Final Project for SW Engineering Class
CSC 648-848 Summer 2020

Team 04

Online Real Estate Marketplace:
Hillow

Wiam Boumaazi, Team Lead, Github master
(wboumaazi@mail.sfsu.edu)
Karina Abad, Backend Lead
Utsav Bhatta, Backend Developer
Katie Kennedy, Backend Developer
Jaren Lynch, Frontend Lead
Mohammadrehan Kanuga, Backend Developer

URL of the Demo
:http://ec2-54-219-200-213.us-west-1.compute.amazonaws.com/index.php

08/06/2020

Table of Contents

2. Product Summary:

Name of product - Hillow Rental

Itemized List:

Our system will allow unregistered users to:

- View all the listings, and search with no parameters, types or zip code.
- Search the listings by type, price and distance to SFSU.
- Register, and login with accepting the terms and conditions to use more features.

Our system will allow registered users to:

- Login and post a listing on the site.
- contact landlords, view their messages and the listings.
- logout if they were logged in.

Our system will allow admin to:

- contact landlords.
- approve a listing before they upload on the site.

Our system will have posts that:

- Contain the type of the listing (Apt, Room, House)
- Contain the addresses, prices, and the distance from the campus of the listing

Our system requires an email ending with (@sfsu.edu) is required for registration and Search text entry shall be validate to allow only up to 40 alphanumeric characters (letters and numbers only)

URL: http://ec2-54-219-200-213.us-west-1.compute.amazonaws.com

3. Milestone Documents -M1-M4:

Milestone 1

Software Engineering CSC 648/848
Summer 2020
Team 4 Milestone 1
Initial Proposal

Hillow App

Developed by Team 4:
Wiam Boumaazi, Team Lead (wboumaazi@mail.sfsu.edu)
Karina Abad, Frontend/Backend Developer
Jaren Lynch, Frontend Lead
Katie Kennedy, GitHub Master/Backend Lead
Utsav Bhatta, Backend Developer
Mohammedrehan Kanuga, Backend Developer

Submitted for review: June 25, 2020

Submission date	Date revised
06 /25/ 2020	07/02/2020

1. Executive Summary

Our new app called Hillow is a solution to a constant request from countless SFSU students and faculties who often struggle to find a suitable house with their desired amenities, as there are too many places to search for rental units and too few services that cater directly to students' needs. Hillow is the solution these people have been looking for, allowing them to easily find housing with minimal time wasted searching. Hillow allows anybody who owns an apartment or house to advertise for rent. Our main goal is to put an end to all students' problems with the rental market. We strive to connect students with rare housing opportunities in the closest neighborhoods to SFSU, where people with the same interests can become roommates. Hillow is the best available market option for students to achieve healthy living and make progress towards their educational goals. Our website takes zero commission and students' satisfaction is always our paramount goal.

Hillow is an online platform exclusively for SFSU students and faculties to rent a house and apartments. Hillow helps students to find the best match to their needs. Hillow offers spacious, pet-friendly, and safe apartments or townhouses. Any SFSU student who is looking to rent an apartment can create an account and start using our variety of housing-related services. SFSU students can search for apartments and find out just in seconds about the availability and location of the apartments on hillow. While there are multiple online platforms in the market providing similar service, hillow is the only online rental platform where houses and apartments are available exclusively for SFSU students and faculties. For years, SFSU students have struggled to find a reliable app that makes their lives easier and promises to provide them with only the best housing choice available. Hillow has its special map that shows the location of all buildings and helps you find the route to SF State.

Hillow will keep users up-to-date if any new locations are available that start leasing. One of the best features that hillow provides is to connect students by their majors, allowing students to learn better if their roommates are also studying the same material as they are because it creates a good learning environment. We believe in user appreciation and all of our services will work accordingly; from searching through hillow to finalizing a lease we will be there to support and help you schedule a meeting with the owner and sign a lease to reduce unnecessary waste of time and make your life earlier and get your work done more quickly. To avoid potential problems, hillow will only allow SFSU students as its users and it wont let any other users sign up and use it.

Hillow is a startup led by 6 SFSU students who work towards their common goal- housing for advancement of students. Being SFSU students, we know the struggle and preference of students and why it is important to find the best place that can help you concentrate on your studies. Our team always works for the betterment of SFSU students and we will have more and useful functionalities developed and added to hillow in future.

2. Personas and use cases

Lucas

Lucas is an SFSU student. He is looking for rent but he is tired of browsing on unorganized networks to find available places. Lucas wants to use a more organized app that will allow him to find rent easily. Lucas had been scammed many times and he lost a lot of money and energy looking for a new place. Lucas want an application that provides the following services:

- An organized application that allows the user to search easily for rent.
- An application that verify the user authentication and avoid scam.
- An application that verifies the location and the availability of the offer.
- An application that provides useful information about the place.

Sarah

Sarah is an SFSU faculty. Sarah spends more time on the campus but also likes to go back home once in every 5 hours to rest, eat or shower. Sarah wants a place very close to campus so she won't waste time in the commute. Sarah prefers to live with SFSU faculties only. She feels more comfortable living with other SFSU faculties. Sarah likes to eat in restaurants and she would like to find the closest grocery places to her place. Sarah wants an application that will allow her to:

- To find the closest available spots to the campus of SFSU.
- To find the closest restaurants and grocery markets to that place.
- To make sure that all tenants in the house are SFSU faculties.

Sam

Sam is an SFSU student. Sam is an independent person who goes to school and work at the same time to cover all her expenses. Sam does small jobs and she works only part time. Sam has some financial struggles and doesn't want to spend too much money in rent. Sam would like to find the cheapest available places for rent. Additionally, Sam likes to spend her free time at home and she also likes to do her homework at home. Sam would like a place that doesn't have a lot of tenants. Sarah wants and app that allows her to:

- To find the cheapest available place.
- To find a place with two or three tenants maximum.
- To find information about tenants living in the house.

Carmen

Carmen is an employee of Hillow who works as a web admin. Her job is to approve users' property postings before they are live on the Hillow website for users to view, making sure language and all other post components comply with Hillow's policy against inappropriate or irrelevant content. Carmen's admin access does not allow her to edit posts, only approve or deny them for listing on Hillow. To streamline Carmen's job, the Hillow web app must allow designated admin users the following functions:

- A complete listing of all the new posts awaiting a decision on its approval status.
- Two buttons allowing the admin to approve or deny each listing.
- A method of deleting listing posts that no longer comply with the company policy or that are expired.

Jeff

Jeff is a house owner, Jeff is interested in renting his house to SFSU students or faculties only. Jeff has been scammed many times and would like a platform more secured and controlled where he can share pictures about his house and a mini description. Jeff would like an app with the following features:

- A secured application that controls the use of data and posts.
- An application allowing rent to SFSU students and faculties only.

Use case for case 5: With our app, the landlords and house owner will be more satisfied with the security measures that we are taking in order to secure their data. Our application has admins who control the use of the site and can identify any illegal use or scams. Our admins verify the identity of the clients before completing any operation and we require the SFSU email as a verification to that.

Use cases

Use case for case 1: Our application will help Lucas to avoid scam. Lucas will be able to see pictures of the place to check if it fits his needs or not before he can go check it in person. Lucas can make sure that the users of the app had been authenticated. Lucas does not needs to create an account in order to be able to make a search and to choose a place. Lucas will be able to contact the house owner or the seller via his phone or email that we will provide. Lucas will be able to check all the information needed about the place he wants to rent. When the house owner is posting a new available place for rent, he will be asked to provide information such as: how many bedrooms are in the house, how many bathrooms, is the place pet friendly? How many tenants are there? Is the rent month-to-month or should the tenant sign the lease?

Use case for case 2: With our app, Sarah can be able to filter the choices for an available place based on the shortest distance to the campus. Sarah will be asked to enter a max distance between the campus and available places. Based on the distance she entered, the places will be filtered allowing her to choose from

the available places with lowest distance to the campus. Sarah can make sure that all the tenants in the house are SFSU faculties. Sarah can check information about the closest restaurants and grocery stores to each place. The Landlord will be asked to provide this information while making a new posting.

Use case for case 3: With our app, Sam will be able to create an account and check for available rent. Sam will be able to choose from the cheapest available places. The app will ask the student to provide a maximum budget they will be able to pay for rent. Based on this selection, the app will filter the best results for the user and allow him to choose from the places with a price lower than the one selected. In our app, the place posted will also have a short description about the tenants who live in the house and it will be updated in case the tenant left the place.

Use case for case 4: With Hillow's special functionality for employee web admins, employees will have an easy way to quickly respond to post requests from users, as users quickly become frustrated while having to wait for approval. Because it is important not to misplace requests from users, all requests for posting (or edits) will be on one page with one-click approval (or denial). Admins will also need an option to delete an existing post on Hillow if it must be removed for some reason. Admins have no need to edit a posting, so there is no option integrated in Hillow's design to do so, keeping the admin's options minimal to reduce mistakes in the post approval system.

3. Main Data Items & Entities

List of Entities:

1. Unregistered User
    a. Account (optional)
        i. Username or Email
        ii. Password
2. Registered User
    a. Account
        i. Username or Email
        ii. Password
3. Admin
    a. Username or Email
    b. Password
    c. Permissions
4. Seller
    a. Account
        i. Username or Email
        ii. Password
    b. Property Listing

Active Entities:

1. Registered Users
    a. Registered users are able to browse the listings within the site, as well as message sellers on each listing that is posted.
    b. Without restrictions, they have access to the database of listings within the site.
    c. Registered users are able to receive notifications on listings that they've saved, new listings posted within price range and location specified, as well as if a listing saved has been taken down.
        i. Username: will serve as an identifier for the user, better interpretation than a primary key number, as usernames are easier to remember and self assigned.
        ii. Password: to log into the system and access specific files, listings, private messages within their account--security and marketing .
2. Unregistered Users
    a. Users are able to browse the listings within the site without an account. They are also able to message sellers on each listing that is posted.
    b. Without restrictions, they have access to the database of listings within the site.
3. Seller
    a. Sellers need an account to post in order for the site to efficiently manage all postings and for verification.
    b. Sellers are able to post listings, talk to potential clients, update their listing, remove listings, and receive notifications that they've saved or updates on the site.
        i. Username: will serve as an identifier for the user, better interpretation than a primary key number, as usernames are easier to remember and self assigned
        ii. Password: to log into the system and access specific files, listings, private messages within their account--security and marketing
        iii. Listing: for posting, deleting, updating property listing with images, descriptions, content, location, pricing and other necessary information needed to list property.
4. Admin
    a. Admin takes care of the well being of the site and making sure that listings are accurate, correct and not fraudulent. They're responsible for making sure that everything on the site is verified and kept tidy in terms of information. They must also approve each post before it goes online.
        i. Username: will serve as an identifier for the user, better interpretation than a primary key number, as usernames are easier to remember and self assigned, but will be assigned due to admin use only
        ii. Password: to log into the system and access specific files, listings, private messages within their account--security and marketing

Data Items:

1. Property Listing
    a. Listing id: for mapping the listings across the site and making sure each one is accessible uniquely.

    b. Description: provides the users a summary of the post, including only important details and a specific number of characters i.e. Apartment for rent 2 miles away from SFSU
    c. Location: map of the property within a 10 mile radius all around
    d. Pricing: provides pricing for property that are in USD, and in time frames of monthly, 3 months, 6 months, 8 months, yearly, to name a few.
    e. Date Availability: provides the exact date that will enable the buyer to move in.
    f. Dates for Showing: a range of dates that lists for open houses.
2. Map
    a. Longitude and latitude components: provides a realistic range for users to be able to navigate the location within the area in proportion
    b. GUI: to be able to depict clearly the proximity of the property among other establishments, neighborhoods, homes, schools.

Data Structures:

1. Property listing: list of properties within a specific neighborhood
2. Neighborhoods: list of neighborhoods available to search from within the site
3. For Sale: list of properties for sale within a specific neighborhood
4. Sellers: list of sellers within database
5. Successful Transactions: List of successful transactions and connections made within the site, for data purposes that can be used for marketing and advertising.
6. Blocked: List of users that affected the site experience negatively.

4. Functional Requirements

1. Unregistered users shall be able to register.
2. Unregistered users shall have viewing access to all listings.
3. Unregistered users shall be able to sort/search listings.

4. Registered users shall be able to do anything an unregistered user can do.
5. Registered users shall be able to create a listing on the site.
6. Registered users shall be able to filter data based on the distance and budget.
7. Registered users shall be able to filter data based on number of rooms.
8. Registered users shall be able to message sellers.
9. Registered users shall be able to message a seller about a specific listing.
10. Registered users shall be able to view their own historical activities.

11. Registered users shall be able to favorite a listing.
12. Registered users shall be notified of changes to favorited listings.
13. Registered users shall be able to change their information

14. Admin Shall be able to contact all customers using this site.
15. Admin Shall be able to inspect all posts.
16. Admin Shal be able to ban registered users.
17. Admin Shall be required to check and approve or remove registered users' listings before they upload on the site.
18. Admin Shall be able to view historical data such as searches, making payment, listings, etc. from all registered users.

5. Nonfunctional Requirements

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).
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
3. Selected application functions must render well on mobile devices
4. Data shall be stored in the team's chosen database technology on the team's deployment server.
5. No more than 50 concurrent users shall be accessing the application at any time
6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
7. The language used shall be English (no localization needed)
8. Application shall be very easy to use and intuitive.
9. Google analytics shall be used
10. No email clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.

12. Site security: basic best practices shall be applied (as covered in the class) for main data items
13. Media formats shall be standard as used in the market today
14. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
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).

6. Competitive Analysis

Our website offers features specifically designed to aid SFSU students in their search for housing, while staying competitive in the market with well established rental companies. Our app offers features that allow students to search for rentals within their budget and desired distance from SFSU's campus, which matter most to students when choosing a rental property. We also aim to stay transparent by offering upfront pricing with no hidden fees and taxes to avoid surprising students. We are making the rental procedure easy by allowing the listing from property owners and also an option to sublease the property with another SFSU student. This is our unique feature that stands out with the rest of the competitors.

Competitors / Features	Zillow	Trulia	Airbnb	Our App
Search by budget	Yes	Yes	No	Yes
Search within SFSU radius range	No	No	No	Yes
Sublease option	No	No	No	Yes
Additional fees and charges	Yes	Yes	Yes	No
Exclusive to SFSU students	No	No	No	Yes

7. Tools & Frameworks

Server Host: AWS EC2

Operating System: Ubuntu 20.04

Database: MySql 8.0

Server-Side Language: NodeJS 12.14.0

Additional Technologies:

IDE: Microsoft Visual Studio

API: Google Maps

Front-end code languages: React, Bootstrap

Supported browsers: Chrome, Firefox

8. Team & Roles

Wiam Boumaazi: Team lead.

Karina Abad: Frontend/Backend developer.

Katie Kennedy: GitHub master/Backend lead.

Jaren Lynch: Frontend lead.

Utsav Bhatta: Backend developer.

Mohammedrehan Kanuga: Backend developer.

9. Checklist

Task	Status

All team members are engaged and attending ZOOM sessions when required.	DONE
Team found a time slot to meet outside of the class.	DONE
GitHub master chosen.	DONE
Team decided and agreed together on using the listed SW tools and deployment server.	DONE
Team is ready and able to use the chosen back and front end frameworks (and those who need to learn are working on learning and practicing).	DONE
Team lead ensured that all team members read the final M1 and agree/understand it before submission.	DONE
Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents, etc.)	DONE

Milestone 2

SW Engineering CSC648/848
Summer 2020

Milestone 2
Online Real Estate Marketplace:
Hillow

Developed by Team 4:

Wiam Boumaazi, Team Lead (wboumaazi@mail.sfsu.edu)
Karina Abad, Front End Developer
Utsav Bhatta, Back End Developer
Katie Kennedy, Back End Lead
Jaren Lynch, Front End Lead
Mohammadrehan Kanuga, Back End Developer

07/08/2020

Submitted for review:

Submission date	Date revised
07/08/2020	07/27/2020

Priorities:

Priority 1:

Unregistered users shall be able to register.

1. Unregistered users shall have viewing access to all listings.
2. Unregistered users shall be able to sort/search listings.
   3.1. Unregistered users shall be able to sort/search based on the shortest distance
   3.2. Unregistered users shall be able to sort/search based on the prices
   3.3 Unregistered users shall be able to sort/search based on the number of rooms
3. All users shall be able to create a listing on the site.
4. All users shall be able to sort/search listings.
   5.1. All users shall be able to sort/search based on the shortest distance
   5.2. All users shall be able to sort/search based on the prices
   5.3 All users shall be able to sort/search based on the number of rooms
   5.4 All users shall be able to message landlords or sellers
5. Admin Shall be able to contact all customers using this site.
6. Admin Shall be able to inspect all posts.
7. Admin Shall be required to check listings
8. Admin shall approve or remove users' listings before they upload on the site.

Priority 2:

1. Unregistered users shall be able to see information about the neighborhood.
2. Unregistered users shall be able to contact the admin.
3. Unregistered users shall be able to rate the website.
4. Unregistered users shall be able to share their user experience in the website.
5. Registered users shall be able to message sellers about a specific listing.
6. Registered users shall be able to change their information.

7. Admin Shal be able to ban registered users.

Priority 3:

1. Registered users shall be able to view their own historical activities.
2. Registered users shall be able to favorite a listing.
3. Registered users shall be notified of changes to favorited listings.
4. Admin Shall be able to view historical data such as searches, listings, etc. from all registered users.

2. List of Main Data Items & Entities

List of Entities:

1. Unregistered User
    a. Account (optional)
        i. Username or Email
        ii. Password
2. Registered User
    a. Account
        i. Username or Email
        ii. Password
3. Admin
    a. Username or Email
    b. Password
    c. Permissions

Active Entities:

1. Registered Users
    a. Registered users are able to browse the listings within the site, as well as message sellers on each listing that is posted.
    b. Without restrictions, they have access to the database of listings within the site.
    c. Registered users are able to receive notifications on listings that they've saved, new listings posted within price range and location specified, as well as if a listing saved has been taken down.

        i. Username: will serve as an identifier for the user, better interpretation than a primary key number, as usernames are easier to remember and self assigned.
        ii. Password: to log into the system and access specific files, listings, private messages within their account--security and marketing .
2. Unregistered Users
    a. Users are able to browse the listings within the site without an account. They are also able to message sellers on each listing that is posted.
    b. Without restrictions, they have access to the database of listings within the site.
3. Admin
    a. Admin takes care of the well being of the site and making sure that listings are accurate, correct and not fraudulent. They're responsible for making sure that everything on the site is verified and kept tidy in terms of information. They must also approve each post before it goes online.
        i. Username: will serve as an identifier for the user, better interpretation than a primary key number, as usernames are easier to remember and self assigned, but will be assigned due to admin use only
        ii. Password: to log into the system and access specific files, listings, private messages within their account--security and marketing

Data Items:

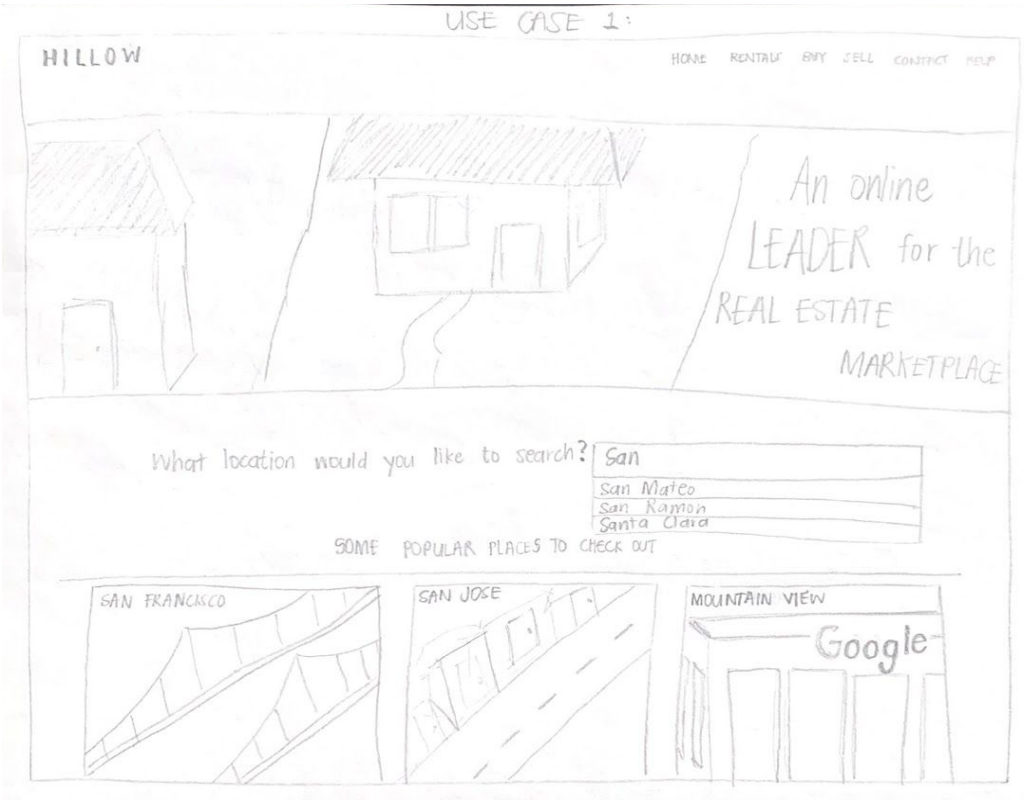
1. Property Listing
    a. Listing id: for mapping the listings across the site and making sure each one is accessible uniquely.
    b. Description: provides the users a summary of the post, including only important details and a specific number of characters i.e. Apartment for rent 2 miles away from SFSU
    c. Location: map of the property within a 10 mile radius all around
    d. Pricing: provides pricing for property that are in USD, and in time frames of monthly, 3 months, 6 months, 8 months, yearly, to name a few.
    e. Date Availability: provides the exact date that will enable the buyer to move in.
    f. Dates for Showing: a range of dates that lists for open houses.
2. Map
    a. Longitude and latitude components: provides a realistic range for users to be able to navigate the location within the area in proportion
    b. GUI: to be able to depict clearly the proximity of the property among other establishments, neighborhoods, homes, schools.

Data Structures:

1. Property listing: list of properties within a specific neighborhood
2. Neighborhoods: list of neighborhoods available to search from within the site
3. For Sale: list of properties for sale within a specific neighborhood
4. Sellers: list of sellers within database
5. Successful Transactions: List of successful transactions and connections made within the site, for data purposes that can be used for marketing and advertising.
6. Blocked: List of users that affected the site experience negatively.

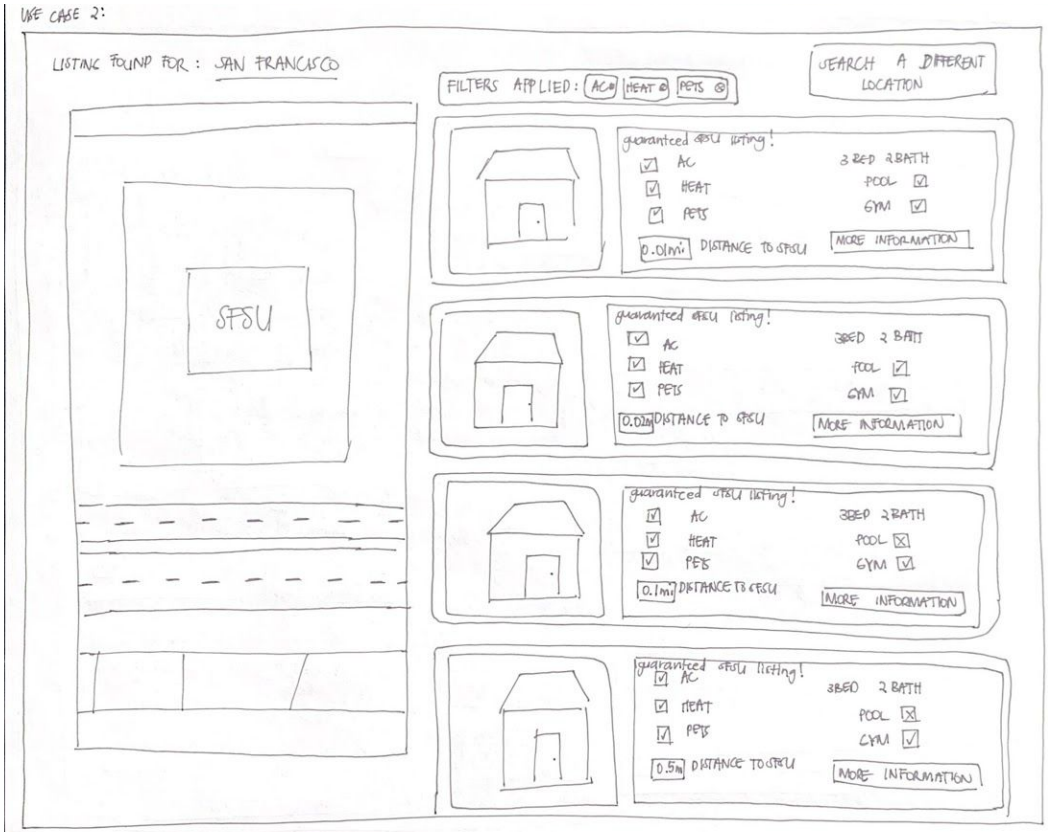
3. UI Mock-ups and Storyboards

Use case 1: Homepage for the site, where a search bar is available to access database of listings

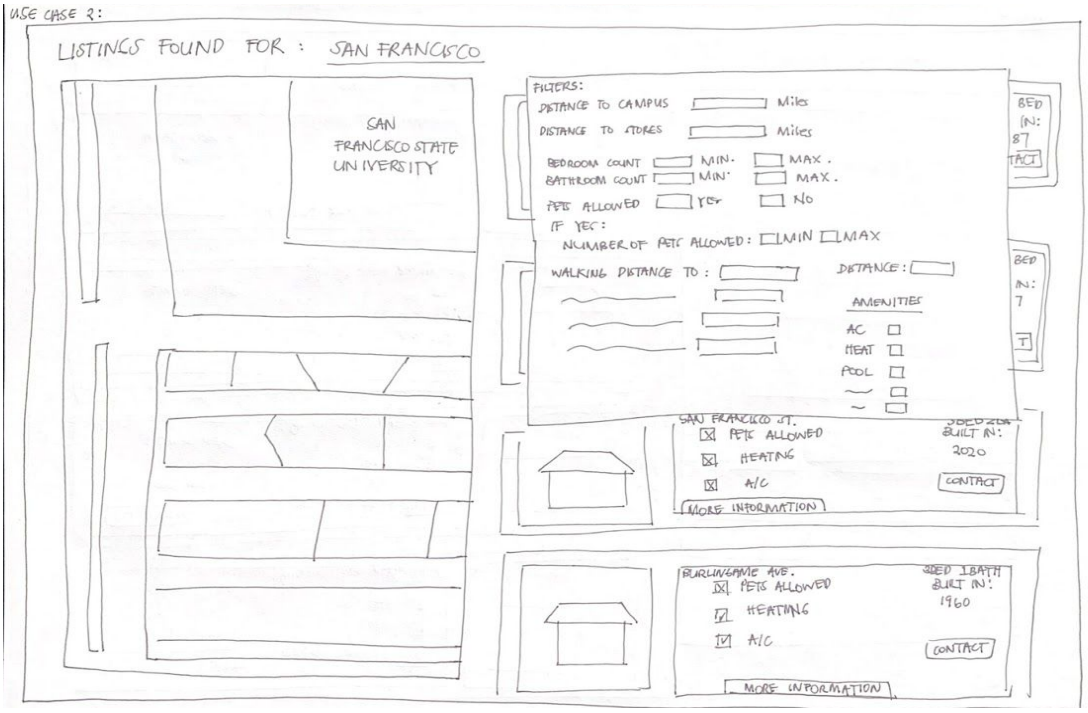


Use case 1: Design of page that specifically populates user's desired location, map and listings are shown on the screen, as well as option to search a different location

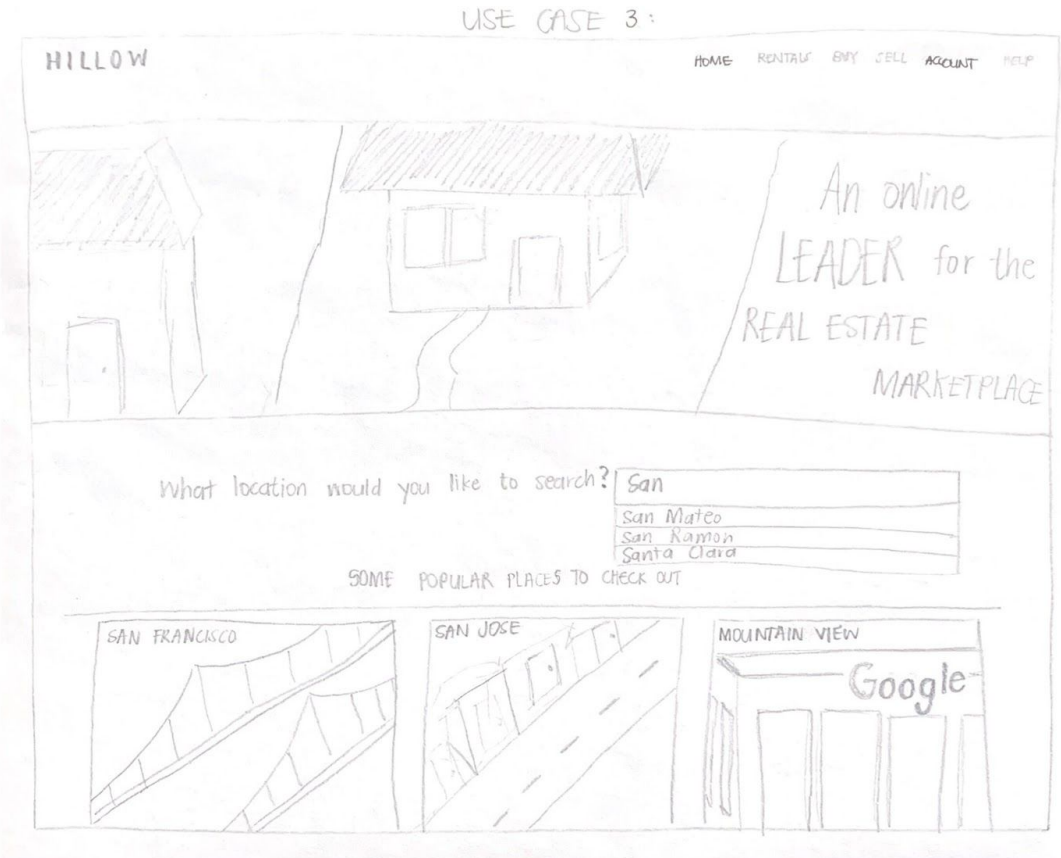
Use case 2: Design of page that specifically populates user's desired location, map and listings are shown on the screen, as well as option to search a different location



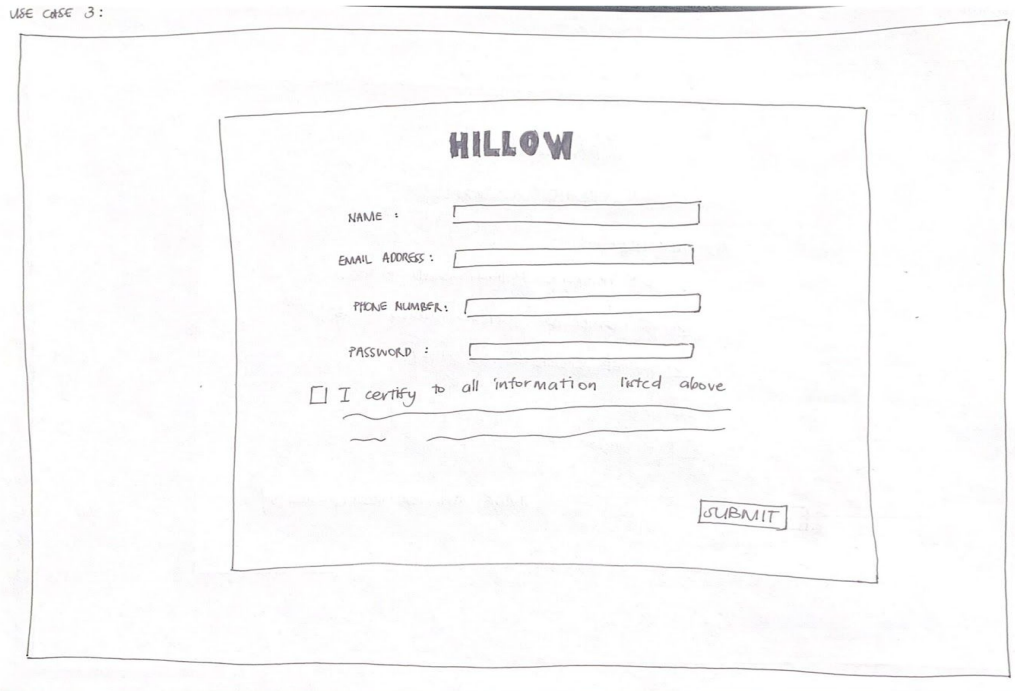
Use case 2: Design of page that specifically populates user's desired location with filtered results



Use case 3: Design home page with account setting to create or access account

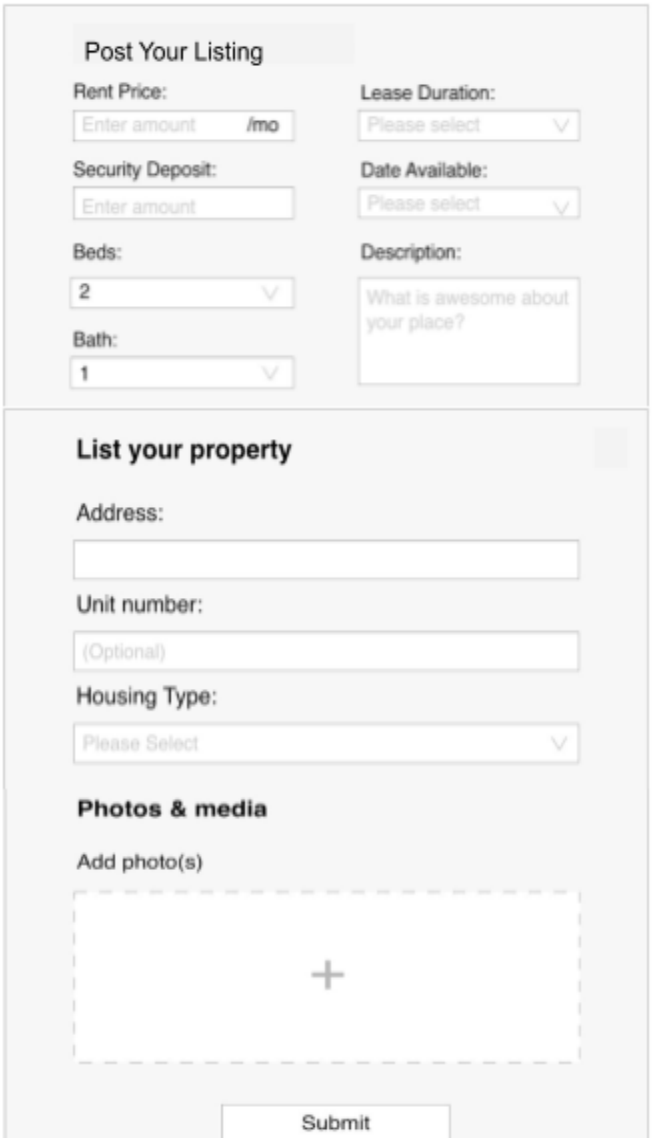


Use case 3: Design of page to create an account

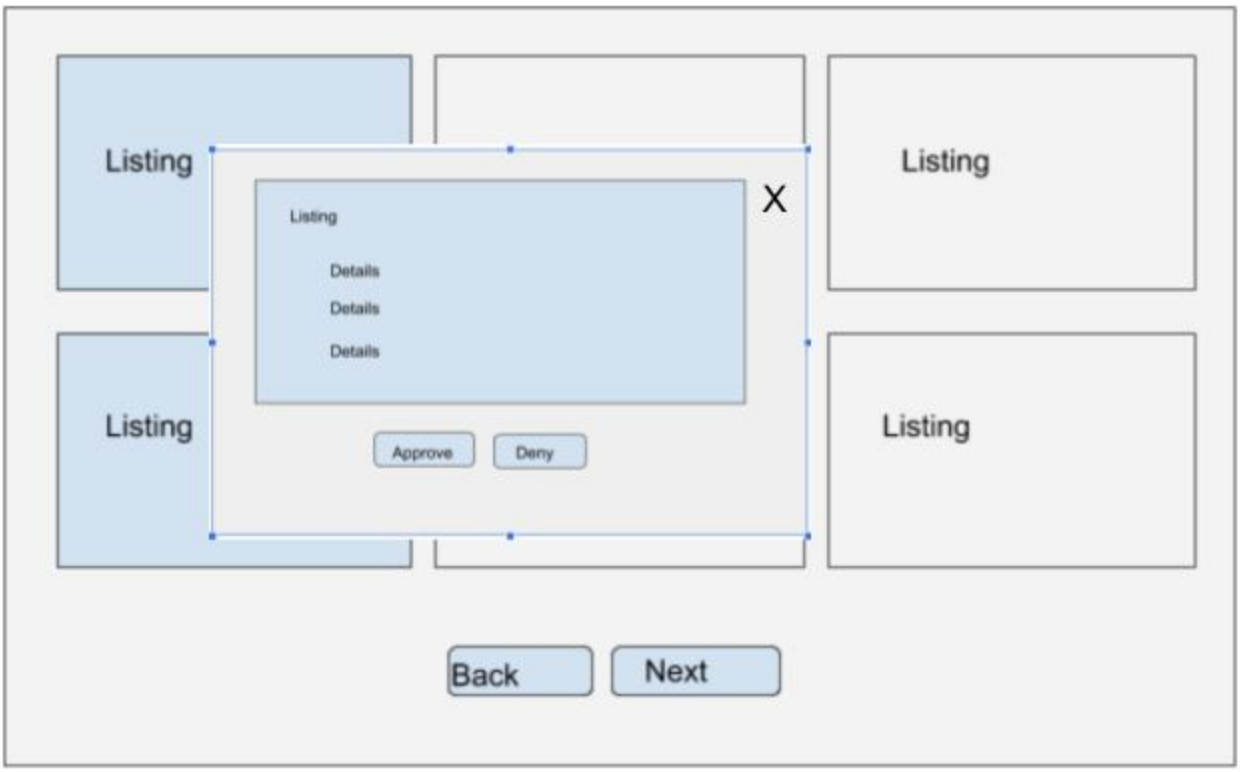


Use case 3: Design of page that will show benefits of having an account

Use Case 4: Users can create listings (1/3)



Use Case 5: Admin approve/deny listings


Listing
Listing
Listing
Details
Details
Details
X
Listing
Listing
Approve
Deny
Back
Next

4. High Level Architecture & Database Organization

1. **DB organization: Describe the main database schema/organization (high level), e.g. list main DB tables (e.g. their titles) and items in each DB table (check instructors' suggestions and class slides on architecture). Make sure the titles and var. names are in easy to understand plain English and consistent with data definitions in Section 1 above.**

Schema to be implemented in our DB:

CREATE TABLE user(
userid INT(10),
userEmail CHAR(15), userPassword CHAR(20)NOT NULL, Blocked BOOLEAN,
PRIMARY KEY (userid));

CREATE TABLE listing(
listingid INT(10),
listingtitle CHAR(15),
listingseller CHAR(15), listingdescribtion CHAR(15),
listingNo CHAR(15),
usepermission FLOAT(5,2) NOT NULL, condition char(15),
listingfiledata BLOB('long'), listingthumbnaildata BLOB('short), pending BOOLEAN,
PRIMARY KEY(listingid),
FOREIGN KEY (posterid) REFERENCES user(userId));

CREATE TABLE Messages(
messageId INT(10),
title CHAR(20)
date DATE(30),
Time TIME,
body CHAR(144),
read BOOLEAN,
FOREIGN KEY (fromId) REFERENCES user(userId));
FOREIGN KEY (toId) REFERENCES user(userId));

CREATE TABLE Blocklist(
FOREIGN KEY (userId) REFERENCES user(userId));

2. **Media storage: Decide if images and video/audio will be kept in file systems or in DB BLOBs (decision on file vs. BLOBs must be made by the end of M2). Describe any other special data format requirements like for video/audio/GPS etc.**

   -We will use BLOBs to store thumbnails, images, and videos. We will be using GPS location using the GPS Exchange Format. For video streaming we may or may not do it depending on the timeline we have.

3. **Search/filter architecture and implementation: what will be the alg/SW for search; how will you organize search items for the user; what DB terms will be searched, how it will be coded and organized in the DB (check instructors' suggestions in the class. OK to use SQL and %like and if you plan to use that simply say so).**

   -To search, Frontend will make an HTTP call to the Backend Server after receiving user input as keywords by transferring the keywords to the Backend. The DB's linked to the Backend. The backend must retrieve the word(s) after receiving the request from Frontend, and perform a Query to the Database. The returned result(s) will then be sent to the Frontend by backend as a JSON answer. The Frontend will then parse Backend 's answer in a helpful way for the user to view.

4. **Your own APIs (if any): Describe and define at a high level any major APIs that you will create other than standard ones provided by tools and frameworks you use.**

API	Input	Output	Description
user_signup	email_id, password	Registration status : Success/Failure	Takes in the input arguments, validates it and if it passes the validation then add it to the user table
block_user	email_id	Status: Success/Failure	Takes and adds the email id to blocklist table
upload_listings	All attributes in listings table	Status: Success/Failure	Takes and adds input attributes to the listings table
send_message	All attributes in messages table	Status: Success/Failure	Takes and adds input attributes to the messages table

Search APIs

API	Input	Output	Description
email_validation	email_id	Boolean	Checks if the user table already includes email id
user_login	email_id, password	Login status: Success/Failure	Takes in input arguments, validate it against the user table and return the status
search_listing	Search string	Array of listings	Takes in the search string and looks for it in listings table
search_video	Search string	Array of listings (video)	Exclusive search only for videos
search_audio	Search string	Array of listings(audio)	Exclusive search only for audios
get_poster	listing_id	Seller details	Takes in the listing_id and retrieves the details of the person who posted it.
get_user_details	user_email	User details from table User	Takes in user_email and retrieves all the details of the user from table "User"
get_listings	user_email	All listings user has posted	Takes in user_email and retrieves all the listings posted by that user
get_message	user_email	All messages for that particular user ID	Takes in the user_email and retrieves all the messages for that user

5. Describe any significant non-trivial algorithm or process if any (like rating, ranking, automatic prioritizing of items etc.)

-We will build a function that will turn the password into hash before storing it to our database.
-Filtering and prioritizing search items will be done by manipulating SQL query.

. 6. If you have changed SW tools and frameworks or added any new one please describe it. Any new SW or framework you will be using has to be approved by CTO in writing by this time.

-We will be using node packages.

- Node JS
- React
- MySQL2
- PHP (for the backend)

5. Key Risks

Skills Risks:

Risks in personal skills include having little to no experience in the technology stack being used by the team. Leads would make sure that each team member is hitting their daily goals for verification of understanding the topic to be able to be functional in cooperating with the project.

Schedule Risks:

The greatest risk would be that the allotted time frame for the project now boils down to approximately 4 weeks. Each project and milestone would have to be hit on time, or even better, early, so that the project is done on a timely manner and could even be greatly improved if necessary tasks are done early by having personal weekly and daily goals, as well as team daily and weekly goals.

Technical Risks:

Our biggest technical unknowns are being able to efficiently use the tools, API's and GUI's provided and chosen, to have them be reliable and work whenever used, and to effectively create a functional website that serves the purpose and more.

Teamwork Risks:

Given the situation where group mates are not able to meet face to face, and not knowing each other's skill set, the biggest risk is for team members to not do their tasks assigned within the time frame needed. Team leads would do necessary checks so daily and weekly, each team member is completing their tasks, and if not, catch them early so for others to help and save.

Legal/Content Risks:

There are no legal and content risks at the time, all necessary frameworks, libraries and resources will be utilized before they are set to expire, such as AWS.

6. Project Management

As a group, we all understand that our time is short and all duties must be done on time, individually and as a group. For this we have decided to best utilize Trello in order to assign tasks daily and weekly, as well as efficiently monitor who has or has not finished with each task. We have also decided to use Zoom as our main source of virtual communication, and will do meetings of 3-4 times per week, in order to ensure that each team member is doing what they're supposed to be doing, on time and also catch up on how everyone's doing, have questions and spend some time getting to know each other better, for a boost in morale and getting the project done in a positive environment. Another resource that the group will be using is Slack for communications and clarifications between sub-groups and the group as a whole. To increase productivity, team leads will meet to discuss each person's task, and make sure that deadlines are hit on time. On top of that, each sub-team lead will be assigning tasks within each sub-group and will be required to meet and consistently keep open communication between each member of the group to ensure that nothing gets left behind and that everything is done early, if not on time. Finally, Github and Google Docs will be our main tools for facilitating projects and milestones per week.

Milestone 3

SW Engineering CSC648/848
Summer 2020

Milestone 3
Online Real Estate Marketplace:
Hillow

Developed by Team 4:

Wiam Boumaazi, Team Lead (wboumaazi@mail.sfsu.edu)
Karina Abad, Front End Developer
Utsav Bhatta, Back End Developer
Katie Kennedy, Back End Lead
Jaren Lynch, Front End Lead
Mohammadrehan Kanuga, Back End Developer

08/05/2020

Submitted for review:

Submission date	Date revised
07/13/2020	08/05/2020

Milestone 3 Feedback:

During Milestone 3, we met with Professor Petkovic and demonstrated our vertical prototype and received feedback on our current progress. The feedback from the meeting was as follows:

- The nav bar should contain the search bar + the filter options + Aboutus + Post + login
- The nav bar should remain the same in all pages
- The home pages should contain example of listings (like the newest listings)
- Searching for listing: The result that will appear after the user click search are either for buying or renting -> means an extra filter is required to choose (buy/rent)
- Basically the summary of filters: (room/apt/house) | price | distance | beds | (buy/rent)
- When showing the results in a new page, the nav bar should be there also and will show what the user looked for or what he entered to get those data.
- User can filter the buy/rent after the results are shown
- For the posting: when a user makes a posting, he should select if this post is for rent or for sell. What the user post, should be stored in the listing database
- The cards of the listing should have the same size
- When the user gets the results after searching, every card should have a contact message button -> when you click on it, if you are not logged in, you will be asked to login or register -> then it will direct you to the page where you can contact the agent about the listing you chose.
- The messaging page shall have: username, place Id, date, message
- Message then will be saved in the agent_message DB and should be shown in the user dashboard
- The user dashboard contains the user listing chosen and messages he made.
- Admin dashboard, should show, all messages (to agent) and posts
- no need to message the admin anything!
- The post page: mark the mandatory fields
- Didn't really understand this part but: he suggested that we export the listing database and put it directly in the admin dashboard.

Milestone 3 Plans:

After receiving the feedback from Professor Petkovic, our team met and discussed the feedback and developed plans to address the issues raised.

First we separated the issues into whether they were frontend, backend issues, or high level issues. After this, we discussed with each lead the issues and possible solutions, and a time frame at which these issues could be completed.

UI Issues:

- The nav bar should contain the search bar + the filter options + Aboutus + Post + login
- The nav bar should remain the same in all pages
- The home pages should contain example of listings (like the newest listings)
- Searching for listing: The result that will appear after the user click search are either for buying or renting -> means an extra filter is required to choose (buy/rent)
- Basically the summary of filters: (room/apt/house) | price | distance | beds | (buy/rent)
- When showing the results in a new page, the nav bar should be there also and will show what the user looked for or what he entered to get those data.
- User can filter the buy/rent after the results are shown
- For the posting: when a user makes a posting, he should select if this post is for rent or for sell. What the user post, should be stored in the listing database
- The cards of the listing should have the same size
- When the user gets the results after searching, every card should have a contact message button -> when you click on it, if you are not logged in, you will be asked to login or register -> then it will direct you to the page where you can contact the agent about the listing you chose.
- The messaging page shall have: username, place Id, date, message
- The post page: mark the mandatory fields

Backend Issues:

- Professor suggested that we export the listing database and put it directly in the admin dashboard.
- Message then will be saved in the agent_message DB and should be shown in the user dashboard

High level issues:

- The state of the project, in that we did not have a successful horizontal prototype.

After discussing with the team, issues were broken down into tasks and P1 priority, and added to Trello. From there respective areas of the team began tackling the issues.

Milestone 4

SW Engineering CSC648/848
Summer 2020

Milestone 4
Online Real Estate Marketplace:
Hillow

Developed by Team 4:

Wiam Boumaazi, Team Lead, Github master (wboumaazi@mail.sfsu.edu)
Karina Abad, Backend Lead
Utsav Bhatta, Backend Developer
Katie Kennedy, Backend Developer
Jaren Lynch, Frontend Lead
Mohammadrehan Kanuga, Backend Developer

Submitted for review:

Submission date	Date revised
08/03/2020	08/05/2020

1) Product Summary

Name of product - Hillow Rental

Itemized List:

Our system will allow unregistered users to:

- Register.
- View all the listings.
- Search for a listing by empty parameters or by zip code.
- Search for a listing by choosing a type (apt, room, house) from the drop down menu.
- Agree the terms before they register.

Our system will allow registered users to:

- Login.
- post a listing on the site.
- contact landlords.
- view their messages.
- view their listings.
- do all unregistered users can do.
- logout if they were logged in.

Our system will allow admin to:

- contact landlords.
- inspect all posts.
- approve a listing before they upload on the site.
- do all unregistered users can do.

Our system will have posts that:

- Contain the type of the listing (Apt, Room, House)

- Contain the address of the listing
- Contain the price of the listing
- Contain the distance from campus of the listing

Our system requires an email ending with (@sfsu.edu) for registration and also the search text entry shall be validate to allow only up to 40 alphanumeric characters (letters and numbers only)

URL: http://ec2-54-219-200-213.us-west-1.compute.amazonaws.com

2) Usability Test Plan:

Test Objectives:

To test the usability of our product, we will be testing the search functionality of our product. This is the function that most users will use, and as such is the most important to get it right. At first, our search only filtered via listing types. The addresses, prices, and images etc. were grouped together. This website is primarily for SFSU students and faculties. While other users can also browse the listings, we felt the need to make that distinction in the search results as well.

We expect our search result will assist users whether they know what they're looking for or not, and keep them from drowning in too many search results if they're hunting for something in specific. Our usability test plan is designed to lay bare what aspects of our website accomplish what we set out to do, and which features need improvement before full deployment.

Test Background and Setup

- System setup

Due to the pandemic circumstances, remote testing shall be a necessity. As such, our testers will need to be those with a computer access. Users will also need to already have a Google Chrome or Firefox browser installed, or be willing to install one. In order to track task completion and efficiency (such as number of clicks, time taken between tasks, etc), testers would also need to have or be willing to install an app capable of screen sharing to allow us to at least take a rough measurement of their clicks, mistakes, and time taken. But since our usability will focus primarily on user satisfaction, those steps are optional.

- Starting point

Testers will be sent the website's url along with their instructions. If they don't have screen sharing capabilities, we'll ask them to send a quick message like "done with #2" upon completion of each step as an approximate measure of time.

- Intended users

Our intended users are San Francisco State University students and faculty.

- System URL, what is to be measured

System URL: http://ec2-54-219-200-213.us-west-1.compute.amazonaws.com

The test shall primarily measure user satisfaction using Likert tests, but task completion time will also be measured approximately.

Usability task description:

- Instructions for testers (4 SFSU students randomly chosen to test the usability of our application.)

Thank you for helping us to test our website! Please use our website to search the apartment you prefer, and contact the seller. Here we will tell our users what to do, not how to do.

How would we measure effectiveness?

- Was the user able to use the search bar to find the listings and contact the seller?
- Out of four testers, if 2 of them complete the task successfully, and the rest are only able to find listings but unable to contact the seller, we may want to measure the effectiveness of our application to be at 75%. If one user completes the tasks successfully and the rest cannot complete the tasks, then the effectiveness of our application will be 25% success rate.

Example of simple feedback form.

Example of simple feedback form:

Task	Were you able to complete the task?		Other (type your answer)
	Yes (put Y)	No (put N)	

Search the listings			
View the listings			
Contact seller			

If you have any further comments or feedback please write them below

How would we measure efficiency?

● We will measure the efficiency of our application using the same users and tasks that were assigned in the previous step. To measure efficiency we will be looking at time taken to complete the task. We would be measuring the total time to complete the task for each person, which we would use to assess an overall average as well as an average between four users. We would also be assessing the number of clicks necessary to complete the task as well as the number of screens visited. The minimum number of clicks needed to complete this task is four and the goal would be that the user does not have to use more than six clicks. The minimum number of screens that can be visited is three and total screens visited should be five or less. Our goal is that each user can complete the task within 20 to 40 seconds.

- User data will be collected as they navigate the task from within the app and we will later pool and assess the different components. Evaluation of efficiency will be based on averaging the total time to complete the task, screens visited, total clicks as well as a short survey. Averages will be compared individually as well as all four users to identify areas that may take one person more time vs another based on visibility and ease of use. The survey will ask each user to evaluate the difficulty of the task, if buttons/screens were navigable and will also allow them to make individual comments or give feedback.

Example of simple feedback form:

Tasks	Number of clicks to complete the task	Time taken to complete the task
Search the listing		
View the listing		
Contact seller		

If you have any further comments or feedback please write them below:
After all data is collected, responses will be pooled for evaluation and data will be visually mapped using bar charts. We will also be using percentages, i.e. what % of users were able to complete the task in 30 seconds or less, 60 seconds or less, or more than one minute. This will be compounded with user feedback on usability and efficiency of the screenflow and UI to identify areas in need of revision.

How do we get the subjective feedback from the users?
We need the subjective feedback from the users to find out how usable our product is. How easy to use the features and how they like it overall and how can we improve from the user point of view.

- Task one: opening the application and navigating to the search bar.
- Task two: search by your type of choice and view the listings.
- Task three: contact the respective seller.

Sample Likert-scale questions:

- I am able to navigate easily to the home page and search bar.

___ Strongly Disagree ___Disagree ___Neutral ___Agree ___Strongly Disagree

- I am able to view the listings in my search result.

___ Strongly Disagree ___Disagree ___Neutral ___Agree ___Strongly Disagree

- Contacting the seller is easy for me.

___ Strongly Disagree ___Disagree ___Neutral ___Agree ___Strongly Disagree.

If you have any further comments or feedback please write them below:

3) QA Test Plans:

Test Objectives:

We will be testing the functional accuracy of the search functionality of the website. This is a vital function, as it should allow our users to find what they are looking for in a straightforward manner without any confusions. Users should be able to receive accurate results from any kind of search. A search with a city name “San Francisco” will be performed, and it should result in a specific listing on the website appearing. Idf we have 10 listings from San Francisco, it should display 10 out of 50 results on the next page. We will also be checking advanced search can be selected and used, with accurate results being produced. Keywords must be tested as well. The site should be able to match user input to the title of the listings that are available. This is another way to make the user’s experience enjoyable, and to ensure that the results they receive from the Hillow search are accurate. We will be testing the cross browser functionality of our website. This will ensure that users have options and do not have to be concerned with the browser they are using, otherwise they will be advised to use a specific kind of browser. These tests will be run on Google Chrome and Firefox.

URL : http://ec2-54-219-200-213.us-west-1.compute.amazonaws.com

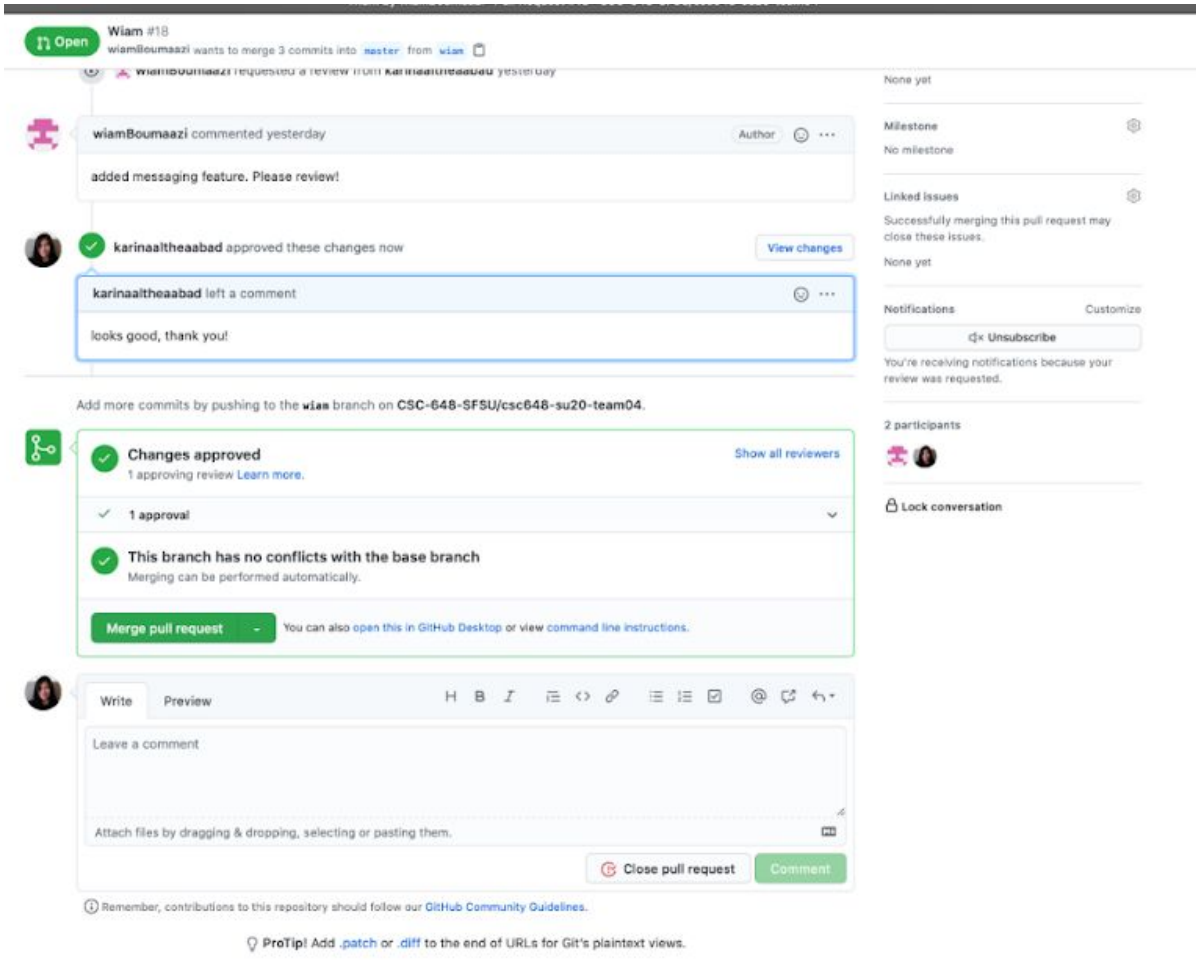
Test #	Title	Description	Test Input	Test Output	Test Result
1.	Empty Search Result	User shall be able to see all available items if the user inputs nothing.	User just hits the search button in the homepage.	User shall see “Search result was empty. Here are all the items” as a result.	Google Chrome - PASS Firefox - PASS

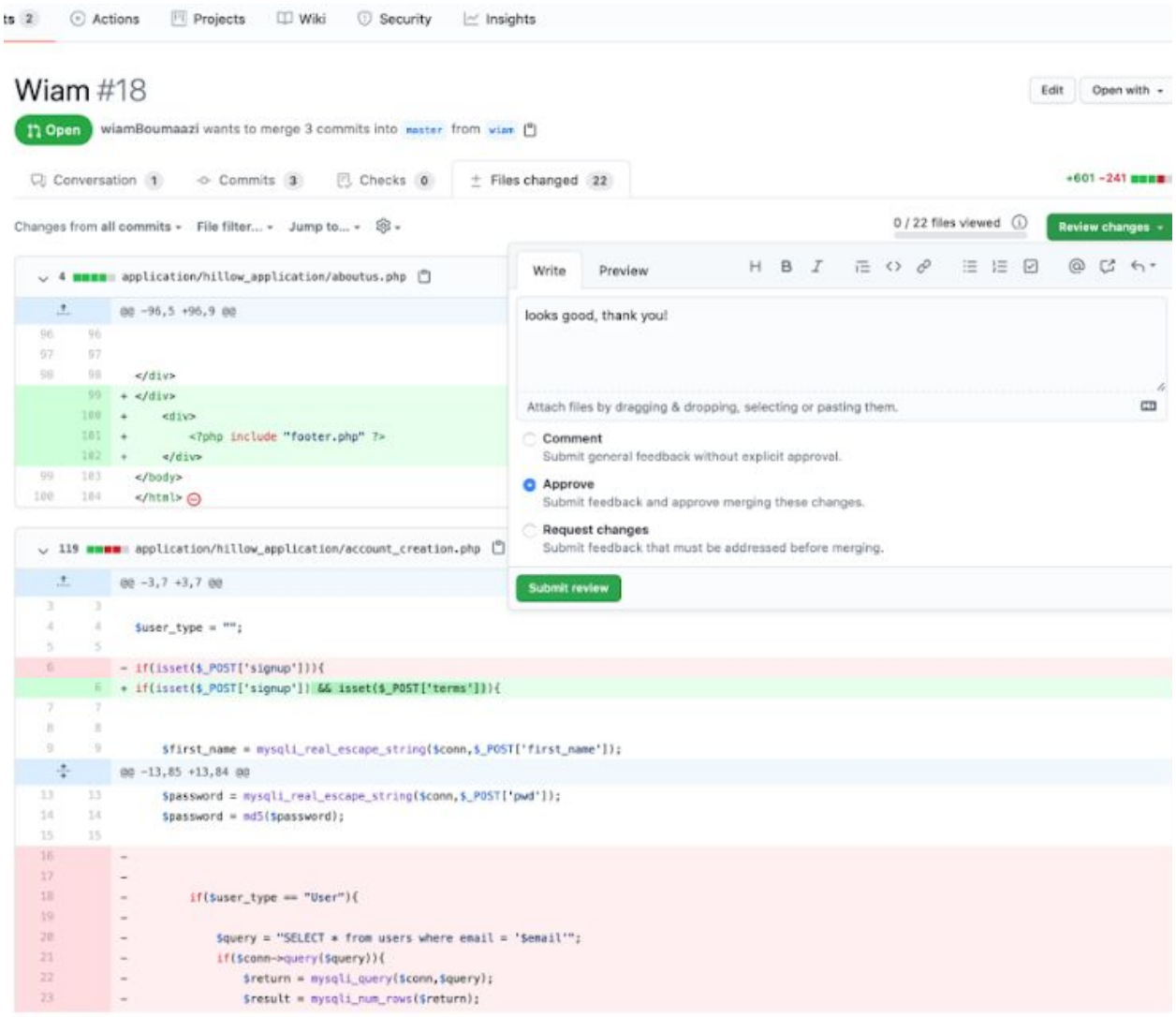
2.	Search by type	User shall be able to choose listings by type and see all available items within that category.	User choose ‘Apartment or room or house ’ in ’drop down button and click search.	User shall be able to see the exact number of listings that are of type chosen from the categories.	Google Chrome - PASS Firefox - PASS
3.	Filter the results	User shall be able to filter all the listings based on the price or the distance.	User choose the ‘price’ or the distance on search bar and Click search	User shall see The exact number of listings that are in the price or the distance range chosen..	Google Chrome - PASS Firefox - PASS

4) Code Review:

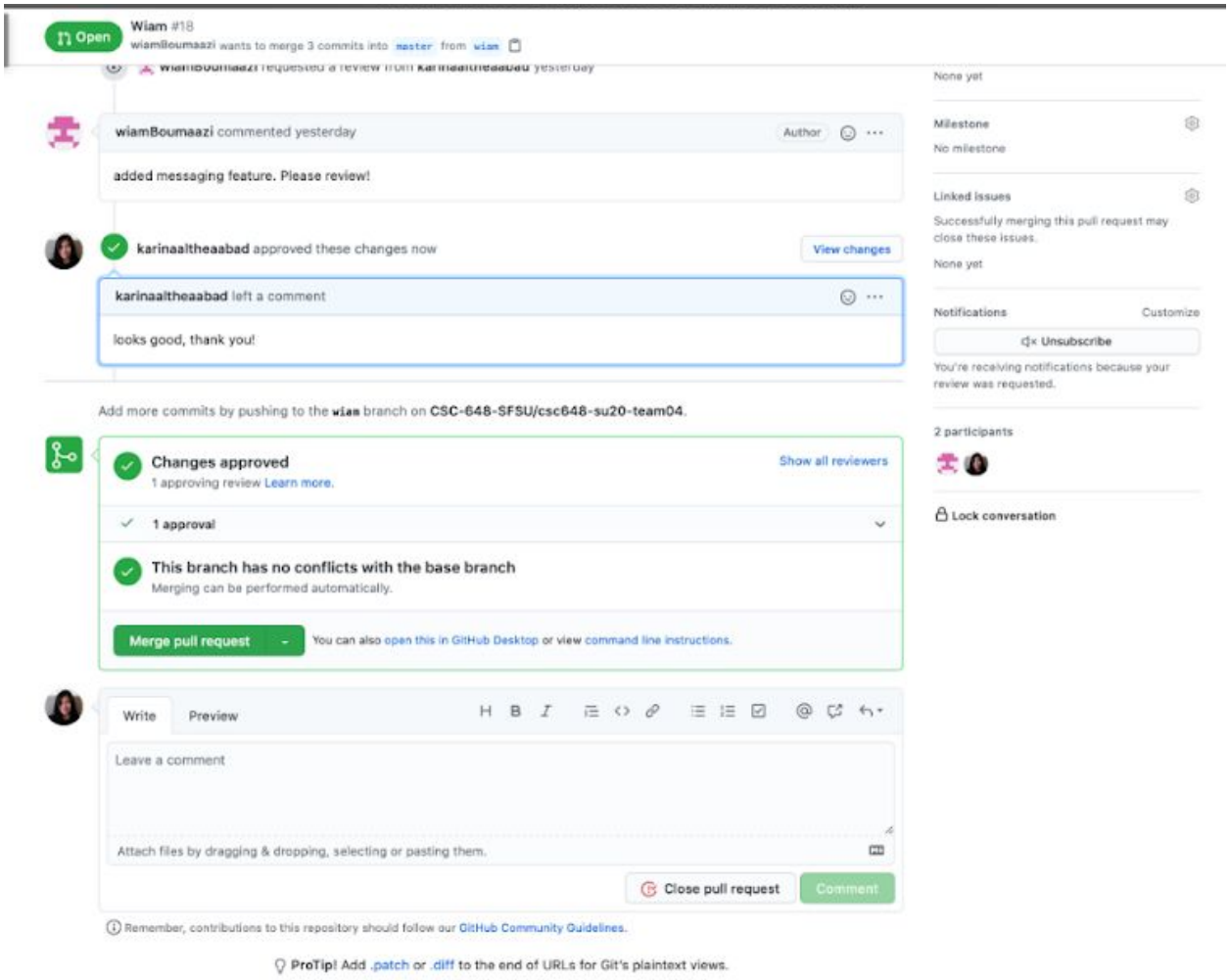
- Our coding style is conventional. We write comments describing code blocks when necessary. We declare variables in a descriptive manner. We are also using Prettier, a javascript library for global styling and implemented Eslint to check automatically for styling errors.

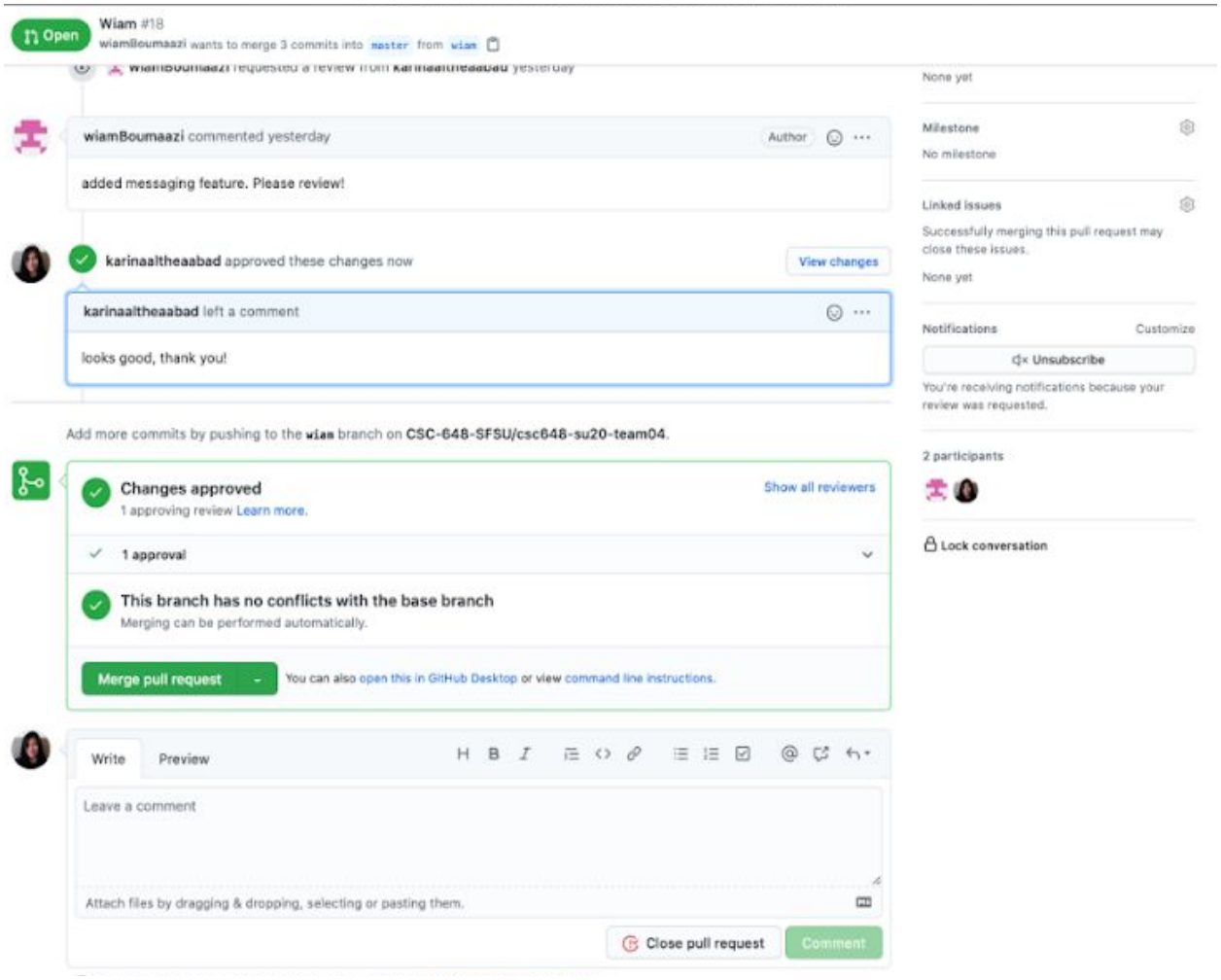
- To ensure code review, we reinforced communication using proper commit messages and feedback in person and over our private Slack channel. This helped us better understand each other's work processes and allowed us to be able to properly address changes that needed to be made.
- Ex:




ts 2
Actions
Projects
Wiki
Security
Insights
Wiam #18
Edit
Open with
Open
wiamBoumaazi wants to merge 3 commits into master from wiam
Conversation 1
Commits 3
Checks 0
Files changed 22
+601 −241
Changes from all commits
File filter...
Jump to...
0 / 22 files viewed
Review changes
4 application/hillow_application/aboutus.php
@@ -96,5 +96,9 @@

+
+
+ <?php include "footer.php" ?>
+
</body>
</html>
119 application/hillow_application/account_creation.php
@@ -3,7 +3,7 @@
$user_type = "";
- if(isset($_POST['signup'])){
+ if(isset($_POST['signup']) && isset($_POST['terms'])){
$first_name = mysqli_real_escape_string($conn,$_POST['first_name']);
@@ -13,85 +13,84 @@
$password = mysqli_real_escape_string($conn,$_POST['pwd']);
$password = md5($password);
- if($user_type == "User"){
- $query = "SELECT * from users where email = '$email'";
- if($conn->query($query)){
- $return = mysqli_query($conn,$query);
- $result = mysqli_num_rows($return);
Write
Preview
looks good, thank you!
Attach files by dragging & dropping, selecting or pasting them.
Comment
Submit general feedback without explicit approval.
Approve
Submit feedback and approve merging these changes.
Request changes
Submit feedback that must be addressed before merging.
Submit review


Open
Wiam #18
wiamBoumaazi wants to merge 3 commits into master from wiam
wiamBoumaazi requested a review from karinaaltheaabad yesterday
wiamBoumaazi commented yesterday
Author
added messaging feature. Please review!
karinaaltheaabad approved these changes now
View changes
karinaaltheaabad left a comment
looks good, thank you!
Add more commits by pushing to the wiam branch on CSC-648-SFSU/csc648-su20-team04.
Changes approved
1 approving review Learn more.
Show all reviewers
1 approval
This branch has no conflicts with the base branch
Merging can be performed automatically.
Merge pull request
You can also open this in GitHub Desktop or view command line instructions.
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Close pull request
Comment
Remember, contributions to this repository should follow our GitHub Community Guidelines.
ProTip! Add .patch or .diff to the end of URLs for Git's plaintext views.
None yet
Milestone
No milestone
Linked issues
Successfully merging this pull request may close these issues.
None yet
Notifications
Customize
Unsubscribe
You're receiving notifications because your review was requested.
2 participants
Lock conversation


Open
Wiam #18
wiamBoumaazi wants to merge 3 commits into master from wiam
wiamBoumaazi requested a review from karinaaltheaabad yesterday
wiamBoumaazi commented yesterday
Author
added messaging feature. Please review!
karinaaltheaabad approved these changes now
View changes
karinaaltheaabad left a comment
looks good, thank you!
Add more commits by pushing to the wiam branch on CSC-648-SFSU/csc648-su20-team04.
Changes approved
1 approving review Learn more.
Show all reviewers
1 approval
This branch has no conflicts with the base branch
Merging can be performed automatically.
Merge pull request
You can also open this in GitHub Desktop or view command line instructions.
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Close pull request
Comment
None yet
Milestone
No milestone
Linked issues
Successfully merging this pull request may close these issues.
None yet
Notifications
Customize
Unsubscribe
You're receiving notifications because your review was requested.
2 participants
Lock conversation

5) Self-check on best practices for security

- **Major assets we are protecting:**
    - User information
        - Passwords are encrypted in the database and can't be seen or accessed by anyone. Users are able to register for an account and login to existing accounts, the account information is saved to the database.
    - Server security
        - Admins have sole access to servers. Admin credentials are protected and not shared over insecure channels.
        - We are protecting the listings, user data, in our website that are crucial to our business and customer trustworthiness.

- **Input data validation:**
    - Upon signing up, all user inputs are being validated and an error message shall appear if the inputs do not match our validation rules. User emails are being checked using JavaScript to ensure the string the user input matches an email format ending with (@sfsu.edu) and an error will occur if the input is not an authentic email address. The search bar input is also being checked by using a percent like algorithm to compare the input to the media in our database. Empty search bar will display all results if a type is not selected from the drop down menu. Search text entry shall be validate to allow only up to 40 alphanumeric characters (letters and numbers only)

6. Self-check for non-functional:

- **Non-functional specs list:**

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). -ISSUE
   . We have changed our server-side language (Javascript) to (php) and we have mentioned that change on M2 document.

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 email clients shall be allowed. Interested users can only message to sellers via 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

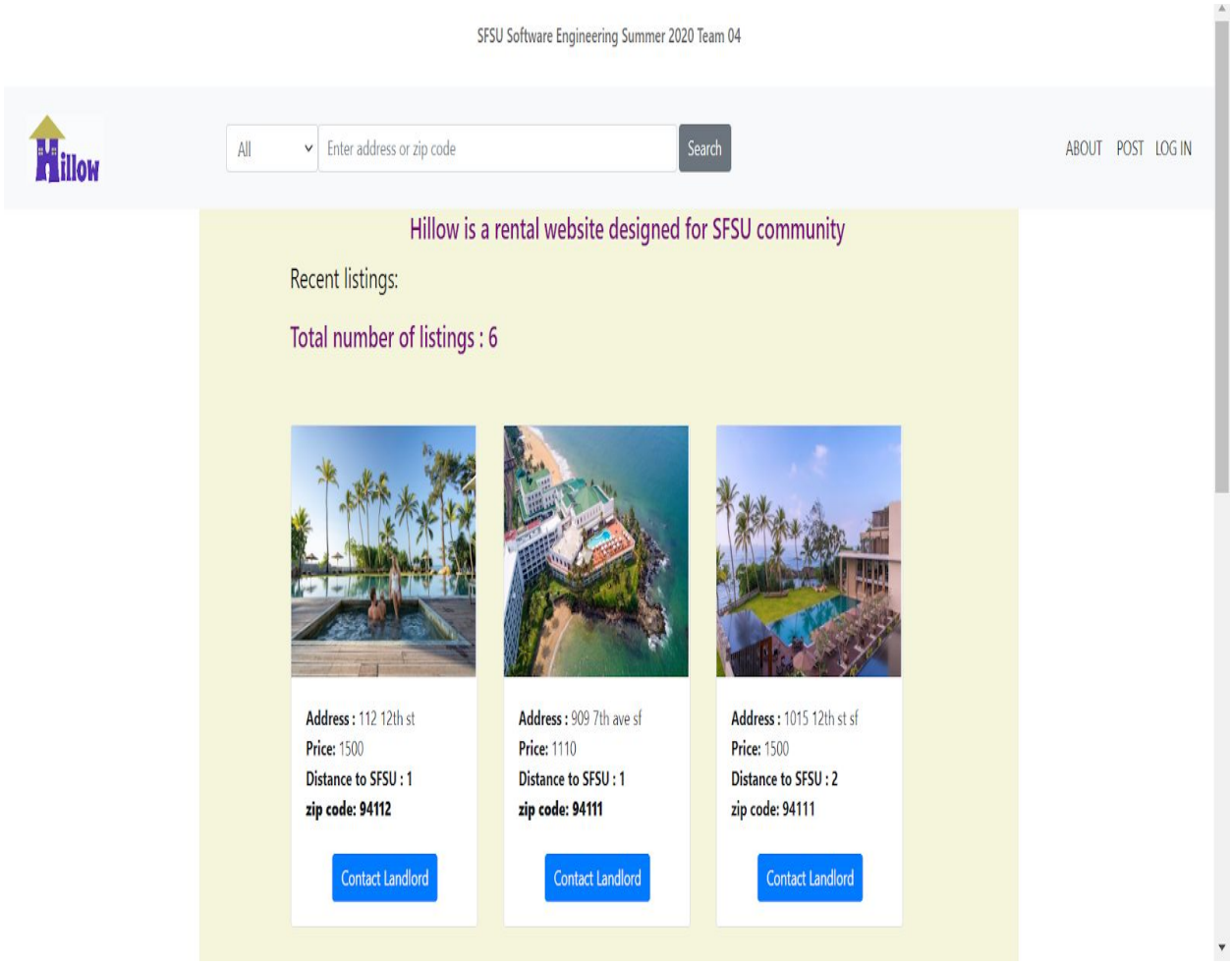
12. Site security: basic best practices shall be applied (as covered in the class) for main data items. -DONE

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

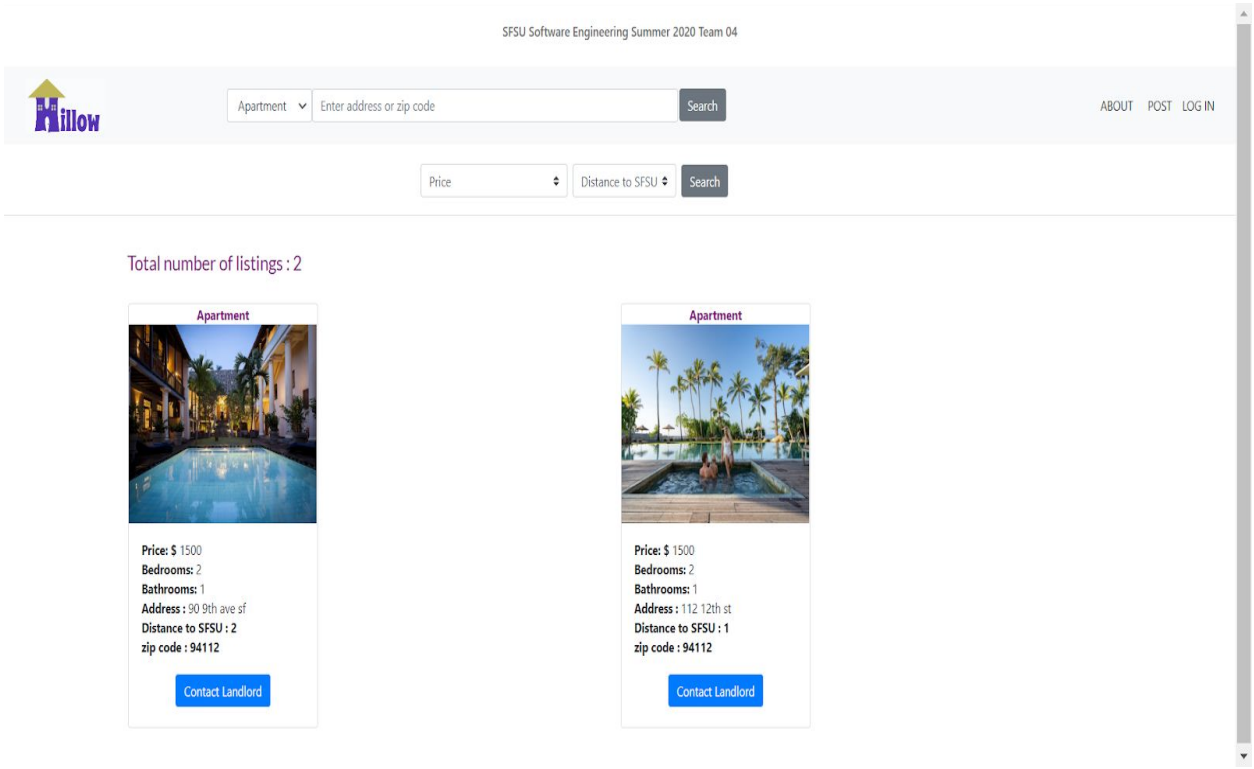
15. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Summer 2020. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application). -DONE

4.Product screenshots:

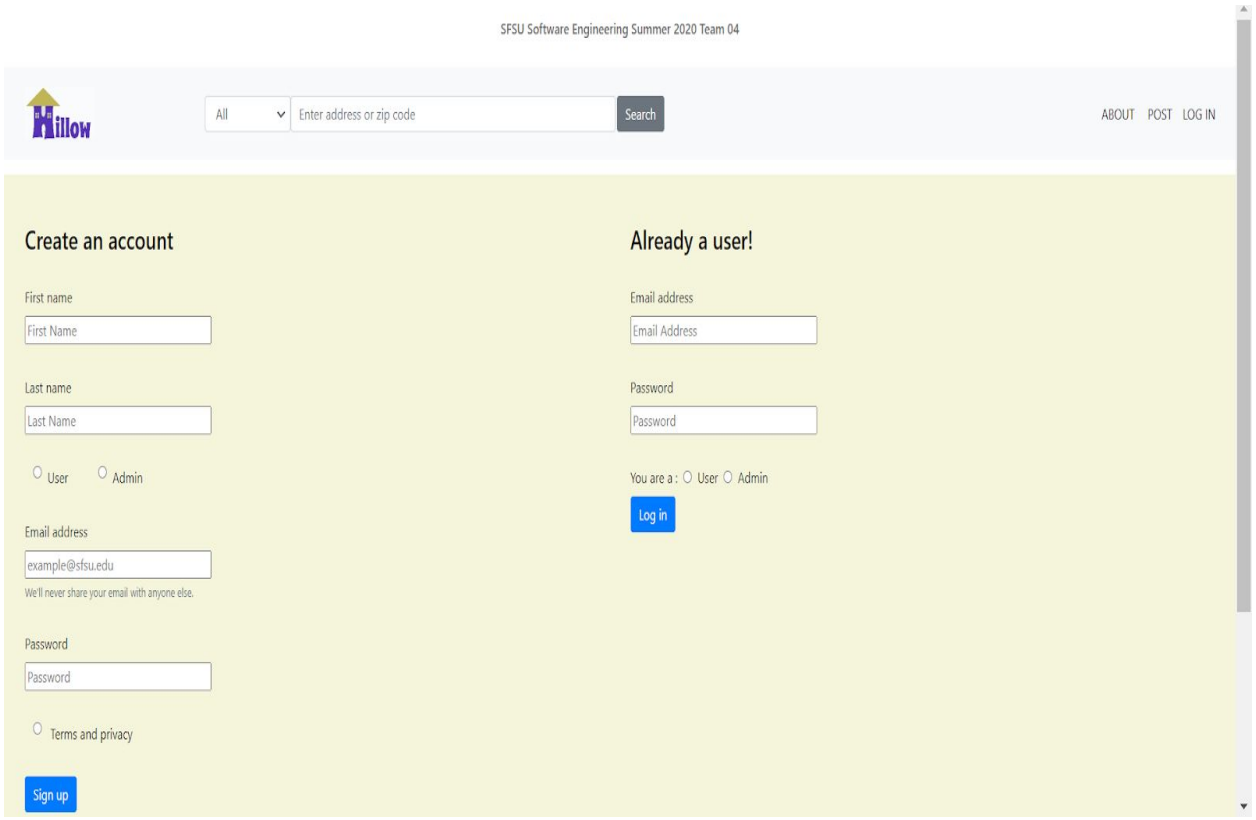
Home Page:



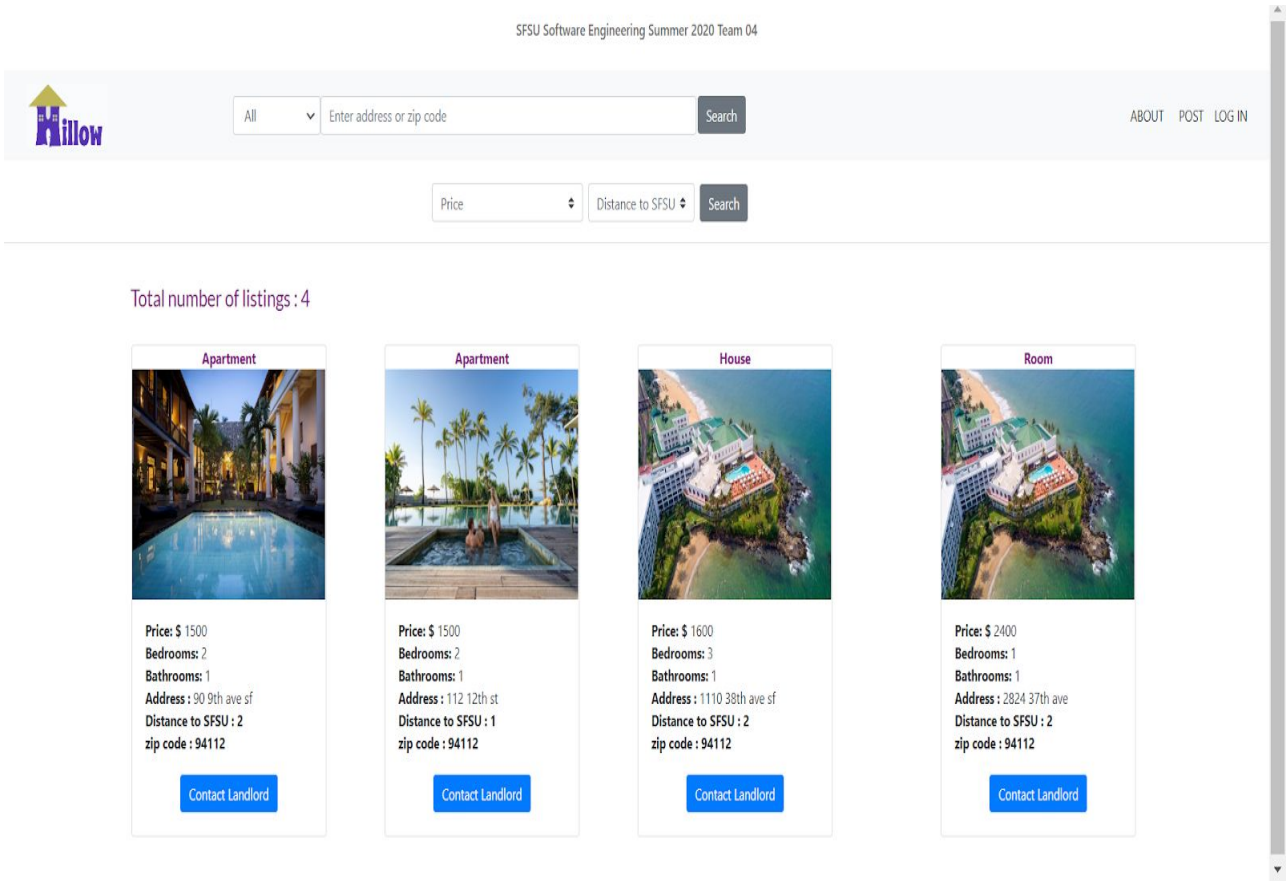
Search Result Page by Type and Zip Code:



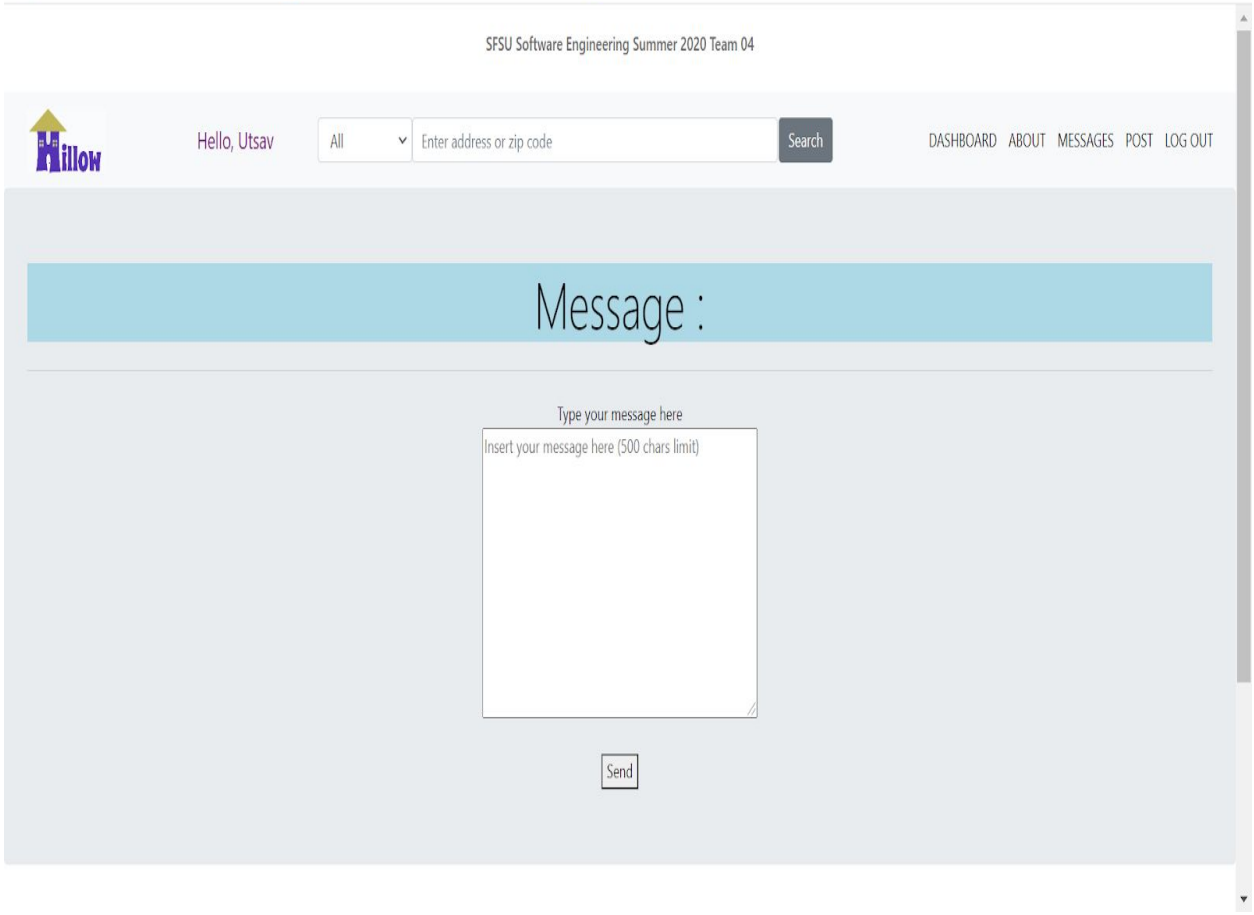
Register and Login Page:



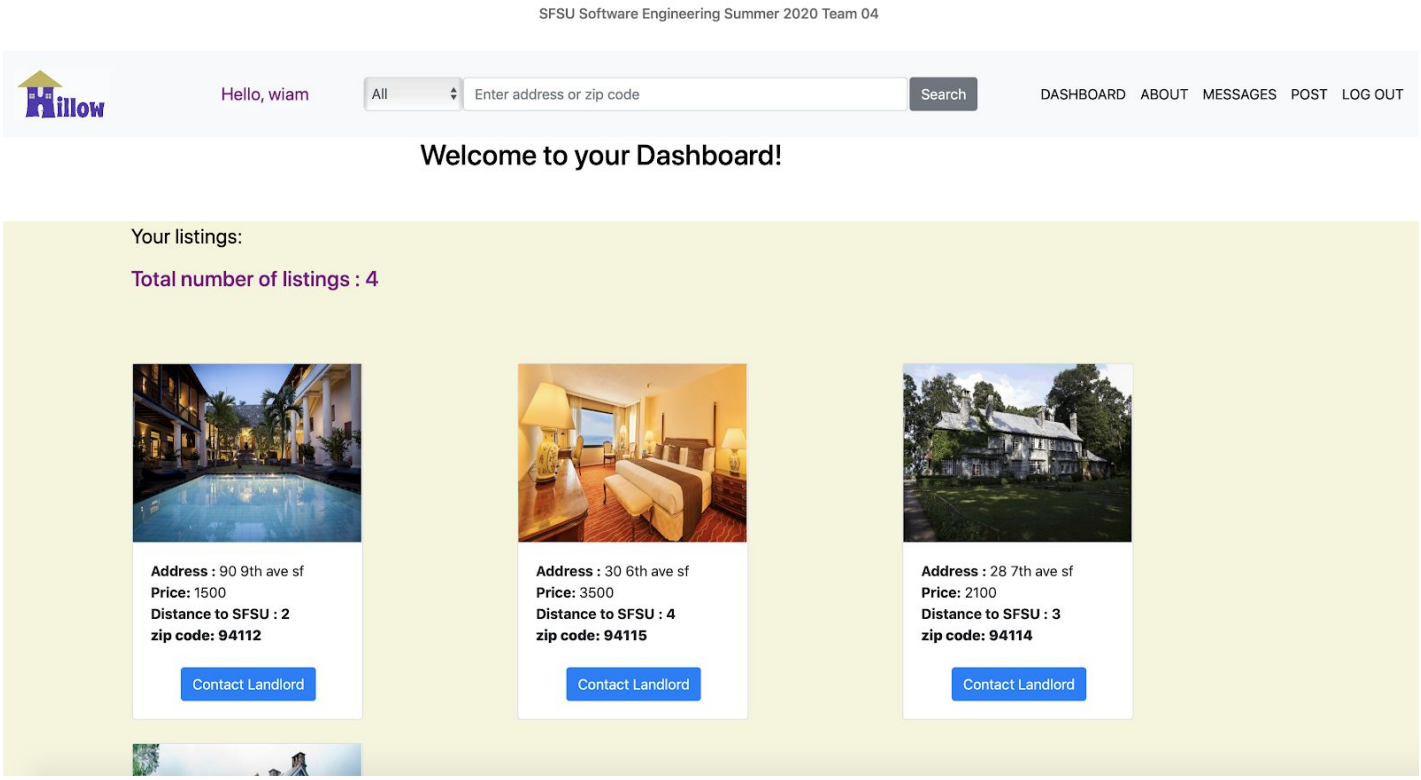
Search Page by distance and price:

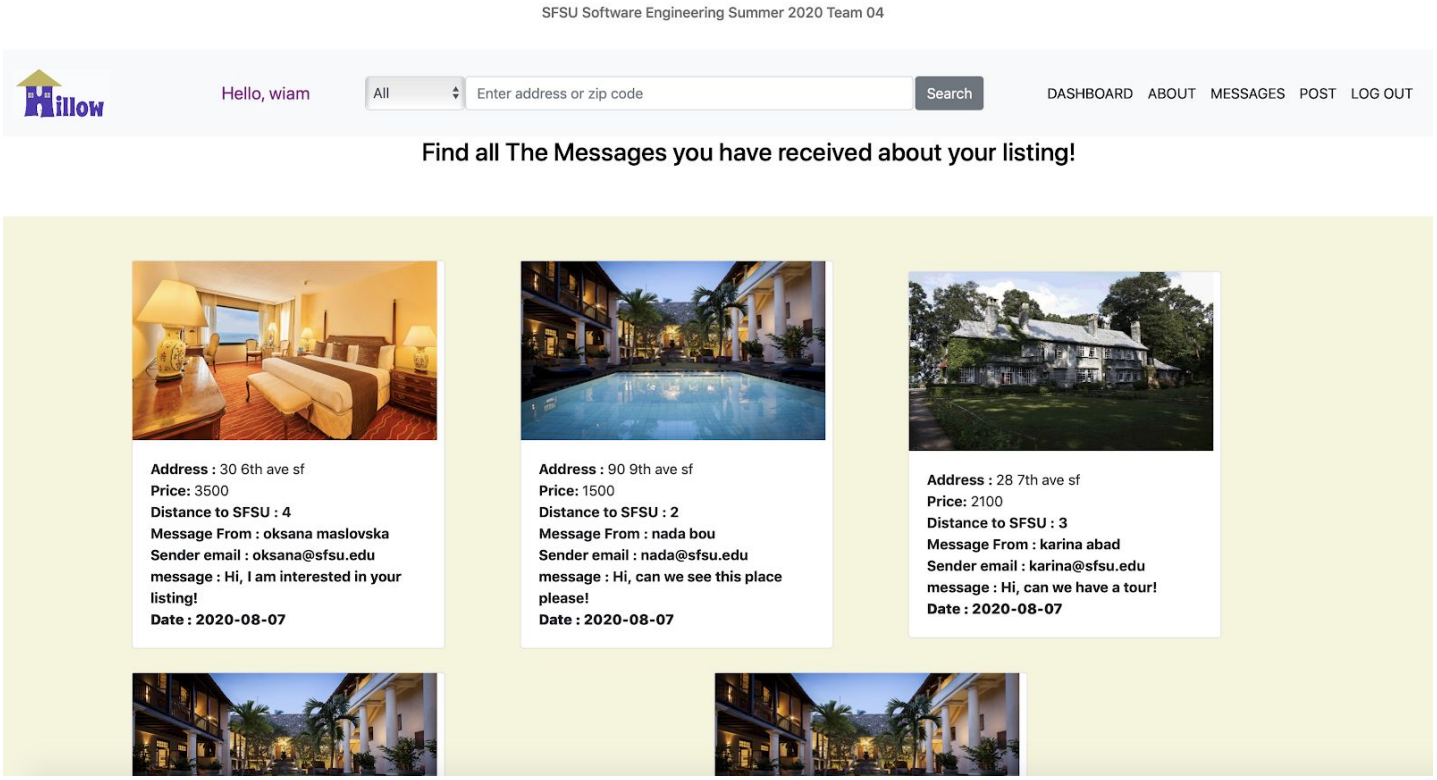


Messaging Screen to contact seller:

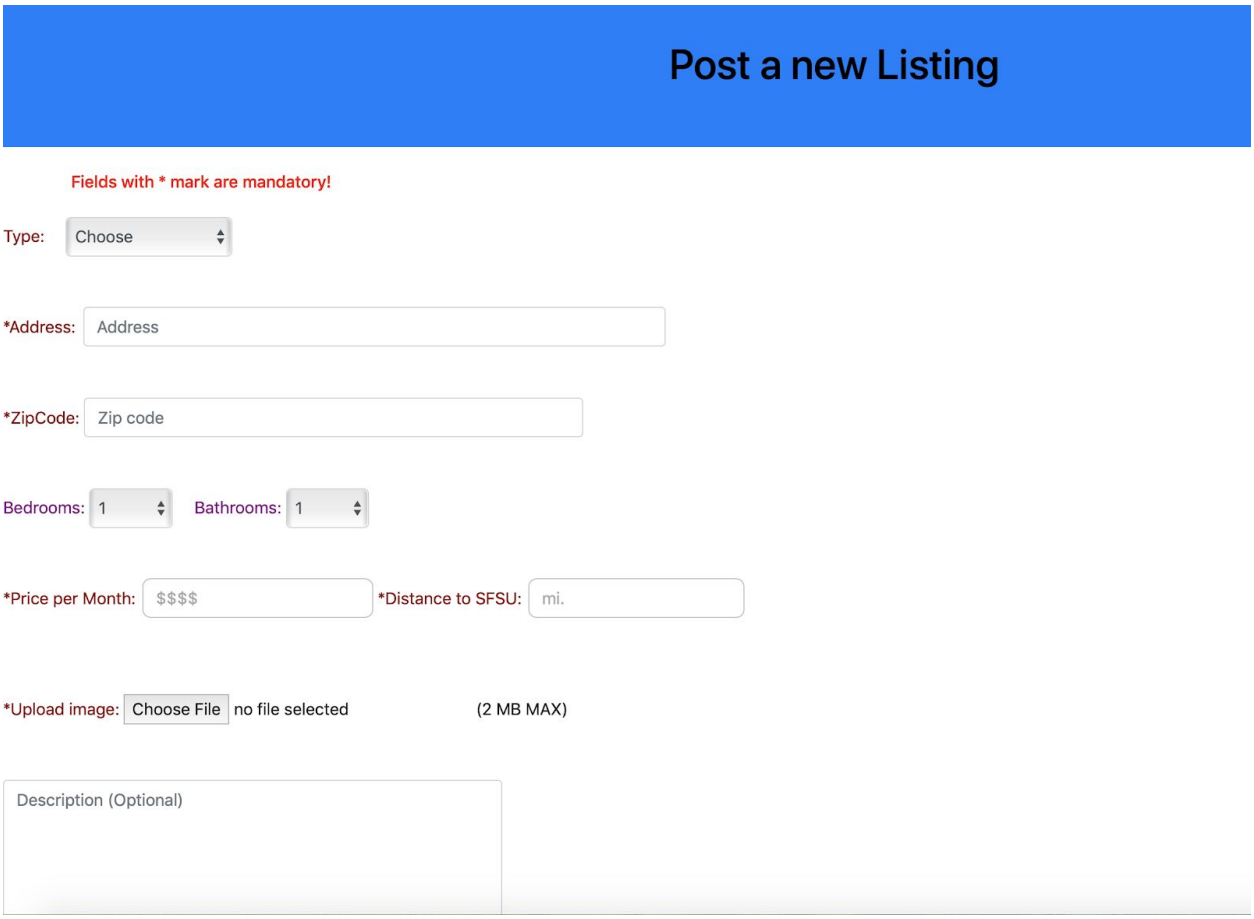


User Dashboard Page:





Create a new listing:



5. Database organization:

Below is our User table:

Options

	uid	first_name	last_name	email	pwd
Edit Copy Delete	1	wiam	boumaazi	wiam@sfsu.edu	202cb962ac59075b964b07152d234b70
Edit Copy Delete	2	karina	abad	karina@sfsu.edu	202cb962ac59075b964b07152d234b70
Edit Copy Delete	3	nada	bou	nada@sfsu.edu	202cb962ac59075b964b07152d234b70
Edit Copy Delete	4	oksana	maslovska	oksana@sfsu.edu	202cb962ac59075b964b07152d234b70

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Listings table:

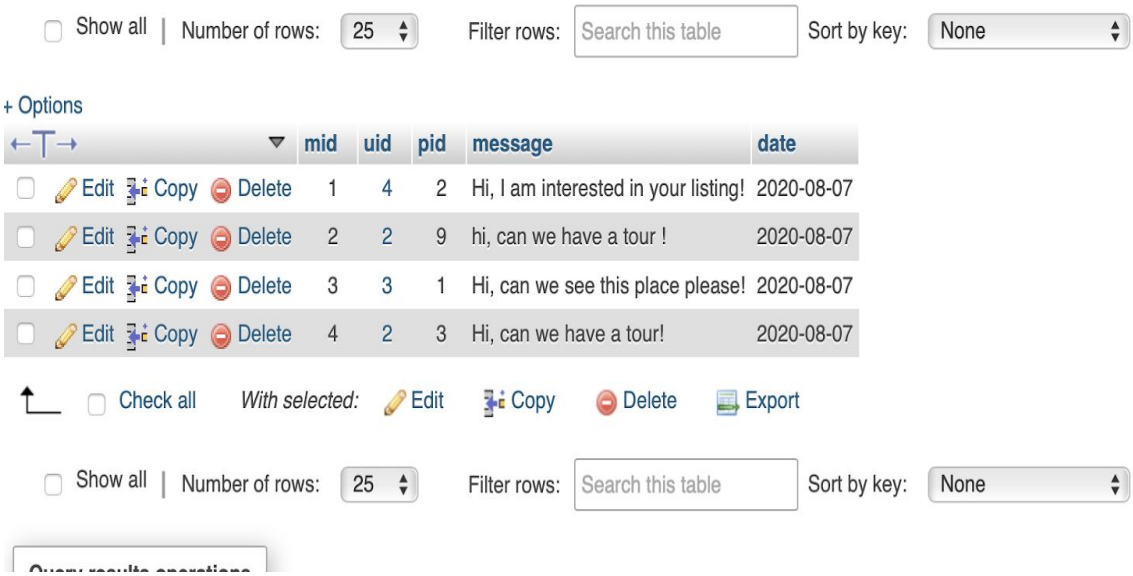
- Options

	pid	landlord	address	zipcode	type	number_bed	number_bath	price	description	distance	image
Edit Copy Delete	1	1	90 9th ave sf	94112	Apartment	2	1	1500		2	[BLOB - 119.5 KiB]
Edit Copy Delete	2	1	30 6th ave sf	94115	House	3	1	3500		4	[BLOB - 62.5 KiB]
Edit Copy Delete	3	1	28 7th ave sf	94114	House	2	1	2100		3	[BLOB - 50.5 KiB]
Edit Copy Delete	4	1	105 brodway ave sf	94114	House	3	2	950		3	[BLOB - 68.4 KiB]
Edit Copy Delete	5	2	70 holloway ave sf	94111	House	2	1	1500		1	[BLOB - 75.4 KiB]
Edit Copy Delete	6	2	1110 38th ave sf	94112	House	3	1	1600		2	[BLOB - 63.8 KiB]
Edit Copy Delete	7	2	2824 37th ave	94112	Room	1	1	2400		2	[BLOB - 63.8 KiB]
Edit Copy Delete	8	3	443 ply ave sf	94114	Apartment	2	1	2500		3	[BLOB - 262.5 KiB]
Edit Copy Delete	9	3	1015 12th st sf	94111	Apartment	2	1	1500		2	[BLOB - 34.8 KiB]
Edit Copy Delete	10	3	909 7th ave sf	94111	Room	1	1	1110		1	[BLOB - 63.8 KiB]

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Messaging table:



6. Google analytics:

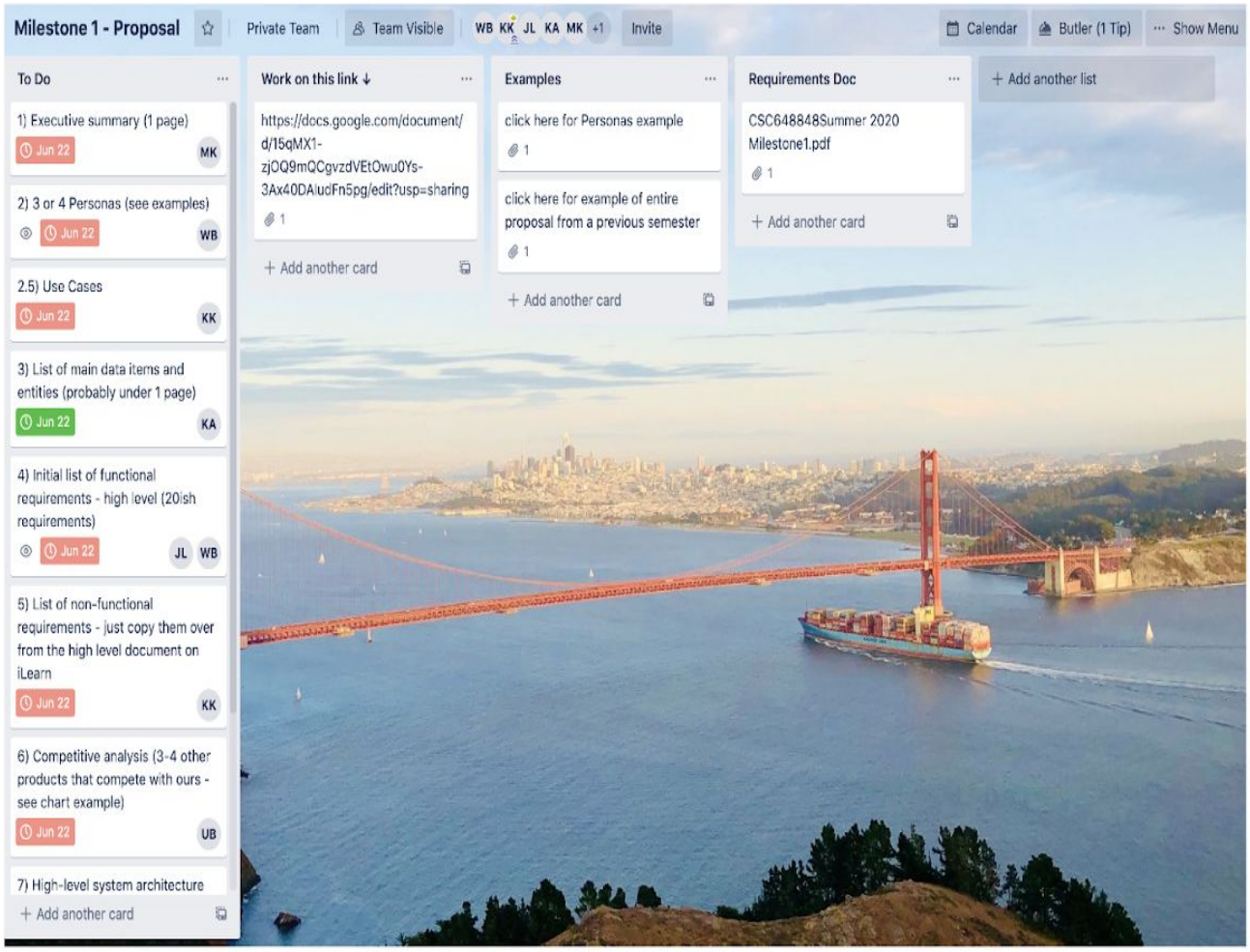
Application Requests:

API Requests:

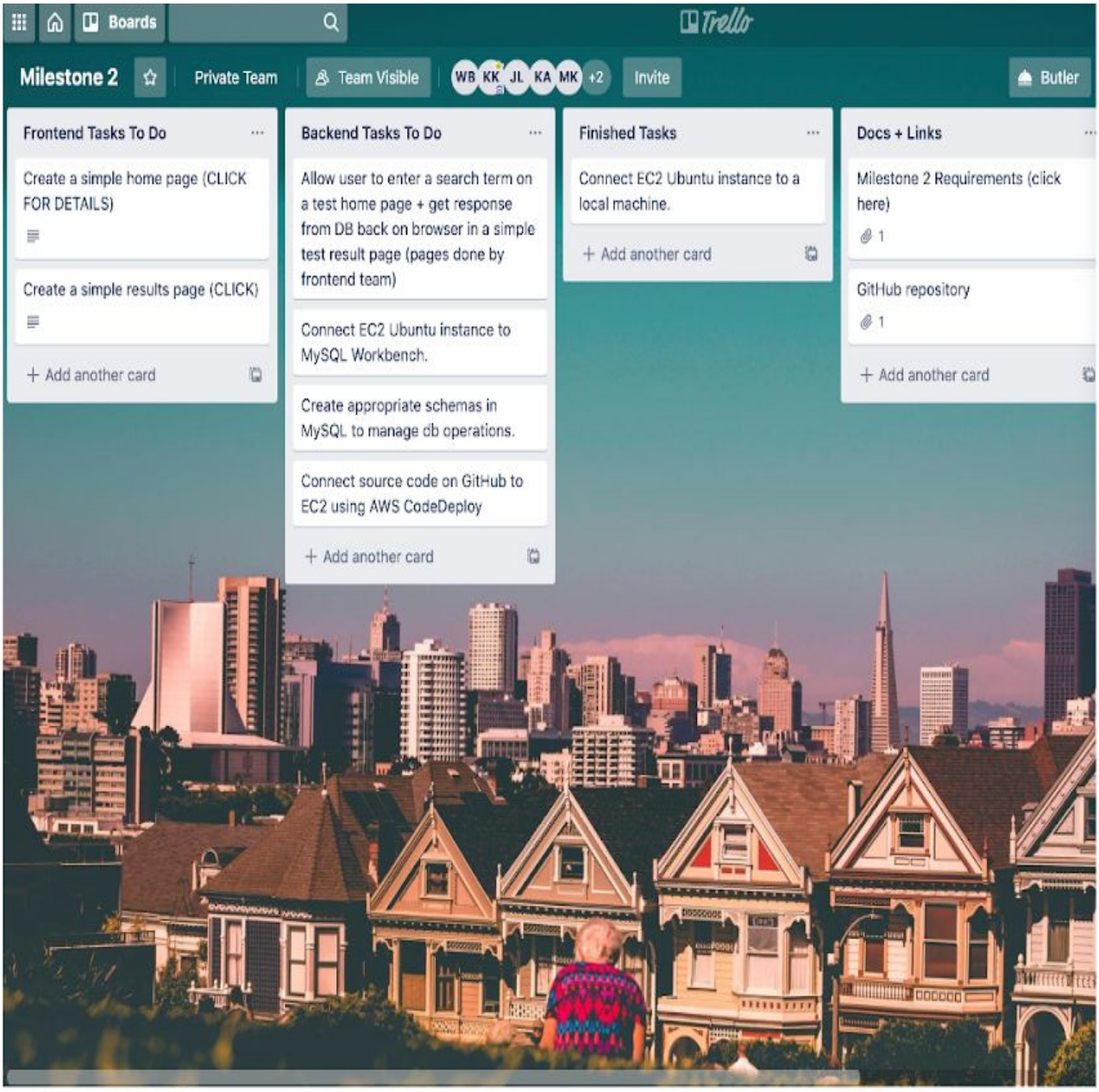
7. Project management

Trello was used to manage to workflow:

Trello Screen 1:

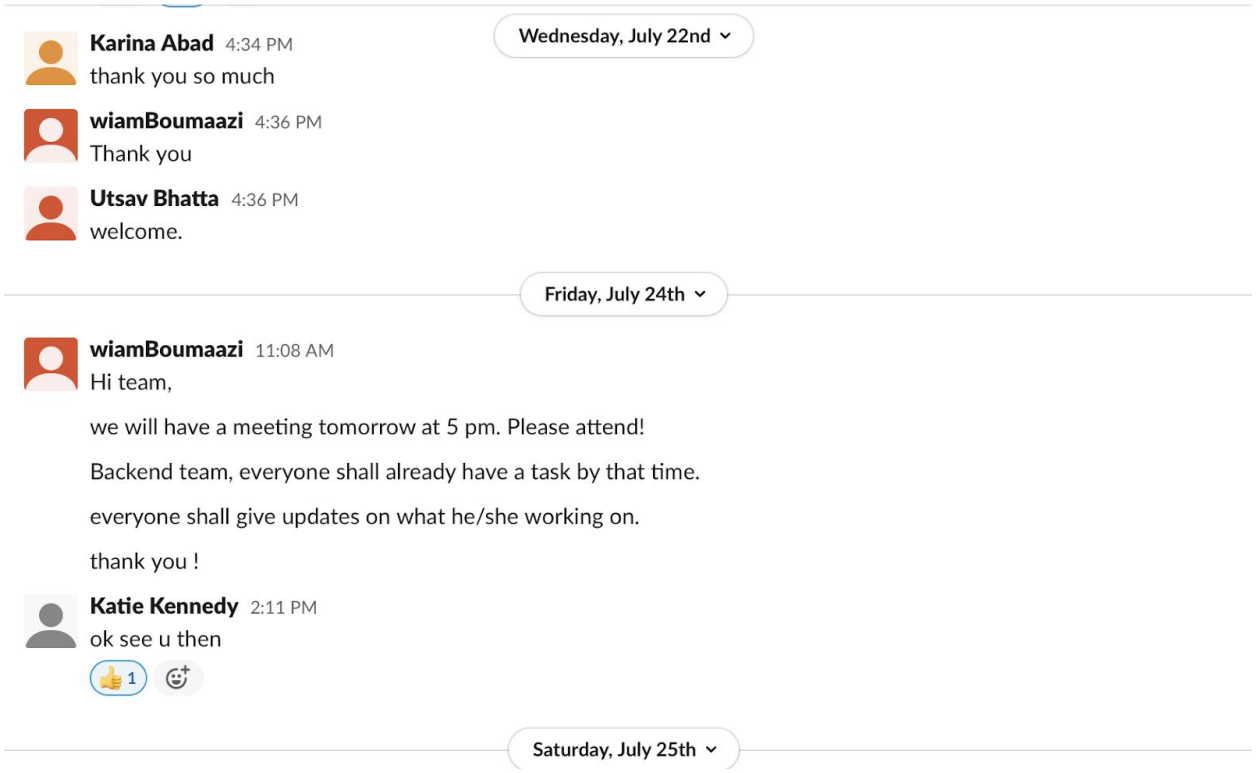


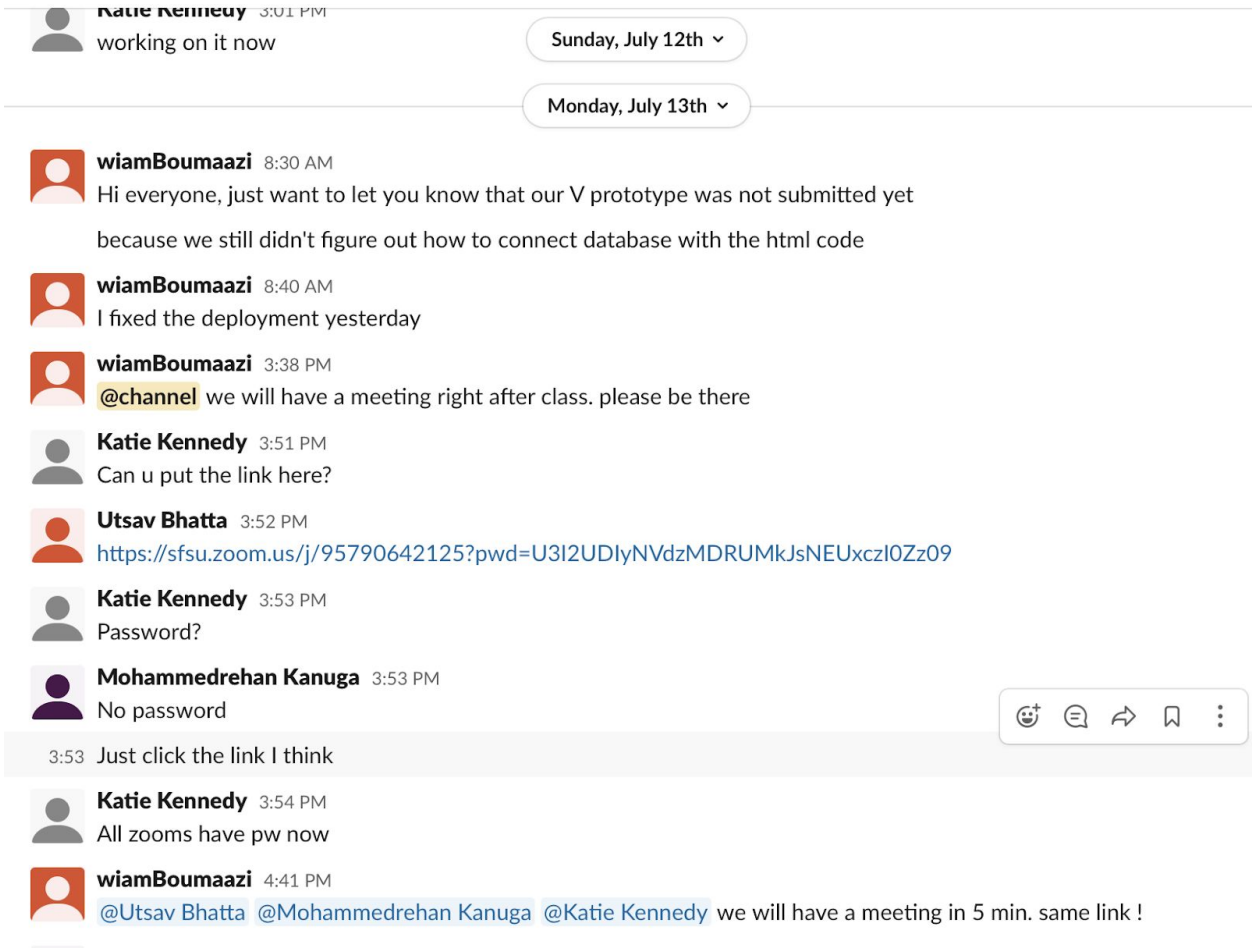
Trello Screen 2:



8. Team members self assessment and contribution:

For this class, and in order to communicate we were using slack channel and we were having meetings 4-5 times a week. Here are some screenshots from our slack channel:




Katie Kennedy 3:01 PM
working on it now
Sunday, July 12th
Monday, July 13th
wiamBoumaazi 8:30 AM
Hi everyone, just want to let you know that our V prototype was not submitted yet
because we still didn't figure out how to connect database with the html code
wiamBoumaazi 8:40 AM
I fixed the deployment yesterday
wiamBoumaazi 3:38 PM
@channel we will have a meeting right after class. please be there
Katie Kennedy 3:51 PM
Can u put the link here?
Utsav Bhatta 3:52 PM
https://sfsu.zoom.us/j/95790642125?pwd=U3I2UDIyNVdzMDRUMkJsNEUxczI0Zz09
Katie Kennedy 3:53 PM
Password?
Mohammedrehan Kanuga 3:53 PM
No password
3:53 Just click the link I think
Katie Kennedy 3:54 PM
All zooms have pw now
wiamBoumaazi 4:41 PM
@Utsav Bhatta @Mohammedrehan Kanuga @Katie Kennedy we will have a meeting in 5 min. same link !

Tip: Try ⌘ F to search this channel ×

Mohammedrehan Kanuga 11:07 AM
Team, #4 is done for our document.

Let us know if we should change/add anything.

wiamBoumaazi 11:15 AM
thank you guys for your hard work

I also worked this morning on the deployment on ec2

I changed a little bit just to make it work

18.216.135.86:8000

Mohammedrehan Kanuga 11:16 AM
Morning, did it work after that?

wiamBoumaazi 11:16 AM
http://18.216.135.86:8000

> **18.216.135.86**
> React App
> Web site created using create-react-app

now it is working but I had to change a little bit in the design

Mohammedrehan Kanuga 11:17 AM
Yes it's working now.
👍 1

Utsav Bhatta 11:26 AM
for accessing the ec2 account, is there a team aws account? or we doing from our personal account? I am little confused about this.

Tip: Try ⌘ F to search this channel ×

wiamBoumaazi 3:12 PM
Hi team

tomorrow we will have a meeting that will take an hour or so

In this meeting, Karina will go over the code she wrote in Php and show us all the steps she followed in the vertical prototype

this meeting is very important, it will make the whole team on the same page

do you prefer morning or afternoon ?

let me know

@channel

Katie Kennedy 3:30 PM
around 5 would be best for me but im flexible

wiamBoumaazi 3:31 PM
guys 5pm is fine with you?

Karina Abad 3:32 PM
Same I'm flexible
👍 1

Mohammedrehan Kanuga 3:32 PM
5pm is fine.
👍 1

Utsav Bhatta 7:20 PM
5pm works best for me.
👍 1

Wiam Boumaazi Sent - Exchange June 8, 2020 at 11:04 AM

CSC648-848 Welcome to team#4 !

To: kkennedy2@mail.sfsu.edu, kabad1@mail.sfsu.edu, jlynch4@mail.sfsu.edu, Utsav Bhatta, mkanuga@mail.sfsu.edu

Hi all,

I hope you all are doing good. My name is Wiam Boumaazi, I am the team lead and I am very excited to be part of this team.
Today we are going to have our first meeting to introduce ourselves and get to know each other. We will also have to choose the frontend/backend/github leads.
It will really be helpful and preferable if we can use video to see each at least for the first meeting (but it is not required!).

Today's meeting will start at 3:30 pm to 4:20 pm you are welcome to join. Here is the link to the meeting:

https://sfsu.zoom.us/j/97211405777?pwd=NGFFb0Nnak9xd0NLZmRURHRiSUc1QT09

Meeting ID: 972 1140 5777

Password: 498287

I have also created a slack channel for us to stay connected. Please join the channel from this link:
https://join.slack.com/t/mail-vxp6883/shared_invite/zt-erc5le0n-90awM08OVZgPNJmHb7Bbmg

If there is any problem or if you have any questions I will be more than happy to help.

I am looking forward to a great session working with you!

Thank you,
Best,
Wiam Boumaazi

The Team:

Utsav Bhatta (Backend Developer):

For this team project I helped my team

- With creating the milestone documents and revising them after instructor's feedback.
- Created a search filter and advanced search filter for our listintings.
- Updated the database to meet the search and advanced search criteria.
- Created a backend for messaging functionality in NodeJs (not used due to switching our backend language to PHP)

I made 30 commits to my github team Dev branch.

The main challenges I encountered during this team project is working remotely from zoom, where having all the team members together was hard. I also got challenged learning PHP and implementing it at the same time.

For the next time, I would like to think about the project from the perspectives of personas and use cases. So, I know who my users are and what functions will be useful. Definitely do low fidelity and high fidelity mockups for my designs, which achieves feedback from users without investing coding in less time. One thing that is important for every project is SCRUM. I will use SCRUM to update my team and myself.

Jaren Lynch (Frontend Developer)

- I was front end lead.
- Worked mainly on front end.
- Worked on M1-M5
- Worked on About, Survey, CreateListing, and Admin page.
- Total commits to Github - 10.
- The main challenges I encountered revolved around communication over slack and zoom. I often found myself out of the loop due to not missing slack notifications.
- In the future I will be more familiar with the remote class and working as a team remotely and will try to stay up to date. Also I will work on side projects involving the technology we used to so as to get better at using them.

Mohammedrehan Kanuga (Backend Developer)

- By contributing to the written milestones we worked together according to the criteria, I supported my team, and then fixed them according to the professor's feedback.
- To delegate tasks and provide feedback on the work assigned to us, I participated in every single team meeting that was held through the zoom.
- Being a backend developer, I wrote Nodejs and added more to what other backend developer wrote however our team then decided to switch from Node Js to php.
- After switching to php, I focused on creating search and advanced search features of our website. I had to look at the front end code and then write the backend so it functions well and gives good results.
- From about page to delivery I worked and delivered my task.

There are about 21 commits from me to github team Dev. branch

Throughout this whole process of starting from scratch and up to delivering the whole website we used zoom learning and it was challenging as it was the first time ever in my life where I had to do work remotely with a team on a big project. Some other challenge was to learn and use new tools and programming languages in order to be efficient.

To do better next time, I would like to focus more on the customers who will use our website and make our website design simple so they can have a very good user experience. Now that I have team experience and details about how a big project is made from use cases to adding databases and functionality to the program I can do better.

Karina Althea F. Abad (Backend Lead)

a. Contributions (per timeline, i.e. developing M1, M2)
    i. created and put together about me's (M1)
    ii. contributed in all sections of M2, drew diagrams for front end, finished the Vertical Prototype, deployed app through AWS, FileZilla, provided resources and tutorials to team members to understand where I learned the backend deployment process
    iii. created google maps function, integrated the front end with backend, created databases, created backend functions for messaging function--for horizontal prototype(time period: from M3-Presentation with Professor alone)
    iv. deployed app across all stages, helped fix back end and helped clean front end UI design

b. Git Submissions: I did a total of 4 submissions to github dev (60 commits)
c. I had a hard time dealing with inconsistent communication that led to helping last minute for submission of a functional app.
d. Along with PHP, SQL, XAMPP, AWS, FileZilla, I learned that a project depends heavily on dealing with people who may or may not be in line with each other, which can present itself as a big challenge. I also value total honesty and consistent communication much more. I will start to implement professional communication between team members often and ensure we all understand each task and deadlines instead of holding back, in trust. Lastly, I learned that every team member will enable each other to continue to grow by aiding in assessing on how to be better as a person individually, as well as a team member.

Wiam Boumaazi (team lead)

My contribution to team project and teamwork:

- Developing frontend pages.
- Developing login/signup system and user dashboard.
- Developing messaging feature.
- Contributing to the post feature.
- Contributing to all milestones and revising them.
- Github master: controlling Github use and supervising the master branch.
- Hosting and managing all the meetings as well as trying to solve any conflict within the team.

Number of my submission made to Github team Dev. Branch: 51

During this class I have learned and developed many skills as a team lead but also I have encountered some challenges. One of the challenges was to ensure good communication within the team and to put the whole team on the same page. Another challenge was the short time we had to learn new technologies in order to develop the team project. I got the chance to learn a new language (php) and my level in that language went from 0 to 4 on a scale of 5.

If I get the chance to take the leader experience again I will make sure to know more of the strengths of each team member in order to be assigned to the right tasks.

Big Thanks to Professor Dragutin Petkovic and Professor Anthony Souza for their help and assistance during this learning journey. Thank you to everyone who contributed to this project.