

# Test Plan and Report

## Cafe Hopper

November 26, 2024

### System Test Scenarios

**User story A:** *As a user, I want to see information about cafes I search for such as name, address, hours, an image of the cafe, and reviews about it.*

Scenario 1: User clicks on cafe pin from map page

Scenario 2: User clicks on cafe from explore list view

Result: User should see the corresponding cafe in the cafe page with all information

**User Story B:** *As a cafe visitor, I want to be able to log my visits to cafes.*

Scenario 1: User clicks on “log a visit” on the cafe page

Result: User can fill out a form for their visit and have it uploaded to the platform

**User Story C:** *As a user, I want to add cafes to my personalized lists (e.g., liked or to-go) and have the action reflected across the platform.*

Scenario 1: User presses the "Done" button to add a cafe to a list.

Scenario 2: User opens and closes the "New List" component to create or manage lists.

Result: The cafe should be added to or removed from the selected list, and the "New List" component should open and close as expected.

**User Story D:** *As a user, I want to see detailed information about a cafe, such as its name, address, hours, rating, tags, and image, so I can decide whether to visit.*

Scenario 1: The ListCard component renders correctly with all information provided.

Scenario 2: The ListCard component renders a default image when no cafe image is provided.

Scenario 3: The ListCard component renders all tags when they are available.

Scenario 4: The ListCard component gracefully handles missing tags.

Result: The ListCard component should display the cafe details accurately or handle missing data gracefully without breaking.

**User Story E:** *As a user, I want to create a new list with a name, description, and privacy settings to organize my cafes.*

Scenario 1: The "New List" component renders correctly with all required elements.

Scenario 2: User closes the modal using the close button.

Scenario 3: User inputs a name and description for the list, and the input fields handle text changes correctly.

Scenario 4: User toggles between "Public" and "Private" privacy options.

Scenario 5: User tries to create a list without entering a name and receives an alert.

Result: The "New List" component should render and behave as expected, ensuring a smooth and functional experience for creating new lists.

**User Story F:** *As a user, I want to explore cafes through a map or list view, with the ability to search for cafes and apply filters to find the ones I'm interested in.*

Scenario 1: User switches to the "Map" view from the toggle button and sees the map.

Scenario 2: User interacts with the search bar to search for cafes or characteristics.

Scenario 3: User moves around in the "Map" view to search for cafes in that given area.

Scenario 4: User sees toggle buttons for switching between "List" and "Map" views.

Scenario 5: User focuses on the search bar and sees filter options appear.

Scenario 6: User selects a filter (e.g., "Open now") and the selected filter is updated.

Result: The "Explore" screen should render the search and view toggling functionality correctly, allowing users to explore cafes seamlessly. The cafes returned should fall under the filters specified by the user. The rendered markers on that map view should continuously update whenever the user stops moving around in the map at that given location.

**User Story G:** *As a user, I want to be able to customize my profile so I can showcase my preferences and make my experience feel more personalized.*

Scenario 1: Profile customization

1. User presses the "Edit Profile" button on their profile page and sees the profile customization page. Default profile has fields filled in as:

Name= <Cafe lover>

Bio=<I love cafes!>

Location=<Cafe>

2. User can edit all these fields to their own discretion using text boxes
3. User presses "Save changes" and the page will exit back to the profile page
4. User should see a pop up message that changes have been saved.

Scenario 2: Check profile

1. User goes to profile tab in navigation bar
2. User should see all their account information [name, bio, location, favorite cafes (if set), lists]

**User Story H:** *As a user, I want to be able to make an account and sign in so that I can review cafes and they will be saved under my identity.*

Scenario 1: User provides an email address and password to create an account on the sign up page.

Scenario 2: User can use their email address and password, after creating an account, to login to the app.

Scenario 3: User may provide incorrect credentials when attempting to login to the app.

Result: Assuming valid email and password is provided for login and sign up, the user is taken to the homepage of the app where they can explore the cafes. If any incorrect credential is provided or there is an error with login, the corresponding notification will alert the user.

**User Story I:** *As someone opening the app for the first time, I want the option to use the app as a guest, without being forced to create an account.*

Scenario 1: User can click on the Continue as guest button on the sign up page.

Result: Clicking on the Continue as guest button will take the user to the app home page where they can use the app. The “guest-level” features are managed by the other pages.

---

## Unit Tests

### Frontend:

Directory: /\_\_tests\_\_

#### CafeTest.tsx

- Render cafe name and address

- Like button functionality

- To-go button functionality

- Log a visit button functionality

- Render average review rating

#### ReviewTest.tsx

- Render profile name, review content, and like review button

#### AddToList.test.tsx

- Renders the component with a loading spinner initially.

- Verifies "Done" button functionality to exit the adding-to-list state.

- Tests the opening and closing of the "New List" component.

#### ListCard.test.tsx

- Renders the ListCard component correctly.

- Displays the cafe name, rating, address, and tags.

- Renders a default image when no cafe image is provided.

- Renders all tags when they are provided.

- Handles missing tags gracefully without breaking.

- Displays a placeholder rating when no rating is provided.

#### NewList.test.tsx

Renders the NewList component correctly, including all modal elements such as input fields, privacy options, and buttons.

Closes the modal when the close button is pressed.

Handles input changes for the list name and description fields correctly.

Toggles privacy settings between "Public" and "Private" accurately.

Displays an alert if the "Create" button is pressed without entering a list name.

### **Explore.test.tsx**

Renders the "Map" view correctly when toggled.

Renders the search bar correctly.

Displays toggle buttons for switching between "List" and "Map" views.

Shows the filter dropdown when the search bar is focused.

Updates the selected hour filter (e.g., "Open now") when clicked.

### **start-test.tsx**

Redirects to tabs page if user is logged in

Renders the "Get started" page if no user is logged in

### **sign-up-test.tsx**

Renders all fields and buttons

Navigates to login screen

Shows an alert if passwords do not match

Signs up user and navigates to tabs if successful

Shows alerts for errors during signup or profile creation

Allows toggling password visibility

### **login-test.tsx**

Renders the login screen correctly

Toggles password visibility

Calls signInWithEmail and navigates on successful login

Shows an alert on login error

Navigates to the sign-up page

## **Backend:**

**Directory: /src/ \_\_tests\_\_**

### **CafeRoutes.test.ts**

Test if searching cafes regularly returns 200 status code with proper response body

Invalid geolocation format should return 400

Invalid sortBy option format should return 400

Searching by tags should return 200 and every cafe should have at minimum the tags provided in the request

Searching without a query should still return cafes as long as geolocation is provided

Searching with a customTime should return 200 and the cafes should be open at that time

Searching with a customTime should allow either day or time to be omitted

Searching with an incorrect customTime provided should return 400

Searching by rating should return 200 and the cafes should have at least that rating