

# System and Unit Test Report

Product Name:

Team: RideShare Price Comparer

Date: 12/8/2020

## System Test Scenarios

### Sprint 1

1. As a user, I want a landing page design, so I know what I am going to build for mobile and web.

#### Scenario:

1. Go the Rideshare Price Comparer website
2. User should see the page header, and the two input boxes labeled Pickup Location and Dropoff Location.

### Sprint 2

1. As a user, I want a mobile friendly web application so that I can easily access this app on the go

#### Scenario

1. Open the Rideshare Price Comparer app on a mobile device
2. User should see the labels for pickup and dropoff stacked on top of each input box

### Sprint 3

No user stories were completed in this sprint

### Sprint 4

1. As a user, I want to visually see which rideshare app is cheaper so I can speed up the checkout process
2. As a user, I want to have a map displaying the locations so that I know they are the correct locations

#### Scenario

1. Open the Rideshare Price Comparer app on any device
2. Type:
  - Pickup Location: San Francisco
  - Press Enter Key or click on the option that says San Francisco, CA
  - User should see the map centered relatively on San Francisco, CA and a red pin on that location with its address.
  - Dropoff Location: San Jose
  - Press Enter Key or click on the option that says San Jose, CA
  - User should see the two location pins, one red and one blue, inside the map, and a line connecting the pins.
3. Click the button labeled "Compare!"
4. After waiting for a few seconds, the user should see the Uber and Lyft Prices lists.

## Unit Tests

### Front End

- Loading Components Tests
  - GET Map element
    - The Google Map must appear on loading the page
  - Form Components defined
    - The Pickup Location input box must appear on loading the page
    - The Drop-off Location input box must appear on loading the page
  - Clicking Button
    - The Compare! Button must appear and be clickable
  - Uber and Lyft Price Lists do not appear on loading
- Address Form Input tests
  - Clicking off a box after typing in it will clear it
    - After typing in the pickup box, the box must not be empty
    - After clicking away from the pickup box, the box must be empty
    - The same goes for the drop-off box
  - A selected address from the drop down will be displayed in the text box
    - After typing part or all of an address in either input box, a drop down menu must appear containing a list of addresses
    - After selecting an address from the menu, the box will display the option that was selected
- Map Tests
  - Both the location pin and its address show up when an address is entered and deleted when an address is deleted
    - For each box, after selecting an address, a location pin should appear on the map pointing to the exact latitude and longitude of the address
  - Two location pins show up when both addresses are selected from each box
- Prices Output Test
  - Display Uber and Lyft prices when both addresses are filled out and the button is clicked

### Back End

- Geocode Middleware Testing Suite
  - when invalid API key is set, it resolves the promise with a 500 error
  - when unknown Google Error exists, it rejects the promise with another 500 error
  - when given correct params, it returns a successful resolved promise
- Geocode Testing Suite
  - when sending preflight request, it returns 200 with empty body
  - when given proper params, it returns 200 OK with proper body

- when query is undefined, it returns 400 BAD REQUEST error with correct fields
  - when API key is unset, it returns 500 Internal Server Error
  - when query is missing, it returns 400 BAD REQUEST with appropriate body
- Middleware Testing Suite
  - when auth key is the correct one, it returns no errors
  - when auth key is not set, it returns 500 error
  - when given incorrect auth token, it returns 401 error
- getUberPrices Testing Suite
  - when given incorrect params, it returns a 400 error with appropriate fields
  - when given correct params, it returns appropriate 200 response
  - when tokens are incorrectly set, it returns a 500 error with correctly set fields
- getLyftPrices Testing Suite
  - when given incorrect params, it returns 400 status code with error message
  - when given correct params, it returns with a with no errors
  - when unknown error happens server side, it correctly displays 500 error message
- Autocomplete Testing Suite
  - when sending preflight request, it returns 200 with empty body
  - when correct params and authentication are present, returns 200 with body
  - when unexpected error happens on google's end, returns 500 error
  - when request is not authenticated, it returns a 401 error
  - when sending incorrect params, it returns a 400 response
- getPrediction Google API Testing Suite
  - when an error happens on google's end, it returns a 500 error
  - when given a valid input, it returns a resolved promise with correct data
- getPrices testing Suite
  - when preflight request is sent, returns a 200 OK response
  - when validating, if incorrect token is supplied, it returns 401
  - When validating, if token is unset in environment, it returns 500 error message with correct fields
  - when given incomplete params and when lyft's params are invalid, it should return 400 bad request error message
  - when given incomplete params and when uber's params are invalid, it throws a 400 error bad request
  - when given correct params, it returns 200 successful response with appropriate fields
  - when server side error happens with uber, it throws a 500 error pointing
  - when server side error happens with Lyft, throws a 500 error pointing