Software Requirements Specification

for

<PEEWEE>

**Version 1.0 approved**

**Prepared by <author>**

**<organization>**

**<date created>**

**Table of Contents**

[**1. Introduction 1**](#_heading=h.1fob9te)

[**1.1 Purpose 1**](#_heading=h.3znysh7)

[**1.2 Document Conventions 1**](#_heading=)

[**1.3 Intended Audience and Reading Suggestions 1**](#_heading=)

[**1.4 Product Scope 1**](#_heading=)

[**1.5 References 1**](#_heading=)

[**2. Overall Description 2**](#_heading=)

[**2.1 Product Perspective 2**](#_heading=h.2s8eyo1)

[**2.2 Product Functions 2**](#_heading=)

[**2.3 User Classes and Characteristics 3**](#_heading=)

[**2.4 Operating Environment 5**](#_heading=)

[**2.5 Design and Implementation Constraints 6**](#_heading=)

[**2.6 User Documentation 7**](#_heading=)

[**2.7 Assumptions and Dependencies 7**](#_heading=)

[**3. External Interface Requirements 8**](#_heading=)

[**3.1 User Interfaces 8**](#_heading=h.2jxsxqh)

[**3.2 Hardware Interfaces 8**](#_heading=)

[**3.3 Software Interfaces 9**](#_heading=)

[**3.4 Communications Interfaces 10**](#_heading=)

[**4. System Features 11**](#_heading=)

[**4.1 Registration 11**](#_heading=)

[**4.2 Sign In 11**](#_heading=)

[**4.3 View Traffic Conditions 12**](#_heading=h.89hmamivgolx)

[**4.4 View Traffic Incidents 13**](#_heading=h.5v6i02678cbi)

[**4.5 Report Traffic Incidents 13**](#_heading=h.6a3lwo9ftfvj)

[**4.6 View Map 14**](#_heading=h.895q1p2vbpgp)

[**4.7 Save and View Routes 14**](#_heading=h.wu15i8vijrsz)

[**5. Other Nonfunctional Requirements 16**](#_heading=h.ngl5fhpd1a2l)

[**5.1 Performance Requirements** 16](#_heading=h.xwhgys52wwke)

[**5.2 Safety Requirements 16**](#_heading=)

[**5.3 Security Requirements 16**](#_heading=)

[**5.4 Software Quality Attributes 16**](#_heading=)

[**5.5 Business Rules 16**](#_heading=)

[**6. Data Dictionary 17**](#_heading=h.1cpi61j1al89)

[**7. Use Cases 18**](#_heading=h.3bcma0dylica)

[**7.1 Use Case Model 18**](#_heading=h.1ru7mcuc5lfb)

[**7.2 Use Case Notations 19**](#_heading=h.f1nr8x4ijhmt)

[**7.3 Use Case Specifications 19**](#_heading=h.5hl81kf13za9)

[**8. Other Requirements 41**](#_heading=)

[**9. System Testing 42**](#_heading=h.7tnslayjd4wf)

[9.1 Black Box Testing 42](#_heading=h.z183mxmpvmer)

[9.2 White Box Testing 46](#_heading=h.t0bvz9drty7a)

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

*<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>*

## Document Conventions

*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

## Intended Audience and Reading Suggestions

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

## Product Scope

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>*

## References

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

# Overall Description

## Product Perspective

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

**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, often 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, through employing the use of advanced predictive algorithms to inform drivers of potential delays. The application feature a user-friendly interface for commuters and communicate with external APIs 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 it completion when the application undergoing through real-world trials, and once successful, it will then receives endorsement from the Land Transport Authority. The approval will mark the readiness of PEEWEE for public release and its contribution to the Smart Nation initiative.

## Product Functions

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>*

**1. User Account Management**

1. Login: Allows the user to securely access their account
2. Register: Enables new user to create new account
3. Forget Password: Assists user in resetting password if forgotten

**2. Traffic Analysis and Display**

1. View Traffic Camera Images: Users can view live images from various traffic cameras to assess current road conditions.
2. Analyse Traffic Images: The system automatically analyzes traffic camera images to detect congestion, incidents, or any abnormalities.
3. 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**

1. Visualize on Map: Users can view their routes on a map, complete with real-time traffic conditions.
2. Create Route: Users can define and save custom routes.
3. Delete Route: Allows users to remove previously saved routes.
4. List Routes: Displays a list of all saved routes for the user.

**4. Incident Reporting and Viewing**

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

## User Classes and Characteristics

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

**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 Travelers:**

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.

## Operating Environment

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>*

**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: Optimized 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: Optimized for Linux server distributions, especially Ubuntu 20.04 and CentOS 8.

Software Components:

Database: Uses PostgreSQL (version 12 and above) for storing user data, routes, and traffic information. It can coexist with other database management systems but requires dedicated resources.

Web Server: Built on the Nginx platform. Compatible with Apache, but dedicated resources are recommended.

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 and Waze, allowing for enhanced route suggestions and data sharing.

## Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

**Corporate or Regulatory Policies:** The PEEWEE system must adhere to GDPR and CCPA data protection and privacy regulations.

**Hardware Limitations:** Due to mobile device variability, the software must be optimized for devices with limited RAM (min 2GB) and storage.

**Interfaces to Other Applications:** The software needs to integrate with Google Maps and Waze for route recommendations.

**Specific Technologies, Tools, and Databases:** The development will utilize PostgreSQL for the database and Node.js for the backend. Any deviation from these technologies requires approval.

**Parallel Operations:** The system should support multitasking to provide real-time updates and simultaneous route recommendations for multiple users.

**Language Requirements:** The primary development language will be JavaScript.

**Communication Protocols:** Data transfer will use the HTTPS protocol (CRUD), ensuring secure communication between the client and server.

**Security Considerations:** Two-factor authentication must be implemented for user logins. All user data must be encrypted both at rest and in transit.

**Design Conventions or Programming Standards:** Development will follow the MVC (Model-View-Controller) design pattern, ensuring separation of concerns.

**Maintenance Responsibility:** Once the PEEWEE system is delivered, the customer's internal IT team will be responsible for its maintenance and updates.

## User Documentation

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

We have decided to place user documentation for consumers to refer to in any case they are lost through github. The github folder would be made accessible to public but only specific files that doesn’t tamper with the data of the PEEWEE app itself.

**User Manuals:** A comprehensive guide detailing all the features and functionalities of the PEEWEE system.

**Online Help:** Context-sensitive help available within the PEEWEE application, guiding users on the feature they are currently using.

**Tutorials:** Video and written guides focusing on key functionalities like setting up route recommendations and integrating with other applications.

**FAQs:** A section in the application and on the official website addressing common queries and issues faced by users.

**Troubleshooting Guides:** A dedicated guide to help users diagnose and solve common technical problems.

**Quick Start Guides:** A brief guide designed for new users to get started with the PEEWEE system quickly.

**Release Notes:** Documented updates for each new version of the PEEWEE system, including new features and bug fixes.

**Documentation Delivery Formats:** All documentation will be available as PDFs for download from the official website. Additionally, Online Help and FAQs will be embedded within the application. Video tutorials would be hosted on the official YouTube channel.

## Assumptions and Dependencies

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

**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 licenses for third-party tools like '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 traffic records.

**Cloud Services:** The project will rely on ‘github’ for backup and data synchronization across devices.

# External Interface Requirements

## User Interfaces

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## Hardware Interfaces

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

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.

## Software Interfaces

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

### 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.

bcrypt - Version 5.1.1 or higher  
bcrypt is a library used to perform hashing on passwords so that they can be stored in our database securely.

### 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

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.

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.

## Communications Interfaces

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

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.

# System Features

## 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 Functional Requirements

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

## 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 Functional Requirements

1. System must allow the user to log in to an existing account
   1. User must enter a valid email address and password
   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.
   1. System must send an 6 digit OTP to the user via the registered email address to reset their password
   2. User must enter a valid OTP to reset password
   3. User must enter a new password that complies with the restrictions on the password complexity outlined in 4.1.3.1.3.1

## 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.

4.2.2 Stimulus/Response Sequences

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

4.2.3 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
   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

## View Traffic Incidents

4.4.1 Description

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

4.2.2 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 Functional Requirement

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

## 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 Functional Requirements

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

## 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 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. System must assess real-time traffic congestion level by analysing real-time traffic images from the Traffic Image API
   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
   1. User must enter a starting point and a destination to create driving routes

## Save and View Routes

4.6.1 Description

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

4.6.2 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.

4.6.3 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

# 

# Nonfunctional Requirements

## Performance Requirements

1. **Real-Time Monitoring**
   1. The system must load pages in under 5 seconds.
   2. The system must detect vehicles in traffic images for congestion analysis with an accuracy of above 70%.
   3. The system should analyse and update traffic conditions every 5 minutes.

1. **Scalability**
   1. The system needs to be able to support at least 100 simultaneous users without decrease in performance.
2. **System Availability**
   1. The system must be compatible with latest versions of web browsers such as Chrome, Firefox, Safari, and Edge.
   2. The system should have an uptime of at least 99.9%.
   3. The system must remain responsive even when external APIs are unresponsive or unavailable.

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

## Safety Requirements

1. Incident Alerts
   1. 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.

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

## Security Requirements

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

1. User Authentication
   1. Users must undergo a secure authentication process before accessing the System.

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

## Software Quality Attributes

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

1. Maintainability
   1. The system’s codebase should be well-documented to facilitate readability.
   2. The system should be modular, facilitating easy maintenance and updates.

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

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

## Business Rules

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

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

# 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. |

# Use Cases

## Use Case Model

## 

## 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

## Use Case Specifications

| Use Case ID: | UC-01 | | |
| --- | --- | --- | --- |
| Use Case Name: | Login | | |
| Created By: | Guang | Last Updated By: | Cheng Yao |
| Date Created: | 05/09/2023 | Date Last Updated: | 25/10/2023 |

# 

| 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: | 1. User provides authentication information such as email address and password 2. User clicks on login 3. The system authenticates User 4. User is brought to the dashboard |
| Alternative Flows: | AF-S1: Credentials are incorrect   1. The system displays an error message indicating an incorrect credentials 2. Flow resets to Step 1 of normal flow   AF-S1: User clicks on forgot password   1. The system redirects User to UC-03 Forgot Password, and User changes to a new password 2. The system brings User back to current flow 3. Flow resets to Step 1 of normal flow   AF-S3: The system fails to authenticate User   1. The system displays an error message indicating the reason of failure 2. Flow resets to Step 1 of normal flow |
| Exceptions: | EX1: Error connecting to server   1. System displays an error message indicating connection is lost 2. Use case ends |
| Includes: | (Extends) UC-03 Forgot Password |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

# 

| 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: | 1. User provides a valid email address 2. User provides a password and confirms the password in the confirm password field 3. The system validates the provided information and registers a new account for User 4. User is redirected to the Dashboard |
| Alternative Flows: | AF-S1: User provides invalid email address   1. The system displays an error alert 2. Flow resets to Step 1 of normal flow   AF-S2: Password and confirm password fields do not match   1. The system displays and error and prompts User to re-enter password 2. Flow resets to Step 1   AF-S2: Entered password does not conform to password policy   1. The system will prompt the User to re-enter a new password, specifying which condition policy has not been satisfied   Flow resets to Step 2  AF-S3: User provides an email address that is already registered   1. The system will alert User 2. The system will prompt User to register with another email address 3. Flow resets to Step 1 |
| Exceptions: | EX1: System is offline and is unable to register a new account   1. The system displays an error message with an error code 2. Use case ends |
| Includes: | (Extends) UC-03 Forgot Password |
| Special Requirements: | For security, passwords must conform to a specific password policy:   1. Minimum length of 8 characters 2. At least 1 uppercase character 3. At least 1 lowercase character 4. At least 1 special character |
| Assumptions: | - |
| Notes and Issues: | - |

### 

# 

| 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: | 1. User provides their email address 2. The system sends an OTP to the User’s email 3. User enters the OTP into the page 4. The system prompts the User for a new password with a password and confirm password field 5. User clicks on change password 6. The system successfully changes User’s password and redirects to the Login use case |
| Alternative Flows: | AF-S1: User provides an invalid email address   1. The system shows an error message and prompts User for a valid email address 2. Flow returns to Step 1 of normal flow   AF-S3: OTP is incorrect or expired   1. The system will prevent the User from changing their password. 2. User can request for a new OTP after a cooldown of 60 seconds, up to 3 times total 3. Flow returns to Step 3 of normal flow   AF-S4: Password and confirm password fields do not match   1. The system displays and error and prompts User to re-enter password 2. User enters the password correctly 3. Flow moves to Step 5 of normal flow   AF-S4: Entered password does not conform to password policy   1. The system will prompt the User to re-enter a new password, specifying which condition policy has not been satisfied 2. Flow returns to Step 4 of normal flow |
| Exceptions: | EX1: OTP entered incorrectly for 3 times   1. The system displays an error message 2. Use case ends |
| Includes: | - |
| Special Requirements: | For security, passwords must conform to a specific requirement:   1. Minimum length of 8 characters 2. At least 1 uppercase character 3. At least 1 lowercase character 4. At least 1 special character   OTPs will expire after 5 minutes |
| Assumptions: | - |
| Notes and Issues: | - |

### 

### 

| Use Case ID: | UC-04 | | |
| --- | --- | --- | --- |
| Use Case Name: | Visualise on Map | | |
| Created By: | Aaron | Last Updated By: | Guang |
| Date Created: | 05/09/2023 | Date Last Updated: | 25/10/2023 |

# 

| Actor: | User |
| --- | --- |
| Description: | * Facilities Users to visually interact with a dynamic map, spotting real-time traffic conditions, traffic cameras, and road incidents. * The map offers an intuitive overlay, giving users a holistic view of the traffic ecosystem, ensuring a seamless navigation experience. |
| Preconditions: | 1. User is authenticated and has access rights. 2. Stable and consistent Internet connection. |
| Postconditions: | 1. Users get a graphical representation of the traffic landscape with key highlights. 2. Relevant traffic markers, including cameras and incidents, are visible to users. |
| Priority: | Essential |
| Frequency of Use: | Regular (before and during commuting.) |
| Flow of Events: | 1. User selects the “Visualise on Map” feature. 2. System does step 1 to 3 in UC-09 normal flow to gather incident reports. 3. System does steps 1 to 3 in UC-11 normal flow to gather the general traffic trends. 4. Application loads a real-time map of the user’s current location or specified area.  * Traffic conditions are color-coded: green for light, orange for medium, and red for heavy. * Markers for traffic cameras and incidents are plotted on the map.  1. User can zoom, pan, and click on individual markers for more detailed information. 2. System plots all incidents when selected. |
| Alternative Flows: | AF-S2: If the map fails to load due to weak connection or API issue:   1. The application advises the user: “Unable to load the map. 2. Please check your connection or try again later”. 3. Return to Step 2 in normal flow.   AF-S2: If the user seeks a region with restricted data:   1. The map displays, but with limited or no data, and notifies: "Information for this area is limited or restricted." 2. Return to Step 2 in normal flow. |
| Exceptions: | EX1: If there's a malfunction with real-time data updating:   1. The system sends an alert: “Some data might be outdated. Proceed with caution.”   EX2: When the application encounters a crash or unexpected failure:   1. User is prompted with: “Sorry for the inconvenience. We’re experiencing technical difficulties. Please retry shortly.” |
| Includes: | UC-11 View Traffic Trends  UC-09 View Reported Incidents |
| Special Requirements: | * Data must be as real-time as possible with refresh intervals not exceeding a few minutes. * The application must handle high-resolution camera feeds without lag. |
| Assumptions: | * The database is constantly updated with fresh traffic, camera, and incident data. * Users have basic knowledge of map interfaces and can understand common traffic symbols. |
| Notes and Issues: | * Ensure consistent uptime for the service to be useful during critical times. * Periodically review and update the map interface to include new roads, landmarks, and traffic management tools. |

| 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: | Allows users to search traffic conditions.   * The feature allows users to search for traffic cameras and traffic incidents within the website. * Data aims to give users a comprehensive understanding of the traffic conditions, easing their efforts in making informed decisions about their desired routes. |
| Preconditions: | * User must be registered and logged into the system. * Internet connectivity is available. |
| Postconditions: | * User is presented with accurate traffic conditions and related statistics for the chosen location or route. * User’s search queries are saved in their search history for future reference. |
| Priority: | High |
| Frequency of Use: | High(High usage during peak commuting hours.) |
| Flow of Events: | **Traffic Search:**   1. User accessing “Traffic Search” feature. 2. System prompts the user to put a specific road, landmark, or search for a traffic camera or reported incident. 3. User input the desired location or criteria. 4. System fetches data and provides information on traffic camera or reported incidents |
| Alternative Flows: | AF-S2: If the user’s input does not match any existing roads or landmarks:   1. The system suggests similar or nearby locations. 2. User can select a suggested location or re-enter their search. 3. User Return to Step 2 in Normal Flow.   AF-S2: If the system cannot retrieve live data to any external issues:   1. The user is notified of the temporary unavailability of data. 2. The system may provide the most recent cached data with a timestamp, if available. 3. User Return to Step 2 in Normal Flow. |
| Exceptions: | EX1: If the user searches for a restricted or private area:   1. The system notifies them: “Data for the selected location is restricted or not available.”   EX2: If the system faces an unexpected internal error:   1. The user is alerted: “An unexpected error occurred. Please try again later.” |
| Includes: |  |
| Special Requirements: | * System ensure data accuracy and update traffic conditions in real-time. * Ensure privacy and security of user search history. |
| Assumptions: | * Traffic data provided is sourced from reliable channels or databases. * Registered users understand basic traffic terms and classifications. |
| Notes and Issues: | * Regularly update the system’s database to include new roads or landmarks. * Monitor for any discrepancies or inconsistencies in traffic data sourced from multiple channels. |

| Use Case ID: | UC-06 | | |
| --- | --- | --- | --- |
| Use Case Name: | Filter Results | | |
| Created By: | Aaron | Last Updated By: | Guang |
| Date Created: | 05/09/2023 | Date Last Updated: | 25/10/2023 |

# 

| Actor: | User |
| --- | --- |
| Description: | Allows users to filter map information.   * By eliminating unwanted incident types and traffic cameras, users can make informed decisions within the system more effectively. |
| Preconditions: | * User must be registered and logged into the system. * Internet connectivity is available. |
| Postconditions: | Traffic conditions on the map are updated to reflect the user’s selected filters.  Users can save filter preferences for future use. |
| Priority: | High |
| Frequency of Use: | High(High usage during peak commuting hours.) |
| Flow of Events: | 1. Accessing Filter Options:  * Once the search results are displayed on the map, a filter icon or button appears on the top or side of the screen. * The user can click on this icon to access filter options.  1. Display of Filter Options:  * A dropdown or slide-out panel appears displaying various filter criteria. * Filters are primarily categorized as "Traffic Cameras" and "Road Incidents." * Under "Road Incidents," options like "Accidents," "Roadwork," and "Closures” are presented.  1. Selecting Filters:  * 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. * Once the desired filters are selected, there's an "Apply" or "Update" button at the bottom of the filter panel.  1. Applying Filters to Map View:  * Clicking Apply, map updates to display the traffic cameras and incidents that match the user’s selected filters. * All irrelevant data points are removed from the map.  1. Clearing and Resetting Filters:  * Option to “Reset” is provided within the panel. * If selected, it restores the map view to the original state, displaying all traffic cameras and incident types.  1. Close the Filter Panel:  * 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. |
| Alternative Flows: | AF-S1: If system faces a delay in updating the map view:   * User is shown a loading indicator with a message “Updating your view, please wait” * System returns to Step 1 in the normal flow. |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: | * System must have an updated database of traffic database and camera locations. * System should be responsive and adaptive to different screen sizes and electronics gadgets. |
| Assumptions: | * Traffic data is frequently updated in the background for the most accurate representation. |
| Notes and Issues: | * Short tutorial would be consider for first-time users being unfamiliar with the filter process. |

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

# 

| Actor: | User |
| --- | --- |
| Description: | Allows users to check traffic conditions. |
| Preconditions: | * User is logged in * User clicks on “Routes” under Map |
| Postconditions: | 1. User exits the app OR 2. User moves to another screen/page |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. User clicks on routes under Map screen 2. System displays the user’s favourite routes. 3. System prompts the user to select a route from their favourited route list or search for a route using Origin and Destination 4. User selects a route from their favourited list 5. System will retrieve route information from Maps API through the Map feature. System will display the traffic conditions along the route and save the searched route in user’s search history 6. If the route has not been favourited, user can click on the unlit favourite icon to save the route 7. If user favourites the route, the system will store it into the user’s favourites list 8. Users can search for another route or click on previously searched routes 9. If user clicks on the exit button, return them to the map page and end the flow |
| Alternative Flows: | AF-S6: If the user’s route is already favourited   1. The System will display a lit up favourite icon 2. Clicking on the favourite icon will unfavourite the route, removing it from their favourites list 3. The System returns to Step 5 in Normal Flow   AF-S3: If the user searches for another route   1. User enters the origin and destination in the map 2. The system verifies the validity of the origin and destination 3. If the origin and destination are valid, the system returns the user to Step 5 in the normal flow 4. If the user searches for a location that has restricted data or is a private data(e.g. SAF restricted zone), system alerts the user: “Information is restricted or not available.” and returns the user to Step 3 in the normal flow |
| Exceptions: | EX1: If there temporary issue or outage with the Maps API:   1. System notifies the user: “There seems to be a temporary issue retrieving data. Please try again in a moment.” 2. Use case ends and the user is returned to UC-04 Visualise on Map |
| 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: | Possibility in minor delays upon fetching real-time data during peak server loads.  Camera data only covers specific locations in a fixed angle, might not be available for all locations. |

| 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: | 4/11/23 |

# 

| 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: | 1. The user selects “Report Incident” in the Traffic Incident page. 2. The system displays the report form and location access message. 3. The user selects the incident type and enters the incident description. 4. The user accepts the system's request to access the user's current location. 5. The system detects and displays the current location detected. 6. The system validates user input and displays a “Submit” button. 7. The user clicks on the “Submit” button. 8. The system saves a record of the time of report submission, incident type, incident location and incident description. 9. The system displays a message on the submission status and the option to submit another form. 10. The user selects to return to the incident page on the navigation bar. |
| Alternative Flows: | AF-S4: If the user rejects system’s request to access current location   1. The system goes back to Step 4 of normal flow   AF-S5: If the system fails to detect the user’s location   1. The system displays a message “Failed to detect location. Redetect current location?” 2. The user selects to redetect the current location. 3. The system goes back to Step 5 of normal flow.   AF-S9: If the system fails to save the incident details   1. The system displays an error message and the option to resubmit form 2. The user selects to resubmit form 3. The system goes back to Step 1 of normal flow |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

# 

# 

# 

# 

| 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: | 1. The user selects “Incidents” in the main menu of the user interface. 2. The system retrieves records of all incidents reported on the day. 3. The system displays all the information on the reported incidents:  * Incident Type * Reported Time * Incident Location * Incident Description |
| Alternative Flows: | AF-S2: If there are no incidents reported on the day   1. The system displays the message “There are no incidents reported today”. 2. The user refreshes the user interface. 3. The system goes back to Step 2 of normal flow   AF-S2: If the system fails to retrieve incident records   1. The system displays a message “Error in loading. Please refresh the page again.” 2. The user refreshes the user interface. 3. The system goes back to Step 2 of normal flow. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

### 

### 

| 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: | 1. User clicks on View Traffic Camera Images in the specific camera page 2. System displays the following information of the camera    * Location    * Traffic Image    * Camera ID    * Timestamp    * Specific Reported Incidents    * Traffic Peakness Level      1. Low      2. Medium      3. High |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | UC-11 View Traffic Trends |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

### 

| 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: | * The system prompts a slide-out panel that shows a graph of the overall traffic trends for Singapore * User can filter by   + Minute (Last Hour)   + Hour (Last Day)   + Day (Last Week) * The graph displays * Count(Number of Vehicles on the rRoad) * Peakness(Count compared to count at other timings) |
| Alternative Flows: | AF-S1: If the user chooses to view the traffic trend of a specific camera on the traffic camera page   1. The system prompts a slide-out panel that shows a overview of the traffic trend of the specific camera 2. Flow returns to Step 2 of normal flow |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

# Design Model

# System Architecture

# Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

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

Expand on AI

Trained on 200 annotated images

Yolov8s model

Example pictures detected vs non detected

Accuracy (metrics graph) meet 70% accuracy

# 

# System Testing

## Black Box Testing

### Register

| Test Case 1 | Valid Registration |
| --- | --- |
| Objective: | To test if a user can successfully register using a valid email and password. |
| Steps: | 1. Navigate to the registration page. 2. Input a valid email address. 3. Input a valid password that adheres to the password policy. 4. Confirm the password in the respective field. 5. Click the "Register" button. |
| Expected Outcome: | The user should be successfully registered and redirected to the Dashboard. |

| Test Case 2 | Invalid Email Format |
| --- | --- |
| Objective: | To test the system's response when an invalid email format is provided. |
| Steps: | 1. Navigate to the registration page. 2. Input an invalid email address (e.g., "user@.com"). 3. Input any password. 4. Confirm the password. 5. Click the "Register" button. |
| Expected Outcome: | The system should display an error alert |

| Test Case 3 | Password and Confirm Password Mismatch |
| --- | --- |
| Objective: | To test the system's response when the passwords provided do not match. |
| Steps: | 1. Navigate to the registration page. 2. Input a valid email address. 3. Input a valid password. 4. Input a different password in the "Confirm Password" field. 5. Click the "Register" button. |
| Expected Outcome: | The system should display an error and prompt the user to re-enter the password |

| Test Case 4 | Password doesn’t meet policy |
| --- | --- |
| Objective: | To test the system's response when the password doesn't conform to the password policy. |
| Steps: | 1. Navigate to the registration page. 2. Input a valid email address. 3. Input a password that doesn't meet the policy (e.g., only 5 characters long). 4. Confirm the password. 5. Click the "Register" button. |
| Expected Outcome: | The system should prompt the user to re-enter a new password, specifying which condition of the policy has not been satisfied |

| Test Case 5 | Register with Already Existing Email |
| --- | --- |
| Objective: | To test the system's response when trying to register with an email that's already in use. |
| Steps: | 1. Navigate to the registration page. 2. Input an email address that has already been registered. 3. Input any valid password. 4. Confirm the password. 5. Click the "Register" button. |
| Expected Outcome: | The system should alert the user and prompt them to register with another email address |

| Test Case 6 | System Offline |
| --- | --- |
| Objective: | To test the system's response when it's offline. |
| Steps: | 1. Ensure the system is offline or simulate an offline scenario. 2. Navigate to the registration page. 3. Attempt to register with any email and password. |
| Expected Outcome: | The system should display an error message with an error code |

### Login

| Test Case 1 | Valid Login |
| --- | --- |
| Input: | Email: test@example.com  Password: testPassword123  //minimum word length, insufficient characters, character restrictions, |
| Expected Output: | User is authenticated and brought to the dashboard. |
| Actual Output: | -filled after testing |
| Status: | PASS/FAIL |

| Test Case 2 | Invalid Email |
| --- | --- |
| Input: | Email: invalid@email.com  Password: ValidPassword123 |
| Expected Output: | System displays an error message indicating incorrect credentials. |
| Actual Output: | -filled after testing |
| Status: | PASS/FAIL |

| Test Case 3 | Valid Login |
| --- | --- |
| Input: | Email: test@email.com  Password: WrongPassword |
| Expected Output: | System displays an error message indicating incorrect credentials. |
| Actual Output: | -filled after testing |
| Status: | PASS/FAIL |

| Test Case 4 | Empty Email and Password Fields |
| --- | --- |
| Input: | Email: ``  Password: `` |
| Expected Output: | System displays an error message indicating fields cannot be empty or similar message. |
| Actual Output: | -filled after testing |
| Status: | PASS/FAIL |

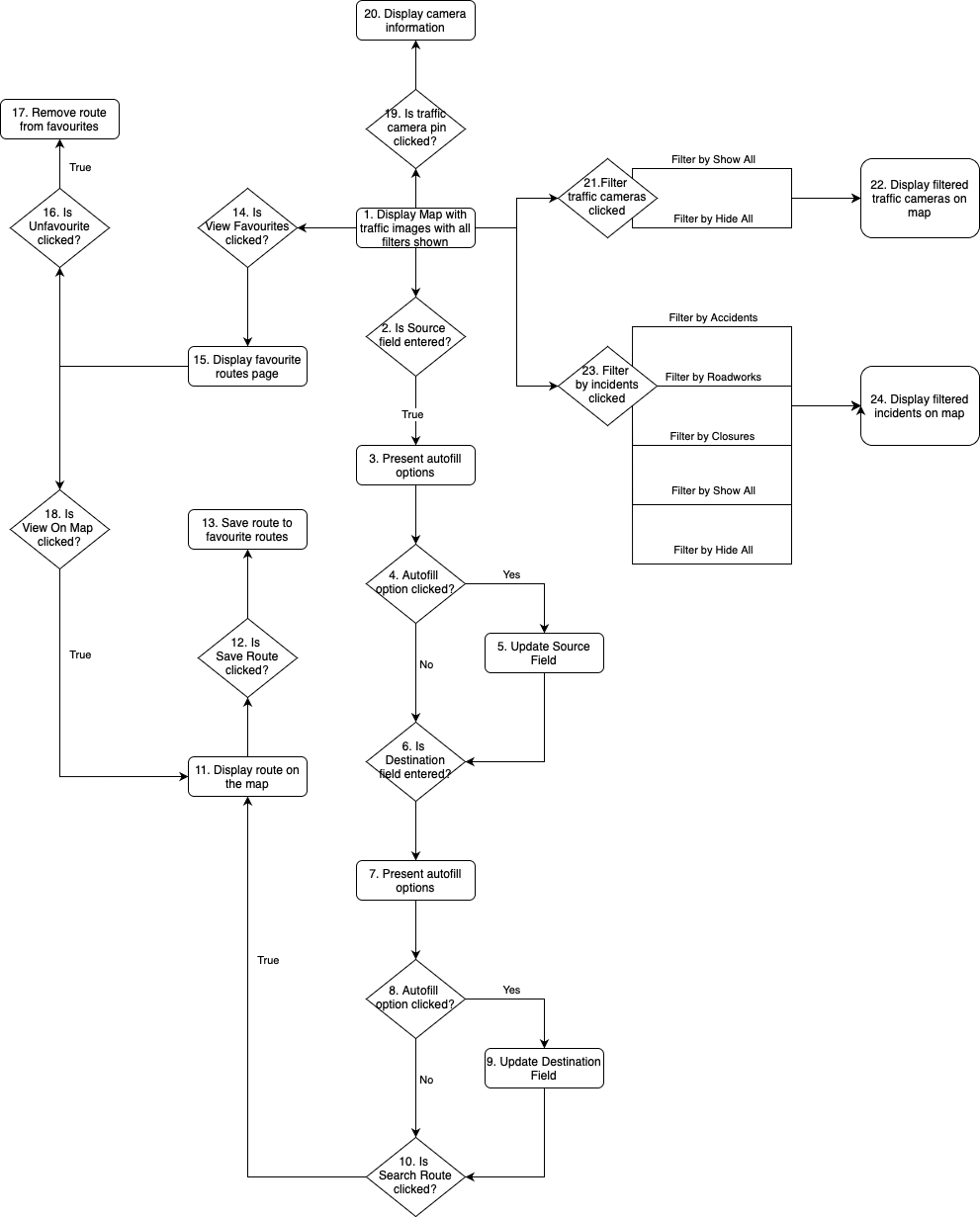
| Test Case 5 | Click 'Forgot Password' |
| --- | --- |
| Action: | Click on "Forgot Password" link. |
| Expected Output: | System redirects to Forgot Password page or flow. |
| Actual Output: | -filled after testing |
| Status: | PASS/FAIL |

| Test Case 6 | Server Connection Error |
| --- | --- |
| Action: | Disconnect from the server or simulate server error and then attempt to login. |
| Expected Output: | User is authenticated and brought to the dashboard. |
| Actual Output: | -filled after testing |
| Status: | PASS/FAIL |

## White Box Testing

### Search Routes

Control Flow Graph



| 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 |
| Source: “Bed”  Click on Bedok autofill  Destination: “Tamp”  Click on Tampines autofill  Click Search Route  Click Save Route | Flow: 1,2,3,4,5,6,7,8,9,10,11,12,13  “Bedok” autofill presented  “Tampines” autofill presented  Source: “Bedok”  Destination: “Tampines”  Route plotted on map | Flow: 1,2,3,4,5,6,7,8,9,10,11,12,13  “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 |
| Source: “Bedok”  Destination: “Tampines”  Click Search Route  Click Save Route | Flow: 1,2,3,4,5,6,7,8,9,10,11,12,13  “Bedok” autofill presented  “Tampines” autofill presented  Source: “Bedok”  Destination: “Tampines”  Route plotted on map | Flow: 1,2,3,4,5,6,7,8,9,10,11,12,13  “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 |
| Source: “Bed”  Destination: “Tamp”  Click Search Route | Flow: 1,2,3,4,5,6,7,8,9,10,11,12,13  Source: “Bedok”  Destination: “Tampines”  Route not plotted on map | Flow: 1,2,3,4,5,6,7,8,9,10,11,12,13  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 View Favourites  Click View on Map | Flow: 1,14,15,18,11  Display Favourite Routes page  Route plotted on map | Flow: 1,14,15,18,11  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 View Favourites  Click Unfavourite | Flow: 1,14,15,16,17  Display Favourite Routes page  Favourite Routes removed | Flow: 1,14,15,16,17  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 on Pin of Camera 6705 | Flow: 1,19,20  Display Traffic Camera Information  “Camera 6705  Vehicle Count: xx  Peakedness: xx%” | Flow: 1,19,20  Display Traffic Camera Information  “Camera 6705  Vehicle Count: 10  Peakedness: 47.6%” |

| Test Case 7: Filter by incidents | | |
| --- | --- | --- |
| Location: “app/src/pages/Map.tsx” | | |
| Inputs | Expected Flow and Outputs | Actual Flow and Outputs |
| Click Hide All  Click Show All  Click Accidents  Click Roadworks  Click Closures | Flow: 1,23,24  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: 1,23,24  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. |

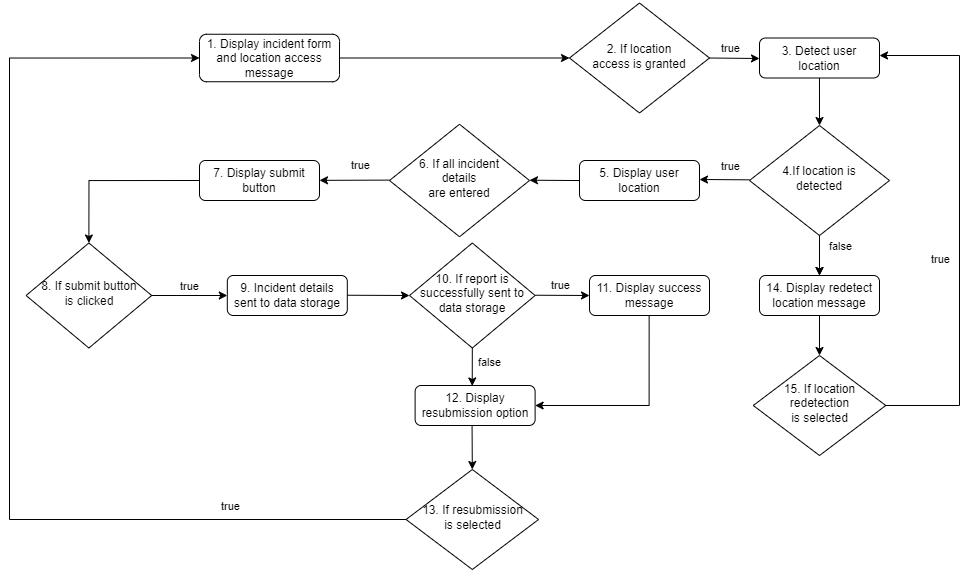
| Test Case 8: Filter by traffic cameras | | |
| --- | --- | --- |
| Location: “app/src/pages/Map.tsx” | | |
| Inputs | Expected Flow and Outputs | Actual Flow and Outputs |
| Click Hide All  Click Show All | Flow: 1,21,22  Hide all traffic cameras on map  Show all traffic cameras on map | Flow: 1,21,22  Hide all traffic cameras on map  Show all traffic cameras on map |

### 

### 

### Report Incident

##### Control Flow Graph



| 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’ | Flow: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12  “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, 12  “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 | Flow: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12  “Incident Location: Rail Corridor” (coordinates: 1.3078, 103.7915)  “Error in report submission” | Flow: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12  “Incident Location: Rail Corridor, Ghim Moh, Queenstown, Central Region, 139350, Singapore” (coordinates: 1.3074, 103.7906)  “Error in report submission” |

**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*

Source: http://www.frontiernet.net/~kwiegers/process\_assets/srs\_template.doc