

## Test Plan and Report

### Hitch

Harrison Nou, Anthony Duong, Josh Defesche,

Andrei Ofilian, Aiden Brem

9 December 2023

## System Test Scenarios:

These system test scenarios show different features users should be able to access.

- Scenario 1: (LOG IN) **PASS**
  - If user does not have an account, the user should be able to plug First Name, Last Name, Username, Email, Phone Number, and Password Twice
  - Users should be able to log into the account and be at the home page.
- Scenario 2: (USER INTERFACE) **PASS**
  - Account is created and user is logged out; and user puts in their username and password to login
  - Users should be able to view a homepage that gives an option button to either create a ride or hitch a ride.
- Scenario 3: (ACCOUNT INFORMATION/ LOGOUT) **PASS**
  - The user presses the user icon in the corner
  - Users should be able to view a page that provides account information with a button that allows them to either log out of their account
- Scenario 4: (SHOW COMPLETED RIDES) **PASS**
  - User presses button 'Show Completed Rides'
  - Users should be able to access a page that allows them to view the completed rides with the user.
- Scenario 5: (CREATE RIDES) **PASS**
  - User presses the 'Driver' icon button on the homepage
  - Users should be able to access a page that allows them to view a map and give them the option to create a ride with a start point and destination.
- Scenario 6: (HITCH A RIDE) **PASS**
  - User presses the "Hitch a Ride" icon button on the homepage
  - Users should be able to access a page that allows them to view a map and give them an option to confirm route.
- Scenario 7: (RIDER VIEW HITCHES RIDES) **PASS**
  - When inside the "Hitch A Ride" page, user clicks the 'View Pending Rides' button at the top of the page
  - User should be able to see a refresh status button and the pending rides that they have in progress
- Scenario 8: (DRIVERS SPECIFY RIDE DETAILS) **PASS**
  - Driver submits the startpoint and destination
  - The driver should be given further inputs to submit maximum rider count, cost per rider, and Maximum distance with an option to submit time and date of when this ride will occur
- Scenario 9: ( DRIVER SUBMITS RIDE) **PASS**

- Driver clicks "Submit Your Route" icon button
- It should give an alert to the driver that the ride is now created and they should be able to view it in the "View Created Rides" page.
- Scenario 10: (RIDER CAN CHOOSE PRICE RIDES) **PASS**
  - User submits startpoint and destination
  - Users should be able to be given choices of the maximum price they are willing to pay and the maximum distance they are willing to be dropped off from their location
- Scenario 11: (RIDER CAN VIEW AVAILABLE RIDES) **PASS**
  - User click "Find Your Ride" button
  - Users should be able to be given choices of rides that are within distance of the given route they had posted or rides they already have pending.
- Scenario 12: (RIDER CAN HITCH) **PASS**
  - User picks a ride to join
  - Users should be notified they are now part of the ride and the ride they had clicked is now part of the Pending Section.
- Scenario 13: (RIDER CAN CANCEL RIDE) **PASS**
  - User goes into the pending ride section and presses the cancel button under the pending ride
  - User's ride should be removed from the pending rides page.

## Unit Testing:

- For unit testing, we were able to create a test file that went through the queries used within the server code to make sure they all worked individually in the database. Having this code allowed for the team to check if the queries worked individually before running them through the server.
- None of these had failed before the system version was released, but it allowed us to verify if certain variables within the database were connected. We were able to catch some unique issues, like if the values taken for each column were accepting the correct values, and if the values were restricted to being unique.
- Testing Environment works in the way that it submits an account, driver info, and creates rides and riders and makes sure they function properly, then removes them from the database so that the test queries do not show up on the interface
  - For example, the test creates hitch rides that could show in the database and in the interface as an available ride for the user, but is removed so that it won't appear for the users.
- The testing during the final stages of the product is able to simulate two different accounts being created and assign a ride and driver to them through a specific ride id that essentially simulates the ride process through the app, which then completes the ride and gives us the information we need to know about our query functions. Any test accounts or rides created in the process are removed at the end