CSC 648-848 Software Engineering Summer 2020
Milestone 4

Freshwater Rentals

Rent, sell, buy housing/apartment website

Available on desktop and mobile.

Team 1 : San Francisco State Coders (SFSC)

Luis Chumpitaz	**Team Lead**	**lchumpit@mail.sfsu.edu**
Garrett Peuse	**Backend Lead**	
Gouri Jamakhandi	**Frontend Lead**	
Katthak Shah	**Github Master**	
Nathan Hildum		
Zachary Ma		
Xinwei Fan		

Edit History

Aug 4, 2020 - Document created

Aug 6, 2020 - Fixed and added all current sections

1) Product summary

With the increasing number of SFSU students and the decreasing availability of living spaces, finding a place to live in San Francisco has never been more critical for students. Our website FreshWater Rentals aims to provide users the ability to find housing opportunities in our current fast paced and competitive market. Registered users on our site can advertise living spaces available for sale or for rent. Other features include a detailed map to campus.

Name of Product: Freshwater Rentals

Our system will allow users to:

- Search and filter listings by Zip code, Rent / Buy, Home Type, pets allowed.
- Post listing.
- Once a user has selected a specific listing, they can see the home's address, price, photos, description and the home location on Google maps.
- Users shall message to landlords.
- Review their listing and messages on the dashboard.
- Sign up by email address, username and password
- Login by username and password

Our system will allow administrators to:

- Review posting by MySQL Workbench.
- Delete malicious or irrelevant posting by MySQL Workbench.
- Delete misappropriate message MySQL Workbench.

Our website URL: **http://freshwater.city/**

2) Usability test plan: Search

Test Objectives (what is being tested and why)

On this usability test we shall test the functionality of the homepage and its main feature: Search.

The purpose of the Usability Test is to ensure simple user experience and allow users to find a result listing quickly. This will ensure the user accomplishes the tasks he/she/them wants to achieve by using our application, and that we provide the right functionalities for this to happen, from loading the page, exploring different listings, narrowing down a search result and exploring results of interest to the user.

Test background and setup
(System setup, starting point, who are the intended users, URL of the system to be tested ad what is to be measured)

- System setup
  The tester's device will connect to the Internet and access to the following accepted internet browser: Google chrome, Firefox or Safari.

- Starting point
  The tester will begin at the site homepage: http://freshwater.city/, where it provides a search text box and button.

- Intended users
  The intended users are property owners in the San Francisco Bay Area and college students and faculty from San Francisco State University.
  Users will be considered the new to the site.

- URL of the system: http://freshwater.city/.

- Measured criteria
  The tester is satisfied with the search result. This is evaluated from a post test questionnaire (Likert tests).

Usability Task description:

Tasks:

- Login to the freshwater.city and explore the current postings.
- Search for a specific type of house/apartment.
- Sort postings by your main concern when looking for a place, price, zip code, bedrooms, distance from sfsu, etc.
- Mix sorting and filters as you may wish.

We would measure the effectiveness of these tasks by the percentage of people who completed the search task in defined time.

We would measure the efficiency of these tasks by

- Time: average time to compete search task;
- Effort: number of clicks;
- Content/Design: number of screens, clicks and pages of instructions.

Usability Questionnaire

I was able to successfully locate a desired type of housing. (Click one)

__ Strong Disagree __ Disagree __ Neither Agree or disagree
__ Agree __ Strongly Agree

I was able to find the specific listing quickly. (Click one)

__ Strong Disagree __ Disagree __ Neither Agree or disagree
__ Agree __ Strongly Agree

I was able to find the listing for the zip code I was searching for (Click one)

__ Strong Disagree __ Disagree __ Neither Agree or disagree
__ Agree __ Strongly Agree

My comments:

_____

3) QA test plan

Test Object

- The object of the QA test is to validate the main functions, i.e. Search.
- The tester shall be given specific instruction to follow, and the result shall be recorded as a PASS/FAIL in each of the individual cases.

Hardware and Software setup

- The tester's devices (PC, cellphone, IPad) will connect to the Internet via Ethernet, Wi-Fi, Cellular, etc.
- The internet connection speed shall be larger than 2.5MBps.
- The tester's devices can access the following accepted internet browser: Google chrome, Firefox or Safari.
- Finally, the tester shall open the website link: http://freshwater.city/, and start the test.

Feature to be tested

This test mainly focuses on the Search feature. The database has 5 properties information by August 6, 2020.

QA test plan

Test #	Test Title	Description	Test Input	Expected Correct Output	Test Results
1	Search field on Each page	Test % like in search field	Type "nice" in search field	Get 2 results, both of them have "nice" in their description	**Pass**
2	Filters on Home Page	Test different combination of the filters available	Select "House" and "Rent" filters	Get 1 result, which is a House to Rent	**Pass**
3	Search and Filter combination on Home Page	Test different combination of the filters and search	Type "nice" in search field and select "Apartment" to "Rent" in filter	Get 1 result for Apartment to Rent, which has "nice" in the description	**Pass**

4	Sort Functionality Test	Test different sorting options available	Select “Distance to SFSU” sorting option	All the 5 results should be sorted by Distance to SFSU, shortest distance listing being first	**Pass**
5	Listing more info	Test one of the listings to check out more info about that listing	Click on “Apartment to Rent” listing with 800 sqft	The more information about this listing should open with correct info. About this listing such as Rent($3000/mo), Distance(1.0 mile from SFSU), map, street address	**Pass**

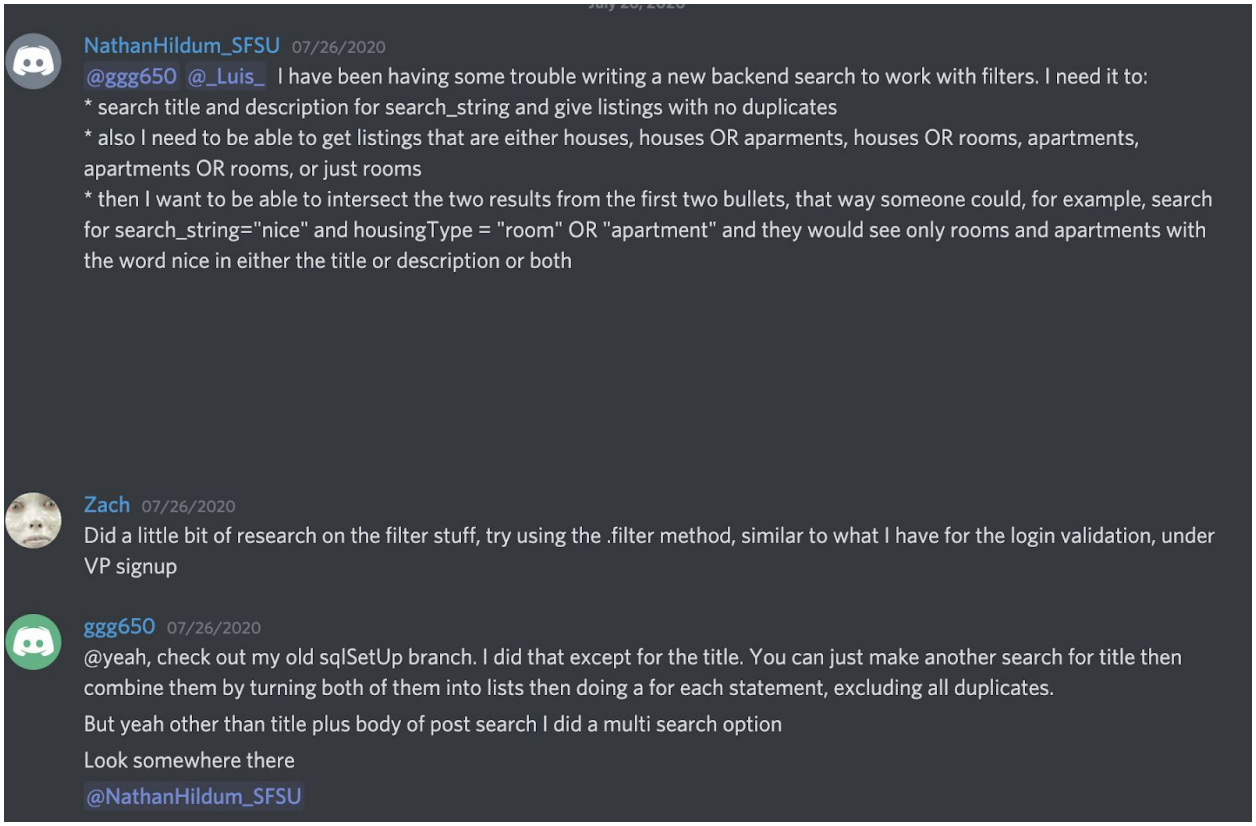
4) Code Review:

The chosen code style is modular programming.
The code to be reviewed about the function that lets user's search by listing address, city and zip code.

```
# searches listings for search_string in title and desc
# and also applies filter options
def backendSearch(search_string=None, housingType=None, sellOrRent=None, petsAllowed=None):
    results = search_title_and_desc(search_string)
    results = filter_in_list(
        results, Listings.houseType, housingType)
    results = filter_in_list(
        results, Listings.sellOrRent, sellOrRent)
    results = filter_in_list(
        results, Listings.petsAllowed, petsAllowed)
    return results

# searches title and description for search string
# and if empty, returns all Listings
def search_title_and_desc(search):
    if search:
        search = "%{}%".format(search)
        results = Listings.query.filter(or_(
            Listings.description.ilike(search),
            Listings.title.ilike(search)))
    else:
        results = Listings.query
    pprint.pprint(results)
    print(type(results))
    return results

# returns sqlalchemy BaseQuery object
# that has been filtered to include only records/rows that have values in list
def filter_in_list(results, table_column, list_):
    if list_:
        results = results = results.filter(table_column.in_(list_))
    pprint.pprint(results.all())
    print(type(results))
    return results
```


NathanHildum_SFSU 07/26/2020
@ggg650 @_Luis_ I have been having some trouble writing a new backend search to work with filters. I need it to:
* search title and description for search_string and give listings with no duplicates
* also I need to be able to get listings that are either houses, houses OR aparments, houses OR rooms, apartments, apartments OR rooms, or just rooms
* then I want to be able to intersect the two results from the first two bullets, that way someone could, for example, search for search_string="nice" and housingType = "room" OR "apartment" and they would see only rooms and apartments with the word nice in either the title or description or both
Zach 07/26/2020
Did a little bit of research on the filter stuff, try using the .filter method, similar to what I have for the login validation, under VP signup
ggg650 07/26/2020
@yeah, check out my old sqlSetUp branch. I did that except for the title. You can just make another search for title then combine them by turning both of them into lists then doing a for each statement, excluding all duplicates.
But yeah other than title plus body of post search I did a multi search option
Look somewhere there
@NathanHildum_SFSU

5) Self-check on best practices for security

Assets we are protecting are:

- User Information via
    - Encrypted Passwords
    - Website Restriction
- General Server Protection via
    - Mysql injection attacks
    - Protection from File uploads

For password encryption we are using Python Flask's Security framework along with Bcrypt for hashing. This process is dual encryption, one on the frontend side and one on the backend side with of course the second hashing of the password stored in the Mysql database. Passwords must be at least 6 characters long upon creation and email must end with @sfsu.edu, @mail.sfsu.edu or @sfsu.com. Also to control the flow of data from authorized users we are deploying session handling that tracks users behavior and restricts users without the correct credentials to certain sites.

For mysql injection attacks we are using sqlAlchemy framework to protect from any for sql injection attack.

For a file uploading, we limit the size and file type to only 'png', 'jpg', 'jpeg', 'gif' to limit malicious script being sent to the server.

6) Self-check: Adherence to original Non-functional specs – performed by team leads

Original Non-Functional Specs	Status
1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO).	**DONE**
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers.	**DONE**
3. Selected application functions must render well on mobile devices	**DONE**
4. Data shall be stored in the team's chosen database technology on the team's deployment server.	**DONE**
5. No more than 50 concurrent users shall be accessing the application at any time.	**DONE**
6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.	**DONE**
7. The language used shall be English (no localization needed)	**DONE**
8. Application shall be very easy to use and intuitive.	**DONE**
9. Google analytics shall be used.	**DONE**
10. No e-mail clients shall be allowed. Interested users can only message to sellers vis in-site messaging. One round of messaging (from user to seller) is enough for this application	**DONE**
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.	**DONE**
12. Site security: basic best practices shall be applied (as covered in the class) for main data items	**DONE**
13. Media formats shall be standard as used in the market today	**DONE**

14. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development	**DONE**
15. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Summer 2020. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application)	**DONE**