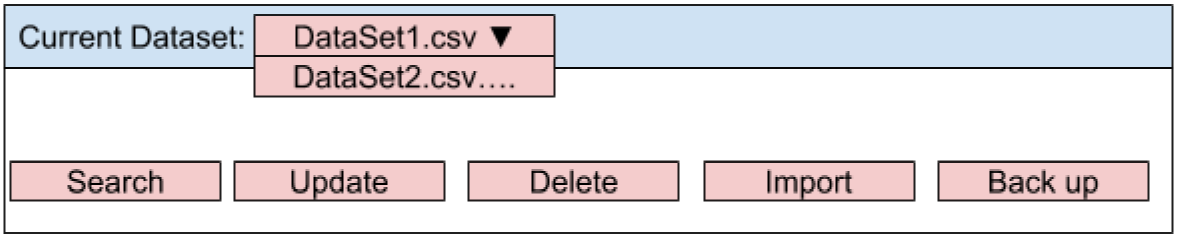
Airbnb - Lab 3 artifact

Features (user stories) to Implement in Next Sprint:

- **Update**
  (listings.csv)
  - **Feature 1**: as a user, I want to update my Airbnb listing's price.
  - **Feature 2**: as a user, I want to update my Airbnb listing's availability_365.
- **Create**
  (listings.csv)
  - **Feature 3**: as a user, I want to create a listing on Airbnb.
- **Delete**
  (listings.csv)
  - **Feature 4**: as a user, I no longer want to have an Airbnb listing.
- **Data**
  (listings.csv)
  - **Feature 5**: as a user, I want to backup my listings by replicating a csv file.
  - **Feature 6**: as a user, I want to import my listings from a specific csv file.

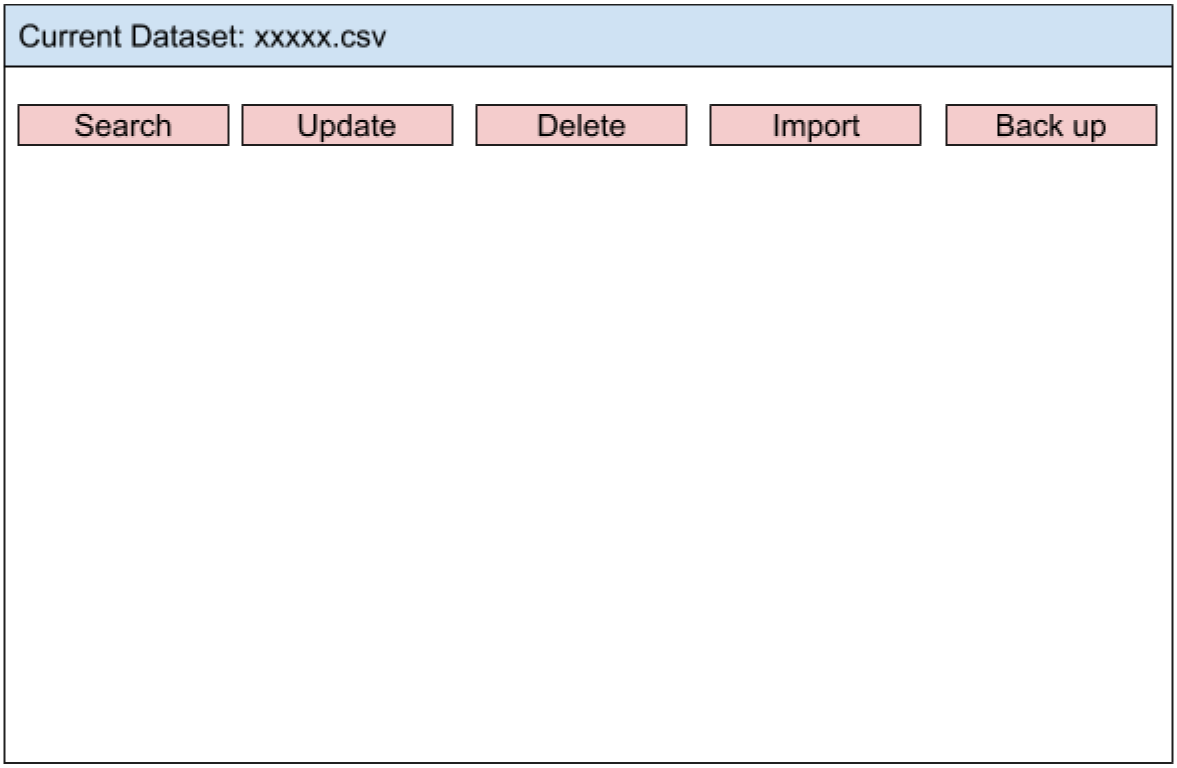
GUI

- **Navigation bar:** In the navigation bar, it has five buttons, one for search, one for updating a listing, one for deleting a listing, one for import, and one for backing up. The backup button ,when clicked, creates a new backup csv. The navigation bar also displays which data set/csv file the user is currently using.



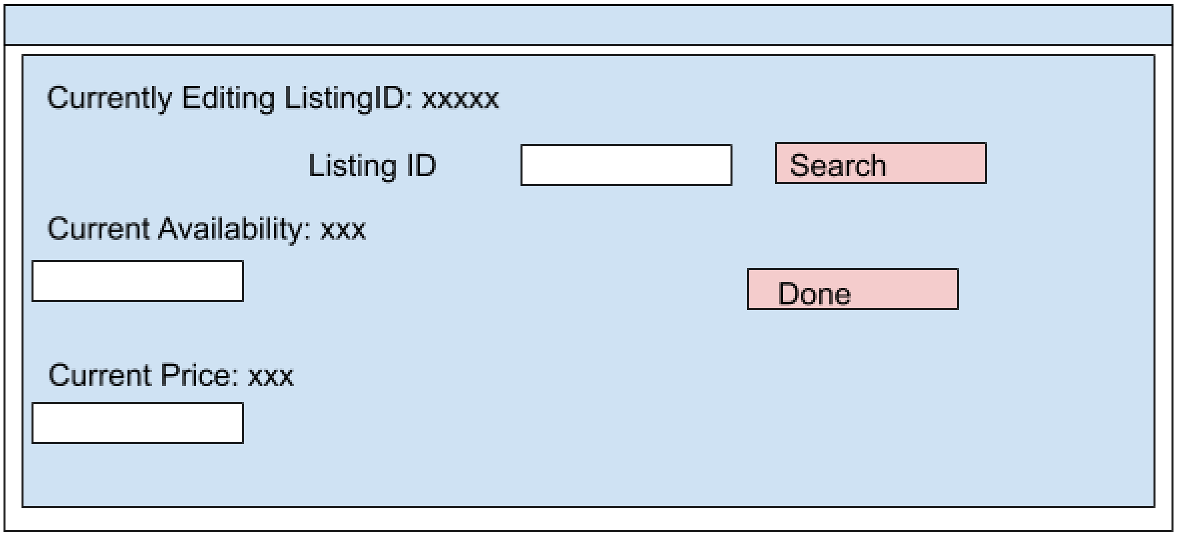
Current Dataset: DataSet1.csv ▼

- **Home page:** Home page that has four buttons, one for search, one for updating a listing, one for creating a listing, and one for deleting a listing. Each button redirects to the corresponding web page. (This is a possible option. The search page might be the home page if it is too difficult to move to this option).



Current Dataset: xxxxx.csv

- **Update page:** Web page has a form with inputs for availaility_365 and price and a button to submit the information.

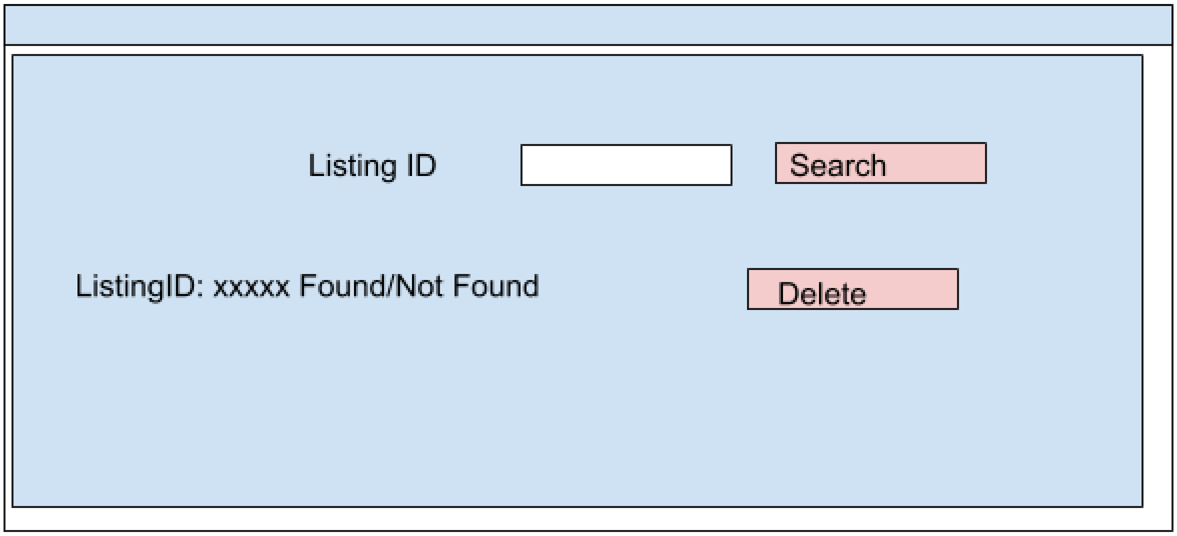


Currently Editing ListingID: xxxxx

Current Availability: xxx

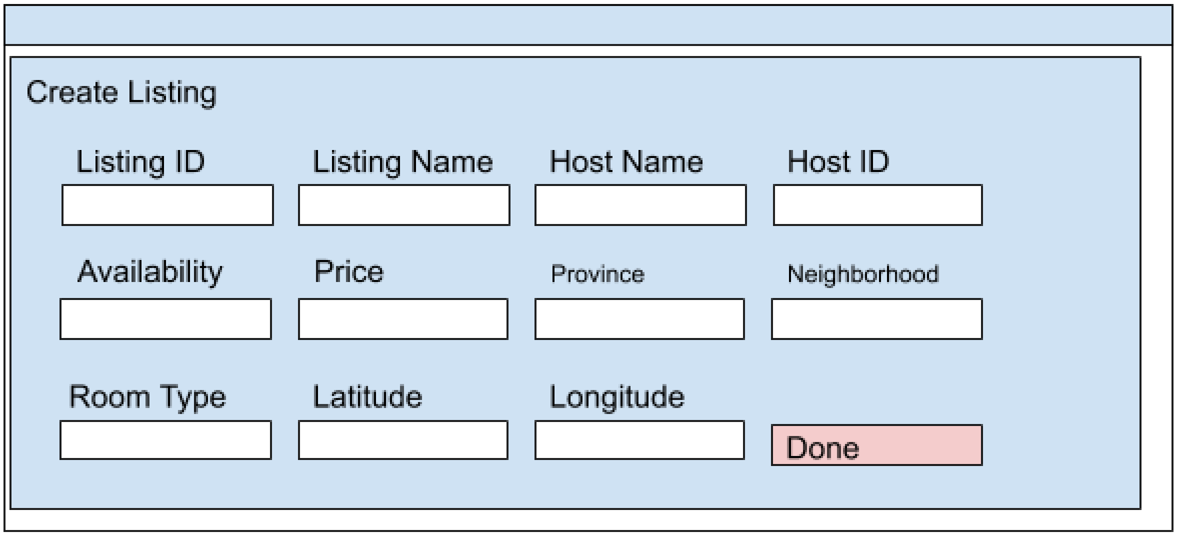
Current Price: xxx

- **Delete page:** Web page has a form with input for id (not host_id) and a button to submit the information.



ListingID: xxxxx Found/Not Found

- **Create page:** Web page has a form with inputs for availaility_365, id, name, host_id, host_name, neighbourhood_group, neighbourhood, latitude, longitude, room_type, and price and a button to submit the information.


Create Listing
Listing ID
Listing Name
Host Name
Host ID
Availability
Price
Province
Neighborhood
Room Type
Latitude
Longitude
Done

Test Cases

- **Feature 1/2 test cases**: as a user, I want to update my Airbnb listing's price/availability_365.
    - Test case 1: as a user, I enter the new price of my listing and submit the form.
        - Correct output: the website sends an update request to the back end and I am shown my updated listing.
    - Test case 2: as a user, I enter the new availability of my listing and submit the form.
        - Correct output: the website sends an update request to the back end and I am shown my updated listing.
    - Test case 3: as a user, I enter the new availability and price of my listing and submit the form.
        - Correct output: the website sends an update request to the back end and I am shown my updated listing.
- **Feature 3**: as a user, I want to create a listing on Airbnb.
    - Test case 1: as a user, I fill in all text fields, leaving no blank text field, and submit the form.
        - Correct output: the website sends a create request to the back end and I am shown my new listing.
    - Test case 2: as a user, I fill in some text fields, leaving 1 blank text field, and submit the form.
        - Correct output: the website sends the user an error message and asks the user to fill in all text fields.
- **Feature 4**: as a user, I no longer want to have an Airbnb listing.
    - Test case 1: as a user, I fill in my listing's ID and submit the form.
        - Correct output: the website sends a delete request to the back end and I am shown a confirmation that the listing I inputted is deleted
    - Test case 2: as a user, I fill in a non-existing listing's ID and submit the form.
        - Correct output: the website sends the user an error message and asks the user to input the correct listing ID.
- **Feature 5**:

    - Test case 1: as a user, I click on the “backup” button to backup my current data.
        - Correct output: the website sends a backup request to the back end and creates a new csv file with the information given up to that time.
- **Feature 6**: as a user, I want to import my listings from a specific csv file.
    - Test case 1: as a user, I want to import a specific data set to use and use the dropdown bar in the navigation bar to select which data set I want to use.
        - Correct output: the website sends an import request to the back end and imports the specified data set. The name of the csv file is updated in the navigation bar.

TODO LIST

Done list of last sprint:

- Front end creation -- finished by Jinhuan Liu
    - Create a frontend form component for searching by price range.
        - [finished by Jinhuan Liu and verified by everyone]
    - Create a frontend form component for searching a listing by room type.
        - [finished by Jinhuan Liu and verified by everyone]
    - Create a frontend form component for searching a listing by province/neighborhood.
        - [split into two separate components]
        - [finished by Jinhuan Liu and verified by everyone]
    - Create a frontend form component for searching a listing by availability.
        - [finished by Jinhuan Liu and verified by everyone]
    - Create a frontend form component for searching a listing by listing name.
        - [finished by Jinhuan Liu and verified by everyone]
- Search function -- finished by Vivian Tsai
    - Backend program to search listing.csv for sorted listings (display 15 results).
        - [finished by Vivian Tsai and verified by everyone]
    - Backend program to search listing.csv for listings within a price range (display 15 results).
        - [finished by Vivian Tsai and verified by everyone]
    - Backend program to search listing.csv for listings with a specific listing name.
        - [finished by Vivian Tsai and verified by everyone]
    - Backend program to search listing.csv for listings within a specific room type (display 15 results).
        - [finished by Vivian Tsai and verified by everyone]

    - Backend program to search listing.csv for listings within a province (display 15 results).
        - [finished by Vivian Tsai and verified by everyone]
    - Backend program to search listing.csv for listings within a neighborhood (display 15 results).
        - [finished by Vivian Tsai and verified by everyone]
    - Backend program to search listing.csv for available listings (display 15 results).
        - [finished by Vivian Tsai and verified by everyone]
- NOT INCLUDED IN LAST WEEK'S ARTIFACT BUT WERE TASKS THOUGHT OF AFTER IT WAS SUBMITTED
    - Backend program to parse csv files.
        - [finished by Qi Liu and verified by everyone
    - Create a frontend form component for two-way sorting the price.
        - [finished by Jinhuan Liu and verified by everyone]
    - Backend REST API to send data from the client to server.
        - [finished by Ruth Navarrete and verified by everyone]
    - Write user test cases for searching by neighbourhood_group.
        - [finished by Yinjie Xie and verified by everyone]
    - Write user test cases for searching by neighbourhood.
        - [finished by Yinjie Xie and verified by everyone]
    - Write user test cases for searching by room_type.
        - [finished by Yinjie Xie and verified by everyone]
    - Write user test cases for searching by name.
        - [finished by Yinjie Xie and verified by everyone]
    - Write user test cases for searching by host_name.
        - [finished by Yinjie Xie and verified by everyone]
    - Write user test cases for searching by price range.
        - [finished by Yinjie Xie and verified by everyone]

- Secondary notes: our Github Kanban board is more accurate as to how work is being divided than the artifacts. Tasks are broken down here and then group based on functionality in GitHub. The overall grouping is then created into an issue and everyone is assigned tasks.

ToDo task list for the next sprint:

- Create a frontend form component for adding a new listing.
  - Acceptance criteria: the form receives data, such as price, host id, room-type, availability_365, and neighborhood to add a new list.
- Create a frontend form component for deleting an existing listing.
  - Acceptance criteria: the form receives a host id from the user and sends the delete request to the server.
- Create a frontend form component for updating a current listing.
  - Acceptance criteria: the form receives data inputted by the user and sends information to the backend when the user submits the form.
- Create a frontend component for the navigation bar:
  - Acceptance criteria: the bar has 5 buttons and a dropdown. The search, update, delete, import, backup buttons navigate to the corresponding page. The backup button and drop bar sends requests to the server.
- Backend program to modify REST API to include POST request for updating a listing.
  - Acceptance criteria: the method updates one listing based on user input.
- Backend program to modify REST API to include PUT/PATCH request for creating a listing.
  - Acceptance criteria: the method creates one listing based on user input.
- Backend program to modify REST API to include DELETE request for deleting a listing.
  - Acceptance criteria: the method deletes one listing based on user input.
- Backend program to modify REST API to include a request for creating a backup csv file.
  - Acceptance criteria: the method creates one new csv backup.
- Backend program to modify REST API to include a request for importing a csv file.

    - Acceptance criteria: the method changes the dataset being used based on user input.

- Create unit tests for Insert, Delete, Update, Backup, and Import.
    - Acceptance criteria: verify that all functions do expected behavior.