Lab 4 Artifacts - team2345

Members: Valeria Garibaldi, Krischin Layon, Jeevan Sandhu, Justin Wang

Features (user stories) to Implement in Next Sprint:

Listings data:

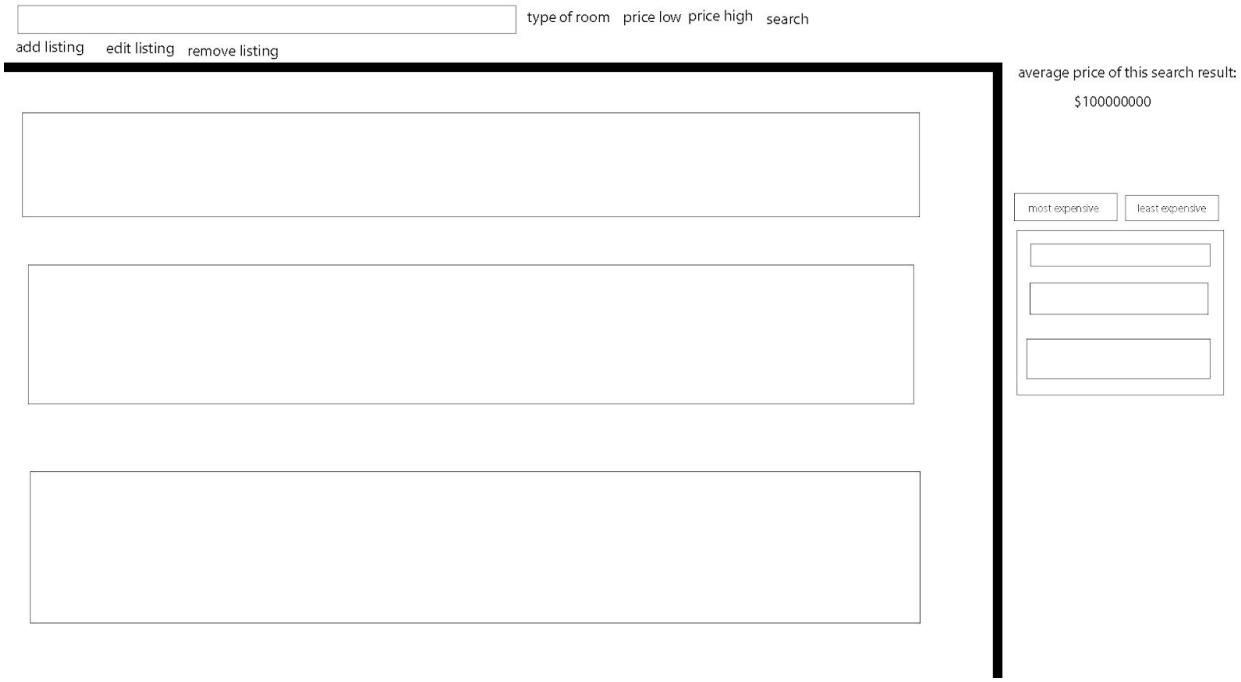
-**Feature 1:** As a user, I want to know the average price for a rental in each neighborhood.

-**Feature 2:** As a user, I want to know which listings are the least expensive in each neighborhood.

-**Feature 3:** As a user, I want to know which listings are the most expensive in each neighborhood.

GUI:

Analytics Tab Page



Test Cases:

- **Feature 1 Test Cases:** As a user, I want to know the average price for a rental in each neighborhood.
    - **Test case 1**: As a user, I search for a neighborhood in the listings page with a certain price range.
      Correct Output: The front end displays on the listing page the average price of all the listings in the selected neighborhood and price range.

    - **Test case 2:** As a user, I search for a room type in addition to neighborhood and price range.
      Correct Output: The frontend displays the average price for the listings that fit the search criteria.

    - **Test case 3:** As a user, I make an empty search.
      Correct Output: The frontend does not display average pricings for any listings.

- **Feature 2 Test Cases:** As a user, I want to know which listings are the least expensive in each neighborhood.
    - **Test case 1**: As a user, I search for a neighborhood in the listings page.
      Correct Output: The front end displays a “least expensive” button. Clicking on the button displays a top 3 list of the least expensive listings in the selected neighborhood.

    - **Test case 2**: As a user, I search for a room type in addition to neighborhood and price range.
      Correct Output: The front end displays a “least expensive” button. Clicking on the button displays a top 3 list of the least expensive listings that fit the search criteria.

    - **Test case 3**: As a user, I make an empty search.

Correct Output: The front end does not display any information for the least expensive listings.

- **Feature 3 Test Cases:** As a user, I want to know which listings are the most expensive in each neighborhood.
    - **Test case 1**: As a user, I search for the neighborhood on the listings page.
      Correct Output: The front end displays a “Most expensive” button. Clicking on the button displays a top 3 list of the most expensive listings in the selected neighborhood.

    - **Test case 2**: As a user, I search for a room type in addition to neighborhood and price range.
      Correct Output: The front end displays a “Most expensive” button. Clicking on the button displays a top 3 list of the most expensive listings that fit the search criteria.

    - **Test case 3**: As a user, I make an empty search.
      Correct Output: The front end does not display any information for the most expensive listings.

To-Do List

Done list of last sprint:

- Added HTML file for Add, Remove, and Edit Listing Functionalities
  [finished by Valeria and verified by Everyone]
- Added container functionality to Listings HTML file to display Neighborhood, Room Type, and Price
  [finished by Valeria and verified by Everyone]
- Implemented Angular Components to handle new API calls (add/edit/remove)
  [finished by Krischin and verified by Everyone]
- Implemented Angular Services to handle new API (add/edit/remove)

[finished by Krischin and verified by Everyone]

- Set up API functionality to receive new POST Requests for Add, Edit, and Remove Listings

  [finished by Justin and verified by Everyone]
- Integrate API with Data Layer functions

  [finished by Justin and verified by Jeevan]
- Implement backend functionality to add, remove, and edit listings in the csv file

  [finished by Jeevan and verified by Everyone]
- Implement data backup functionality so that updated data is saved.

  [Finished by Justin and Jeevan and verified by Everyone]
- Implement data import so that updated data from a previous session is still available in a new session.

  [Finished by Justin and Jeevan and verified by Everyone]
- Add Highest ID tracking file “id.txt” for adding new listings

  [finished by Jeevan and Justin and verified by Jeevan and Justin]
- Conducted and confirmed Feature 1, 2, and 3 Test Cases from Sprint-3

  [finished by Jeevan and Justin and verified by everyone]
- Completed Sprint 3 Artifact

  [finished by everyone and verified by everyone]
- Record Sprint 3 Demo

  [finished by everyone and verified by everyone]

To-Do for <u>next sprint</u>:

- Revised Home Tab
  - Sidebar for analytics.
  - Spot to view cheapest + most expensive listings.
  - Add a spot for displaying the average price.
  - Add buttons for displaying cheapest and most expensive listings.
- Search HTTP Response includes analytics for:
  - Average Price of the whole search result
  - Top 3 cheapest listings of the whole search result

    - Top 3 most expensive listings of the whole search result
    - API Response
        - {
            “listings”: [(list of listing objects)],
            “average_price” : null,
            “cheap_listings”: [(list of 3 listing objects)],
            “expensive_listings”: [(list of 3 listing objects)]
        }
- Implement backend analytics for calculating average price of each search
- Implement backend analytics for calculating 3 cheapest listings of search result
- Implement backend analytics for calculating 3 most expensive listings of search result

To-do if there’s time:

- Add listing id section in container to show for each listing in the listing search results
- Start on calendar parser for finding calendar analytics
- Add confirmation message when deleting, adding, and editing listings.