

Lab 3 Artifacts - team2345

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

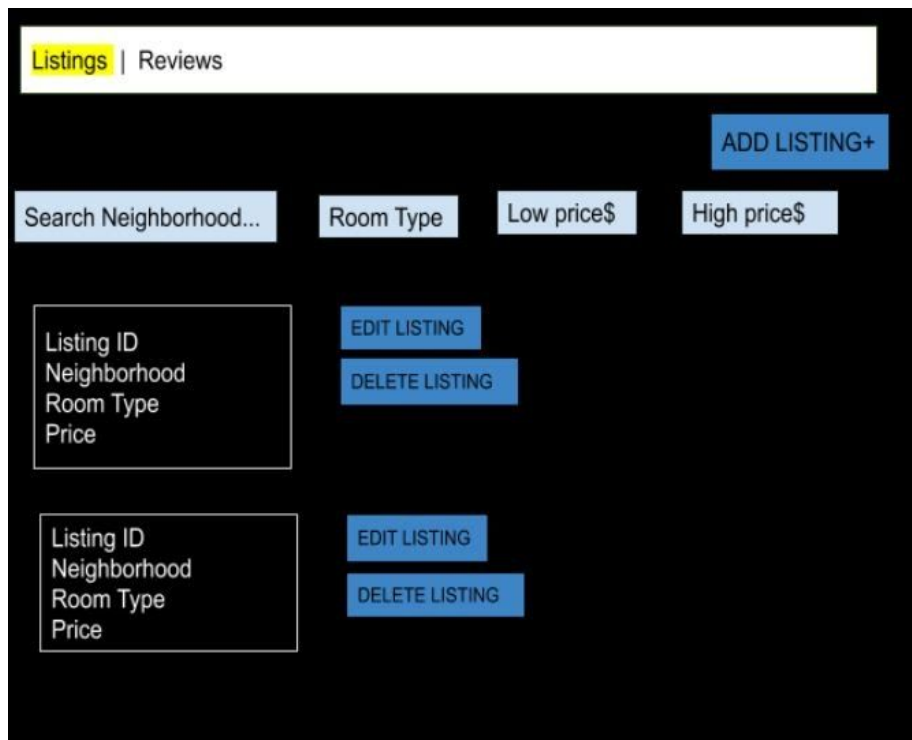
Features (user stories) to Implement in Next Sprint:

Listings data:

- Feature 1:** As a user, I want to add a new listing.
- Feature 2:** As a user, I want to delete an existing listing.
- Feature 3:** As a user, I want to edit an existing listing.
- Feature 4:** As a user, I want to view the id of the listings that are searched.
- Feature 5:** As a user, I want to make sure my data is backed up after refreshing the page.
- Feature 6:** As a user, I want to import updated data that was backed up in a previous session

GUI:

Listing Home Page (To implement)



Test Cases:

- **Feature 1 Test Cases:** As a user, I want to add a new listing.
 - Test case 1:** As a user, on the home page I first select the “Add Listing” button. Then I input the information of the new listing.
 - Correct Output: The backend assigns a new unique listing id to the new listing. The frontend flashes a successful “add” message and displays the information of the new added listing.
 - Test case 2:** As a user, I try to add a listing with missing fields.
 - Correct Output: The frontend displays a validation error, and no data is sent to the backend.
- **Feature 2 Test Cases:** As a user, I want to delete an existing listing.
 - Test case 1:** As a user, on the home page I first search for a listing. Then I click the “Delete Listing” button.
 - Correct Output: The frontend flashes a successful “delete” message, as well as the information of the listing that was just deleted. Searching the same parameters now will not display the listing that was just deleted.
- **Feature 3 Test Cases:** As a user, I want to edit an existing listing.
 - Test case 1:** As a user, on the home page I first search for a listing. Then I click the “Edit Listing” button and input the updated listing information.
 - Correct Output: The backend appropriately edits the specified listing information. The frontend flashes a successful “edit” message and displays the information of the newly edited listing.
- **Feature 4 Test Cases:** As a user, I want to view the id of the listings that are searched.

-**Test case 1:** As a user, on the home page I select a search criteria for a listing.

Correct Output: The backend returns the id of each matched listing. The frontend displays the id next to each listing.

- **Feature 5 Test Cases:** As a user, I want to see my backed up data after I refresh the page.
 - **Test case 1:** Enter new data (add/edit/remove), refresh the page, and search for the entered data from the previous session.

Correct Output: The data was backed up automatically and searching upon refreshing outputs the updated data (add/edit/remove).
- **Feature 6 Test Cases:** As a user, I want to import updated data that was backed up in a previous session.
 - **Test case 1:** Update data with add/edit/remove, back-up the data, and restart the application.

Correct Output: The updated data is imported, and searching the data outputs it on the screen.

To-Do List

Done list of last sprint:

- Set up an HTML file for listings GUI structure
[finished by Valeria and verified by Krischin]
- Set up a CSS file for listings GUI design
[finished by Valeria and verified by Krischin]
- Set up Angular Components to handle API calls
[finished by Krischin and verified by everyone]
- Set up Angular Services to handle API calls
[finished by Krischin and verified by everyone]
- Set up API to handle listing POST request
[finished by Justin and verified by everyone]
- Integrate API with Data Layer csv parser

- [finished by Justin and verified by Jeevan]
- Integrate data backup by updating the csv file in parallel with local client data
 - [finished by Justin and Jeevan and verified by everyone]
- Designed “listing” classes to store parsed data from CSV files
 - [finished by Jeevan and verified by everyone]
- Set up parser to parse listings.csv
 - [finished by Jeevan and verified by Justin]
- Conducted and confirmed Feature 1, 2 and 3's Test Cases from Sprint-2
 - [finished by Krischin and Justin and verified by everyone]
- Complete Sprint 2 Artifact
 - [finished by everyone and verified by everyone]
- Record Sprint 2 demo
 - [finished by everyone and verified by everyone]

To-Do for next sprint:

- Modify HTML file to accommodate new features
 - Add (displayed as a button at top of list)
 - Remove (displayed as button next to each listing)
 - Edit (displayed as button next to each listing)
 - These buttons automatically backup the data to the backend
- Add import functionality to the application that imports latest backup of data
- Add container functionality to HTML to separate/differentiate displayed listings from one another
 - Displayed values in container:
 - Listing ID
 - Neighborhood
 - Room Type
 - Price
- Add POST Requests for
 - Add
 - listings/add

- Remove
 - listings/remove
- Edit
 - listings/edit
- Implement Angular Components and Services to handle new API calls
- Add API functionality to receive new POST Requests
 - Feed request data to Data Layer
 - Return correct status codes
- Implement new functions
- Implement backend functionality to add a new listing.
- Implement backend functionality to edit an existing listing.
- Implement backend functionality to delete a listing.
- Implement Data backup functionality to the csv file when add/edit/delete is called

To-do if there's time:

- Set up html and css files for Review tab
- Add search by id functionality for reviews.csv
 - Implement frontend post request
- Connect delete functionality with reviews.csv file to delete all reviews associated with a listing that was just deleted
- Implement add/delete/edit functionality for reviews