

# Passenger Mobile App

# Introduction



- Part of the FLEV Booking System
- Ride-hailing and delivery app
  - Initially developed for Local Government Units
- Services
  - Private Trip
  - Shared Trip
    - Service Unit Trip
    - Scheduled Trip
  - Delivery Request
- Accommodates or allocates passengers accordingly
- [Overall FLEV Booking System short video](#)

# Technologies

## Programming Languages

- Javascript, Typescript

## Frameworks

- [React Native](#)
- [Expo](#)

## Query Language for API

- [GraphQL](#)

# Setup and configurations

## Requirements

- Node v14

## Setup

- Copy `.env.example` to `.env` and fill in the values.
- `npm i` to install dependencies

## Running the App

- `npm run development` to run the app in development.
- `npm run start` to run the app in production.

## Testing GraphQL queries

- `npm run test` to validate the queries

# File directory structure

## Packages

- `Package.json`

## Environment variables

- `.env*`

## Compilers

- `jsconfig.json`, `tsconfig.json`

## Coding standards

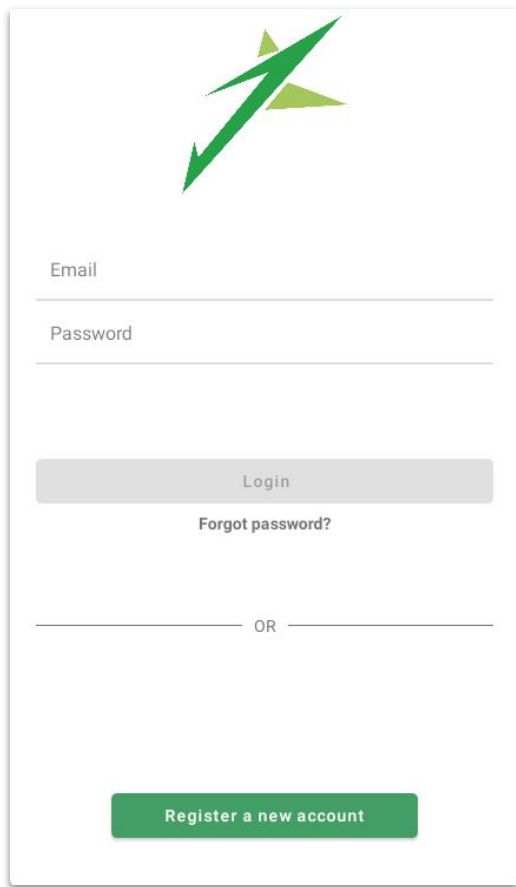
- `.eslintrc.js`, `.prettierrc.js`, `.editorconfig`

## Directories

- [directories](#)



# Unrestricted Screens

# Login




The login screen features a green logo at the top left, consisting of a stylized 'Z' or '7' shape. Below the logo are two text input fields: 'Email' and 'Password'. A 'Login' button is positioned below the password field, with a 'Forgot password?' link underneath it. A horizontal line with the text 'OR' in the center separates the login section from the registration section. At the bottom, there is a green button labeled 'Register a new account'.

## Login Screen

- Screen that authenticates the passenger.
- “Email” text field
  - Should be existing and approved passenger
  - Should easily be cleared by pressing “x” when filled up
- “Password” text field
  - When filled up
    - hide password - 
    - show password - 
- “Login” button
  - Uses `passengerLoginByPassword` mutation call to get **passenger details** and **token**
  - Goes to **Service Options Screen** when passenger is authenticated otherwise, shows an error message
- “Forgot Password?” label
  - When pressed, goes to **Forgot Password Screen**
- “Register a new account” button
  - When pressed, goes to **Name Screen**

# Login



Email

Password

Login

Forgot password?

OR

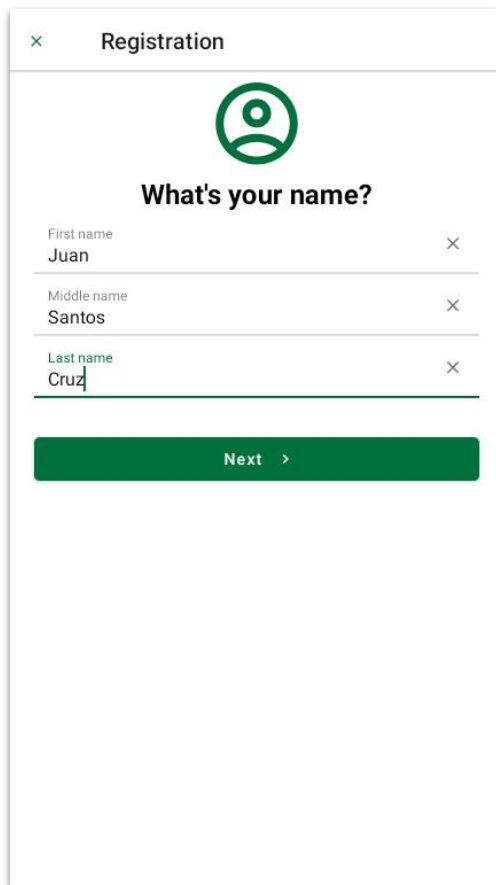
Register a new account

## Login Screen

- Possible Errors
  - Internal server error - Something has gone wrong in the application
  - Invalid credentials error - Invalid email or password.
  - Pending user registration error - Passenger have a pending registration.
  - Declined user registration error - Passenger registration has been declined.



# Registration

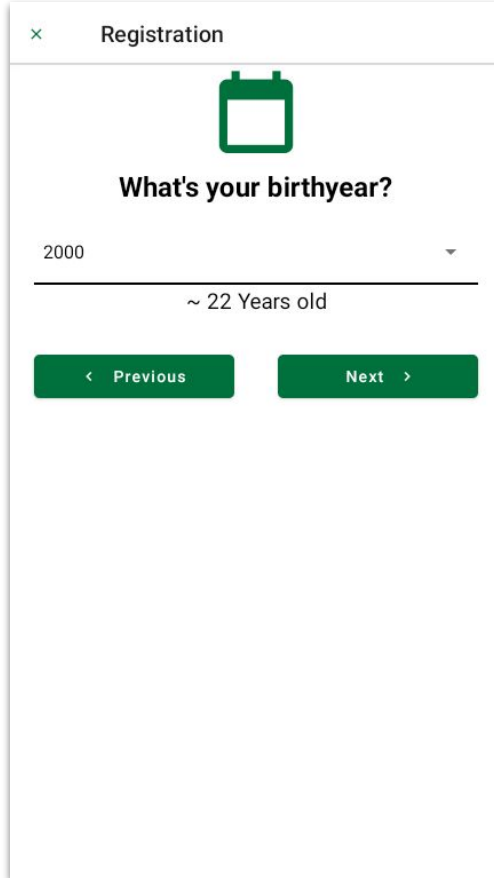


The image shows a mobile app registration screen titled "Registration" with a close button (X) in the top left. Below the title is a green circular icon with a white person silhouette. The main heading is "What's your name?". There are three text input fields: "First name" with the value "Juan", "Middle name" with the value "Santos", and "Last name" with the value "Cruz". Each field has a small "X" button to its right for clearing the text. At the bottom is a large green button with the text "Next >".

## Name Screen

- Screen that asks for passenger's name. Part of the registration screens.
- Text fields (Passenger name)
  - Should easily be cleared by pressing "X" when filled up
  - Should only contain letters a-z and ñ
  - Should only contain single space(s)
  - If there is/are error(s), it should be visible below the text field.
- "Next" button
  - Should be pressable when all fields are valid
  - Proceeds to **Birthyear Screen**
- "X" header button
  - Should go back to **Login Screen**

# Registration

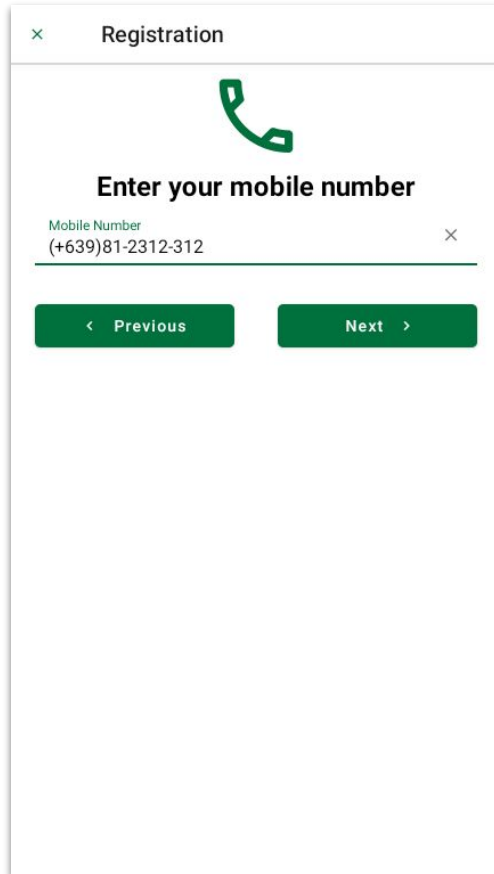


The image shows a mobile app registration screen titled "Registration" with a close button (X) in the top left. Below the title is a green calendar icon. The main text asks "What's your birthyear?". Below this is a year picker showing "2000" with a dropdown arrow. A horizontal line separates the year picker from the age estimate "~ 22 Years old". At the bottom are two green buttons: "< Previous" and "Next >".

## Birthyear Screen

- Screen that asks for passenger's birthyear. Part of the registration screens.
- "Year" picker
  - Should be able to select from 1940 to the current year
  - Should display the passenger's age.
  - Passenger's age should be  $\geq 18$  and  $< 100$ .
- "Previous" button
  - Goes back to **Name Screen**
- "Next" button
  - Should be pressable when all fields are valid
  - Proceeds to **Mobile Screen**
- "X" header button
  - Should go back to **Login Screen**

# Registration

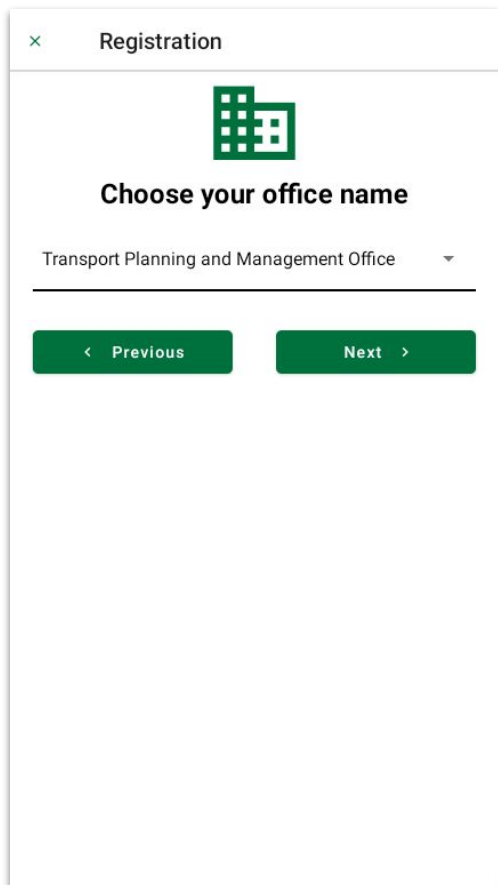


The image shows a mobile application screen titled "Registration" with a close button (X) in the top left corner. Below the title is a green telephone handset icon. The main heading is "Enter your mobile number". Below this is a text input field with the placeholder text "Mobile Number" and the example number "(+639)81-2312-312". A small "X" icon is on the right side of the input field. At the bottom, there are two green buttons: "Previous" with a left arrow and "Next" with a right arrow.

## Mobile Number Screen

- Screen that asks for passenger's mobile number. Part of the registration screens.
- "Mobile Number" text field
  - o Should have a (+639) prefix
  - o Should be automatically formatted in this style (+639)xx-xxxx-xxx
  - o Should be 9 digit.
  - o Should easily be cleared by pressing "X" when filled up
  - o If there is/are error(s), it should be visible below the text field.
- "Previous" button
  - o Goes back to **Birthyear Screen**
- "Next" button
  - o Should be pressable when all fields are valid
  - o Proceeds to **Office Name Screen**
- "X" header button
  - o Should go back to **Login Screen**

# Registration

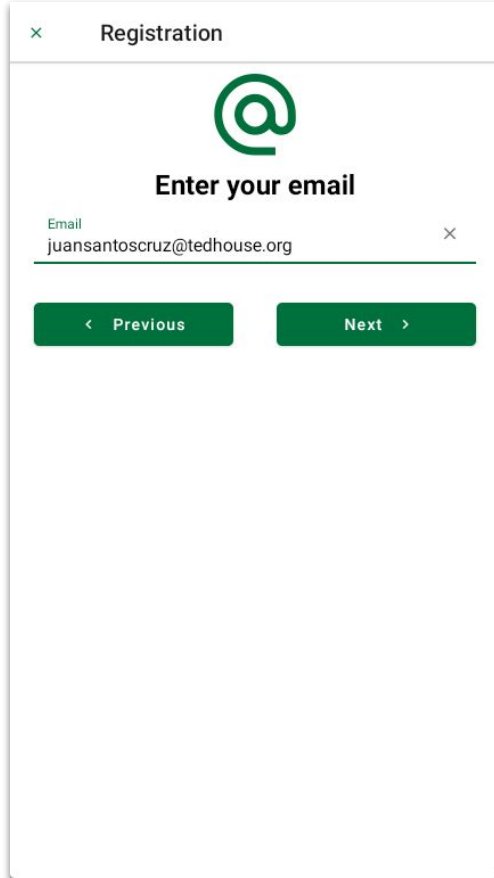


The screenshot shows a mobile application interface for the 'Registration' process. At the top, there is a header bar with a close button (an 'x' icon) and the title 'Registration'. Below the header, there is a green icon representing a building. The main heading is 'Choose your office name'. Underneath, there is a dropdown menu currently displaying 'Transport Planning and Management Office'. At the bottom of the screen, there are two green buttons: 'Previous' with a left arrow and 'Next' with a right arrow.

## Office Name Screen

- Screen that asks for passenger's office name. Part of the registration screens.
- "Office Name" picker
  - Should include the following: Offices of General Services, Transport Planning and Management Office, Others
- "Previous" button
  - Goes back to **Mobile Number Screen**
- "Next" button
  - Proceeds to **Email Screen**
- "X" header button
  - Should go back to **Login Screen**

# Registration

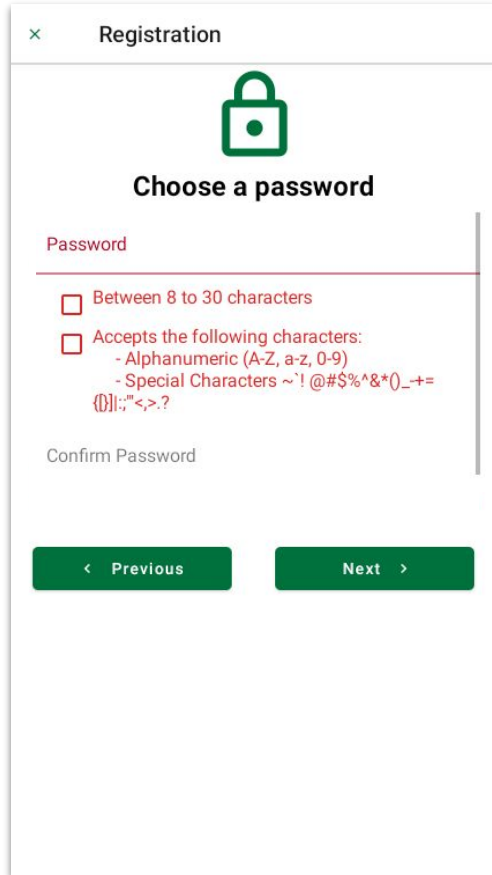


The mockup shows a mobile app screen titled "Registration" with a close button (X) in the top left. Below the title is a large green "@" icon and the text "Enter your email". A text input field contains the email "juansantoscruz@tedhouse.org" with a small "Email" label above it and a clear button (X) to its right. At the bottom are two green buttons: "< Previous" and "Next >".

## Email Screen



- Screen that asks for passenger's email. Part of the registration screens.
- "Email" text field
  - Should be valid email address
  - Should easily be cleared by pressing "X" when filled up
  - If there is/are error(s), it should be visible below the text field.
- "Previous" button
  - Goes back to **Office Name Screen**
- "Next" button
  - Should be pressable when all fields are valid
  - Proceeds to **Password Screen**
- "X" header button
  - Should go back to **Login Screen**

# Registration




The mockup shows a registration screen with a title bar 'Registration' and a close button. Below is a green padlock icon and the heading 'Choose a password'. A 'Password' text field is present. Below it, a checklist of password requirements is shown with red checkboxes and text: 'Between 8 to 30 characters' and 'Accepts the following characters: - Alphanumeric (A-Z, a-z, 0-9) - Special Characters ~!@#\$%^&\*()\_+={[]|:;'\<,>.?'. Below the checklist is a 'Confirm Password' text field. At the bottom are two green buttons: '< Previous' and 'Next >'.

## Password Screen

- Screen that asks for passenger's password. Part of the registration screens.
- "Password" text field
  - When filled up
    - hide password - 
    - show password - 
  - Should be between 8 to 30 characters
  - Accepts the following characters:
    - Alphanumeric (A-Z, a-z, 0-9)
    - Special Characters ~!@#\$%^&\*()\_+={[]|:;'\<,>.?}
- "Password" checklist
  - Should be checked when password criteria is matched
- "Confirm Password" text field
  - Should matched "Password". Otherwise, error will show below the text field.

# Registration

× Registration



Choose a password

Password

☐ Between 8 to 30 characters

☐ Accepts the following characters:

- Alphanumeric (A-Z, a-z, 0-9)
- Special Characters ~`! @#\$%^&\*()\_+= {}|;:~" <, >, . ?

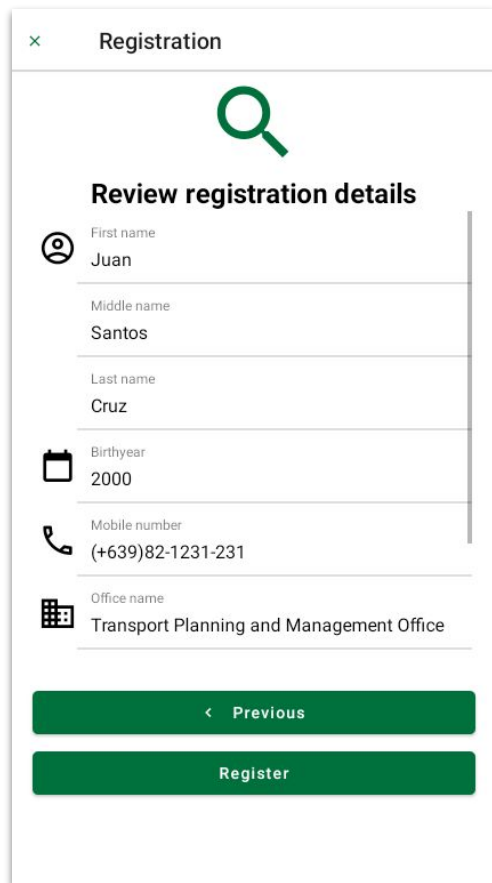
Confirm Password

< Previous Next >

## Password Screen



- Screen that asks for passenger's password. Part of the registration screens.
- "Previous" button
  - Goes back to **Email Screen**
- "Next" button
  - Should be pressable when all fields are valid
  - Proceeds to **Send Registration Screen**
- "X" header button
  - Should go back to **Login Screen**

# Registration



The image shows a mobile app registration screen. At the top, there's a header with a close button (X) and the title 'Registration'. Below the header is a green magnifying glass icon. The main section is titled 'Review registration details'. It contains several input fields with labels and icons: 'First name' (Juan), 'Middle name' (Santos), 'Last name' (Cruz), 'Birthyear' (2000), 'Mobile number' ((+639)82-1231-231), and 'Office name' (Transport Planning and Management Office). At the bottom, there are two green buttons: 'Previous' and 'Register'.


## Send Registration Screen

- Screen that lets the passenger review their details. Part of the registration screens.
- Fields
  - Should be scrollable.
  - should reflect the input provided by the passenger
- “Password” text field
  - When filled up
    - hide password - 
    - show password - 
- “Previous” button
  - Goes back to **Email Screen**
- “Register” button
  - Sends the registration to server
  - Uses `createUserRegistration` mutation send registration to server.
  - Goes to **Send Notification Screen** when successful. Otherwise, shows an error message
- “X” header button
  - Should go back to **Login Screen**




# Registration

× Registration



**Review registration details**


 First name  
Juan

Middle name


Santos

Last name


Cruz

 Birthyear

2000

 Mobile number

(+639)82-1231-231

 Office name

Transport Planning and Management Office

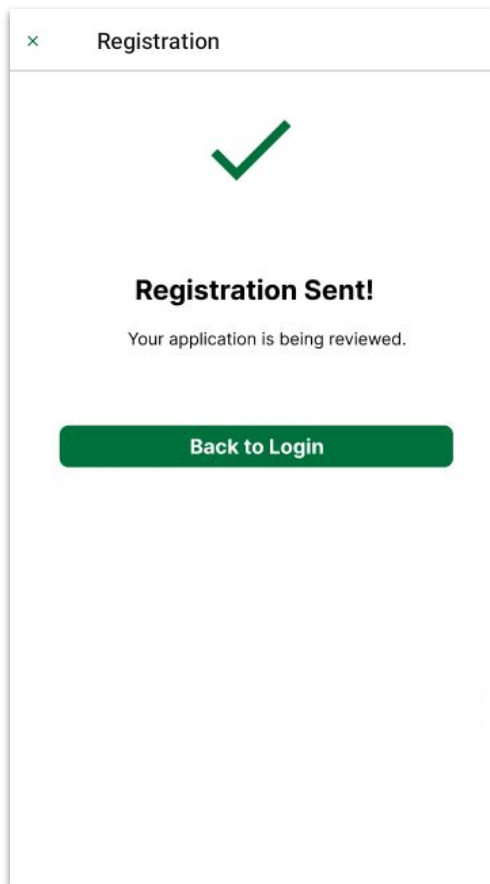
< Previous

Register

## Send Registration Screen

- Possible errors when sending registration
  - User email already exists error - Email address already exists or is taken by another passenger.
  - User mobile number already exists error - Mobile number already exists or is taken by another passenger.

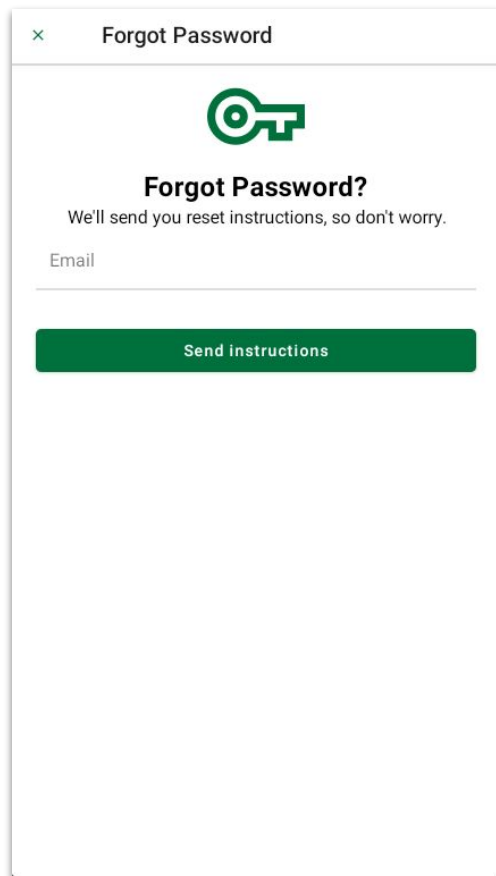
# Registration



## Send Notification Screen

- Screen that notifies the passenger that the registration is sent to the server.
- "X" header and "Back to Login" button should go back to **Login Screen**

# Forgot Password

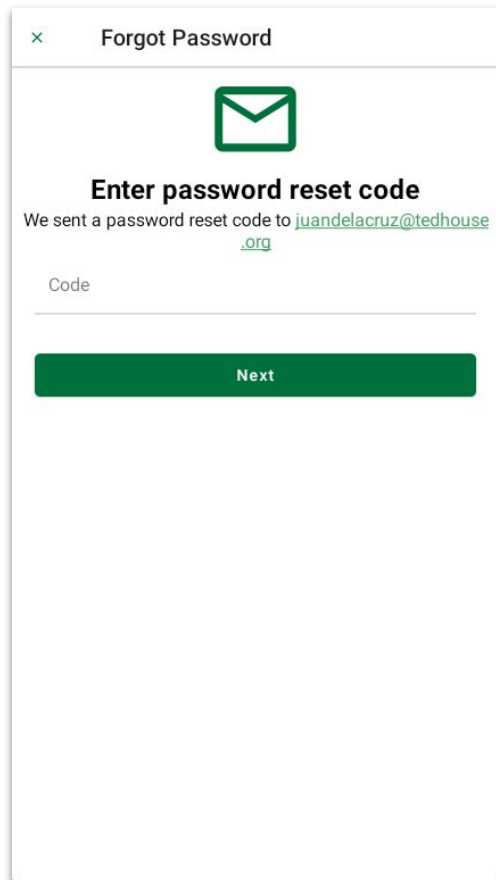


The image shows a mobile app screen titled "Forgot Password". At the top left is a close button (an 'x' icon). Below the title is a green key icon. The text "Forgot Password?" is centered, followed by the subtitle "We'll send you reset instructions, so don't worry." Below this is a text input field labeled "Email". At the bottom is a large green button labeled "Send instructions".

## Email Screen

- Screen that asks for passenger's email. Server will send a reset instructions to the email.
- "Email" text field
  - Should be existing and approved passenger
  - Should easily be cleared by pressing "x" when filled up
- "Send Instructions" button
  - Should be pressable when all fields are valid
  - Should send instructions to the specified email
- "X" header button should go back to **Login Screen**

# Forgot Password




A mobile app screen titled "Forgot Password" with a close button (X) in the top left. Below the title is a green envelope icon. The main heading is "Enter password reset code". Below this, a message states: "We sent a password reset code to [juandelacruz@tedhouse.org](mailto:juandelacruz@tedhouse.org)". There is a text input field labeled "Code" with a light blue border. At the bottom is a large green button labeled "Next".

## Code Screen

- Screen that asks for the code that is sent on the email.
- "Code" text field
  - o Numeric 4 digit
- "Next" button
  - o Should be pressable when all fields are valid
- "X" header button should go back to **Login Screen**

# Forgot Password

× Forgot Password



**Set new password**

Password

- ☐ Between 8 to 30 characters
- ☐ Accepts the following characters:
  - Alphanumeric (A-Z, a-z, 0-9)
  - Special Characters ~!@#\$%^&\*()\_+={}|;:,<.>?

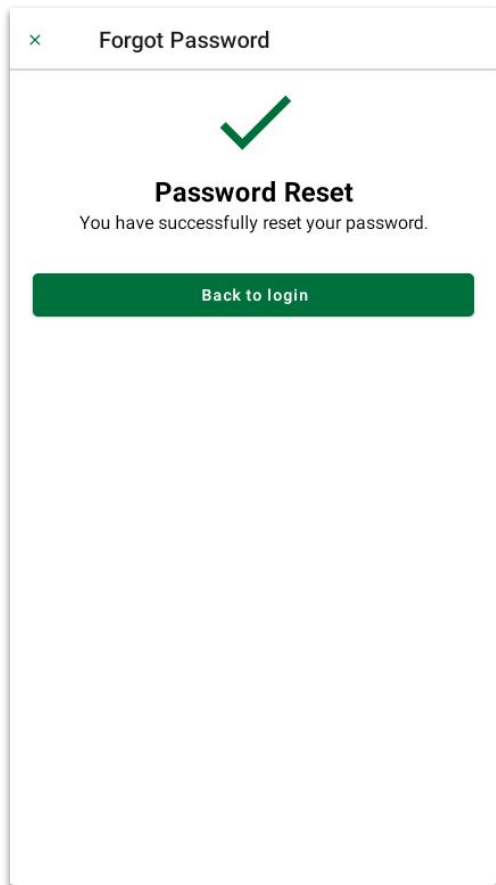
Confirm Password

Reset password

## Set New Password Screen

- Screen that asks for a new password.
- “Password” text field
  - When filled up
    - hide password -
    - show password -
  - Should be between 8 to 30 characters
  - Accepts the following characters:
    - Alphanumeric (A-Z, a-z, 0-9)
    - Special Characters ~!@#\$%^&\*()\_+={}|;:,<.>?
- “Password” checklist
  - Should be checked when password criteria is matched
- “Confirm Password” text field
  - Should be matched “Password”. Otherwise, error will show below the text field.
- “Reset password” button
  - Should be pressable when all fields are valid
  - Should send request for reset password
    - Proceeds to **Reset Password Screen** if successful. Otherwise, shows an error message

# Forgot Password



## Set New Password Screen

- Screen that informs the passenger that their password has been reset successfully.
- "X" header and "Back to Login" button should go back to **Login Screen**

# Bottom Navigation



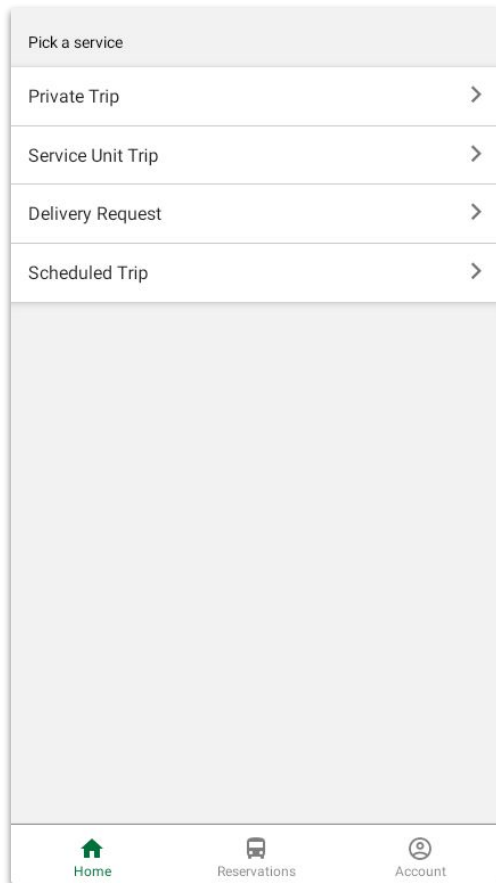
## Bottom Navigation Buttons

- Buttons that lets the passenger navigate to different tabs.
- If selected, button is highlighted.

# Service Options Screens



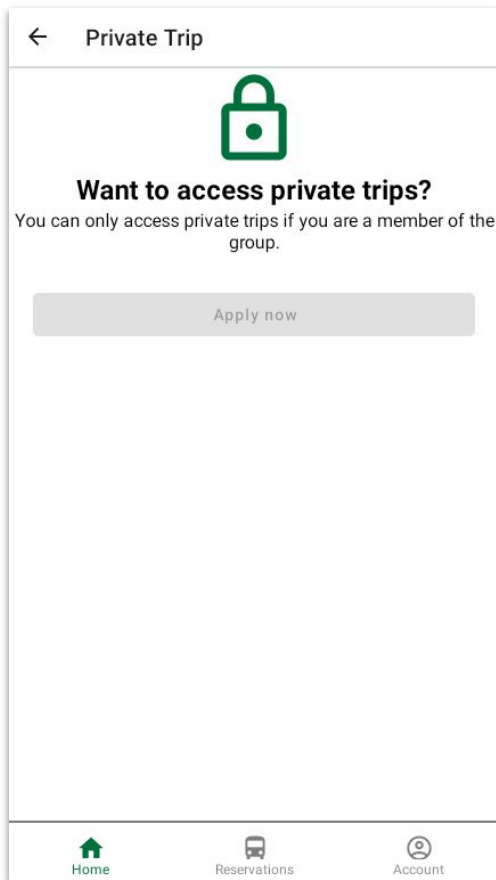
# Service Options



## Service Options Screen

- Screen that shows the services options for passenger.
- Services:
  - o **Private Trip** - Urgent vehicle trips are trips that require immediate attention or action. Not shared.
  - o **Service Unit Trip** - Trips that are shared or accessible by multiple passengers.
  - o **Delivery Request** - FLEV picks up a package at point A and delivers it to point B."
  - o **Scheduled Trip** - Scheduled trips are trips that are planned in advance and have a specific time and place for pick-up and drop-off. These trips are shared with other passengers, but unlike **Service Unit Trip**, the passengers are required to be present at the designated time and place.

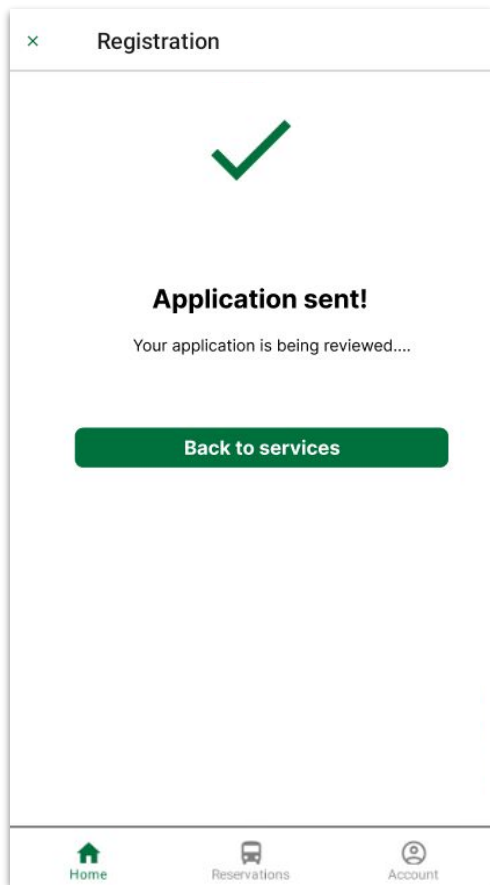
# Private Trips



## Restricted Service Screen

- The private trips access **Restricted Service Screen** is displayed when the passenger does not have permission to access private trips
- To gain access, passengers must apply for the service.
- **"Apply now"** button
  - o Should send a request to the server.
  - o Disabled if the passenger is restricted to private trips.
  - o Proceeds to **Send Application Screen** if successful. Otherwise, shows an error.

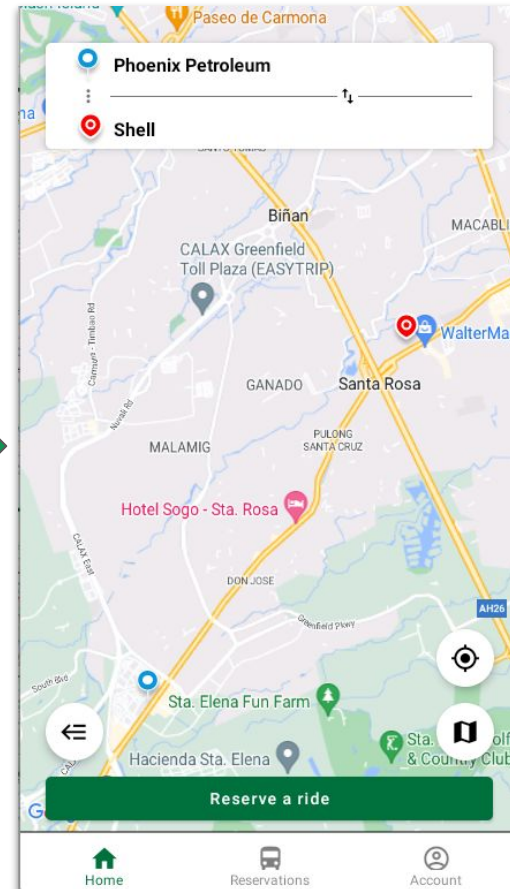
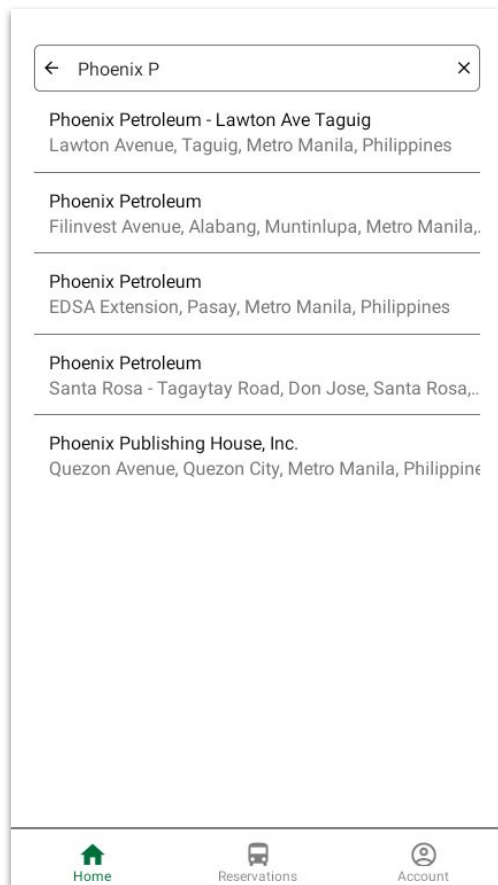
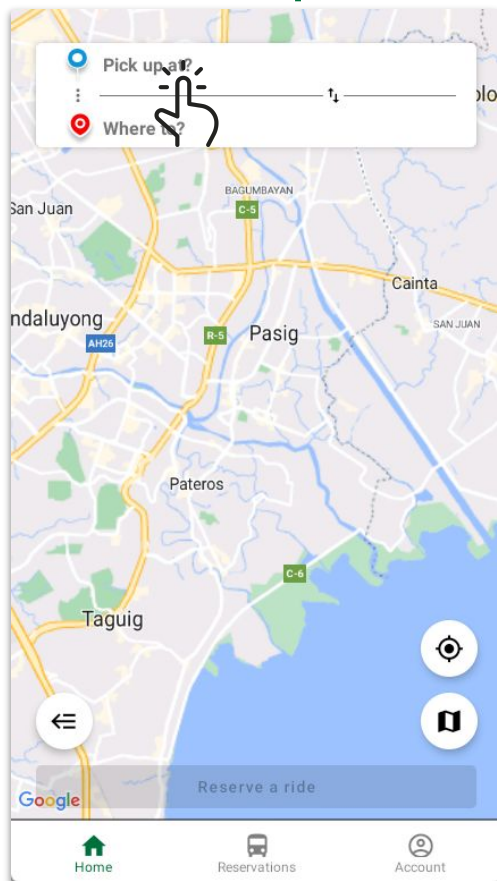
# Private Trip



## Send Notification Screen

- Screen that notifies the passenger that the application is sent to the server.
- "X" header and "Back to services" button should go back to **Services Options Screen**

# Private Trips Flow



# Map Search Buttons



- Back to service options

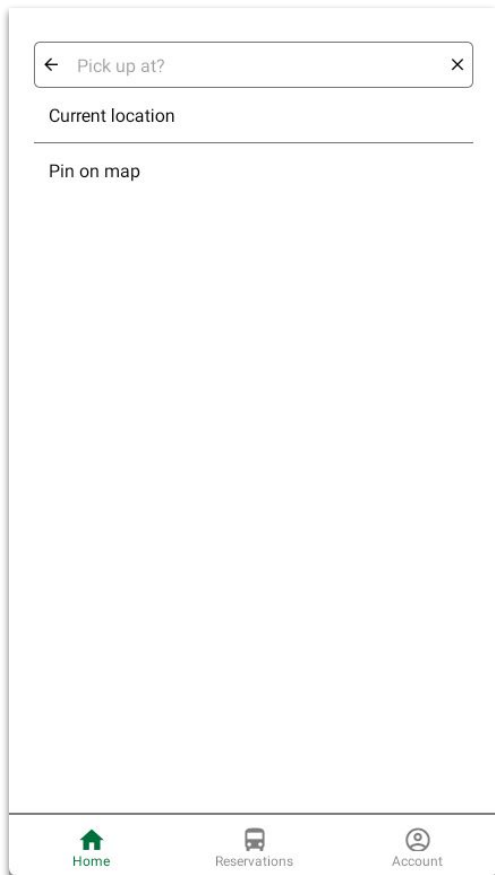


- Go to user location



- Display origin or destination or origin and destination on the center of the map

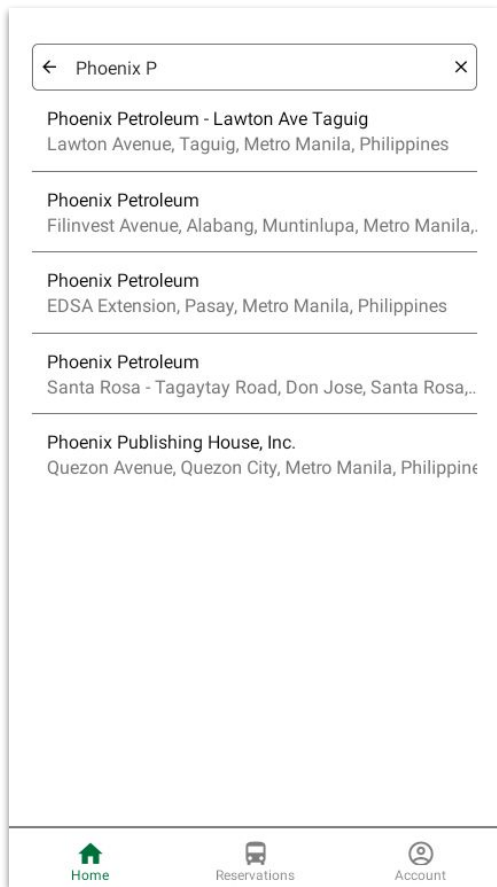
# Search Location



## Search Location Index Screen

- The screen allows the passenger to search for or select their location and set it as the origin or destination.
- Current Location
  - Passenger picks their location.
- Pin on map
  - Passenger pinpoint their location in the map.

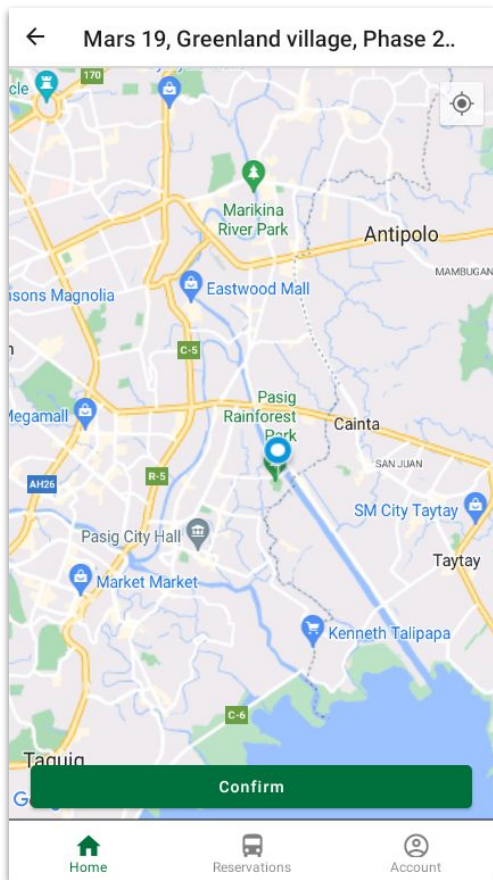
# Search Location



## Search Location Index Screen

- The screen allows the passenger to search for or select their location and set it as the origin or destination.
- **Google Places Autocomplete** text field
  - Passenger searches for their location using Google Places Autocomplete, and then selects their location from the list provided by Google.
  - Should easily be cleared by pressing “x” when filled up

# Search Location

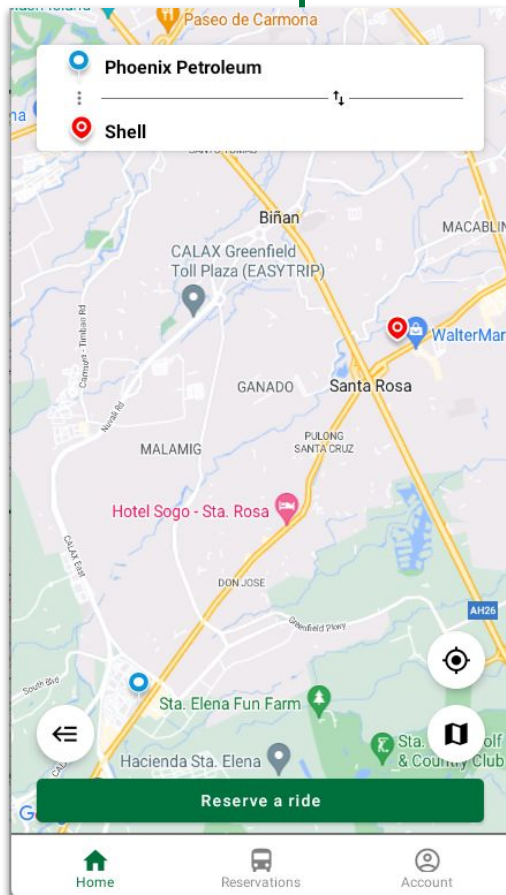


## Pin Map Screen

- The screen allows the passenger pinpoint their location and set it as the origin or destination.
- **Confirm** button
  - o Sets the location where the pin is located.
- **Target** button
  - o Located at the top right corner of the map.
  - o Locates and moves to the location of the passenger.



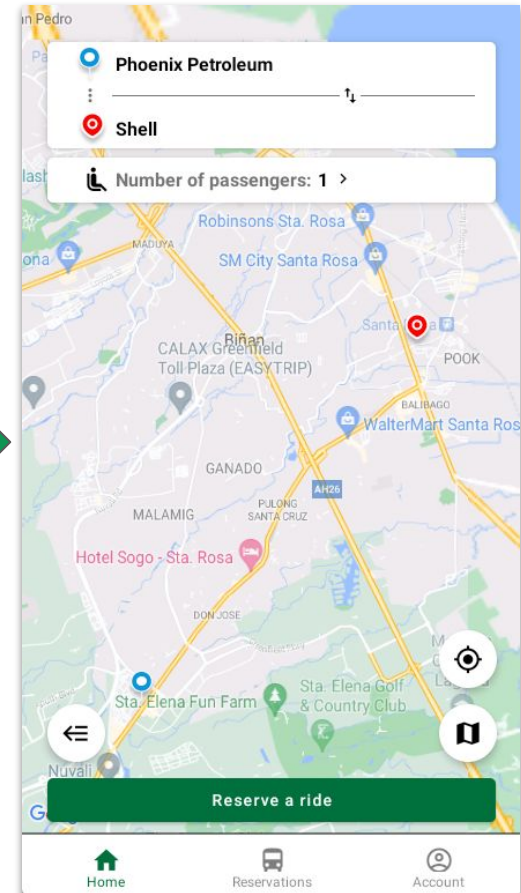
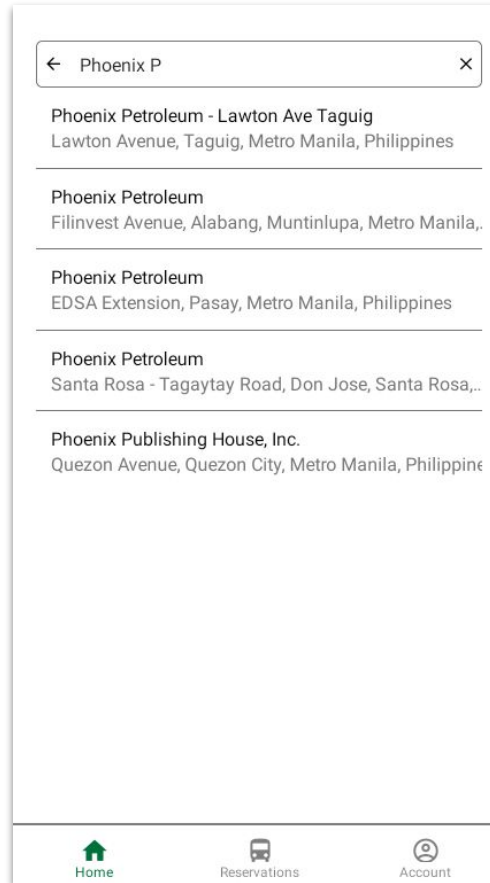
# Private Trip



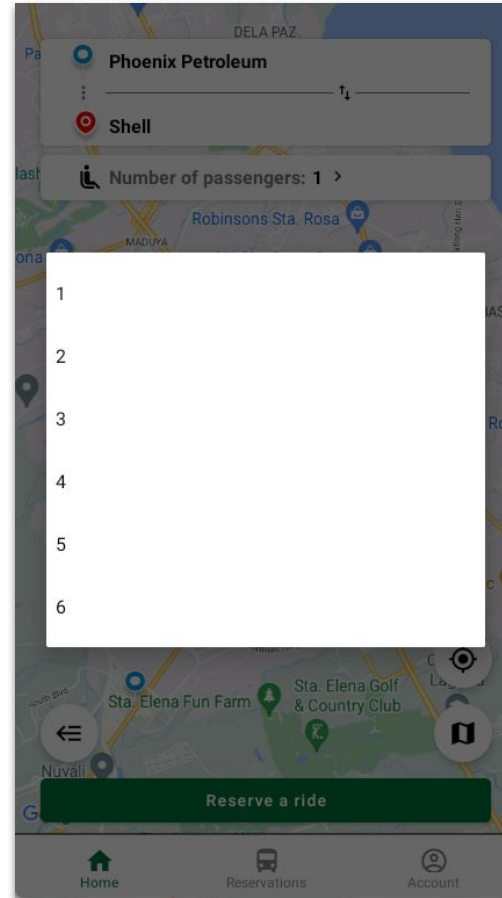
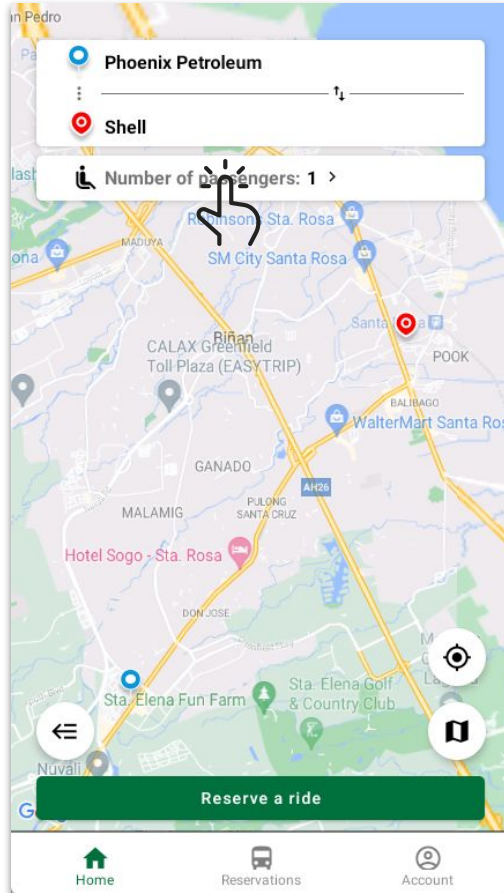
## Private Trip Screen

- Screen where the passenger searches trips for Private Trips.
- Reserve a ride
  - When origin (Pick up at?) and destination (Where to?) is filled up, the button should be pressable
  - If successful
    - It should go to the Reservations tab and the passenger should see their trip request.
    - The passenger awaits for the trip request confirmation popup or notification and should see the status of the request, whether it was successful (Driver accepts the request) or failed (Failed to find a trip).
  - If failed (Request failed or didn't send)
    - It should show a popup dialog containing the error message.
  - Uses createTripRequest mutation call to create trip request.
    - Passes origin and destination details along with the trip type PRIVATE.

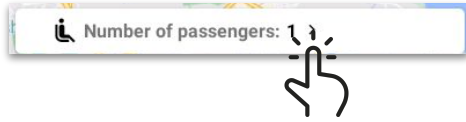
# Service Unit Trips Flow



# Service Unit Trips Flow



# Number of passengers

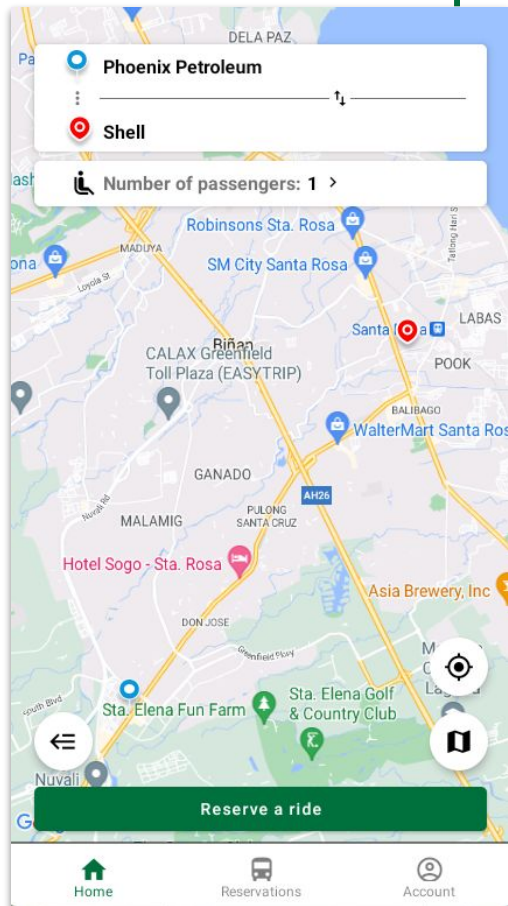


## Number of passengers card

- If pressed, it will show a numbered list.
- Passenger selects the number of passengers they want to book for the reservation.



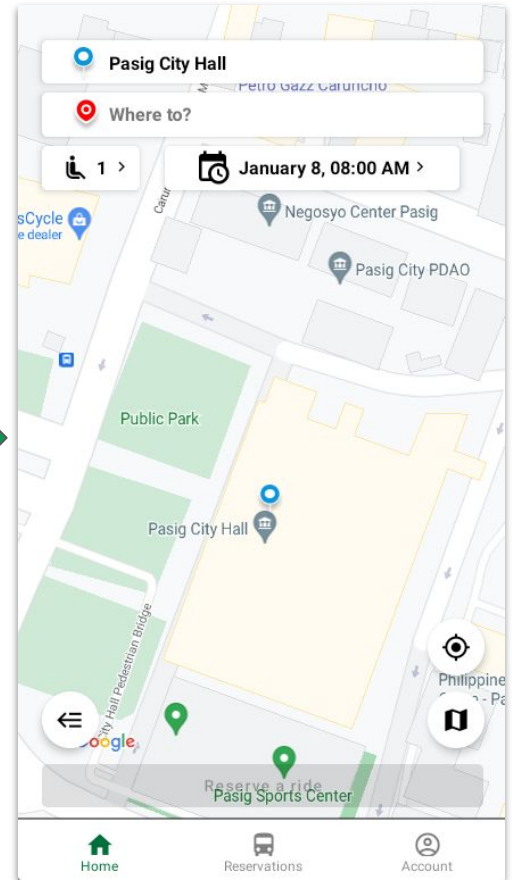
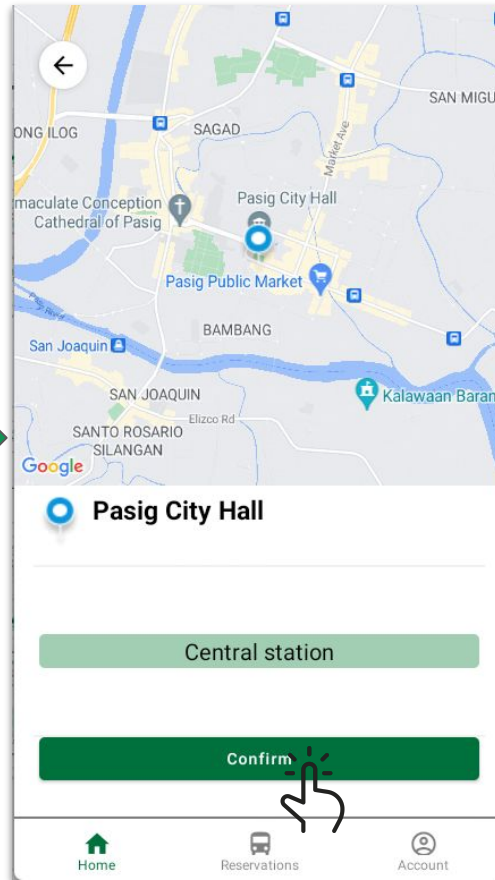
# Service Unit Trip



## Service Unit Trip Screen

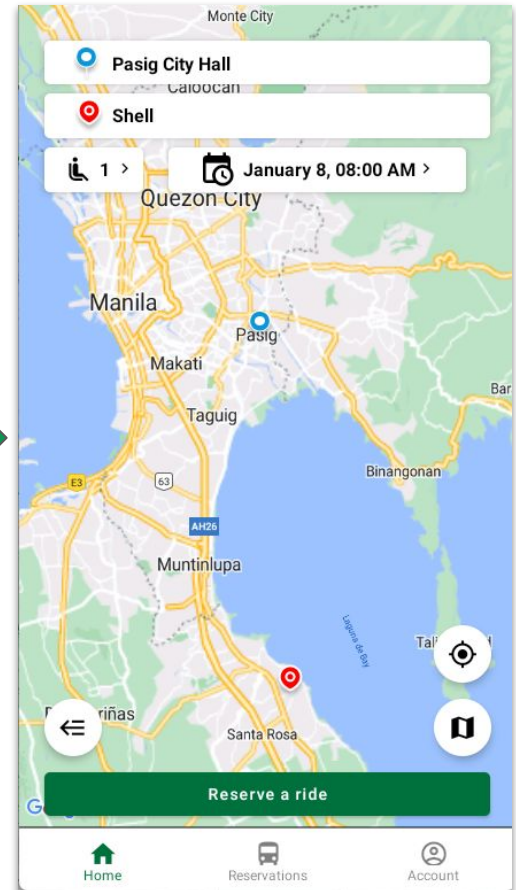
- Screen where the passenger searches trips for Service Unit Trips.
- Reserve a ride
  - o When origin (Pick up at?), destination (Where to?), and number of passengers is filled up, the button should be pressable
  - o If successful
    - It should go to the Reservations tab and the passenger should see their trip request.
    - The passenger awaits for the trip request confirmation popup or notification and should see the status of the request, whether it was (Driver accepts the request) or failed (Failed to find a trip).
  - o If failed (Request failed or didn't send)
    - It should show a popup dialog containing the error message.
  - o Uses createTripRequest mutation call to create trip request.
    - Passes origin and destination details along with the trip type SERVICE\_UNIT and number of passengers.

# Scheduled Trip Flow

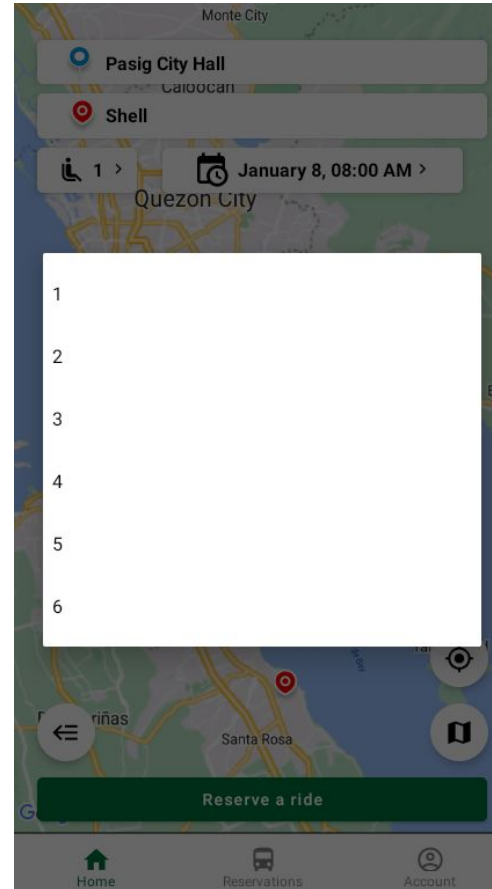
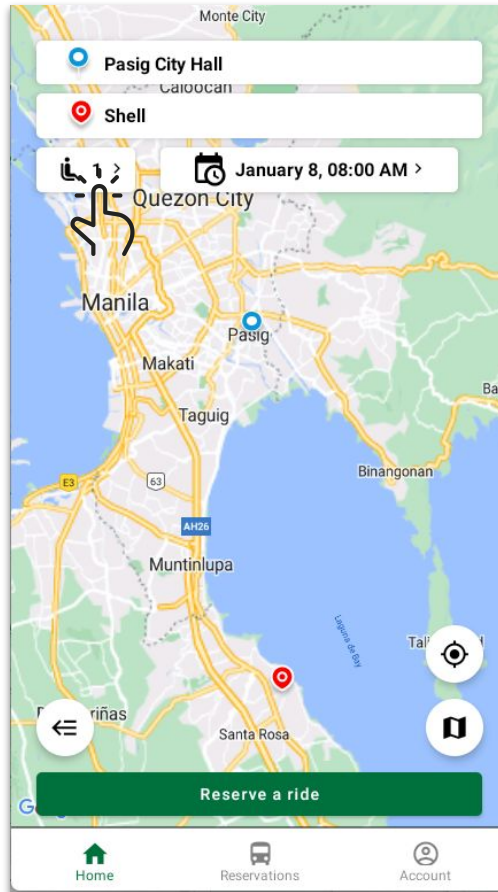




# Scheduled Trip Flow

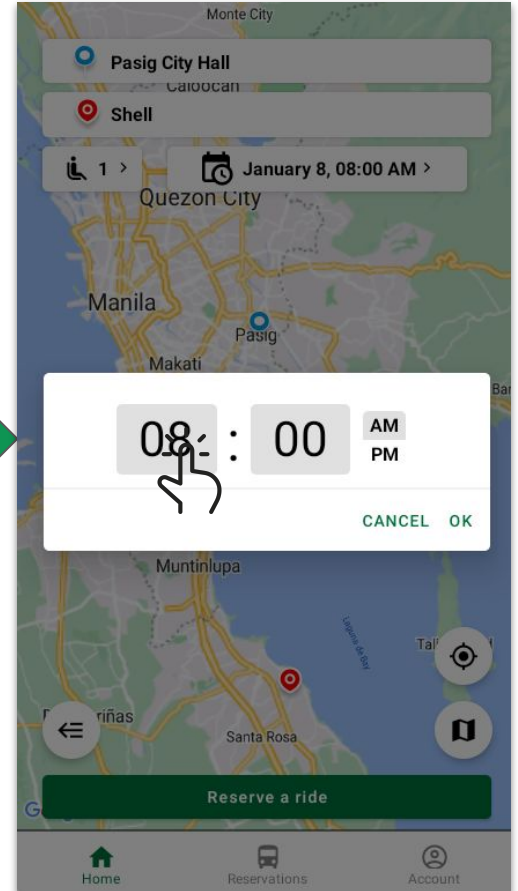
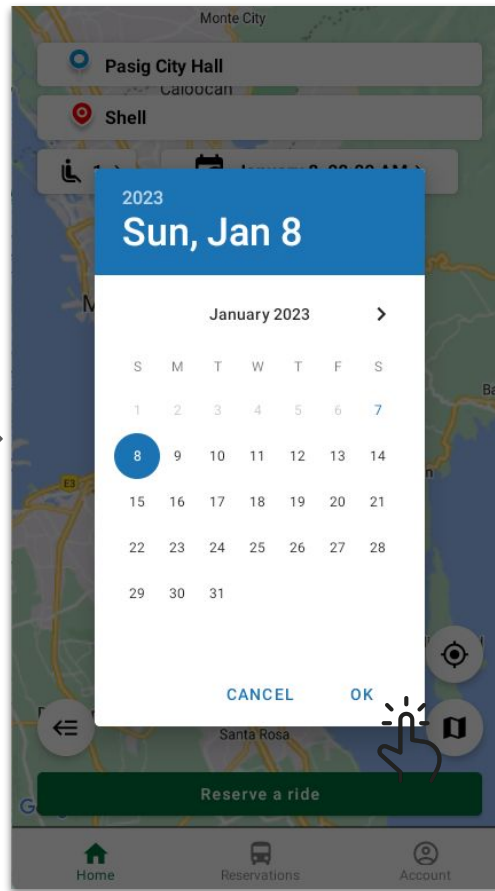
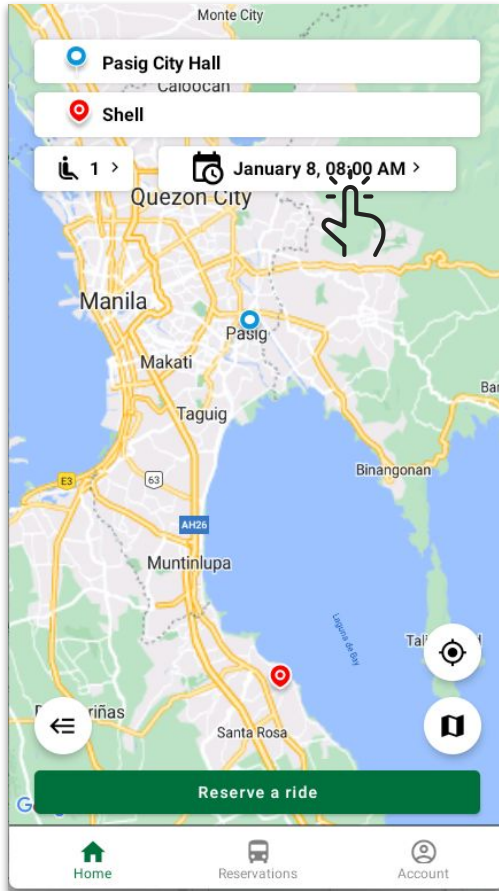


# Scheduled Trip Flow

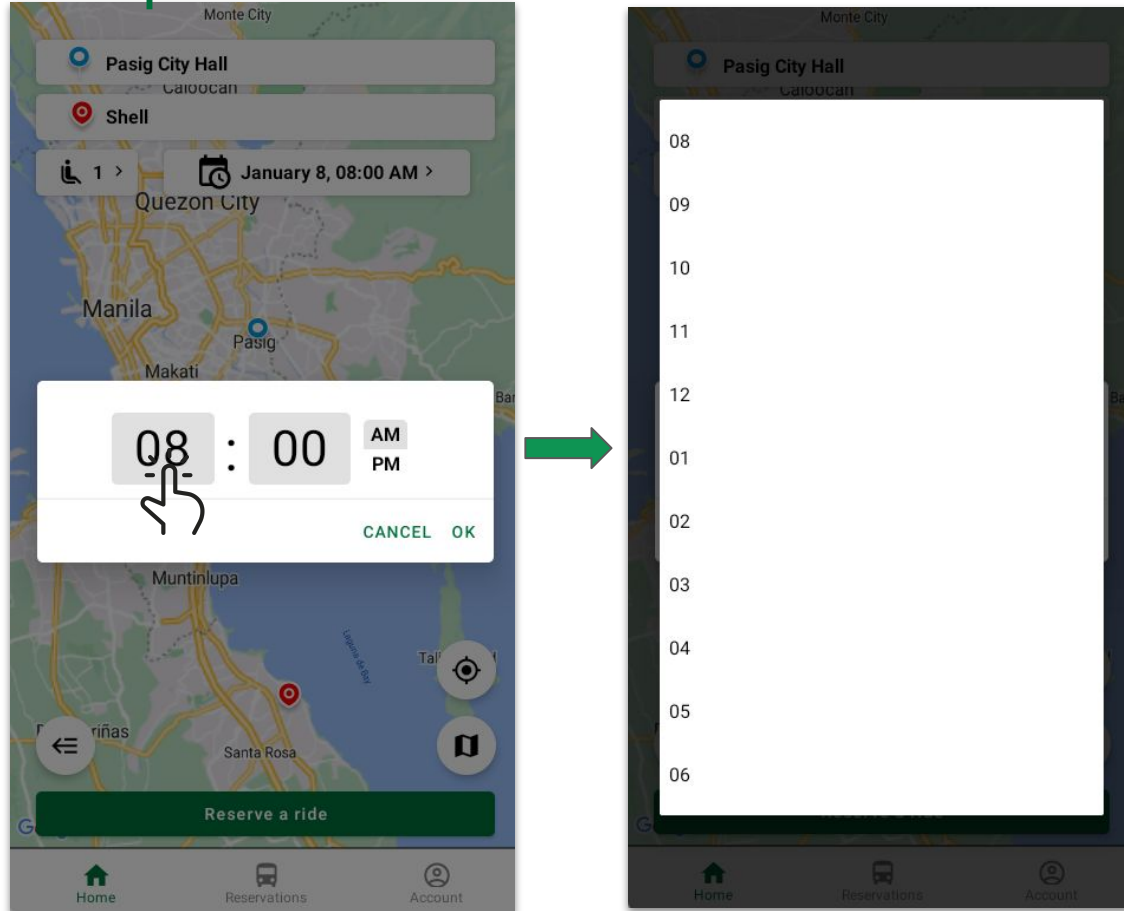




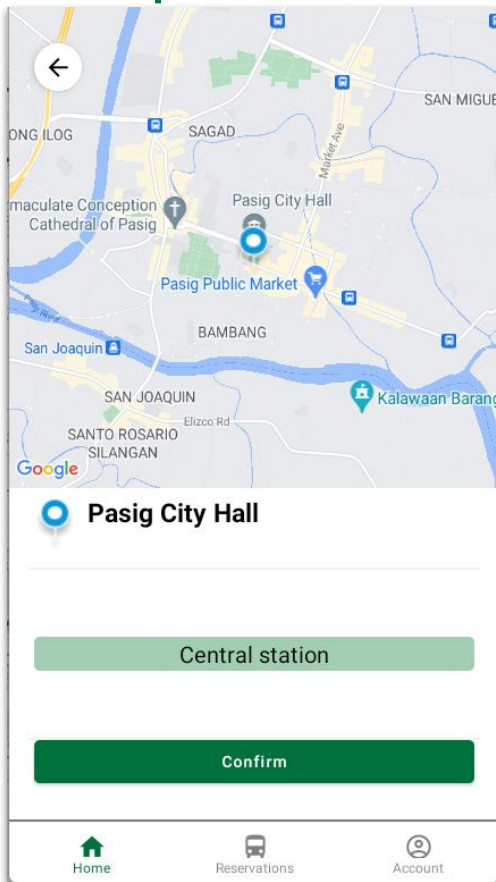
# Scheduled Trip Flow



# Scheduled Trip Flow



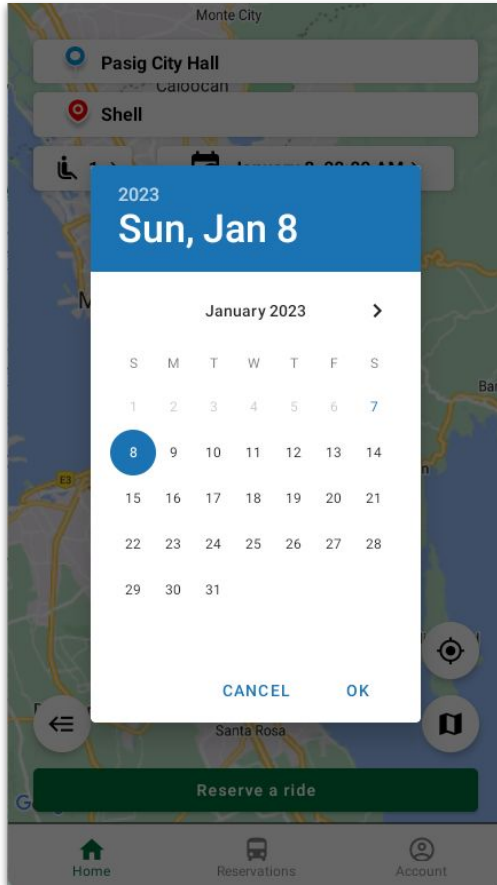
# Pickup Location



## Pickup Locations Screen

- Screen where the passenger selects from a list of pickup options.
- Pressing the "Confirm" button sets the highlighted pickup location (in this case, Central station) as the origin.

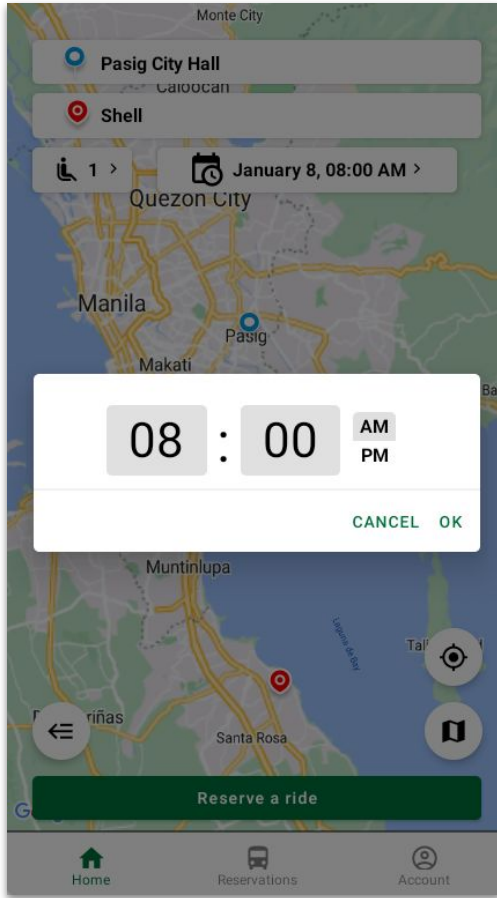
# Date time



## Date Picker

- If pressed, a calendar with selectable dates from tomorrow to 30 days after tomorrow will be displayed.
- Pressing “OK” will display the **Time Picker**.
- Pressing “CANCEL” will close the **Date Picker**.

# Date time



## Time Picker

- Shown after the date is set.
- Pressing “OK” will set the date time.
- Pressing “CANCEL” will close the **Time Picker**.

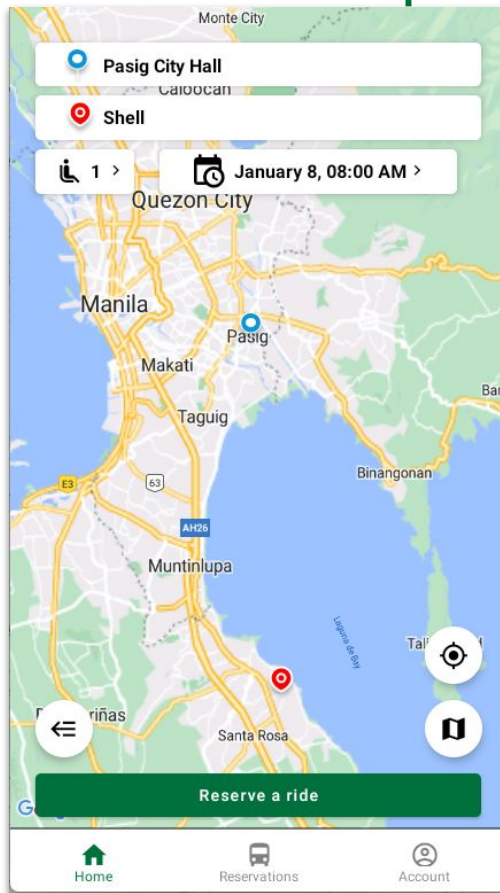
# Date time



## Time Picker

- Shown after the date is set.
- Pressing "OK" will set the date time.
- Pressing "CANCEL" will close the **Time Picker**.

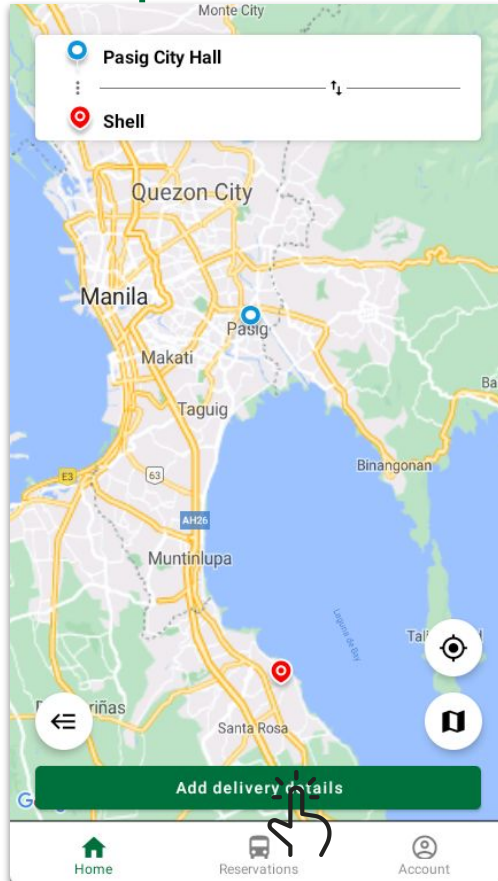
# Scheduled Trip



## Scheduled Trip Screen

- Screen where the passenger searches trips for Scheduled Trips.
- Reserve a ride
  - When origin (Pick up at?), destination (Where to?), number of passengers, and date time is filled up, the button should be pressable.
  - If successful
    - It should go to the Reservations tab and the passenger should see their scheduled trip request.
    - The passenger awaits for the trip and should see the status of the request, whether it is ongoing or not.
  - If failed (Request failed or didn't send)
    - It should show a popup dialog containing the error message.
  - Uses createScheduledTripRequest mutation call to create scheduled trip request.
    - Passes origin, destination, selected date, and number of passengers.

# Delivery Request Flow



## Delivery Details

**Sender**

Name: Juan Dela Cruz  
Phone or Tel. Number:  
Address: Pasig City Hall

**Destination**

Name:  
Phone or Tel. Number:  
Address: Shell

**Item details**

Type of item: Document  
Total weight (kg): 10  
Item Dimensions (w x l x h - in): 10 x 10 x 10

**Delivery schedule**

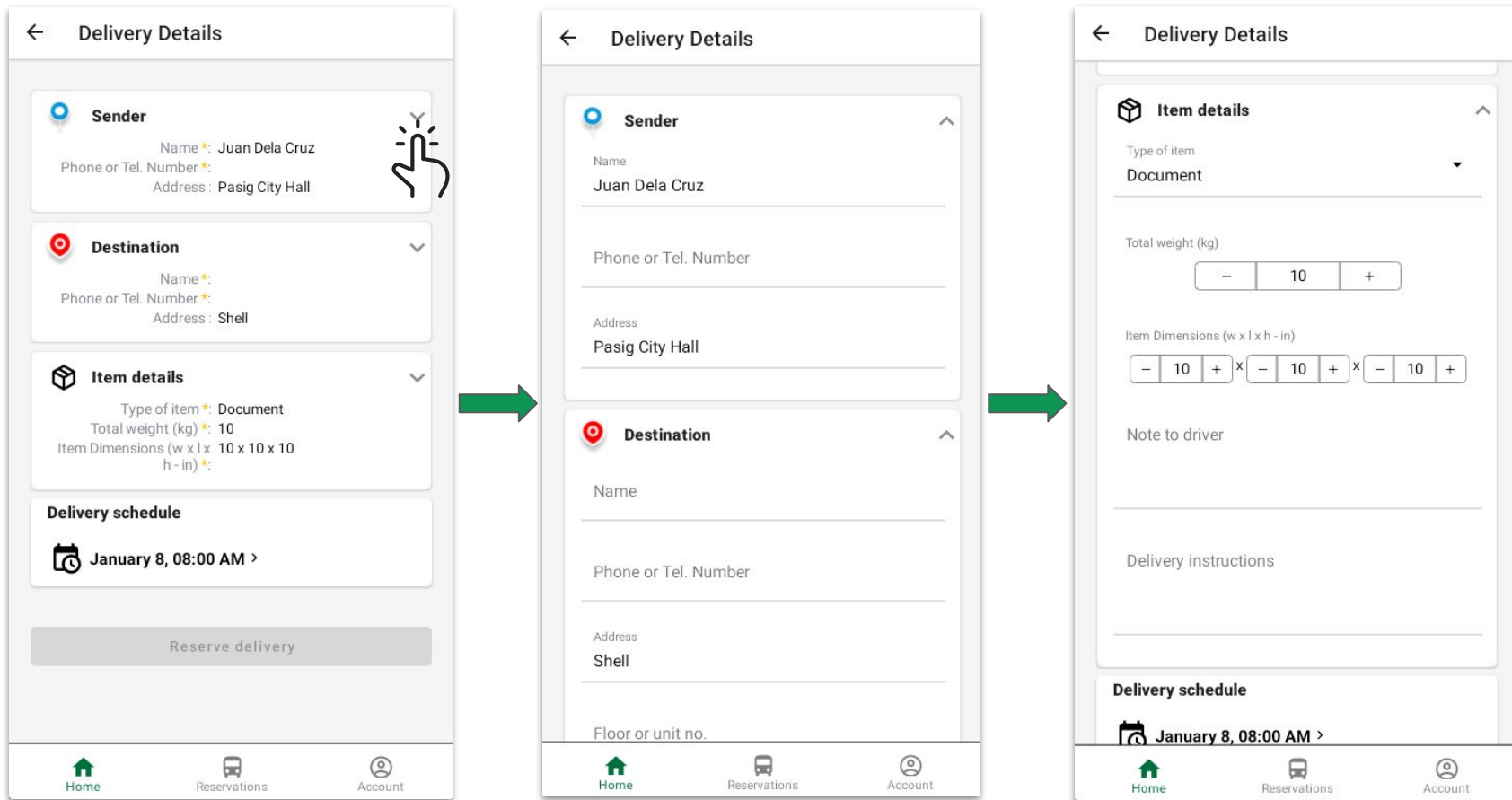
January 8, 08:00 AM >

Reserve delivery

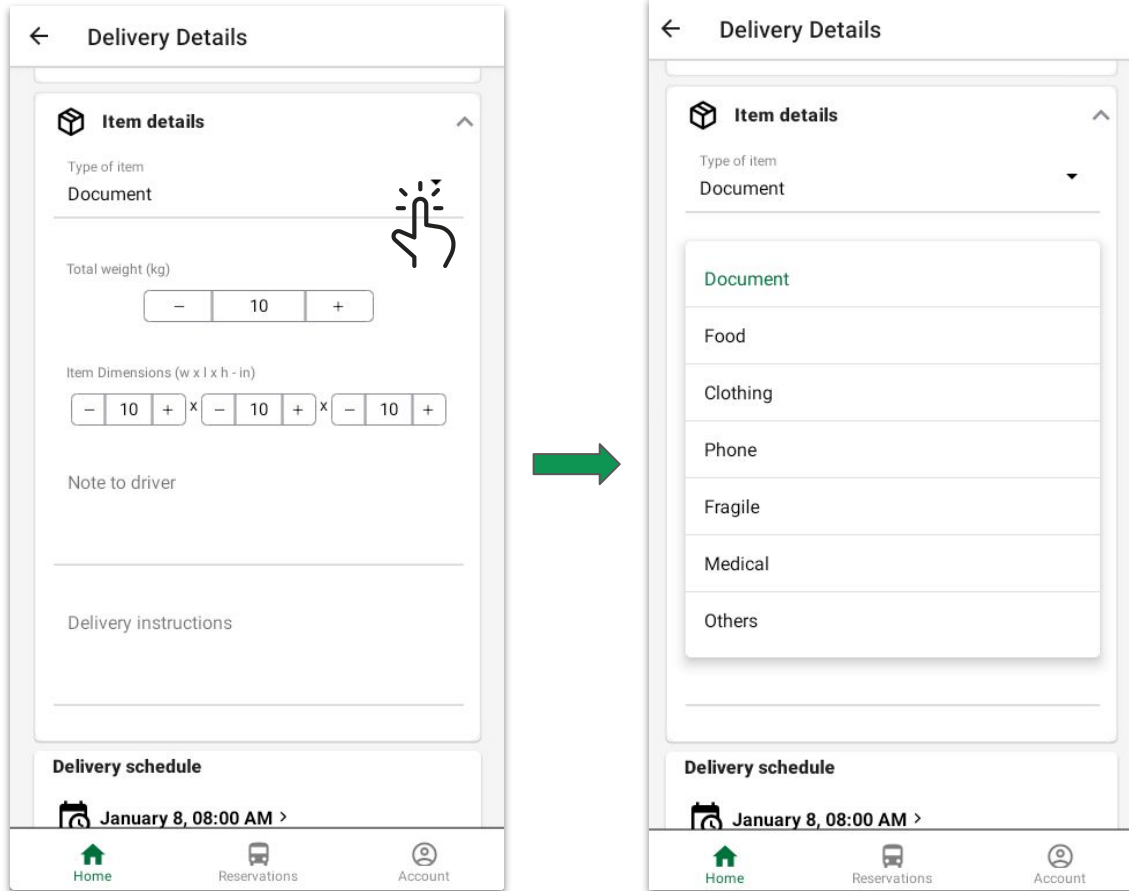
Home Reservations Account



# Delivery Request Flow



# Delivery Request Flow



# Delivery Request Flow

←

Delivery Details

Sender

▼

Name\*: Juan Dela Cruz  
Phone or Tel. Number\*: +639123123121  
Address: Pasig City Hall

Destination

▼

Name\*: Maria Clara  
Phone or Tel. Number\*: +639253888911  
Address: Shell

Item details

▼

Type of item\*: Document  
Total weight (kg)\*: 10  
Item Dimensions (w x l x h - in)\*: 10 x 10 x 10

Delivery schedule

January 8, 08:00 AM >

Reserve delivery

Home

Reservations

Account



←

Delivery Details

Sender

▼

Name\*: Juan Dela Cruz  
Phone or Tel. Number\*: +639123123121  
Address: Pasig City Hall

Destination

▼

Name\*: Maria Clara  
Phone or Tel. Number\*: +639253888911  
Address: Shell

Item details

▼

Type of item\*: Document  
Total weight (kg)\*: 10  
Item Dimensions (w x l x h - in)\*: 10 x 10 x 10

Delivery schedule

January 8, 08:00 AM >

Reserve delivery

Home

Reservations

Account

2023

Mon, Jan 9

January 2023 >

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

CANCEL OK

# Accordion

The screenshot shows a mobile application interface for "Delivery Details". At the top is a back arrow and the title "Delivery Details". Below this is an accordion component titled "Item details" with a cube icon and an upward arrow. The accordion is currently expanded, revealing several input fields: "Type of item" with a dropdown menu showing "Document", "Total weight (kg)" with a numeric keypad showing "10", "Item Dimensions (w x l x h - in)" with three separate numeric keypads each showing "10", "Note to driver" with a text input field, and "Delivery instructions" with a text input field. At the bottom of the screen is a "Delivery schedule" section showing a calendar icon, the date "January 8, 08:00 AM", and a right-pointing chevron. The very bottom of the screen features a navigation bar with three icons: a house for "Home", a bus for "Reservations", and a person for "Account".

## Accordion

- A list of items stacked vertically, in this case a list of fillable details.
- Each item can be expanded or collapsed to show or hide the content associated with it.
- Helps present content in a compact format, allowing passengers to see a summary of their filled-out details.

# Delivery Item Details

Type of item  
Document ▼

Document

Food

Clothing

Phone

Fragile

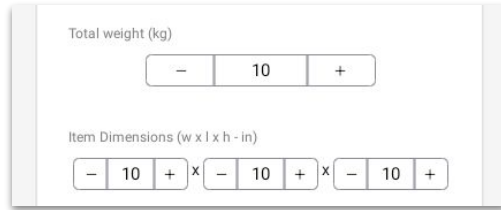
Medical

Others

## Type of item

- Determines the type of item to be delivered
- Should include the following: Document, Food, Clothing, Phone, Fragile, Medical, Others

# Delivery Item Details



The form is titled "Delivery Item Details" and contains two input sections. The first section is labeled "Total weight (kg)" and features a single input field with a minus sign, the number "10", and a plus sign. The second section is labeled "Item Dimensions (w x l x h - in)" and features three identical input fields, each with a minus sign, the number "10", and a plus sign, separated by "x" characters.

Total weight (kg)

- 10 +

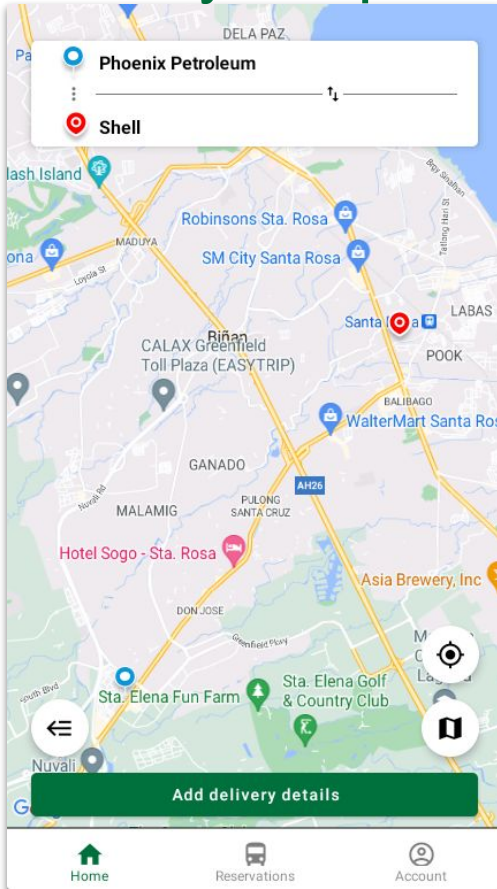
Item Dimensions (w x l x h - in)

- 10 + x - 10 + x - 10 +

## Item weight and dimensions

- Passenger estimates the weight and dimensions of the item. This is used to arrange the item in the Electric Vehicle.
- Ideally, this should be a list of selectable items.

# Delivery Request



## Delivery Request Screen

- Screen where the passenger searches trips for Delivery Requests.
- Add delivery details
  - o When origin (Pick up at?) and destination (Where to?) the button should be pressable.
  - o When pressed, it should go to the **Delivery Details Screen**.

# Delivery Request

← Delivery Details

**Sender** ▼

Name\*: Juan Dela Cruz  
Phone or Tel. Number\*: +639123123121  
Address: Pasig City Hall

**Destination** ▼

Name\*: Maria Clara  
Phone or Tel. Number\*: +63925388911  
Address: Shell

**Item details** ▼

Type of item\*: Document  
Total weight (kg)\*: 10  
Item Dimensions (w x l x h - in)\*: 10 x 10 x 10

**Delivery schedule**

📅 January 8, 08:00 AM >

**Reserve delivery**

Home Reservations Account

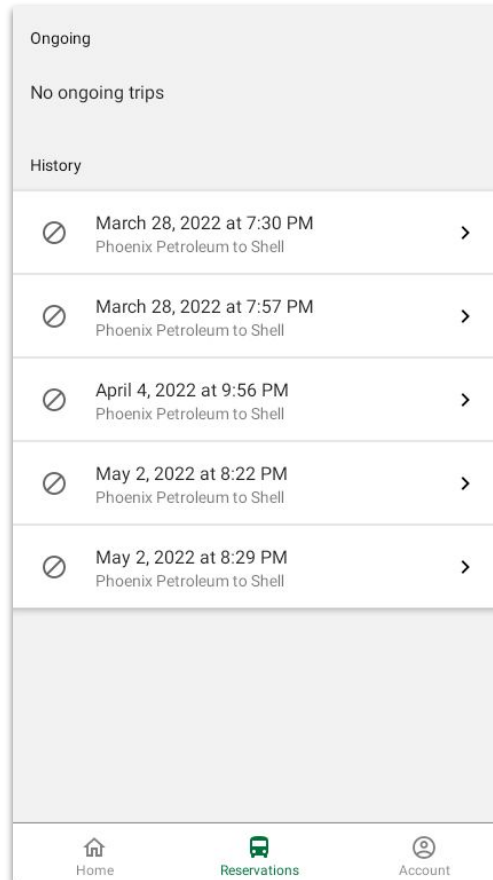
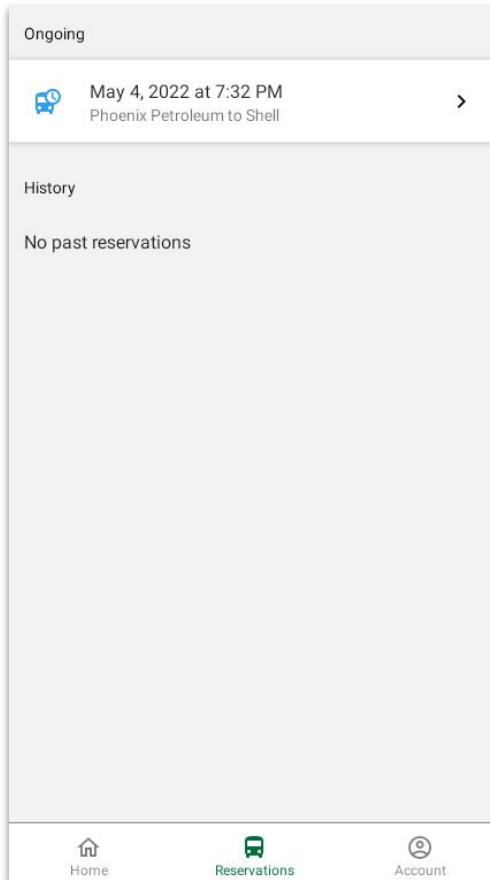
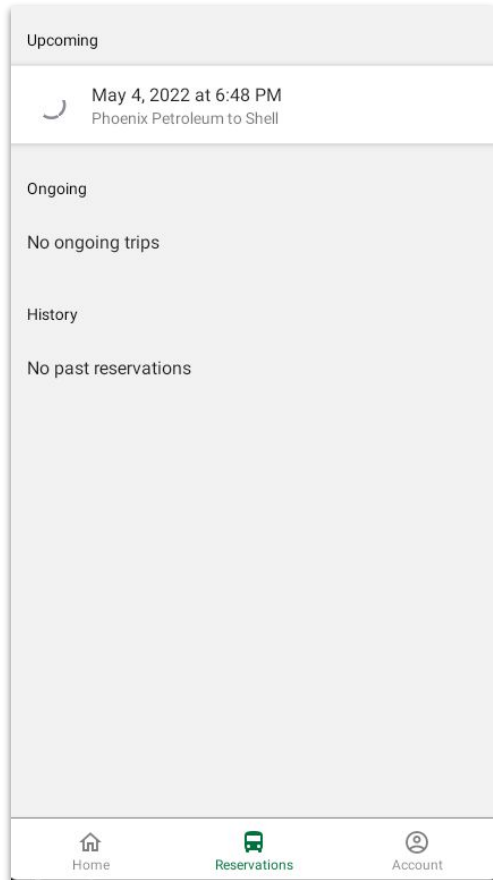
## Delivery Details Screen

- Screen where the passenger fills up the details for delivery request.
- Details with an asterisk (\*) are required fields. Passengers cannot reserve a delivery if these are not filled out.
- Reserve delivery
  - Should be pressable when \* fields are filled out.
  - If successful
    - It should go to the Reservations tab and the passenger should see their delivery request.
    - The passenger awaits for the delivery request confirmation popup or notification and should see the status of the request, whether it was (Driver accepts the request) or failed (Failed to find a trip).
  - If failed (Request failed or didn't send)
    - It should show a popup dialog containing the error message.
  - Uses createTripRequest mutation call to create trip request.
    - Passes the delivery details along with the trip type DELIVERY\_REQUEST.

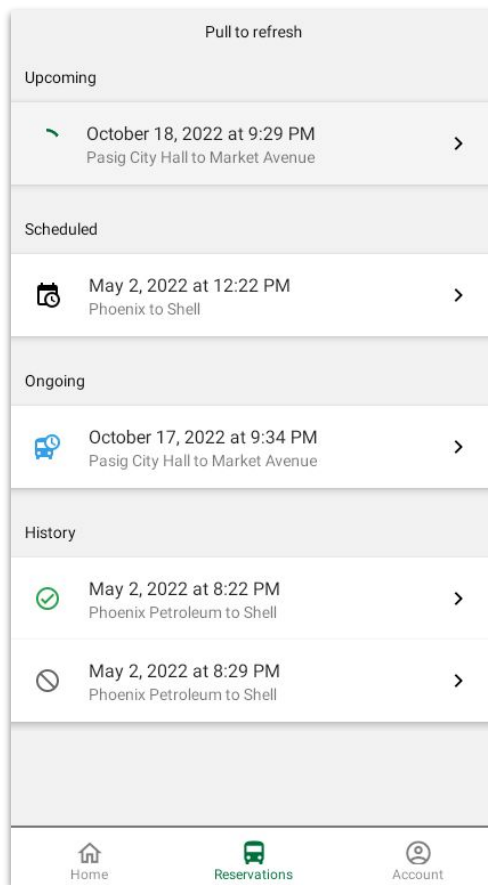


# Reservation Screens

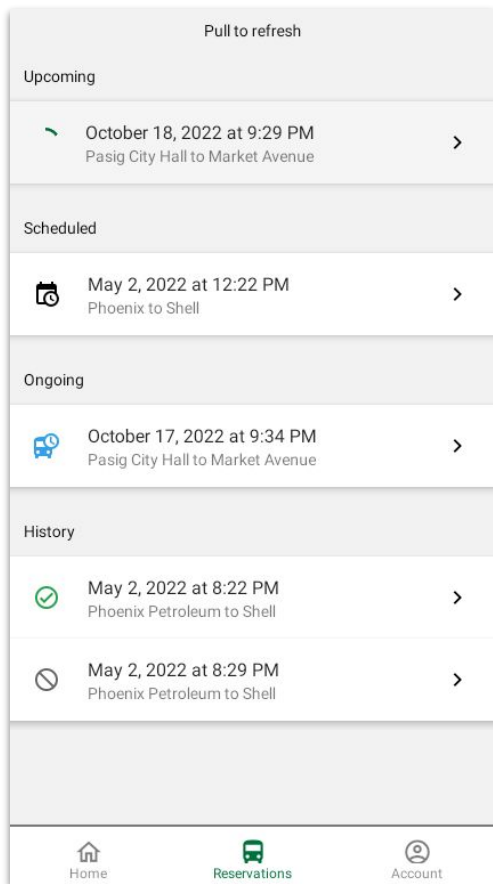
# Reservations



# Reservations



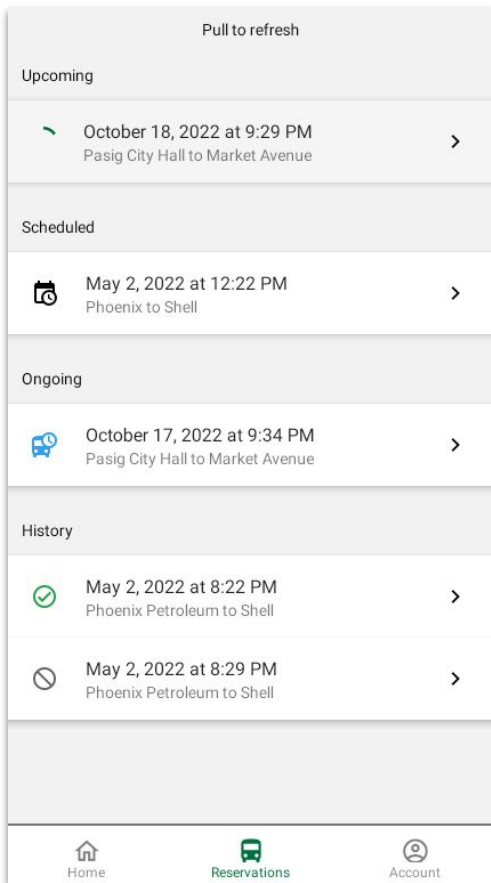
# Reservations



## Reservation List Screen

- Screen where the passenger sees their **Upcoming, Ongoing, Scheduled, and Past reservations** sections.
- Uses tripRequest and reservation query call to get the trip request and reservation data.
- The data retrieved is then arranged into upcoming, ongoing, scheduled, and past reservations based on their status.
- Trip request are requests that have not yet been accepted.
- Scheduled trip request indicates that the trip has not yet been accepted or has not yet started.
- Reservation are accepted requests.

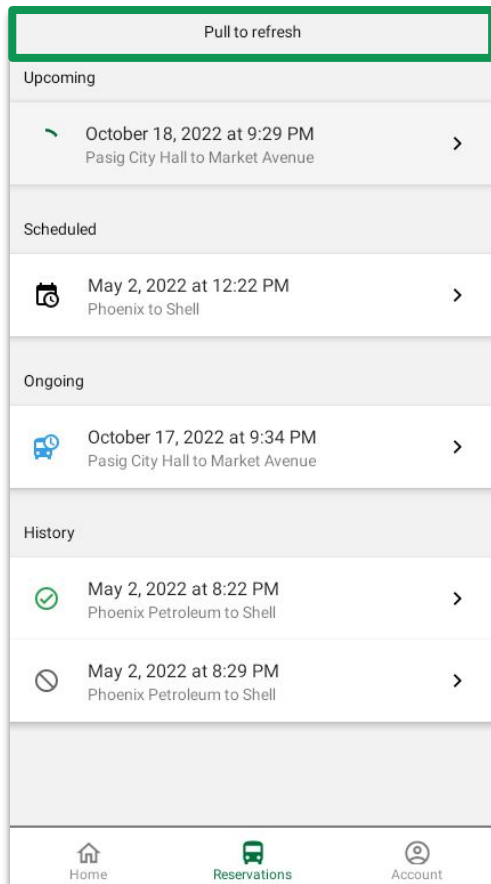
# Reservations



## Reservation List Screen

- **Reservation List Screen** also “subscribes” for updates in trip request.
  - It uses `onTripRequestUpdate` to receive live updates from the server.
  - If there is an update
    - It re-queries the trip requests and reservations using the `tripRequest` and `reservation` queries

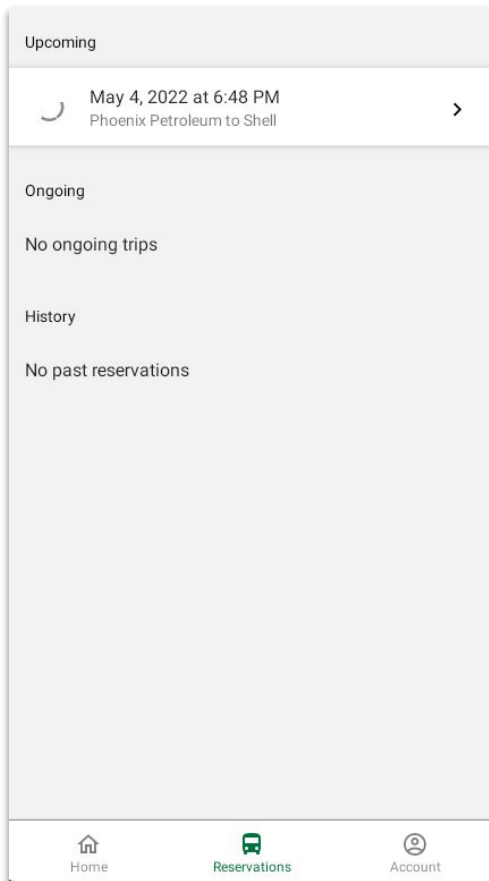
# Reservations



## Reservation List Screen

- If the updates didn't reflect, the passenger can "Pull to Refresh" to re-query.

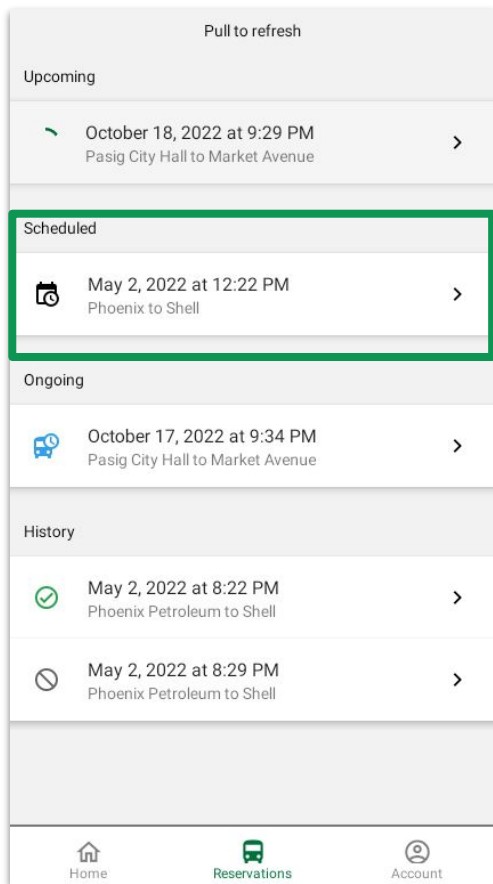
# Reservations



## Reservation List Screen

- If the trip request status is **CREATED**, **PENDING**, or **SEARCHING**, it will be displayed in **Upcoming** section.

# Reservations

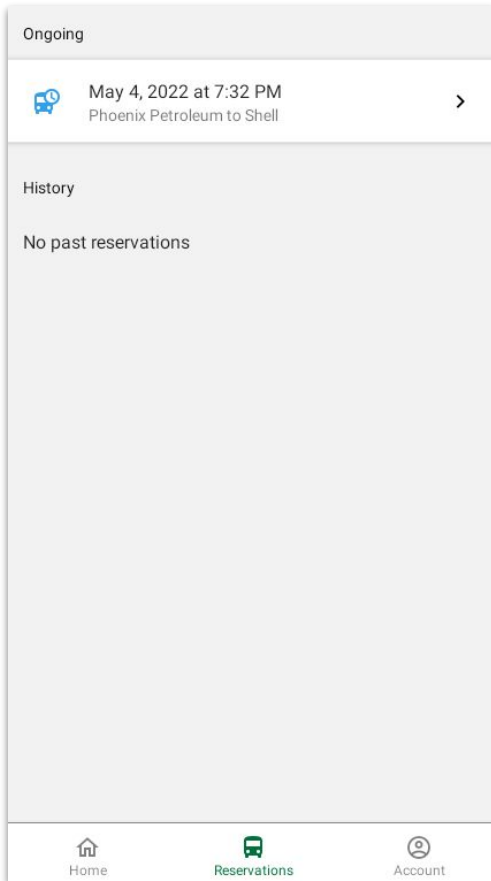


## Reservation List Screen



- Scheduled trip request are in the **Scheduled** section.



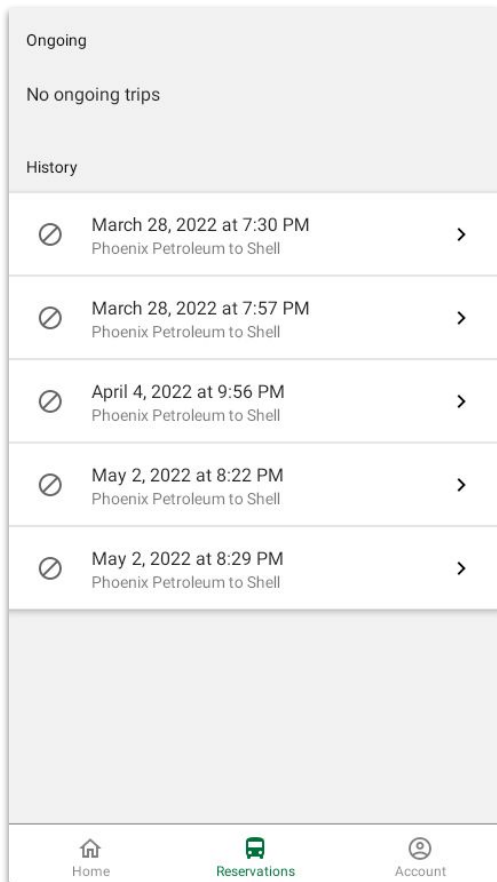
# Reservations





## Reservation List Screen

- If the reservation status is **BOARDED** and **CONFIRMED**, it will be displayed in **Ongoing**.
- **CONFIRMED**
  - Icon - 
  - Driver has not yet picked up the passenger.
- **BOARDED**
  - Icon - 
  - The driver has picked up the passenger and is en route to the passengers' destination.

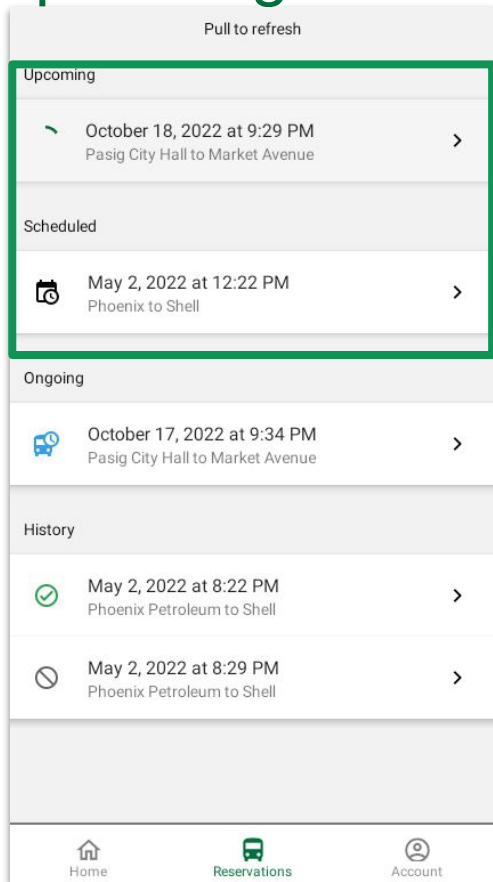
# Reservations



## Reservation List Screen

- If the reservation status is **CANCELED** or the trip status **COMPLETED**, it will be displayed in **History**.
- **COMPLETED**
  - o Icon - 
  - o Passenger trip is completed.
- **CANCELED**
  - o Icon - 
  - o Passenger cancels the **CONFIRMED** reservation.

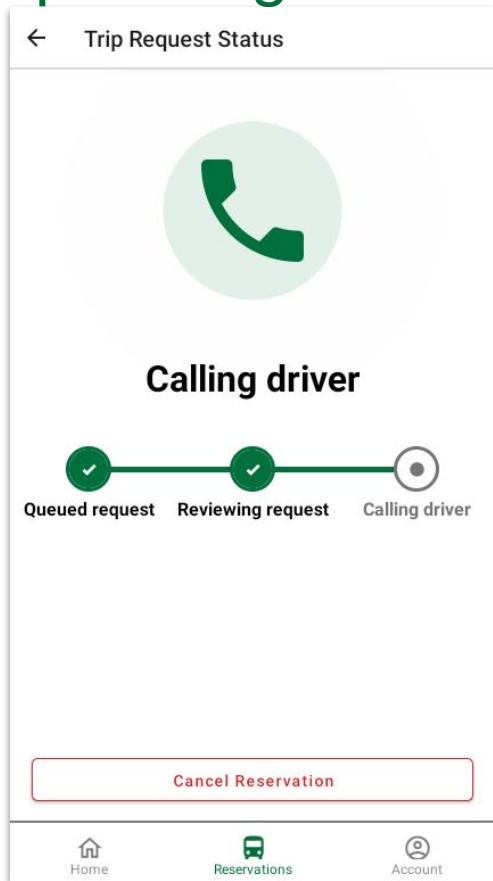
# Upcoming and Scheduled



## Reservation List Screen

- Pressing the **Upcoming** and **Scheduled** reservations will navigate to the **Trip Request Progress Screen** along with the trip request details.

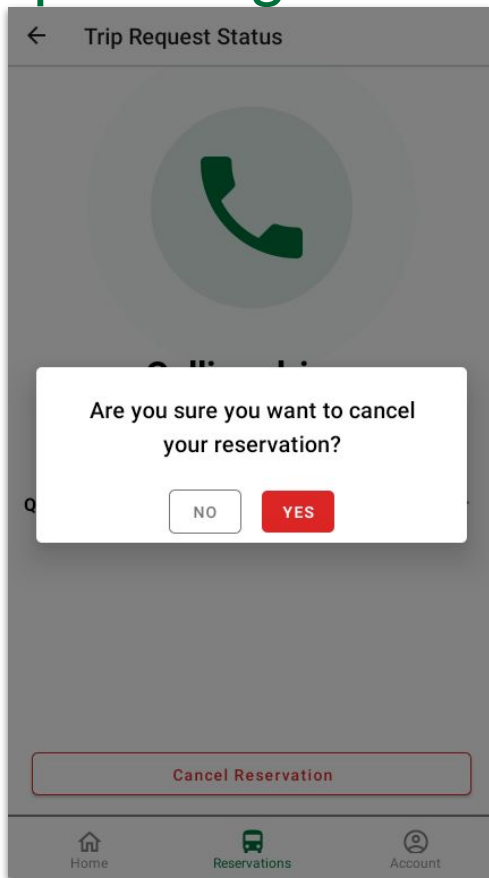
# Upcoming and Scheduled



## Trip Request Status Screen

- Screen where the passenger sees their request for reservation status.
- "Cancel Reservation" button
  - o Cancels the reservation
  - o If pressed, it will ask the passenger for confirmation.

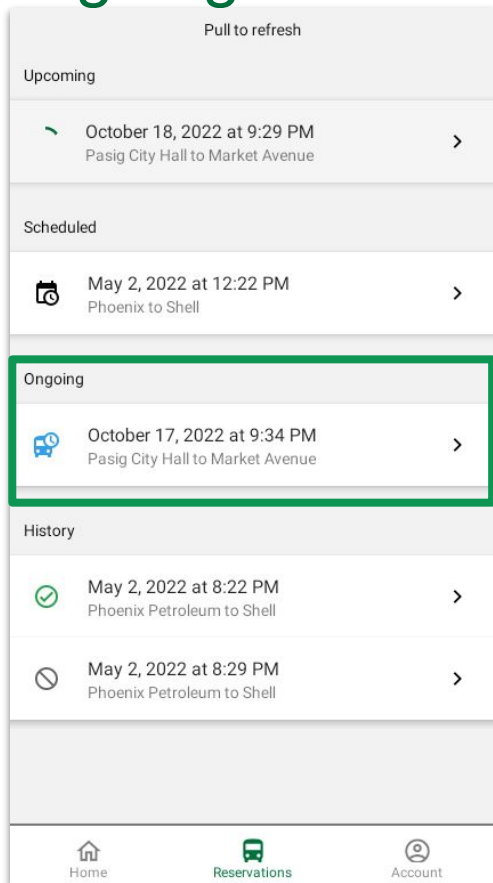
# Upcoming and Scheduled



## Trip Request Status Screen

- If the passenger presses "YES", it will send a status update to the server using the `updateReservationStatus` mutation call and pass the `CANCELED` reservation status.

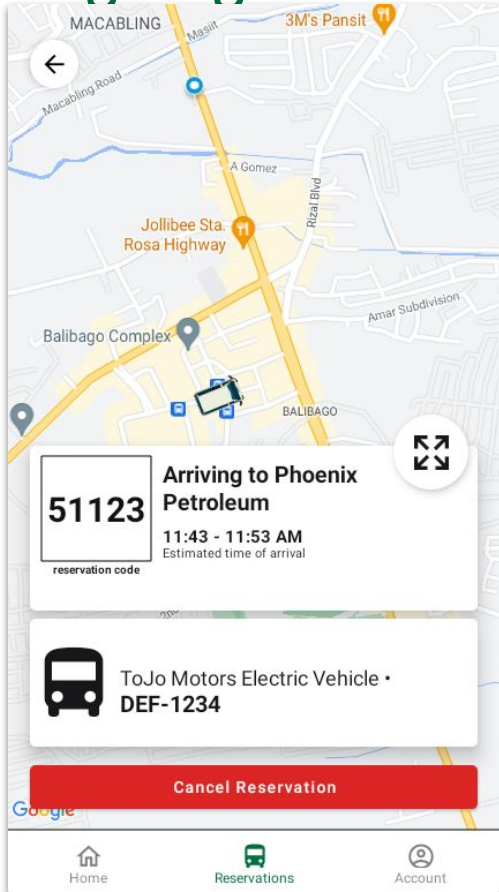
# Ongoing



## Reservation List Screen

- Pressing the **Ongoing** reservations will navigate to the **Booking Details Screen**.
- Pressing on the **Ongoing** reservations will also query the trip data using the trip query call.

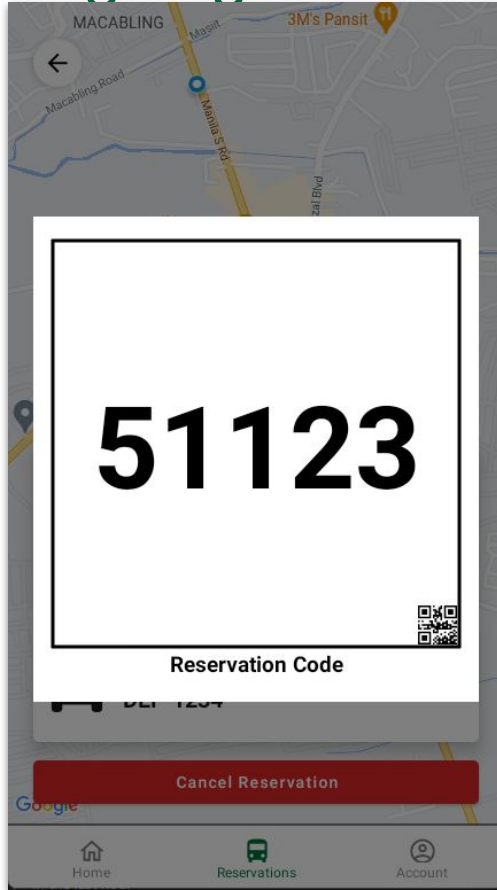
# Ongoing



## Booking Details Screen

- Screen where the passenger can see the vehicle for the trip, location of the vehicle in real-time, its details, its destination, and the trip details.
- Uses `onVehicleLocationUpdated` subscription to get the live location of the vehicle.
- Uses `vehicle` query to get the vehicle details.
- “Cancel Reservation” button is the same as in [Slide 71: Upcoming and Scheduled](#)
- Vehicle destination icon will depend on the status of the Reservation
  - If it's `CONFIRMED`, the icon will be **origin icon**
  - If it's `BOARDED`, the icon will be **destination icon**
- Popup reservation code
  - If “Expand” button is pressed, a popup containing reservation code will appear.
  - If the vehicle is  $\leq 500$  meters, the popup will also appear.

# Ongoing

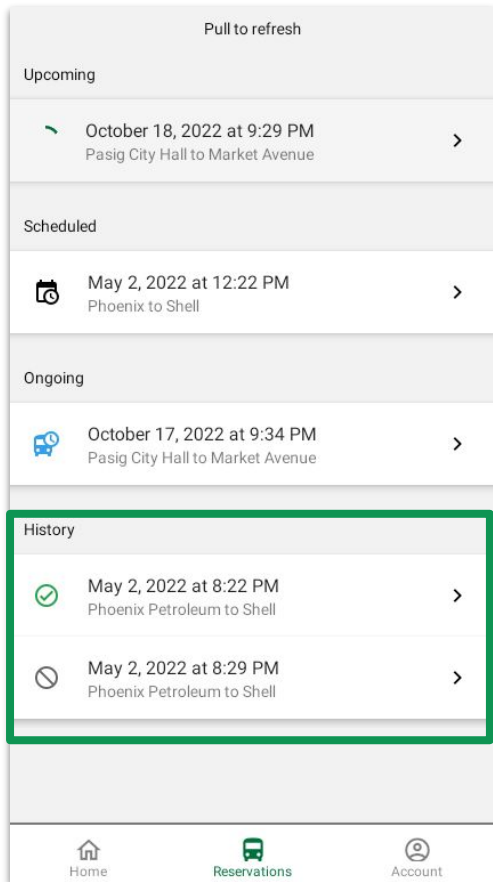


## Booking Details Screen

- Pressing the code will expand the QR code and minimize the text code, and vice versa.



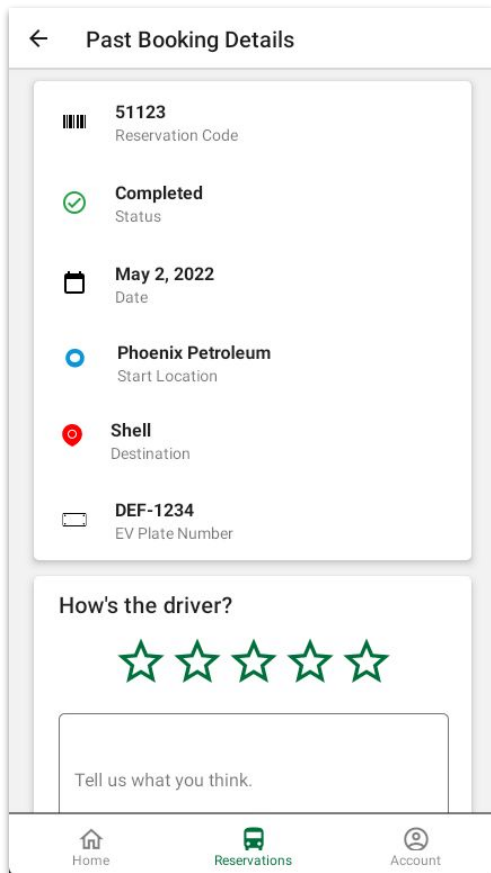
# Past Reservation



## Reservation List Screen

- Pressing the **History** reservations will navigate to the **Past Booking Details Screen**.

# Past Reservation



## Past Booking Details Screen

- Screen where the passenger can view the details of their past reservation.
- If the status of reservation is “COMPLETED” and the passenger still has to review, it will show the **Driver Rating Card**.

# Past Reservation

← Past Booking Details

✓ Completed  
Status

📅

May 2, 2022  
Date

📍

Phoenix Petroleum  
Start Location

📍

Shell  
Destination

📋

DEF-1234  
EV Plate Number

How's the driver?

☆ ☆ ☆ ☆ ☆

Tell us what you think.

0

SEND

Home

Reservations

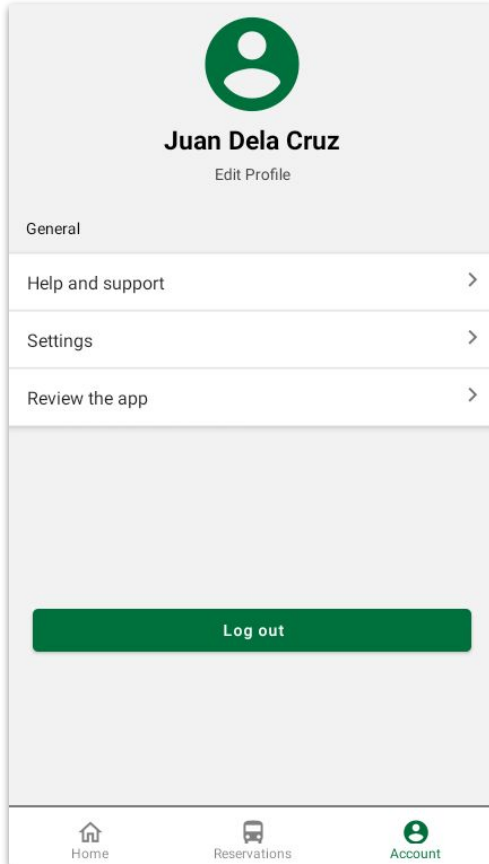
Account

## Driver Rating Card

- The passenger can rate the driver from 1 to 5 stars and can also provide a text feedback of up to 500 characters.
- “Send” button
  - If pressed, it will send the feedback through the `createDriverFeedback` mutation call.
    - If successful, it should return a success message. Otherwise, it will show an error message.

# Account Screens

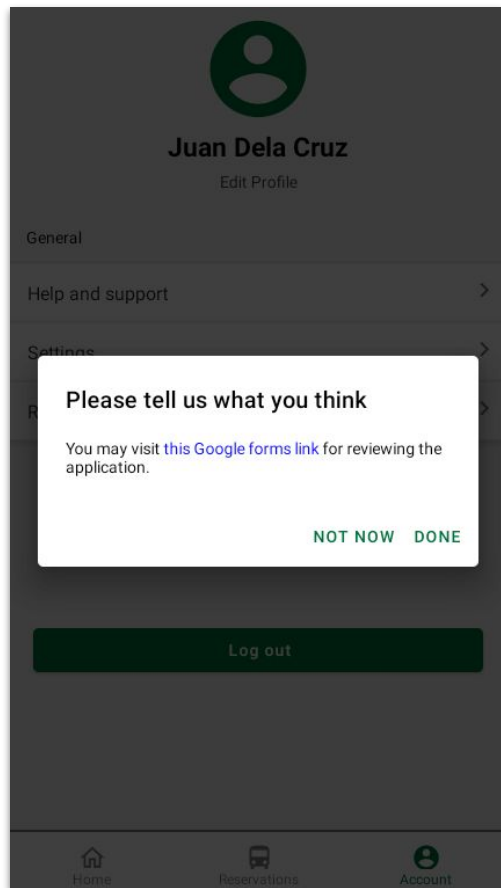
# Account



## Account Index Screen

- Screen where the passenger can see the general settings, edit their profile, and review the app.
- “Log out” button
  - When pressed, it will logged out the user.
  - Deletes the stored token.
- “Review the app” item
  - When pressed, it a review dialog will appear.
- “Edit item” button
  - When pressed, it go to **Edit Profile Screen**.

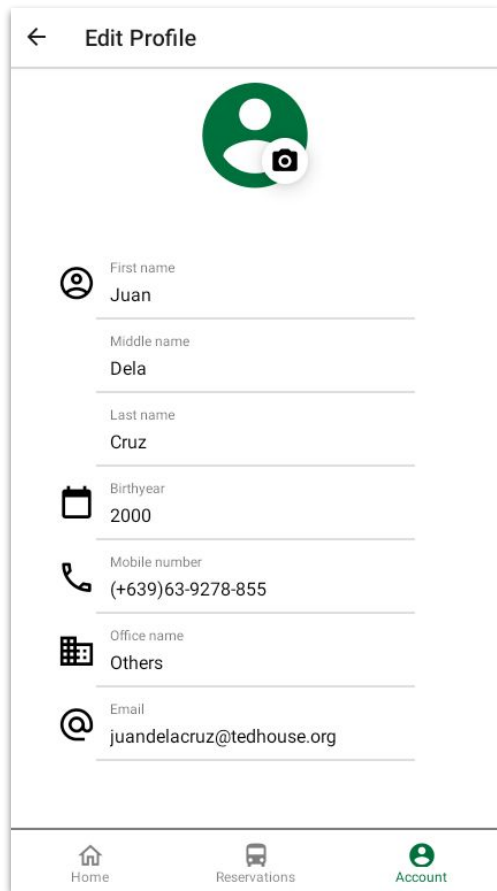
# Account



## Review Dialog


- It prompts the passenger to review the application through the Google form [link](#). When pressed, it will redirect to Google forms.
- Pressing the “NOT NOW” button will postpone the review 1 month.
- Pressing the “DONE” button will close the dialog.
- The **Review Dialog** will appear every month.


# Account





The image shows a mobile app interface for editing a user profile. At the top, there is a back arrow and the title 'Edit Profile'. Below this is a large green circular profile picture placeholder with a camera icon. The form contains several input fields, each with an icon and a label: a person icon for 'First name' (filled with 'Juan'), a calendar icon for 'Birthyear' (filled with '2000'), a phone icon for 'Mobile number' (filled with '(+639)63-9278-855'), a calendar icon for 'Office name' (filled with 'Others'), and an email icon for 'Email' (filled with 'juandelacruz@tedhouse.org'). There are also empty input fields for 'Middle name' (filled with 'Dela') and 'Last name' (filled with 'Cruz'). At the bottom, there is a navigation bar with three icons: a house for 'Home', a bus for 'Reservations', and a person for 'Account' (which is highlighted).


← Edit Profile





 First name  
Juan


 Middle name  
Dela

 Last name  
Cruz

 Birthyear  
2000

 Mobile number  
(+639)63-9278-855

 Office name  
Others

 Email  
juandelacruz@tedhouse.org

Home Reservations Account

## Edit Profile Screen

- Screen where the passenger can see and edit their details.