Lab 5 Artifacts - team2345

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

Features (user stories) to Implement in Next Sprint:

Listings data:

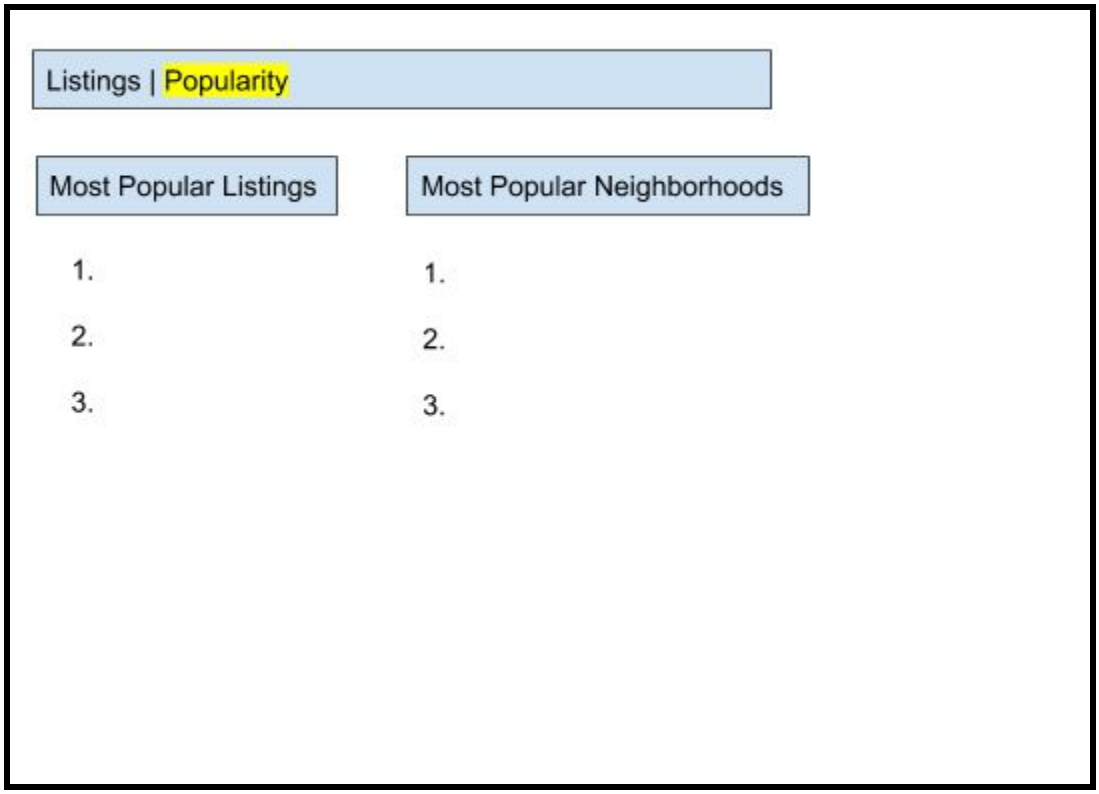
-**Feature 1:** As a user, I want to see the most popular AirBnB listings in all of Madrid.

-**Feature 2:** As a user, I want to see what are the most popular neighborhoods among AirBnB listings in Madrid.

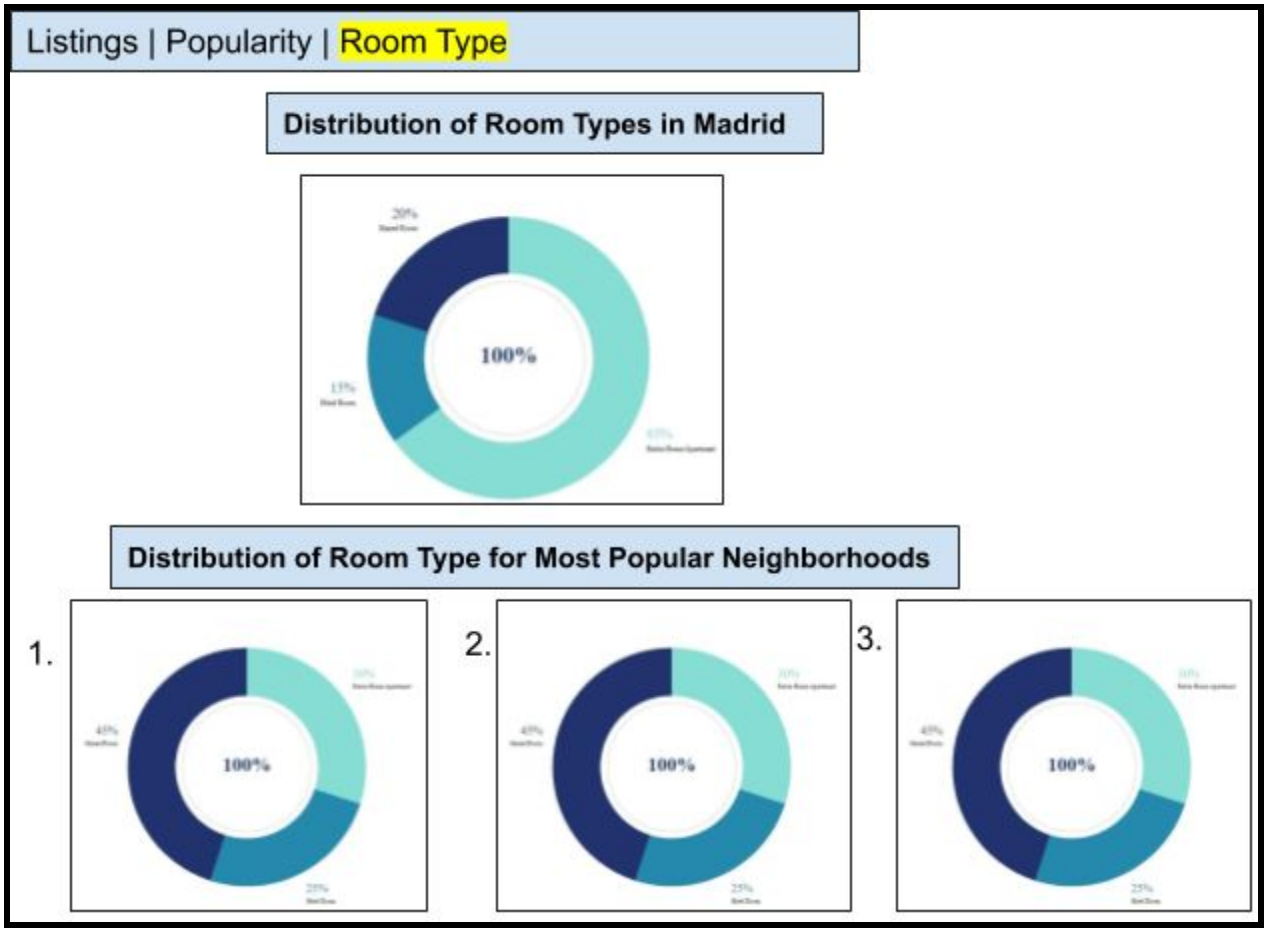
-**Feature 3:** As a user, I want to see the distribution of room types available in AirBnbs in Madrid.

GUI:

Popularity



Room Distribution



Test Cases:

- **Feature 1 Test Cases:** As a user, I want to see the most popular AirBnB listings in all of Madrid.
    - **Test case 1**: As a user, I click on the popularity tab.
      Correct Output: The front end displays on the Popularity tab and displays the top three most popular listings as well as the number of reviews for each listing.

- **Feature 2 Test Cases:** As a user, I want to see what are the most popular neighborhoods among AirBnB listings in Madrid.

- **Test case 1**: As a user, I click on the popularity tab.
  Correct Output: The front end displays on the Popularity tab and displays the top three most popular neighborhoods as well as the number of total reviews for each neighborhood.

Feature 3 Test Cases: As a user, I want to see the distribution of room types available in AirBnbs in Madrid.

- **Test case 1**: As a user, I click on the Room Type Tab.
  Correct Output: The front end displays the room type distributions for madrid AirBnBs and the three most popular neighborhoods.
- **Test case 2**: As a user, I verify that the popular neighborhoods match the ones on the popularity tab.
  Correct Output: The front end displays matching most popular neighborhoods on the Room Type Tab and Popularity Tab.

To-Do List

Done list of last sprint:

- Modified HTML file for Average price, most expensive listings, and cheapest listings analytics
  [finished by Valeria and verified by Krishcin]
- Modified CSS file for average price, most expensive listings, and cheapest listings analytics (Part 1 Analytics)
  [finished by Valeria and verified by Krishcin]
- Modified container functionality to Listings HTML file to display ID
  [finished by Krischin and verified by Everyone]
- Implemented Angular Components to handle modified API call (Part 1 Analytics)
  [finished by Krischin and verified by Everyone]
- Implemented Angular Services to handle modified API call (Part 1 Analytics)
  [finished by Krischin and verified by Everyone]
- Set up API functionality to send new JSON with analytic data upon receiving POST Request

[finished by Justin and verified by Everyone]

- Integrate API with Data Layer analytic functions

  [finished by Justin and Jeevan and verified by Justin and Jeevan]

- Implement backend functionality to calculate average price, most expensive listings, and cheapest listings analytics

  [finished by Jeevan and verified by Everyone]

- Conducted and confirmed Feature 1, 2, and 3 Test Cases from Sprint-4

  [finished by Jeevan and Justin and verified by everyone]

- Completed Sprint 4 Artifact

  [finished by everyone and verified by everyone]

- Record Sprint 4 Demo

  [finished by everyone and verified by everyone]

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

- Room Type Distributions Pie Chart
  - Acquire a Front-End Visualization Framework for this.
  - Save data, send only the counts of the different types of rooms
- Most Popular Neighborhoods (in /analytics)
  - Number of Reviews
- Top 3 Listings (in /analytics)
  - Number of Reviews
- New /analytics tab
- Post request to /analytics

  Json

  {

  top_neighborhoods: []

  top_listings: []

  room_dist_data: [

  {“Madrid”:[1,2,3]}, //dictionary w/ a list of pie chart data

  “Sol”:[1,2,3]},

  {“Embajadores”: [4,5,6]}

.... (Top 3 neighborhoods)

]

}

To-do if there's time:

- Implement fourth drop down for room type - Shared room
- Add confirmation message when deleting, adding, and editing listings
- Implement add review function to web app
    - Reviews page with review post request /review
        - Json
            - {

                "id": 0000 //id of listing to increment num reviews

              }
    - Backend to receive post request and increment num reviews in csv
    - Backup/import