

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

Team 5

Roles:

- Garrett Johnson *gjohnson3@mail.sfsu.edu* (team lead)
- Buu Phan (github)
- Sarah Nafees (frontend)
- Mantasha Khan (backend)
- Nick Brown
- Kamelia Shaharova

URL: <http://3.17.193.189/>

7/8/2020

Product Summary

Name: Housing State

Committed functions

- All users shall have the capability of searching and viewing listings
- All users are required to register to contact seller or create a new listing
- Users shall register or login using their SFSU email and a password
- All users shall be able to access search from every page
- All users shall be able to reach out to renters and sellers through a message dashboard
- Users shall provide information of the listing like the cost, and picture of the house
- Registered users shall have a dashboard page where they can see their messages along with the listings they have posted.
- Users shall be contacted/be able to contact by one in-site message
- System shall not support any type of payment system.
- System shall show all descriptions of listings to all users

URL: [Housing State](http://3.17.193.189/) (http://3.17.193.189/)

Milestone 1

SW Engineering CSC648/848 Summer 2020

Housing State

Team 5

- Garrett Johnson (lead)
- Buu Phan (github)
- Sarah Nafees (frontend)
- Mantasha Khan (backend)
- Nick Brown
- Kamelia Shaharova

Executive Summary

San Francisco is one of the most expensive cities in the U.S. right now. Rent and/or buying a home has skyrocketed from affordable to downright outrageous. Whether you're dealing with flakey landlords or getting outbid on a studio with no bathroom, it can be extremely hard. Luckily a new option will be available to everyone attending San Francisco State University called Housing State. Housing State is a web application designed for students and faculty to find a new cozy home ideally close to their school or workplace. There are many other applications already available to help one find an apartment to rent or buy, but Housing State is unique because it is created specifically for students and faculty. Housing state takes in account how close knit our community is and through alumni, networking, and outside benefactors it will help one find the perfect home for the right price, in the right place.

What sets Housing State apart from other web applications is the community in which it has been created, but also the simplicity that the app provides. Looking at other large websites can get overwhelming with all the options that developers provide. Housing state is simplistic and straight forward. Here is what's out there, and here is what you can get. Simplicity is key to ensuring that all ages can use the site functions without having to think too much on how they work. Other sites also will let scams and fraudulent postings poison their website. We realize how important students and faculties information is, so our application takes this into account. Due to outside participants that may not attend SFSU, whether its landlords or housing companies, having the ability to post onto the application we came up with the feature called "administrator double check". This feature will make it so every post has to be screened and approved by an administrator. Once the administrator ensures that only viable non fraudulent information is being posted it will then be added to the website. Another key factor that our application will provide is that ability to filter through rentals and see if students or faculty are looking for roommates. When running around campus one will notice fliers for vacancies in a house to fill a bedroom. Since rent is so high this is sometimes the best route to take when wanting to cut costs. When signing a rental up on the application we will have an option that must be filled out to specify if the rental is a full house or if it's a student/faculty member looking for roommates. This will ensure that all available options are being provided to our community. Our app will also take into account if it's a student looking for roommates or faculty. This is set in place to provide an optimal living situation for both groups. By doing so students will only be able to live with fellow students and vice versa for faculty. Using our web application will provide optimal avenues for students and faculty alike. By bringing in outside avenues and also using the community to rent, buy, or sell a home anyone attending or working at SFSU should be able to find a buyer or new place to live.

Our team understands the hardships and misfortunes that one must go through when finding a new home. This has motivated us to come up with something streamline and easy for our users to use, while also benefiting students, faculty, and buyers alike. Being that some of us live in the San Francisco and other bay area locations, we realize what makes a great app.

From using the community of students within SFSU to outside avenues, we will be able to facilitate a good experience for all.

Personae

Dylan

- Characteristics
 - SFSU student
 - Avid gamer
 - Transfer student, will be a junior
 - Not loyal to brands
 - Willing to try new things
 - Skateboarder
- Goals
 - To find places and set up tours before moving to SF
 - To live with other like minded students
- Skills
 - More mobile than www experience
 - Above average techy-ness



Image from thispersondoesnotexist.com

Sophie

- Characteristics
 - SFSU student
 - Works at SFSU housing office
 - Involved in several student organizations
- Goals
 - Excel in her classes
 - Have a positive impact on the student body
- Skills
 - Average tech experience
 - Most familiar with www sites



Image from thispersondoesnotexist.com

Minji

- Characteristics
 - Professor and grad student at SFSU
 - Doing research while teaching
 - Avid Biker
- Goals
 - Find affordable housing
 - Live close within biking distance to campus
- Skills
 - Knows a small amount of programming
 - Most familiar with www sites



Image from thispersondoesnotexist.com

Francis

- Characteristics
 - Retired homeowner
 - Wants to live abroad
 - Father
- Goals
 - Rent house for passive income
 - Keep house in good condition to leave to his kids
- Skills
 - Not tech savvy



Image from thispersondoesnotexist.com

Use Cases

Dylan is a transfer student coming to SFSU who completed his first two years of school at a community college near his home in San Diego, California. Eager for more freedom and independence, Dylan wants to live in an off-campus house- hopefully with roommates who share his passion for architecture. Dylan is looking to start his search for housing from the comfort of his home in San Diego. Dylan starts his search off on craigslist but is frustrated with the huge number of listings to sort through, as well as the myriad of mislabeled posts. Dylan heads over to Housing State and immediately notices that all the postings are by SFSU students, for SFSU students. Dylan visits the site and filters for postings within skateboarding distance from school, and within his price range. After reading the bios of the postings shown, Dylan narrows his list down to 10. He responds to each posting to set up an in-person tour for his first week in SF.

Sophie volunteers her time at the SFSU student housing office to help ensure her fellow classmates' housing needs can be met. After living through a less than ideal roommate situation, Sophie became committed to making sure everyone can find safe and affordable living options. The SFSU student housing office maintains a list of housing sites that have been condoned by the school. Sophie would like to take this further and use a service that was built specifically for SFSU students. Sophie was contacted by the site administrators and asked to vet the site to make sure it contained only safe and relevant postings. Sophie was also given admin privileges to remove any postings that did not comply with site rules. At the end of every day, Sophie reads through the unapproved listings and reviews them one by one.

Minji just finished her undergrad at SFSU but is looking to further her education and receive her master's degree. To help cover her costs she decides to teach a class as well. Minji wants to live on her own, and within biking distance of campus. She logs onto the site and searches for single bedroom listings within a 5 mile radius of campus. After reading through some descriptions, she identifies a quaint studio for a reasonable price. She contacts the owner of the posting and they set up time to take a tour of the house.

Francis has just left the workforce and is starting his first year of retired life. He hopes to cross off his bucket list goal of living in another country for a couple of years, but eventually wants to return home to be with his extended family. Francis hears about the site and realizes he can post his home for rent, since it is close to campus. Francis creates a posting and is informed he was to wait for admin approval before his posting goes live. After the posting is approved the next day, Francis is contacted by over 20 students expressing interest in renting his house.

List of main data items and entities

1. Admin: The user whose approval will be required for a post to show in the listings. This user will have access to all the information stored in the database of the other users who will add listings to the website.
2. User: A user who will be able to register and create an account only with the SFSU email address. The email address will be validated since this website will be exclusively for SFSU students and faculties. The user will be able to create or inquire about a listing by contacting the seller.
3. Registered User: A user who will be able to post listings for SFSU people to buy or rent. They will be able to create separate accounts that only allow them to login, post and contact the buyers.
4. Feed: This is the page where all the listings could be seen with the latest post being on the top of the page. This page will have all general functions such editing the profile, searching through the post, and logging out of the account
5. Listing: This is an entity that will contain all the information required for a place on sale. It could be a house, apartment or studio. All the listings will appear on the feed for the registered users. They can contact the seller via the contact information provided or by replying to that post by message.
6. Message: The seller and the buyer can chat over here regarding any concerns for the listings.
7. Locating item: This will be a Map that will show where the listing is located which could be used to evaluate the distance of that place from the SFSU campus.
8. Profile: All the user will be able to see their account information and edit any preferences.

Initial list of functional requirements

1. All users shall be able to search and view listings.
2. Users are required to register to contact seller or create a new listing
3. Users shall register or login using their SFSU email and password
4. Registered users shall have a profile page and be able to upload/change/delete their profile picture, and provide a short biography
5. Registered users shall be able to edit their listing preferences and see their account information
6. Registered users shall be notified when another user sends a message or comments on their listing
7. Users shall provide information of the listing like the cost, pictures of the rooms, and contact info
8. Registered users shall be able to edit or delete their post
9. Users shall be able to keep records of chats after closing or reopening the webpage.
10. User shall be able to report another user for inappropriate language
11. Admin shall have access to the database and most content. They need to login or register.
12. A new listing shall be approved by the admin to be posted on feed
13. Admins shall be able to suspend the users
14. Admins shall not be able to view registered users' password
15. System shall store all data of users and room listings in a secure database.
16. System shall not support any type of payment system.
17. System shall provide a map of where the listing is located
18. System shall show all descriptions of listings to all users
19. System shall be able to provide a secure way to communicate
20. System shall be able to provide listings similar to user's preference

List of non-functional requirements

1. The application shall be in English.
2. The application shall be simple and user friendly.
3. The application shall display the exact text "SFSU Software Engineering Project CSC 648-848, Spring 2020" on top of the page.
4. The application shall be deployed in AWS.
5. The application shall render well in both desktop and mobile browsers.
6. The application shall only be for SFSU students and teachers.
7. All users' data shall be stored in MySQL.
8. All users' data shall be encrypted and protected equally. Only the user themselves can review and edit their data.
9. All posts shall be reviewed by admins before posting.
10. Google API shall be used.
11. No payment function shall be added.
12. Best practices for security and privacy shall be applied.

Competitive Analysis

	Our website	Zillow	Redfin	Facebook Housing
Search Bar (neighborhood, address, city, etc.)	X	X	X	
Rent Housing/Sell Housing	X	X	X	X
Buy Housing	X	X		
Roommate Option	X			X
Agent Finder		X	X	
Administrator Double Check	X	X	X	
For SFSU Students/Faculty Only	X			

Our main competitor is Zillow, and although we share similar features we have a “find roommate” feature that will make housing much easier for people. This feature will allow the user to filter through listings and have the option to choose student/faculty roommates. They will also be allowed to filter through people who share similar interests, and this will benefit users by giving them their ideal living situation.

High level system architecture and technologies used

Tech Stack

Frontend

- React
- Express
- SCSS
- Webpack

Backend

- Node.js
- MySQL

API

- Google maps

Deployment

- AWS EC2

Team and Roles

- Garrett Johnson (lead)
- Buu Phan (github)
- Sarah Nafees (frontend)
- Mantasha Khan (backend)
- Nick Brown
- Kamelia Shaharova

Checklist

Item	Status
All team members attending Zoom	✓
Found a slot outside class to meet	We did a doodle poll, but not everyone was free at the same time. We have a couple times most of us are free though.
Decided on stack	✓
Organized github	✓
Team learning about frameworks to be used	Honestly not sure about this one. I will be following up with my team regarding this.

Milestone 2

SW Engineering CSC648/848 Summer 2020

Housing State

7/8/2020

Team 5

- Garrett Johnson (lead)
- Buu Phan (github)
- Sarah Nafees (frontend)
- Mantasha Khan (backend)
- Nick Brown
- Kamelia Shaharova

Prioritized Functional Requirements

Must have

1. All users shall be able to search and view listings.
2. Users are required to register to contact seller or create a new listing
3. Users shall register or login using their SFSU email and password
4. Users shall provide information of the listing like the cost, pictures of the rooms, and contact info
5. A new listing is required to be approved by the admin to be posted on feed
6. Users shall provide information of the listing like the cost, pictures of the rooms, and contact info
7. Admin shall have access to the database and most content. They need to login or register.
8. Admins shall not be able to view registered users' password
9. System shall not support any type of payment system.
10. System shall store all data of users and room listings in a secure database.
11. System shall show all descriptions of listings to all users
12. System shall be able to provide a secure way to communicate

Desired

1. Registered users shall have a profile page and be able to upload/change/delete their profile picture, and provide a short biography
2. Registered users shall be able to edit their listing preferences and see their account information
3. Registered users shall be able to check their dashboard for any messages/comments
4. Registered users shall be able to edit or delete their post
5. Users shall be contacted/be able to contact by one in-site message or by email

Opportunistic

1. User shall be able to report another user for inappropriate language
2. Admins shall be able to suspend the users
3. System shall provide a map of where the listing is located
4. System shall be able to provide listings similar to user's preference

Main Data Items & Entities

Mantasha 7/3

1. User: A user who will be able to register and create an account only with the SFSU email address. The email address will be validated since this website will be exclusively for SFSU students and faculties. The user will be able to create or inquire about a listing by contacting the seller. There will be different types of users that are admin and registered users.
2. Admin: The user whose approval will be required for a post to show in the listings. This user will have access to all the information stored in the database of the other users who will add listings to the website. Admin will be able to go through the records that were stored from the registration.
3. Registered User: A user who will be able to post listings for SFSU people to buy or rent. They will be able to create separate accounts that only allow them to login, post and contact the buyers.
4. Feed: This is the page where all the listings could be seen with the latest post being on the top of the page. This page will have all general functions such editing the profile, searching through the post, and logging out of the account
5. Listing: This is an entity that will contain all the information required for a place on sale. It could be a house, apartment or studio. All the listings will appear on the feed for the registered users. They can contact the seller via the contact information provided or by replying to that post by message.
6. Message: The seller and the buyer can chat over here regarding any concerns for the listings.
7. Locating item: This will be a Map that will show where the listing is located which could be used to evaluate the distance of that place from the SFSU campus.
8. Profile: All the user will be able to see their account information and edit any preferences.

UI Mockups & Storyboards

Sarah & Kamelia todo by 7/7

Home page: users are able to view updated feed of listings, search a location, or sign in/register.

Home Page

Housing State

[Sign In](#)

Name

2 Bed/2 Bath available. Message for details.

Like/Comment

Sign in page: Returning users will sign in with SFSU email/password. New users can register.

Sign in Page

Welcome!

Home [Sign in](#) Messages

Sign in

SFSU Email

Enter Email

Password

Enter Password

☐ Remember me [Log In](#)

Register

SFSU Email

Create Email

Password

Create Password

Enter Password Again

☐ I accept Housing State's terms and conditions [Register](#)

Add Listing: Registered users can create a new listing to be posted on the home page after admin's approval.

Create New Listing

Add Listing

Home **Profile** Messages

- Your Listings
- Post Listings
- Settings
- Your Listings
- Likes
- Sign Out

Post Your Listing!

Are you student or faculty?

☐ Student

☐ Faculty

Enter Address

Write Description

File browser

Chose file Browse

Post

Users Listing: Registered users can view their posted listings under the Profile tab

Users Listings

Housing State

Home **Profile** Messages

- Your Listings
- Post Listings
- Settings
- Your Listings
- Likes
- Sign Out

Your Listings

oodbridge Lane San Francisco, CA 94112

1 Posted on 2/23/19

Edit Delete

582 Golden Ridge Road San Francisco, CA 94121

2 Posted on 8/13/20

Edit Delete

7182 River Ave. San Francisco, CA 94121

3 Posted on 12/18/20

Edit Delete

2445 North Avenue San Francisco, CA 94121

4 Posted on 4/10/20

Edit Delete

Messaging: Registered users are able to see/reply to messages or compose a new message.

Messages

Messages

[Home](#) [Profile](#) [Messages](#)

Your Messages

[New Message](#)

Your Messages!

John Smith

Hi, can I get more info?

[Go to Profile](#) [Go to Message](#)

Madelin Mitchell

Hello! I'm interested.

[Go to Profile](#) [Go to Message](#)

Alex W.

Is this listing still available?

[Go to Profile](#) [Go to Message](#)

New message

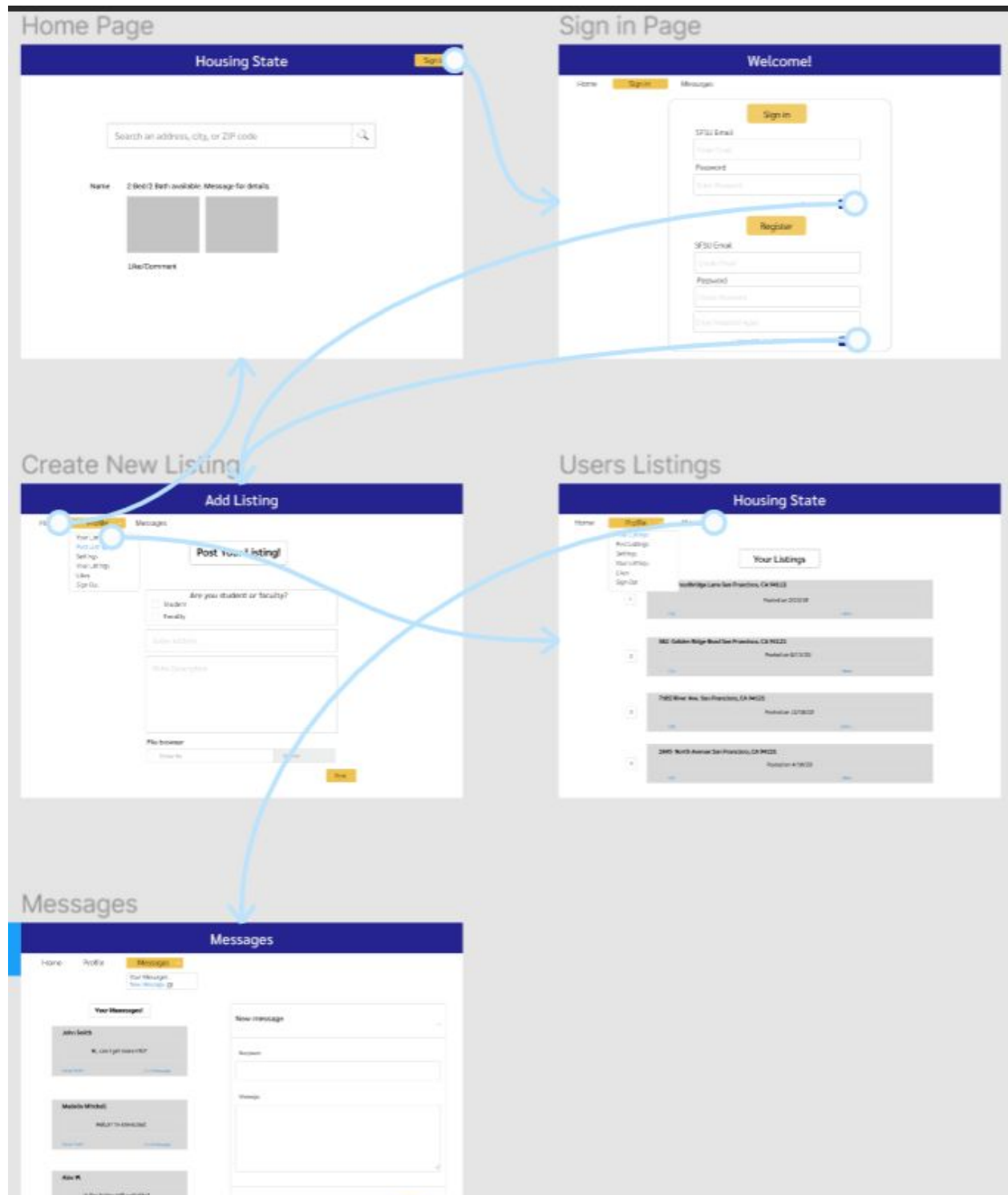
Recipient:

Message:

Cancel

Send

Storyboard



High Level Architecture & Database Organization

Mantasha, Nick todo

Tables

User

Name	Type	Kind	Description
id	tinyint	pk	
email	varchar	pk	
first_name	varchar	nn	
last_name	varchar	nn	
Password	varchar	nn	
Type	varchar		(stu/user)
Verified	bool		T/F
created_at	timestamp		
updated_at	timestamp		

Listing

Name	Type	Kind	Description
id	tinyint	pk	
user_id	tinyint	fk	
address	varchar	nn	
zip_code	int	nn	
Street	varchar	nn	
apt_suite	varchar		

state	char	nn	
cost	int	nn	
type	varchar	nn	house/apt/studio
lease_length	int		Number of months
offer_type	varchar		rent/sell
bedroom_num	int	nn	
bathroom_num	float	nn	
furnish	boolean		furnished/ not furnished
area	float		
available_at	date		
created_at	timestamp		
updated_at	timestamp		
Description	varchar(1000)		description of posting
media_image	tinyint		

Message:

Name	Type	Kind	Description
id	tinyint	pk	
to_user	int	fk	
from_user	int	fk	
text	varchar(10000)		
created_at	timestamp		
updated_at	timestamp		

Media Storage

Name	Type	Kind	Description
media_id	tinyint	pk	
relative_path	varchar	nn	pathway to image
object	BLOB		

We are going to try and store our media with BLOBs.

Search Filter Implementation

For our search implementation we will be using SQL %LIKE then the results will be filtered down by category searched.

We are checking against 5 parameters:

- Address entered in search bar: (fuzzy match with %like)
- Unit type: (apartment, condo, house)
- Offer type: (rent, buy)
- Number bedrooms
- Cost

In order for the result to be returned, at least one of the conditions must be met

API's

Every database object (user, listing, message, media) will have GET (get), POST (add), PATCH (update), and DELETE (delete endpoints)

We will also be using the google maps API to return distance from SFSU for each listing.

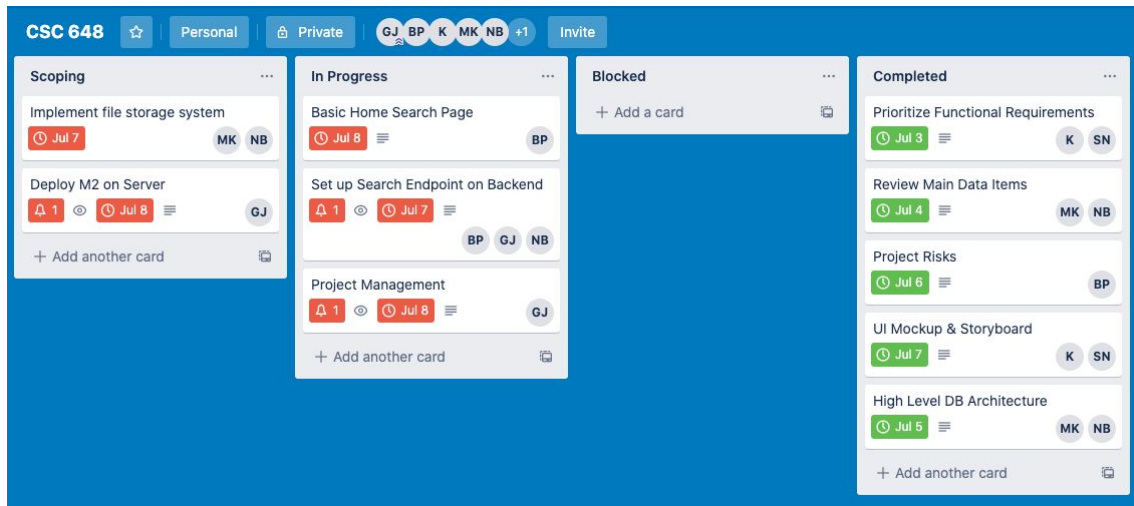
Project Risks

1. As the pandemic continues, real-life conflicts or unforeseen events could happen to the team and can delay the overall production speed. One absent member can lead the whole team to fall apart as the team no longer has the right mindset to continue or that they get angry because they have to take over another member's role. To counter this problem, we will start early and finish whatever we can before anything could happen to avoid as many issues as possible. We will also need to make sure that each member is ready to take over any member's part in case if anything happens.
2. Since we are no longer able to meet up with each other in real life, lack of communication and engagement might occur as we can only meet online via Zoom. Members can become too shy to speak up about their problems or too afraid to ask for help. By using Slack, we don't need to force people to speak up and minimize any fear or shyness that people might have since typing is less pressure for many people. We can also improve our team disengagement by using Slack since we can chat, share images, or post any discussions that we want to share without the need to use email to reach out to each other.
3. Not everyone in the team is familiar with the tools that we are using for this project and required some time to familiarize themselves with the tools before they get to start building the product. Especially when everything is online today, learning by themselves can be difficult for some people and will require a lot of time to learn the basics. If everyone is taking a long time to familiarize themselves with the material, then we will have less time for team discussing or review sessions as we have to be rushing to finish our product before the deadline. As a result, our final product might be poorly made and have tons of bugs. Learning for many people can get a lot easier if they can get some help from other people. As a team, we need to make sure that everyone is on the same page with whatever they need to learn and help each other out if possible. We also need to give those who need to learn a suitable and comfortable learning time to get familiar with the materials to ensure that they know what they are doing rather than letting them copy and paste.
4. As we add more and more features to the product, we spend more time and effort on writing codes for them rather than testing out our current product. Lack of testing could lead our final product to become unstable and not function correctly. To avoid this problem, we will test and review our product multiple times to ensure that everything is fully working before we add a new feature to it. Doing so will allow us to have more time to fix any bugs we have before the deadline and be less stressed when we add more features to it.

Project Management

In order to stay aligned and execute effectively, we are doing the following:

- Tracking all of our major tasks on Trello



- Meeting after class on Monday/Wednesdays in a “standup” fashion to discuss completed tasks, current tasks, and future tasks.
- Team lead hosts working sessions to collaborate and help each group member share knowledge. This can also create more buy-in from team members.
- Assign one team member a lighter load and use them as a “flex” worker who can go around and help others who have become blocked.
- Make sure the team has consensus on any new tools added to the tech stack
- Engage in rigorous PR review

Milestone 3 review summary and plans

CSC 648-848 Summer 2020 team 5

M3 Review Date: 7/29/2020

Date of this summary and plan document: 7/30/2020

Summary of feedback and tasks to do:

After review with the Professor, we noticed that there are many opportunities for improvement to further our application and make it more functional. Some of the tasks that were pointed out varied from our Search/Posting pages to a more user friendly interface. As we went page by page, there were a lot of new things that we missed or haven't yet implemented. Starting first with our home page, no title was provided and a logo was missing to signify our design. Secondly our search interface was a little wonky. We needed to clean up our search bar by moving it to the top and nesting it within our navigation bar to provide searching capabilities throughout. This also ensured that the user could easily find it and access it without too much thought. It was also pointed out that we should have the newest listings be provided on our home page at the top to make it easily accessible and clean up the design. 3-5 listing is more than enough to provide this new feature.

From there we went onto the searching capabilities to show the listings in our database. Within the listings page the Professor pointed out that there was a lot of whitespace. To clean this up he suggested to either add more info/attributes to the listing cards or even put two listings side by side and make each row two cards instead of one. We also noticed, similar to our home page, that the search bar needed to be nested into the navigation bar to provide easier access and better usability. It was also suggested that when a user is searching for a listing to have the number of listings called out at the top to provide the user with how many results have been found for what they are looking for.

Moving onto the postings page, which also had areas of opportunity. We originally designed to have the user go through three steps to set up a new listing/posting. The professor suggested getting rid of the next button, add a title to this page, and put everything all on one page. This ensures that the user does not lose focus on the task at hand. Another area for opportunity is when a user creates a new posting we have an automatic reply telling the user that their posting will show up on the application in the next 24 hours. We also noticed that there was no way for the user to access their dashboard from this page and that we should create a new tab in the navigation bar or a dropdown for easy access. The dashboard will be where the user can access their messages and also see the posts that they have created. We had the message dashboard implemented, but needed to add a split screen for the user to see both of these entities.

After taking a look at the main pages of our application we moved onto our groups Github and MySQL database. Our database looked well designed with appropriate, easy to use naming conventions. So no suggestions were made, just to keep up what we were doing with that. Github was similar, the professor just wanted to ensure that our commits were thorough and easy for someone else to know what we were implementing for each commit. The feedback we

received was to keep up on what we were doing and also be descriptive when committing our code.

Primarily, this was a very helpful Milestone. There was a lot of constructive feedback provided to make our application more functional. After taking a look at all the features and the suggestions our primary tasks are focusing on the User interface and making all our pages flow better. As listed above there were many things pointed out that we have taken into consideration and are working on.

List of tasks team chose to focus on and implement for final delivery:

1. All users shall be able to search and view listings.
2. All users shall be able to filter the listings by address, cost, bedrooms, unit type and offer type
3. Users are required to register to contact seller or create a new listing
4. Users shall register or login using their SFSU email and password
5. Users shall provide information of the listing like the cost, address, pictures of the rooms, area, offer type and contact information
6. System shall store all data of users and room listings in a secure database
7. System shall show all descriptions of listings to all users
8. System shall be able to provide a secure way to communicate
9. Registered users shall be able to check their dashboard for any new messages
10. System shall store all the messages between the users in the database
11. Users shall be able to view all the conversations with other users with most recent one on the top
12. Registered users shall be able to edit or delete their post
13. Users shall be able to view all recent posts on the home page
14. Users shall be contacted/be able to contact by one in-site message or by email provided

List of product P1 functions agreed at meeting

1. All users shall be able to search and view listings.
2. Users are required to register to contact seller or create a new listing
3. Users shall register or login using their SFSU email and password
4. Users shall provide information of the listing like the cost, pictures of the rooms, and contact info
5. Users shall provide information of the listing like the cost, pictures of the rooms, and contact info
6. All passwords will be securely encrypted
7. System shall not support any type of payment system.

8. System shall store all data of users and room listings in a secure database.
9. System shall show all descriptions of listings to all users
10. System shall be able to provide a secure way to communicate
11. Users shall be contacted/be able to contact by one in-site message
12. Search function shall be accessible anywhere on the page
13. Each page shall have a page title
14. Home page shall show a few latest postings
15. Posting page shall be one page long

Milestone 4

SW Engineering CSC648/848 Summer 2020

Housing State

8/3/2020

Team 5

- Garrett Johnson (lead)
- Buu Phan (github)
- Sarah Nafees (frontend)
- Mantasha Khan (backend)
- Nick Brown
- Kamelia Shaharova

Product Summary:

Name: Housing State

Committed functions

- All users shall have the capability of searching and viewing listings
- All users are required to register to contact seller or create a new listing
- Users shall register or login using their SFSU email and a password
- All users shall be able to access search from every page
- All users shall be able to reach out to renters and sellers through a message dashboard
- Users shall provide information of the listing like the cost, and picture of the house
- Registered users shall have a dashboard page where they can see their messages along with the listings they have posted.
- Users shall be contacted/be able to contact by one in-site message
- System shall not support any type of payment system.
- System shall show all descriptions of listings to all users

URL: [Housing State](http://3.17.193.189/) (http://3.17.193.189/)

Usability Test Plan

Test Objectives

The main motive of this evaluation is to assess and evaluate the Housing State website. The major function that we have selected for this usability testing is the search function. The usability test is done to evaluate if the product is user friendly, easy, usable, and fast. The test objectives for the usability study is to evaluate the search function on Housing State website which should be relative to the user's ability to:

- Search available housing either by address, city, or ZIP code
- Search desired listing using a filter

Through this testing, we are also expecting some room for improvements on usability and functionality which we can do by getting feedback from the users and evaluating the user's perception.

Test Background and Setup

The test is designed to evaluate the Housing State website in terms of user ability to easily access and search the desired listings. The website must be able to run in multiple browsers which are Google Chrome, Mozilla Firefox, and Safari. The intended users of this website are SFSU students and faculty.

To begin the usability test we will use the Google Chrome browser and start the user off at the home page of the website. We will then give the required task for the user to complete, but will not provide any assistance in completing those tasks. Once the user completes the tasks we will use the criterias in Usability Task Description table and the Likert test to determine whether the function passes the usability test.

The URL of the website is [Housing State](#) which is also the URL of the search page we are testing. The user can input the valid search query which can include the exact address, city, or ZIP code, and can also filter out searches by using the dropdown menu lists. The usability test is successful and completed if the user finds the appropriate result by searching the desired house listings available. For example, when a user opens the Housing State website and searches for all available listings in San Francisco, the user will be able to view the options related to their search query. After loading the appropriate search result according to their criteria, we can say that our usability test has been successfully completed.

Usability Task Description

TASK	DESCRIPTION
Task	Search listings in San Francisco

Machine state	Search listings loaded
Successful completion criteria	Shows related search results according to the search query.
Benchmark	Completed in 10 seconds

Effectiveness: I would measure effectiveness by looking at the percentage of people who completed the defined task in the time frame, the errors counted per task, and would allow comments or suggestions on the effectiveness of the test.

Efficiency: I would measure efficiency by looking at the average time it took to complete the task and find out the average time of those who completed the task/ average time of all users. Efficiency in effort could be known by the number of clicks it took to search a listing.

("X" applicable column)

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I was able to find the search bar easily.					
I was able to search for the desired listings.					
I found a related search result.					

QA test plan:

- Test objectives
 - Assessing the website's content through its search feature for both registered and unregistered users while validating the accuracy of the search results.
- HW and SW setup
 - Hardware:
 - MacOS running Docker
 - System setup:
 - The setup consists of first using the Google Chrome browser to run the website where we begin on the home page which is also the search page. After the application runs correctly we check if the search bar and search button exists and if some of the demo content exists. We then use the Firefox browser to run the website where we begin on the home page which is also the search page. After the application runs correctly we check if the search bar and search button exists and if some of the demo content exists.
 - Intended users are the students and faculty of San Francisco State University
 - URL: [Housing State](#)
- Feature to be tested:
 - Search function
- QA test plan (suggested format for QA test plan table)

When we performed the test, we followed the table and tested the search feature on two different types of browsers (Chrome and Firefox)

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Chrome	Firefox
1	Search Bar Exists	The search bar should be visible in the main page of the application at all time	N/A	Visible Search Bar	PASS	PASS
2	Search Button Works	The search button should update the main page content when pressed	Type "San Francisco" in the search bar and press search	Update in the main page	PASS	PASS
3	Search Results are Relevant	The main page results should be relevant to the	Type "San Francisco" in the search bar and press search.	Should show listings that contain "San Francisco" in address (3 listings)	PASS	PASS

Code Review:

Done over email as specified. Screenshots below

Code review done over email as specified. Original code in black, review comments in **red**

```
import { Form, Formik } from 'formik';
import axios from 'axios';
import { TextInput, Select } from '../utils/inputs';

// good use of functional components
// perhaps we should be consistent with our backend code and use async/await instead of these promises
const searchbar = ({ setListings }) => {
  const searchListings = (query, { setSubmitting }) => {
    axios
      .get('/api/listing', { params: query })
      .then((res) => {
        setListings(res.data);
      })
      .catch((error) => {
        console.log(error);
      })
      // good use of finally!
      .finally(() => setSubmitting(false));
  };
};

// insert comment about these being default values
const formData = {
  search_term: '',
  unit_type: '',
  offer_type: '',
  bedrooms: 0,
  cost: 0,
};
```

```

return (
  // good use of dynamic react sizing!
  <div className='row container-fluid align-item-center justify-content-center'>
    <div className='col-md-6'>
      <div className='card mb-4'>
        <div className='card-body'>
          <Formik
            initialValues={{ ...formData }}
            onSubmit={ (values, { setSubmitting }) => {
              setSubmitting(true);
              searchListings(values, { setSubmitting });
            }
          >
            {{{ isSubmitting }}} => (
              <Form>
                <div className='form-group'>
                  <TextInput
                    name='search_term'
                    className='form-control'
                    placeholder='Enter an address, city, or ZIP code'
                    maxLength='40'
                  />
                </div>
                <div className='mt-5 mb-1'>
                  <p className='text-left text-muted'>Advanced Search</p>
                </div>
                // maybe the "advanced search" features should be hidden by default?
                <div className='form-row'>
                  <div className='form-group col-sm-3 text-left'>
                    <label className='label-text' htmlFor='unit_type'>
                      Unit Type
                    </label>
                    <Select
                      id='unit_type'
                      name='unit_type'
                      className='form-control border drop-text'
                    >
                      <option value=''>All</option>
                      <option value='house'>House</option>
                      // Should be spelled "apartment"
                      <option value='apartment'>Apartment</option>
                      <option value='townhouse'>Townhouse</option>
                    </Select>
                  </div>
                  <div className='form-group col-sm-3 text-left'>
                    // maybe call this "Buy or Rent"
                    <label className='label-text' htmlFor='offer_type'>
                      Offer Type
                    </label>
                    <Select
                      id='offer_type'
                      name='offer_type'
                      className='form-control border drop-text'
                    >
                      <option value=''>All</option>
                      <option value='buy'>Buy</option>
                      <option value='rent'>Rent</option>
                    </Select>
                  </div>
                </div>
              </Form>
            )
          </div>
        </div>
      </div>
    </div>
  </div>
)

```

```

<div className='form-group col-sm-3 text-left'>
  <label className='label-text' htmlFor='bedrooms'>
    Bedrooms
  </label>
  <Select
    id='bedrooms'
    name='bedrooms'
    className='form-control border drop-text'
  >
    <option value=''>All</option>
    <option value='1'>1</option>
    <option value='2'>2</option>
    <option value='3'>3</option>
    <option value='4'>4</option>
    <option value='5'>5</option>
  </Select>
</div>
<div className='form-group col-sm-3 text-left'>
  <label className='label-text' htmlFor='cost'>
    Cost
  </label>
  <Select
    id='cost'
    name='cost'
    className='form-control border drop-text'
  >
    <option value=''>Any</option>
    <option value='750'>$750</option>
    <option value='1000'>$1000</option>
    <option value='1500'>$1500</option>
    <option value='3000'>$3000</option>
  </Select>
</div>
</div>
<button
  // Really cool that search button is disabled until the response is received. Is this
  // to prevent spamming the submit button?
  type='submit'
  disabled={isSubmitting}
  className='btn btn-dark float-right mt-3'
>
  Search
</button>
</Form>
  })
</Formik>
</div>
</div>
</div>
</div>
  );
};

export default searchbar;

```

```

const Listing = require('../models/listing');
const Sequelize = require('sequelize');
const Op = Sequelize.Op;

// Nice job specifying the route of the endpoint
// route: GET /api/listing
// It looks like "next" is not used, maybe we should remove it?
exports.searchListings = async (req, res, next) => {
  try {
    let where = {
      // is the "and" condition here applying to each item below? could you be more specific here?
      [Op.and]: [
        // match and address and city or zip and unit_type and offer_type and bedrooms and cost
        // non-numeric fields use like so that we can match on a blank value ''
        { full_address: { [Op.like]: `>${req.query.search_term}` } },
        { unit_type: { [Op.like]: `>${req.query.unit_type}` } },
        { offer_type: { [Op.like]: `>${req.query.offer_type}` } },
        { bedrooms: { [Op.gte]: req.query.bedrooms } },
        { cost: { [Op.gte]: req.query.cost } },
      ],
    };
    // good job defining the "where" above, it makes it much cleaner just reference it here
    const listings = await Listing.findAll({ where: where });
    res.send(listings);
    // thanks for catching any potential errors
  } catch (error) {
    console.log(error);
    res.sendStatus(500);
  }
};

```

Self-check - security:

Major assets

- User information – user information is a top priority. With many threats such as confidentiality of a system, and integrity, we took this seriously. We used a hook that takes the users password and encrypts it into hash.
- Listing information – this was also another top priority. Just like user information, threats of confidentiality and integrity of our system could be prevalent without the correct precautions. Since User id is a foreign key for our system and as stated above, we are using hash encryption to protect them. Our Databases are also password protected, so without the correct password one cannot access our MySQL database.

Security is a primary thing and needs to be taken seriously. We also were coming up with checks and balances for admin approval on all listings, but the professor told us to drop this and only work with it if we got everything else done.

Within our backend we ensured that when a new user signs up their password is converted into hash code so that it will be private only to that user. Since the data is encrypted in the backend, the real password never touches the database, only the secure, hashed password is entered.

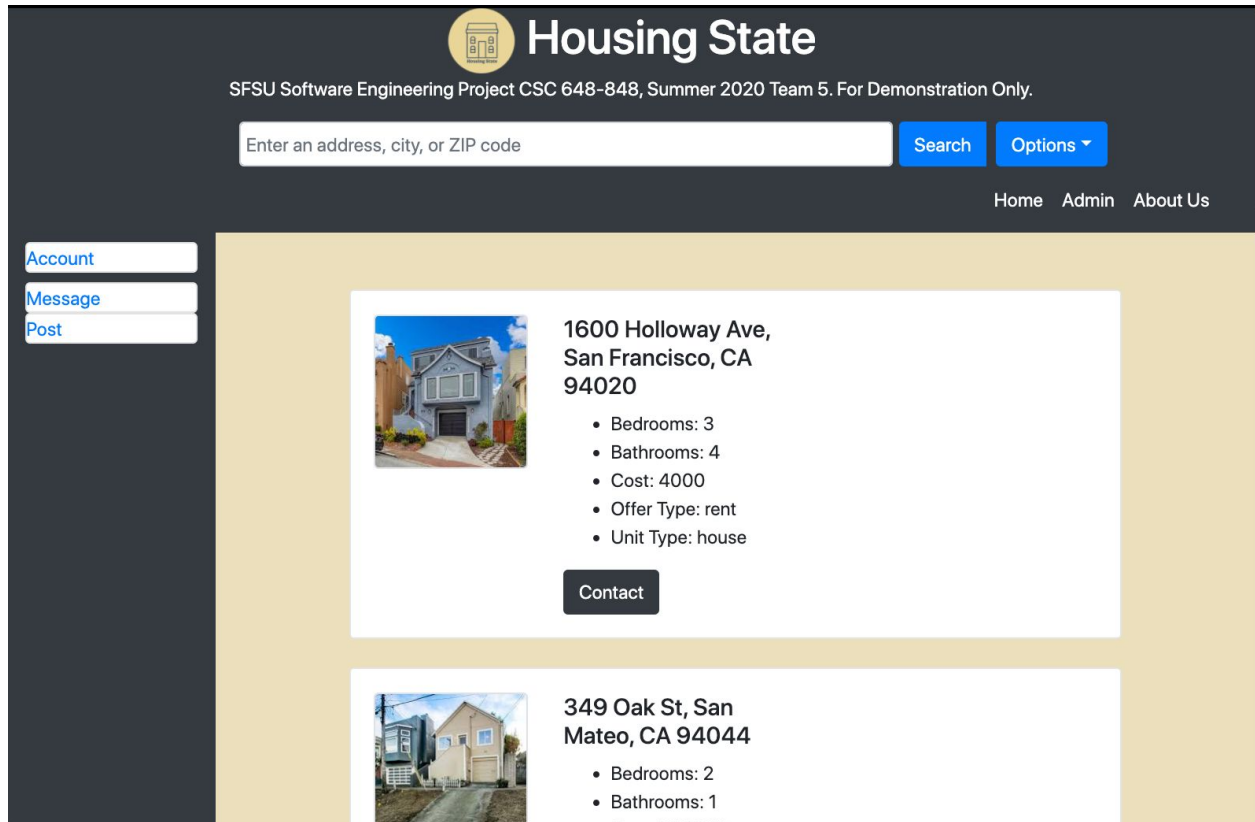
Adherence to original Non-functional specs

Specs from Milestone 1


Spec	Status	Comment
The application shall be in English.	DONE	
The application shall be simple and user friendly.	ON TRACK	
The application shall display the exact text "SFSU Software Engineering Project CSC 648-848, Spring 2020" on top of the page.	ON TRACK	
The application shall be deployed in AWS	DONE	
The application shall render well in both desktop and mobile browsers	DONE	
The application shall only be for SFSU students and teachers.	DONE	
All users' data shall be stored in MySQL	DONE	
All users' data shall be encrypted and protected equally. Only the user themselves can review and edit their data.	DONE	
All posts shall be reviewed by admins before posting.	ISSUE	Since we were slightly behind during our M3 review, you suggested we skip admin functionality. This is the first thing we will do if we finish our P1 requirements
Google API shall be used.	ISSUE	We decided that integrating the google API is a P2, so we will do this only if we have time
No payment function shall be added.	DONE	
Best practices for security and privacy shall be applied.	ON TRACK	

Product Screenshots

Home Page



Message Page



Housing State

SFSU Software Engineering Project CSC 648-848, Summer 2020 Team 5. For Demonstration Only.

SearchOptions ▾

[Home](#) [Admin](#) [About Us](#)

[Account](#)[Message](#)[Post](#)

Chat		sfsu
sfsu	2020-08-07	<div>2020-08-07sfsu</div> <div>Hey I like your house!</div>
		<div>2020-08-07sfsu</div> <div>When can I come by and see?</div>
		<div>2020-08-07sfsu</div> <div>New House who dis!?</div>
		<div>2020-08-07sfsu</div> <div>Hey thanks. Do you want to come by and see it?</div>

Post Listing

Message

Post

Post a Listing

Building No.

Street

City

State

Zip


Unit Type

Pick

Offer Type

Pick

Admin Panel



Housing State


SFSU Software Engineering Project CSC 648-848, Summer 2020 Team 5. For Demonstration Only.

[Home](#) [Admin](#) [About Us](#)

[Account](#)

[Message](#)

[Post](#)



**569 Vernon St,
Oakland, CA 94101**

- Bedrooms: 3
- Bathrooms: 4
- Cost: 700000
- Offer Type: buy
- Unit Type:
apartment

About Us



Garrett Johnson

[About Me](#)



Sarah Nafees

[About Me](#)




Buu Phan

[About Me](#)



Sign Up

 **Housing State**

SFSU Software Engineering Project CSC 648-848, Summer 2020 Team 5. For Demonstration Only.

[Search](#) [Options ▾](#)

[Home](#) [Login](#) [Sign Up](#) [About Us](#)

Sign Up

First Name

Last Name

Email

Password

[Sign Up](#)

Database Organization

Users

100%		1:2						
Result Grid		Filter Rows:	Q Search	Edit:	Export/Import:			
id	email	first_name	last_name	password	type	createdAt	updatedAt	
1	sfsu.student@mail.sfsu.edu	sfsu	student	\$2a\$08\$q3GU5pmeTytN66P192Y1wuzYUQGwBMd72bRRNiZf4ZsXs9mgt...	admin	2020-08-07 21:17:02	2020-08-07 21:17:02	
2	john.doe@mail.com	john	doe	\$2a\$08\$emv6w1YxKhm2BbbCJyUhOU7dZ54ZAYaKSjC.FDlpZxzV2OI5.3l2	student	2020-08-07 21:20:53	2020-08-07 21:20:53	
3	other.student@mail.sfsu.edu	other	student	\$2a\$08\$M8tFik9kEd4voJhC8Lvlen4SSPBzrwedcW3Ei2/ksm4rum4FdOa2	student	2020-08-07 21:26:39	2020-08-07 21:26:39	
4	random.person@mail.com	random	person	\$2a\$08\$5QsVAbYL3zhtK5h0nhevYeSdeHTxdAaBE9jvqN1fs986.GCWqAq46	student	2020-08-07 21:38:39	2020-08-07 21:38:39	
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

Listings

Result Grid

Filter Rows:

Q Search

Edit:

Export/Import:

id	user_id	building_num	street	city	state	zip_code	apt_suite	cost	unit_type	lease_length	offer_type	bedrooms	bathrooms	furnished	sq_footage
1	1	1600	Holloway Ave	San Francisco	CA	94020	NULL	4000	house	4	rent	3	4.0	NULL	1500
2	1	349	Oak St	San Mateo	CA	94044	NULL	500000	apartment	12	buy	2	1.0	NULL	1200
3	2	987	Loyolla St	Daly City	CA	70404	NULL	4500	apartment	4	buy	2	2.0	NULL	2100
4	3	437	Poplar Ave	South San Francisco	CA	94403	NULL	700	apartment	1	rent	3	2.0	NULL	340
5	4	569	Vernon St	Oakland	CA	94101	NULL	700000	apartment	4	buy	3	4.0	NULL	3500
6	4	665	Nice Dr	Daly City	CA	94002	NULL	500	townhouse	6	buy	2	3.0	NULL	2400
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Messages

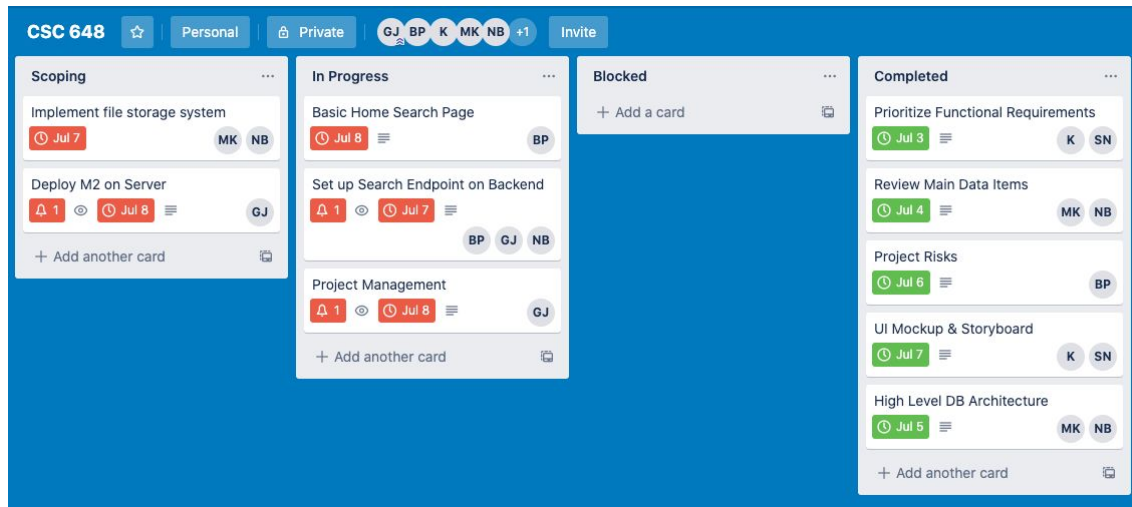
Result Grid		Filter Rows:	Q Search	Edit:	Export/Import:			
id	from_user	to_user	text	createdAt	updatedAt			
3	1	1	Hey thanks. Do you want to come by and see it?	2020-08-07 21:25:20	2020-08-07 21:25:20			
4	4	1	Hey I really like this house!	2020-08-07 21:47:59	2020-08-07 21:47:59			
5	4	1	Wow what a great house!	2020-08-07 21:48:08	2020-08-07 21:48:08			
6	4	1	Nice houes!	2020-08-07 21:48:16	2020-08-07 21:48:16			
7	2	1	Can I come by and see this house?	2020-08-07 21:49:52	2020-08-07 21:49:52			
8	2	1	I'm interested!	2020-08-07 21:50:02	2020-08-07 21:50:02			
9	3	1	Hey I'm interested In this!	2020-08-07 21:53:45	2020-08-07 21:53:45			
10	3	1	Can I come by and see!	2020-08-07 21:53:55	2020-08-07 21:53:55			
11	1	1	New House who dis!?	2020-08-07 22:01:03	2020-08-07 22:01:03			
12	1	1	When can I come by and see?	2020-08-07 22:01:15	2020-08-07 22:01:15			
13	2	1	Wow I really like this house!	2020-08-07 22:02:10	2020-08-07 22:02:10			
14	2	1	Can I see this in person?	2020-08-07 22:02:45	2020-08-07 22:02:45			
15	1	1	Hey I like your house!	2020-08-07 22:20:56	2020-08-07 22:20:56			
NULL	NULL	NULL	NULL	NULL	NULL			

Google Analytics


(site not live on AWS yet - Will update this doc once we do)

Project Management

Trello



Slack

 **Garrett Johnson** 5:51 PM
TODOs:

Milestone 3

- **Everyone** - Add more bullets to the end of the [new Milestone 3 doc](#), lets try to get 30 items. See the M3 doc on iLearn for more.
- **Nick** - Summary of feedback and tasks to do. 1 page
- **Mantasha** - List of tasks team chose to focus on and implement for final delivery. 1 page
- **Buu** - List of product P1 functions agreed at meeting. 1 page

Milestone 4



- [Doc here](#)
- **Nick** - Product Summary. 1/2 page
- **Sarah** - Usability test plan. 1.5 pages
- **Kamelia** - QA test plan. 1.5 pages
- **Garrett** - Code review
- **Nick** - Security 1/2 page
- **Garrett** - Self check 1 page

Deadlines

- **7/31** - Completed Milestone 3 doc
- **7/31** - A slack from everyone with their action items + when they will be done by!! (post in this channel) [PLEASE DO THIS](#)
- **8/3** - Completed Milestone 4 doc

(edited)

👍 2 🗨️

  **2 replies** Last reply 7 days ago

Meeting Minutes

Important Links

- [Repo](#)
- [Trello](#)
- [Zoom](#)

6/17

Updates

- About page and all subpages are done
- Website is hosted on AWS

Next Steps

- Install MySQL
- Divvy up M1 requirements
 - Executive Summary: Nick 18th
 - Personae and main use cases: Garrett 19th
 - List of main data items and entities: Mantasha
 - List of functional requirements: Sarah N
 - List on non-functional requirements: Buu 20-21
 - Competitive analysis: Kamelia 20-21
 - High level system architecture 22
 - Team and roles
 - Checklist

6/15

Updates

Questions:

- Do we need a domain or just local hosting on AWS?
- **Answer:** local hosting on AWS through a port is fine
- Does every member need their own page or can we just dynamically change the content of a single page?
- **Answer:** every member should have their own page

Self Assessments

Garrett Johnson

Hey everyone,

Thank you for all the work you put into this project! It wasn't easy, but we made it through, and learned a lot in the process.

1. I was team lead on this project. I ran meetings, doing my best to make sure we were all aligned and that we came away with clear action items. I laid down the frontend + backend framework to give everyone a good starting point, and continued to heavily contribute to the codebase thereafter. I provided feedback and direction on all milestones (technical and non-technical). I offered myself as a resource and hosted working sessions to try and keep everyone up to speed.

2. 66 commits to master. Over 100 commits across all branches.

3. In my opinion, my biggest challenge (which became my biggest shortcoming) was failing to establish strong buy in from everyone early on. Given the fast pace nature of this class, anyone who becomes a straggler will have difficulty catching up, and I did not do an adequate job ensuring everyone was on the same page.

When coding up the initial frontend + backend framework, I felt trapped between a rock and a hard place. I wanted to do a lot of work to make sure we had a rock solid foundation from which to build off of that wouldn't give us any trouble down the line, but I also didn't want to start off doing too much and leave my teammates feeling like they were not involved early in the process, or put them in a position where they were overwhelmed by the technical aspects of the project from the start.

In retrospect, I feel that I did play too heavy handed a role in the frontend + backend setup, and that caused my teammates to become overwhelmed and lose buy-in.

Coordinating this project over Zoom was also very challenging, and I struggled to build a strong rapport with my teammates in the short time we had together.

4. This class taught me the importance of teamwork above technical expertise. Based on my experience in this class I would have more checkin meetings to keep us organized and on track, and more semi-structured working sessions where we could share knowledge and build camaraderie.

Buu Phan

Hello friends,

Now that the suffering is finally coming to an end, I would like to take this opportunity to thank everyone for your hard work and dedication throughout this semester. Special shoutout to Garrett for being such an incredible team leader who did way too much on his own to ensure that we complete and turn in our assignment on time. The project was overall pretty stressful for me as there were a lot of things to do, but I had fun coding it nevertheless. It was nice working with you all, and I wish everyone the best of luck in all your future endeavors!

A. I mostly worked on the frontend team throughout the semester and rarely touched any of the backend stuff. For the final project, I made the message/chat component and styled some of the pages for our site. I also did a few milestone reports here and there. I also helped Garrett look over our Github and make sure every pull request is working correctly before merging to the dev branch.

B. I made about 4 pull requests.

C. One of the main challenges I have working with the frontend is becoming familiar with React and Bootstrap. Although I have used both React and Bootstrap in my previous class, there are still a lot of things I don't know and needed time to utilize their utilities. I also struggle with time management and always overestimate the time needed for my part. I tend to do assignments last minute, so I always stay up late at night to make sure I will be able to finish my part before the deadline.

D. After going through all this teamwork for the past three months, I learned that I need to take care of my teammates and help them with their part more whenever I have a chance. During some time of development, when I finished my part early, I didn't care about how much progress my teammates have made and thinking that they will get it done eventually. This selfishness later leads to consequences as we almost didn't finish our project before it was due and would have been a disaster to turn in an incomplete project. I could have helped my team a lot earlier when they were struggling with their part while I had a lot of free time. In the future, I will try to finish my part as soon as possible so that I can spend the rest of my free time helping my teammates out whenever they are struggling to make sure that we finish our project days before it is due.

Again, thank you everyone for your great work for the past 3 months. Best of luck in the future!

Best,
Buu Phan

Sarah Nafees

Hey guys,

Wanted to start off by appreciating my team for all the hard work and effort everyone put in toward this project. We all had our challenges but it is good to see what we were able to accomplish and the knowledge we gained in such a short amount of time, despite everything going on around us. It wasn't easy for any of us but we definitely learned a lot through teamwork.

I worked on frontend aspects of this project. I came up with the initial UI design for the website. We created our prototype using Figma, and made contributions in creating the UI of our website. I completed the milestone tasks assigned by our team lead which were to create prioritized functional requirements, the mockup and storyboards of the UI, and a usability test plan.

I made about 3 pull requests.

My main challenge was my unfamiliarity with JavaScript. I came to this class with a little bit of experience in React android development but the approach taken in a team where things have to be done a certain way was where I encountered challenges. I also had trouble figuring out MySQL, but thanks to my members' help, I was able to work around the problem to finish my given tasks.

There are many things I could have done better. Starting with managing my time and taking the time to hone my JavaScript skills before diving into this project. In order to prepare for the class I took up a React course but was not able to be consistent with it due to the workload of my 3 other classes, along with other personal commitments. But I learned that communication is important and can make things a lot easier. It's always better to talk about the problems so there is room for understanding in team members and you can come up with solutions as a team.

Thank you,
Sarah Nafees

Kamelia Shaharova

Hey guys,

I wanted to say thank you so much for an amazing team, I learned a lot in this challenging course. Thanks so much to everyone and Garrett who was an amazing team lead! Good luck with the rest of your finals, and hope you all have a good break until the Fall!

1. I was on the frontend team. We started off creating the design of the website using Figma, and then I contributed to the UI of the website. The milestone tasks were assigned to us by our team lead, I worked on the QA test plan, UI Mockups & Storyboards, and competitive analysis in terms of milestones.

2. The number of pull requests I made was about 2 I believe.

3. My main challenge I encountered was using React and becoming familiar with that. I'm also not the strongest programmer so it was difficult in general, but I was definitely challenged in this class and learned a lot. Our team lead, Garrett, contributed a lot and was really helpful in explaining things when I needed assistance.

4. In the future, I will try to manage my time better and learn all of the necessary tools prior to the class. I had a heavy load this semester and began an internship so I was pretty busy this summer, but found this class to be a good challenge - I learned more about frontend work and teamwork. I had a lot of fun with the UI design and doing my part on the milestones and believe this class helped me further figure Out what I would like to do in the future.

Best,
Kamelia Shaharova

Mantasha Khan

Hello everyone,

First of all, thank you to everyone for your contribution in this project. This project has been the most stressful amongst all my other 3 classes this summer for me. But I'm happy that we're almost there. And I wish you all good luck with everything in future.

I worked mostly on the backend of the project being the backend lead. I set the database in SQL with the help of our team lead. Made the tables for all the required data entities. Our team lead mostly assigned tasks for each Milestone. For the milestone documents I worked on main data items and entities, list of tasks team chose to focus on and implement for final delivery and made changes according to the feedback provided. I also contributed towards making endpoints for messaging functionality. Also, I made the form for new listings.

Since the code was not directly committed to the original dev branch instead separate branches with pull request was created prior to merging any code in the original dev branch. The number of submissions I made to the github dev branch should be around 4.

The main challenge I encountered for this project was working with new tools mainly sequelize. I never learnt this part of MySQL before it was really hard to work with it. I did a lot of research to learn it in order to help my team, but I think I still need more practice to fully understand the stuff. Although our team lead worked really hard and helped us throughout the project.

In future I will try to learn any new tools that are about to be used in project beforehand so that at the time of development I would not need to spend much of energy and time learning it.

Thank you,

Mantasha

Nick Brown

Hey guys,

For milestone 5 we all need to send an email to one another talking about our contributions that we did for this project. So I'm just going to list them out while trying to be short and sweet. I also wanted to say thank you to everyone. This project was extremely stressful for me, so I know other people were probably feeling this as well, but we gave it our all and thank you guys for all the effort that everyone put into it.

8) Team member self-assessment and contributions

a.) For this team project, I worked on the back-end side of things. This was extremely difficult due to not knowing SQL and not being super familiar with our stack. I originally worked on how to store the images with BLOB formatting using sequelize and storing it into our MySQL database. Last minute we decided to change this and store the images differently via the relative path. I also helped on different reports for each Milestone. Our Team lead assigned us with different roles for each one of those. Following that I worked on the back-end for users. By helping with the hash algorithm to change user's passwords into hash, writing the code for the APIs within, and worked on the user authentication.

b.) The number of submissions that I contributed to the dev branch I believe was 3 or 4. I am not hundred percent sure if I looked this up correctly. But my code was also merged into others before going into the dev branches.

c.) For myself, this project was extremely hard and stressful. I am working full time and had to learn almost everything that was being presented on my own. I spent many hours doing research and also trying to learn new applications such as Postman, which was a huge help, but took a lot of time for myself to learn. I want to give a huge shout out to Garrett, our Team Lead, for spending a lot of hours outside of class showing us new things and helping us throughout this project, and also my team for really giving it their all throughout the process.

d.) I think I could've contributed more in the front end side of things. I felt like front-end did a huge amount of work. I also had a lot going on in life, so I wish I had more time to contribute to the project itself. But it was a great learning process and I learned a lot from it.

Thanks guys and it was a fun semester.

Nick Brown