

Software Requirements Specification



Prepared by: Darren, Shao Qi, Quang, Bryan and Jin Jie

<SC2006 – REP Team 39>

Table Of Contents

Table Of Contents

- 1. Introduction
 - 1.1 Purpose
 - 1.2 Document Conventions
- 1.3 Intended Audience
- 1.4 Product Scope
- 2. Overall Description
 - 2.1 Product Perspective
 - 2.2 Functional Requirements
 - 2.3 Non-functional Requirements
- 3. Data Dictionary
- 4. Complete Use Case Model
 - 4.1. Use Case Diagram
 - 4.2. Use Case Descriptions
- 5. Class Diagrams
- 6. System Architecture
- 7. Design Patterns
- 8. Sequence diagram
- 9. Dialog map
- 10. Implementation
- 11. UI Mockups (Web Application)
- 12. UI Mockups (Android)
- 13. Demo video
- 14 Testing
 - 14.1 Test Cases
 - 14.2 Control Flow Graph
- 15. Other Requirements
 - 15.1 International Requirements
 - 15.2 Legal Requirement
 - 15.3 Reusability

1. Introduction

1.1 Purpose

This document outlines the software requirements for **Road Guardian**. This SRS outlines the functionality, features and constraints of the platform, with the aim to provide detailed and comprehensive framework for development. The scope of this SRS comprises the entire digital system and its sub-system for motorist and administrator. This SRS is integral for guiding the development, testing and maintenance of the digital system, ensuring project objective are met.

1.2 Document Conventions

This SRS follows standard documentation convention to ensure consistent and clarity for the readers. The key convention includes:

- Font style: Arial for body and header
- Highlighting: Bold for emphasis
- Requirement Prioritization: Follow the software development lifecycle from planning phase the cascade down to maintenance phase
- Requirement Identification: Each requirement is uniquely identified for easy reference.

1.3 Intended Audience

- **Project Managers and Stakeholders:** Begin with the Introduction to understand the purpose and scope, then proceed to the Overall Description for a high-level overview of the system
- **Developers:** After the introductory sections, delve into the System Features for detailed descriptions of the platform's functionalities, and the External Interface Requirements for integration specifics.
- **Testers:** Focus on the System Features for test case development and refer to the Nonfunctional Requirements for performance and security testing guidelines.
- **User Experience Designers:** The User Interface sections within the External Interface Requirements will be of particular interest, providing details on the interaction between the system and its users.
- **Technical Writers:** The entire document is relevant, with emphasis on User Documentation and Assumptions and Dependencies to ensure accurate and comprehensive user guides and help documentation.

1.4 Product Scope

Road Guardian is a digital solution designed to reduce road accidents caused by road hazards on the road. Its primary purpose is to detect road hazards and notify motorists of the impending road danger when they are within the range of the detected road hazard. This platform will allow motorists to view all road hazards on the map and provide them ways to report a road hazard.

2. Overall Description

2.1 Product Perspective

This SRS specifies the requirements for Road Guardian to create a safer driving experience. It is not a follow-on member of a product family nor a replacement for existing systems but is a pioneering effort to reduce accidents on roads. The platform functions independently and integrates existing solutions via APIs. The functional and non-functional requirements are provided to identify what a product must do and what its features and functions are, and general properties of a system.

2.2 Functional Requirements

- **Users** must be able to **log in** to the application using their email account
 - The **login system** must **verify the user's login credentials** against its database
 - The **login system** must **grant user access** to the system if credentials match those in its database
 - The **login system** must **deny access and display 'Invalid Credentials' message** if credentials does not match those in its database
 - **Users** must be able to **register an account** for the application with their email address through the **login system**
 - **Users** must be able to **reset their account password** via the **login system**
 - The **register system** must **display invalid email address** if the email address entered exists in the database
 - The **register system** must **display invalid password** when user password does not meet the password security requirement
- Admin must be able to use the **Road Damage Detector** must be able to **extract real time road-side cameras images** from data.gov.sg

- The Road Damage Detection system must analyse the images to determine if there are road damages
 - The Road Damage Detection system must extract the location of the reported road damage from the real time image
 - The Road Damage Detection system must classify the type of detected road damages
 - The Road Damage Detection system must store the picture, location and type of the damage in the system's database
- Road users must be able to view reported road damages around Singapore using the application
 - The application must continuously retrieve data on report road damages from the system's database
 - The application must be able to notify drivers road damages within 150m of their GPS location via notification
 - The application must be able to notify drivers road damages within 150m of their GPS location via a moving map display
- Road users must be able to report road damages using their application
 - Road users must be able to take pictures of road damages via the application
 - Road users must be able to specify the location of the damage manually
 - Road users must be able to specify the location of the damage automatically by GPS
 - Road users must be able to add a brief description of the road damage
- Authorities must be able to monitor the status of each damage report using the system's webpage
 - The webpage must continuously retrieve data on report road damages from the system's database
 - Authorities must be able to update or close the status of a reported damage

2.3 Non-functional Requirements

Android Application
Performance
<ul style="list-style-type: none"> ● The application must load the map and display markers of the location of road damages within 10 seconds
Scalability
<ul style="list-style-type: none"> ● The android application must be able to handle 20 damage reports being made simultaneously

<p>Localization</p> <ul style="list-style-type: none"> • The system must support integration with various local authorities across different region of Singapore
<p>Usability</p> <ul style="list-style-type: none"> • The user interface must be intuitive and easy to use, even for non-technical users • The application must be simple for reporting damages, with minimal steps required
<p>Security</p> <ul style="list-style-type: none"> • Motorist data, including location and login details, must be securely stored and transmitted in the firestore database • The application must protect against unauthorised access and data breaches
<p>Portability</p> <ul style="list-style-type: none"> • The application should be compatible across various android devices and versions • The application code should be well-documented and modular to facilitate future updates and maintenance
<p>Compliance</p> <ul style="list-style-type: none"> • The application must comply with local regulations and data protection laws in Singapore

Web Interface
<p>Performance</p> <ul style="list-style-type: none"> • The map and entries table must load and display within 10 seconds
<p>Scalability</p> <ul style="list-style-type: none"> • The web interface must be able to handle 10 administrators simultaneously
<p>Localization</p> <ul style="list-style-type: none"> • The system must support integration with various local authorities across different region of Singapore
<p>Usability</p> <ul style="list-style-type: none"> • The web interface must be intuitive and easy to use, even for non-technical users • The interface must be simple for adding new cases with minimal steps required • The interface must be simple for editing and closing existing cases with minimal steps required
<p>Security</p> <ul style="list-style-type: none"> • Administrator login details, must be securely stored in the firestore database • The interface must protect against unauthorised access and data breaches
<p>Portability</p> <ul style="list-style-type: none"> • The interface should be compatible across various PCs and browsers • The interface code should be well-documented and modular to facilitate future updates and maintenance
<p>Flexibility</p> <ul style="list-style-type: none"> • The machine learning model should be easy to retrain and update as new data becomes available

Compliance

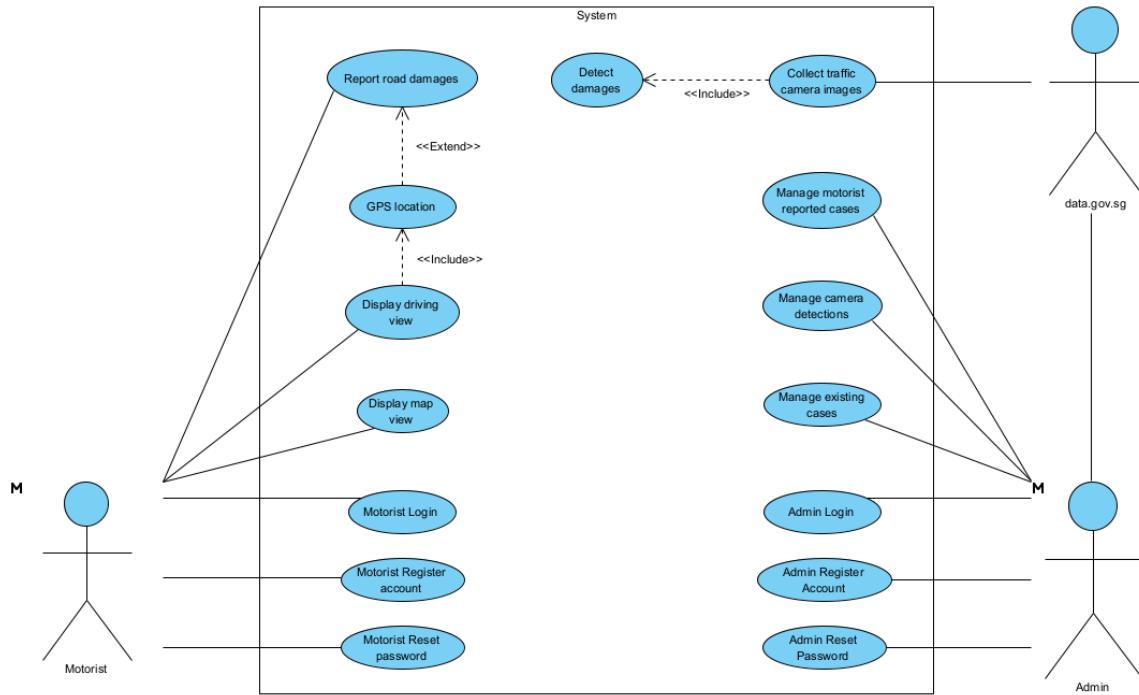
- The application must comply with local regulations and data protection laws in Singapore

3. Data Dictionary

Term	Description
Login System	A backend authentication service hosted by the authorities which verifies user credentials sent by the application via the user database, and also registers new user accounts into the user database.
Road Guardian	A backend service hosted by LTA which processes road images from users and road cameras to detect road damages and determine the severity of detected road damages. If a road damage is detected, the location and severity of the road damage is logged into the database and the system sends an alert to app users about the road damage.
Application	A client-facing application that can be downloaded from app stores. It provides the interface through which road users can interact with the app's features and services. The app allows road users to report road damages, receive alerts of road damages both visually and verbally, and see a GPS map with road damage locations highlighted.
Webpage	A web frontend interface that allows the authorities to monitor and manage the status of road damages. The webpage includes a dashboard that displays damage statistics and a list of damage reports and options to update their status. Access is restricted to authorities only.
Password security requirement	Password must be at least 6 characters (Firebase requirements)
Road Damages	Deterioration or degradation of a road's surface and structure caused by weather conditions, traffic load, poor construction, or lack of maintenance. Pot-holes, cracks and bumps are guidelines used for our classification
Camera Detections/Detections	Images and metadata (Location, Description, Date of Occurrence) of road damages detected by the traffic camera
Motorist Reports/Reports	Images and metadata (Location, Description, Date of Occurrence) of road damages reported by motorist using mobile phones
Cases	Images and metadata (Location, Description, Date of Occurrence, Date of Repair, Status) of road damages logged by Admin from detections/reports database
Motorists	Drivers on the road using their mobile application
Admin	Land & Transport Authority (LTA) - Managing road infrastructures and orders repair for road damages

4. Complete Use Case Model

4.1. Use Case Diagram



4.2. Use Case Descriptions

Motorist Use Cases - MTR

Use Case ID	MTR-001		
Use Case Name	Motorist Login		
Created By	Bryan	Last Updated By	Bryan
Date Created	28/8/24	Date Last Updated	7/11/24

Actor	Motorist
Description	Motorists login to their accounts
Pre Conditions	Motorists must enter a username-password combination that exists in the database
Post Conditions	1. If login succeeds, motorists are brought to the application's menu

	<p>page if login is successful</p> <p>2. If login fails, an error message is displayed, and the application remains on the login page</p>
Frequency of Use	Very often
Flow of Events	<ol style="list-style-type: none"> 1. Motorist opens the system's application 2. Application opens the login page 3. Application prompts motorist for his/her username and password 4. Motorist enter his/her username - password combination 5. Motorist clicks the "Login" button 6. The application verifies the username - password combination in its motorist database 7. If the combination exists in the database, the application opens the menu page
Alternative Flows	<ol style="list-style-type: none"> 1. If the username - password combination does not exist in the database, the application notifies this to the motorist via an error message 2. The application remains at the login screen
Exceptions	<p>Case 1: Multiple attempt failed</p> <ol style="list-style-type: none"> 1. After five failed attempts, the system locks the motorist out for five minutes. 2. The application will prompt "Press 'Forgot Password' if you have forgotten"
Includes	NIL
Extends	NIL
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

Use Case ID	MTR-002		
Use Case Name	Motorist Register Account		
Created By	Quang	Last Updated By	Bryan
Date Created	28/8/24	Date Last Updated	7/11/24

Actor	Motorist	
Description	Motorist register an account with the system	
Pre Conditions	<ol style="list-style-type: none"> 1. Motorist must have a valid email address 2. Motorist must not have an account with the system 	

Post Conditions	<ol style="list-style-type: none"> 1. If username and email do not exist in the system, the new username, password and email combination will be saved in the motorist's database 2. If username/email exists in the system, the system will display "Username exists" or "Email exists" to the motorist respectively
Frequency of Use	Rarely
Flow of Events	<ol style="list-style-type: none"> 1. Motorist presses 'Register here' button and app opens registration page 2. The system prompts the motorist for his/her email, username, password and to confirm password. 3. Motorist enter his/her email, username, password and same password for confirm password. 4. Motorist clicks the "Register" button 5. The system validates the registration information provided by the motorist 6. The email, username, password are saved into the motorist database 7. The application displays "Registration successful, please login."
Alternative Flows	<p>Case 1: Duplicate email/username</p> <ol style="list-style-type: none"> 1. If the email or username exists in the system's database, the application displays "Username/Email exists" 2. The application returns to Step 2 <p>Case 2: Invalid registration details</p> <ol style="list-style-type: none"> 1. If the format of email address or password entered is invalid, the application displays "Invalid email format"/"Password does not meet security requirements" respectively 2. The application returns to Step 2 <p>Case 3: Confirm password input is not the same as Password</p> <ol style="list-style-type: none"> 1. 'Register' button is not clickable until they are the same
Exceptions	NIL
Includes	NIL
Extends	NIL
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

Use Case ID	MTR-003		
Use Case Name	Motorist Forgot Password		
Created By	Bryan	Last Updated By	Bryan
Date Created	28/8/24	Date Last Updated	7/11/24

Actor	Motorist
Description	Motorist reset their account password in the system's database
Pre Conditions	Motorist must enter a username that exists in the database
Post Conditions	Motorist's new password is updated in the database
Frequency of Use	Rarely
Flow of Events	<ol style="list-style-type: none"> 1. Motorist navigates to the password reset page 2. Application prompts motorists for their email address 3. Motorist enter their email address. 4. Application verifies email address. 5. If the email exists in the database, an email is sent to the email address with a link to reset their password 6. Motorist clicks the link and is brought to a page to enter their new password and confirm password 7. Motorist enters a valid password and same password for confirm password. 8. Password is saved to database.
Alternative Flows	<p>Case 1: Email address does not exist in the database</p> <ol style="list-style-type: none"> 1. 'User does not exist' message is shown and app remains in Forget Password page. <p>Case 2: Invalid password</p> <ol style="list-style-type: none"> 1. 'Password does not meet security requirements' is shown <p>Case 3: Confirm password input is not the same as Password</p> <ol style="list-style-type: none"> 1. 'Change Password' button is not clickable.
Exceptions	NIL
Includes	NIL
Extends	NIL
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

Use Case ID	MTR-004		
Use Case Name	Display Map View		
Created By	Quang	Last Updated By	Bryan
Date Created	28/8/24	Date Last Updated	7/11/24

Actor	Motorist
Description	Display locations of reported road damage on application's map
Pre Conditions	1. Motorist must be logged into the application
Post Conditions	Motorists see locations of report road damages on their application's map
Frequency of Use	Very Often
Flow of Events	<ul style="list-style-type: none"> 1. Motorist open application 2. Application extracts the map of Singapore from its map reader 3. Application displays the extracted map on motorists' screen 4. Application retrieves all reported road damages from its database 5. Application overlays the location of the reported damages on its map
Alternative Flows	NIL
Exceptions	NIL
Includes	NIL
Extends	NIL
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

Use Case ID	MTR-005		
Use Case Name	Display Driving View		
Created By	Jin Jie	Last Updated By	Bryan
Date Created	28/8/24	Date Last Updated	7/11/24

Actor	Motorists	
Description	Display location of report road damages on the application's moving map	
Pre Conditions	<ul style="list-style-type: none"> 1. Motorist must be logged into the application 2. Motorist must select "Navigation" 	
Post Conditions	Motorists see location of reported road damages ahead of their path on their moving map	
Frequency of Use	Very Often	
Flow of Events	<ul style="list-style-type: none"> 1. Application reads the current GPS location of the motorist 2. Application extracts partial map of 500 feet radius based on the 	

	<p>motorist's location</p> <ol style="list-style-type: none"> 3. Application measures the compass heading of the motorist 4. Application plots an arrow on the partial map denoting the location and direction of movement of the motorist 5. Application retrieves the location, classification and status of reported road damages within 500 feet radius of the motorist from its database 6. Application overlays the location of the reported damages on the partial map 7. Application plays notification to alert the motorist if the road damage is within 500 feet of motorist 8. Application repeats from Step 1 every 5 seconds
Alternative Flows	NIL
Exceptions	NIL
Includes	NIL
Extends	NIL
Special Requirements	NIL
Assumptions	<ol style="list-style-type: none"> 1. Motorist's phone does not have notifications muted or off
Notes and Issues	NIL

Use Case ID	MTR-006		
Use Case Name	Report Road Damage		
Created By	Jin Jie	Last Updated By	Jin Jie
Date Created	1/9/24	Date Last Updated	7/11/24

Actor	Motorists
Description	Motorists report road damages spotted
Pre Conditions	<ol style="list-style-type: none"> 1. Motorists must be logged into the application 2. Motorists must select "Damage Reporting" page
Post Conditions	The newly reported road damage is added to its database as unlogged for verification by admin
Frequency of Use	Very Often
Flow of Events	<ol style="list-style-type: none"> 1. Motorists select "Take Photo" 2. Application opens the camera function 3. Motorists take picture of the road damage they spot 4. Application prompts motorists to set type of hazard and enter the location of the damage 5. Motorists enter the location of the road damage manually or by pressing the "GPS" icon

	<p>6. (Optional) Motorists add a brief description of the road damage</p> <p>7. Motorists click "Send"</p> <p>8. Application adds the case as unlogged into the case database</p> <p>9. Application returns to the Menu page</p>
Alternative Flows	<p>Case 1: Location input is outside of Singapore</p> <ol style="list-style-type: none"> 1. "Location must be within Singapore" is displayed 2. Application remains at ReportHazard page <p>Case 2: Motorist cancels report</p> <ol style="list-style-type: none"> 1. Motorist presses "Cancel" 2. Application returns to "Menu View"
Exceptions	No location entered before clicking "Send" <ol style="list-style-type: none"> 1. Error message "Please enter location" displayed 2. Application remains in ReportHazard page
Includes	Classify damages
Extends	GPS location
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

Admin Use Cases - ADM

Use Case ID	ADM-001		
Use Case Name	Admin Login		
Created By	Darren	Last Updated By	Darren
Date Created	1/9/24	Date Last Updated	7/11/24

Actor	Admin
Description	Admin login to their accounts
Pre Conditions	Admin must enter a username-password combination that exists in the database
Post Conditions	<ol style="list-style-type: none"> 1. If login succeeds, admin will be brought to the application's menu page if login is successful 2. If login fails, an error message is displayed, and the application remains on the login page
Frequency of Use	Very often
Flow of Events	<ol style="list-style-type: none"> 1. Admin opens the system's application

	<ol style="list-style-type: none"> 2. Application opens the login page 3. Application prompts admin for his/her username and password 4. Admin enter his/her username - password combination 5. Admin clicks the “Login” button 6. The application verifies the username - password combination in its system database 7. If the combination exists in the database, the application opens the menu page
Alternative Flows	<p>ALT 7: Invalid credentials below 5 attempts</p> <ol style="list-style-type: none"> 1. If the email- password combination does not exist in the database, the webpage displays “Invalid Credentials - X”, where X is the number of failed attempts. 2. The webpage returns to step 3
Exceptions	<p>Case 1: Admin forgot password</p> <ol style="list-style-type: none"> 1. Admin click on “Reset Password” 2. The webpage goes to Use Case ADM-003 <p>Case 2: Multiple attempt above 5 attempts</p> <ol style="list-style-type: none"> 1. After five failed login attempts, the system locks the admin out for one minute. 2. The application will prompt “Press ‘Forgot Password’ if you have forgotten” <p>Case 3: Admin has no account with the system</p> <ol style="list-style-type: none"> 1. Admin clicks the “Register Account” button in the login page 2. The webpage goes to Use Case ADM-002
Includes	NIL
Extends	NIL
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

Use Case ID	ADM-002		
Use Case Name	Admin Register Account		
Created By	Darren	Last Updated By	Darren
Date Created	1/9/24	Date Last Updated	7/11/24

Actor	Admin	
Description	Admin register an account with the system	
Pre Conditions	<ol style="list-style-type: none"> 1. Admin must have an valid email address 2. Admin must not have an account with the system 	

Post Conditions	1. The admin's email and password is stored in the admin's database
Frequency of Use	Rarely
Flow of Events	<ol style="list-style-type: none"> 1. Admin navigates to the registration page 2. The system prompts the admin for his/her email and password 3. Admin enter his/her email, username 4. Admin clicks the "Register" button 5. The system validates the registration information provided by the admin 6. The email, password are saved into the admin database 7. The application displays "Registration successful, please login."
Alternative Flows	<p>ALT 4a: Duplicate email</p> <ol style="list-style-type: none"> 1. If the email or username exists in the admin's database, the webpage displays "Email exists" 2. The webpage returns to Step 2 <p>ALT 4b: Invalid email</p> <ol style="list-style-type: none"> 1. If the format of email address is invalid, the webpage displays "Please include a @ in the email" 2. The webpage returns to Step 2
Exceptions	NIL
Includes	NIL
Extends	NIL
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

Use Case ID	ADM-003		
Use Case Name	Admin Reset Password		
Created By	Darren	Last Updated By	Darren
Date Created	1/9/24	Date Last Updated	7/11/24

Actor	Admin
Description	Admin reset their account password in the system's database
Pre Conditions	Admin must enter a username that exists in the database
Post Conditions	Admin new password is updated in the database
Frequency of Use	Rarely

Flow of Events	<ol style="list-style-type: none"> 1. Admin navigates to the password reset page 2. Application prompts admin for their email address 3. Admin enter their email address. 4. Application verifies email address. 5. If the email exists in the database, the webpage sends a password reset link to the admin <ol style="list-style-type: none"> a. The webpage displays "Please check inbox" 6. Admin clicks on the email link in his/her inbox 7. The link opens a new browser that prompts the admin for new password 8. Admin input a new password. 9. Application updates the new password in the admin database.
Alternative Flows	<p>ALT 8:</p> <ol style="list-style-type: none"> 1. If the password is less than 6 characters, display "The password must be at least 6 characters long". 2. Repeat step 8
Exceptions	NIL
Includes	NIL
Extends	NIL
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

Use Case ID	ADM-004		
Use Case Name	Manage existing cases		
Created By	Shao Qi	Last Updated By	Shao Qi
Date Created	1/9/24	Date Last Updated	7/11/24

Actor	Admin
Description	Manage logged cases of road damages in Singapore
Pre Conditions	<ol style="list-style-type: none"> 1. Admin must be logged into the webpage 2. Admin must select the Pending Action button
Post Conditions	Admin sees the list of logged cases of road damages and their details
Frequency of Use	Very Often
Flow of Events	<ol style="list-style-type: none"> 1. Webpage displays a list of logged cases with their location, description, date of occurrence, date of repair and status on the bottom table display 2. Webpage display markers specifying the location of cases on the top

	<p>left map display</p> <ol style="list-style-type: none"> 3. Admin selects a specific case 4. Webpage displays the picture of the damage on the top right display 5. Admin set the date of repair via the date field input box 6. Admin changes the status of the case to 'Repair Completed' via the status dropdown box 7. The website updates the status of the case in the case database 8. Admin repeats from step 3
Alternative Flows	<p>ALT 3:</p> <ol style="list-style-type: none"> 1. Admin selects a marker specifying the location of the case on the map 2. Website display the image of the damage on the top right display <p>ALT 6:</p> <ol style="list-style-type: none"> 1. Admin changes the status of the case to 'Delete' via the status dropdown box 2. The website deletes the case from the case database
Exceptions	NIL
Includes	NIL
Extends	NIL
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

Use Case ID	ADM-005		
Use Case Name	Manage camera detections		
Created By	Shao Qi	Last Updated By	Shao Qi
Date Created	1/9/24	Date Last Updated	7/11/24

Actor	Admin
Description	Manage camera detection of road damages in Singapore
Pre Conditions	<ol style="list-style-type: none"> 1. Admin must be logged into the webpage 2. Admin must select the Camera Detections button
Post Conditions	Admin sees the list of camera detection of road damages and their details
Frequency of Use	Very Often
Flow of Events	<ol style="list-style-type: none"> 1. Webpage displays a list of camera detections with their location, description, date of occurrence and action on the bottom table display 2. Webpage display markers specifying the location of cases on the top

	<p>left map display</p> <ol style="list-style-type: none"> 3. Admin selects a specific detection 4. Webpage displays the picture of the damage on the top right display 5. Admin changes the action of the row to 'Add Case' via the action dropdown box 6. The website moves the detection from the detection database to the case database 7. Admin repeats from step 3
Alternative Flows	<p>ALT 3:</p> <ol style="list-style-type: none"> 1. Admin selects a marker specifying the location of the case on the map 2. Website display the image of the damage on the top right display <p>ALT 5:</p> <ol style="list-style-type: none"> 1. Admin changes the action of the row to 'Dismiss' via the action drop down box 2. The website deletes the detection from the detections database
Exceptions	NIL
Includes	NIL
Extends	NIL
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

Use Case ID	ADM-006		
Use Case Name	Manage motorist reports		
Created By	Shao Qi	Last Updated By	Shao Qi
Date Created	1/9/24	Date Last Updated	7/11/24

Actor	Admin
Description	Manage motorist reports of road damages in Singapore
Pre Conditions	<ol style="list-style-type: none"> 1. Admin must be logged into the webpage 2. Admin must select the Motorist Report button
Post Conditions	Admin sees the list of motorist reports of road damages and their details
Frequency of Use	Very Often
Flow of Events	<ol style="list-style-type: none"> 1. Webpage displays a list of motorist reports with their location, description, date of occurrence and action on the bottom table display 2. Webpage display markers specifying the location of cases on the top

	<p>left map display</p> <ol style="list-style-type: none"> 3. Admin selects a specific report 4. Webpage displays the picture of the damage on the top right display 5. Admin changes the action of the row to 'Add Case' via the action dropdown box 6. The website moves the report from the report database to the case database 7. Admin repeats from step 3
Alternative Flows	<p>ALT 3:</p> <ol style="list-style-type: none"> 1. Admin selects a marker specifying the location of the case on the map 2. Website display the image of the damage on the top right display <p>ALT 5:</p> <ol style="list-style-type: none"> 1. Admin changes the action of the row to 'Dismiss' via the action dropdown box 2. The website deletes the report from the reports database
Exceptions	NIL
Includes	NIL
Extends	NIL
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

Use Case ID	ADM-007		
Use Case Name	Collect traffic images		
Created By	Shao Qi	Last Updated By	Shao Qi
Date Created	1/9/24	Date Last Updated	7/11/24

Actor	Admin
Description	Collect traffic images from data.gov.sg
Pre Conditions	<ol style="list-style-type: none"> 1. Admin must be logged into the webpage 2. Admin must select the Extract Camera Images button
Post Conditions	Images are extracted from data.gov.sg for road damage detection
Frequency of Use	Very Often
Flow of Events	<ol style="list-style-type: none"> 1. Webpage runs a python script to extract the image from data.gov.sg

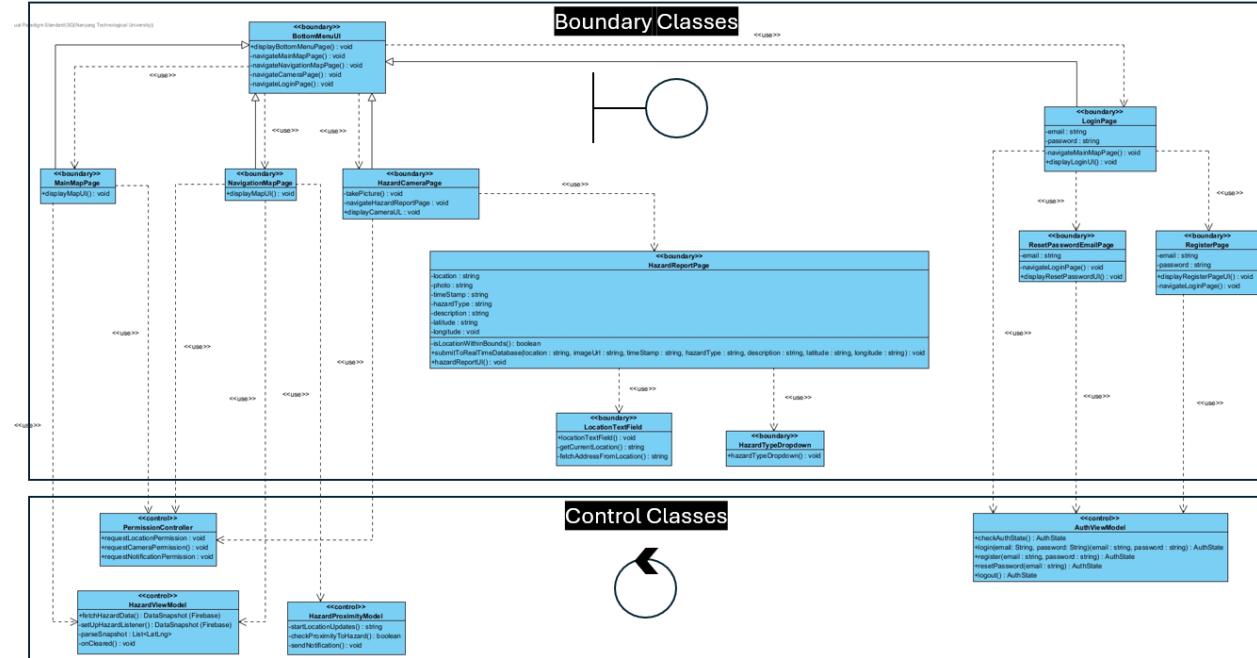
Alternative Flows	NIL
Exceptions	NIL
Includes	Detect Damages
Extends	NIL
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

Use Case ID	ADM-008		
Use Case Name	Detect Damages		
Created By	Shao Qi	Last Updated By	Shao Qi
Date Created	1/9/24	Date Last Updated	7/11/24

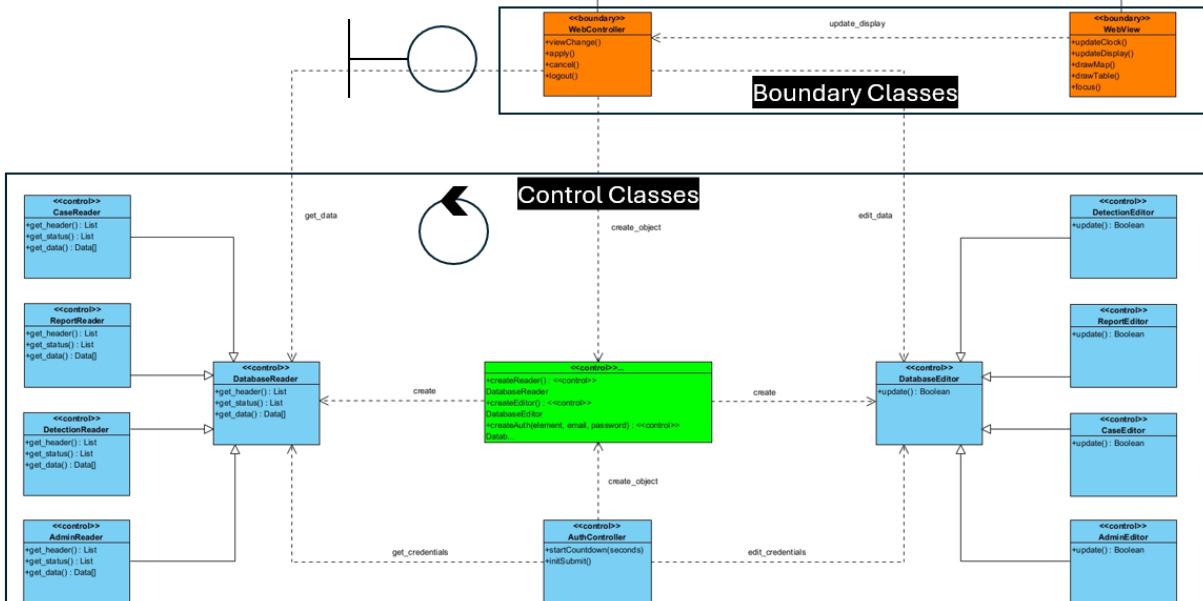
Actor	Admin
Description	Detect road damages from traffic images collected from data.gov.sg
Pre Conditions	Admin must click the 'Extract Camera Images' button
Post Conditions	Images of detect road damages are stored in the detections database
Frequency of Use	Very Often
Flow of Events	<ol style="list-style-type: none"> 1. Webpage runs a python script to detect road damages from images collected from data.gov.sg 2. The python script stores data and images of road damages in the detections database
Alternative Flows	NIL
Exceptions	NIL
Includes	NIL
Extends	NIL
Special Requirements	NIL
Assumptions	NIL
Notes and Issues	NIL

5. Class Diagrams

After several iterations we've deemed Entity Classes unnecessary in our design, as we've adopted Firebase Realtime database to store our data.



Key Boundary-Control classes (Android)



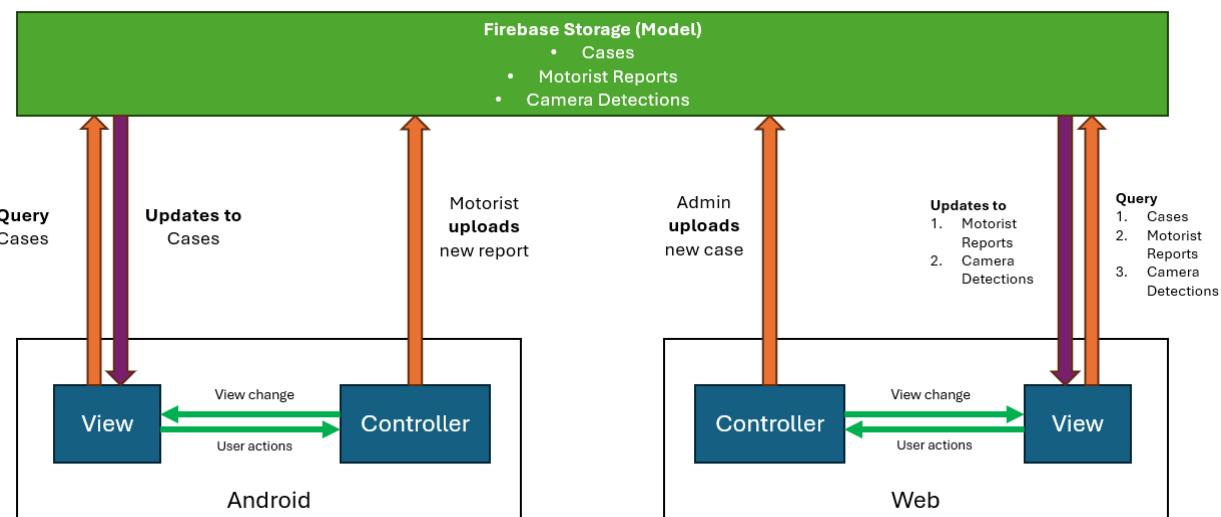
Key Boundary-Control classes (Web)

The **Single Responsibility Principle** is adopted in our class design. Each class has one and only one responsibility (i.e. to read a certain type of data in DatabaseReader, to manage

authentication control in AuthController). This reduces coupling between components and protects classes from unintended changes when parts of other code changes. It also makes code easier to understand and debug.

Our class design also follows the **Open-Closed Principle**. As such, new functions can be added without changing existing code. For instance, a subclass of DatabaseReader can be implemented to read a new type of data, without affecting DatabaseReader and the context that calls it (Although the new type of data needs to be specified in ObjectFactory).

6. System Architecture

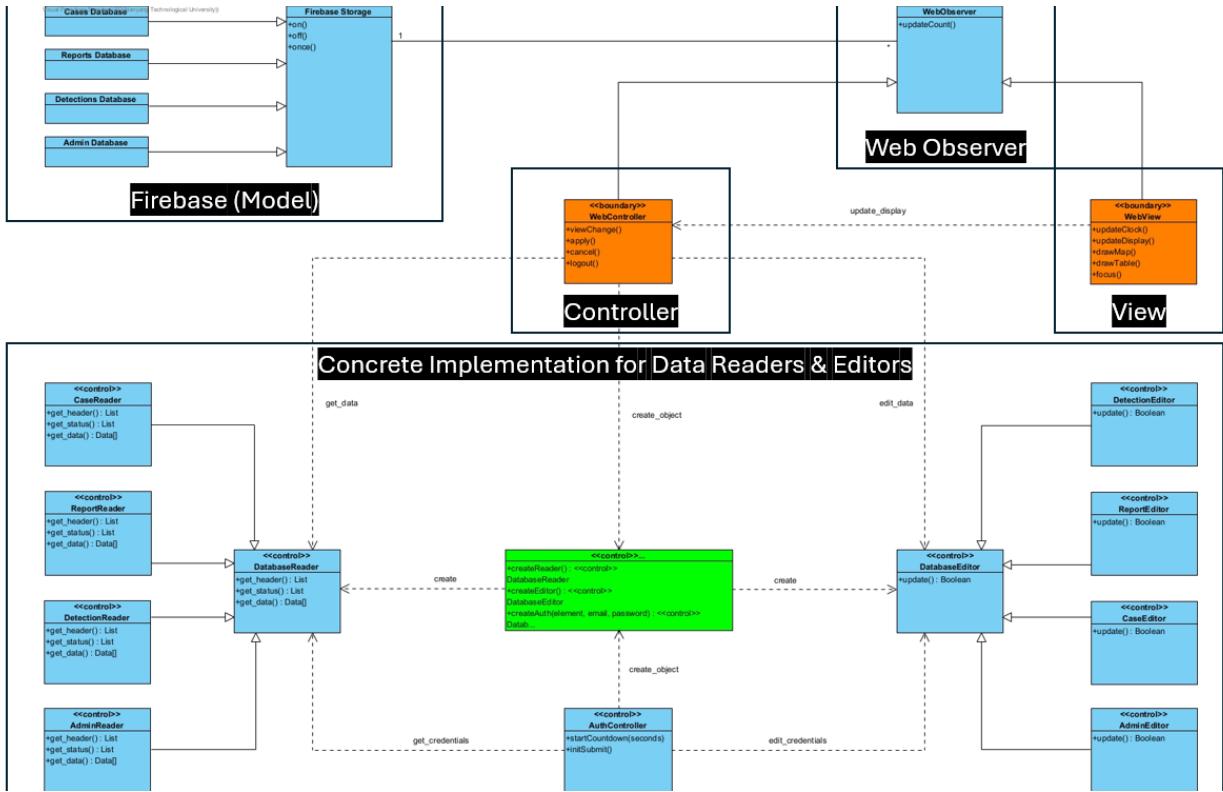


Model-View/Controller Architecture abstract diagram

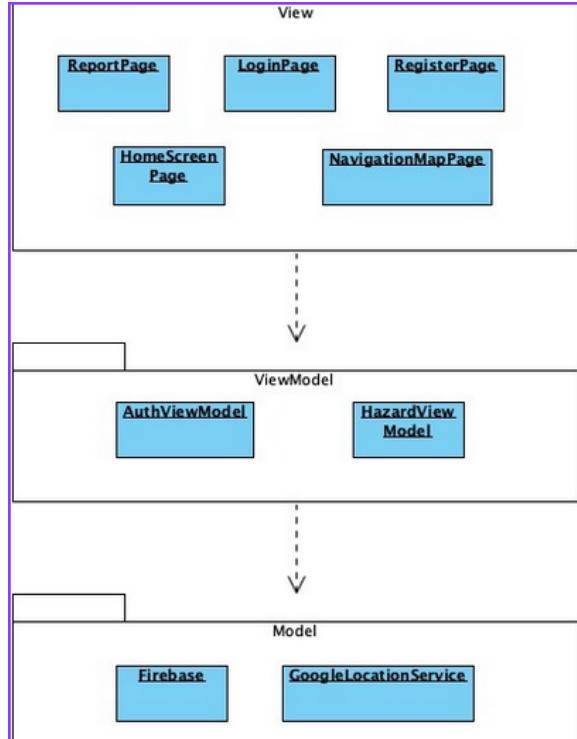
Model-View/Controller (MVC) architecture is adopted to address the problem of **one data being represented by multiple views**. In the Web-interface, Cases/Motorist Reports/Camera Detections each can be displayed in the form of 1) Google Map marker overlays, 2) Entries table consisting of coordinates and damage type and 3) Picture of the damage. In the Android application, Cases are displayed in Google Map marker overlays for 1) Map View, which displays all verified road damages around Singapore and 2) Navigation View, which displays verified road damages within 150m of the motorist's proximity.

When data (e.g. Cases) is updated in the Model, changes are propagated to the Observers (Android App & Web Interface) by calling their corresponding View classes. In theory, the model calls an update() method in each Observer, and the concrete implementations of each Observer's update() methods will call the get_data() of the Model to retrieve the data. In Firebase, this is done implicitly by adding a listener for new data on each call to .on() (subscribe). Upon application/webpage loading, methods in each View class populates its display with the information retrieved.

When an admin initiates changes to the database (or a motorist submits a new report), its corresponding controller class is called, which affects the changes via its concrete data reader and editor implementation, completing the MVC cycle.



Model-View/Controller Architecture detailed diagram (Web)

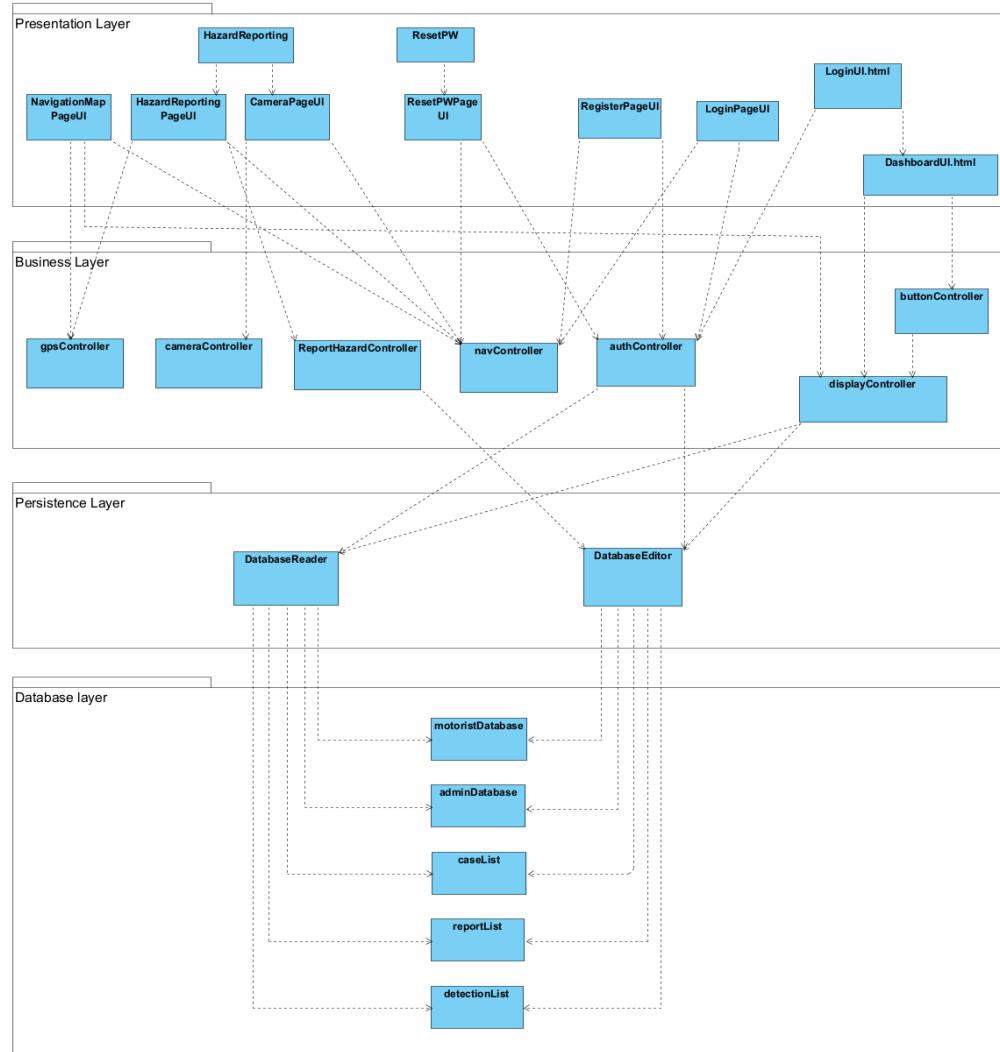


Model-View-ViewModel of Road Guardian Android (simplified)

In the mobile application, we decided to use a slightly different architecture, called Model-View-ViewModel. Here the Model still represents the data and business logic, which is essentially what the Firebase backend is, and the View still manages the presentation layer and the user interface. The difference however lies in the **ViewModel**. Normally, in the MVC architecture pattern the Controller has a reference to the View, which makes architecture **tightly coupled**. In MVVM architecture, however, the ViewModel interacts with the View through a technique called **data binding**, which supports two-way communication and allows the View to observe data directly. The ViewModel is completely unaware of the View's structure, allowing for **better decoupling and smoother flow of data**.

This is crucial for features like **real-time updates** of the mobile map, as seen in the interaction between the HazardViewModel, the HomeScreenPage and the NavigationMapPage. Here the HazardViewModel keeps a StateFlow object representing changes in the database. It's then passed to the various Views to listen to the changes in StateFlow to automatically rerender the page. In a way, this could be seen as a combination of **push-based observer pattern and MVC architecture**.

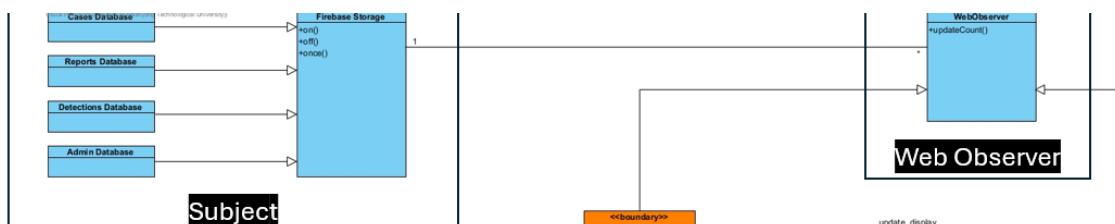
Layered architecture was chosen for our design initially. However as the problem of data being represented by multiple views arise and lower layers calling the upper layers when data is being updated (i.e. Model at the bottom calling update() in the Observer at the upper layers), we decided the Model-View/Controller architecture is a better approach.



Initial Layered Architecture design

7. Design Patterns

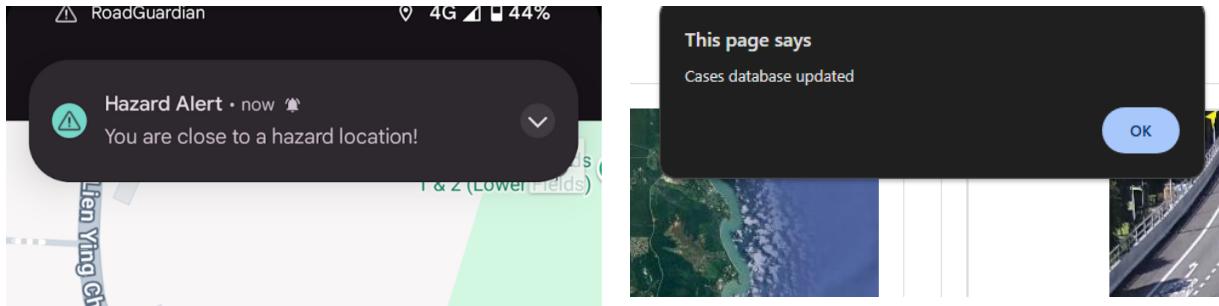
Observer Pattern



Subject-Observer (Web)

Subject-Observer Pattern is employed in our system to ensure **loose coupling**. When data change is detected, the Web and Android observers are notified to retrieve the necessary data

when required. As such, motorists and administrators can refrain from constantly monitoring new cases, but will receive notification when one comes.

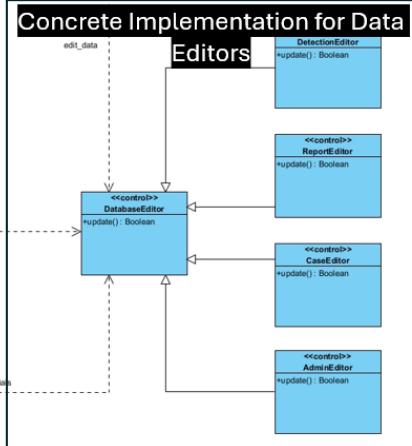
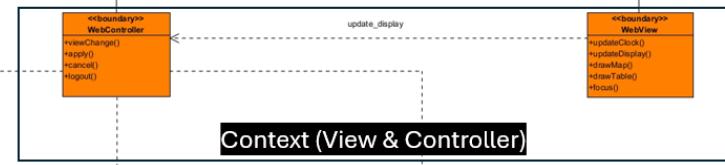
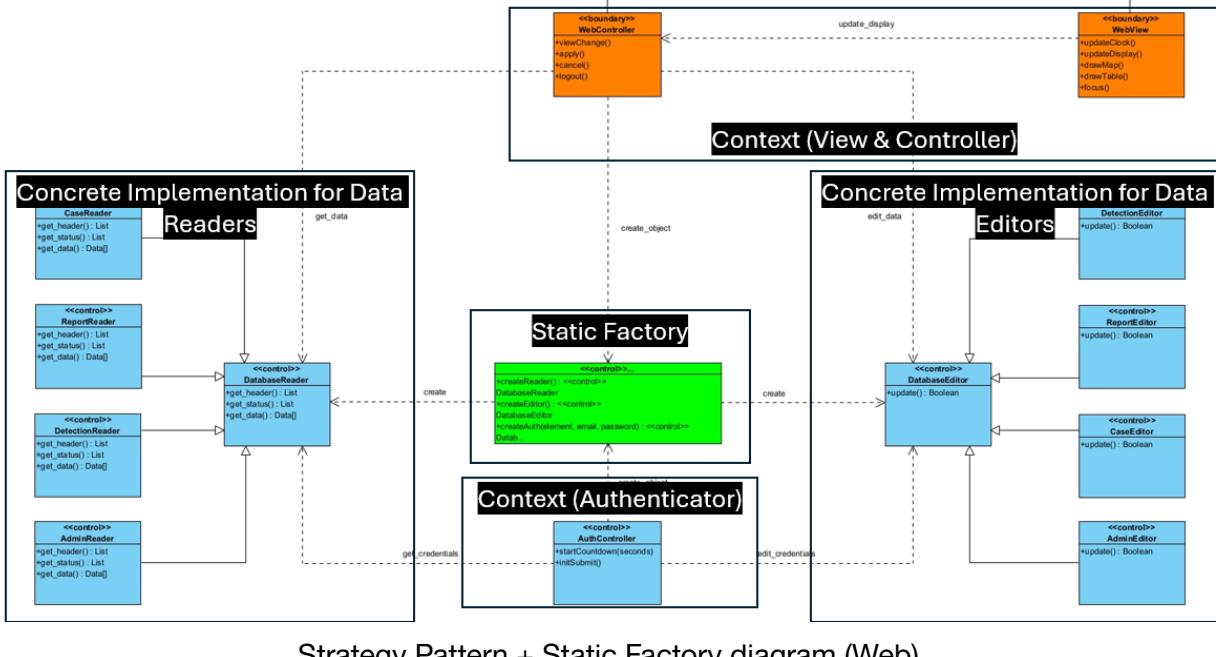


Hazard notification on Road Guardian Android (Left) and Web (Right)

In our implementation, the Firebase Realtime Database storage is regarded as our model. The following critical Model methods are noted in order to implement our Observers (Web & Android)

- .on(): Firebase's subscribe() for Web, called during webpage loading
- .off(): Firebase's unsubscribe() for Web, called when admin logouts/closes the browser
- .once(): Firebase's get_data() for Web, called when data retrieval is needed
- notify() is done implicitly in Firebase onto any Web observer that subscribes to it
- addSnapshotListener(): Firebase's subscribe() for Android, called during upon motorist login
- onCleared(): Firebase unsubscribe() for Android, called upon motorist logout
- get_data() & notify() is done implicitly on any Android observer that subscribes to it

Strategy Pattern + Static Factory



Strategy Pattern + Static Factory diagram (Web)

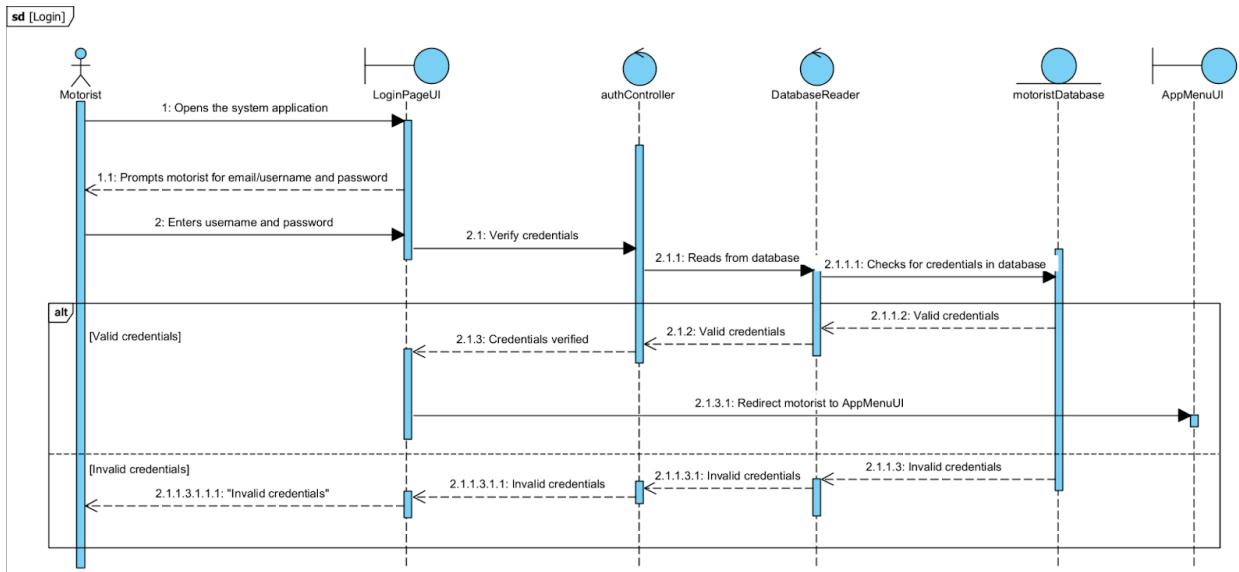
Strategy pattern is employed in our web-interface to address the problem of **interchangeable displays**. When Pending Action (Cases)/Motorist Reports/Camera Detections buttons are clicked, the map markers overlay and entries table are interchanged with the Cases/Reports/Detections data. This hides the actual implementation of how specific data is being retrieved and allows the data displayed to be changed at runtime.

Static Factory is added to **decouple class selection and object creation**. When a new concrete data reader/editor class is added, only the factory class needs to be amended (and minimally), minimising impact of change to code.

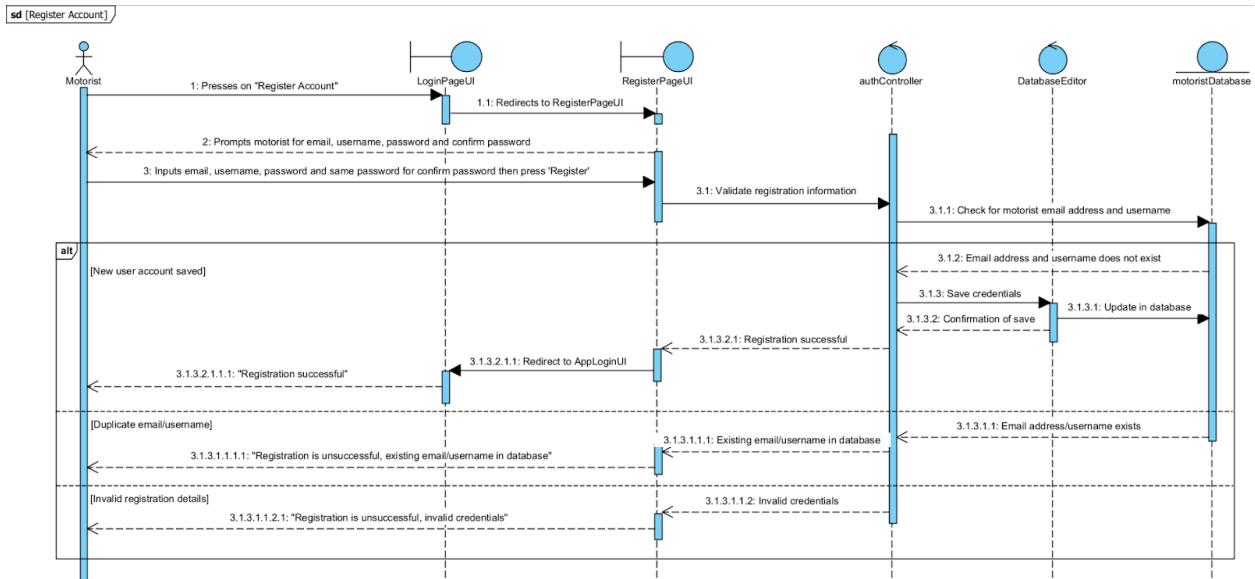
8. Sequence diagram

*Take note that although entity class symbols are used to represent databases, there are no actual entity classes in our implementation.

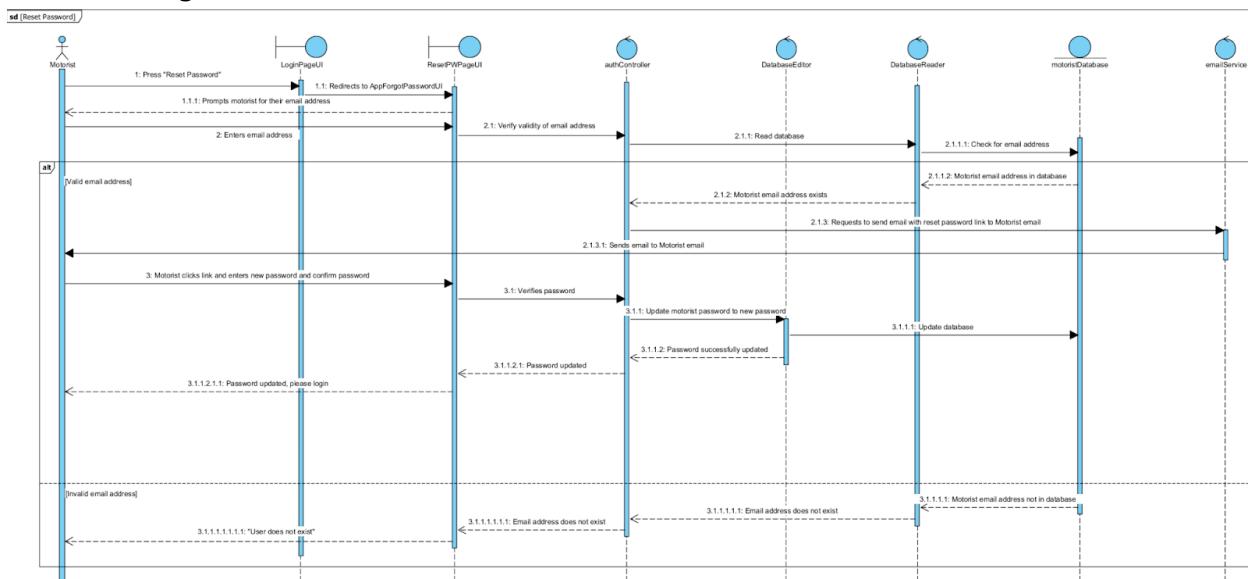
Motorist Login



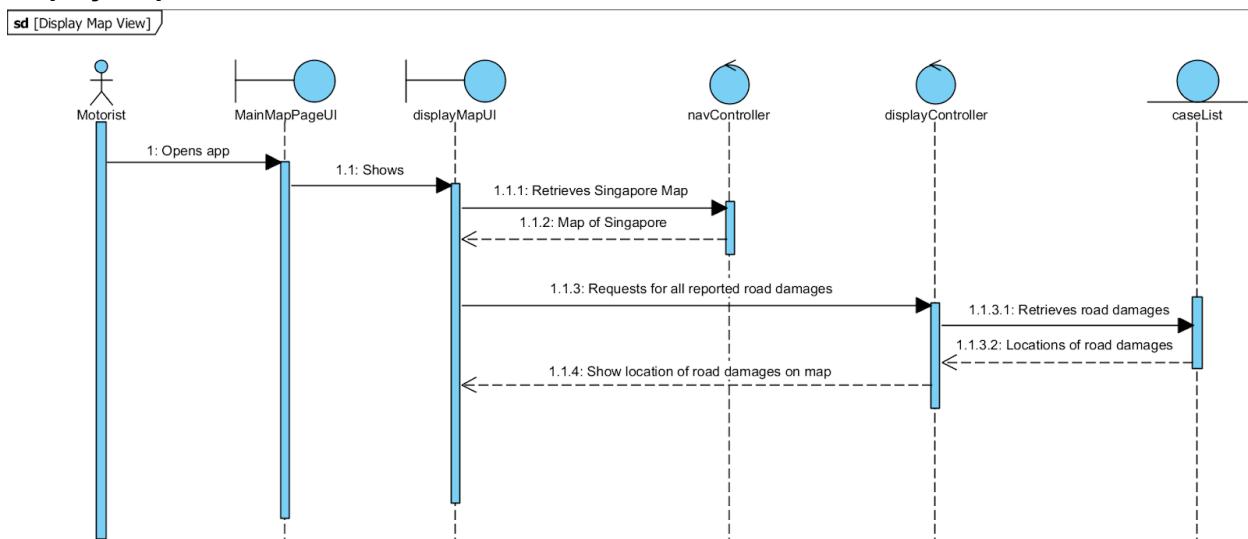
Motorist Register Account



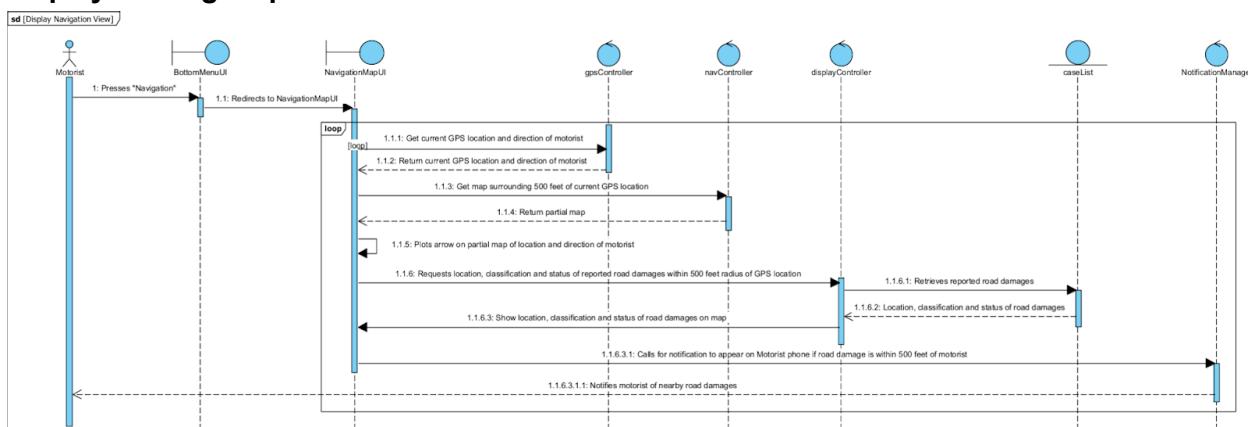
Motorist Forgot Password



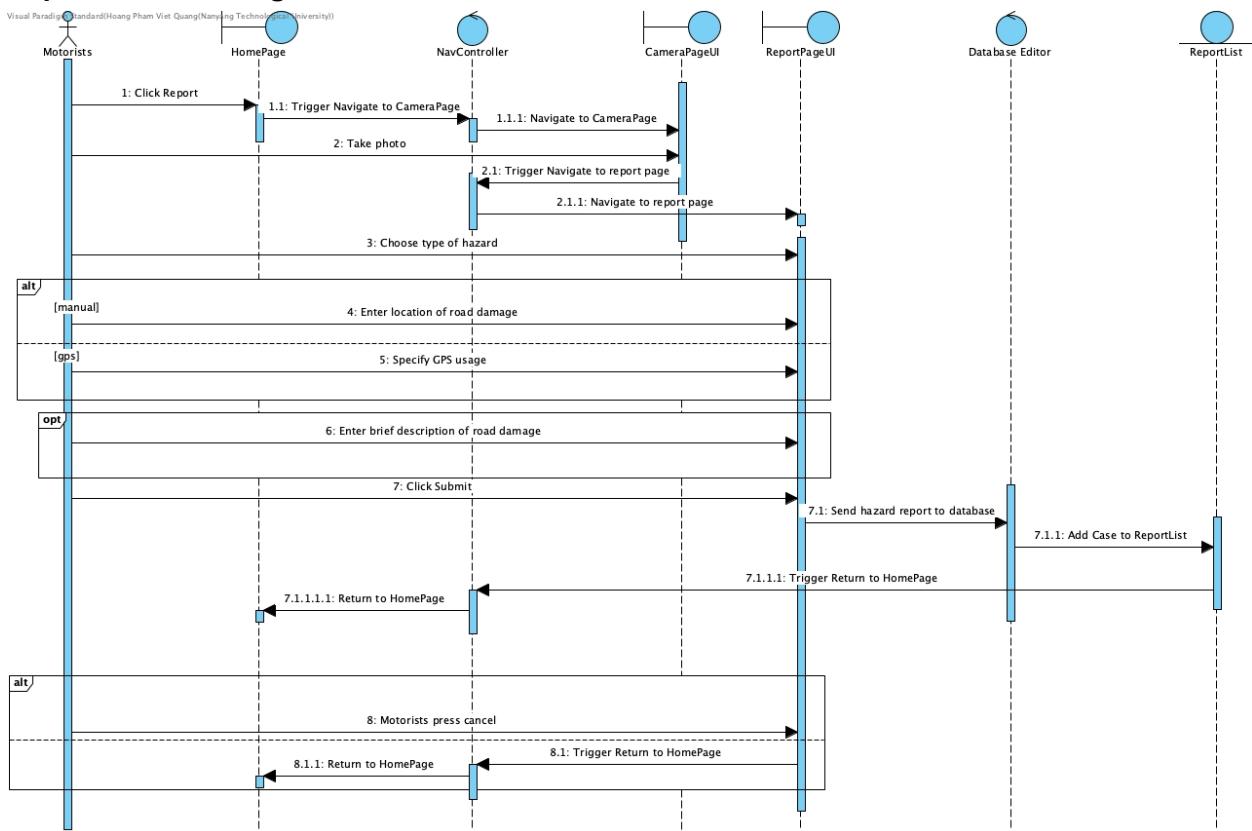
Display Map View



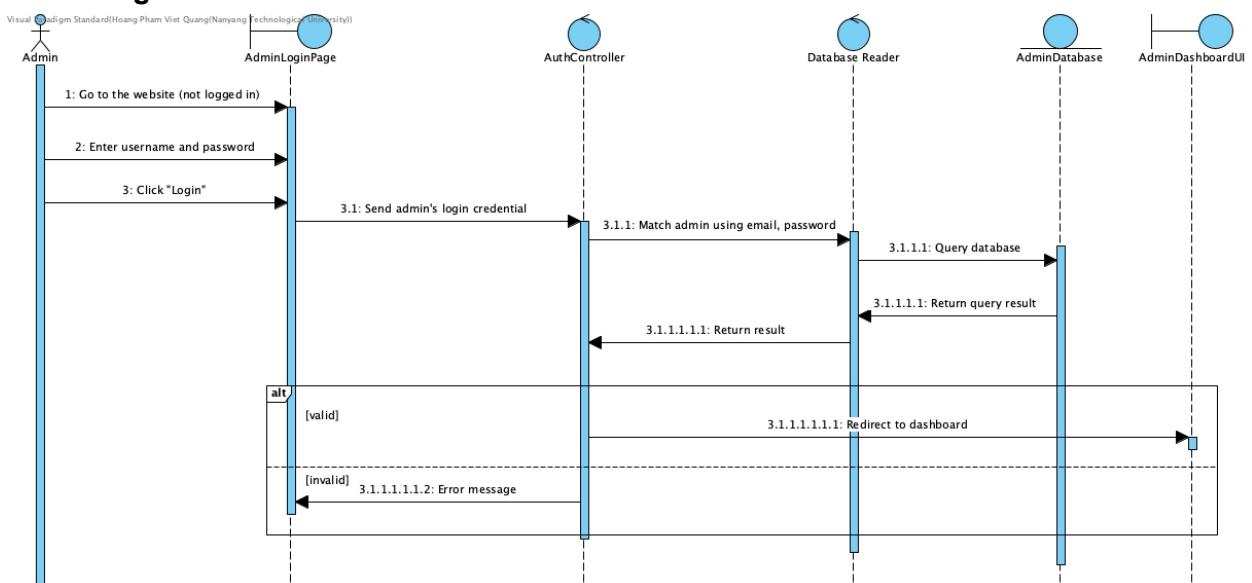
Display Moving Map



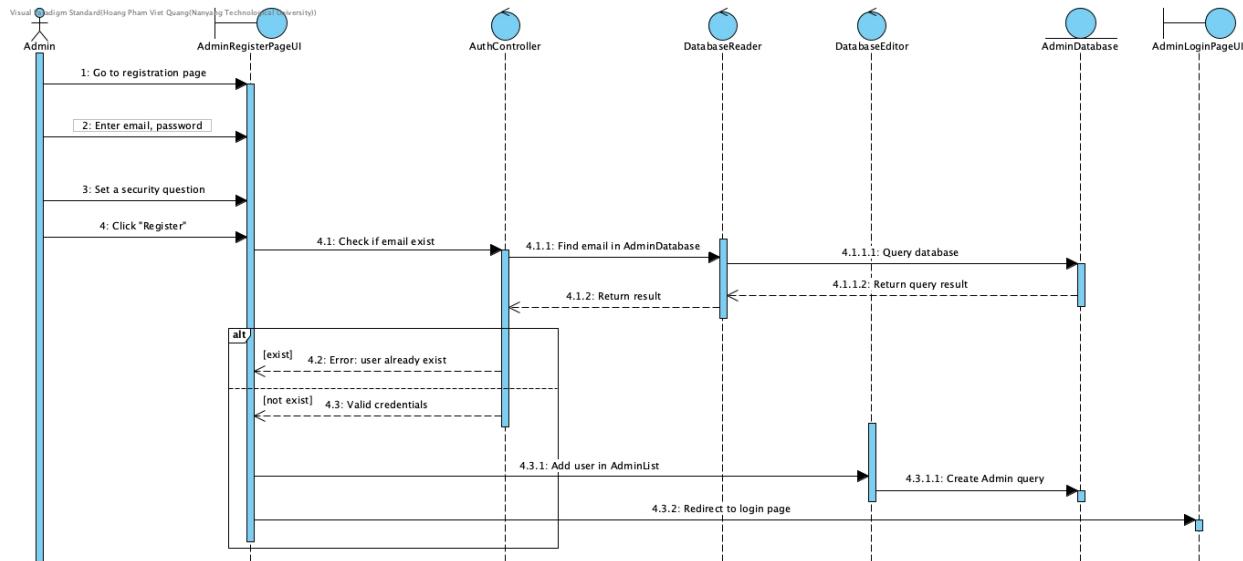
Report Road Damage



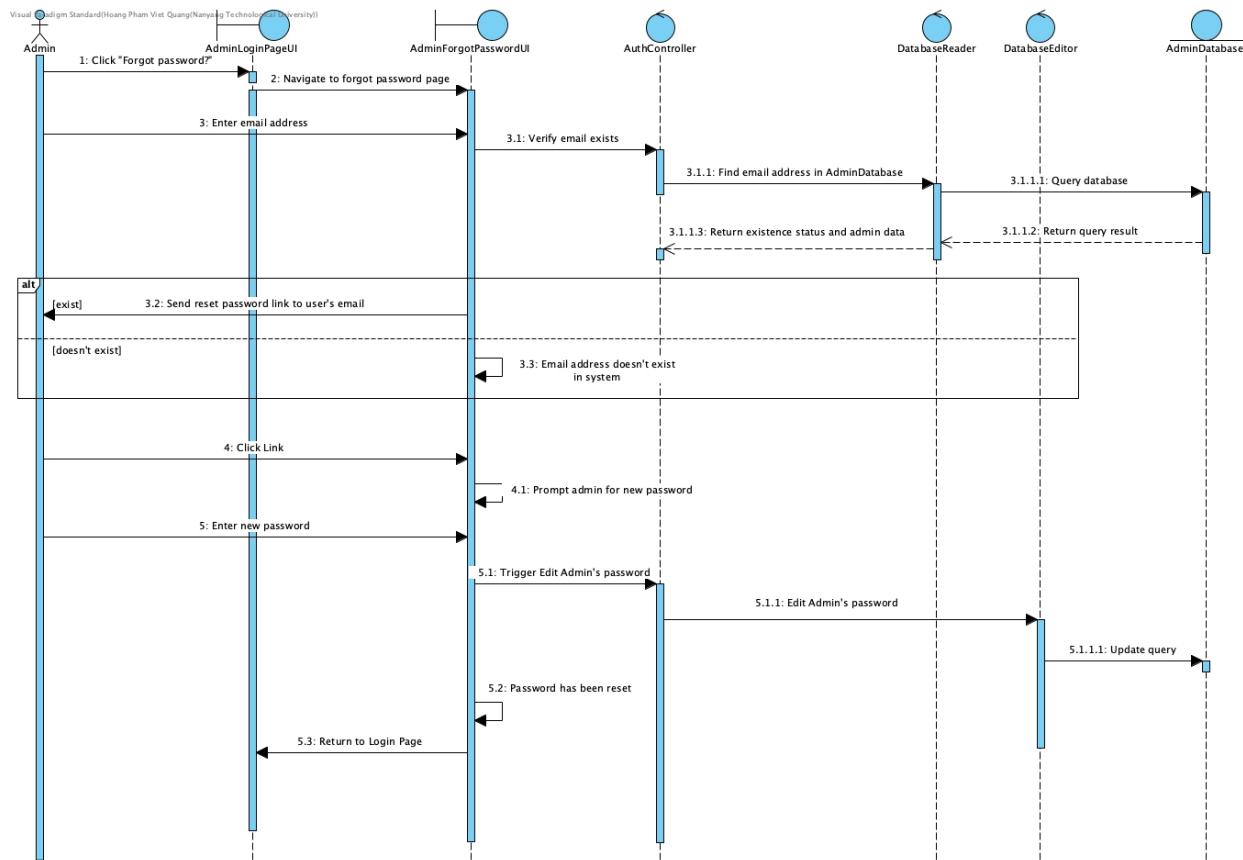
Admin Login



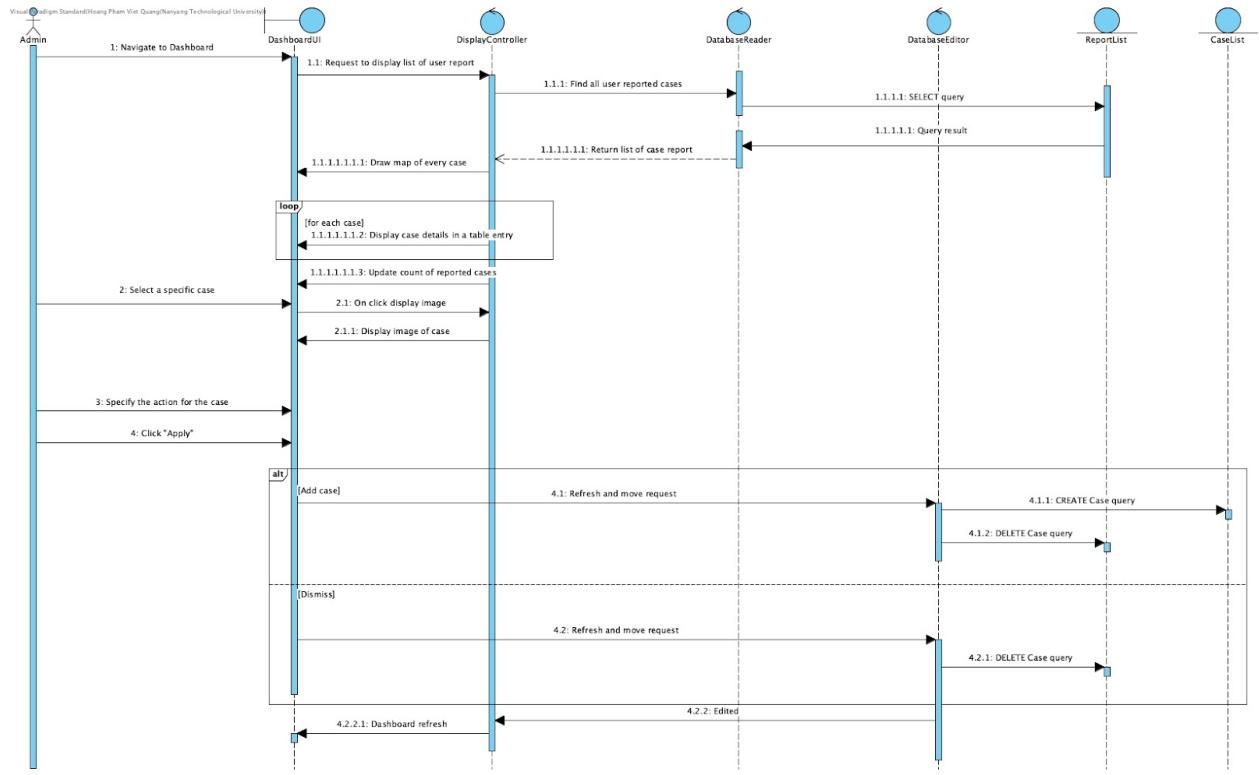
Admin Register Account



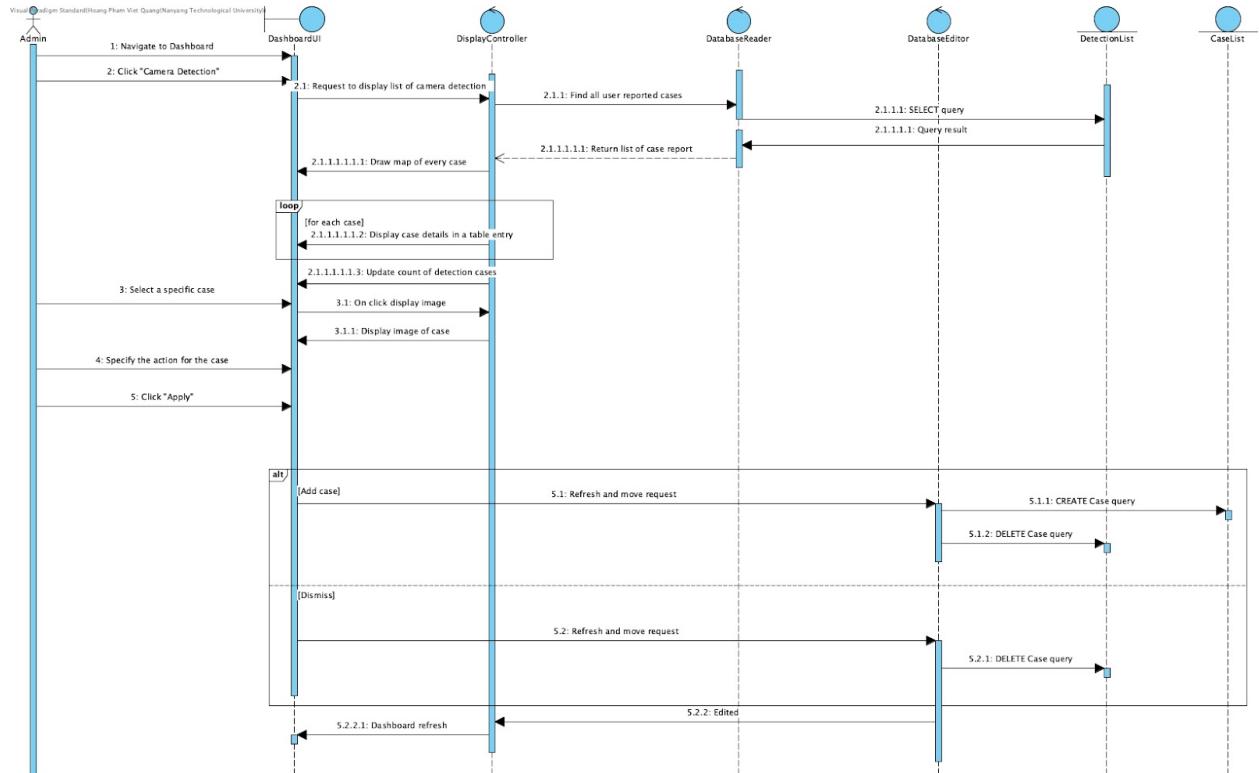
Admin Forgot Password



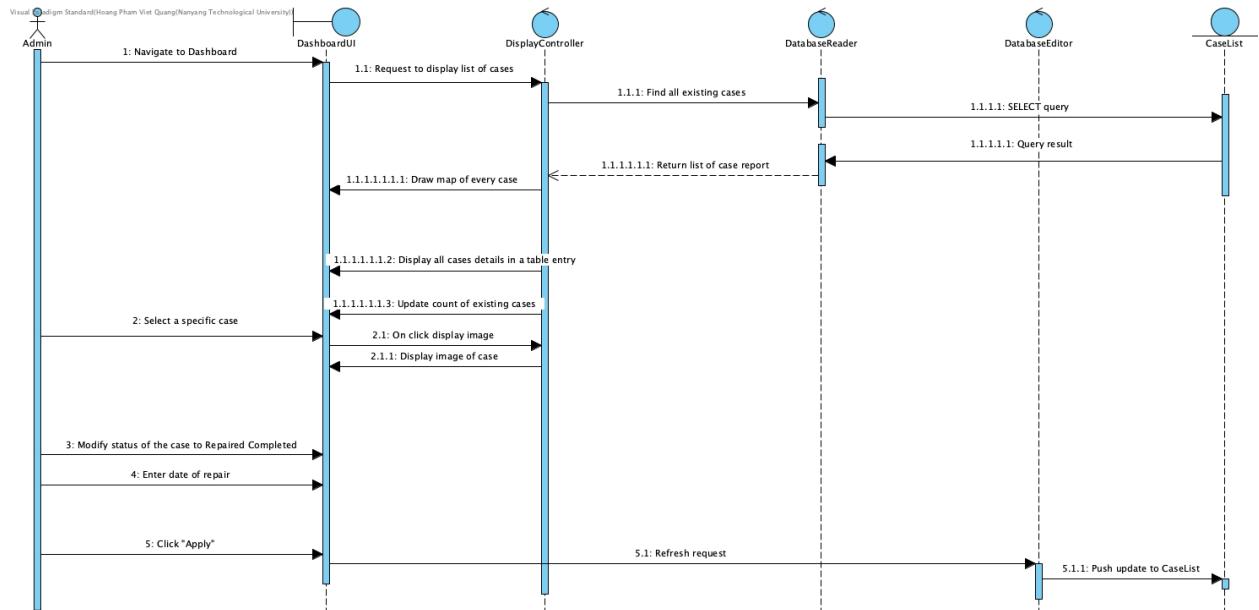
Manage motorist reported cases



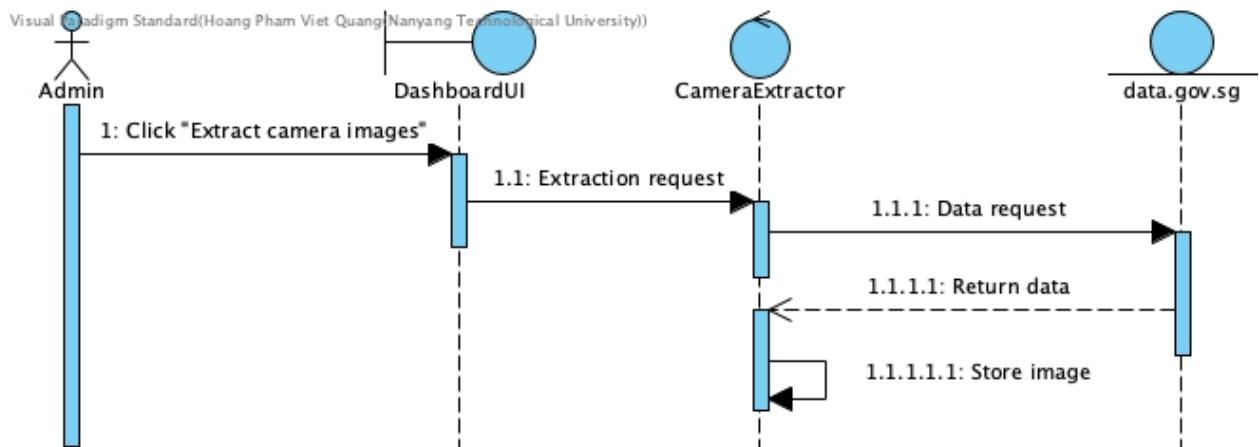
Manage camera detection



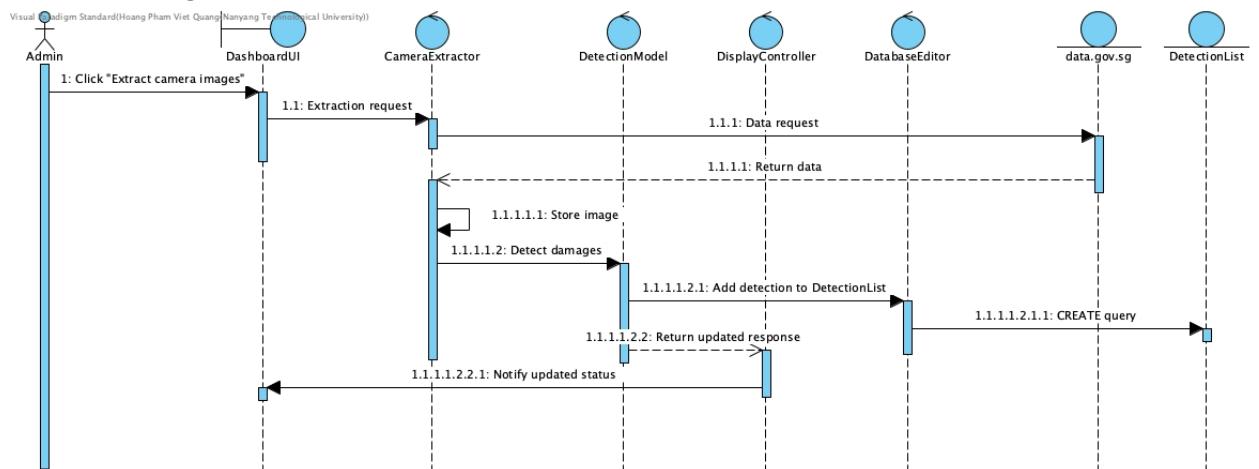
Manage existing cases



Collect traffic images

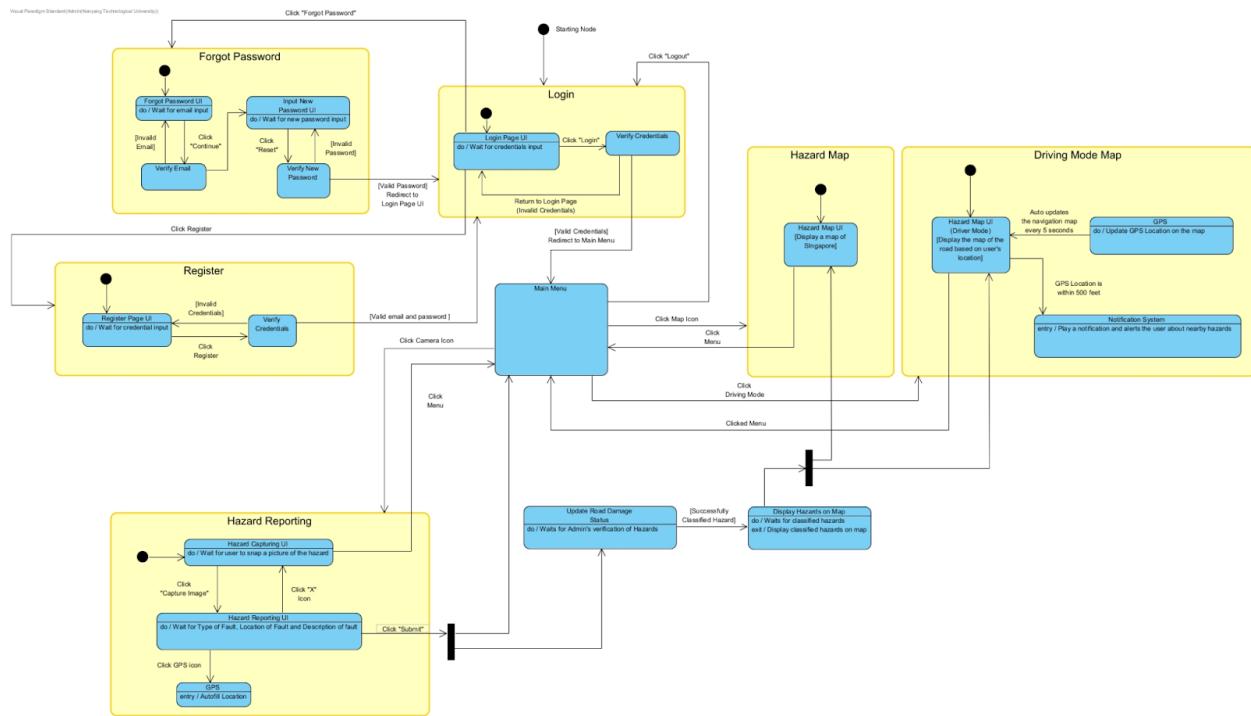


Detect damages

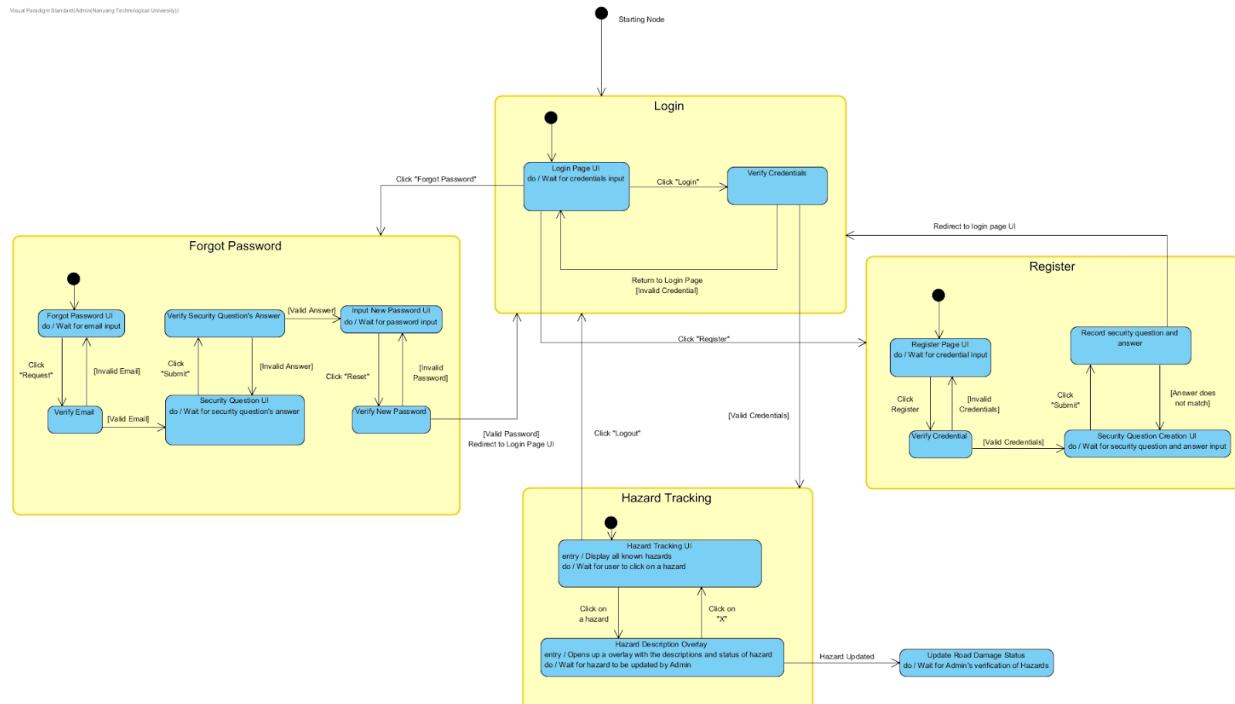


9. Dialog map

Dialog Map for user application



Dialog map for Admin webpage



10. Implementation

10.1. Operating System

- Windows 10 and above, Mac OS 14 and above
- Android OS 14 Upside Down Cake and above (For Road Guardian Android)

10.2. Browser (Road Guardian Web)

- Firefox/Chrome Browser for Windows, Safari Browser for Mac OS

10.3. Databases

- Firebase Realtime Database (For storing verified cases, motorist reports and camera detections)
- data.gov.sg (To extract raw traffic camera image for road damage detection)

10.4. Languages

- HTML, CSS, JavaScript (Road Guardian Web)
- Java, Kotlin (Road Guardian Android)
- Python 3.11.9 (Deep learning for road damage detection)

10.5. Deep Learning Tool*

- YOLOv7-Tiny with PyTorch v2.3.1

*As YOLOv7 (and other YOLO versions) require specific python dependencies, we've decoupled the road damage detection function from the main web interface. This simplifies the set up of YOLO to one (or a few) PC while supporting critical Admin functions across multiple platforms.

11. UI Mockups (Web Application)

RoadGuardian 08:42:41 Logout

			Pending Action: 19	Motorist Reports: 4	Camera Detections: 25		
-OBO0F0925XNjk_ZBAA0w	https://storage.googleapis.com/test-cb666.appspot.com/images/6cabaf131-9ae1-11ef-90c9-f0d4150e1e20.jpg	1.323604823	103.8587802	peeling off premix	2024/11/05 03:17:25	dd/mm/yyyy	Pending Repair
-OBO0aJGP-dFeOqCNgfym	https://storage.googleapis.com/test-cb666.appspot.com/images/6cd20a3-9ae1-11ef-aa9d-f0d4150e1e20.jpg	1.326024822	103.905625	peeling off premix	2024/11/05 03:17:25	dd/mm/yyyy	Pending Repair
-OBO0aRFk04lARAC9t94S	https://firebasestorage.googleapis.com/v0/b/test-cb666.appspot.com/o/images%2FmzvwYBuhiR6VT4MCpKEJfPVNg%2F17308231370289alt=media&token=19f12d33-f199-4d95-aea7-1988b2e173cf	1.3460299	103.6873037	j	2024/11/06 00:12:18	dd/mm/yyyy	Pending Repair Repair Completed Delete

Entries

Apply Cancel

Road Guardian Web displaying Admin verified cases of road damages

RoadGuardian 08:42:11 Logout

			Pending Action: 19	Motorist Reports: 4	Camera Detections: 25	
	cb666.appspot.com/o/images%2Fc3Xwl4UzccWvOk18sKDwU4TCxR32%2F1730899784373?alt=media&token=f92443b-e040-4df0-a8ff-95556eac5739			NTU Hall 1	21.29.46	
-OB9TnMDfkSz0NYIEam	https://firebasestorage.googleapis.com/v0/b/test-cb666.appspot.com/o/images%2FSS7lawwV05bCXDrYgiWDL8d7W2%2F1731047661666?alt=media&token=a514cf54-d7dc-48c1-b8ea-6ac18410c571	1.3462911	103.6824696	pothole near bus stop	2024-11-08 14:34:28	Pending Review Add Case Dismiss
-OB94B6cp0HtkhrnXZ3V	https://firebasestorage.googleapis.com/v0/b/test-cb666.appspot.com/o/images%2FekJTB262lOGtYKyexu7iZafyAm1%2F1731051933239?alt=media&token=e672d574-8e59-4d83-9112-d3eb9b446c92	1.3462498	103.6825261	wer	2024-11-08 15:45:34	

Entries

Apply Cancel

Road Guardian Web displaying motorist reports of road damages

Entries

ID	URL	Latitude	Longitude	Damage Type	Date	Action
15	https://storage.googleapis.com/test-cb666.appspot.com/images/d4129b1f-9da7-11ef-9d80-d25e00e016f8.jpg	1.332691	103.770278	alligator crack	2024/11/08 16:02:41	
16	https://storage.googleapis.com/test-cb666.appspot.com/images/1d6d97bad-9da7-11ef-92cd-d25e00e016f8.jpg	1.356299	103.716071	pothole with crack	2024/11/08 16:02:46	
17	https://storage.googleapis.com/test-cb666.appspot.com/images/d7d41291-9da7-11ef-84ce-d25e00e016f8.jpg	1.322893	103.6635051	peeling off premix	2024/11/08 16:02:48	
18	https://storage.googleapis.com/test-cb666.appspot.com/images/d8f3a706-9da7-11ef-8437-d25e00e016f8.jpg	1.354245	103.963782	alligator crack	2024/11/08 16:02:49	
19	https://storage.googleapis.com/test-cb666.appspot.com/images/d9ff1a69-9da7-11ef-a0b5-d25e00e016f8.jpg	1.354245	103.963782	multi crack	2024/11/08 16:02:50	

Pending Action: 19 Motorist Reports: 4 Camera Detections: 25

Pending Review
Add Case
Dismiss

Apply Cancel

Road Guardian Web displaying camera detection of road damages

Login

Email

Password

Login

Register Account

Reset Password

Register Account

Email

Password

Register

Go To Login

Reset Password

Reset Password

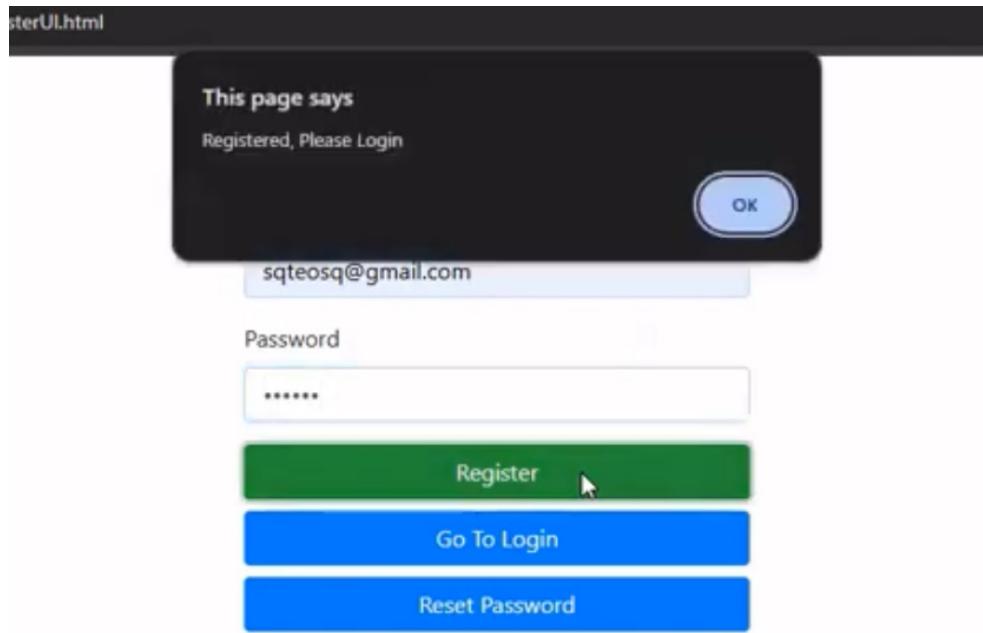
Email

Reset Password

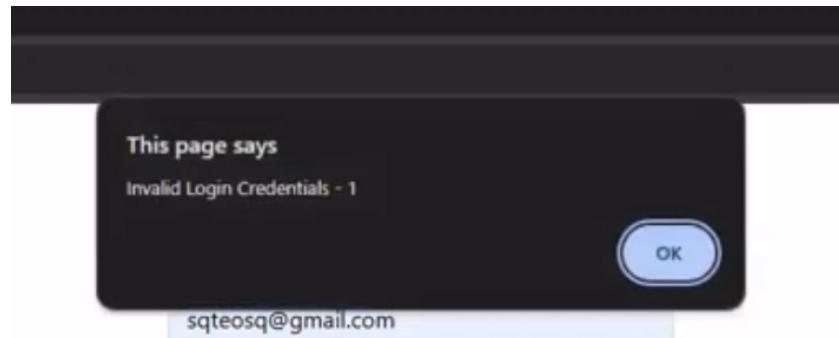
Register Account

Go To Login

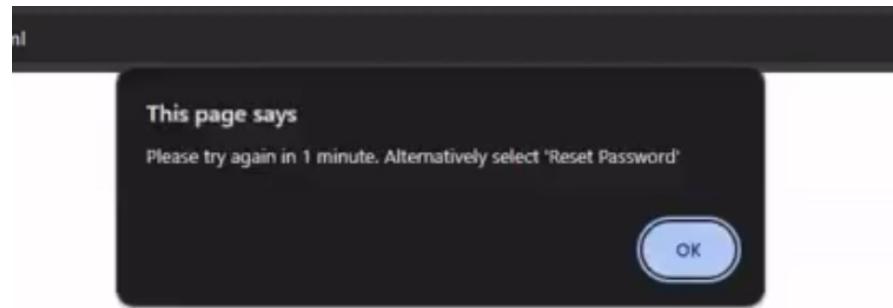
Road Guardian Web 1) Login, 2) Register Account and 3) Reset Password pages



Road Guardian Web successful registration



Road Guardian Web wrong login credentials with number of failed attempts



Password

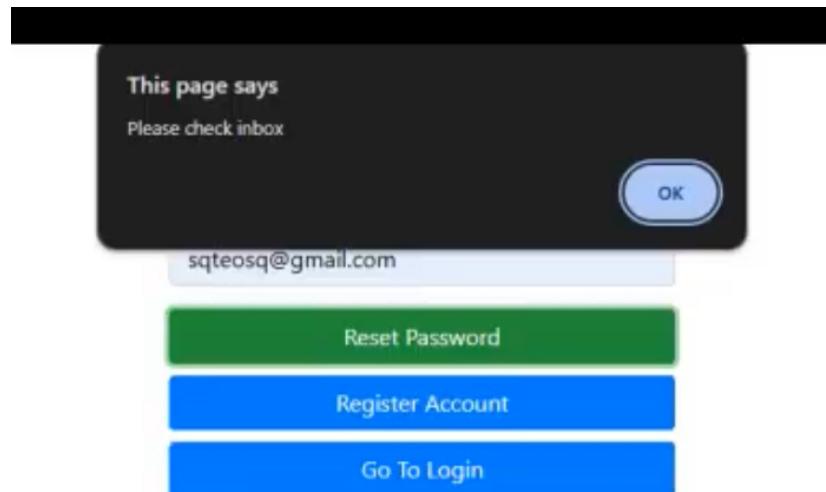
...

Login

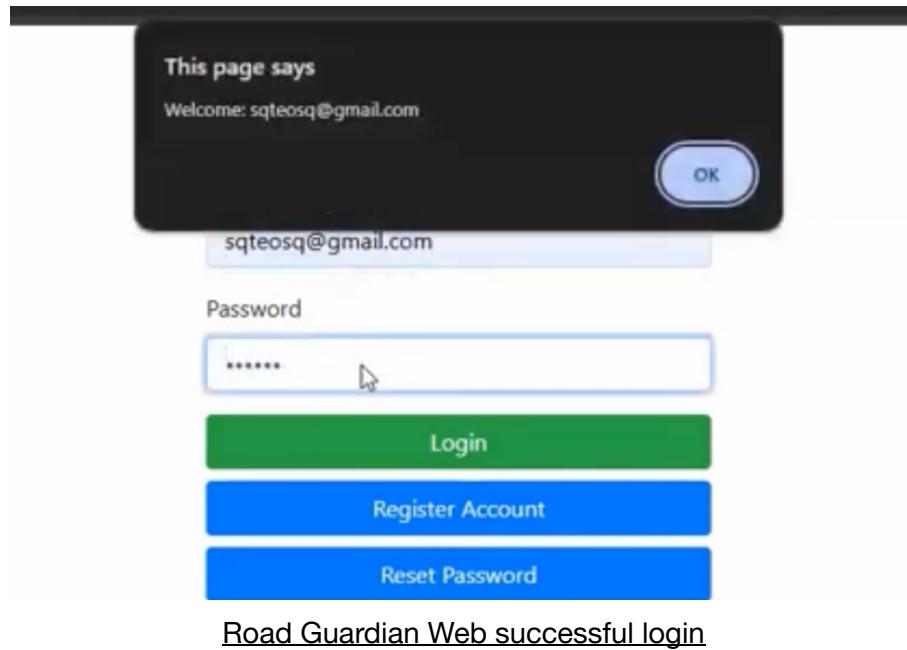
Register Account

Reset Password

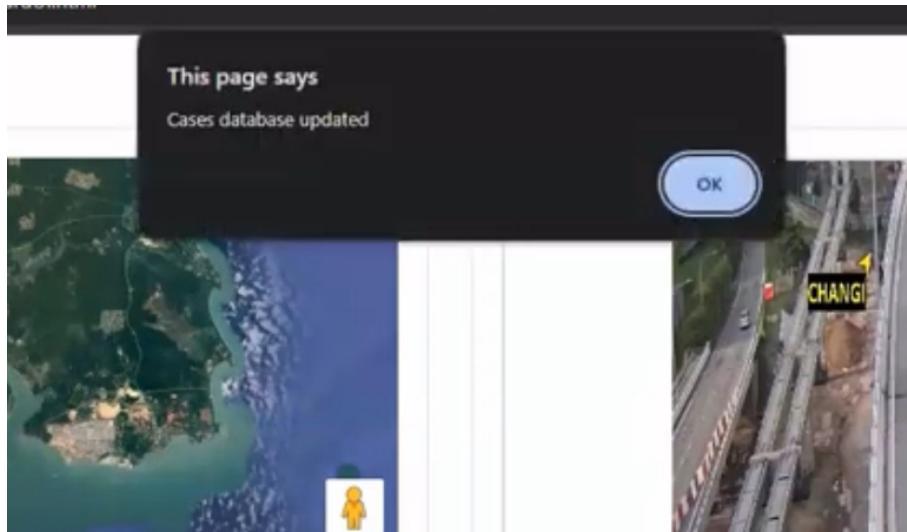
Road Guardian Web five failed login attempts



Road Guardian Web reset password notification



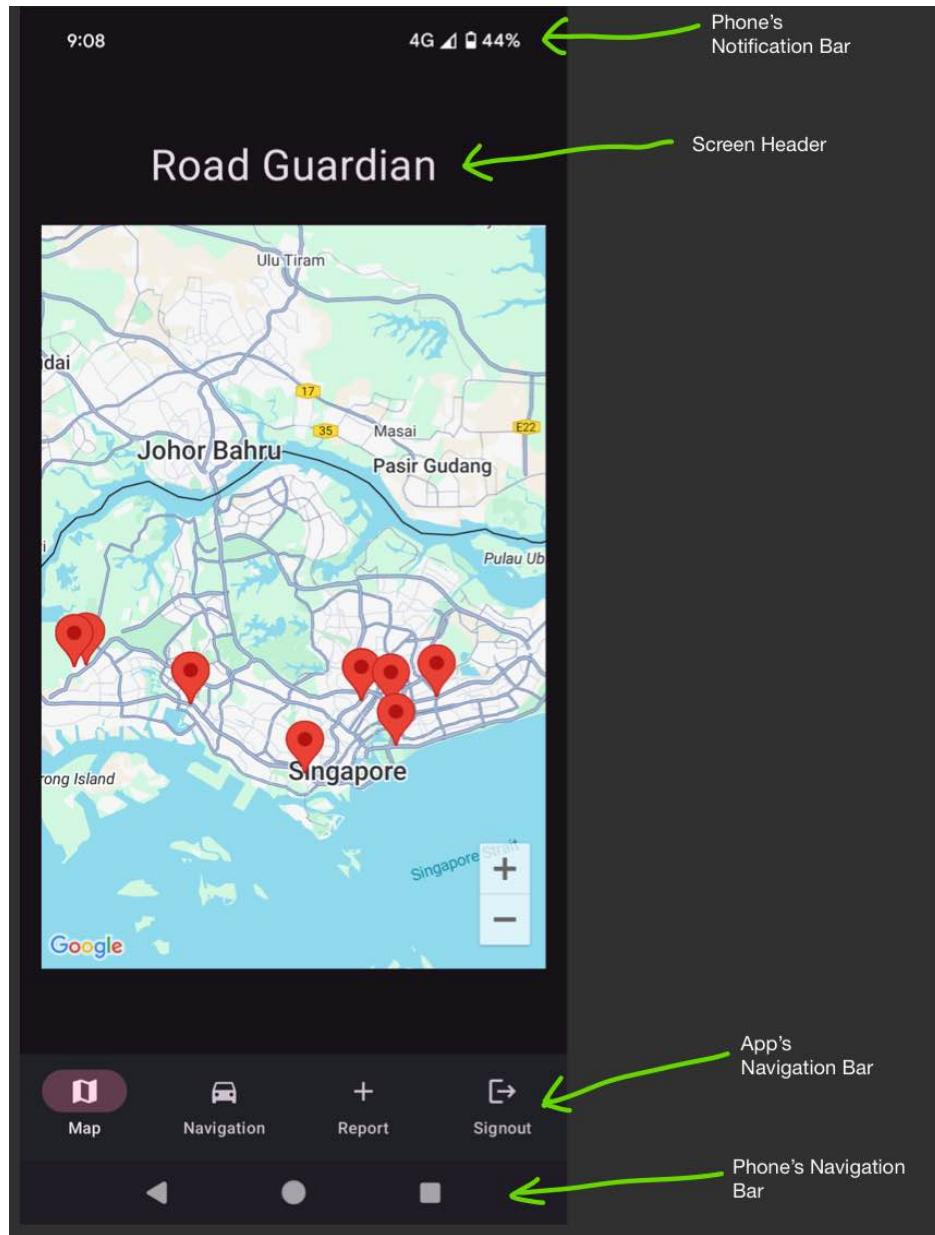
Road Guardian Web successful login



Road Guardian Web update notification message

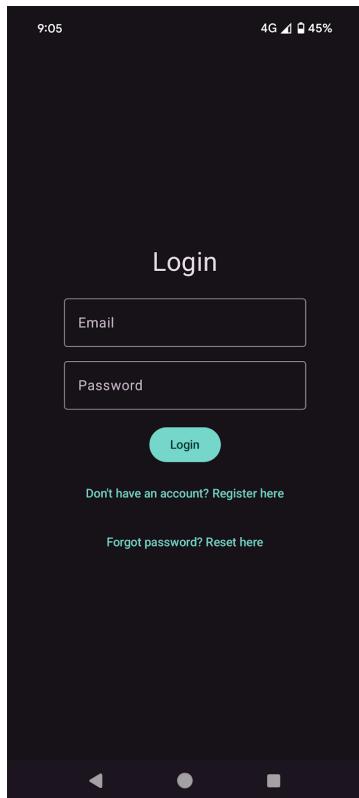
12. UI Mockups (Android)

12.1: App Screen Template

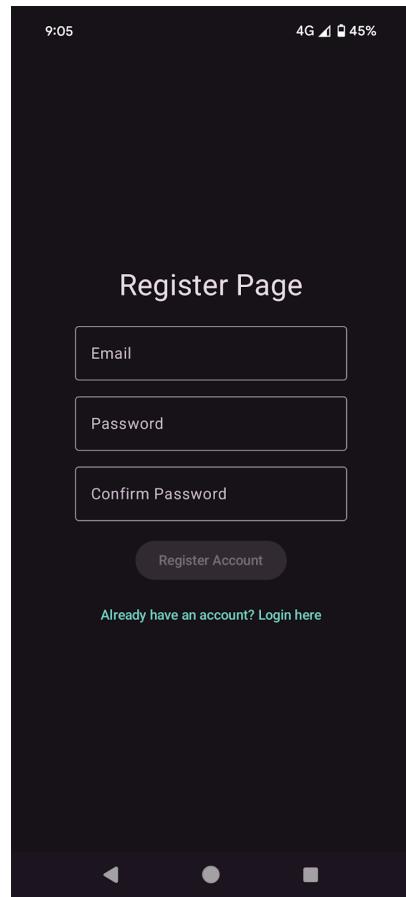


12.1: App Screen Template

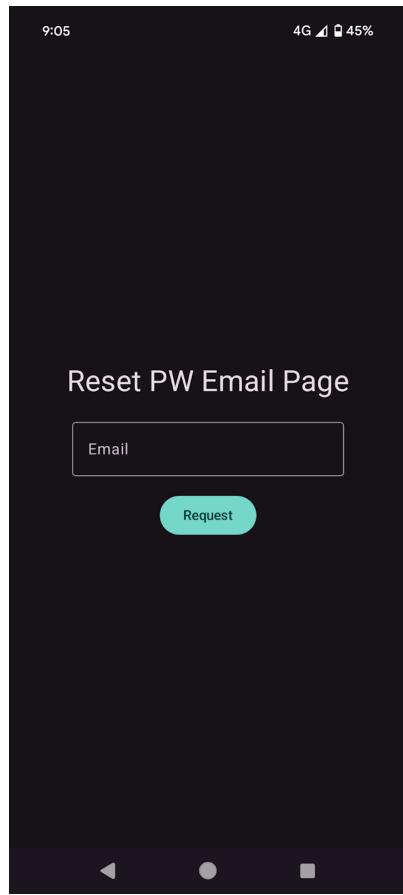
12.2: Authentication (Login, Register, Reset Password):



12.2.1: Login UI

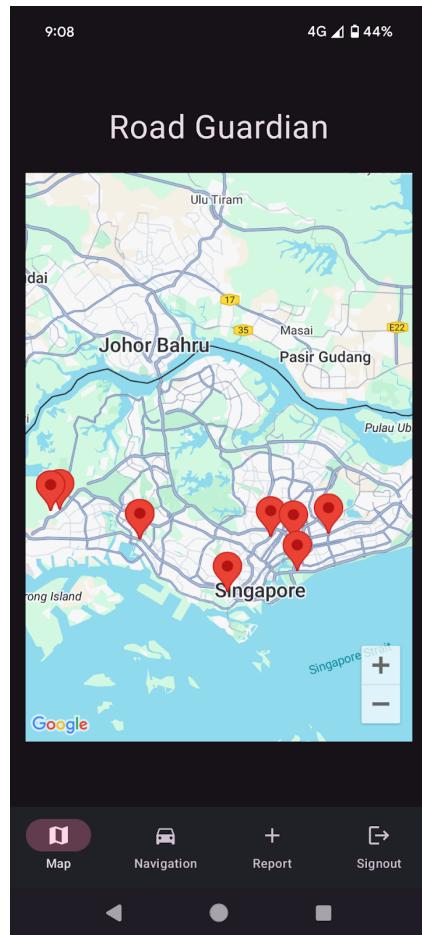


12.2.2: Register UI

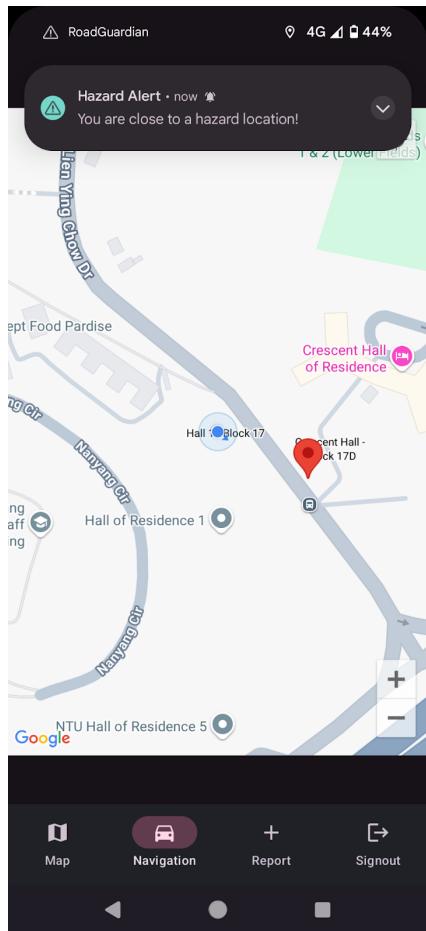


12.2.3: Reset Password UI

12.3: Main Map & Navigation:



12.3.1: Main Map UI

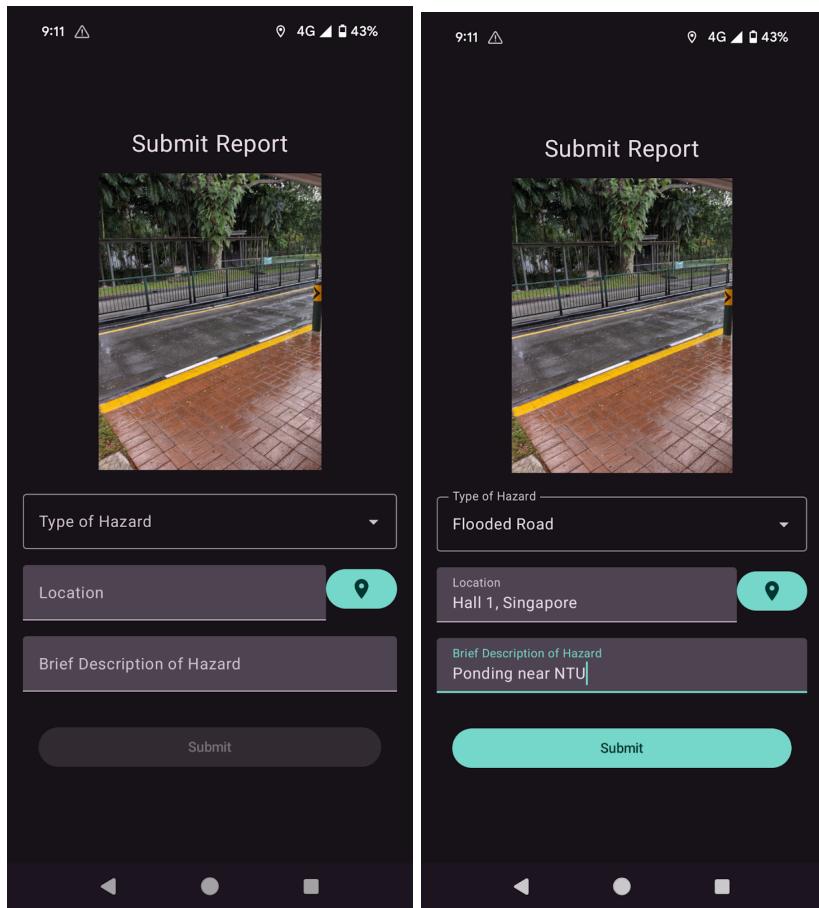


12.3.2: Motorist Navigation UI

12.4: Hazard Camera and Hazard Reporting:

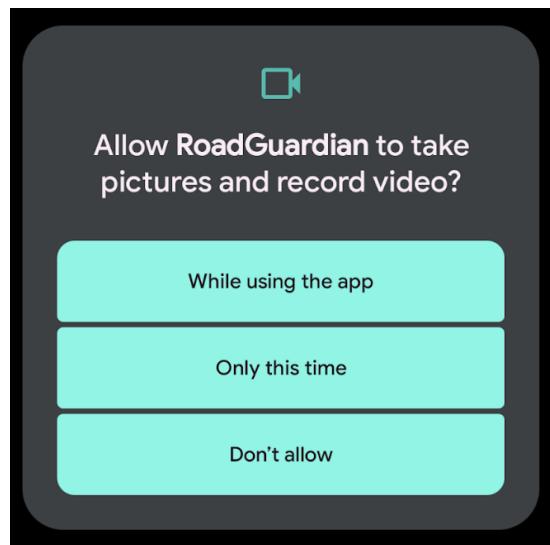


12.4.1: Hazard Camera UI

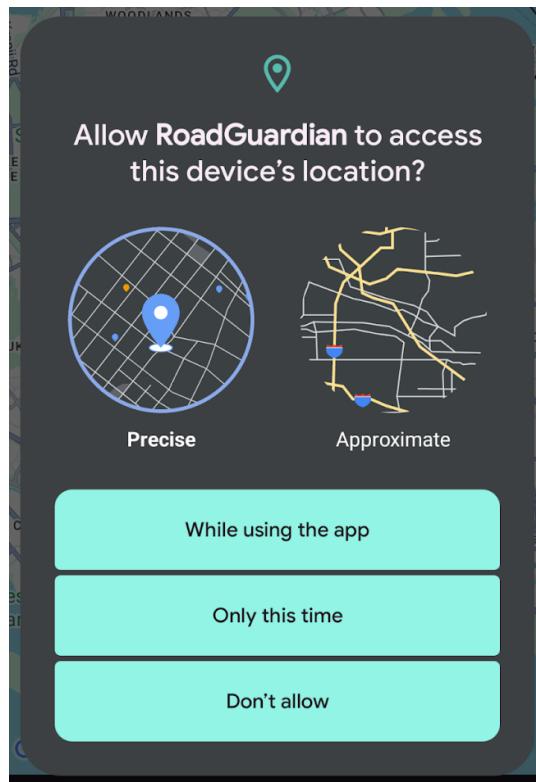


12.4.2: Hazard Reporting UI

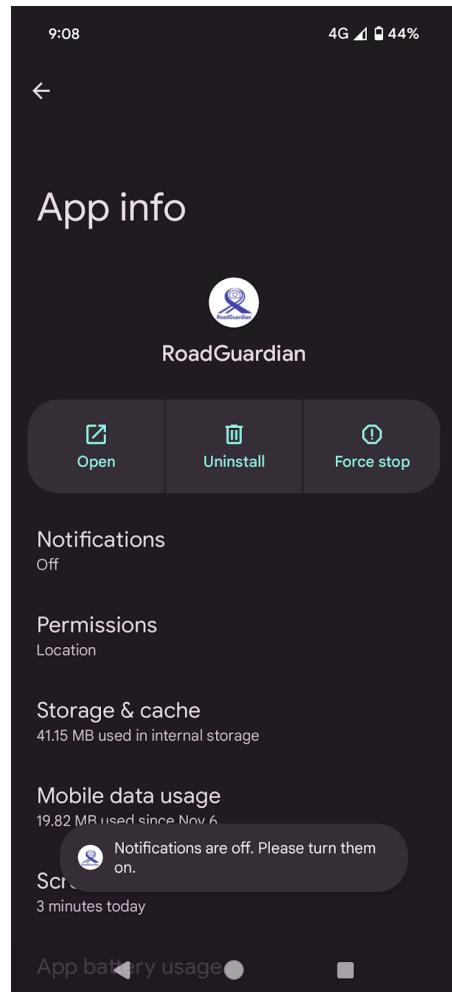
12.5: Permission Handler:



12.5.1: Camera Permission Handler UI

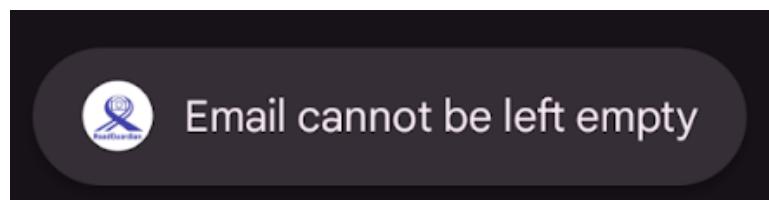


12.5.2: Location Permission Handler UI

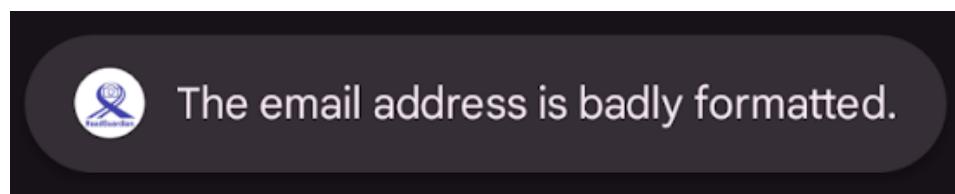


12.5.3: Notification Permission Handler UI

12.6: Success/ Error Messages Display (Toast Message)



12.6.1: Error: Empty email message



12.6.2: Error: Badly formatted email message



The email address is already in use by another account.

12.6.3: Error: Email already exist in the database message



Password must contain at least 8 characters, including uppercase, lowe...

12.6.4: Error: Password does not meet the security requirements message



Account registered successfully!

12.6.5: Success: Account is successfully registered message



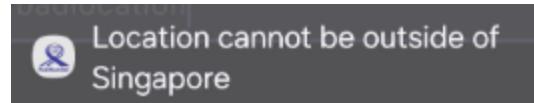
Check email to reset your password.

12.6.6: Success: Link to reset password has been sent to user's email message



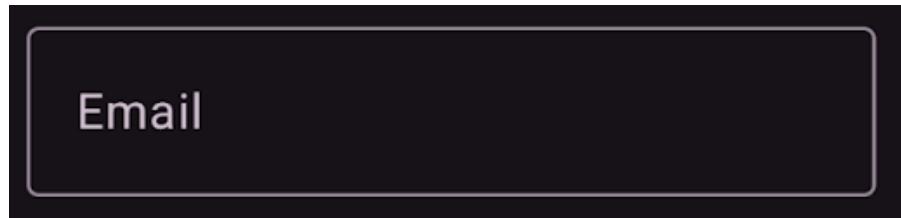
Report submitted

12.6.7: Success: Hazard report is successfully submitted message

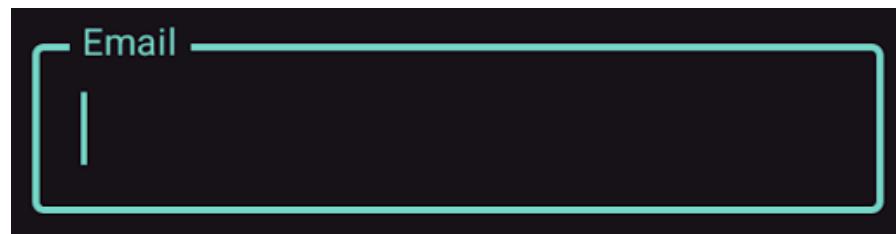


12.6.8: Error: Location input is outside Singapore message

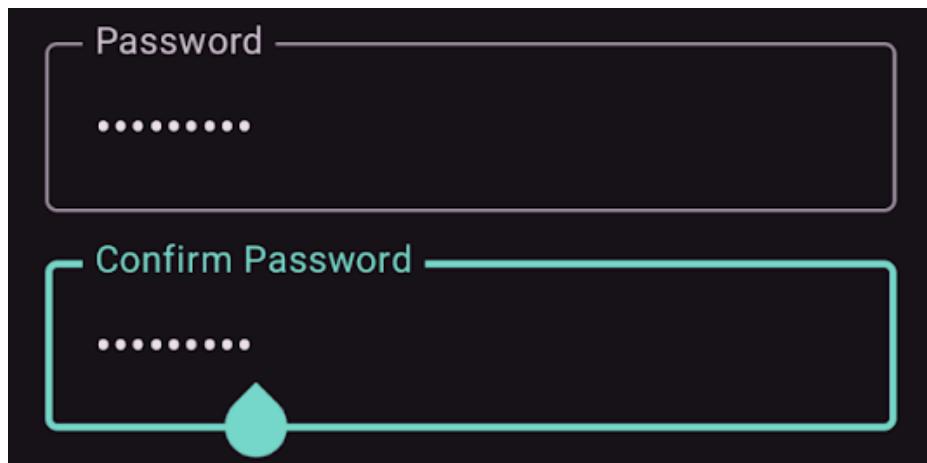
12.7: Common Component



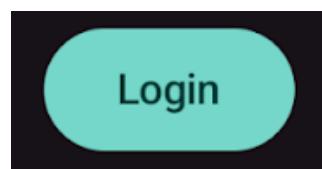
12.7.1: Form Text Input



12.7.2: Form Text Input when selected



12.7.3: Masked Form Text Input For Password

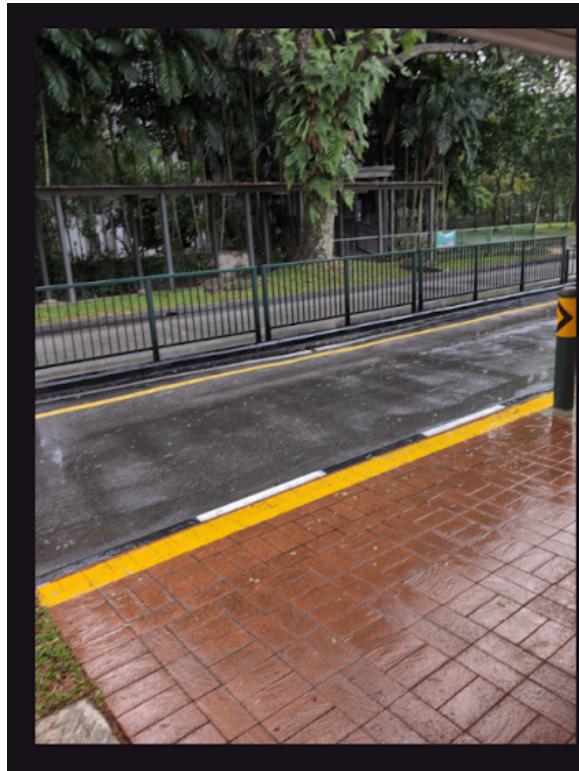


12.7.4: Form Submit Button

[Don't have an account? Register here](#)

[Forgot password? Reset here](#)

12.7.5: Form Navigation Buttons



12.7.6: Image Preview Box

Type of Hazard

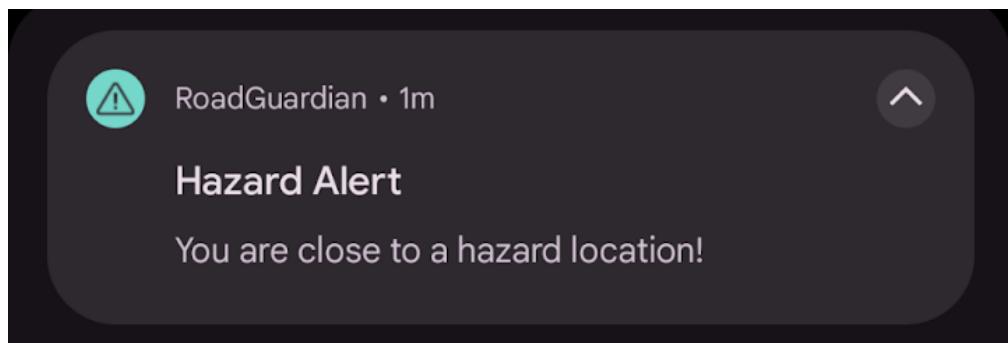


12.7.7: Form Dropdown Input

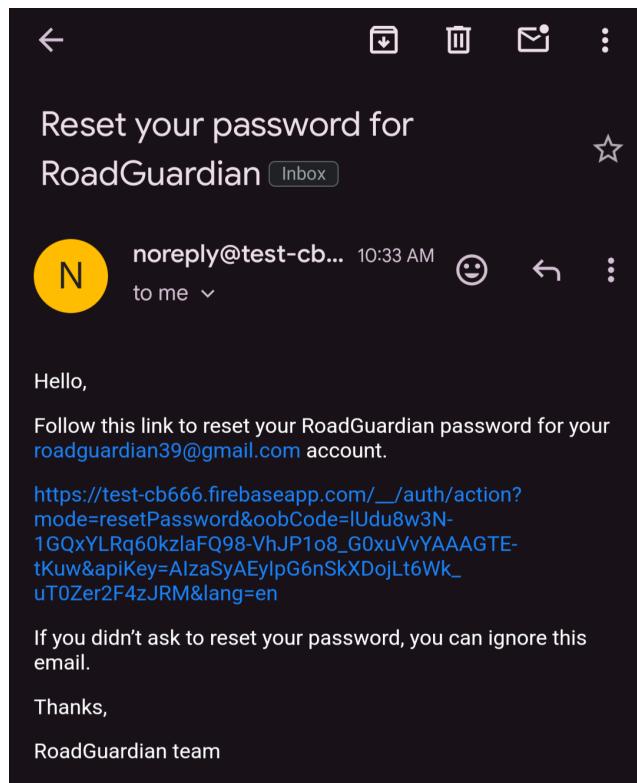


12.7.8: Form Location Input

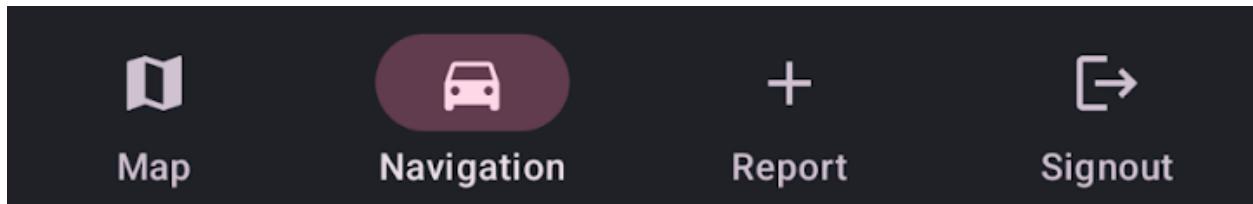
12.8: Misc



12.8.1: Hazard Notification



12.8.2: Email link to reset password



12.8.3: App Navigation Bar

13. Demo video

https://youtu.be/W_SvEwXV2a0

14 Testing

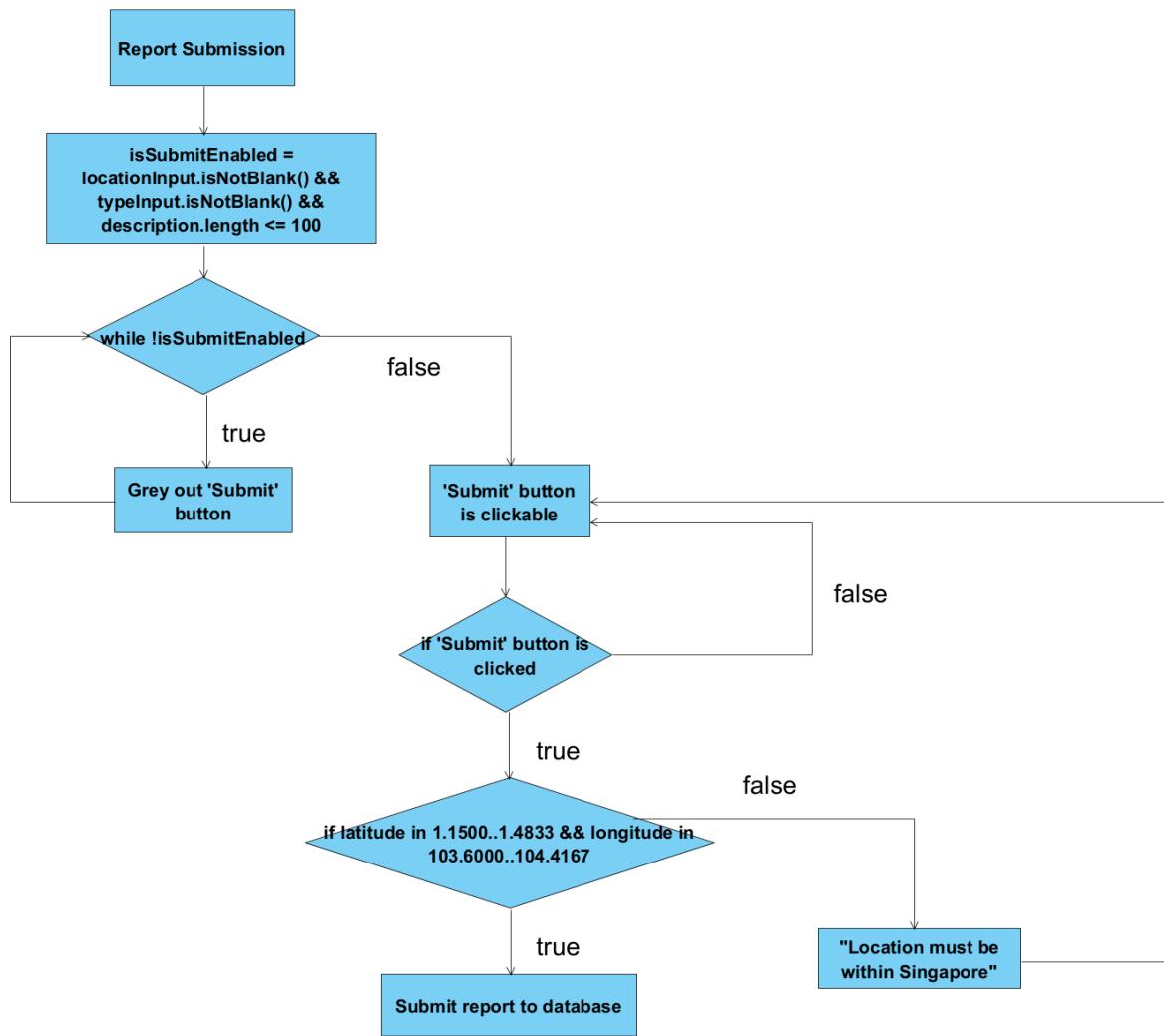
14.1 Test Cases

ID	Test case	Test Input	Expected Output	Actual Output
U01	New user can successfully register	Email: roadguardian39@gmail.com Password: SC2006Group39@	User is able to register	User is able to register
U02	User registration with an email that already exists	Email: roadguardian39@gmail.com Password: SC2006Group39@	User is not able to register with email that already exist	User is not able to register with email that already exist
U03	Test if a registered user can log in	Email: roadguardian39@gmail.com Password: SC2006Group39@	Registered user is able to login	Registered user is able to login
U04	User login with an incorrect password	Email: roadguardian39@gmail.com Password: <u>wrongpassword</u>	User is not able to login with incorrect password	User is not able to login with incorrect password
U05	User login with unregistered email	Email: testingwrong@gmail.com Password: <u>thispassword</u>	User is not able to login with unregistered email	User is not able to login with unregistered email
M01	Display user current Location on Map	GPS enabled	User location is displayed on the map	User location is displayed on the map
M02	Hazard markers are displayed on the map	-	Hazard markers are shown on the map	Hazard markers are shown on the map

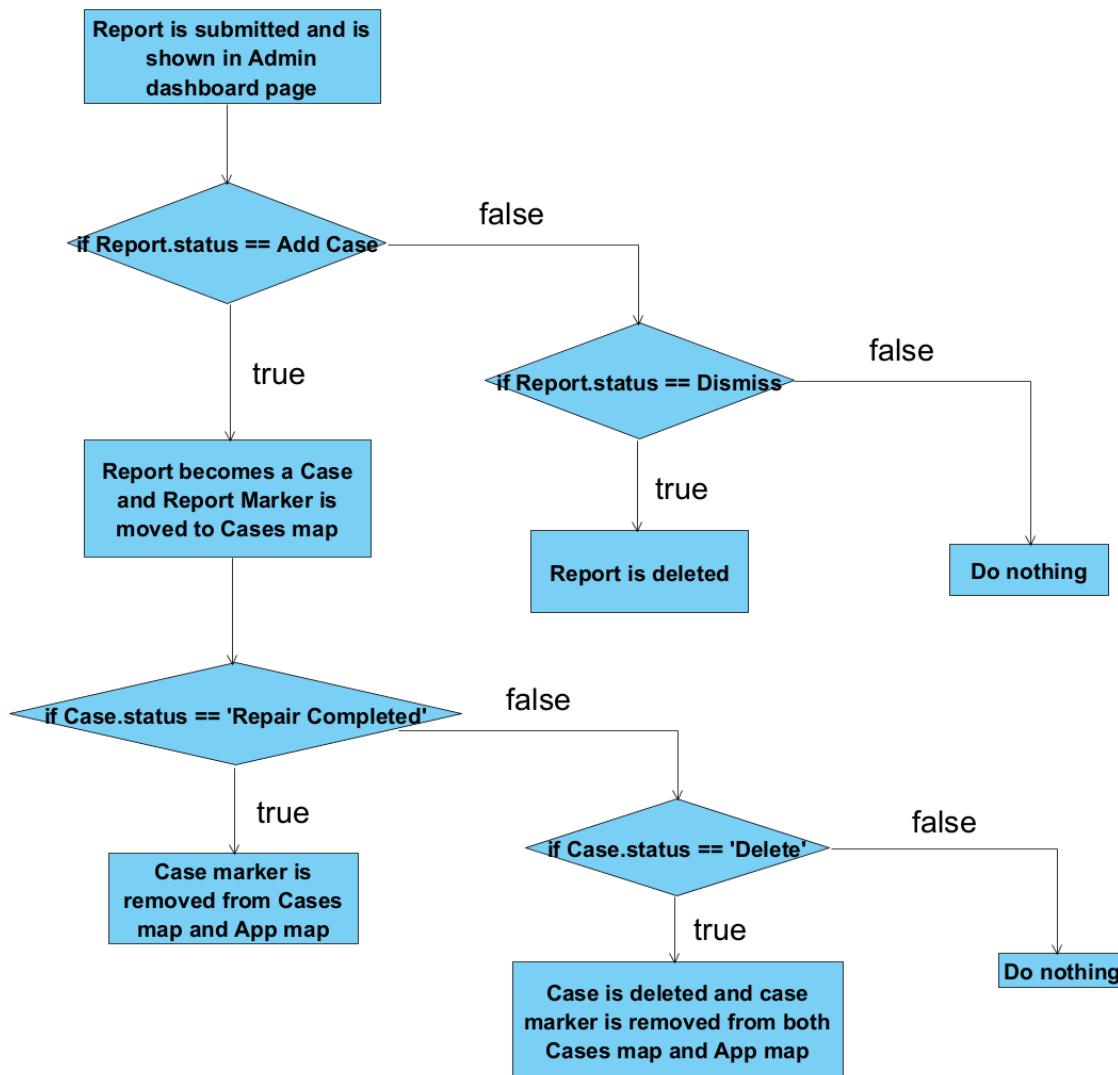
M03	Mobile is able to display notification	Hazard is within 500 feet from the user	Hazard Alert. You are close to a hazard location	Hazard Alert. You are close to a hazard location
H01	Submit Hazard Report with All Valid Data	Image: Photo Hazard type: Other Location: NTU Description: 'Shorter than 100'	Report submitted	Report submitted
H02	Submit Hazard Report without Hazard Type	Image: Photo Hazard type: NOT SELECTED Location: NTU Description: 'Shorter than 100'	Unable to Submit Report, Button is disabled	Unable to Submit Report, Button is disabled
H03-1	Submit Hazard Report with Location outside North Singapore	Image: Photo Hazard type: Other Location: 1.49,103.8 Description: 'Shorter than 100'	Location cannot be outside of Singapore	Location cannot be outside of Singapore
H03-2	Submit Hazard Report with Location outside South Singapore	Image: Photo Hazard type: Other Location: 1.08,103.8 Description: 'Shorter than 100'	Location cannot be outside of Singapore	Location cannot be outside of Singapore
H03-3	Submit Hazard Report with Location outside East Singapore	Image: Photo Hazard type: Other Location: 1.35,104.5 Description: 'Shorter than 100'	Location cannot be outside of Singapore	Location cannot be outside of Singapore

			<p>A message 'Reports database updated' should appear'</p> <p>The item should appear in the case database and its map marker should be added.</p> <p>A message 'Cases database updated' should appear'</p>	<p>A message 'Reports database updated' should appear'</p> <p>The item should appear in the case database and its map marker should be added.</p> <p>A message 'Cases database updated' should appear'</p>
A07	Admin can remove an item from reports	'Status' drop down box selection 'Apply' button clicking	<p>The line item from the reports database and its map marker should be removed.</p> <p>A message 'Reports database updated' should appear'</p>	<p>The line item from the reports database and its map marker should be removed.</p> <p>A message 'Reports database updated' should appear'</p>
A08	Admin can mark an existing case as completed	'Status' drop down box selection 'Apply' button clicking	<p>The status column of the line item will be marked 'Repair Completed', and its map marker should be removed.</p>	<p>The status column of the line item will be marked 'Repair Completed', and its map marker should be removed.</p>
A09	Admin can set the date of repair of an existing case	'Date' field editing 'Apply' button clicking	<p>The date column of the line item will be updated with the newly input date</p>	<p>The date column of the line item will be updated with the newly input date</p>
A10	Admin can remove an existing case	'Status' drop down box selection 'Apply' button clicking	<p>The line item from the cases database and its map marker should be removed.</p> <p>A message 'Cases database updated' should appear'</p>	<p>The line item from the cases database and its map marker should be removed.</p> <p>A message 'Cases database updated' should appear'</p>
A101	Run a damage detector python script that detect road damages in traffic camera images from data.gov.sg and their hazard type	<ol style="list-style-type: none"> 1. Traffic camera images from data.gov.sg 2. Trained YOLOv7 model 	<p>Detected road damages and their images are uploaded to their detections database</p>	<p>Detected road damages and their images are uploaded to their detections database</p>

14.2 Control Flow Graph



14.2.1: Control Flow Diagram for Hazard Reporting



14.2.2: Control Flow Diagram for managing hazard cases

15. Other Requirements

15.1 International Requirements

- Currently only supports English Language. Future release will consider multi-language such as Chinese, Tamil and Malay so that wider audience in Singapore who does not understand English is able to use this application
- Date and Time is in the Singapore time zone (GMT +8)

15.2 Legal Requirement

- Compliance with PDPA when handling user data

15.3 Reusability

- Application uses MVC approach allowing application to be portable to another system easily
- APIs used in the application are designed with industry standard facilitate external usage of our platform