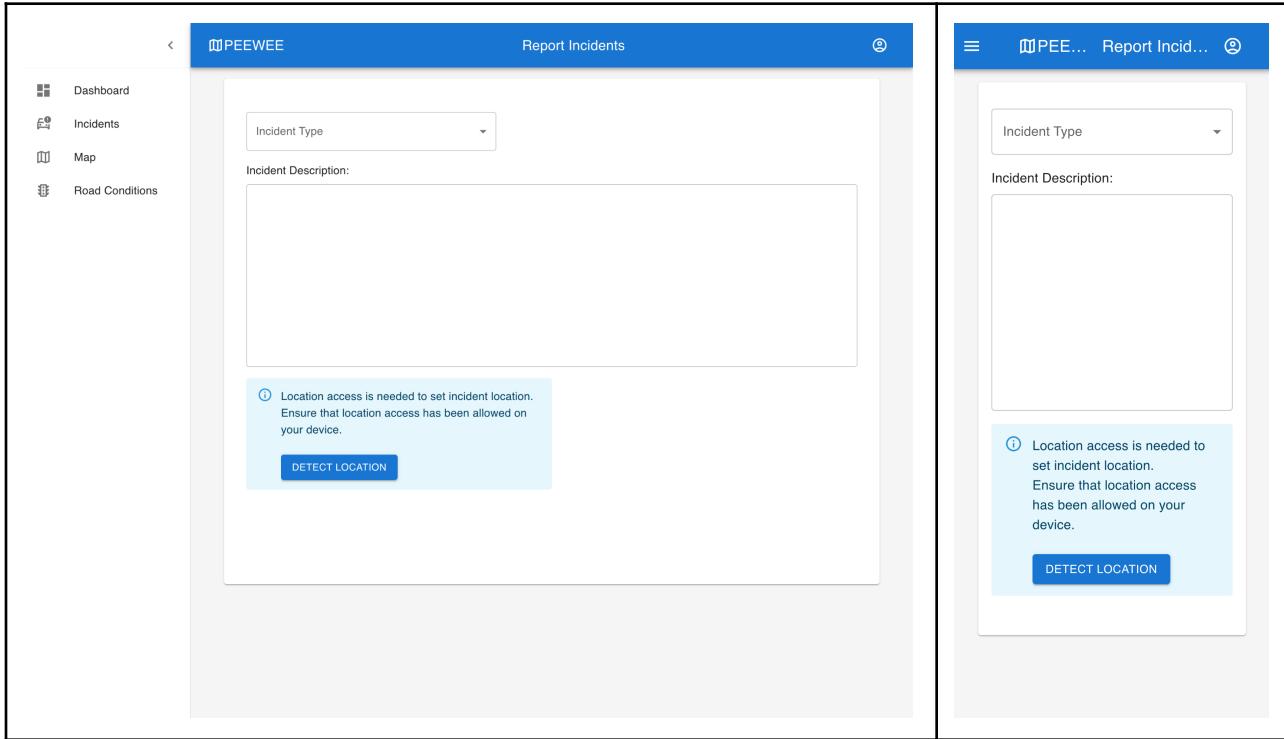
## Incidents



**Figure C2:** Incidents page

The Incidents page contains all available incidents reported by all users across the application. It shows the incident types which include accidents, road works and road closures. It also contains the location of the incident as well as its description. There is an option for the user to report a new incident by clicking on the Report Incident button.

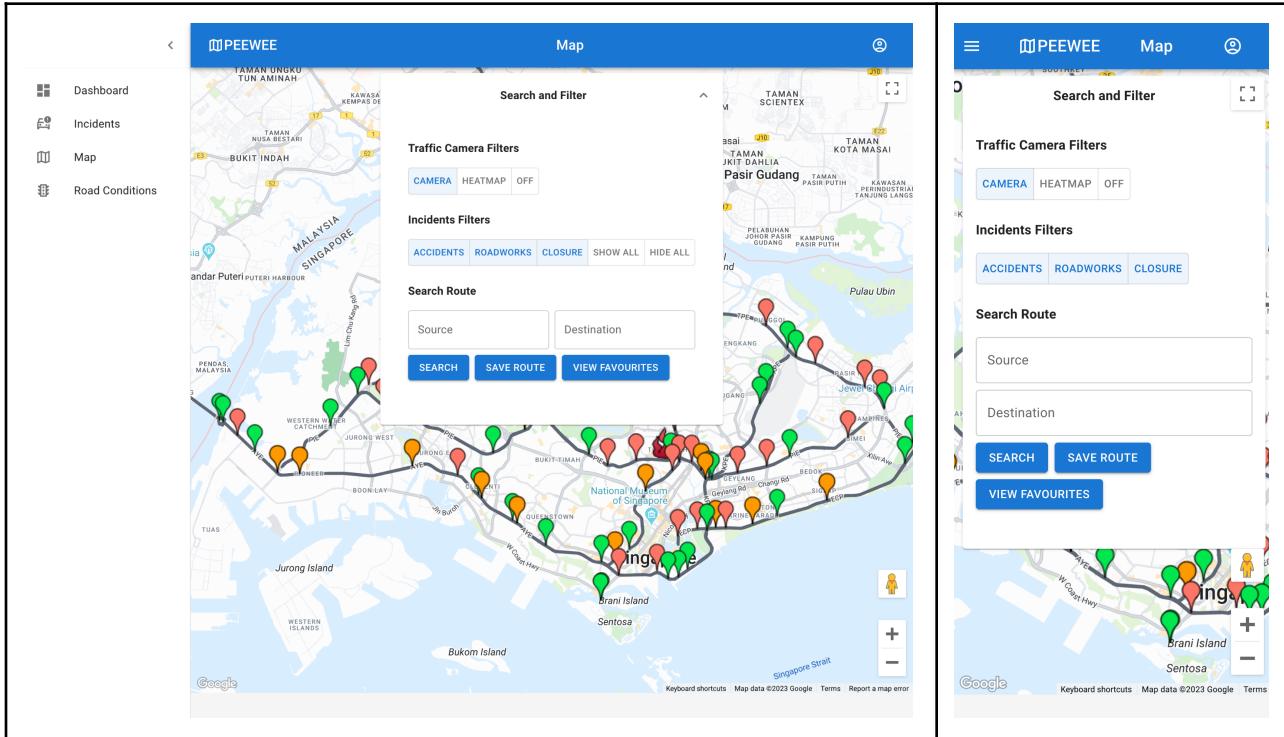
## Report Incidents



**Figure C3:** Report Incidents page

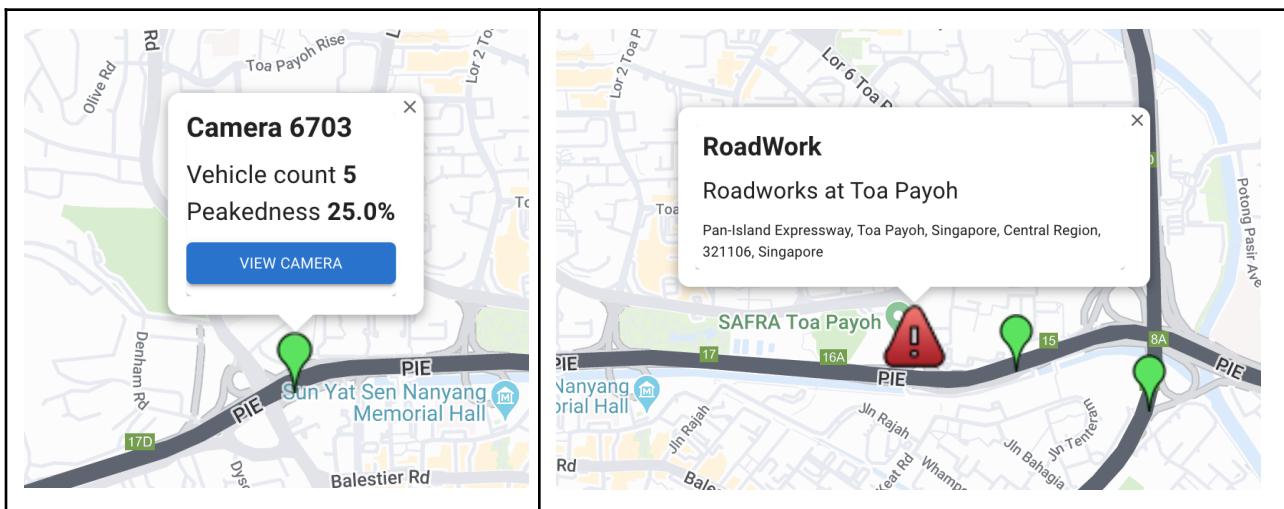
The Report Incidents page allows the user to enter a new incident into the system, alerting all other users on the platform. Information such as the Incident Type, Description, as well as the user's current location are captured. Before capturing the user's location, Peewee will prompt the user to allow location services from their device making it possible to fetch the user's current location. Once the user submits, a new incident will be plotted on the Map.

## Map



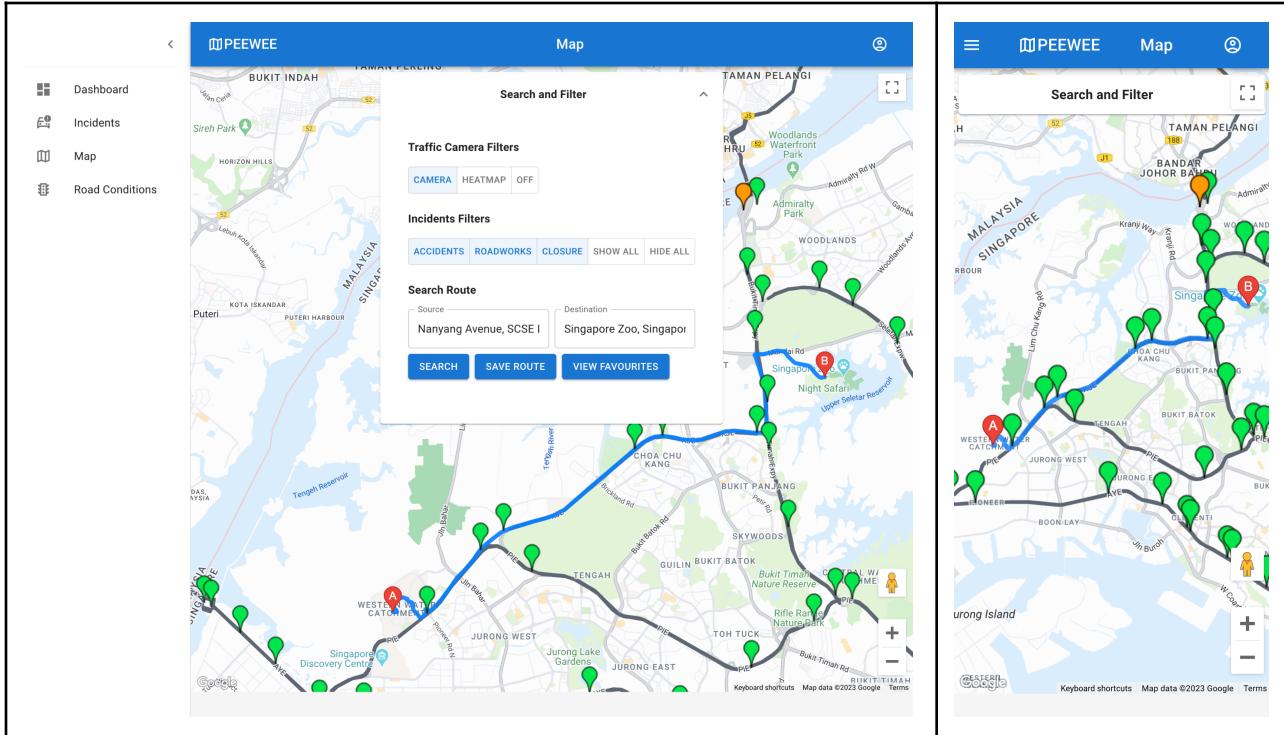
**Figure C4:** Map page

The Map contains the core functionality of our application and thus, is the primary focus of our UI. It provides a fast, intuitive, and dynamic representation of current road conditions visualised by a collection of different map elements. Google Maps API is used to render the mapping platform on our web application due to its fast and data rich features. Live traffic cameras and traffic incidents are plotted on the map. Filtering options are available, to selectively display the individual traffic cameras, detailed heatmap, and different types of incidents available in the system,



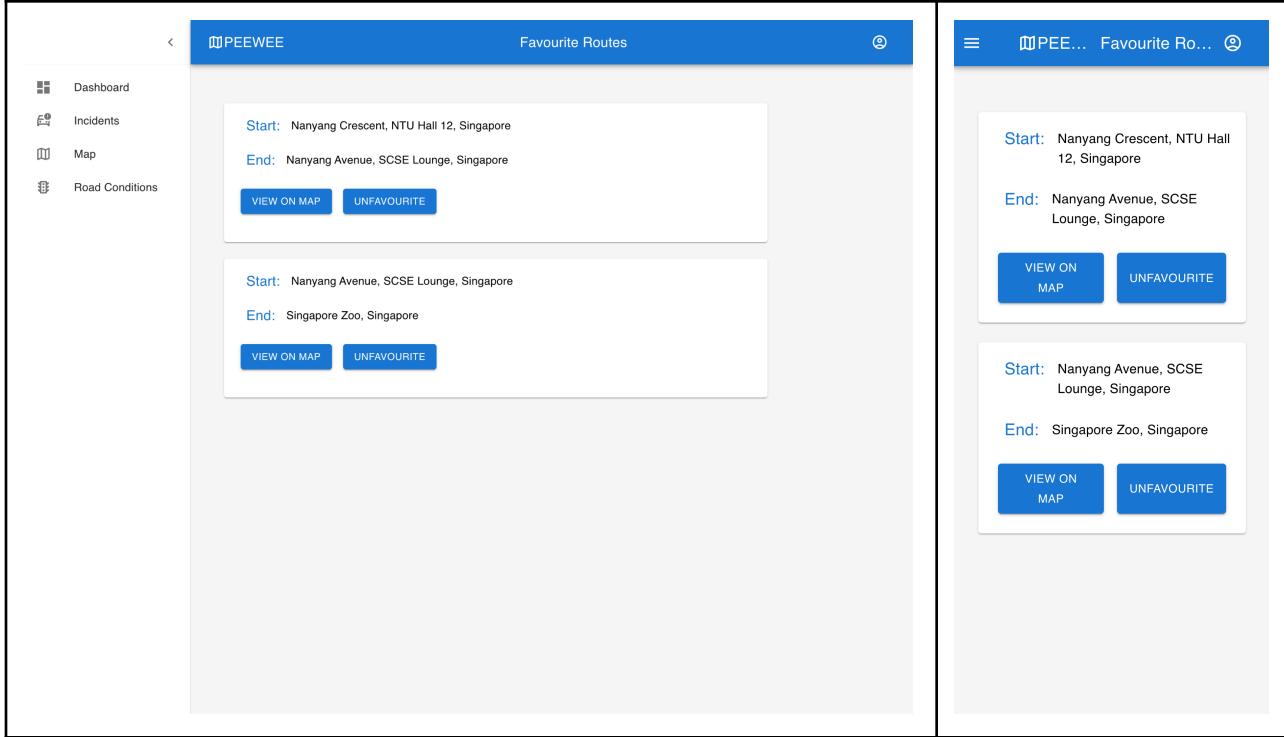
**Figure C5:** Informative Windows displaying information on Map Markers

Upon clicking onto a plotted Map Marker component, an information window will pop up from the Marker to show information on the traffic camera or traffic incident, depending on the type of Marker. The traffic camera InfoWindow has an additional button to view more information on a specific camera by redirecting to the individual camera's Road Conditions page.



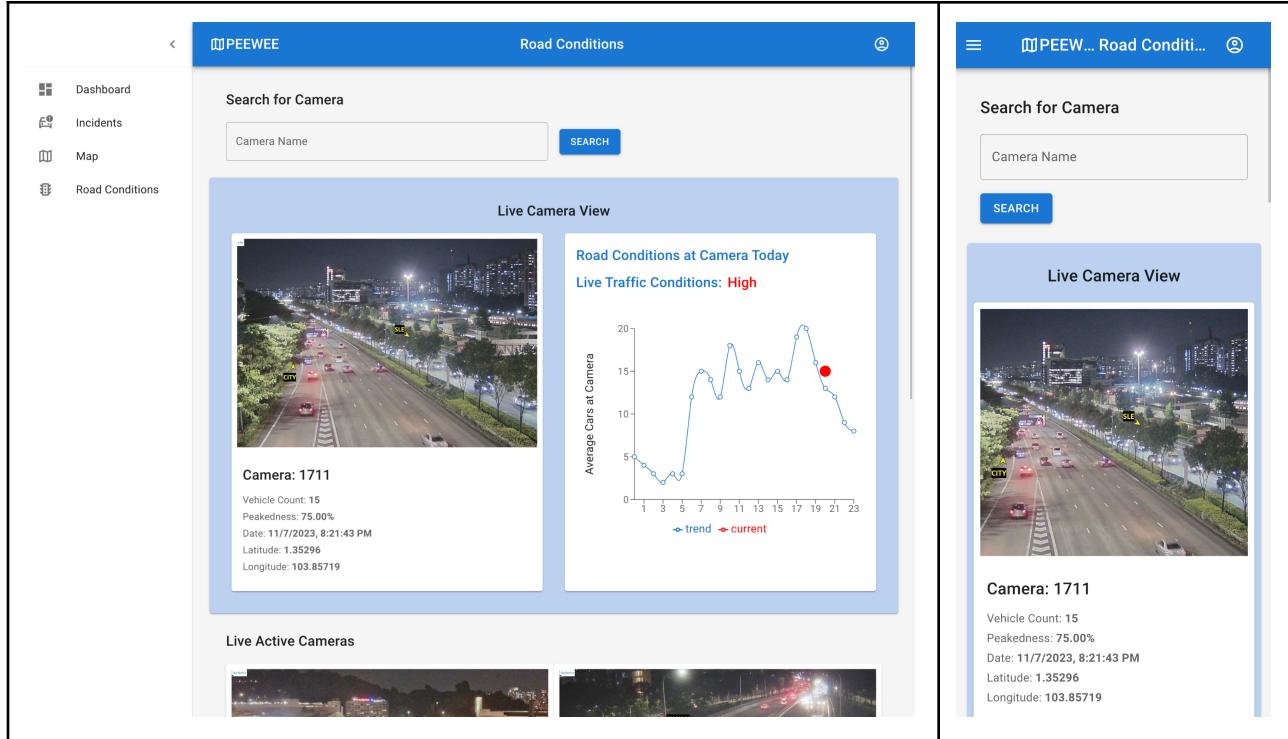
**Figure C6:** Searching for a route on the Map

Routes can be displayed on the Map by searching for a source and destination limited within Singapore. Upon a successful search, the map will hide UI elements and pan to the route being displayed, providing minimal distractions to get the user to their destination. Traffic cameras can be seen along the route, with their Marker colour being an indication of peakedness levels on their journey. Routes can also be saved and viewed in the future for ease of access. A list of the user's favourite routes can be seen upon clicking the “View Favourites” button.



**Figure C7:** A list of favourite routes saved by the user

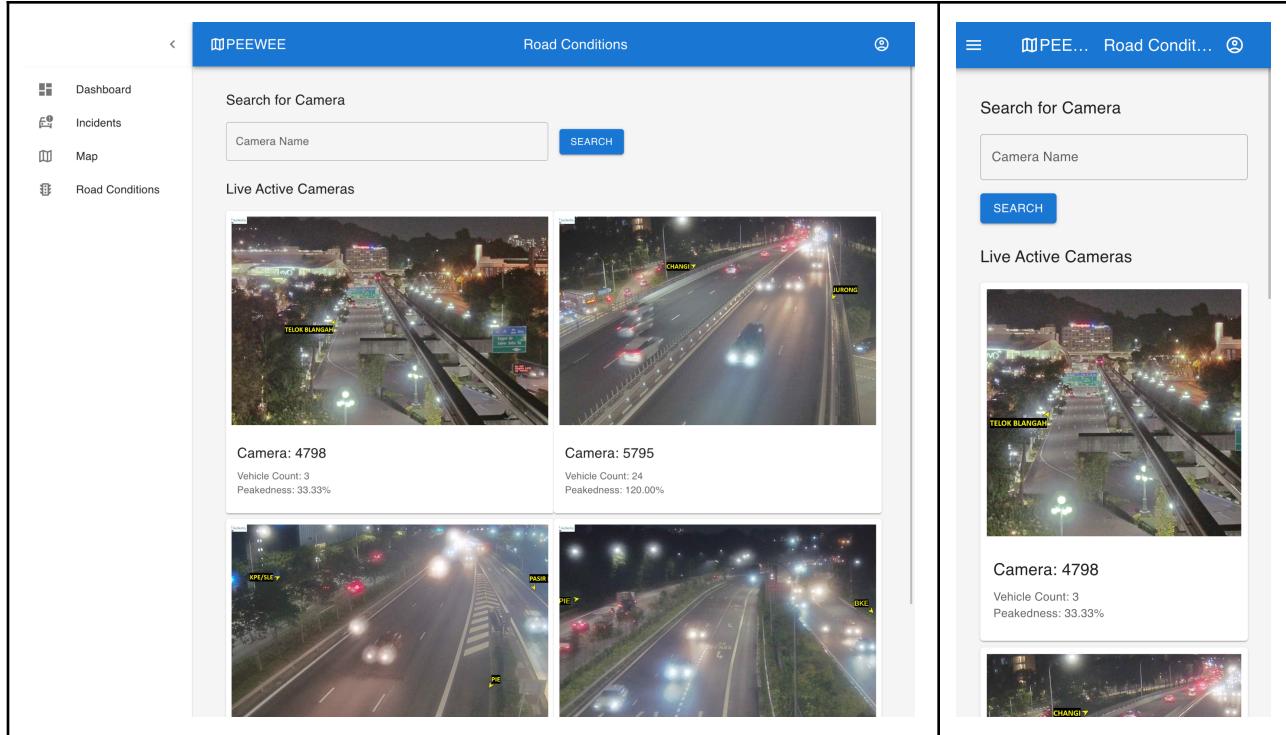
## Individual Road Conditions



**Figure C8:** Individual Road Conditions

Clicking on the “View Camera” button on the Map will redirect the user to the Road Conditions page, where information on an individual is shown. A picture is displayed along with details such as vehicle count, the camera’s peakedness level, date the picture is taken, and location information. Vehicle count traffic trends are populated based on the current camera being displayed, and are specific to that camera. The current vehicle count is also plotted on the trend data, showing the current values compared to the trend.

## Road Conditions



**Figure C9:** Road Conditions

The Road Conditions page will also show a few live views of traffic cameras around Singapore, with their respective vehicle counts and peakedness values. Users can search traffic cameras based on their camera names to view individual camera information identical to the Individual Road Conditions page.