**SW Engineering CSC 648/848 Spring 2019**



(temp)

**Team 13**

**Amari Bolmer - Team Lead - amarirules@gmail.com**

**Brian Ho, Simon Tan, Adeel Bhatti,**

**Kim Wang, Kurtis Hoang, Sushil Kumar Plassar**

[**https://github.com/CSC-648-SFSU/csc648-sp19-team13/tree/master**](https://github.com/CSC-648-SFSU/csc648-sp19-team13/tree/master)

**Milestone 2**

**March 19, 2019**

History Table:

Date Submitted: March 22, 2019

**Table of Contents**

[**Data Definitions**](#_rtlwr8wnp4hg) **3**

[**Functional Requirements**](#_jr5oqkcp2gg4) **3**

[**UI Mockups and Storyboards (high level only)**](#_9gxe26shxr8t) **3**

[**High level Architecture, Database Organization**](#_aac9bwlijlr7) **3**

[**High Level UML Diagrams**](#_d9zg04e715qz) **3**

[**Identify actual key risks**](#_dcj5bav21rfq) **3**

[**Project management**](#_e3wkytcx4p3s) **3**

# Data Definitions

A. Unregistered Users:

Users who have not signed up or logged in. They can use the website to browse the listings and house details. They cannot send renting requests, post a listing, nor contact landlords.

B. Registered User:

Registered users can be both customers and landlords at the same time, but they shall have different dashboard to view specific activities. They can browse the website and listing details. Also, they can contact landlords, send renting requests, and post their properties for renting.

C. Administrator:

Users who have access to the database and perform administrative tasks including approving postings before they go live and blocking users.

D. Listing

Listed properties on the website. Listing types include:

* Houses
* Apartments
* Condos
* Townhomes
* Single rooms

E. Landlord Dashboard

A page a user can see related information as a landlord. Information includes posted listings, orders, and messages to current or prospective customers. All users shall have both landlord dashboard and customer dashboard.

F. Customer Dashboard

A page a user can see related information as a customer. Information includes orders and messages to landlords. All users shall have both landlord dashboard and customer dashboard.

G. Order

Orders are defined as completed renting transactions. Since a user can both rent a property from our website and post a property on our web for others to rent, there can be two types of orders for a user - orders from a customer perspective, and orders from a landlord perspective. Therefore, we separate these two types of orders and showed them on both customer dashboard and landlord dashboard; on customer dashboard, a user can only see their orders from a customer perspective, while on landlord dashboard, a user can only see their order from a landlord perspective.

H. Message

Similar as “Order” data, message can also have two types: message from customer to landlord, and message from landlord back to customer. We display message the same way as we do for orders; on customer dashboard, a user can only see their message conversations with other landlord users, while on landlord dashboard, a user can only see their message conversations with other customer users.

# 

# Functional Requirements

**Priority 1:**

A. Unregistered User

1. Shall be able to register an account
2. Shall be able to browse the available houses
3. Shall be able to view listing details
4. Shall be able to filter listings by price, size and distance range
5. Shall be able to sort listings by price, size and distance

B. Registered User

Shall be able to do what unregistered users can do plus:

1. Shall be able to login
2. Shall be able to contact landlords
3. Shall be able to make a request to rent properties
4. Shall be able to send email to the landlord
5. Shall be able to send email to the person looking to rent
6. Shall be able to post listing
7. Shall be able to make add, delete, and edit renting properties

C. Administrators:

Shall be able to do what regular registered users can do plus:

1. Shall be able to access the database
2. Shall be able to delete listings
3. Shall be able to approve listings for posting

**Priority 2:**

A. Unregistered User

1. Shall be able to sign up using their google and facebook account

B. Registered User

Shall be able to do what Unregistered users can do plus:

1. Shall be able to upload the profile picture
2. Shall be able to Edit the profile or change the notification preferences

**Priority 3:**

A. Unregistered User

1. Shall be able to see a promoted items on the front page

B. Registered User

Shall be able to do what Unregistered users can do plus

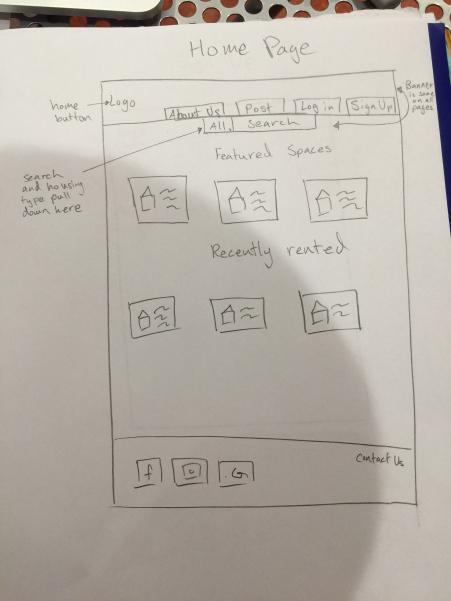
1. Shall be able to send message to other registered users
2. Shall be able to message landlord through inbuilt web chat app
3. Shall be able to contact person looking to rent through inbuilt web chat
4. Shall be able to use reset/forgot password functionality
5. Shall be able to refer a friend
6. Shall be able to see travel times to school from location

# 

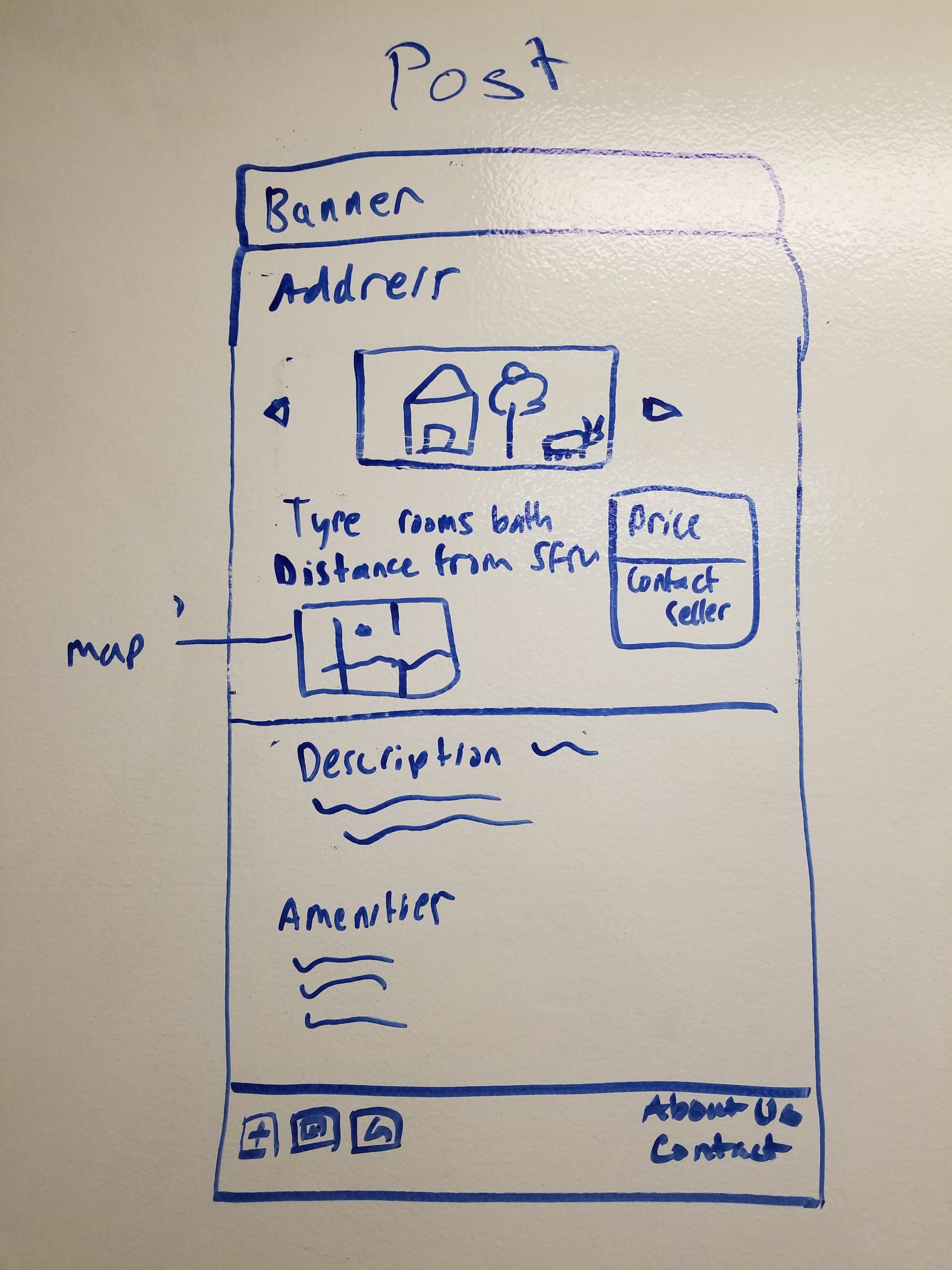
# UI Mockups and Storyboards (high level only)

# Home page

When any user enters our site, they see the home page. There is a banner at the top that stays the same throughout all pages. Search is also in the same place in each page. When a user clicks a house icon, they will be taken to a post.

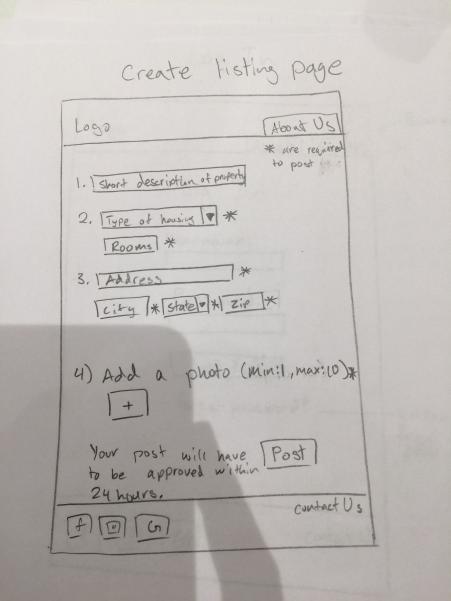


# Listing page



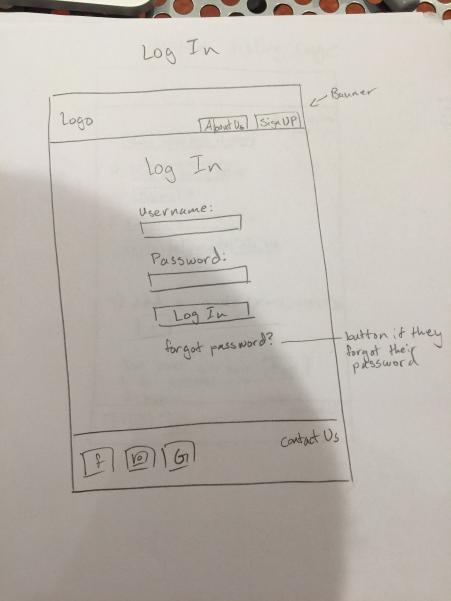
The post shows relevant information at the top. Then the user will see any other information that the landlords want them to see.

# Create listing page



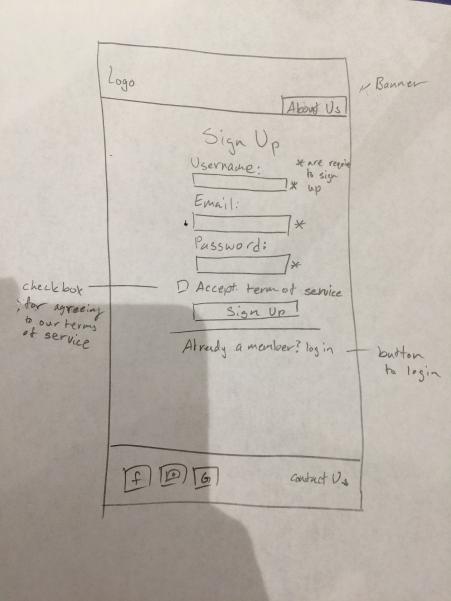
When a registered user presses the post button, they get taken to the create listing area. This form will have uniform fields and fonts to make the user experience better.

# Login page



The log in page will be pretty simple. There are just 2 fields for username and password. If a user is forgetful like most normal user, they can choose to reset their password.

# Signup page



If a user does not have an account and tries to post something, they will be taken to this sign up page. This is because the user has already invested time into creating a post and they would not want to waste their efforts. There will also be a button for registered users to log in.

# Send message page

# 

# This is a prototype for messaging screen. We will try to make it look like an instant messenger. If we can’t figure it out we will make it look like an email client.

Admin Storyboard

The admin storyboard will consist of:

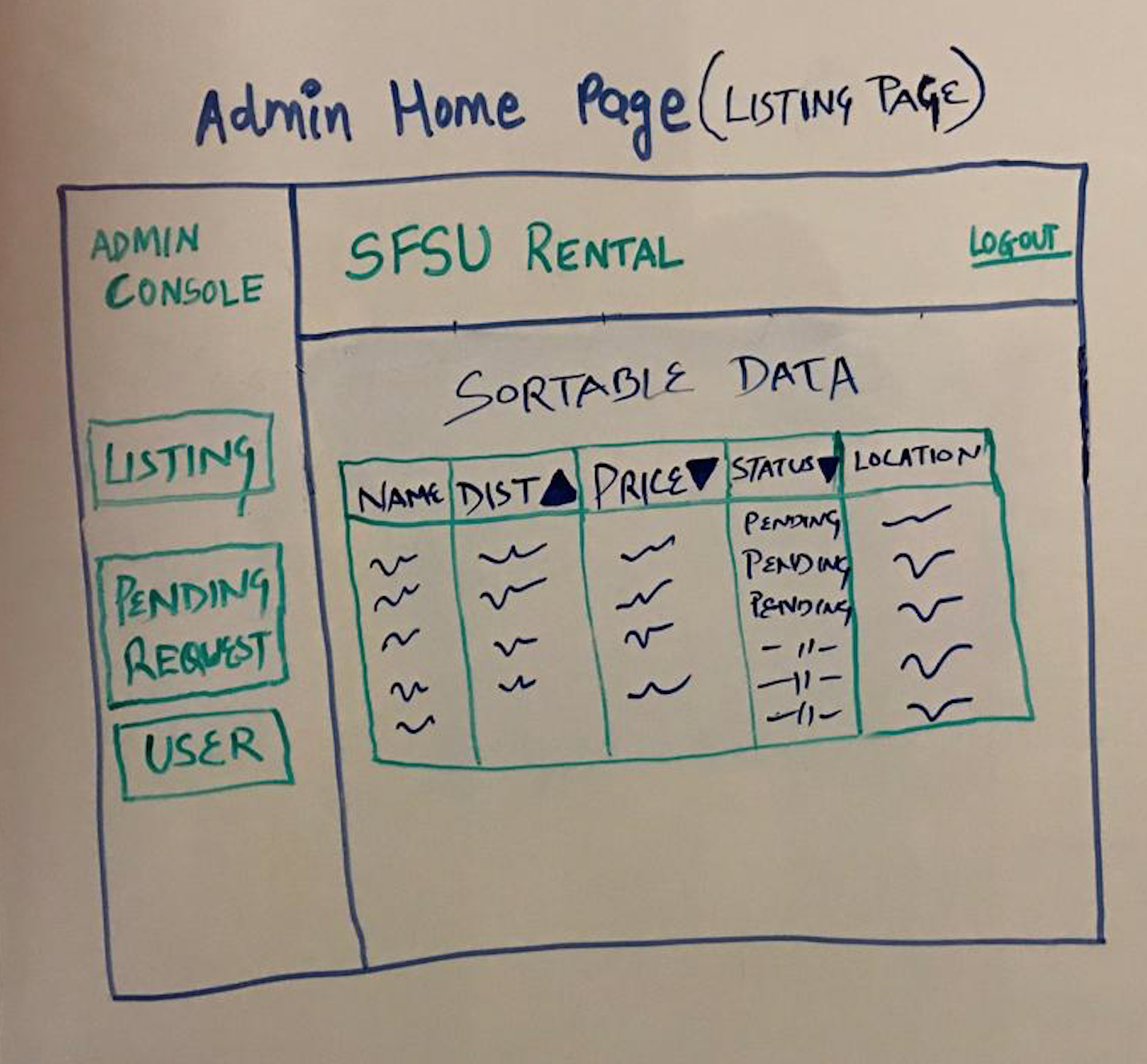
i) Admin Home Page

ii) Admin Approval Page

iii) View History Page

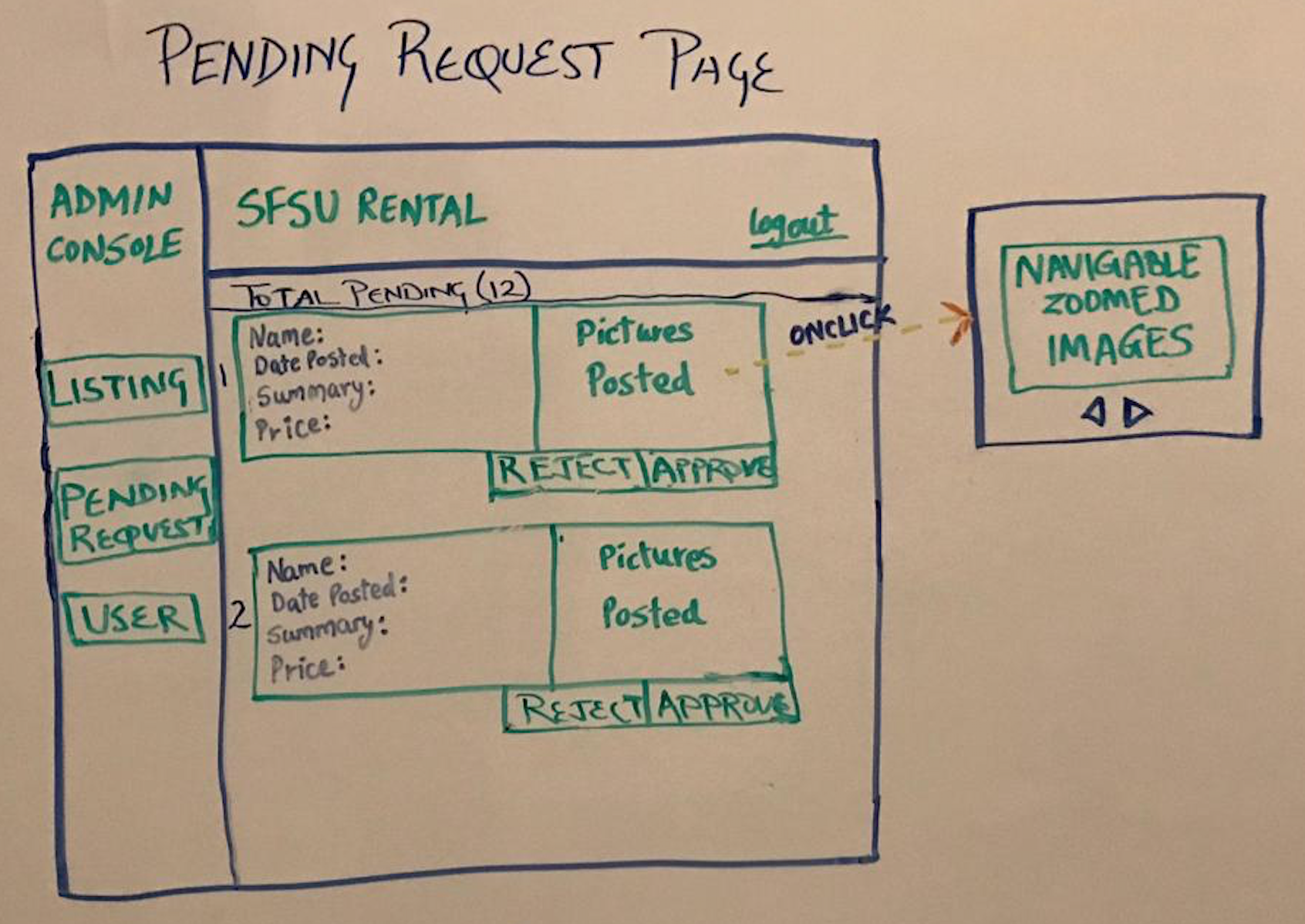
## Admin Home Page

The below screenshot shows the landing page for the admin. It will have the links for him/her to view all the listings, approve/reject the new postings under ‘Pending Requests’ and see the User details under User



## Pending Request page

The admin can view the new posts for approval or rejection under this page. The below screenshot shows the mock page layout:



## 

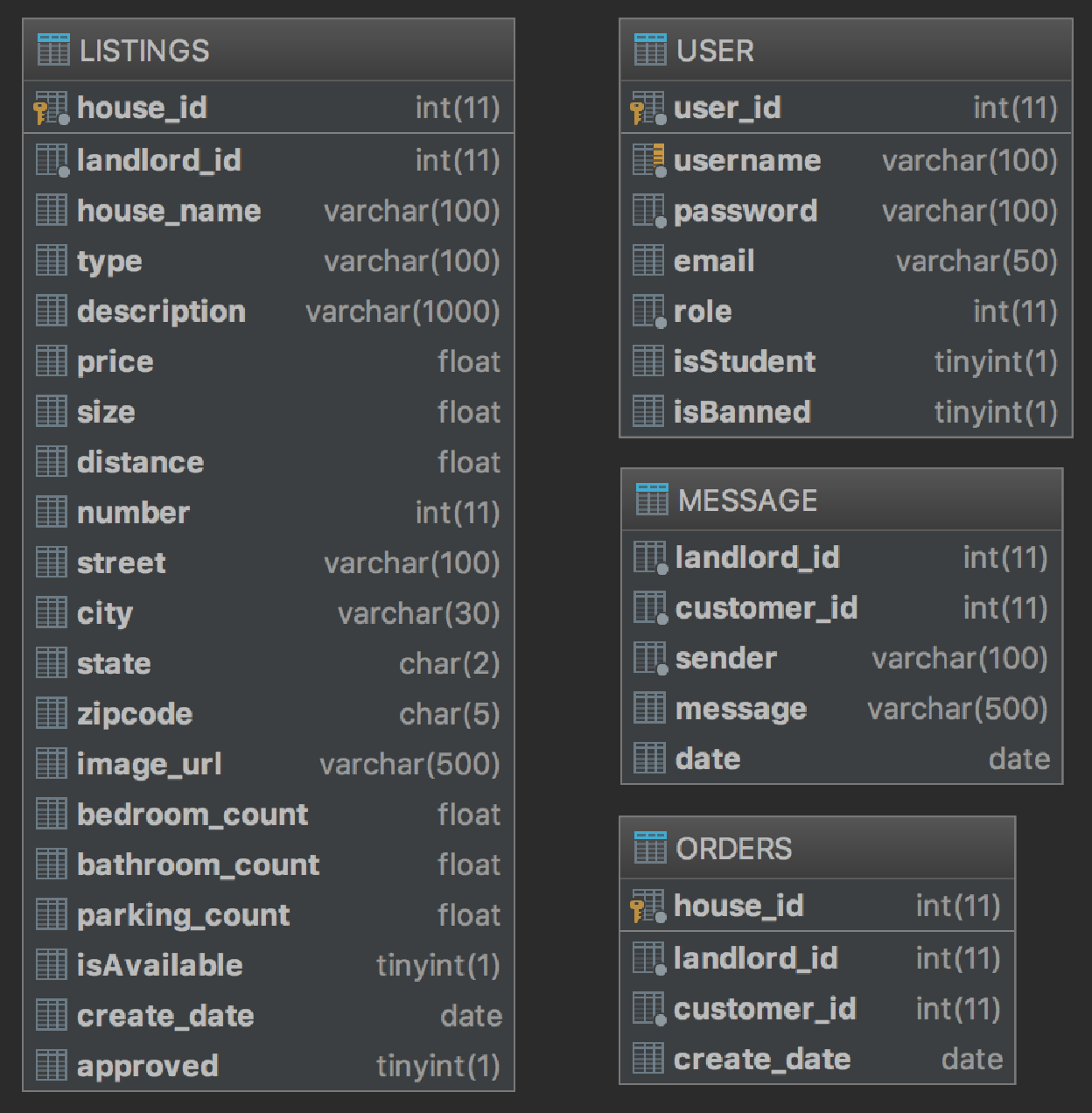
## User Detail Page

The admin can view user details along with the posted ads from the user on this page. Below screenshot shows the mock dashboard for the same



# High level Architecture, Database Organization

# **Database Organization**



# 

# **Media storage**

Media Storage: BLOBs (Binary Large Objects)

Images: Images shall be in the format .JPG or .PNG

Audio: All audio files shall be .MP3

Video: All videos shall be .MP4

GPS: GPS and maps shall be implemented using Google Maps

# **Search/filter architecture and implementation**

Filters:

* **Distance**: Users shall be able to search for listings based on a distance to a certain location using google maps geocode/geolocation to find the distance. The location of the listing will be computed once and stored in the listing database.
* **Major**: Users shall be able to search for listings based on housemates with similar majors.
* **Listing types**: Users shall be able to search by types of listings (i.e apartment, house, single bed rooms, shared bedroom)

Search:

* **%Like:** MySQL built-in search architecture,%like, shall be used to search the database based on the filters.

# 

# High Level UML Diagrams

# Class diagram

# 

# 

# Deployment diagram

# 

# Identify actual key risks

# **Skill Risk & Scheduling Risk**

- Skill risk associated within our team project relies on the inability at current of members being unskilled in various languages. As we begin, the complexity of the project depends upon members ability to quickly learn the skills necessary for their role. This is negated by giving certain coding practices to study to various members.

- Only one member of our team is fluent in python, our backend lead, yet our backend lead is not very well skilled in AWS. Tutorials will be used to help with understanding.

- Scheduling Risk, and may pose an issue when troubleshooting the server if the backend lead is unavailable to help and we are debugging an issue within a language we are not familiar with.

- Skill Resolution, this issue will be handled by having multiple members working on the backend, as the backend lead is gifted in python, this will allow other members to focus on understanding Amazon Web Server.

- Scheduling Risk, the time slot we originally planned is unable to be met by all members.

- Scheduling conflicts will be resolved by members in front-end and backend teams making appropriate separate times to meet in seperate groups to allow deliberation on their respective task if they cannot make the appropriate group discussions.

# 

# Project management

# **Management Details**

Current Tools:

* Trello (Task Assigning and Progress Checker)
* Discord App ( Voice & Messaging)
* Discord App (Screen Sharing Feature)

Future tools will remain from the current tools being implemented. Project deadlines are important so pushing the team to complete assignments early allows for revision time. If any comments or changes would like to be addressed, there is time. Trello is easy for assigning task and monitoring but will stress the importance of checking Trello at least twice a week. Moving forward, with discord we have the ability to @everybody which sends a push notification to all members of the team and can remind members to complete task. Members may forget deadlines, but using direct communication for reminders, team members are quick to remain on task. Tutorials and videos are being used to ensure members understand their assignments.