Use Cases

for

TransitEase

Version 1.0 approved

Prepared by Ashwin, Dave, Jun Heng, Jonathan

Nanyang Technological University

03/09/24

Revision History

Name	Date	Reason For Changes	Version
Ashwin	3 Sept		1.0
Dave	3 Sept		1.0
Jonathan	3 Sept		1.0
Goh Jun Heng	3 Sept		1.0

Table Of Contents

1. Users

1.1. Table of Users

User Types	Description
Daily Commuters	Individuals who drive to work or school and need to find parking regularly.
	People attending events, concerts, or sports games who need nearby parking.
	University or college students who drive to campus and need to find cheaper parking.
	People with disabilities who require accessible parking spaces/disabled parking spaces

2. Requirements

2.1. Functional Requirements

2.1.1. The Application must be able to create an account for a new User using the Firebase API.

2.1.1.1. The User must enter a UserID.

2.1.1.1.1. The UserID must be a string of at least 1 character and at most 512 characters.

2.1.1.1.2. The UserID must not be shared with any existing User.

2.1.1.2. When the Application is creating an account, the User must enter a UserPassword.

2.1.1.2.1. The UserPassword must be a string of at least 8 characters and at most 512 characters.

2.1.1.2.2. The password must contain at least one uppercase letter, at least one lowercase letter, at least one number and at least one special character.

2.1.1.3. After the User enters a valid UserPassword, the Application must add the UserID to the user database.

2.1.1.4. After the User enters a valid UserID and a valid UserPassword, the Application must add the UserPassword to the user database.

2.1.1.5.

2.1.2. The Application must be able to login an existing User to the application.

2.1.2.1. User must provide the following details

2.1.2.1.1. Email

2.1.2.1.2. Password

2.1.3. The Application must be able to logout a user from the application.

2.1.4. The Application must detect the user's current location using GPS or other location services.

2.1.5. The Application must integrate with a map service to visually display the location of car parks.

2.1.6. The Application must be able to retrieve information on nearby car parks based on the user's location.

2.1.6.1. Information Include:

2.1.6.1.1. Distance to Carpark

2.1.6.1.2. Available lots

2.1.6.1.3. Cost of parking

2.1.6.1.4. Electric Vehicle (EV) Charging

2.1.7. The Application must show real-time information about car park availability and pricing.

2.1.8. The application must allow users to filter car parks

2.1.8.1. FIlters include:

2.1.8.1.1. Distance to current user location

2.1.8.1.2. Pricing

2.1.8.1.3. Availability of lots

2.1.8.1.4. EV Charging

2.1.9. The Application must must be able to store user preferences

2.1.9.1. Preferences include:

2.1.9.1.1. Default distance for search

2.1.9.1.2. Favorite car parks

2.1.9.1.3. EV Charging

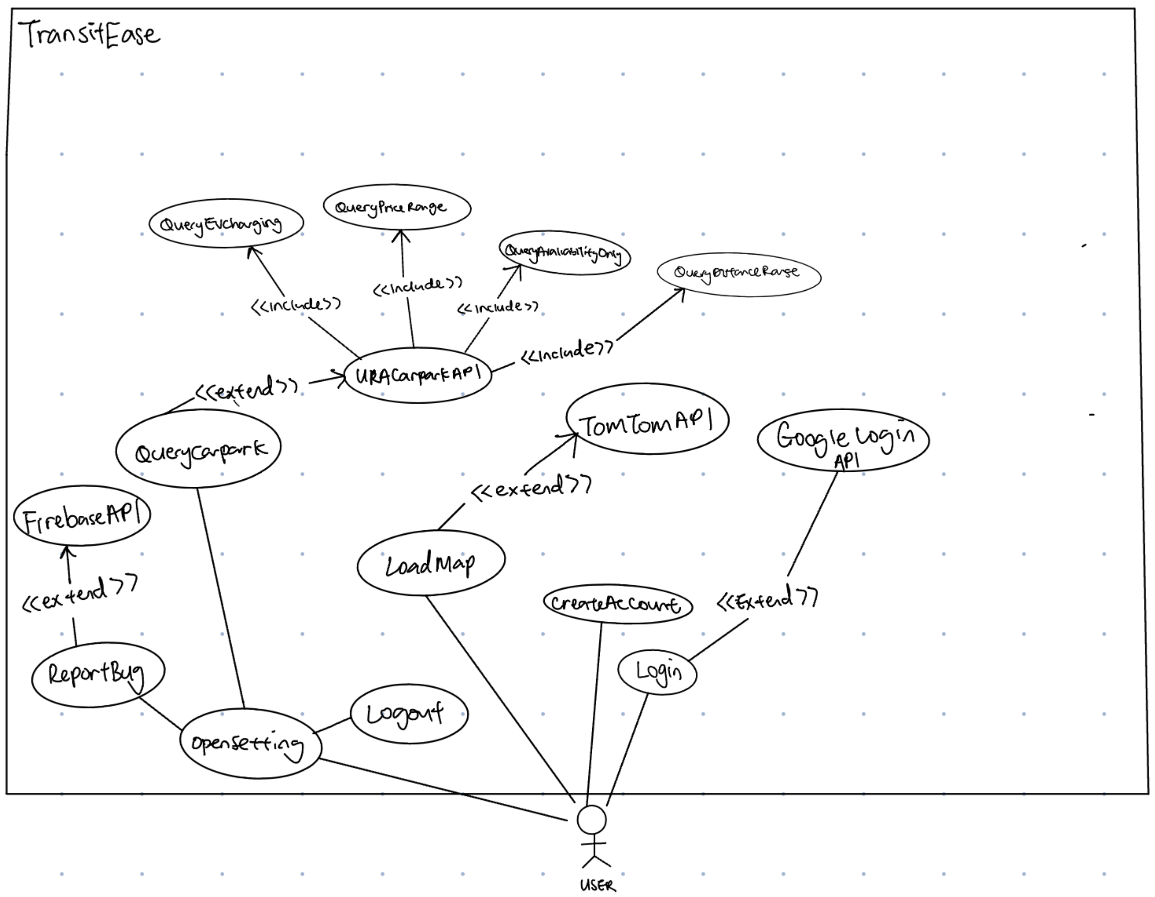
2.1.9.1.4. AvailabilityOnly

2.2. Non-Functional Requirements

ID	Requirement	Description
1	Performance	Ensure the app loads carpark data and responds to user actions within a few seconds.
2	Scalability	Handle an increasing number of users and data without a decline in performance.
3	Security	Securely handle and store user data, including location and payment information.
4	Reliability	Ensure consistent functionality without crashes, even under different network conditions.
5	Usability	Provide an intuitive user interface that is easy to navigate for users of all technical abilities.
6	Compatibility	Be compatible with both iOS and Android devices.
7	Data Privacy	Implement data privacy practices to comply with relevant laws and regulations.
8	Battery Efficiency	Optimize the app to minimize battery usage, especially when using location services.
9	Maintainability	Organize and document the codebase to allow for easy updates and maintenance.
10	Accessibility	Ensure the app is accessible to users with disabilities, following best practices for mobile app accessibility.
11	Language	Content must have support for different languages other than English

3. Use Case

A. Use Case Diagram


TransitEase
QueryEvcharging
QueryPriceRange
QueryAvailabilityOnly
QueryDistanceRange
<<include>>
<<include>>
<<include>>
<<include>>
URACarparkAPI
<<extend>>
QueryCarpark
TomTomAPI
Google Login API
FirebaseAPI
<<extend>>
LoadMap
<<extend>>
CreateAccount
<<Extend>>
ReportBug
Login
Logout
OpenSetting
USER

1. Login with Email

Use Case ID:	1		
Use Case Name:	LogInEmail		
Created By:	Ashwin Suresh	Last Updated By:	Ashwin Suresh
Date Created:	03/09/24	Date Last Updated:	03/09/24

Actor:	User
Description:	Allow user to create a new account
Preconditions:	User must not exist
Postconditions:	None
Priority:	High
Frequency of Use:	1
Flow of Events:	User selects “Login with Email” in the user interface User enters username and password User is granted access
Alternative Flows:	NIL
Exceptions:	1. Incorrect password 2. Username does not exist
Includes:	
Special Requirements:	1. Firebase authentication
Assumptions:	1. User has created an account before
Notes and Issues:	

2. Login with Google

Use Case ID:	2		
Use Case Name:	LoginWGoogle		
Created By:	Ashwin Suresh	Last Updated By:	Ashwin Suresh
Date Created:	03/09/24	Date Last Updated:	03/09/2024

Actor:	User
Description:	Allow user to log in with google account
Preconditions:	User must have a google account
Postconditions:	None
Priority:	High
Frequency of Use:	1
Flow of Events:	1. User selects “Login with Google” in the user interface 2. User allows access to app in google account permissions 3. User logs in
Alternative Flows:	NIL
Exceptions:	1. User denies access of app to google account
Includes:	
Special Requirements:	1. Google authentication API
Assumptions:	1. User has valid google account
Notes and Issues:	

3. Create Account with Email

Use Case ID:	3		
Use Case Name:	CreateAccount		
Created By:	Ashwin Suresh	Last Updated By:	Ashwin Suresh
Date Created:	03/09/24	Date Last Updated:	03/09/2024

Actor:	User
Description:	Allow user to create a new account with Email
Preconditions:	User must not exist
Postconditions:	None
Priority:	High
Frequency of Use:	1
Flow of Events:	1. User selects “Create Account” in the user interface 2. User inputs UserID, UserPassword, ReenterUserPassword into System 3. User successfully creates an account
Alternative Flows:	NIL
Exceptions:	1. User email already exists 2. UserPassword and ReenterUserPassword does not match
Includes:	
Special Requirements:	1. Firebase API
Assumptions:	1. User is not a prior user 2. User has a valid email address 3. User types in both passwords correctly
Notes and Issues:	

4. LoadMap

Use Case ID:	4		
Use Case Name:	LoadMap		
Created By:	Ashwin Suresh	Last Updated By:	Ashwin Suresh
Date Created:	03/09/24	Date Last Updated:	03/09/2024

Actor:	User
Description:	Loads Map API onto screen
Preconditions:	1. User has allowed location permissions when requested
Postconditions:	None
Priority:	High
Frequency of Use:	1
Flow of Events:	1. User is prompted to give app permission for location 2. User grants location 3. Map localizes map view to user location 4. Icon representing current user location is shown
Alternative Flows:	
Exceptions:	1. User does not give access to location permissions 2. User does not have network connectivity

Includes:	1. TomTom API
Special Requirements:	1. Permission Locations
Assumptions:	
Notes and Issues:	

5. QueryCarpark

Use Case ID:	5		
Use Case Name:	QueryCarpark		
Created By:	Ashwin Suresh	Last Updated By:	Ashwin Suresh
Date Created:	03/09/24	Date Last Updated:	03/09/2024

Actor:	System
Description:	Loads nearby carparks onto map API
Preconditions:	1. Network connectivity
Postconditions:	None
Priority:	High
Frequency of Use:	High
Flow of Events:	1. System queries URA Carpark API for carpark information 2. System displays carpark information from nearest to furthest from user 3. System displays information about carpark a. Information includes: i. Distance ii. Rate iii. EV Charging capability iv. Capacity
Alternative Flows:	
Exceptions:	1. No Network connectivity 2. No location permissions granted
Includes:	1. URA Carpark API 2. TomTom API
Special Requirements:	
Assumptions:	
Notes and Issues:	

6. reportBug

Use Case ID:	6		
Use Case Name:	reportBug		
Created By:	Ashwin Suresh	Last Updated By:	Ashwin Suresh
Date Created:	03/09/24	Date Last Updated:	03/09/2024

Actor:	User
Description:	Allow user to create a report about bug
Preconditions:	User must be a valid account
Postconditions:	None

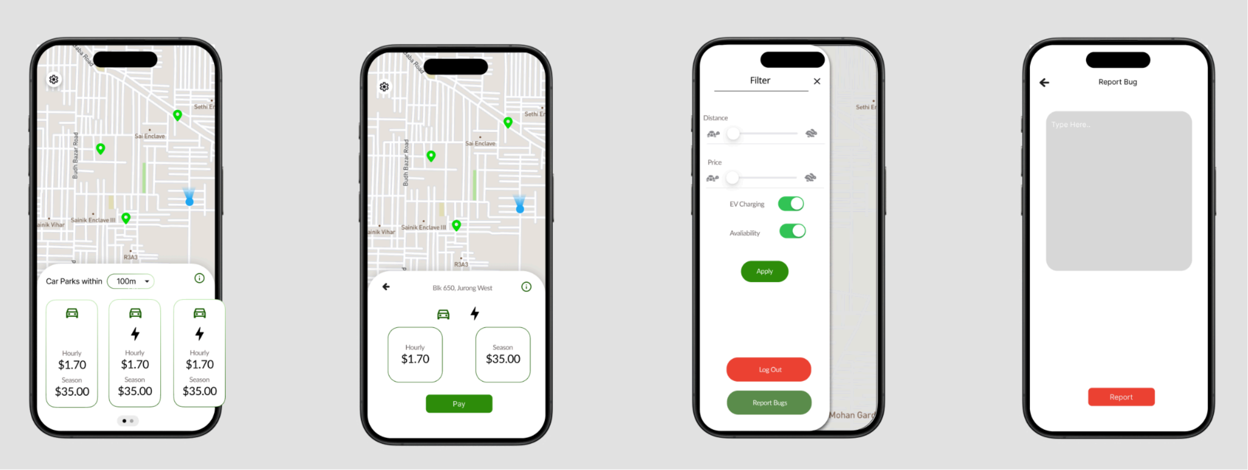
Priority:	High
Frequency of Use:	1
Flow of Events:	1. User clicks setting icon 2. User clicks report button 3. User types report or bug in text field 4. User clicks submit 5. Report is sent to database
Alternative Flows:	
Exceptions:	1. Data in textfield has unexpected characters 2. Data in textfield is longer than 1000 characters 3. User does not have network connectivity
Includes:	1. Firebase API
Special Requirements:	
Assumptions:	
Notes and Issues:	

4. Data Dictionary

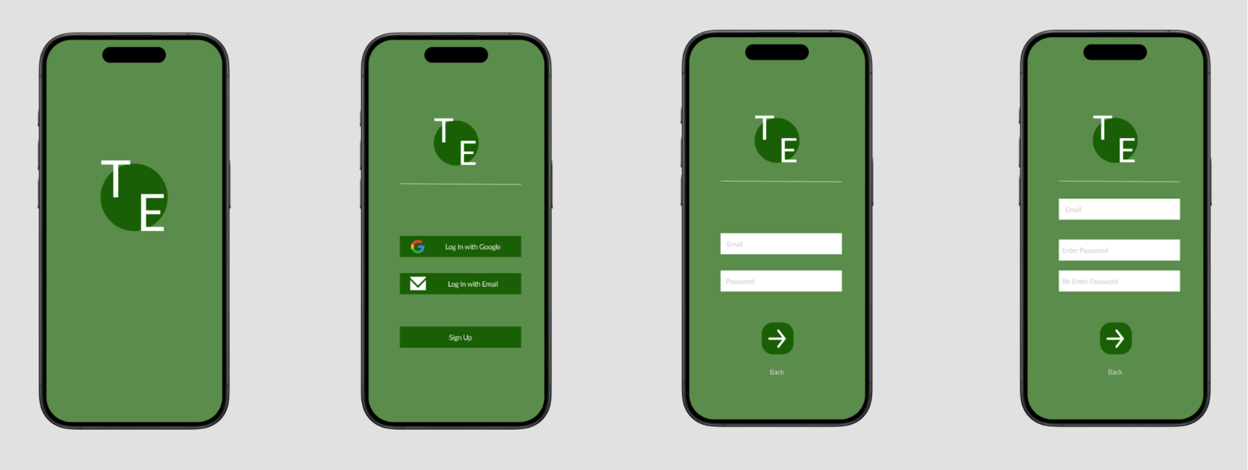
1.1. Current Data Dictionary

Entity	Column Name	Data Type	Description
User	UserID	String	Unique identifier for each user
	UserPassword	String	Password associated with user
	LocationCoordinates	JSON	Current GPS coordinates (latitude and longitude) of the user
	Preferences	JSON	User's preferences for carpark selection (e.g., price range, proximity)
	NotificationsEnabled	Boolean	Indicates whether the user has enabled notifications for car park alerts
Carpark	CarparkID	String	Unique identifier for each car park
	Name	String	Name of the car park
	LocationCoordinates	JSON	GPS coordinates of the car park (latitude and longitude)
	Capacity	Integer	Total number of parking spots available in the carpark
	CurrentOccupancy	Integer	The current number of occupied parking spots
	PricePerHour	Decimal	Cost of parking per hour at the carpark

	RealTimeAvailability	Boolean	Indicates if the carpark supports real-time availability updates
SearchFilters	DistanceRange	Integer	Maximum distance (in meters or kilometers) from the user's current location to search for car parks
	PriceRange	Decimal	Range of parking costs that the user is willing to consider
	AvailabilityOnly	Boolean	Indicate to show only car parks with available spots
	EVCharging	Boolean	Indicates whether to show only car parks with EV Charging
MapSettings	MapService	String	The map service used for displaying car park locations (e.g., Google Maps, Apple Maps)
	Directions	JSON	Directions from the user's current location to a selected carpark
Notifications	NotificationID	String	Unique identifier for each notification sent to the user
	UserIDKey	String	Foreign key referencing the UserID in the Users table
	NotificationType	String	Type of notification (e.g., Car Park Available, Session Expiring)
	Message	String	The content of the notification message
Backend	FireBase		



(Left to Right) Figure 1. User Interface Figure 2. Carpack Selected Figure 3. Filter Selected Figure 4. Report Bug Selected



(Left to Right) Figure 5. Splashscreen Figure 6. Login Options Screen Figure 7. Login Screen Figure 8. Create Account Screen