

SW Engineering CSC648/848 Spring 2019
GatorState
Team 6

Team Members:

Team Lead: Rowvin Dizon (email: rdizon1@mail.sfsu.edu)

Front End Lead: Jonathan Gurdal

Front End: Daisy Sun

Back End Lead: Marlon Johnson

Full Stack: Minho Cha

Full Stack: Rene Elias

GitHub Master: Kayla Musleh

Milestone 2
April 2nd, 2019

History Table

| | |
|------------------------|----------------|
| Date Submitted: | 3/22/19 |
| Date Revised: | 4/2/19 |

1. Data Definitions V2

Classes:

- **Home Page:** Default page that provides access to login, navigation of site, and search
- **User Account:** Able to see basic info of listings, favorite/bookmark listings, can create listings. Can be created by Unregistered users. Unregistered Users can do this by going to the “Register” page. Here they can create a username and password for their account.
 - Database Elements:
 - Name: A string of User’s full name, can be displayed on User profile
 - Username: A string for User’s ID credentials that allows user to be found via search
 - Password: A string of characters to authorize user at login
 - Photo/Image: One .jpg or .jpeg file for User Profile Picture
 - Listings: A list that contains user’s own posted listings
 - Saved Listing: A list that contains user’s bookmarked listings
- **Listing:** Holds the basic and necessary information of a home. Can only be created by Registered User. Registered User can create this by clicking the option called “Add a Listing” and will then be redirected to a page where they can input all fields of necessary information about their intended listing.
 - Database Elements:
 - Address: String that holds the physical address/location of listing
 - Images: Multiple .jpg or .jpeg files of Listing
 - Home Type: Falls under either apartment, condo, house category.
 - Number of Rooms
 - Square Feet: Size of listing
 - Price: Cost of listing per month, or total overall cost.
 - User ID: A String of the original poster’s User ID.
- **Search:** Retrieves specific information from the database depending on User’s input. Unregistered Users and Registered Users can both use search by inputting the desired zip code and/or address into the search bar. Users can also filter search results by checking what filters suit their needs the most.
- **Messaging:** Gives buyers and sellers a means of contacting each other. Can only be used by Registered Users. Registered Users can access this by clicking on the “Messages” tab on the Dashboard. They can use it by inputting the name of the user they wish to contact and typing a text message to the intended receiver.
 - Database Elements:
 - SenderName: string of full name of User
 - ReceiverName: string of full name of user receiving messages
 - FromUser: string text message sent by the current user to another user
 - ToUser: string text message sent to the current user from another user
 - Timestamp: recorded time of when a message is delivered
 - Photo/Image: User’s Profile Picture shown along with user’s full name

2. Functional Requirements V2

Priority 1 (Must have):

Unregistered Users:

1. Search

- a. Partial Address
 - i. Zip Code
 - ii. City
- b. Full Address
 - i. Everything in partial
- c. Filter

2. Login/Registration

- a. Email
- b. Password

3. Filter

- a. Distance
- b. Price
- c. Type of Building

4. View Listing

- a. Square footage
- b. Price
- c. Address
- d. General Description
- e. Contact

5. Filter

- i. Distance
- ii. Price
- iii. Type of Building

Registered Users:

5. Rent

- a. Be able to see multiple listings.
 - i. Filters shall be applied to narrow down posting selection.
- b. When you want to rent a listing there will be a button called 'Request to Apply' where you submit your name, number, and email.

6. Sell

- a. being able to post listings
 - i. can go to a form that allows user to input posting details such as name, address, etc.
 - 1. At the end, if the user isn't logged in (or unregistered), they shall get a registration/login pop up.
 - a. The registration/login pop up requires username and password.
 - ii. After the user has submitted all the information, a notification will appear that their posting is being reviewed before being made public.

7. Messaging

- a. Ability to message the seller or buyer directly with direct messages.

8. Posting a Listing

Administrators:

1. Delete User Account

- a. shall be required to approve all pending posts before they go live

Priority 2 (Desired):

Registered Users:

1. Save Listing

- a. Save listings to your profile page to allow it to be viewed again later.

2. Account (Priority 2)

- a. Name
- b. Photo
- c. Personal Info/Bio
- d. View your listings
- e. Edit Profile
- f. Save Listing (Priority 2)

Priority 3 (Opportunistic):

1. Compare

- a. Ability to select two listings and compare them side by side to easily compare the different attributes of the listings.

2. Things to do

- a. General recommendations of things to do in the area based on your location.

3. Directions

- a. Different methods of transportation to get from your listing to SFSU.

4. Reviews

- a. Depending on the type of listing, you are able to leave reviews on the listing for everyone to read about the listing.

5. Tutorial

- a. Step by step instructions on the posting page to help less technical users post the listings with greater ease.

6. Feedback

- a. Feedback button to allow users to message the developers with feedback.

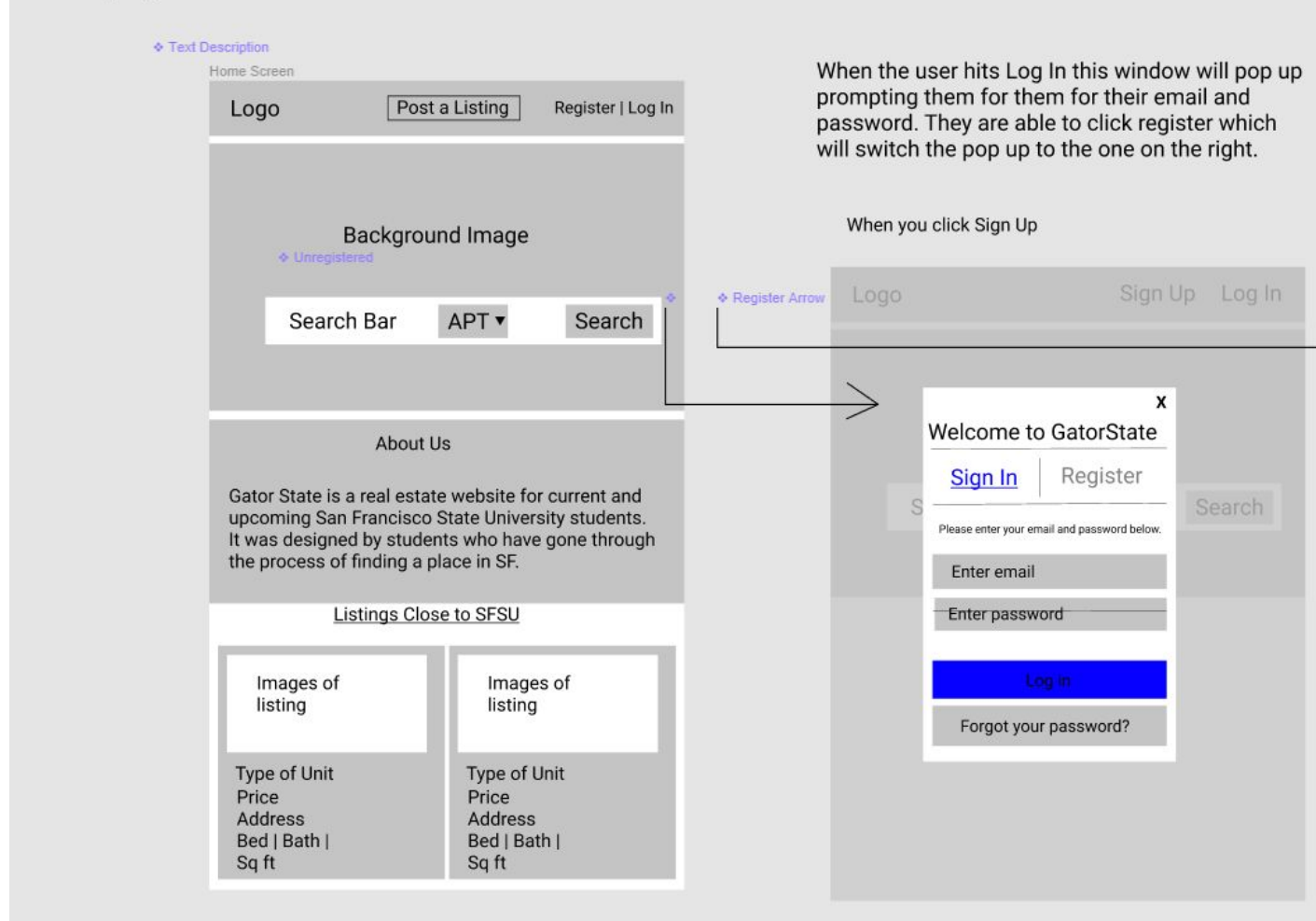
3. UI Mockups and Storyboards (high level only) : [View in Figma \(Recommended, 2 pages\)](#)

*If you choose to view in Figma, go from left to right

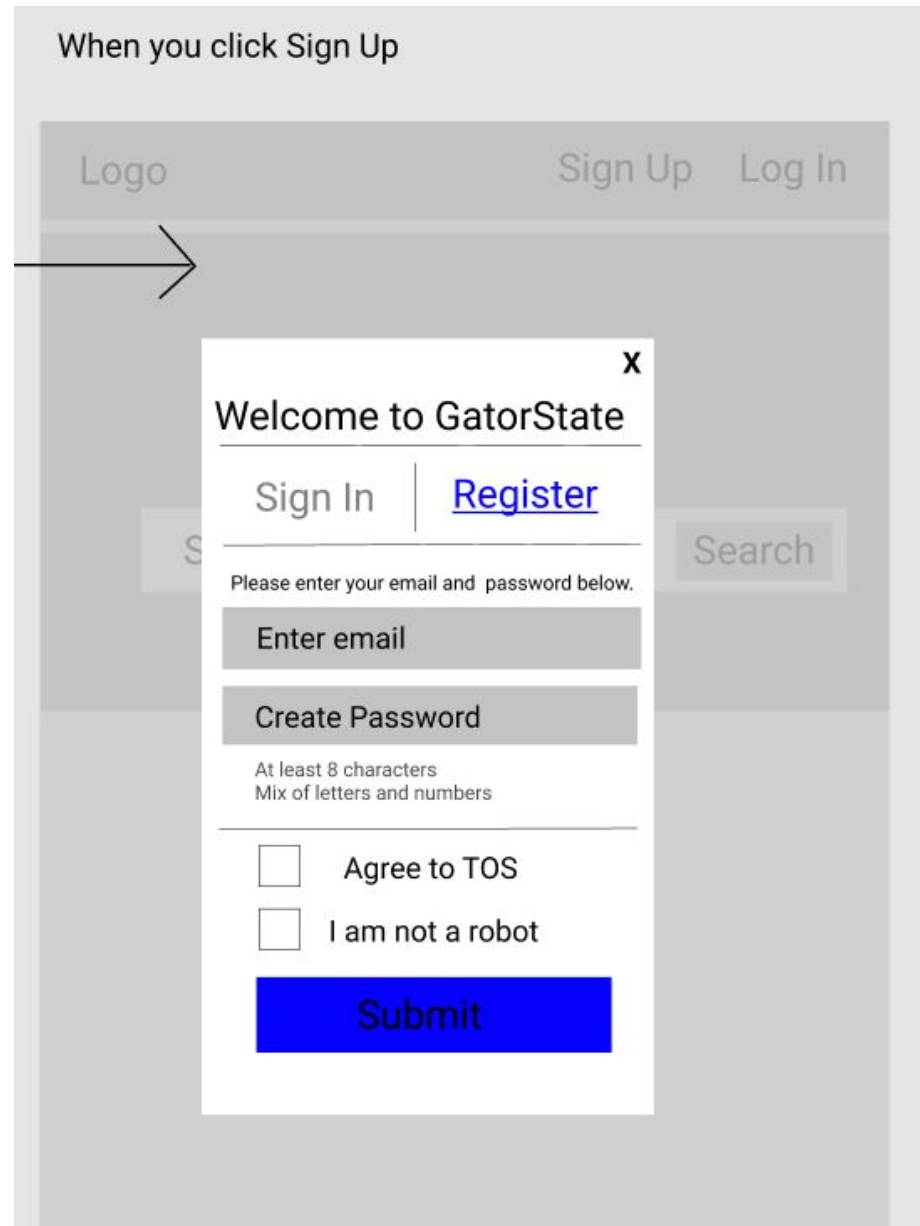
Unregistered Users

*Stretched to make it easier to view

Home page for UNREGISTERED USERS



Added



This is the the other page that will pop up if the user clicks register instead. You are able to switch between the two in the pop up.

[Back](#)

Click Here for assistance ?

Details and Description

* Address: Unit:

* Rent Price: * Home Type:

Beds Bath

* Lease Duration

Date Available

* Description

Property Type Dropdown

House

Apartment

Dorm

Condo

Contact Information

Name

Email

Amenities and Rules

Check all that apply (Optional)

☐ A/C

☐ Balcony/Deck

☐ Furnished

☐ Hardwood Floor

☐ Garage Parking

☐ Off-Street Parking

Laundry

☐ None

☐ In Unit

☐ Shared/ In-building

Pets

☐ No pets allowed

☐ Cats OK

☐ Dogs OK

* Photos

Save Listing

Post Listing

When an unregistered user clicks either of these buttons it will cause the register/sign in window to pop up.

When the user clicks the “Post a Listing” button on the Home Page they will be redirected to the following page asking for additional information.

The Unregistered User will be able to provide

- Price
- # Bed
- # Bath
- Lease Duration
- Date Listing is Available
- Description of Listing

The Unregistered User will need to provide their

- Name
- Email

So that they can be contacted for their listings.

Amenities and Rules will be a check box where the Unregistered User can specify more information about their listings, the is optional.

The Unregistered User can upload images of their listings and afterwards can choose to Save Listing to complete later or Post Listing to be reviewed before uploaded to site. When either button is clicked a pop up for registration/sign in will appear.

| | |
|--|---|
| <div><div>Upload Listing</div><div><div>Rent ▼</div><div>Sell ▼</div><div>Register Log In</div></div><div><div>Upload Listing</div><div><div><div>Photos*</div><div>+upload image(s) of listing</div></div><div><div>Address*</div><div><div></div><div></div><div>City*</div><div>Zip Code*</div><div></div><div></div></div></div></div><div><div>Done</div></div></div></div> | <p>After the user clicks Photos from the previous diagram they will be redirected to the following page asking for additional information.</p> <p>The Previously Unregistered User will be able to provide</p> <ul style="list-style-type: none">• Photos• Address• City• Zip Code <p>The Previously Unregistered User can press Done to upload images of their listing.</p> |
|--|---|

Thank you for Posting!

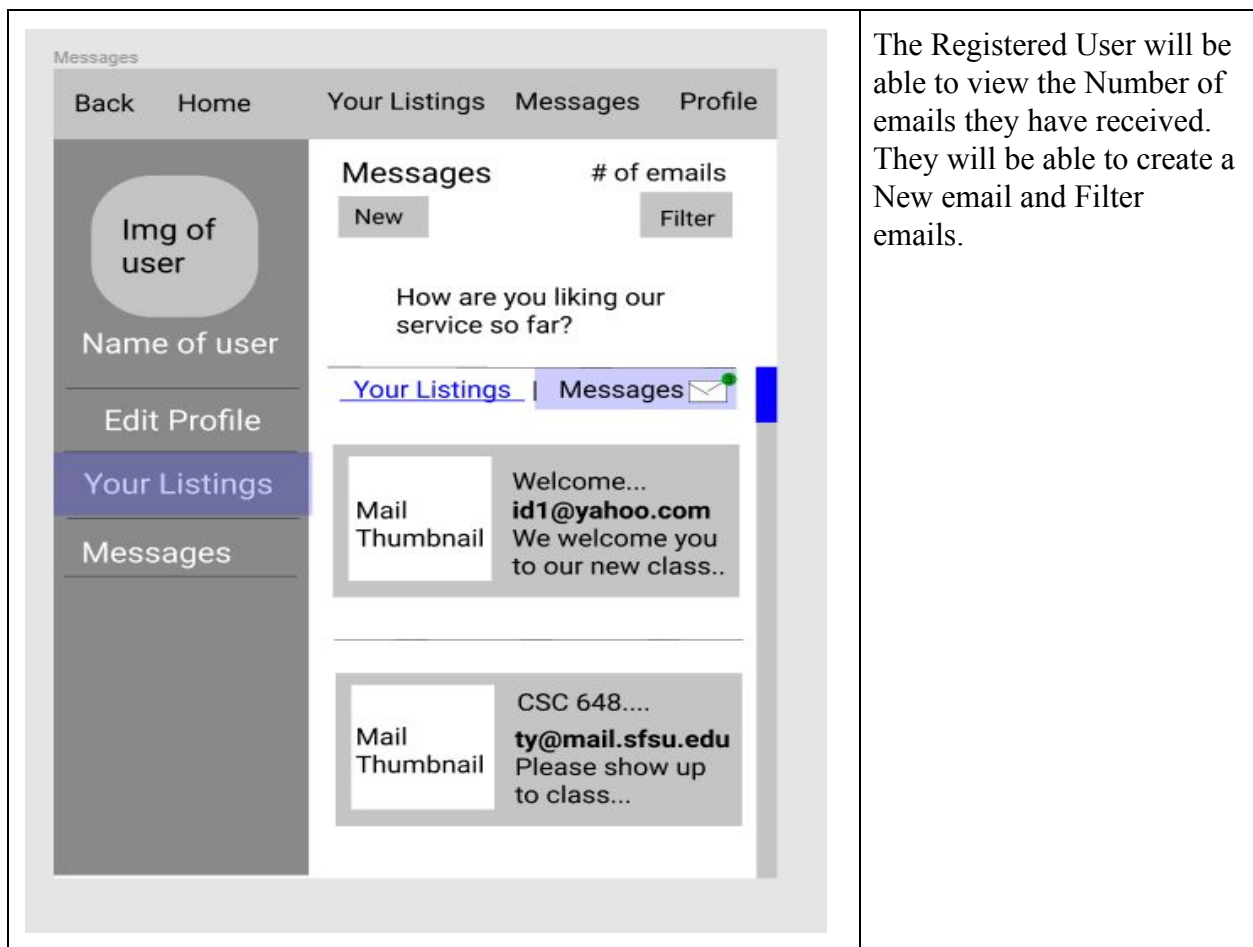
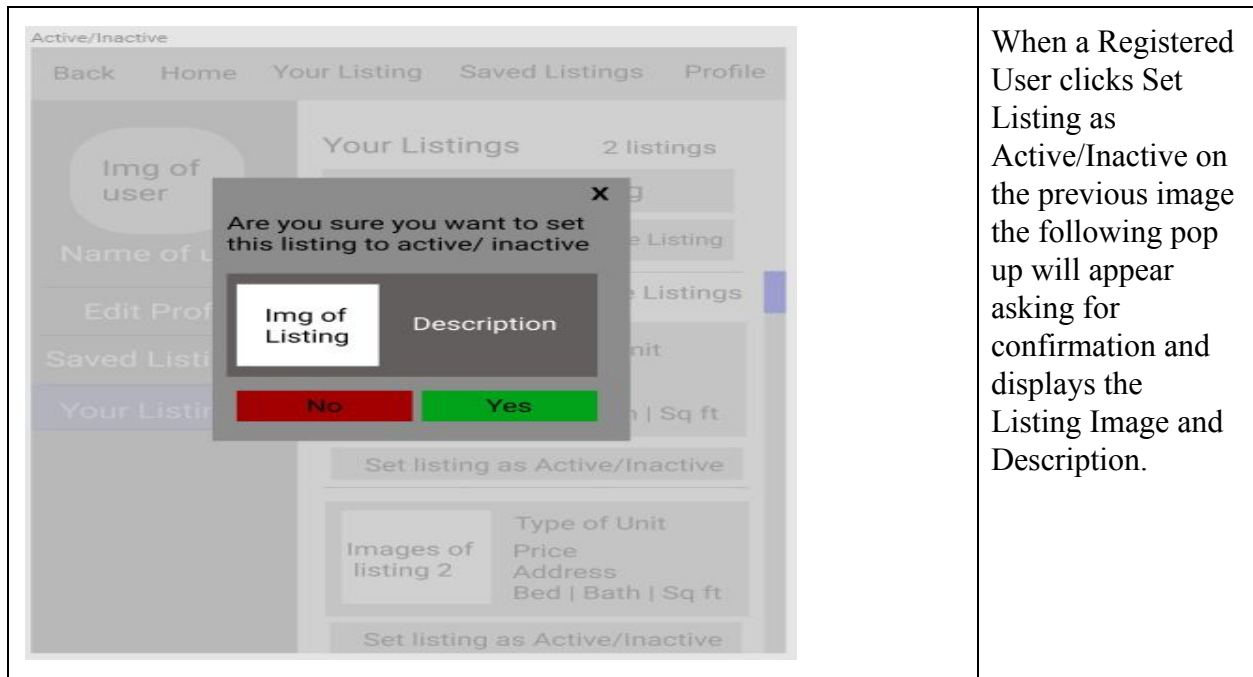
Thank you for posting a listing.

Your upload will be approved
within the next 24h hrs.

After the user clicks Post Listing from the previous diagram they will be redirected to the following page with a message that thanks them and tells them their posting will be approved within 24 hours

Registered Users

| | |
|--|--|
| <div data-bbox="250 291 1015 1304"><p>Your Listings</p><p>Back Home Your Listings Messages Profile</p><div><div>Img of user</div><div>Name of user</div><div>Edit Profile</div><div>Your Listings</div><div>Messages</div></div><div><div>Your Listings<div>New Edit Remove Listing</div></div><div># of listings</div><div>You have new Messages!</div><div><div>Your Listings Messages</div></div><div><div>Images of listing 1</div><div>Type of Unit Price Address Bed Bath Sq ft</div><div>Set listing as Active/Inactive</div></div><div><div>Images of listing 2</div><div>Type of Unit Price Address Bed Bath Sq ft</div><div>Set listing as Active/Inactive</div></div></div></div> <div data-bbox="250 1350 1015 1587"><p>When viewing your listings you'll be able to easily post new listings, edit or delete current listings. The user is able to view all their listings that they have up and scroll through them. Active listing and inactive listing will have the same look to it, the only difference is Active listings will be able to be viewed by the public while inactive listings will only be able to be viewed by the current user.</p></div> | <p>The Registered User will be able to view the Number of Listings they have posted. They will be able to Post, Edit, or Remove Listings from this page. Messages they have received from other User from their Listings.</p> <p>Registered User will be able to change their listings from Active Listing or Inactive Listing and select which Listings they want to view on that page.</p> |
|--|--|



Admin

| | |
|---|--|
| <div><p>Login Page</p><div><h3>GatorState</h3><div><div>Logo</div><h3>Administrator</h3><div><div>Username</div><input type="text"/></div><div><div>Password</div><input type="password"/></div><div><div>Log In</div></div><div>Forgot Password?</div></div></div></div> | <p>Separate Login Page for Admin of GatorState. It prompts the Admin for their Username and Password. If the admin forgot their password and they click the button it will lead to the following page.</p> |
|---|--|

| | |
|--|--|
| <div data-bbox="207 197 1026 856"><p>Forgot Password</p><h2>Reset your password</h2><p>Please enter your email address to request a password reset</p><p>Email address</p><input data-bbox="334 621 854 693" type="text"/><p><input data-bbox="696 716 854 774" type="button" value="Send"/></p></div> | <p>The Admin will need to supply the Email address and if there is a Admin with that email in our database then we will send a password reset email.</p> |
|--|--|

*This is currently not a part of M2 but I would like your feedback on the idea behind the admin page.

*Explanation of the image above:

Side Navbar

Profile

- Home page with listings to review
 - The listings can be filtered by
 - Newest
 - Flagged
 - Updated by User

Review Listings

- View listings that have been flagged for additional review by other admins.

Message

- Hub of messages
- Admins can message user regarding information on their listings that need to be changed before it can be made public on the site.
- Admins can see the messages between them and developers

Contact Developer

- Quick way to message developers
- If the message is regarding a listing it will need to be attached with the message.

Buttons beneath the Listing

Approve

- If an admin finds no fault with a listing he can make it public to the site.

Request info from user

- Quick way for the admin to message the user on information that needs to be changed regarding their listing.

Flagged

- If a listing needs additional review from an admin
- Listings that are flagged will remain Inactive
- If a user has enough Listings that are flagged their account may be frozen.

Inactive / Delete?

- Listing cannot public until further changes by the admins
- The user will receive a notification on why their listings are not being made public.

4. High level Architecture, Database Organization

DB Organization:

These are the each of the database tables along with the attributes contained within them:

❖ User

- UserID
- Name
- Username
- Email
- Password
- Age
- Photo
- Messages
- UserFavoritesListID

❖ Messages

- FromUserID
- ToUserID
- Message
- TimeSent

❖ Listing

- ListingID
- UserID
- StreetAddress
- City
- ZipCode
- State
- NumberOfRooms
- NumberOfBathrooms
- Images
- HomeType - can be apartment, house, or dorm
- SquareFeet
- Price
- Rating
- ThumbnailPhoto
- Active

- ❖ UserFavoritesList
 - UserFavoritesListID
 - ListingID(1)...ListingID(n)

Media Storage:

We have decided to use a file system to store our data, images as well as gps data.

Search/filter architecture and implementation:

We shall organize the searches we display first and foremost around the specification of the user. We shall display listings that are within the range of the filters specified. Additionally we can show recommended listings that are close to the users specifications but are slightly* out of range. (We need to define what slightly means)

Algorithm Overview:

Entered in from user search:

1. Address
 - a. If given, show the building of the exact address that was given
2. Zip code
 - a. If we are only given a zip code, show all listings in that given zip code
3. City
 - a. Show Listings in that city across the city.

Filter priority rating:

- High: Only find listing that are within what the user specified.
- Low: Show listings that are within what the user input, however if a listing has been found within the filtered input except this one, then dismiss this one.

Filters that can be added:

1. Price
 - a. High priority
 - b. If a user specifies price our algorithm will then only show listings within that range.
2. Distance
 - a. High priority
 - b. Stay within the users input.

3. Home type
 - a. Low priority
 - b. Show only the type of buildings the user specifies
 - i. Dorm
 - ii. Apartment
 - iii. House
 - iv. Condo
4. Bed
 - a. Low priority
 - b. Lower priority
5. Bath
 - a. Low priority
6. Square Feet
 - a. Low priority

If the user changes a filtered search after the initial search, prioritize showing all the new listings that fall within the filter.

How information will be retrieved from the database:

Depending on which filters are specified, only the necessary attributes needed to determine which listings are desired to populate in the search results and in which order they should be displayed will be initially queried. The requesting app will then parse through the information provided by the database, and send a second request which will have the desired order, will request more information on the listings to display, and will omit any listings from the initial query that are undesired. Here are couple of examples of how this will generally work:

1. No filters specified:
 - a. Return results based solely of zip code, city, or address entered in search bar.
 - b. Example query:

```
SELECT street_address, city, zip_code, state, images,
home_type, price

FROM listings

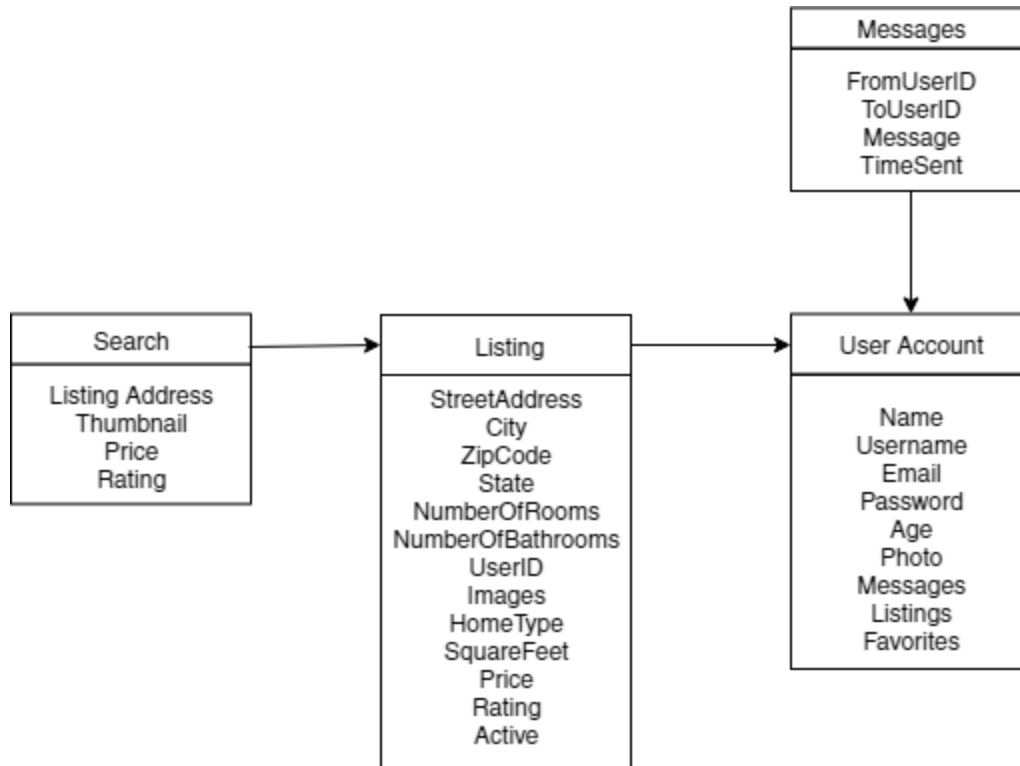
WHERE city LIKE %${search_value}
```
2. Filters specified:
 - a. Return results based off what was entered in the search bar and what was selected as a filter.

b. Example query:

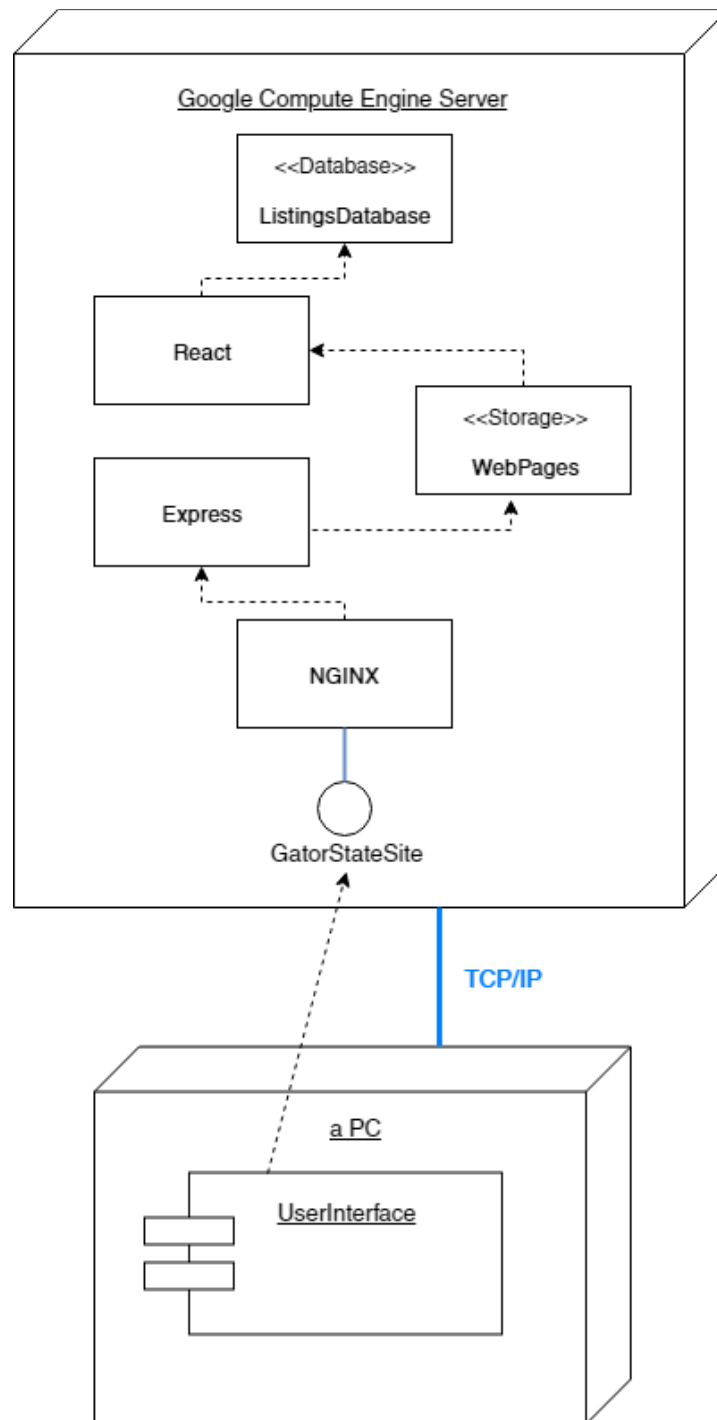
```
SELECT street_address, city, zip_code, state, images,  
home_type, price  
  
FROM listings  
  
WHERE city LIKE %${search_value} AND home_type = ${filter}
```

5. High Level UML Diagrams

High-Level UML:



Component/Deployment Diagram:



6 .Identify *actual* key risks for your project at this time

Skills (Risks)

- Back End
 - Not properly structure the database entities
 - Having clear entity name that relate to their function and using proper naming conventions.
 - Race conditions
 - Mutex, semaphore locks.
 - Setting proper access depending on user priority.
 - Write the right rules for each priority level of the users.
- Front End
 - Having improper states and calling improper functions
 - Setting obvious and clear names for our states and functions.
 - Not properly linking sources to our back end.
 - Actively debug and do code reviews. Create test cases for functions in our front end.
- Scheduled risks
 - Recently pushed code breaks the site
 - Restore to the previous version of the site.
- Technical Risk
 - Pushing bad code to the main branch
 - Push to the development branch and make sure it runs properly before pushing it to the main branch
 - Map API
 - Look into other programs that integrate google map api and use it as a reference.
 - Linking front end and back end while using react.
 - Study react in depth and find how others use react when having the front end communicate with the back end.
 - SQL injections
 - Proper checks on any data that is being inputted by the user.
 - Properly protecting sensitive information that the users may have.
 - Identify what the information is and set proper checks and restrictions on that data.
 - Prevent bots from flooding the sites with postings to crash it.
 - Implement a captcha when posting

- Teamwork Risk
 - People are doing something they aren't comfortable coding
 - Clear communication among teammates and discussing what people need help with or changing the tasks people are doing.
- Legal
 - Make sure users are posting listings that are actually theirs and not just someone else's home.
 - Have a team to verify the listing before listings are posted.
 - Have a report button that users can use.
 - Properly licensing any material that is not ours, i.e. SFSU logo
 - Don't use them and create our own, or contact the owner of the material to see if we can get the rights to use the material.

7. Project management

We managed all of Milestone 2 requirements by organizing all the tasks that were assigned into Trello. From there we meet up as a group to discuss all the requirements and agreed on how to separate the tasks that we felt should be handled by front end and back end. We organized the priorities of the tasks that needed to be completed first in order to make the rest of the tasks easier. From there we assigned tasks to team members that felt the most confident with completing them. After a team member has completed what they were assigned, they marked that it was done on Trello, which allowed prompted other team members to review what was completed.