

Non-functional requirements

1. Usability Requirements

- 1.1. The system must reduce short-term memory load.
 - 1.1.1. To keep the display simple.
- 1.2. The system must display all contents in English.
- 1.3. The system must offer informative feedback.
 - 1.3.1. When the system detects invalid inputs, the system must display an error message.
 - 1.3.2. When certain processes fail, the system must display an error message according to each error.
 - 1.3.2.1. After showing the error message, the system returns to the initial display for each functionality.
- 1.4. The system must strive for consistency.
 - 1.4.1. The system must retain a consistent visual layout.
(e.g.
 - fonts
 - colours
 - placements of buttons)
- 1.5. The system must allow users to sign-up in less than 10 minutes.

2. Reliability Requirements

- 2.1. The fetched website must be secured.
 - 2.1.1. The system must fetch websites protected by utilising SSL.
- 2.2. The system must show users up-to-date information fetched from APIs.
 - 2.2.1. The system allows up to 15 seconds of delay.
- 2.3. The system must be maintained with little or no downtime occurring.
- 2.4. The system must support the internal locus of control.
 - 2.4.1. The system must respond without any lag or latency.

3. Performance Requirements

- 3.1. The system must load data within 5 seconds upon launching.
- 3.2. The system must not crash when the user opens the application.

- 3.3. When a user launches the route search functionality, The system must display the result within 30 seconds.
- 3.4. The system must detect the difference in the number of car parks every 5 minutes.
- 3.5. When the system displays a notification, the system must show a pop-up window in less than 3 seconds.
- 3.5.1. The pop-up must be with a button for closing the pop-up window.
- 3.6. When the system migrates to another display without access to a database, the system must display the consequent display in up to 8 seconds.

4. Supportability Requirements

- 4.1. When the network connection is not available, the system must display an informative pop-up box.
- 4.1.1. The pop-up box must inform the user to retry.
- 4.2. The system application must be developed via Java and JavaScript.
- 4.3. The system must be developed in React Native.
- 4.4. The system must store data collected or input from users on Firebase Realtime Database.
- 4.4.1. The system must store the user's unconfirmed information in a database named 'temporal table'.
- (e.g.
- a raw input of userID/password
 - a raw input to a departure/destination blank
 - a raw input to feedback/comment
-)
- 4.4.2. The system must store the user's confirmed information in a database named 'user information table'.
- (e.g.
- a confirmed user account information
 - a confirmed search input
 - a confirmed feedback/comment
-)