

---

# Software Requirements Specification

for

## OTW Lah

**Version 1.0**

**Prepared by**

Fan Tianyu,  
Teo Hong GuanBrian,  
Lee Wern Jie Eyan,  
Guo Zhiqi,  
Gu Shucheng

**Nanyang Technological University, Team Bla Bla Bus**

**5/9/2023**

# Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose of the Project .....	1
1.2 Document Conventions .....	1
1.3 Intended Audience and Reading Suggestions.....	2
1.4 Product Scope .....	2
1.5 References.....	2
<b>2. Requirements.....</b>	<b>3</b>
2.1 Functional Requirements .....	3
2.2 Non-Functional Requirements .....	7
<b>3. External Interface Requirements .....</b>	<b>9</b>
3.1 User Interfaces .....	9
<b>4. System Use Cases .....</b>	<b>19</b>
4.1 Use Case Diagram .....	19
4.2 Use Case Description.....	20
<b>Appendix A: Data Dictionary.....</b>	<b>41</b>

# 1. Introduction

## 1.1 Purpose of the Project

In Singapore, where the weather is both hot and unpredictable, traveling by bus can be a challenge. Lengthy waits under the sun, combined with sudden rain showers upon reaching the destination, make cars a preferred choice for many, despite the environmental implications.

Our Real-time Transportation App OTA Lah is set to transform this dynamic. By tapping into the Singapore LTA API and Google Maps API, we minimize outdoor waiting times, allowing commuters to stay sheltered until their bus's imminent arrival. More crucially, the app predicts the weather at the destination upon the estimated arrival time, letting users brace for sudden showers or intense heat.

Additionally, our 'Party Mode' streamlines group travel. Users input a shared destination, and the app provides individualized departure times. This ensures synchronized arrivals, and with real-time location and weather tracking, everyone stays informed and prepared.

In essence, our app aims to make bus commutes in Singapore efficient, weather-aware, and eco-friendly, encouraging a shift from cars to public transport while catering to the nation's unique climate challenges.

## 1.2 Document Conventions

**Software Requirement Specification Format:** This document follows IEEE standard. Priorities of higher level requirements are inherited by detailed level requirements.

**Font:** Roboto

**Main Header:** Size 18, Bold

**Subsection Header:** Size 14, Bold

**Content:** Size 12

Further conventions on the terms used could be found at **Appendix A – Data Dictionary Section**

### **1.3 Intended Audience and Reading Suggestions**

This document is designed for all stakeholders associated with the Real-time Transportation App. This includes:

1. Users of the app, encompassing daily commuters and group travellers.
2. Software developers tasked with building the app.
3. Project managers overseeing its development.
4. Marketing teams responsible for its promotion.
5. Testers ensuring the app's robustness.
6. Business partners, notably those from Singapore LTA and Google Maps API.

The document elucidates the app's purpose, its myriad features, and the rationale behind each inclusion.

For Users, Testers, Marketing Teams, and Business Partners: It's recommended to read the document sequentially.

For Software Developers and Project Managers: Begin with the general overview for context. Then, dive into individual features for technical insights, rounding off with specific requirements, both functional and non-functional, for a holistic view.

### **1.4 Product Scope**

Our product will be released as an Android mobile application. Boasting an intuitive user interface, the app integrates real-time data from Singapore LTA and Google Maps API, enhancing the commuting experience in Singapore. Users receive personalized departure suggestions and journey estimates, alongside weather updates for their destination. The 'Party Mode' streamlines group travels, ensuring synchronized arrivals. Aimed to counter Singapore's unpredictable weather, the app seeks to make bus commutes more efficient and predictable.

### **1.5 References**

NA

## 2. Requirements

### 2.1 Functional Requirements

#### Account Creation and Login:

1. Users shall be able to create an account upon entering the application.
  - 1.1. Users must enter necessary data for account creation.
    - 1.1.1. The data for account creation must consist of the user's username.
    - 1.1.2. The data for account creation must consist of the user's email address or contact number.
    - 1.1.3. The data for account creation must consist of the user's chosen password.
2. If the user has previously created an account, the user shall login using their account credentials or using One-Time-Password (OTP).
  - 2.1. The user must enter their username previously used during account creation.
  - 2.2. If the user wants to login via password, then the user must enter the corresponding password set for their account.
    - 2.2.1. If the password entered is valid, the app will log the user into their account.
  - 2.3. If the user wants to login via an OTP, upon selecting this option, the app must send the OTP to the user's registered email or contact number.
    - 2.3.1. The user must enter the OTP into the system.
      - 2.3.1.1. If the OTP entered is valid, the app shall log the user into their account.
3. The app shall not give the user access to the application features if the user has not logged in.
  - 3.1. Users not registered shall be redirected to "Create Account" page.
4. The app shall provide a "Forgot Password" option to help users recover their account.
5. The app must integrate LTA & Google Maps API post successful login to fetch real-time data.

## **Travel Mode Selection:**

1. Users must choose between Individual Mode and Party Mode functionality.
  - 1.1. Upon selection of Individual Mode, the user will be prompted to fill in necessary data fields.
    - 1.1.1. The user must enter their departure location.
      - 1.1.1.1. The app must allow the option of using the user's current location for their departure location.
        - 1.1.1.1.1. The user must allow the application to access their device's GPS location services.
    - 1.1.2. The user must enter their destination of choice.
    - 1.1.3. The user can enter their preferred time of arrival at the destination.
  - 1.2. Upon selection of Party Mode, the user must choose between creating a party and joining a party.
    - 1.2.1. Upon selection of Create Party, the app must display the options to create a party.
      - 1.2.1.1. Users must be able to enter necessary details for the party.
        - 1.2.1.1.1. The detail must include the destination of choice.
          - 1.2.1.1.1.1. The user must be able to choose the destination from saved frequently visited places.
        - 1.2.1.1.2. The detail must include the preferred time of arrival at the destination.
      - 1.2.1.2. The app shall provide a QR code or Pin Number for other users to join the party.
      - 1.2.1.3. Users must be able to connect with multiple peers to synchronize travel.
        - 1.2.1.3.1. The app must show Realtime location of all peers.
        - 1.2.1.3.2. The app must send notifications if a member hasn't departed on time.

- 1.2.1.3.2.1. The app shall send the notification to all connected users
  - 1.2.1.3.2.1.1. The notification must include the affected user's username
  - 1.2.1.3.2.1.2. The notification must include the delayed estimated time of arrival.
- 1.2.2. Upon selection of Join Party, the app shall ask the user to input a QR code or Pin Number.
  - 1.2.2.1. The user must provide a valid QR code or Pin Number.
  - 1.2.2.2. Given a valid QR code or Pin Number, the app must connect the user to the party.
    - 1.2.2.2.1. The app must track real-time location of participants.
    - 1.2.2.2.2. The app must display real-time route of other participants.
- 1.2.3. After all users have joined the party, the app shall set the party creator as the party leader.
  - 1.2.3.1. The party leader must be able to change their destination of choice.
  - 1.2.3.2. The party leader must be able to change their preferred time of arrival at the destination.

### **Travel Details & Timings:**

- 1. Upon starting of the journey, the app must display a map with detailed information.
  - 1.1. The app must show information of different recommended routes to user.
    - 1.1.1. The app must show each recommended route path drawn on the map.
    - 1.1.2. The app must calculate and display the estimated time of arrival for each recommended route.
      - 1.1.2.1. The information must be showed to user on an overlay layer on the map.
    - 1.1.3. The app must calculate and display the waiting time at each bus stop along to route for each recommended route.

- 1.1.3.1. The information must be showed to user on an overlay layer on the map.
- 1.1.4. If the user is in party mode, the app must calculate and display individual arrival times so that each user knows who will arrive late.
  - 1.1.4.1. The information must be displayed under each user's avatar
- 1.1.5. The app must send reminder notifications to the respective user 10 minutes before their stipulated departure time.
- 1.2. The user must select a route among the recommended routes.
  - 1.2.1. Upon selection, the app must display the real-time location of all connected users within the party on the shared map.
  - 1.2.2. Upon selection, the app shall display the time the user can stay at home before needing to leave.
    - 1.2.2.1. The route must minimize user's wating time at the first bus stop.
  - 1.2.3. Upon selection, the app must calculate and display the time user needed to wait at the first bus stop, and every subsequent bus stop.
  - 1.2.4. Upon selection, the app must display the estimated arrival time on the screen.
    - 1.2.4.1. If the user is in party mode, the app must display if the user will arrive on time.
  - 1.2.5. Upon selection, the app must query and display the weather at the destination at the time of arrival.
- 2. The app should provide accurate predictions of bus arrival times based on real-time data for the next stop.
  - 2.1. The app must present a list of real-time bus arrival times.
  - 2.2. All times must be shown in a 24-hour format.

### **Weather Integration:**

- 1. The app must provide real-time weather updates for the entered destination at estimated arrival time.
  - 1.1. The app must provide real-time weather conditions.



- 1.2. The app must allow users to check weather without necessarily setting a destination.
2. The app shall send notifications and alerts if the latest forecast indicates there will be inclement weather at the destination at the chosen arrival time.

### **User Preferences:**

1. The map must allow users to save frequently visited locations.

## **2.2 Non-Functional Requirements**

### **Usability:**

1. 80% of first-time users must be able to query a route within 2 minutes of starting to use the app.
2. Users must be able to query a location using both Chinese and English.

### **Security:**

1. The app must mandate passwords that contain at least one uppercase letter, one lowercase letter, one special character, and have at least 8 characters.
2. After 5 consecutive failed login attempts, the account shall be temporarily locked for 10 minutes.
3. After 5 consecutive failed login attempts, an alert email or SMS must be sent.
4. All user location and data must be encrypted using methods no weaker than AES-128-bit encryption.
5. Quarterly security audits shall be conducted to confirm the safety and integrity of user data.
6. All requests must be sent using HTTPS or using TLS encryption, no clear text communication is allowed.

### **Compatibility:**

1. The app shall be compatible with all Android devices, ranging from Android version 9.0 to the latest available version.

**Performance:**

1. Real-time data fetching must not exceed a latency of 250ms.
2. The server must sustain a minimum request rate of 100 requests per second (RPS).

**Storage:**

1. Implement a scalable storage system that can accommodate an increase in user data including but not limited to saved routes, frequent locations, and party mode sessions.

**Reliability:**

1. The system's yearly downtime must not exceed 1%, guaranteeing 99% uptime.
2. After a system reboot, the full system functionality must be restored within 5 minutes.

**Scalability:**

1. The system must accommodate a 200% surge in user numbers during peak hours without degradation in performance.

**Network:**

1. Ensure seamless app functionality on both Wi-Fi and 4G/5G cellular data connections.

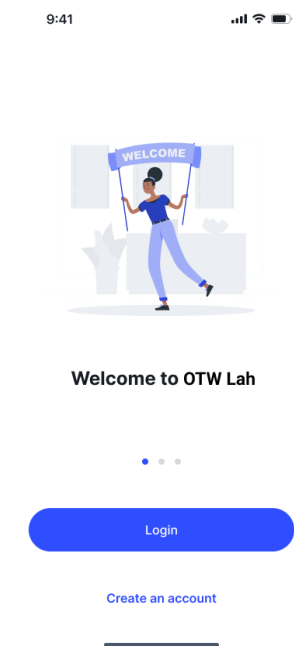
**Supportability:**

1. The database is a relational database, and it must be replaceable with any commercial product supporting standard SQL queries.

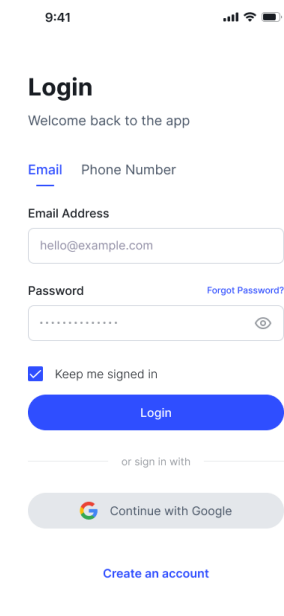
## 3. External Interface Requirements

### 3.1 User Interfaces




#### UI 1.1 Onboarding



#### UI 1.2 Email Login with Password



## UI 1.3 Phone Login with Password


9:41   

### Login


Welcome back to OTW Lah

Email [Phone Number](#)

Phone number [Login with OTP](#)

 +1 | Phone number


Password [Forgot Password?](#)

..... 

☒ Keep me signed in




Login

or sign in with

 Continue with Google

[Create an account](#)


## UI 1.4 Phone Login with OTP

9:41   

### Login

Please enter your phone number

Phone number

 +1 | Phone number

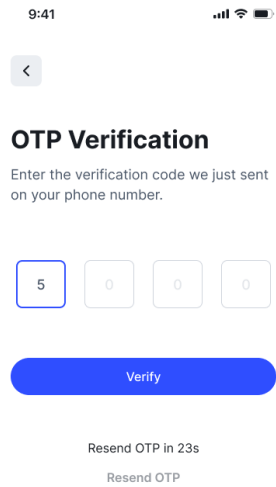
☒ Keep me signed in

Get OTP

or sign in with

Login with email

## UI 1.5 Verify OTP



9:41

<

### OTP Verification

Enter the verification code we just sent on your phone number.

5 0 0 0

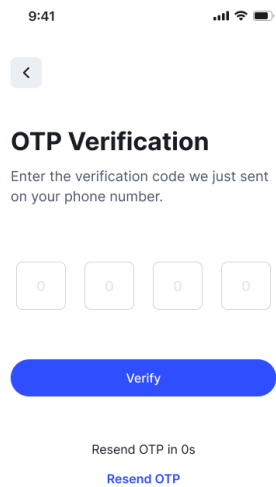
Verify

Resend OTP in 23s

Resend OTP

This screenshot shows the 'Verify OTP' screen. At the top, the status bar displays the time 9:41 and signal/battery icons. Below the back arrow, the title 'OTP Verification' is followed by the instruction 'Enter the verification code we just sent on your phone number.' There are four input fields for the code; the first contains '5' and the others are empty. A blue 'Verify' button is positioned below the inputs. At the bottom, a message states 'Resend OTP in 23s' with a 'Resend OTP' link.

## UI 1.6 Resend OTP



9:41

<

### OTP Verification

Enter the verification code we just sent on your phone number.

0 0 0 0

Verify

Resend OTP in 0s

Resend OTP

This screenshot shows the 'Resend OTP' screen. The layout is identical to UI 1.5, but the input fields are all empty (0 0 0 0). The 'Resend OTP in 0s' message is displayed, and the 'Resend OTP' link is active (blue text).

## UI 1.7 Forgot Password

9:41

<

**Forgot Password?**

Enter your email address to get the password reset link.

Email Address

hello@example.com

Password Reset

Create an account

## UI 1.8 Reset Password

9:41

**Reset Password**

Enter your new password twice below to reset a new password

Enter new password

152@@##PAss




Re-enter new password

.....

Reset Password

Create an account

## UI 1.9 Signup


9:41   

### Create an account

Name

Email Address


Password

By continuing, you agree to our [terms of service](#).

[Sign up](#)




or


 Continue with Google


Already have an account? [Sign in here](#)

---

## UI 1.10 Registered

9:41   





### Congratulations!

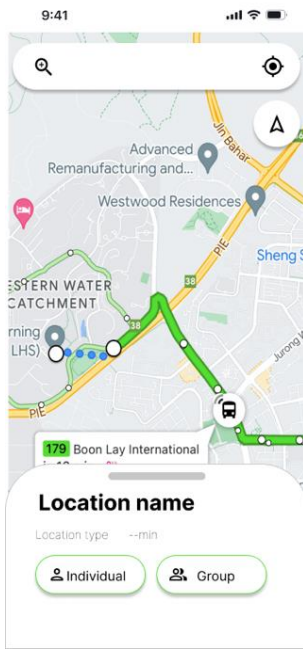
We've sent you a verification email, please check your inbox and follow the instructions to verify your account.

Thank you for signing up with us!

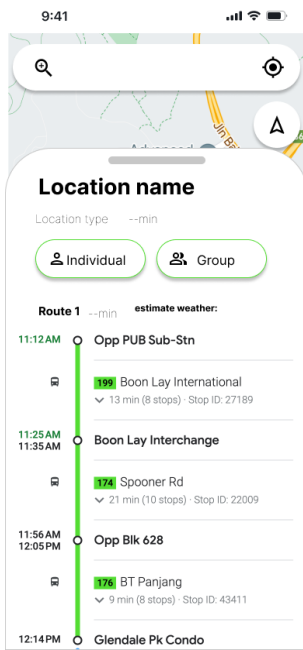
[Sign in here](#)

---

## UI 2.0 Group/Individual

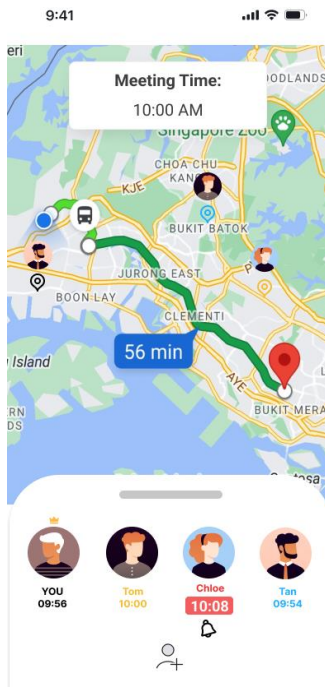


## UI 2.0 Group/Individual





## UI 2.1 View Route



## UI 2.2 Set Destination

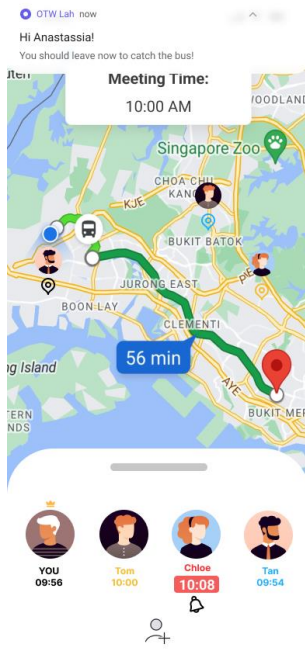
9:41

**Destination**  
Search

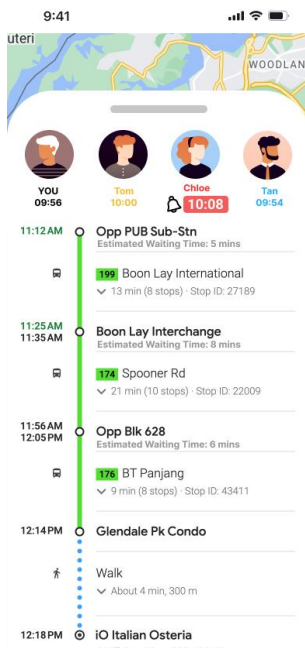
**Departure**  
Search

**Time**  
Date


## UI 2.3 Notification



## UI 2.4 Show Trip Details



### UI 3.1 Create Party

9:41   

**Invite friends to join!**



**PIN: 123456**



---

### UI 3.2 Join Party

9:41   

**Join session**

Party PIN

**OR**

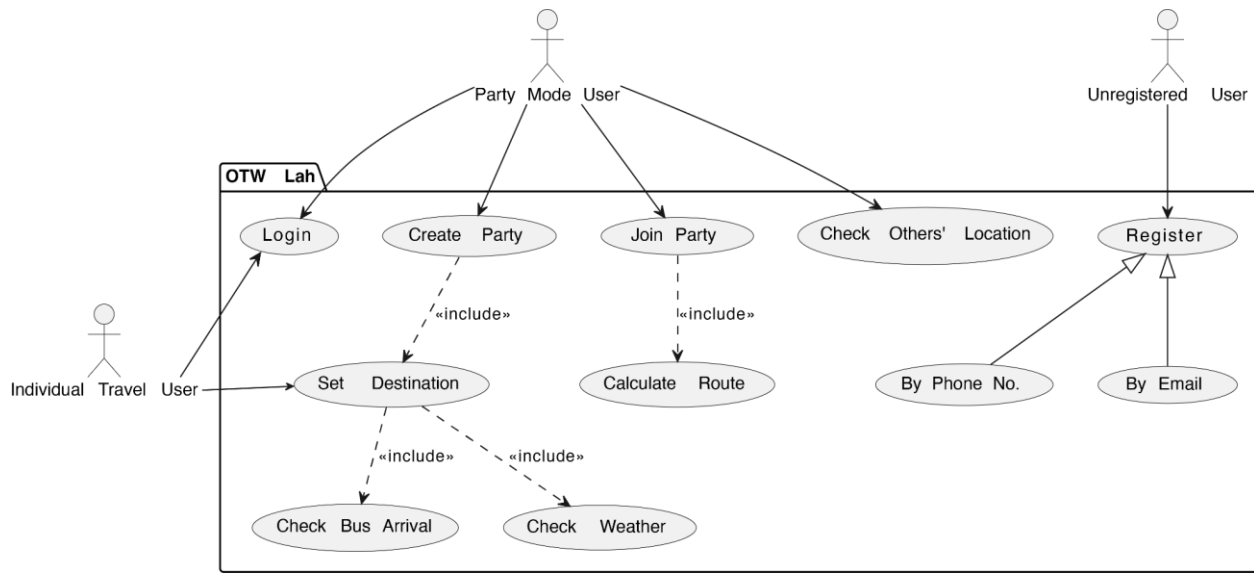


### UI 3.3 Scan QR Code To Join Party



## 4. System Use Cases

### 4.1 Use Case Diagram



## 4.2 Use Case Description

Use Case ID:	UC_001		
Use Case Name:	Create Account		
Created By:	Brian	Last Updated By:	Brian
Date Created:	29/8/2023	Date Last Updated:	29/8/2023

Actor:	Unregistered User
Description:	Unregistered user will need to register by email or phone number before starting to use the application
Preconditions:	-
Postconditions:	1. User will be registered to the application and will be able to login
Priority:	High
Frequency of Use:	One time usage
Flow of Events:	<ol style="list-style-type: none"> <li>1. User launches the application</li> <li>2. System will show a splash screen then prompt the user to register for an account either by phone number or email</li> <li>3. The user will enter their preferred phone number or email and enter a strong password</li> <li>4. The system verifies the data</li> <li>5. The system logs the user into their new account</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. User fails to provide a valid phone number or email:</li> <li>2. The system will display an error message telling the user that it is an invalid phone number/email</li> <li>3. User fails to provide a strong password:</li> <li>4. The app shall tell the user that a password must contain at least one uppercase letter, one</li> </ol>

	lowercase letter, one special character, and have at least 8 characters.
Exceptions:	-
Includes:	-
Special Requirements:	1. The user data must be stored in a secure database
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC_002		
Use Case Name:	Set Destination		
Created By:	Brian, Tianyu	Last Updated By:	Brian, Tianyu
Date Created:	29/8/2023	Date Last Updated:	29/8/2023

Actor:	Party Mode/Individual Travel User
Description:	The user can set a destination and get the best commuting options based on real-time data, weather conditions, and other factors.
Preconditions:	<ol style="list-style-type: none"> <li>1. The user has registered and logged into the app.</li> <li>2. The app has access to real-time data from the Singapore LTA API and Google Maps API.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The user receives an optimal route suggestion based on real-time data.</li> <li>2. The user is informed about the current weather conditions and bus arrival timings.</li> </ol>
Priority:	High
Frequency of Use:	Multiple times daily, depending on the user's commuting needs.
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user logs into the app.</li> <li>2. The user sets a desired destination.</li> <li>3. The app fetches and processes real-time data from integrated APIs.</li> <li>4. The app calculates the optimal route based on current conditions.</li> <li>5. The user is presented with a route suggestion, bus arrival timings, and weather conditions.</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. User adjusts the destination or prefers a different route than the suggested one.</li> </ol>



	<ol style="list-style-type: none"> <li>The user checks only bus arrival timings without setting a destination.</li> <li>The user checks only weather conditions without setting a destination.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>Failed login attempts.</li> <li>API unavailability or errors.</li> <li>Lack of internet connection.</li> <li>Real-time data not available for a particular route or transport mode.</li> </ol>
Includes:	<ol style="list-style-type: none"> <li>Calculate Route</li> <li>Check Bus Arrival</li> <li>Check Weather</li> </ol>
Special Requirements:	<ol style="list-style-type: none"> <li>The app needs real-time integration with Singapore LTA API and Google Maps API.</li> <li>The app must be able to process data quickly for a seamless user experience.</li> <li>The app should be capable of providing alternatives when real-time data isn't available.</li> </ol>
Assumptions:	<ol style="list-style-type: none"> <li>Users have an active internet connection.</li> <li>The APIs provide accurate and timely data.</li> <li>Users prefer the most efficient route based on time and not other factors (like scenery, fewer stops, etc.).</li> </ol>
Notes and Issues:	<ul style="list-style-type: none"> <li>The app's reliance on external APIs means there's potential for errors or inaccuracies beyond our control.</li> <li>Ensuring data privacy and security is crucial, especially when integrating multiple data sources.</li> </ul>

- |  |  |
|--|--|
|  | <ul style="list-style-type: none"><li>- As with all real-time data apps, there's a potential delay between data retrieval and data display which could affect the accuracy of route suggestions.</li><li>- This description aims to cover the primary functionalities and scenarios an "Individual Travel User" would experience. Adjustments may be needed based on further feedback or iterative design processes.</li></ul> |
|--|--|

Use Case ID:	UC_003		
Use Case Name:	Login		
Created By:	Brian, Tianyu	Last Updated By:	Brian, Tianyu
Date Created:	29/8/2023	Date Last Updated:	29/8/2023

Actor:	Party Mode/Individual Travel User
Description:	User can login to the application after they are registered via phone number or email and their password
Preconditions:	<ol style="list-style-type: none"> <li>1. The user must be registered in the system</li> <li>2. The user's device has an active internet connection</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The user's session is established</li> </ol>
Priority:	High
Frequency of Use:	Every time a user wants to access the application
Flow of Events:	<ol style="list-style-type: none"> <li>1. User launches the application.</li> <li>2. User is presented with a login screen.</li> <li>3. User enters their mobile number or email address.</li> <li>4. User is provided with the option to enter a password or receive an OTP.</li> <li>5. If user chooses the password option: <ol style="list-style-type: none"> <li>a. User enters their password.</li> <li>b. The system validates the credentials.</li> <li>c. If valid, the user is granted access to the app.</li> </ol> </li> <li>6. If user chooses the OTP option: <ol style="list-style-type: none"> <li>a. The system sends an OTP to the entered mobile number or email address.</li> <li>b. User enters the received OTP.</li> </ol> </li> </ol>

	c. If the OTP matches, the user is granted access to the app.
Alternative Flows:	<ol style="list-style-type: none"> <li>1. If the user is not registered:</li> <li>2. The system prompts the user to create an account.</li> <li>3. The user is redirected to the "Create Account" page.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. If the entered password is incorrect: <ol style="list-style-type: none"> <li>a. The system notifies the user and prompts to re-enter the password or opt for the OTP method.</li> </ol> </li> <li>2. If the entered OTP is incorrect or expired: <ol style="list-style-type: none"> <li>a. The system notifies the user and provides an option to resend the OTP.</li> </ol> </li> <li>3. If the system fails to send the OTP due to technical issues: <ol style="list-style-type: none"> <li>a. The user is notified of the issue and prompted to try again later or use the password method.</li> </ol> </li> </ol>
Includes:	<ol style="list-style-type: none"> <li>1. Create Account (If the user needs to register)</li> <li>2. Integrate LTA &amp; Google Maps API (Post successful login to fetch real-time data)</li> </ol>
Special Requirements:	<ol style="list-style-type: none"> <li>1. The system must ensure that user credentials are stored securely.</li> <li>2. The OTP mechanism should be time-sensitive for enhanced security.</li> </ol>
Assumptions:	<ol style="list-style-type: none"> <li>1. The user has access to the mobile number or email address they used to register.</li> </ol>

	<ol style="list-style-type: none"><li>2. There's a mechanism in place to handle multiple failed login attempts to prevent brute force attacks.</li></ol>
Notes and Issues:	<ol style="list-style-type: none"><li>1. There should be an easy way for users to retrieve/reset their passwords in case they forget.</li><li>2. For the OTP mechanism, there might be potential delays due to network issues which should be considered</li></ol>

Use Case ID:	UC_004		
Use Case Name:	Calculate Route		
Created By:	Brian	Last Updated By:	Brian
Date Created:	29/8/2023	Date Last Updated:	29/8/2023

Actor:	Party Mode/Individual Travel User
Description:	The system will display the best route to be taken by the user after a destination has been set with information such as estimated travel time and bus to take.
Preconditions:	<ol style="list-style-type: none"> <li>1. The user has a registered account and is authenticated.</li> <li>2. The user has set their travel details, including the starting point and destination.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The route is displayed to the user on the application with the estimated time and bus to take</li> </ol>
Priority:	High
Frequency of Use:	Multiple times daily, depending on the user's commuting needs.
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user logs into the application</li> <li>2. The user sets their travel details, including the starting point and destination.</li> <li>3. The system retrieves the user's previously set starting point and destination.</li> <li>4. The system calculates and displays the recommended travel route from the starting point to the destination.</li> <li>5. The user can view the route, including directions, estimated travel time, and transportation options.</li> </ol>

Alternative Flows:	<ol style="list-style-type: none"><li>1. If the user wants to change their destination after initiating this use case:</li><li>2. The user can update the destination, and the system recalculates the route accordingly.</li></ol>
Exceptions:	<ol style="list-style-type: none"><li>1. If there are no available routes between the provided starting point and destination:</li><li>2. The system notifies the user that no routes are available for the specified locations.</li></ol>
Includes:	-
Special Requirements:	<ol style="list-style-type: none"><li>1. The system must have access to real-time transportation data and maps, likely through the integration of LTA (Land Transport Authority) and Google Maps APIs</li></ol>
Assumptions:	<ol style="list-style-type: none"><li>1. The user has successfully created an account, logged in, and set their travel details before initiating this use case.</li><li>2. The integration of external APIs for map and transportation data is available and functional.</li></ol>
Notes and Issues:	<ol style="list-style-type: none"><li>1. This use case is crucial for providing users with accurate and up-to-date travel route information within the application</li><li>2. The accuracy and availability of route information depend on the data provided by external APIs, such as LTA and Google Maps.</li><li>3. Users can benefit from the convenience of having their travel routes calculated automatically based on their specified destinations.</li></ol>

Use Case ID:	UC_005		
Use Case Name:	Check Bus Arrival		
Created By:	Brian	Last Updated By:	Brian
Date Created:	29/8/2023	Date Last Updated:	29/8/2023

Actor:	Party Mode/Individual Travel User
Description:	This use case outlines the process by which a user checks the estimated arrival time of a bus at a specified location within the application
Preconditions:	<ol style="list-style-type: none"> <li>1. The user has a registered account and is authenticated.</li> <li>2. The user has set their travel details, including the starting point and destination.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The system provides the user with the estimated arrival time of a bus at the specified location.</li> </ol>
Priority:	Medium
Frequency of Use:	Occasional as it depends on whether the user takes the bus often
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user sets their travel details, including the starting point and destination.</li> <li>2. The system prompts the user to enter the location or bus stop where they want to check bus arrival.</li> <li>3. The user provides the required information.</li> <li>4. The system retrieves real-time data from the integrated LTA (Land Transport Authority) and Google Maps API.</li> <li>5. The system displays the estimated arrival time of the next bus at the specified location.</li> </ol>



Alternative Flows:	1. The user wants to check the bus arrival time at another bus stop
Exceptions:	1. There are no bus arrival times at selected location
Includes:	-
Special Requirements:	1. The system must have access to real-time transportation data from external APIs, such as LTA and Google Maps.
Assumptions:	-
Notes and Issues:	<ol style="list-style-type: none"><li>1. This use case enhances the user experience by providing real-time information about bus arrivals, helping users plan their journeys more effectively.</li><li>2. The accuracy and availability of bus arrival information depend on the data provided by external APIs, such as LTA and Google Maps.</li></ol>

Use Case ID:	UC_006		
Use Case Name:	Check Weather		
Created By:	Brian	Last Updated By:	Brian
Date Created:	29/8/2023	Date Last Updated:	29/8/2023

Actor:	Party Mode/Individual Travel User
Description:	The user can check the weather preemptively through the application for the destination before travelling
Preconditions:	<ol style="list-style-type: none"> <li>1. The user has a registered account and is authenticated.</li> <li>2. The user has set their travel details, including the starting point and destination.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The system provides the user with current weather information for the specified travel destination.</li> </ol>
Priority:	Medium
Frequency of Use:	Occasional
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user sets their travel details, including the starting point and destination.</li> <li>2. The system retrieves real-time weather data for the specified destination using integrated weather services.</li> <li>3. The system displays the current weather conditions, including temperature, humidity, wind speed, and forecasts, if available, for the destination.</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. User wants to check weather for another location</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. Weather API is down and could not check the weather of location</li> </ol>

Includes:	-
Special Requirements:	1. The system must have access to real-time weather data from integrated weather services.
Assumptions:	1. The integration of external weather services is available and functional.
Notes and Issues:	<ol style="list-style-type: none"><li>1. This use case enhances the user experience by providing real-time weather information for their travel destination, helping users make informed decisions for their journeys.</li><li>2. The accuracy and availability of weather information depend on the data provided by external weather services.</li></ol>

Use Case ID:	UC_007		
Use Case Name:	Create Party		
Created By:	Brian	Last Updated By:	Brian
Date Created:	29/8/2023	Date Last Updated:	29/8/2023

Actor:	Party Mode User
Description:	The user can create a party and add their friends using a PIN or a QR code generated by the application.
Preconditions:	<ol style="list-style-type: none"> <li>1. The user has a registered account and is authenticated.</li> <li>2. The user has the application installed with proper permissions enabled.</li> <li>3. The user's device has an active internet connection and GPS/location services enabled.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. A party event is created with a unique PIN and QR code.</li> <li>2. Participants joining the party using the PIN or QR code get individualized departure times.</li> <li>3. The user and participants can track each other's real-time location.</li> </ol>
Priority:	High
Frequency of Use:	Occasional
Flow of Events:	<ol style="list-style-type: none"> <li>1. User creates a party.</li> <li>2. User sets a destination and desired arrival time.</li> <li>3. The app generates a unique PIN and QR code.</li> <li>4. User shares the PIN or QR code with friends.</li> <li>5. Friends use the PIN or QR code to join the party.</li> <li>6. The app calculates and displays personalized departure times for each participant.</li> </ol>

	7. Users can track the real-time location of other participants.
Alternative Flows:	1. If the real-time data API is down or unavailable, the app can use average wait times and travel data to compute estimated departure times.
Exceptions:	<ol style="list-style-type: none"> <li>1. In case of data discrepancies from the LTA or Google Maps API, the app might not be able to provide accurate departure suggestions.</li> <li>2. If a participant's location services are turned off or inaccessible, their real-time location won't be available for tracking.</li> </ol>
Includes:	<ol style="list-style-type: none"> <li>1. Integrate LTA &amp; Google Maps API</li> <li>2. Team Up Using QR Code</li> <li>3. Team Up Using PIN</li> </ol>
Special Requirements:	<ol style="list-style-type: none"> <li>1. Securely manage user location data.</li> <li>2. The application must have access to updated transportation schedules and real-time transit data.</li> <li>3. Ability to generate unique QR codes and PINs for each party event.</li> </ol>
Assumptions:	<ol style="list-style-type: none"> <li>1. Real-time data from the LTA and Google Maps API is accurate and up-to-date.</li> <li>2. Participants have location services turned on when using the app.</li> </ol>
Notes and Issues:	<ol style="list-style-type: none"> <li>1. Privacy concerns need to be addressed, especially concerning real-time location tracking.</li> <li>2. Ensure there's a feature to opt-out from location sharing within the party mode.</li> </ol>

	3. Potential issues if multiple parties are being organized simultaneously by a user.
--	---

Use Case ID:	UC_008		
Use Case Name:	Join Party		
Created By:	Brian	Last Updated By:	Brian
Date Created:	29/8/2023	Date Last Updated:	29/8/2023

Actor:	Party Mode User
Description:	The user can join a party created by their friend using the QR code or PIN that was generated. The user can also monitor the real-time location of other participants to determine their progress and potential delays.
Preconditions:	<ol style="list-style-type: none"> <li>1. User is logged into the application.</li> <li>2. The application has access to real-time transportation data from the Singapore LTA API and Google Maps API.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. A party event is created with a unique PIN and QR code.</li> <li>2. Participants joining the party using the PIN or QR code get individualized departure times.</li> <li>3. The user and participants can track each other's real-time location.</li> </ol>
Priority:	High
Frequency of Use:	Occasional
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user selects the 'Party Mode' feature from the main app interface.</li> <li>2. The user is prompted to either scan a QR code or enter a PIN.</li> </ol>

	<ol style="list-style-type: none"> <li>3. The user scans the QR code or inputs the PIN.</li> <li>4. The app connects to the server, retrieves the shared destination and estimated arrival time.</li> <li>5. The app calculates the optimal route and departure time for the user, ensuring minimal wait times at transit stops.</li> <li>6. The displayed route provides detailed waiting times for possible routes, including the first bus stop and any transfers.</li> <li>7. The user is also presented with the real-time locations of other participants in the party mode session.</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. If the QR code scan fails or the PIN is incorrect, the user is prompted to try again or seek a valid QR code or PIN.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. Server connectivity issues prevent fetching party details.</li> <li>2. Real-time data for some routes might be unavailable.</li> </ol>
Includes:	<ol style="list-style-type: none"> <li>1. Real-time Journey Estimates</li> <li>2. Personalized Departure Suggestions</li> </ol>
Special Requirements:	<ol style="list-style-type: none"> <li>1. Access to a camera (for scanning the QR code)</li> <li>2. Real-time connectivity to the server</li> <li>3. Updated route and transit data</li> </ol>
Assumptions:	<ol style="list-style-type: none"> <li>1. The app's real-time data is consistently updated and accurate.</li> <li>2. The user has good internet connectivity to access real-time features.</li> </ol>

Notes and Issues:	<ol style="list-style-type: none"><li data-bbox="609 195 1333 342">1. The app may need to consider the reliability and frequency of real-time data updates from the Singapore LTA API and Google Maps API.</li><li data-bbox="609 363 1317 464">2. Privacy concerns regarding the sharing of real-time locations among users.</li><li data-bbox="609 485 1360 705">3. Potential challenges in ensuring that all participants arrive simultaneously, especially when there's a large variation in individual starting locations or transit options.</li></ol>
-------------------	--



Use Case ID:	UC_009		
Use Case Name:	Check Others' Location		
Created By:	Brian	Last Updated By:	Brian
Date Created:	29/8/2023	Date Last Updated:	29/8/2023

Actor:	Party Mode User
Description:	The user will be able to check the real-time location of their party members once they are in the party
Preconditions:	<ol style="list-style-type: none"> <li>1. The user has logged into the application.</li> <li>2. The user is part of a group where participants' locations are being tracked.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The system provides the user with real-time location information for other participants in the group.</li> </ol>
Priority:	High
Frequency of Use:	Occasional as it depends on how many times the user is part of a party
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user joins or creates a group for a particular journey</li> <li>2. The system retrieves and displays the real-time location information of other participants in the same group on a map.</li> <li>3. The user can see the locations of fellow group members, possibly with markers or icons representing their positions.</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. Lack of permissions to the user's device location for any of the party members</li> </ol>
Exceptions:	-
Includes:	-

Special Requirements:	-
Assumptions:	-
Notes and Issues:	<ol style="list-style-type: none"><li>1. This use case enhances the user experience during group travel by providing a way to monitor the locations of fellow travelers, improving coordination and safety.</li><li>2. Privacy and consent considerations should be taken into account when implementing location-sharing features within the application.</li></ol>

## Appendix A: Data Dictionary

Term	Description	Data Type	Constraints
<b>1. User</b>			
<b>Username</b>	Text. Unique identifier for users.	String	Unique
<b>Email Address</b>	Text. Used for communication, recovery, and OTP verification.	String	Unique
<b>Contact Number</b>	Text. Used for OTP verification.	String	Unique
<b>Password</b>	Encrypted Text. Used for authentication.	String	Has at least one uppercase letter, one lowercase letter, one special character, and have at least 8 characters.
<b>OTP</b>	One time password. Numeric. Temporary password for authentication.	String	Fixed length
<b>Preferred Locations</b>	List of Text. Frequently visited places for faster selection.	String list	NA
<b>2. API Integration</b>			
<b>LTA &amp; Google Maps Token</b>	Encrypted Text. Access tokens for API integrations.	String	NA
<b>API Response Data</b>	Dynamic. Real-time transportation and map data.	String	NA
<b>3. Travel Mode Functions</b>			
<b>Mode Type</b>	Text. Individual or Party mode.	String/Enum	Limited to two types
<b>Departure Location</b>	Text. Point of the start of the journey.	String	Valid location
<b>Destination</b>	Text. Point of the end of the journey.	String	Valid location
<b>Preferred Arrival Time</b>	Time. Desired time to reach the destination.	Time	NA
<b>4. Party Mode Extra Functions</b>			
<b>Party ID (PIN)</b>	Numeric. Unique identifier for parties.	Int	Unique
<b>QR Code</b>	Encoded Data. Used for adding members to the party.	QR code	NA
<b>PIN Number</b>	Numeric. Alternative to QR Code for adding members.	Int	Unique (in valid time)
<b>Party Members</b>	List of Users. Users in a particular party.	User list	Valid User ID
<b>Party Leader</b>	User. Creator of the party.	User	Valid User ID
<b>5. Travel Details &amp; Timings</b>			

<b>Bus Arrival Timings</b>	Time. Real-time data for bus arrivals.	Time list	NA
<b>Route Suggestions</b>	List of Routes. Possible routes to the destination.	Route Class	
<b>Selected Route</b>	Route. Chosen route by the user to the destination.	Route Class	
<b>Alternative Routes</b>	List of Text. Other possible routes to the destination.	Route Class	
<b>Route History</b>	List of Text. Previously selected routes.	Route Class	
<b>6. Weather Integration</b>			
<b>Time</b>	Weather information time	Time	NA
<b>Temperature</b>	Temperature of given time and location	Double	In °C
<b>Precipitation</b>	Amount of precipitation	Double	$\geq 0, \leq 1$
<b>Weather condition</b>	Description of weather condition	Enum	
<b>Grid data</b>	List of data describe weather condition of certain location	TBC	NA
<b>Weather Map</b>	Static map of weather in given time, retrieve from data source	TBC	NA
<b>Weather Data</b>	Dynamic Data. Real-time weather information.	TBC	
<b>Destination Weather</b>	Text. Weather at the destination.	TBC	
<b>Alerts</b>	Text. Notifications for bad weather.	TBC	
<b>7. Login &amp; Authentication</b>			
<b>Login Method</b>	Text. Either using a password or OTP.	Boolean	
<b>Session</b>	Dynamic. User's active session after login.	String	
<b>Failed Attempts</b>	Numeric. Count of failed login attempts.	Short	< Attempt Limit
<b>8. Party Creation &amp; Management</b>			
<b>Party Status</b>	Text. Active, completed, or canceled.	Enum	
<b>Participant Locations</b>	List of Dynamic. Real-time locations of party members.	Dynamic Location List	
<b>Shared Map</b>	Dynamic. Live map view shared among party members.	XML Data	