

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

SC2006 – Software Engineering

Lab 1 Deliverables

Lab Group	SCSI-30
Team	TEAM 5
Members	Tan Wei Song Russell Bryan (U2421844K)
	Justin Woon Thean Woon (U244641B)
	Ethan Jared Chong Rui Zhi (U2421895B)
	Evelyn Theresia Cuaca (U2320523L)
	Parvez Kurniawan Wijaya (U2423845G)

Table of Contents

- 1. Documentation of functional and non-functional requirements**
 - A. Functional Requirements
 - B. Non-Functional Requirements
- 2. Data dictionary**
- 3. Initial Use Case Model, consisting of Use Case diagram and Use Case descriptions**
 - A. Use Case Diagram
 - B. Use Case Descriptions
 - I. For Functional Requirement #1
 - I.I SignUp
 - I.II Login
 - I.III AuthenticateUser
 - II. For Functional Requirement #2
 - II.I ManageSavedPlaces
 - II.II ViewMap
 - II.III SearchDestination
 - II.IV SearchSuggestion
 - II.V CompareTransportation
 - II.VI Report
 - II.VII SuggestRoute
 - III. For Functional Requirement #3
 - III. I View Driving Metrics
 - III. II View Public Transport Metrics
 - III. III View Walking Metrics
 - III. IV View Cycling Metrics
 - III. V Bike Rental Information
 - IV. For Functional Requirement #4
 - IV.I Fetch Traffic Data
 - IV.II Manage Account User
 - IV.III ManageObstructionReport
 - IV.IV ManageSystemReport
 - IV.V Manage Suggested Route
 - V. For Functional Requirement #5
 - V.I External System Integration
- 4. UI Mockups**
 - Login
 - Signup
 - User Profile
 - Home Screen
 - Compare modes of travel
 - Destination page (Driving) and Destination page (Walking)

Destination page (Cycling) and Destination page (Public Transport)

Nearest Carpark with Availability

Nearest Bicycle Rental

User suggestions/contributions

User Incident Report

User Saved/ Favourites

Suggested Route by Users Forum

Approved Suggested Routes

Admin Manage Page

Admin User Management Page

Admin Forum Management Page

Admin Manage Cost Models Page

Admin Manage System Reports Page

Admin Manage Obstruction Reports Page

Admin System Health Dashboard

1. Documentation of functional and non-functional requirements

A. Functional Requirements

1. TripTally shall authenticate Users and Admins and authorize access to the application
 - 1.1. TripTally shall allow Users to input their name, email, password and contact number for the creation of a User account with User privileges.
 - 1.1.1. If the User inputs an email already attached to an account, Trip Tally shall inform the User to log in.
 - 1.1.2. If the User inputs a contact number already attached to an account, Trip Tally shall inform the User to log in.
 - 1.1.3. The name inputted by the User shall not exceed 512 characters.
 - 1.1.4. The email inputted by the User shall not exceed 254 characters.
 - 1.1.5. The password entered by the User must be complex enough for security purposes.
 - 1.1.5.1. The password entered by the User must have at least 8 characters.
 - 1.1.5.2. The password entered by the User must have at least 1 alphabet.
 - 1.1.5.3. The password entered by the User must have at least 1 number.
 - 1.1.5.4. The password entered by the User must have at least 1 special character.
 - 1.1.6. If TripTally detects no internet, TripTally shall prompt the User to retry the process of creating an account.
 - 1.2. TripTally shall allow Users to have the option to create an account through Apple ID.
 - 1.2.1. If the User has previously created an account through Apple ID, then TripTally shall allow the User to log in using Apple ID.
 - 1.3. TripTally shall allow Users to have the option to create an account through Google.
 - 1.3.1. If the User has previously created an account through Google, then TripTally shall allow the User to log in using Google.
 - 1.4. If the User has previously created an account using his email, he must enter his email together with his matching password for the account to log in successfully.
 - 1.4.1. Users must enter the correct password within five tries.

- 1.4.2. Upon the fifth incorrect attempt, the system shall lock the account for one hour before permitting additional login attempts.
- 1.5. If the User has previously created an account using his contact number, he must enter his contact number together with his matching password for the account to log in successfully.
 - 1.5.1. Users must enter the correct password within five tries.
 - 1.5.2. Upon the fifth incorrect attempt, the system shall lock the account for one hour before permitting additional login attempts.
- 1.6. TripTally shall allow Admins to input their name, email, password and contact number for the creation of a Admin account with Admin privileges.
 - 1.6.1. If the Admin inputs an email already attached to an account, Trip Tally shall inform the User to log in.
 - 1.6.2. If the Admin inputs a contact number already attached to an account, Trip Tally shall inform the User to log in.
 - 1.6.3. The name inputted by the Admin shall not exceed 512 characters.
 - 1.6.4. The email inputted by the Admin shall not exceed 254 characters.
 - 1.6.5. The password entered by the Admin must be complex enough for security purposes.
 - 1.6.5.1. The password entered by the Admin must have at least 8 characters.
 - 1.6.5.2. The password entered by the Admin must have at least 1 alphabet.
 - 1.6.5.3. The password entered by the Admin must have at least 1 number.
 - 1.6.5.4. The password entered by the Admin must have at least 1 special character.
 - 1.6.6. If TripTally detects no internet, TripTally shall prompt the Admin to retry the process of creating an account.
- 1.7. The Admin shall have the option to create an account through Apple ID.
 - 1.7.1. If the Admin has previously created an account through Apple ID, then TripTally shall allow the Admin to log in using Apple ID.
- 1.8. The Admin shall have the option to create an account through Google.
 - 1.8.1. If the Admin has previously created an account through Google, then TripTally shall allow the Admin to log in using Google.
- 1.9. If the Admin has previously created an account using his email, he must enter his email together with his matching password for the account to log in successfully.
 - 1.9.1. Admins must enter the correct password within five tries.

- 1.9.2. Upon the fifth incorrect attempt, the system shall lock the account for one hour before permitting additional login attempts.
 - 1.10. If the Admin has previously created an account using his contact number, he must enter his contact number together with his matching password for the account to log in successfully.
 - 1.10.1. Admins must enter the correct password within five tries.
 - 1.10.2. Upon the fifth incorrect attempt, the system shall lock the account for one hour before permitting additional login attempts.
2. TripTally shall allow Users to perform user-specific tasks.
 - 2.1. TripTally shall allow Users to search for routes to their destination.
 - 2.1.1. TripTally shall have a search bar for Users to search for their destination.
 - 2.1.2. TripTally shall display the user's last 5 distinct searches in the search menu.
 - 2.1.3. TripTally shall show up to 10 suggested places updated at every keystroke based on the current query and current trends.
 - 2.1.4. TripTally shall provide a list of different modes of transportation to the destination, with the time taken for each mode next to it.
 - 2.1.4.1. The time taken shall be displayed as integers in minutes or hours.
 - 2.1.4.2. The list of different modes of transportation shall include driving.
 - 2.1.4.3. The list of different modes of transportation shall include motorbiking.
 - 2.1.4.4. The list of different modes of transportation shall include public transportation.
 - 2.1.4.5. The list of different modes of transportation shall include walking.
 - 2.1.4.6. The list of different modes of transportation shall include cycling.
 - 2.1.5. TripTally shall provide a list of suggestions for the User at any point in their search.
 - 2.1.6. TripTally shall provide a preview of the route on the map once the destination is confirmed.
 - 2.1.7. If no route exists for a mode, TripTally shall display that mode with "Not available" instead of an ETA.
 - 2.1.8. At each decision point (turn, merge, exit, roundabout, or transit transfer), TripTally shall inform the user on a maneuver instruction.
 - 2.1.8.1. The instruction shall include the action.
 - 2.1.8.2. The instruction shall include the next street/stop name.
 - 2.1.8.3. The instruction shall include the distance in meters.
 - 2.2. At least 75% of the trips that TripTally estimated shall be within 10% or 2 minutes of the actual time to take the trip, whichever is larger.

- 2.3. TripTally shall provide a map control that, when activated, re-centers the map based on the user's current location.
- 2.4. TripTally shall allow Users to view certain metrics of driving or motorbiking to their destination.
 - 2.4.1. TripTally shall estimate the total travel time to the destination selected based on traffic jams, road closures et cetera.
 - 2.4.1.1. The time taken shall be displayed as integers in minutes or hours.
 - 2.4.2. TripTally shall calculate ERP charges to be incurred by the User by analysing the time of travel and the route being taken.
 - 2.4.3. TripTally shall calculate parking charges to be incurred by the User at their destination carpark of choice.
 - 2.4.4. TripTally shall calculate the cost of fuel required to propel the User's vehicle to the destination.
 - 2.4.5. TripTally shall calculate cost of fuel for a round trip (if selected by User).
 - 2.4.6. TripTally shall calculate an estimate, of the User's carbon footprint in choosing to drive to their destination
 - 2.4.7. TripTally shall add up all three aforementioned monetary costs and display it as a metric named "Calculated Total Cost" for the User's reference.
 - 2.4.8. TripTally shall provide a full breakdown of the total cost value upon request from the User.
- 2.5. TripTally shall allow Users to view certain metrics of taking public transportation to their destination.
 - 2.5.1. TripTally shall estimate the total travel time to the destination selected based on waiting times, transfer times, closures et cetera.
 - 2.5.1.1. The time taken shall be displayed as integers in minutes or hours.
 - 2.5.2. TripTally shall calculate total fare cost of the trip and present it to the User.
 - 2.5.3. TripTally shall provide a breakdown on the total cost value upon request from the User.
 - 2.5.4. TripTally shall calculate an estimate of the User's carbon footprint in choosing to take public transport to their destination.
- 2.6. TripTally shall allow Users to view certain metrics of walking to their destination.
 - 2.6.1. TripTally shall estimate the total travel time to the destination selected.
 - 2.6.1.1. The time taken shall be displayed as integers in minutes or hours.
 - 2.6.2. TripTally shall show the User their reduction in carbon footprint as compared to public transportation and driving.

- 2.6.3. TripTally shall show the User the money they save as compared to taking public transportation or driving.
 - 2.6.4. TripTally shall provide an estimation for the amount of calories that the User will burn in choosing to walk to their destination.
- 2.7. TripTally shall allow Users to view certain metrics of cycling to their destination.
- 2.7.1. TripTally shall estimate the total travel time to the destination selected.
 - 2.7.1.1. The time taken shall be displayed as integers in minutes or hours.
 - 2.7.2. TripTally shall show the User their reduction in carbon footprint as compared to public transportation and driving.
 - 2.7.3. TripTally shall show the User the money they save as compared to taking public transportation or driving.
 - 2.7.4. TripTally shall provide an estimation for the amount of calories that the User will burn in choosing to cycle to their destination.
- 2.8. TripTally shall allow Users to compare various modes of transportation to their destination.
- 2.8.1. When a User finalises their destination and selects a mode of transportation, the User shall be able to compare this mode of transportation to another mode of his choice.
 - 2.8.2. When this other mode of transportation is selected, TripTally shall show all the relevant metrics side-by-side with the currently selected mode of transportation.
 - 2.8.3. The cost and carbon footprint metrics shall be colour-coded in the comparison to better show the differences between the modes of transportation.
 - 2.8.4. TripTally shall recommend a mode of transportation to the User, factoring in cost, time taken, time of day, availability et cetera.
 - 2.8.5. TripTally shall display the recommendation as a non-intrusive label on the recommended mode of transportation's icon.
- 2.9. TripTally shall allow Users to provide feedback on app-generated routes.
- 2.9.1. TripTally shall allow Users to tap/click on their map to indicate an obstruction/warning.
 - 2.9.2. Upon tapping the spot on their map, a menu of icons that correspond to various different obstructions/warnings shall pop up for the User to select.
 - 2.9.3. TripTally shall allow other Users passing by the area to verify that the obstruction/warning is still there.
 - 2.9.4. TripTally shall allow Users to suggest better/alternate walking/cycling routes.
 - 2.9.5. Suggested routes shall be subject to approval from Admins before being published to other Users

- 2.9.6. TripTally shall allow Users to provide pictures or videos of the suggested walking/cycling routes.
 - 2.9.7. TripTally shall allow Users to report discrepancies in the calculation of all available cost metrics (fare cost, parking cost et cetera).
- 2.10. TripTally shall have a forum page with User-submitted routes.
- 2.10.1. TripTally shall have a page for User-submitted walking and cycling routes.
 - 2.10.2. TripTally shall allow Users submitting routes to add various tags; for example “time-saving” or “sheltered” et cetera.
 - 2.10.3. TripTally shall allow the route-submitting User to write a description about the route.
 - 2.10.4. TripTally shall show a verification mark on routes that have been verified by Admins.
 - 2.10.5. TripTally shall allow Users perusing the forum page to filter their results by tags.
 - 2.10.6. TripTally shall allow Users perusing the forum page to filter their results by search terms.
 - 2.10.7. TripTally shall allow Users perusing the forum page to filter their results by verification status.
 - 2.10.8. TripTally shall provide a search function for Users perusing the forums to use keywords to find routes.
 - 2.10.9. TripTally shall provide Users with the ability to “like” on User-submitted routes.
 - 2.10.10. TripTally shall provide Users with the ability to “comment” on User-submitted routes.
 - 2.10.11. TripTally shall allow Users to load User-submitted routes onto their map for navigation.
3. TripTally shall allow Admins to perform admin-specific and moderative tasks.
- 3.1. TripTally shall allow Admins to manage forum posts.
 - 3.1.1. TripTally shall allow Admins to view all User-submitted walking/cycling route suggestions.
 - 3.1.2. TripTally shall check User-submitted routes for accuracy before they can be verified by Admins.
 - 3.1.3. TripTally shall check that User-submitted routes include correct tags and descriptions before they can be verified by Admins.
 - 3.1.4. TripTally shall allow Admins to verify User-submitted routes once all required criteria have been passed.
 - 3.1.5. TripTally shall allow Admins to remove User-submitted routes with a provided valid reason.

- 3.1.6. TripTally shall allow Admins to remove comments that violate fair use guidelines.
 - 3.1.7. TripTally shall allow Admins to edit user suggested routes if necessary.
 - 3.2. TripTally shall allow Admins to manage obstructions and warning reports.
 - 3.2.1. TripTally shall allow Admins to view all obstructions and warnings reported by Users.
 - 3.2.2. TripTally shall allow Admins to verify or dismiss the reported obstructions.
 - 3.2.3. TripTally shall allow Admins to flag false obstruction reports.
 - 3.2.4. TripTally shall allow Admins to remove false obstruction reports.
 - 3.2.5. TripTally shall allow Admins to mark obstruction reports as “resolved” once the incidents have been resolved.
 - 3.3. TripTally shall allow Admins to manage user accounts.
 - 3.3.1. TripTally shall allow Admins to view all registered user accounts.
 - 3.3.2. TripTally shall allow Admins to suspend user accounts.
 - 3.3.3. TripTally shall allow Admins to deactivate user accounts permanently if required.
 - 3.4. TripTally shall allow Admins to manage cost models and external data.
 - 3.4.1. TripTally shall allow Admins to update the ERP charge information.
 - 3.4.2. TripTally shall allow Admins to update parking rate information.
 - 3.4.3. TripTally shall allow Admins to update fuel price information.
 - 3.4.4. TripTally shall allow Admins to update public transportation fare rates.
 - 3.4.5. TripTally shall allow Admins to update carbon footprint calculation factors
 - 3.4.6. TripTally shall allow Admins to view logs of external API errors.
 - 3.5. TripTally shall allow Admins to monitor technical failures/difficulties.
 - 3.5.1. TripTally shall allow Admins to view error logs from external system errors.
 - 3.5.2. TripTally shall allow Admins to issue warnings to Users about potential issues.
4. TripTally shall be integrated with external systems to provide accurate and updated route, cost, carpark availability, incident reports and environmental information.
- 4.1. TripTally shall be integrated with mapping and geocoding services/APIs.
 - 4.1.1. TripTally shall use external map providers such as Google Maps to geocode user-entered destinations.
 - 4.1.2. TripTally shall use external routing services to retrieve available routes.
 - 4.1.3. TripTally shall use external mapping APIs to render route previews on the map for users.

- 4.2. TripTally shall integrate with traffic and road closure/incidents alert systems.
 - 4.2.1. TripTally shall receive real-time traffic conditions from external traffic APIs.
 - 4.2.2. TripTally shall retrieve information on road closures, accidents and diversions.
 - 4.2.3. TripTally shall update estimated driving times based on data of traffic conditions of the road and any road incidents.
 - 4.2.4. TripTally shall integrate with external systems for traffic conditions, accidents and road blockage reports to enhance route reliability and efficiency.
- 4.3. TripTally shall integrate with the ERP system.
 - 4.3.1. TripTally shall retrieve ERP gantry location from external ERP providers.
 - 4.3.2. TripTally shall retrieve ERP pricing tables by time of the day and day of the week.
 - 4.3.3. TripTally shall calculate total ERP costs for a given driving route using the retrieved data (ERP prices and time).
- 4.4. TripTally shall integrate with parking information systems.
 - 4.4.1. TripTally shall retrieve parking rate information from the different carparks around Singapore from external systems.
 - 4.4.2. TripTally shall retrieve real-time carpark availability data from external systems.
 - 4.4.3. TripTally shall display carpark lots availability to the users for the various carparks around Singapore.
 - 4.4.4. TripTally shall update parking charges dynamically if there are any changes to parking availability and rates.
- 4.5. TripTally shall integrate with fuel pricing systems.
 - 4.5.1. TripTally shall retrieve current fuel prices from external fuel price providers.
 - 4.5.2. TripTally shall update route cost calculations for drivers based on the retrieved fuel data of their selected vehicle.
- 4.6. TripTally shall integrate with carbon footprint calculation services.
 - 4.6.1. TripTally shall retrieve emission factors for each mode of transportation.
 - 4.6.2. TripTally shall calculate the estimated carbon footprint for each trip using the retrieved data.
 - 4.6.3. TripTally shall update emission factors when new values are provided by the external systems.

- 4.7. TripTally shall integrate with public transportation systems.
 - 4.7.1. TripTally shall retrieve bus and train schedules from external public transport APIs.
 - 4.7.2. TripTally shall retrieve fare tables from external public transport fare providers.
 - 4.7.3. TripTally shall calculate total public transport fares using the retrieved fare tables.
 - 4.7.4. TripTally shall calculate total public transport journey times using retrieved schedule data.
- 4.8. TripTally shall ensure data reliability from external systems.
 - 4.8.1. TripTally shall retry failed EPI calls automatically with exponential backoff.
 - 4.8.2. TripTally shall use fallback values from historical data when external systems are unavailable.
 - 4.8.3. TripTally shall log all external system errors for review by the Admins.
- 4.9. TripTally shall shortcut to various ride-hailing services.
 - 4.9.1. TripTally shall have a taxi ride option that shows options for various ride-hailing apps such as Grab and CDG.
 - 4.9.2. TripTally shall set the travel time for ride-hailing trips equal to the estimated driving time.
 - 4.9.3. Upon pressing one of the ride-hailing options, the User shall be redirected to the app (if they have it installed) or the app's store page.
- 4.10. If any external providers fails or times out (≥ 15 seconds), TripTally shall show "Not available" for that provider and keep others.

B. Non-Functional Requirements

Usability	<p>Responsive User Interface</p> <ul style="list-style-type: none">• Actions respond in < 5 seconds.• Users can navigate and complete tasks with a response time of < 5 seconds per step. <p>Mobile Responsiveness</p> <ul style="list-style-type: none">• Users must be able to view 100% of the contents regardless of what mobile phones they use. <p>Browser Compatibility</p> <ul style="list-style-type: none">• The system shall support the latest 2 major versions of Chrome and Safari. <p>Different Languages</p> <ul style="list-style-type: none">• Content must be able to be displayed in the four main languages used in Singapore (English, Chinese, Malay, Tamil). <p>Iconography</p> <ul style="list-style-type: none">• Each icon must be correctly interpreted by at least 85% of users within 5 seconds when shown without text.
Reliability	<p>Idempotent Actions</p> <ul style="list-style-type: none">• Repeated API calls from network retries must not duplicate submissions or charges.

Performance	<p>Map Interactivity</p> <ul style="list-style-type: none"> The system shall maintain ≥ 30 FPS when zooming in and out of the map on a normal mobile device. <p>Server Capacity</p> <ul style="list-style-type: none"> System shall sustain ≥ 500 concurrent active users. <p>Caching and Prefetch</p> <ul style="list-style-type: none"> Repeat search hit cache $\geq 70\%$. The system shall prefetch alternate routes after the first result without blocking UI.
Supportability	<p>Zero Downtime Deploys</p> <ul style="list-style-type: none"> New versions shall be deployed without taking the application offline. <p>Versioning</p> <ul style="list-style-type: none"> Public APIs must be versioned. Last 3 versions must remain supported. <p>Logging</p> <ul style="list-style-type: none"> Errors must be logged with time and request ID. Sensitive data must never written to logs.

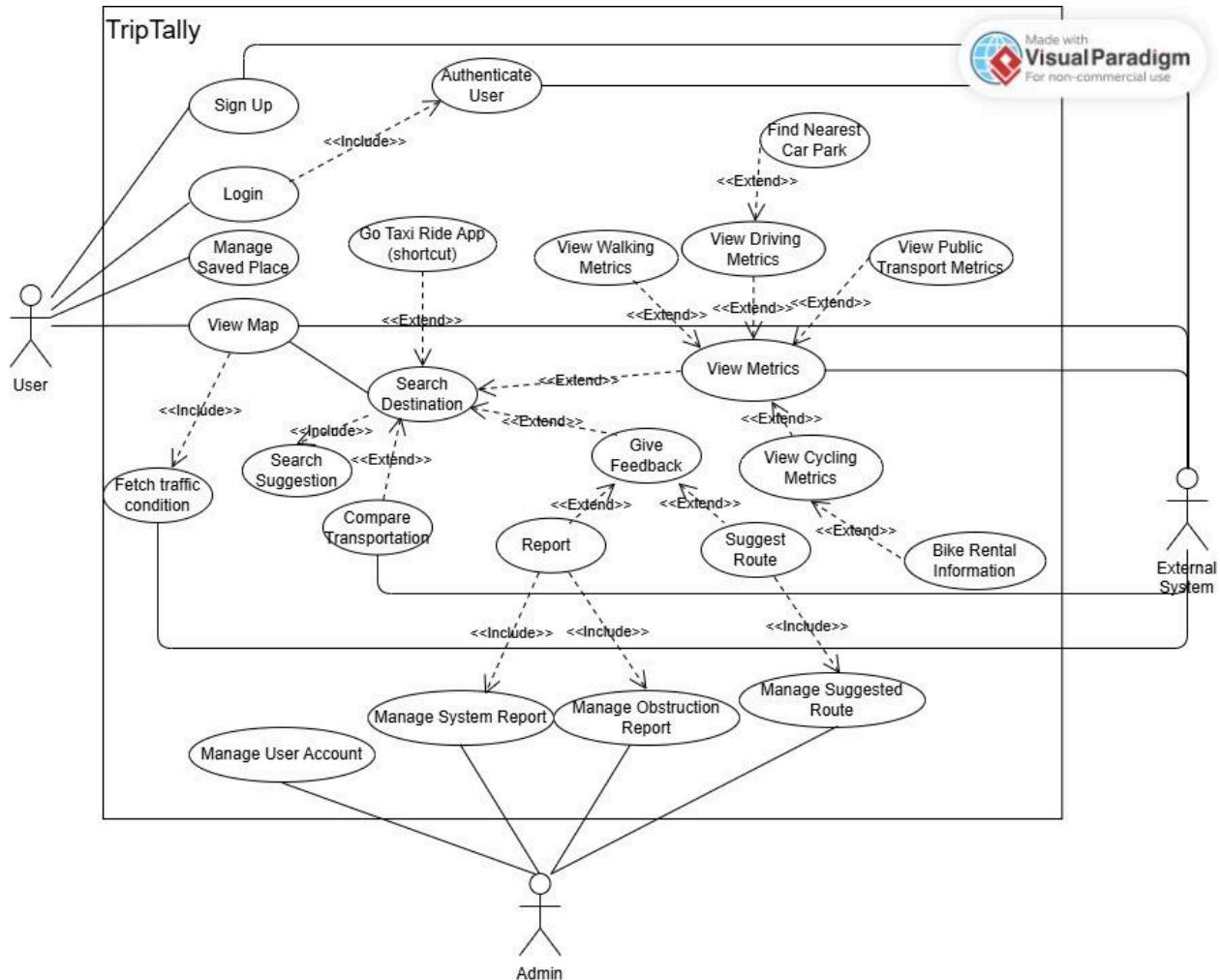
2. Data dictionary

Term	Definition
User	A registered individual who uses TripTally to search, plan and compare routes.
Admin	System administrator who reviews reports, approves routes and manages user accounts and any updates or discrepancies.
Special Character	A non-alphanumeric character that can be typed on a computer keyboard.
External Systems	Third party providers that TripTally relies on such as Map APIs, Traffic Data, Public Transport Data, Google and Apple login APIs etc.
Destination	The end location chosen by the user for their journey.
Account Lock	Temporary blocking of access into the account after multiple failed login attempts.
Route Preview	A pathway on the visual map showing the suggested route to the destination.
Driving Time	Estimated travel time for driving that takes into account current traffic conditions and obstructions.
ERP Charges	Electronic Road Pricing fees incurred along the chosen driving route.
Fuel Cost	Estimated fuel usage to complete the journey based on current fuel prices.
Cost Breakdown	Detailed list of all the charges such as ERP charges, parking and fuel cost that makes up the total cost.
Walking Time	Estimated duration taken to reach the destination on foot.
Cycling Time	Estimated time required to cycle to the destination.
Money Saved	Difference in cost compared to the different modes of transportation (Public Transport, Driving, Walking, Cycling)

Compare Button	Functionality that allows users to compare metrics between 2 transport modes.
Incident Report	User submitted reports on road closures or traffic accidents.
Route Approval	Admin review process to validate user suggested routes before approving or rejecting it.

3. Initial Use Case Model, consisting of Use Case diagram and Use Case descriptions

A. Use Case Diagram



B. Use Case Descriptions

I. For Functional Requirement #1

I.I SignUp

Use Case ID:	#1-1		
Use Case Name:	SignUp		
Created By:	Russell Tan	Last Updated By:	Russell Tan
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	User		
Description:	Creation of a new TripTally account of a user		
Preconditions:	<ol style="list-style-type: none">App must be installed and the user's email/contact number has not been registered.Internet connection must be available.		
Postconditions:	<ol style="list-style-type: none">New account created and verified.Confirmation message sent to the users email.		
Priority:	Low		
Frequency of Use:	High		
Flow of Events:	<ol style="list-style-type: none">User selects Sign UpUser enters name, email, password and contact number. User can also sign up through Apple ID or Google.System validates that the email exists and is valid and checks the password strength.User agrees to the Terms and Services of the application.Account is then created and confirmation is sent to the email that has created the account.		
Alternative Flows:	AF-1: Sign up available with Apple/Google.		
Exceptions:	EX-1: If there is a duplicate email account, system will prompt that account already exists		

	<p>and prompts user login</p> <p>EX-2: A weak password would result in the system prompting to change the password to have at least 8 characters with a mix of alphabets, numbers and special characters.</p> <p>EX-3: No internet would result in a retry option.</p>
Includes:	<ol style="list-style-type: none"> 1. Login 2. AuthenticateUser
Special Requirements:	Passwords must be a minimum of length 8, containing both alphabets, numbers and at least one special character.
Assumptions:	None
Notes and Issues:	None

I.II Login

Use Case ID:	#1-2		
Use Case Name:	Login		
Created By:	Russell Tan	Last Updated By:	Russell Tan
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	User		
Description:	Start a user session with email and the corresponding password.		
Preconditions:	<ol style="list-style-type: none"> 1. User has a registered account and clicks the Login button. 2. Internet connection must be available 		
Postconditions:	<ol style="list-style-type: none"> 1. User has been authenticated and secure session active 		
Priority:	Low		
Frequency of Use:	Low		
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks Login button. 2. User enters email/contact number with the correct password. 3. AuthenticateUser is called. 		

	4. User is redirected to the Home page of the application if AuthenticateUser is successful.
Alternative Flows:	<p>AF-1: Wrong password/ username</p> <ul style="list-style-type: none"> • If User enters their password incorrectly for 5 tries, their account would be locked for an hour. • Whenever the User gets their credentials wrong before exceeding 5 tries, a message saying “invalid login/password” will be displayed instead. • The system will prompt user to re-enter their username and password <p>AF-2: Network issues would bring back the user to the sign in page.</p>
Exceptions:	None
Includes:	AuthenticateUser
Special Requirements:	None
Assumptions:	Assume user enters the right password on the first try
Notes and Issues:	None

I.III AuthenticateUser

Use Case ID:	#1-3		
Use Case Name:	AuthenticateUser		
Created By:	Russell Tan	Last Updated By:	Russell Tan
Date Created:	01/09/2025	Date Last Upd ated:	01/09/2025
Actor:	ExternalSystem		
Description:	Process of verifying that the user's account exists and matches the password before logging them in.		
Preconditions:	1. Login has been initiated. 2. Internet connection must be available.		
Postconditions:	1. User securely authenticated and login successful.		
Priority:	High		

Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. System receives the login credentials from the User. 2. System fetches data matching the account. 3. System checks the status of the account and ensures that it is not locked. 4. System then verifies the user's password. 5. System resets the retry fail count. 6. System returns a true and user is authenticated
Alternative Flows:	AF-1: If the account is locked, system displays "Your account has been locked" to the User.
Exceptions:	EX-1: If user cancels at the account selection, display a retry option
Includes:	<ol style="list-style-type: none"> 1. SignUp 2. Login
Special Requirements:	None
Assumptions:	Account that the user is logging in exists.
Notes and Issues:	None

II. For Functional Requirement #2

II.I ManageSavedPlaces

Use Case ID:	#2-1		
Use Case Name:	ManageSavedPlaces		
Created By:	Russell Tan	Last Updated By:	Russell Tan
Date Created:	01/09/2025	Date Last Upd ated:	01/09/2025
Actor:	User		
Description:	User have the ability to manage their saved locations. (Home, Work, Favourites)		
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged in. 2. Internet connection must be available 		

Postconditions:	1. User has successfully added/edited/deleted the saved location.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. User selects the “Saved Places” button. 2. User is able to add, edit or delete the location selected by the user. 3. System updates database and syncs profile. 4. Newly added, edited or deleted location is displayed for the user.
Alternative Flows:	AF-1: Duplicate places shall prompt a rename of the location to the user.
Exceptions:	EX-1: If invalid location is selected, an error will be displayed and bring the user back to the home page.
Includes:	None
Special Requirements:	None
Assumptions:	User logged in.
Notes and Issues:	None

II.II ViewMap

Use Case ID:	#2-2		
Use Case Name:	ViewMap		
Created By:	Russell Tan	Last Updated By:	Russell Tan
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	User, ExternalSystem		
Description:	Displays interactive map that is centered on the user's location.		
Preconditions:	<ol style="list-style-type: none"> 1. User is logged in. 2. Internet connection must be available. 		

Postconditions:	1. Map is displayed with the current location displayed to the user.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. User opens the home page/ clicks on the explore button. 2. System fetches current location and Maps API. 3. System displays an interactive map with context.
Alternative Flows:	AF-1: User can manually input their address when GPS is off.
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	User mobile devices have granted permission to view the user's current location.
Notes and Issues:	None

II.III SearchRouteToDestination

Use Case ID:	#2-3		
Use Case Name:	SearchDestination		
Created By:	Russell Tan	Last Updated By:	Russell Tan
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	User, ExternalSystem		
Description:	Displays the selected location to the user.		
Preconditions:	<ol style="list-style-type: none"> 1. Origin and Destination must be known. 2. Internet connection must be available. 		

Postconditions:	<ol style="list-style-type: none"> 1. The selected location is displayed on the map for the user with the information regarding the location.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. System displays cached recent searches. 2. User keys in their destination in the search bar. 3. System queries geolocation API to search for the destination. 4. Location is displayed with its information and directions for the user. 5. Users can then select mode of transportation for the directions to their destination. 6. Route calculation is triggered. 7. Multiple routes to the users destination is displayed
Alternative Flows:	AF-1: Destination picked is from recent or Saved Places
Exceptions:	EX-1: Geocoding API is down. EX-2: Invalid location is inputted.
Includes:	<ol style="list-style-type: none"> 1. ViewMap
Special Requirements:	None
Assumptions:	Interactive map is loaded.
Notes and Issues:	None

II.IV SearchSuggestion

Use Case ID:	#2-4		
Use Case Name:	SearchSuggestion		
Created By:	Russell Tan	Last Updated By:	Russell Tan
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	User, ExternalSystems		
Description:	Display live search suggestions as the user types into the search bar.		
Preconditions:	<ol style="list-style-type: none"> 1. User has keyed in a location in the search bar. 2. Internet connection must be available. 		
Postconditions:	<ol style="list-style-type: none"> 1. Search options displayed for users to select. 		
Priority:	High		
Frequency of Use:	High		
Flow of Events:	<ol style="list-style-type: none"> 1. User keys in a destination in the search bar. 2. System queries an external search API. 3. Suggestions based on the user's search are displayed. 4. User is able to select one of the suggestions that is displayed. 		
Alternative Flows:	None		
Exceptions:	EX-1: Location/API unavailable would display a "No match/ Try again" text to the user.		
Includes:	<ol style="list-style-type: none"> 1. ViewMap 2. SearchDestination 		
Special Requirements:	None		
Assumptions:	User has keyed in a valid location in the search bar.		
Notes and Issues:	None		

II.V CompareTransportation

Use Case ID:	#2-5		
Use Case Name:	CompareTransportation		
Created By:	Russell Tan	Last Updated By:	Russell Tan
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	User, ExternalSystem		
Description:	Side by side comparison of two modes of transportation. (Time, Cost, Carbon Footprint)		
Preconditions:	<ul style="list-style-type: none"> 1. Origin and destination must be known. 2. Internet connection must be available 		
Postconditions:	<ul style="list-style-type: none"> 1. User able to see side by side comparison of the 2 modes of transportation selected. 		
Priority:	High		
Frequency of Use:	High		
Flow of Events:	<ul style="list-style-type: none"> 1. User select the 2 modes of transportation that they would like to compare. 2. System fetches data of the different modes of transportation from their various APIs. 3. System compares the selected modes of transportation (Driving, Public Transport, Walking, Cycling) 4. System calculates the ETA, distance, cost and carbon footprint. 5. This data is then displayed to the user for comparison. 		
Alternative Flows:	<ul style="list-style-type: none"> 1. AF-1: User is able to view a brief comparison of all 4 metrics if the compare button is clicked without any selection of the 2 modes. 		
Exceptions:	EX-1: If one mode is down due to the API being down, display historical data and inform users that data is based on past information.		
Includes:	<ul style="list-style-type: none"> 1. ViewMap 2. SearchDestination 		
Special Requirements:	None		
Assumptions:	Origin and destination known.		
Notes and Issues:	None		

II.VI Report

Use Case ID:	#2-6		
Use Case Name:	Report		
Created By:	Russell Tan	Last Updated By:	Russell Tan
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	User		
Description:	User can submit report about incident on road or system error faced while using the app		
Preconditions:	1. The user is logged in. 2. Network is available.		
Postconditions:	1. User is saved in the system with "Submitted" status 2. Admin are automatically notified and the report is placed in the review queue		
Priority:	High		
Frequency of Use:	Medium		
Flow of Events:	1. User open submit report option 2. Systems prompt user to choose type of report (road incident/system error) 3. User select reporting road incident or system error 4. The system displays the form where users can describe the issue and upload relevant supporting material such as picture/video. 5. System validate, store the report and confirm submission. 6. System notify admin queue for review.		
Alternative Flows:	AF-1:Road incident report 1. Users input the location of the incident. 2. System prompt user to select incident categories. 3. Users can optionally add description and photos/videos. 4. Once submitted, report is saved with location, time, and marked as submitted 5. Admins are notified and incident is placed into the review queue AF-2:System error report 1. The User selects the category of error (fare calculation error, route error, data error etc.)		

	<ol style="list-style-type: none"> 2. The User is prompted to provide a description of the error. 3. User can optionally provide photo/video
Exceptions:	EX-1: Missing required information- System will highlight the required fields and prevent submission. EX-2: File upload error- System provide option to submit report without
Includes:	<ol style="list-style-type: none"> 1. ManageObstructionReport 2. ManageSystemReport
Special Requirements:	Status of Report includes: Submitted, Pending, Resolved
Assumptions:	None
Notes and Issues:	Verified incidents will affect the routing.

II.VII SuggestRoute

Use Case ID:	#2-7		
Use Case Name:	SuggestRoute		
Created By:	Russell Tan	Last Updated By:	Russell Tan
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	User		
Description:	User submits different walking /cycling routes from one location to another than the ones generated by the system.		
Preconditions:	<ol style="list-style-type: none"> 1. User is logged in to a valid account 2. Internet connection must be available 		
Postconditions:	1. User gets confirmation that their suggested route has been successfully uploaded.		
Priority:	Low		
Frequency of Use:	Low		

Flow of Events:	<ol style="list-style-type: none"> 1. The User selects the option to upload a new route. 2. The User selects from a range of tags that accurately describe the type of route they are uploading. 3. The User is then prompted to search for their start and end points of their destination using the search bar. 4. Once both points are confirmed, the User will tap on the screen to place “checkpoint” pins on the map to show exactly where the route will lead. 5. At each checkpoint the User will have the option to indicate whether the route will continue indoors or outdoors, and also upload helping images. 6. Once all the checkpoints and details are confirmed, the User is prompted to give the route a name and short written description. 7. The User will then tap on the “upload” button to upload the route to the forum.
Alternative Flows:	None
Exceptions:	EX-1: If the forum’s servers are down, return a message stating “The servers are currently down, please come back later”.
Includes:	<ol style="list-style-type: none"> 1. ViewMap 2. SearchDestination
Special Requirements:	None
Assumptions:	The User is not banned from posting routes.
Notes and Issues:	None

III. For Functional Requirement #3

III. I View Driving Metrics

Use Case ID:	#3-1		
Use Case Name:	ViewDrivingMetrics		
Created By:	Ethan Jared Chong Rui Zhi	Last Updated By:	Ethan Jared Chong Rui Zhi
Date Created:	1/8/2025	Date Last Updated:	1/8/2025

Actor:	User
Description:	User is shown metrics for driving to their destination.
Preconditions:	<ul style="list-style-type: none"> 1. User has searched for a destination and selected the “driving” mode of transportation.
Postconditions:	<ul style="list-style-type: none"> 1. The User gets shown the corresponding metrics for driving.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ul style="list-style-type: none"> 1. User searches for destination 2. User confirms destination 3. User selects “driving” as mode of transportation 4. Retrieve ERP pricing data, destination parking lot data (call FindNearestCarpark), fuel consumption data, fuel price data, vehicle CO2 emissions data, travel duration. 5. Calculate total cost. 6. Display drop-down table that shows CO2 emission, travel time and total cost.
Alternative Flows:	None
Exceptions:	<p>EX-1: If driving cannot reach the destination, the option to select “driving” will be disabled.</p> <p>EX-2: If some required information cannot be fetched, “note: error retrieving live data” message will be displayed in the corresponding table row and display historical data.</p>
Includes:	FindNearestCarpark
Special Requirements:	User's phone location services are enabled for TripTally.
Assumptions:	None
Notes and Issues:	None

III. II View Public Transport Metrics

Use Case ID:	#3-2		
Use Case Name:	ViewPublicTransportMetrics		
Created By:	Ethan Jared Chong Rui Zhi	Last Updated By:	Ethan Jared Chong Rui Zhi
Date Created:	1/8/2025	Date Last Updated:	14/8/2025
Actor:	User		
Description:	User is shown metrics for taking public transport to their destination.		
Preconditions:	<ul style="list-style-type: none"> 1. User has searched for a destination and selected the "public transport" mode of transportation. 		
Postconditions:	<ul style="list-style-type: none"> 1. The User gets shown the corresponding metrics for taking public transportation. 		
Priority:	High		
Frequency of Use:	High		
Flow of Events:	<ol style="list-style-type: none"> 1. User searches for destination 2. User confirms destination 3. User selects "public transport" as mode of transportation 4. Retrieve total trip fares, CO2 emissions data, travel duration. 5. Calculate total cost. 6. Display drop-down table that shows CO2 emission, travel time and total cost. 		
Alternative Flows:	None		
Exceptions:	<p>EX-1: If any required information is missing or invalid, an error message is displayed.</p> <p>EX-2: If public transportation cannot reach the destination, the option to select "public transportation" will be disabled.</p>		
Includes:	None		
Special Requirements:	User's phone location services are enabled for TripTally.		
Assumptions:	None		
Notes and Issues:	None		

III. III View Walking Metrics

Use Case ID:	#3-3		
Use Case Name:	View Walking Metrics		
Created By:	Ethan Jared Chong Rui Zhi	Last Updated By:	Ethan Jared Chong Rui Zhi
Date Created:	1/8/2025	Date Last Updated:	1/8/2025

Actor:	User
Description:	User is shown metrics for walking to their destination.
Preconditions:	<ul style="list-style-type: none"> 1. User has searched for a destination and selected the “walking” mode of transportation.
Postconditions:	<ul style="list-style-type: none"> 1. The User gets shown the corresponding metrics for walking to their destination.
Priority:	
Frequency of Use:	
Flow of Events:	<ul style="list-style-type: none"> 1. User searches for destination 2. User confirms destination 3. User selects “walking” as mode of transportation 4. Retrieve calories burnt and time taken 5. Display drop-down table that shows CO2 emission (zero), travel time and total cost (zero).

Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	User's phone location services are enabled for TripTally.
Assumptions:	None
Notes and Issues:	None

III. IV View Cycling Metrics

Use Case ID:	#3-4		
Use Case Name:	ViewCyclingMetrics		
Created By:	Ethan Jared Chong Rui Zhi	Last Updated By:	Ethan Jared Chong Rui Zhi
Date Created:	1/8/2025	Date Last Updated:	1/8/2025
Actor:	User		
Description:	User is shown metrics for walking to their destination.		
Preconditions:	1. User has searched for a destination and selected the “cycling” mode of transportation.		
Postconditions:	1. The User gets shown the corresponding metrics for cycling to their destination.		
Priority:	High		
Frequency of Use:	Medium		
Flow of Events:	1. User searches for destination 2. User confirms destination 3. User selects “cycling” as mode of transportation 4. Retrieve calories burnt and time taken 5. Display drop-down table that shows CO2 emission (zero), travel time and total cost (zero).		
Alternative Flows:	None		
Exceptions:	None		
Includes:	None		
Special Requirements:	User's phone location services are enabled for TripTally.		
Assumptions:	None		
Notes and Issues:	None		

III. V Bike Rental Information

Use Case ID:	#3-5		
Use Case Name:	BikeRentalInformation		
Created By:	Ethan Jared Chong Rui Zhi	Last Updated By:	Ethan Jared Chong Rui Zhi
Date Created:	1/8/2025	Date Last Updated:	1/8/2025
Actor:	User		
Description:	Shows all available bike-sharing spots near the User.		
Preconditions:	1. The User has selected “cycling” as their chosen mode of transportation. 2. The User has opted to show the available bike-sharing points.		
Postconditions:	1. Successfully display availability of bikes at the various spots. 2. Successfully display the locations of possible return points.		
Priority:	Medium		
Frequency of Use:	High		
Flow of Events:	1. Query bike-sharing API(s) 2. Show nearby bike-sharing points 3. Show the estimated number of available bicycles 4. Retrieve hourly/half-hourly rental cost and calculate cost to get to destination based on time taken to walk to bike-sharing point and then reach the destination. 5. Provide redirect to bike sharing application		

Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	User's phone location services are enabled for TripTally.
Assumptions:	None
Notes and Issues:	None

IV. For Functional Requirement #4

IV.I Fetch Traffic Data

Use Case ID:	#4-1		
Use Case Name:	Fetch Traffic Data		
Created By:	Evelyn Theresia Cuaca	Last Updated By:	Evelyn Theresia Cuaca
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	External System		
Description:	The system is able to receive and process real-time traffic incidents such as accidents, closures, or roadworks from an external feed.		
Preconditions:	<ol style="list-style-type: none"> 1. Connection to traffic alert API is active. 2. The user is viewing a map, searching, or navigating. 		
Postconditions:	<ol style="list-style-type: none"> 1. Incident data ingested and displayed in the app. 2. Routes and ETAs updated accordingly. 3. Logs and metrics recorded for monitoring. 		
Priority:	High		
Frequency of Use:	High		
Flow of Events:	<ol style="list-style-type: none"> 1. The Traffic Alert System (External system) alerts the system about traffic conditions (accident, road closure). 2. The system normalizes and shows the data. 3. The incident is stored in the TripTally database including the timestamp and validity period 4. Routing engine checks whether the incident affects: <ul style="list-style-type: none"> o Active trips in progress. o New trip requests. 5. If an active route is affected, drivers receive a soft reroute suggestion. 6. Map view automatically overlays icons for incidents (warning symbols, red markers, etc.). 		

Alternative Flows:	<p>AF-1: API Failure: If the traffic feed is unavailable, the system falls back to the last known incidents .User is warned that traffic data may be outdated.</p> <p>AF-2: High Latency: If data fetching takes more than 60s, the routing service uses default routes and applies the update once available.</p>
Exceptions:	None
Includes:	None
Special Requirements:	Access to traffic API
Assumptions:	The System has access to traffic alert API
Notes and Issues:	None

IV.II Manage Account User

Use Case ID:	#4-2		
Use Case Name:	Manage Account User		
Created By:	Evelyn Theresia Cuaca	Last Updated By:	Evelyn Theresia Cuaca
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	Admin		
Description:	Admin can view, deactivate, delete account		

Preconditions:	<ol style="list-style-type: none"> 1. Admin must be authenticated with proper role/ privileges 2. The target user account exist
Postconditions:	<ol style="list-style-type: none"> 1. User account data is successfully updated in the database. 2. All changes are recorded with timestamp, admin ID, and reason for change 3. If the account is deactivated/ deleted, the user losses access immediately
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Admin logs into the system 2. Admin navigates to the User Account Management panel 3. Admin searches for the target user account by ID, email, or name 4. System retrieves and displays user profile and account status. 5. Admin selects an action: <ul style="list-style-type: none"> • Edit user details • Deactivate account • Delete account 6. Admin confirms action 7. System validates the request and applies changes
Alternative Flows:	<p>AF-1: Reset Password: Instead of editing details, Admin triggers a password reset email/SMS.</p>
Exceptions:	None
Includes:	Login
Special Requirements:	<ol style="list-style-type: none"> 1. Deactivated users must immediately lose access to the system.
Assumptions:	Only authorized admins can access this feature.
Notes and Issues:	<ol style="list-style-type: none"> 1. Account deletion is irreversible. 2. Sensitive data (like passwords) is never directly editable, only resettable.

IV.III ManageObstructionReport

Use Case ID:	#4-3		
Use Case Name:	ManageObstructionReport		
Created By:	Parvez Kurniawan Wijaya	Last Updated By:	Parvez Kurniawan Wijaya
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	Admin		
Description:	Admin verifies road incidents that are reported by Users.		
Preconditions:	<ul style="list-style-type: none"> 1. Admin received information of road incident reported by users 2. Incident report by the User is stored in the system as Unverified 		
Postconditions:	<ul style="list-style-type: none"> 1. Incident record is updated with a status: Verified, Rejected, or Pending Review 2. Verified incidents are shown on the user's map and factored into routing 3. Rejected incidents are excluded from the routing system 		
Priority:	High		
Frequency of Use:	Medium		
Flow of Events	<ul style="list-style-type: none"> 1. The user submits an incident report through the app . 2. The system stores the incident as Unverified by default. 3. Admin reviews and updates the status: Verified / Rejected / Pending. 4. The system updates the incident record. 5. If Verified, the incident is factored into ETA and rerouting logic. 6. The user is notified of the outcome (e.g., "Your report has been verified"). 		
Alternative Flow	None		
Exceptions	None		
Include	<ul style="list-style-type: none"> 1. Login 2. ViewMap 		
Special Requirement	Only verified incidents affect routing and reroute notifications		
Assumptions	Not all API alerts are accurate, some require Admin judgement		

Notes and Issues	None
------------------	------

IV.IV ManageSystemReport

Use Case ID:	#4-4		
Use Case Name:	ManageSystemReport		
Created By:	Evelyn Theresia Cuaca	Last Updated By:	Evelyn Theresia Cuaca
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	Admin		
Description:	Admin can view, categorize, assign, resolve, or archive reports to ensure that issues are addressed promptly		
Preconditions:	<ol style="list-style-type: none"> Admin must be authenticated with the correct privileges. At least one report must exist in the system. 		
Postconditions:	<ol style="list-style-type: none"> Reports are updated with new statuses All admin actions are recorded with timestamp and admin ID. Resolved reports no longer appear in the report queue. 		
Priority:	Medium		
Frequency of Use:	Medium		
Flow of Events:	<ol style="list-style-type: none"> Admin logs into the system Admin navigates to the Reports Management Module System displays a list of reports Admin selects a report from the list Admin chooses an action: <ul style="list-style-type: none"> Resolve the report Assign report to another admin/ technical team Archive report if it is invalid or duplicate Admin's action is recorded by the system Confirmation message shown to Admin 		
Alternative Flows:	AF-1: Escalation Flow: If the issue is critical, Admin hand it over to higher-level support or incident response team.		
Exceptions:	EX-1: Report Not Found: Report may have been deleted or already resolved by another admin.		

Includes:	None
Special Requirements:	<ol style="list-style-type: none"> Reports must always have one of the following status values: Submitted, Pending, Resolved. Critical reports should trigger notifications to Admin
Assumptions:	<ol style="list-style-type: none"> Reports follow a linear lifecycle (Submitted → Pending → Resolved).
Notes and Issues:	<ol style="list-style-type: none"> “Submitted” is the default state for new reports. Once marked “Resolved,” reports can still be reviewed in the history/archive. Users will receive notifications when their report changes status.

IV.V Manage Suggested Route

Use Case ID:	#4-5		
Use Case Name:	ManageSuggestedRoute		
Created By:	Evelyn Theresia Cuaca	Last Updated By:	Evelyn Theresia Cuaca
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	Admin		
Description:	Admin can review and validate user-suggested routes against traffic data and safety constraints, and either approves, modifies, or rejects them		
Preconditions:	<ol style="list-style-type: none"> The user has submitted a suggested route to the forum. Admin is authenticated and authorized. Suggested route data (start, end, waypoints) is stored in the system. 		
Postconditions:	<ol style="list-style-type: none"> Approved routes are integrated into the routing database or flagged for testing. Verification of suggested route in the forum 		
Priority:	Low		
Frequency of Use:	Medium		

Flow of Events:	<ol style="list-style-type: none"> 1. Admin logs into the system 2. Navigates to the Suggested Routes Forum 3. The system displays a list of user-submitted routes with detailed information(origin–destination, date submitted, status). 4. Admin selects a suggested route for review 5. The system shows details 6. Admin evaluates the suggestion against: <ul style="list-style-type: none"> • Safety (road closures, traffic rules). • Efficiency (time, cost). • Relevance (popular demand) 7. Admin takes one of the following actions <ul style="list-style-type: none"> • Approve : verify the suggested route • Reject : mark invalid with reason. 8. The Status of suggested route is updated (Verified, Rejected)
Alternative Flows:	AF-1: Auto-Approval: Low-risk routes may be auto-verified.
Exceptions:	EX-1: Invalid Data - Suggested route is missing waypoints or corrupted EX-2: Duplicate Suggestion- Route already exists in Forum
Includes:	None
Special Requirements:	<ol style="list-style-type: none"> 1. Suggested routes must have a status (Submitted, Pending Review, Verified, Rejected). 2. Status of verification must be shown in the Forum. 3. Users must download and load alternate routes from the Forum before use. 4. If no alternate route is loaded, the app defaults to the normal system-generated route. 5. Approved routes must not violate traffic/safety rules.
Assumptions:	None
Notes and Issues:	Forum acts as a shared library for alternate user-suggested routes

V. For Functional Requirement #5

V.I External System Integration

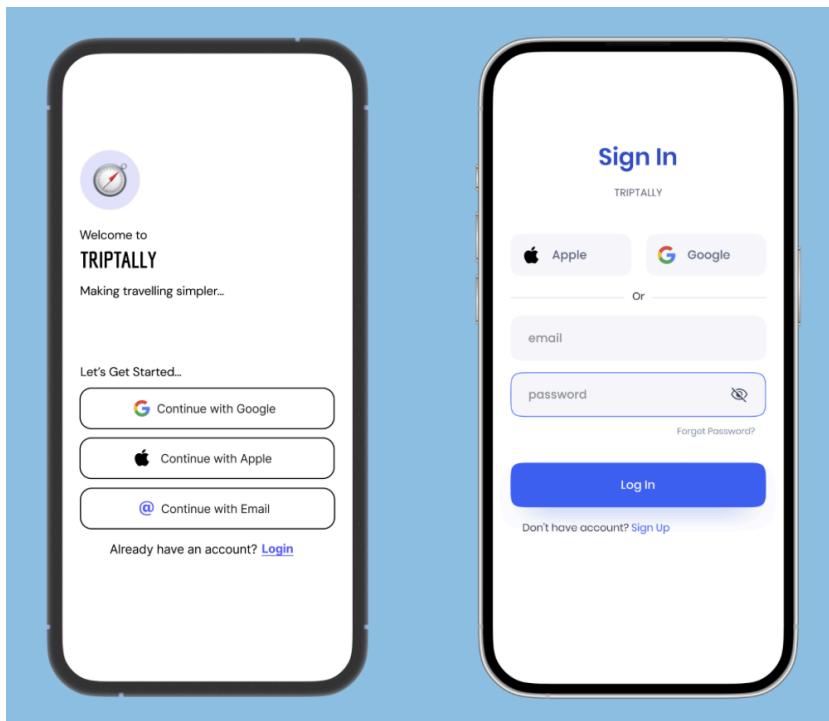
Use Case ID:	#5-1		
Use Case Name:	ExternalSystemIntegration		
Created By:	Evelyn Theresia Cuaca	Last Updated By:	Evelyn Theresia Cuaca
Date Created:	01/09/2025	Date Last Updated:	01/09/2025
Actor:	External System / Data Provider		
Description:	Integrate with external systems (data provider) to get real-time data like routes, traffic incidents, ERP rates, carpark availability, and fuel prices,etc.		
Preconditions:	<ol style="list-style-type: none"> 1. Valid API keys/credentials for each external provider 2. Network access to provider endpoints. 3. Data structure and mappings are defined 		
Postconditions:	<ol style="list-style-type: none"> 1. Data retrieved from external systems is made available to TripTally service 		
Priority:	High		
Frequency of Use:	High		
Flow of Events:	<ol style="list-style-type: none"> 1. Mapping <ul style="list-style-type: none"> • System calls Google Map API to identify user position • The API fetch available driving, walking, cycling, and public transportation routes • Map APIs will render preview of the routes 2. Traffic and Incidents <ul style="list-style-type: none"> • TripTally get real-time traffic conditions and road closures from external traffic APIs • Incident data is applied to update travelling time and rerouting logic. 3. ERP integration <ul style="list-style-type: none"> • The system retrieves ERP gantry locations and rate tables by time/day from LTA ERP providers. • ERP costs calculated per driving route. 		

	<p>4. Parking Integration</p> <ul style="list-style-type: none"> • Carpark APIs provide real-time availability and rates • The system will update parking charges and lot availability. <p>5. Fuel Prices</p> <ul style="list-style-type: none"> • TripTally retrieves current fuel price data from providers. • Driving cost calculations is updated based on user's vehicle profile <p>6. Carbon Footprint</p> <ul style="list-style-type: none"> • The average emission factors per transport mode is retrieved from carbon services. • The system calculates estimated CO₂ footprint per trip. <p>7. Public Transport Data</p> <ul style="list-style-type: none"> • Bus and train schedules retrieved from LTA DataMall. • Fare tables pulled from fare providers, travelling times and costs are computed. <p>8. Calorie API</p> <ul style="list-style-type: none"> • Retrieve calorie expenditure estimates for active travel modes (walking, cycling). • Use external health/fitness APIs to compute calories burnt. <p>9. Ride-Hailing Links</p> <ul style="list-style-type: none"> • Taxi/ride-hailing APIs provide service availability . • The user is presented with options (Grab, CDG), tapping will redirect to app or app store.
Alternative Flows:	AF-1: Data not available
Exceptions:	EX-1: Timeout - Return fallback values after threshold exceeded.
Includes:	<ol style="list-style-type: none"> 1. ViewMap 2. ViewMetrics 3. CompareTransportation
Special Requirements:	<ul style="list-style-type: none"> • Critical feeds must be updated within ≤ 60s.

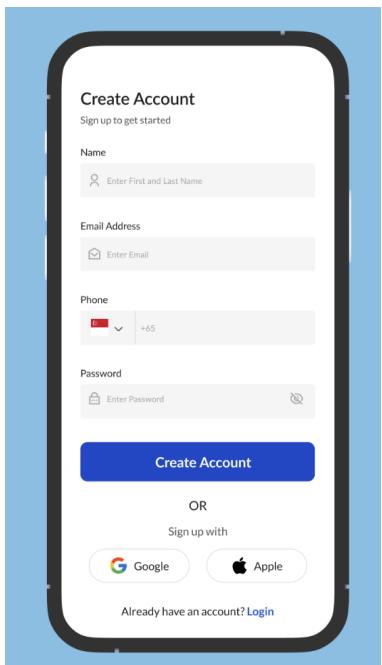
	<ul style="list-style-type: none"> • API responses must integrate into the app within ≤ 2.5s
Assumptions:	<ol style="list-style-type: none"> 1. External providers maintain reliable APIs and SLAs. 2. TripTally has access to provider data. 3. Users may need to install third-party ride-hailing apps for deep linking.
Notes and Issues:	<p>System must support multi-provider fallback Some private carparks may lack reliable data</p>

4. UI Mockups

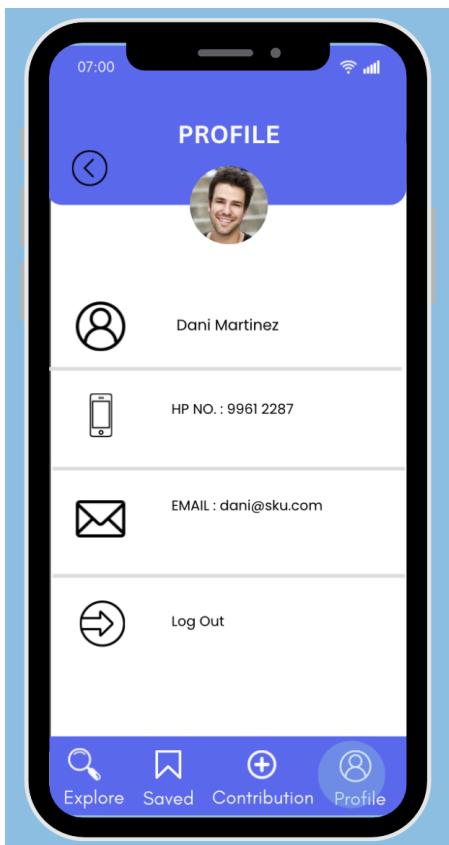
Login



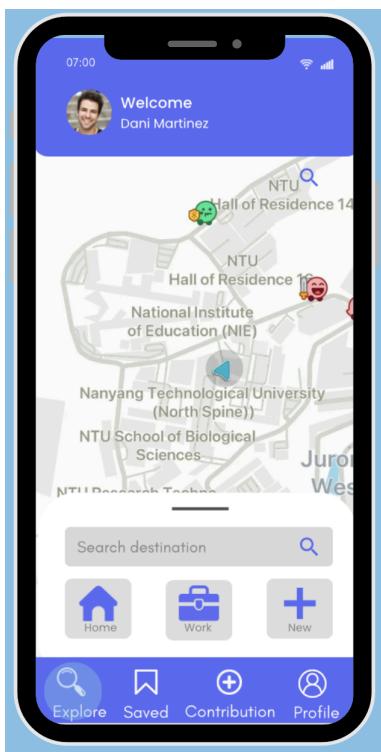
Signup



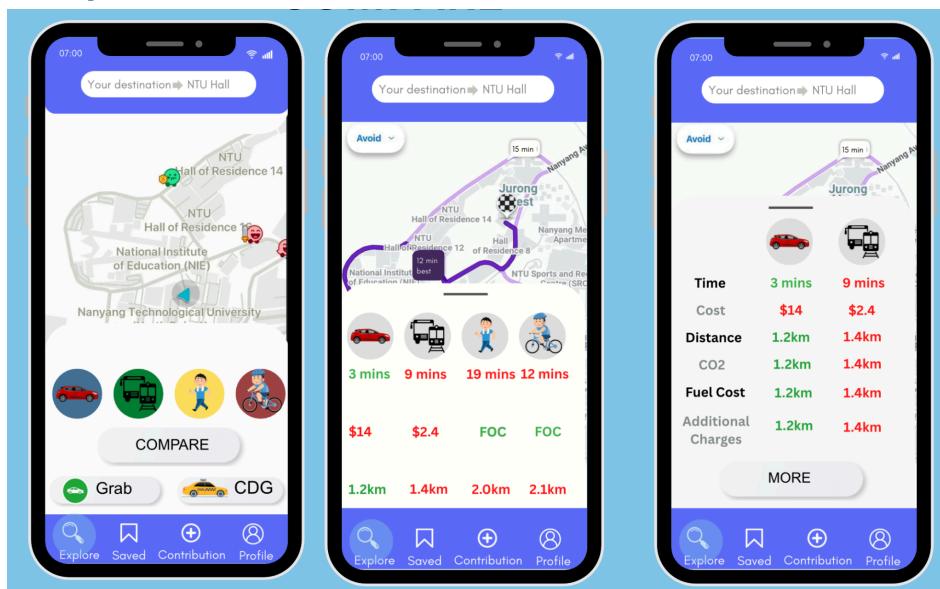
User Profile



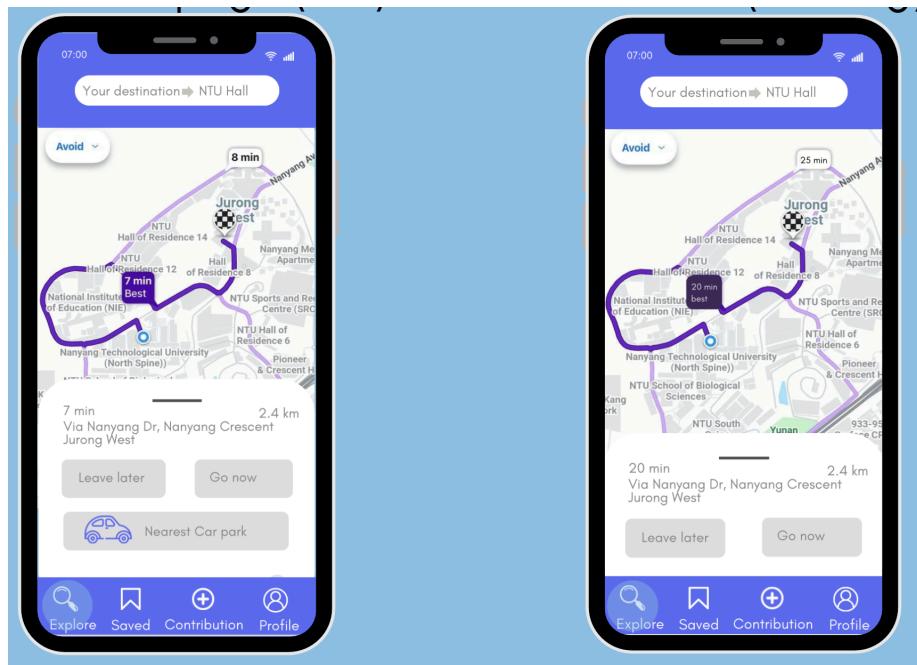
Home Screen



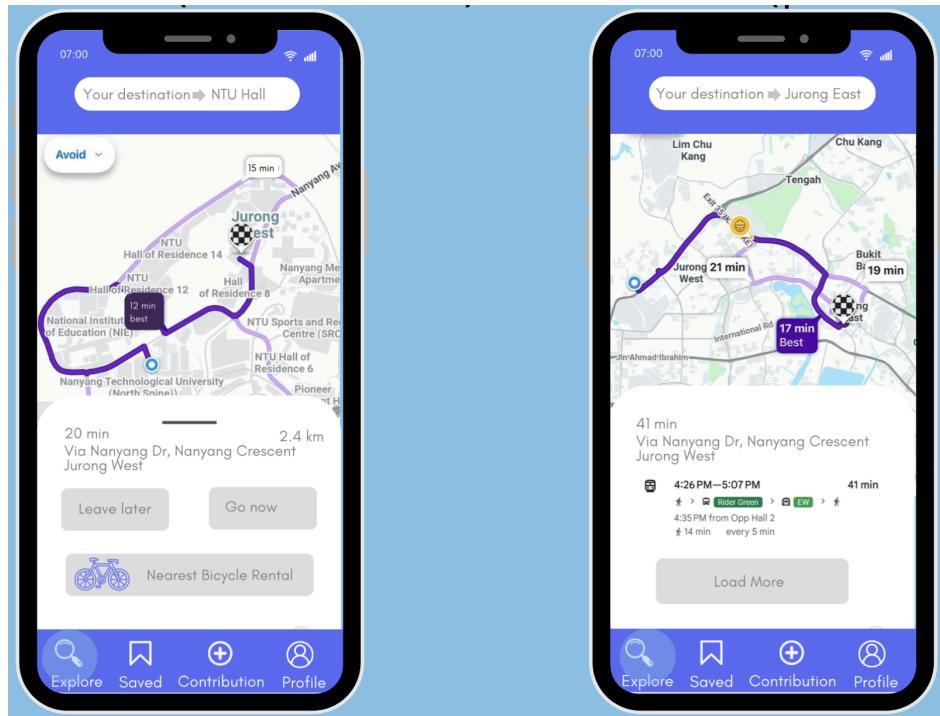
Compare modes of travel



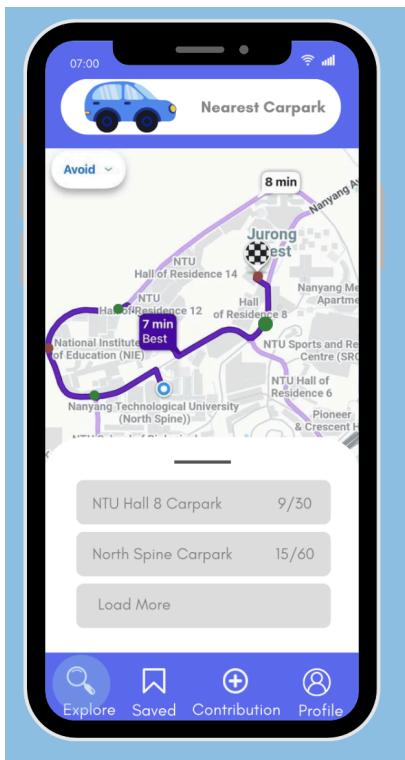
Destination page (Driving) and Destination page (Walking)



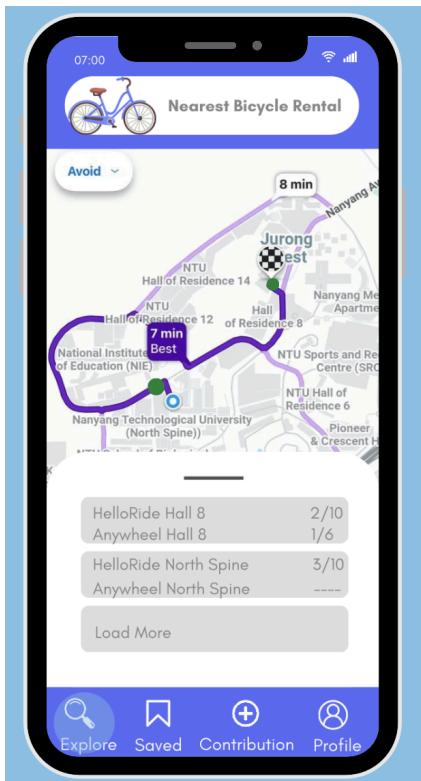
Destination page (Cycling) and Destination page (Public Transport)



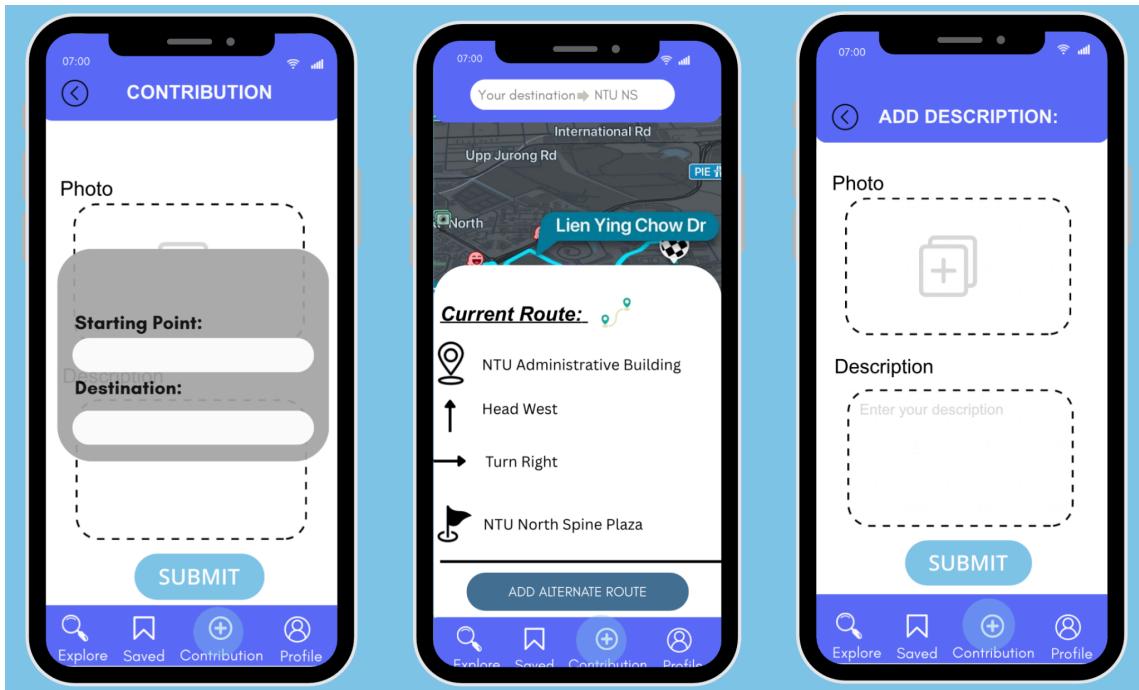
Nearest Carpark with Availability



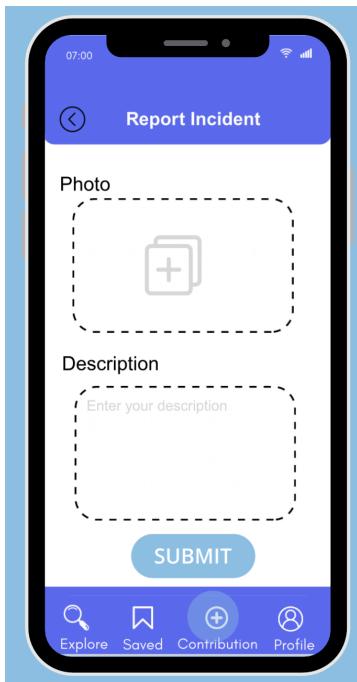
Nearest Bicycle Rental



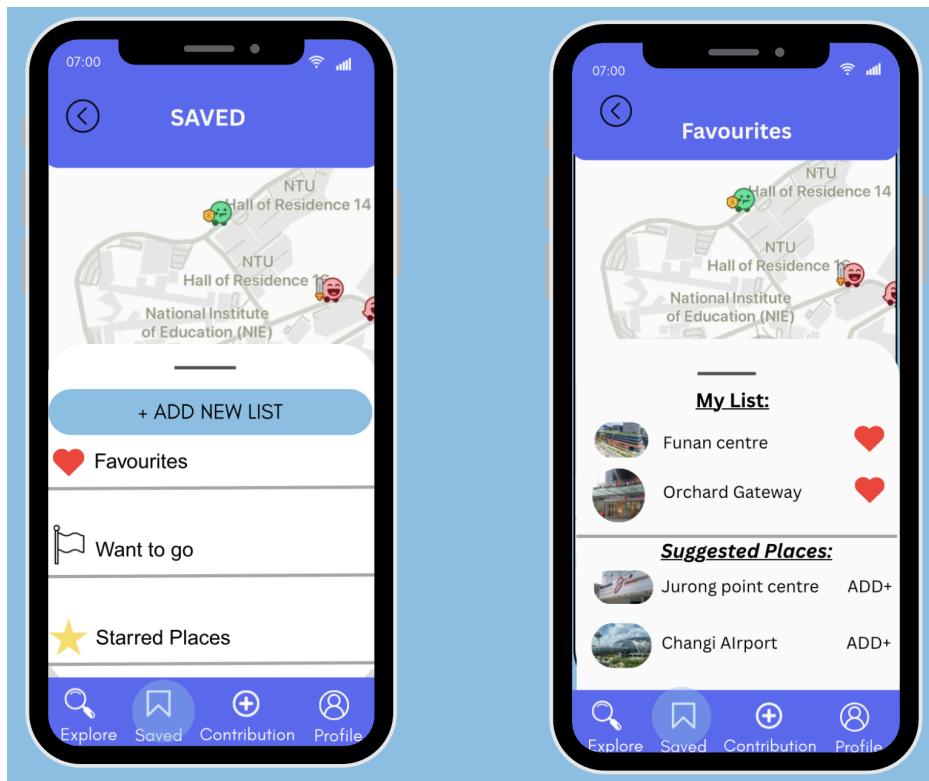
User suggestions/contributions



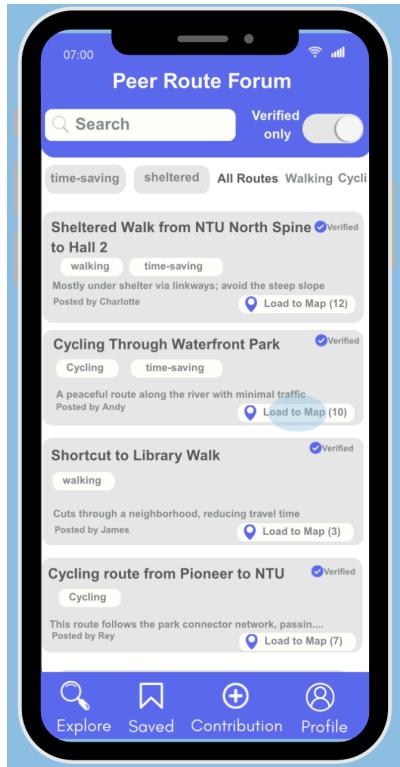
User Incident Report



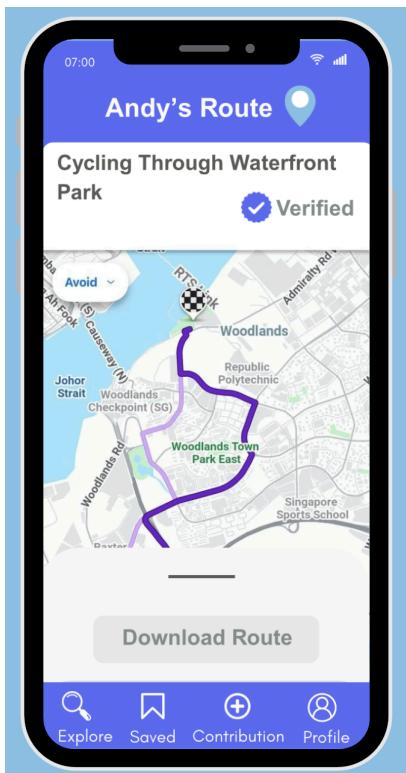
User Saved/ Favourites



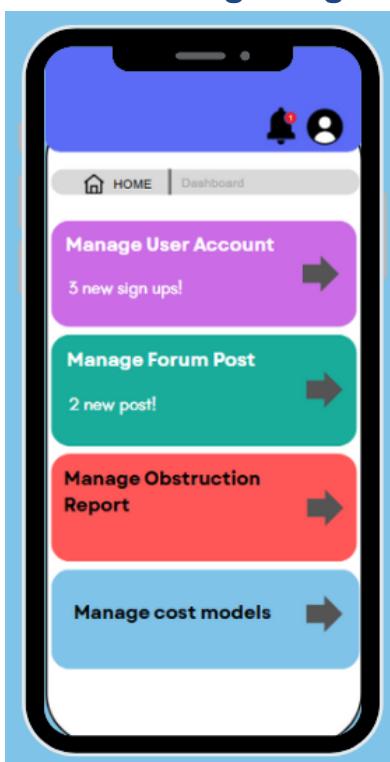
Suggested Route by Users Forum



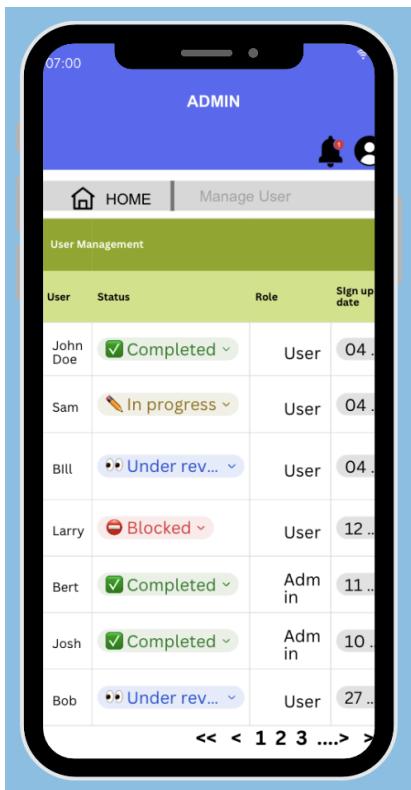
Approved Suggested Routes



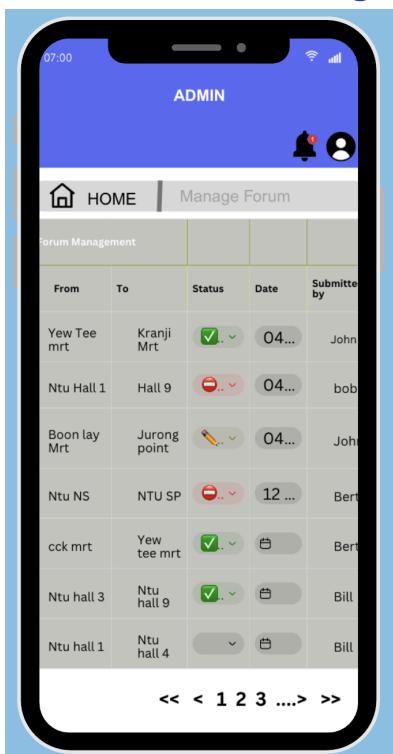
Admin Manage Page



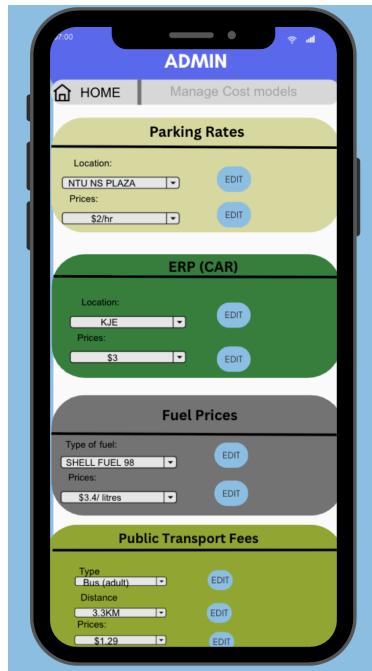
Admin User Management Page



Admin Forum Management Page



Admin Manage Cost Models Page



Admin Manage System Reports Page

The screenshot shows a mobile application interface titled 'ADMIN' at the top. Below it is a navigation bar with 'HOME' and 'Manage System Reports'. There are icons for a bell and a user profile. The main content area is a table showing system reports. The table has columns for 'Report ID', 'Issues', 'Description', 'Status', 'Report Time', and 'Action'. Each row contains a report ID, the issue description, the status (Pending or Submitted), the report date, and a 'Review' button.

Report ID	Issues	Description	Status	Report Time	Action
U1023	Fare discrepancy	"I was charged \$3.20 but the"	Pending	Jun 25, 2025	Review
U1456	Wrong route suggestion	The app told me to go through.....	Submitted	Jun 24, 2025	Review
U1789	App lagging	"The app keeps freezing when....."	Pending	Jun 24, 2025	Review
U1122	Login failure	"I reset my password but I	Pending	Jun 25, 2025	Review
U1345	Payment not processed	"The money got deducted from	Submitted	Jun 25, 2025	Review
U1678	Map not loading	"The map doesn't load when	Submitted	Jun 23, 2025	Review
U1900	Incorrect ETA	The app said the bus will come in..	Submitted	Jun 24, 2025	Review

<< < 1 2 3> >>

Admin Manage Obstruction Reports Page

The screenshot shows a tablet displaying the 'Manage Obstruction Reports' section of an admin dashboard. The interface has a blue header bar with the word 'ADMIN' in white. Below the header is a navigation bar with icons for 'HOME', a bell with a red notification dot, and a user profile icon. The main content area is a table titled 'Manage Obstruction Reports' with columns: Report ID, Issues, Description, Status, Report Time, and Action. There are seven rows of data, each containing a report ID, a brief issue description, the full issue text in a tooltip, the status (Pending or Submitted), the report date, and a 'Review' button. At the bottom of the table is a pagination control with arrows and page numbers (1, 2, 3, ...).

Report	Issues	Description	Status	Report Time	Action
U1023	Fare discrepancy	"I was charged \$3.20 but the app says it's \$3.00."	Pending	Jun 25, 2025	Review
U1456	Wrong route suggestion	The app told me to go through a crowded area.	Submitted	Jun 24, 2025	Review
U1789	App lagging	"The app keeps freezing whenever I scroll."	Pending	Jun 24, 2025	Review
U1122	Login failure	"I reset my password but I still can't log in."	Pending	Jun 25, 2025	Review
U1345	Payment not processed	"The money got deducted from my account but nothing happened."	Submitted	Jun 25, 2025	Review
U1678	Map not loading	"The map doesn't load when I try to view it."	Submitted	Jun 23, 2025	Review
U1900	Incorrect ETA	The app said the bus will come in 10 minutes but it's late.	Submitted	Jun 24, 2025	Review

Admin System Health Dashboard

The screenshot shows a tablet displaying the 'System Health Dashboard' section of an admin dashboard. The interface has a blue header bar with the word 'ADMIN' in white. Below the header is a navigation bar with icons for 'HOME', a bell with a red notification dot, and a user profile icon. The main content area displays several system status cards. From top-left to bottom-right: 1) 'System Status' card with a green checkmark icon and the text 'All good !!'. 2) 'Server Status' card showing 'Normal'. 3) 'API Connections' card showing 'All API Connected'. 4) 'Database' card showing 'Running smoothly'. 5) 'User Reports' card showing '3 new reports today'. 6) A button labeled 'Enable Maintenance Mode' with a gear icon. 7) A warning card with a yellow exclamation mark icon and the text 'Issue Warning to User'.

