

# SW Engineering CSC648/848 Summer 2020

## Chillow

## Team 6

### Team Members:

Valdemar Puente-Gonzalez (Team Lead)

[vpuentegonzalez@mail.sfsu.edu](mailto:vpuentegonzalez@mail.sfsu.edu)

Daniel Belmeur (Front-End Lead)

Sean Ellis

Philip Smith (Back-End Lead)

Salah Ali

Karan Gurung (Git Master)

Milestone 2

8 July 2020

Revised:
10 July 2020

## Table of Contents

<b>Functional Requirements .....</b>	<b>3</b>
<b>List of Main Data Items and Entities.....</b>	<b>4</b>
<b>UI Mockups and Storyboards .....</b>	<b>5</b>
<b>HIGH Level Architecture, Database, Organization .....</b>	<b>17</b>
<b>Project Key Risks .....</b>	<b>18</b>
<b>Project Management .....</b>	<b>20</b>

---

## ***Functional Requirements***

---

### **1. Priority 1 (Must Have)**

- 1.1 Unregistered users shall be able to browse homes on our marketplace
- 1.2 Unregistered users shall be able to search and browse by categories
- 1.3 Unregistered users shall be able filter by:
- 1.4 Unregistered users shall be able to filter by Price
- 1.5 Unregistered users shall be able to filter by Address
- 1.6 Unregistered users shall be able to filter by Zip Code
- 1.7 Unregistered users shall be able to filter by # of Bed/Bath
- 1.8 Unregistered users shall be able to filter by Square footage
- 1.9 Unregistered users shall be able to filter by Type of View
- 1.10 Unregistered users shall be able to filter by Parking
- 1.11 Unregistered users shall be able to see rental listing information
- 1.12 Unregistered users shall be able to view reviews
- 1.13 Unregistered users shall be able to see ratings
- 1.14 Registered users have all functionality of a non-registered
- 1.15 Registered users shall be able to login and register to our site user
- 1.16 Registered users shall be able to sell, buy or rent property
- 1.17 Registered users shall be able to send messages to other users
- 1.18 Registered users shall be able to post listings
- 1.19 Registered users shall be able to reset their username or password
- 1.20 Registered users shall have all personal sensitive data saved on our servers encrypted
- 1.21 Admin users shall have all functionality of a registered user
- 1.22 Admin shall be required to approve all posts
- 1.23 Admin shall ban suspicious users
- 1.24 Admin shall block suspicious posts

### **2. Priority 2 (Desired)**

- 2.1 Registered users shall have a watchlist of favorite properties
- 2.2 Registered users shall have the ability to leave a rating after doing business
- 2.3 Registered users shall be able to write reviews
- 2.4 Registered users shall have a transaction history on profile
- 2.5 Unregistered users shall be able to share postings with friends
- 2.6 Unregistered users shall be able to compare homes qualities
- 2.7 Unregistered users shall be able to filter by animal
- 2.8 Unregistered users shall be able to filter by spoken language
- 2.9 Unregistered users shall be able to determine transportation availability

### **3. Priority 3 (Opportunistic)**

- 3.1 Unregistered users shall be able to utilize an in-home viewer
  - Unregistered users shall be able to expand the animal filter specifying animal type and size
- 3.3 Unregistered users shall be able to determine local activities around property

---

## *List of Main Data Items and Entities*

---

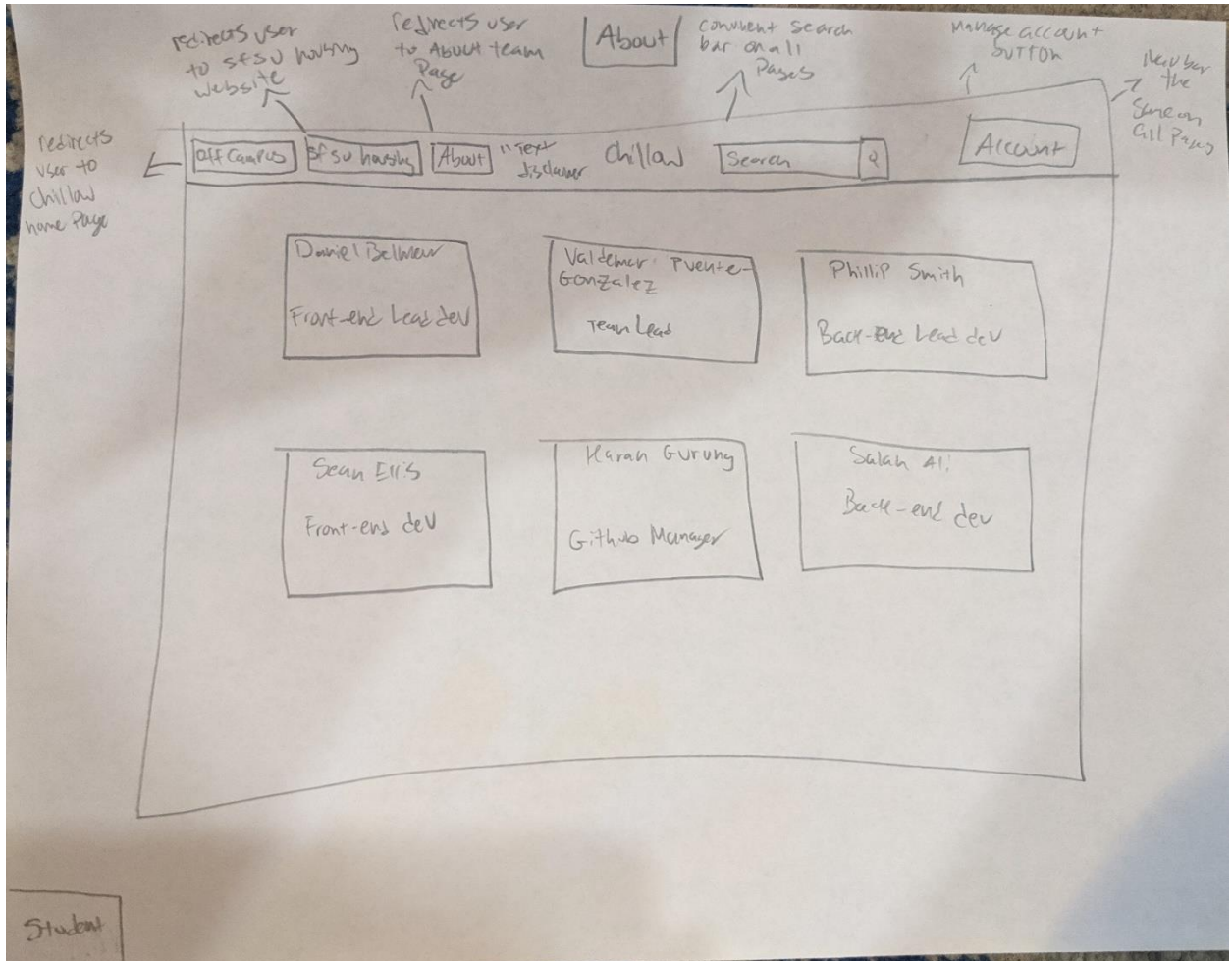
1. Registration record
  - Permissions ID [1, 2, or 3] (permissions of registration classified as follows):  
  
Administrator (1)
    - Attributes: First Name, Last Name, Email, PasswordSFSU User (2)
    - Attributes: First Name, Last Name, Email (must contain '@sfsu.edu'), PasswordRegistered User (3)
    - Attributes: First Name, Last Name, Email, Password
2. Rental listing
  - Attributes: Address, Description, Pictures, Price, Unique ID of listing user, CATEGORY:(apt, room, house), TYPE: (sell/rent) Approved by Admin (Y/N)
3. Messages
  - User ID of sender (must be student or faculty), User ID of recipient (listing user), Message Content, Date/Time Stamp, ID of listings
4. Login Queue
  - List of unique user IDs of users logged in (capped at 50)

---

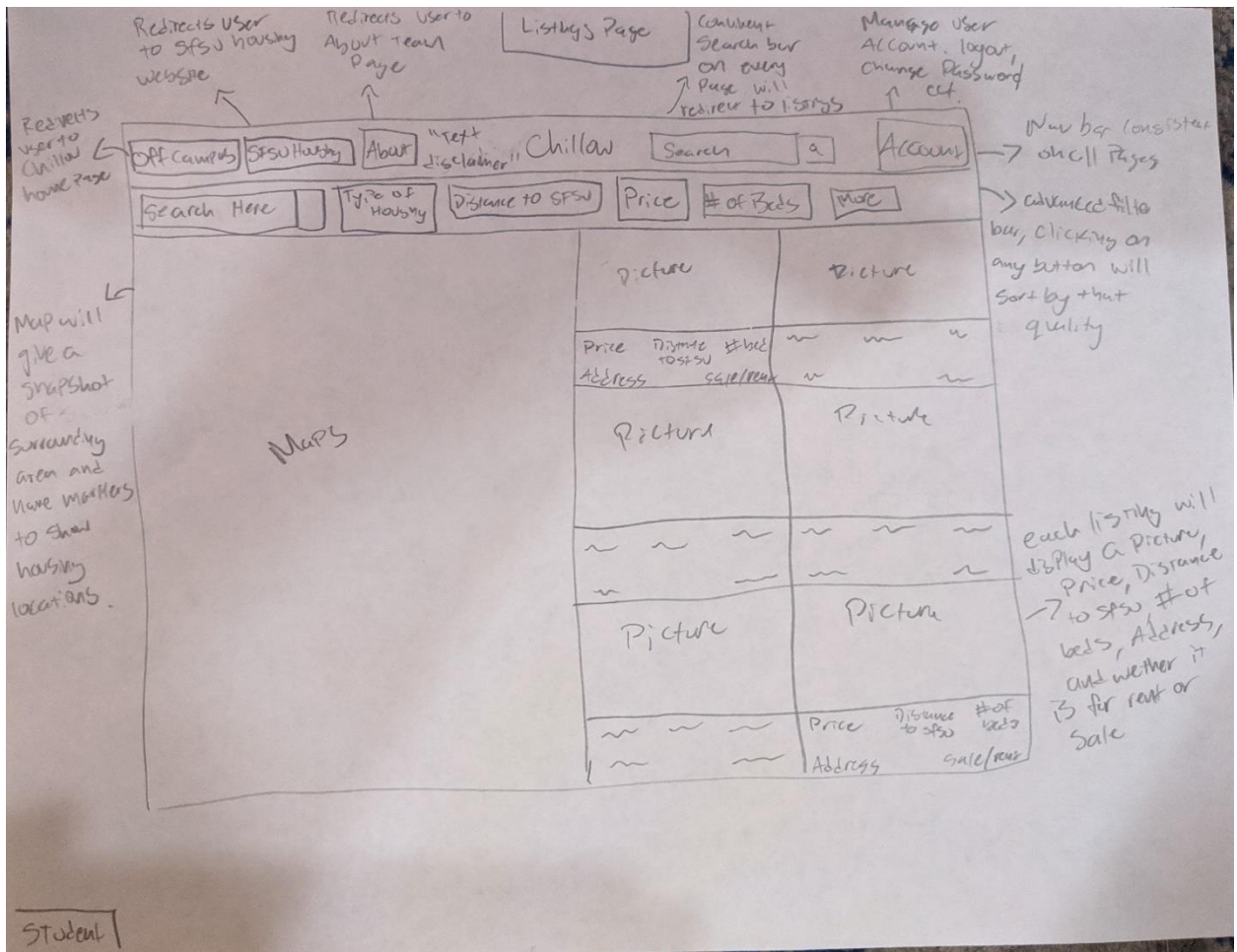
## UI Mockups and Storyboards

---

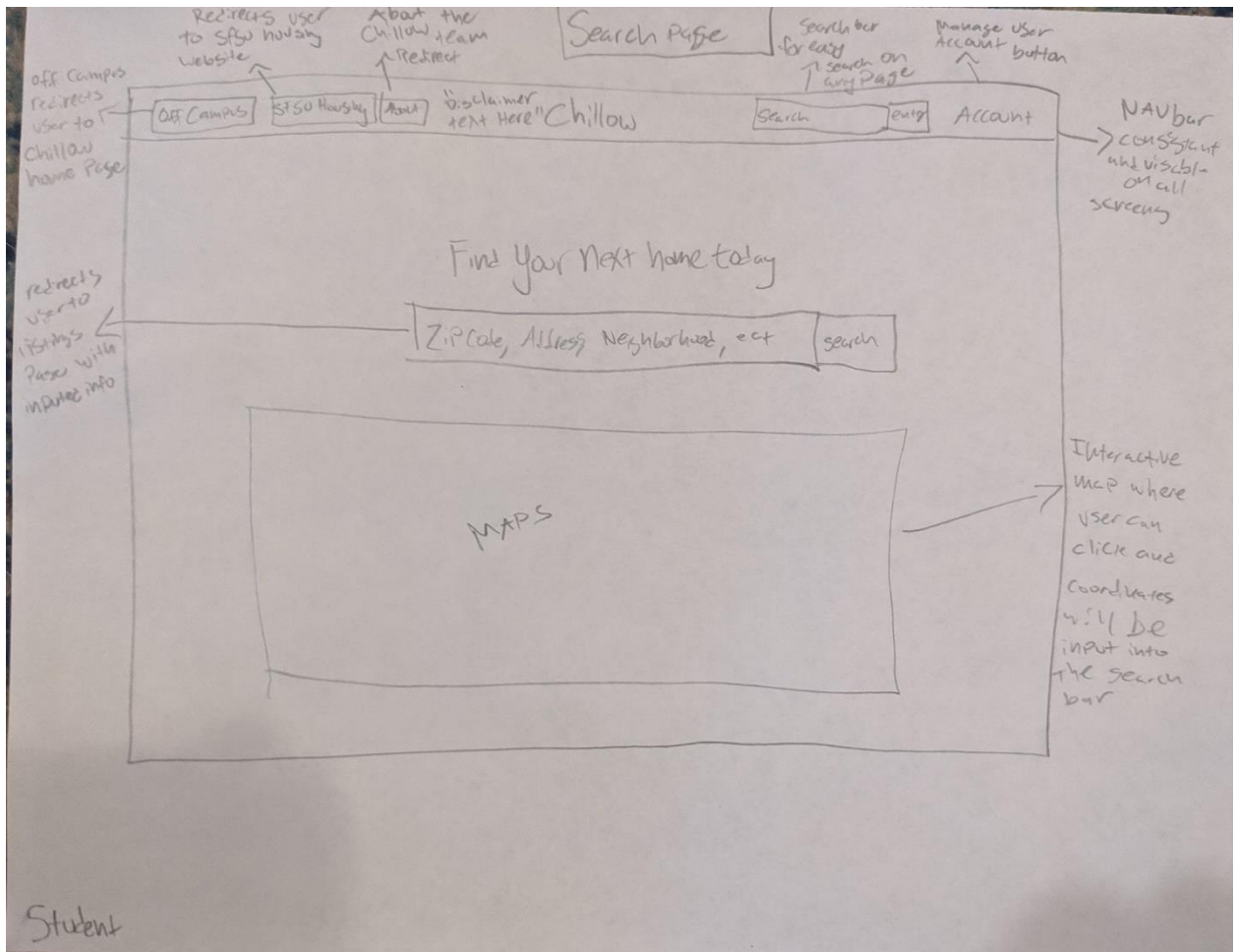
### 1. Student



This is the about page as it looks from the student view



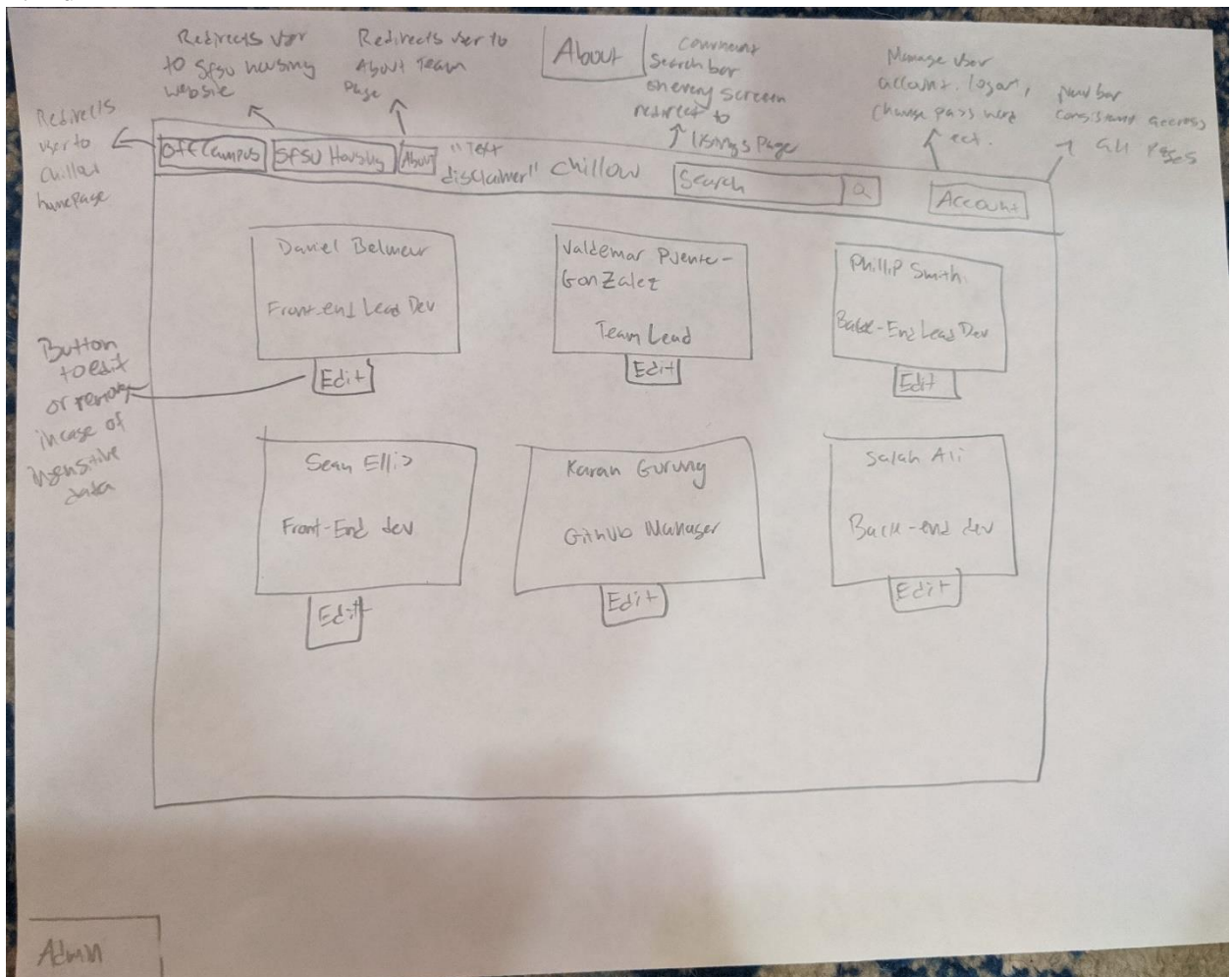
This is the listings page in student view



This is the home page in student view

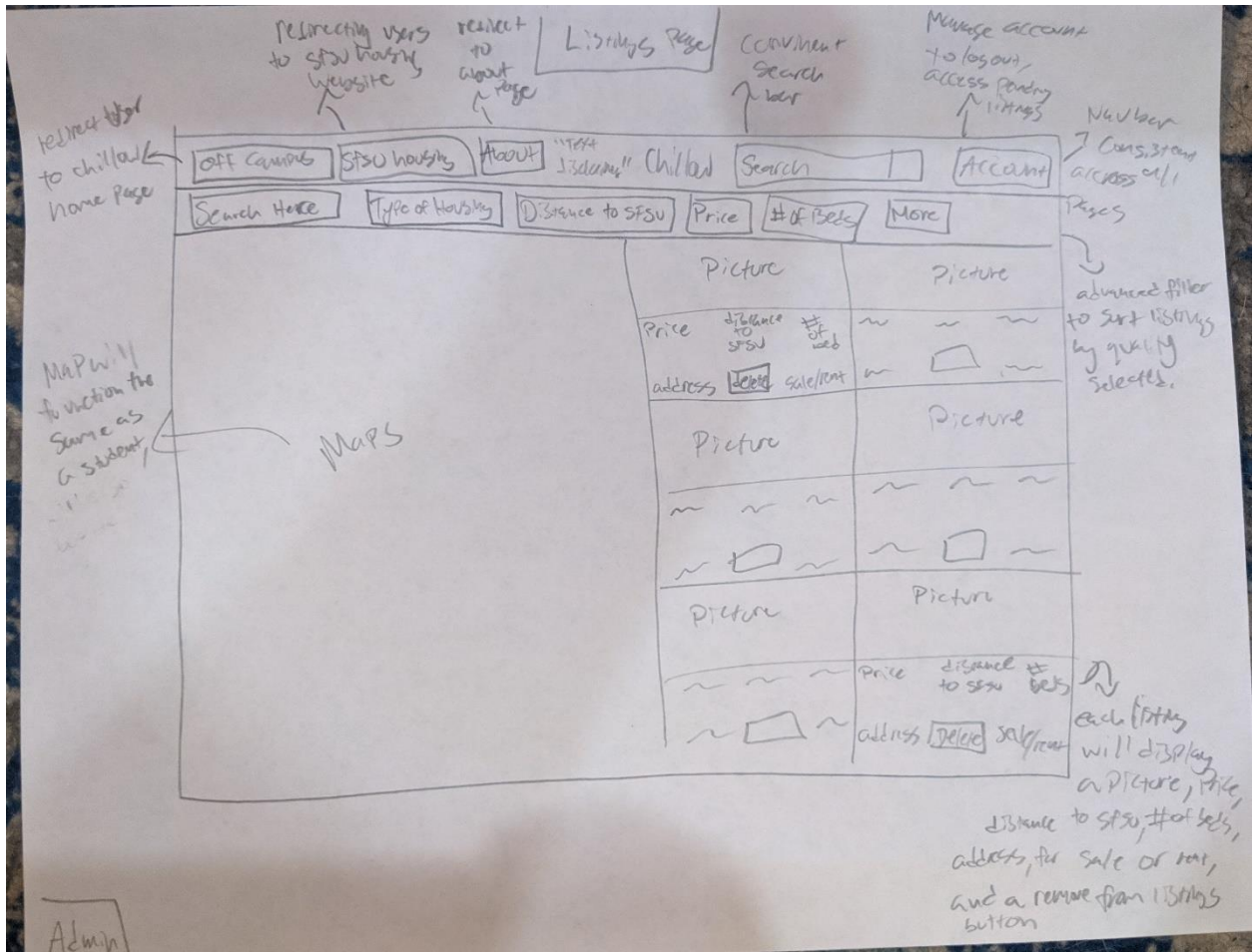


## 2. Admin

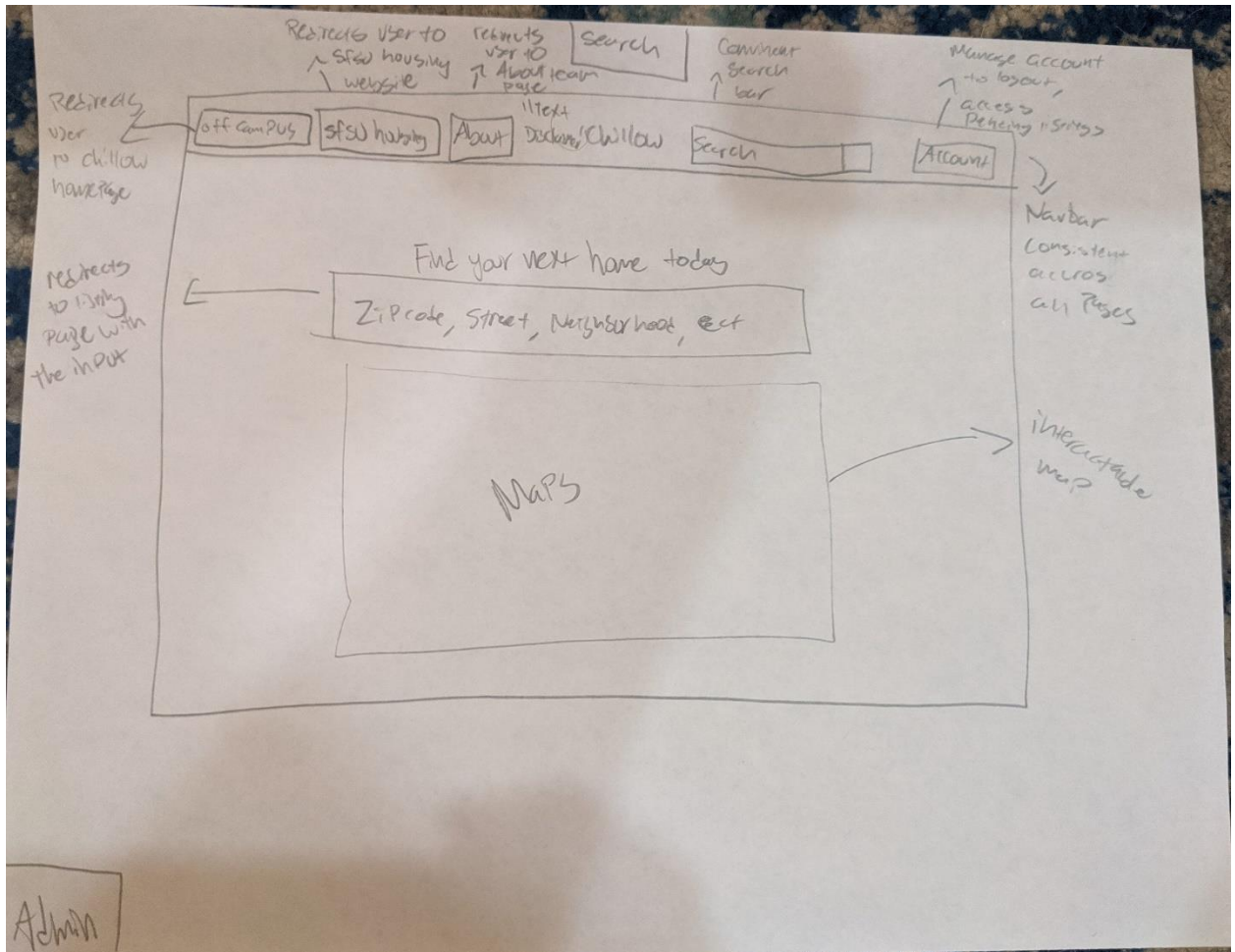


This is the about page in Admin view



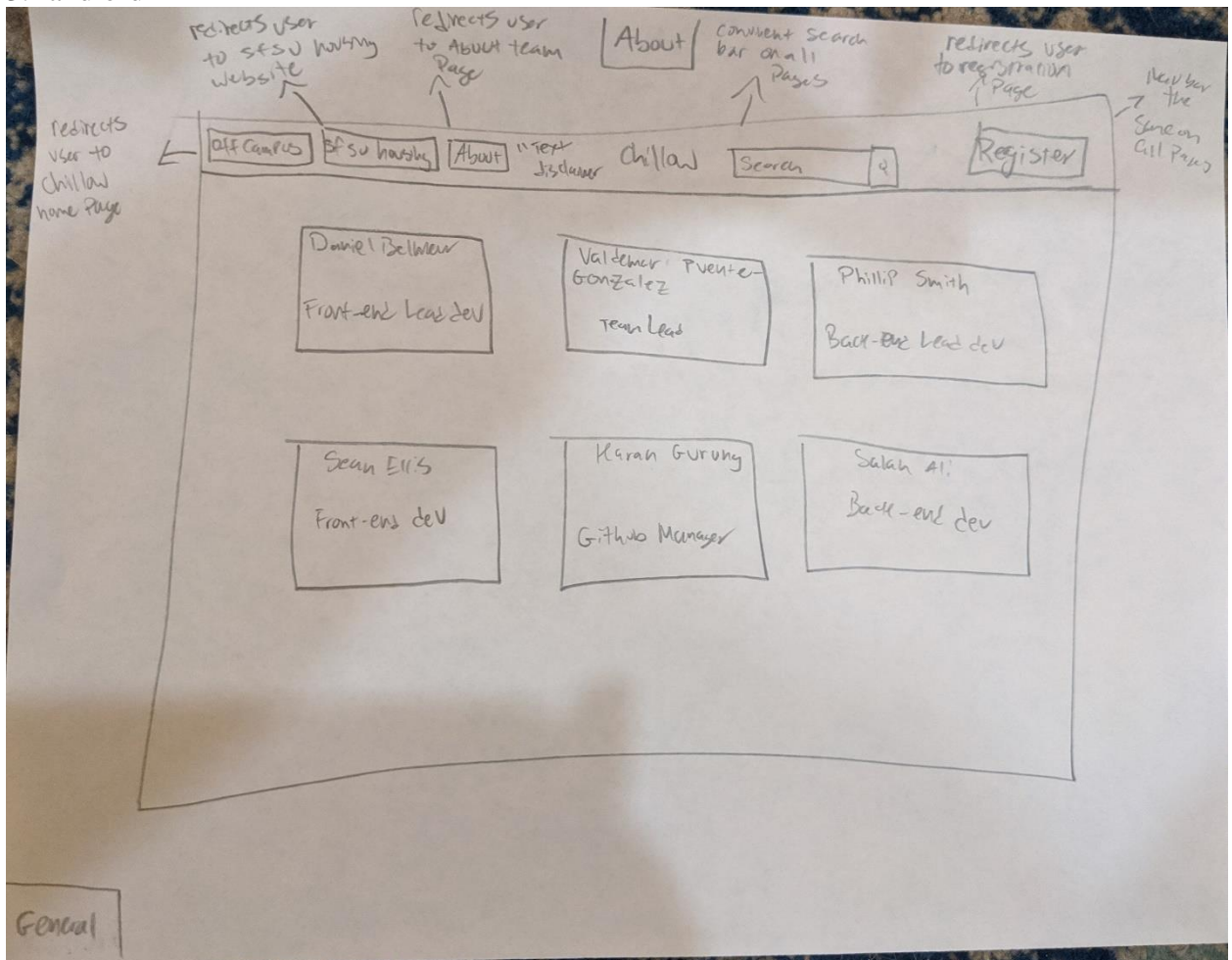


This is the listings page in Admin view

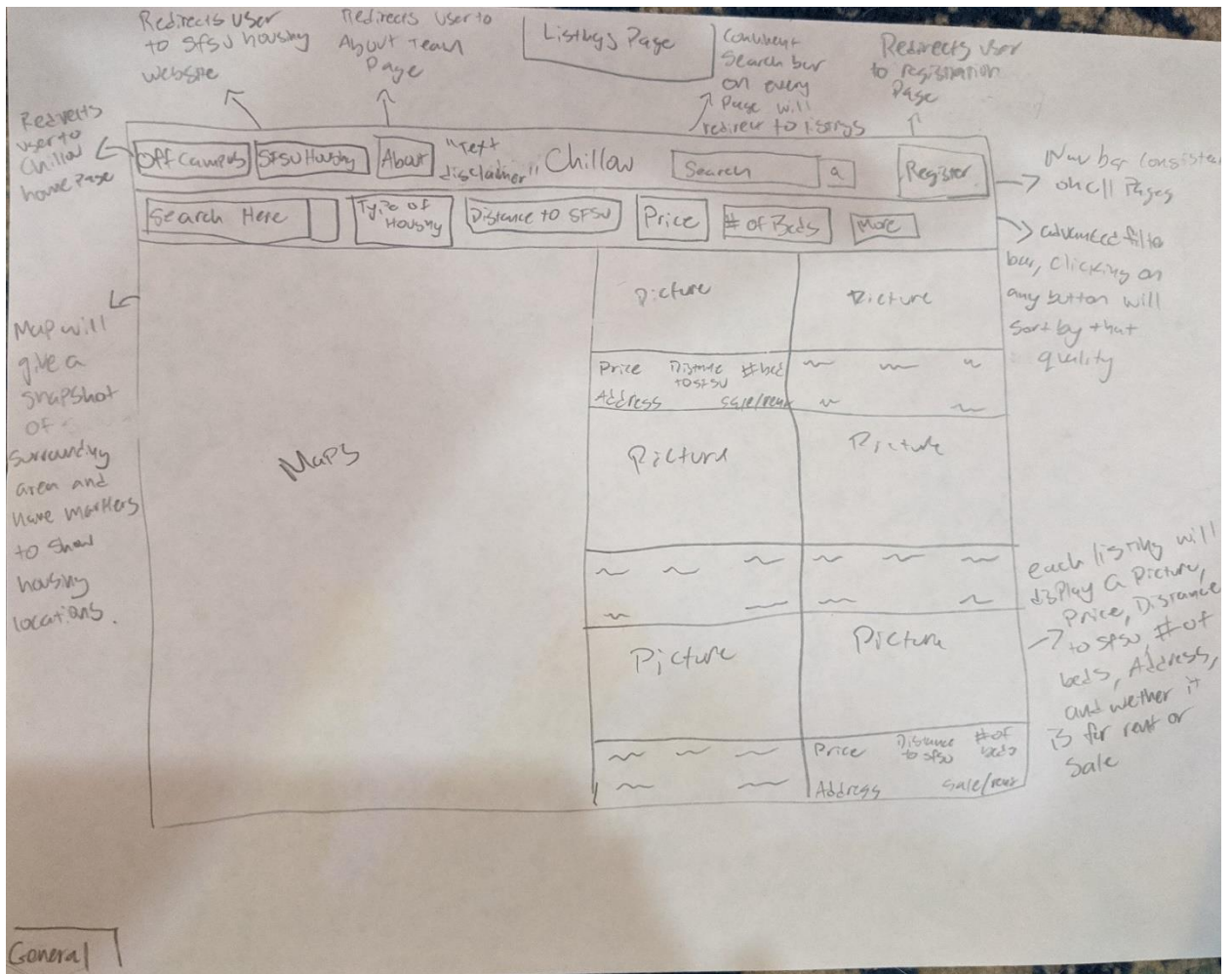


This is the home page in Admin view.

### 3.Landlord

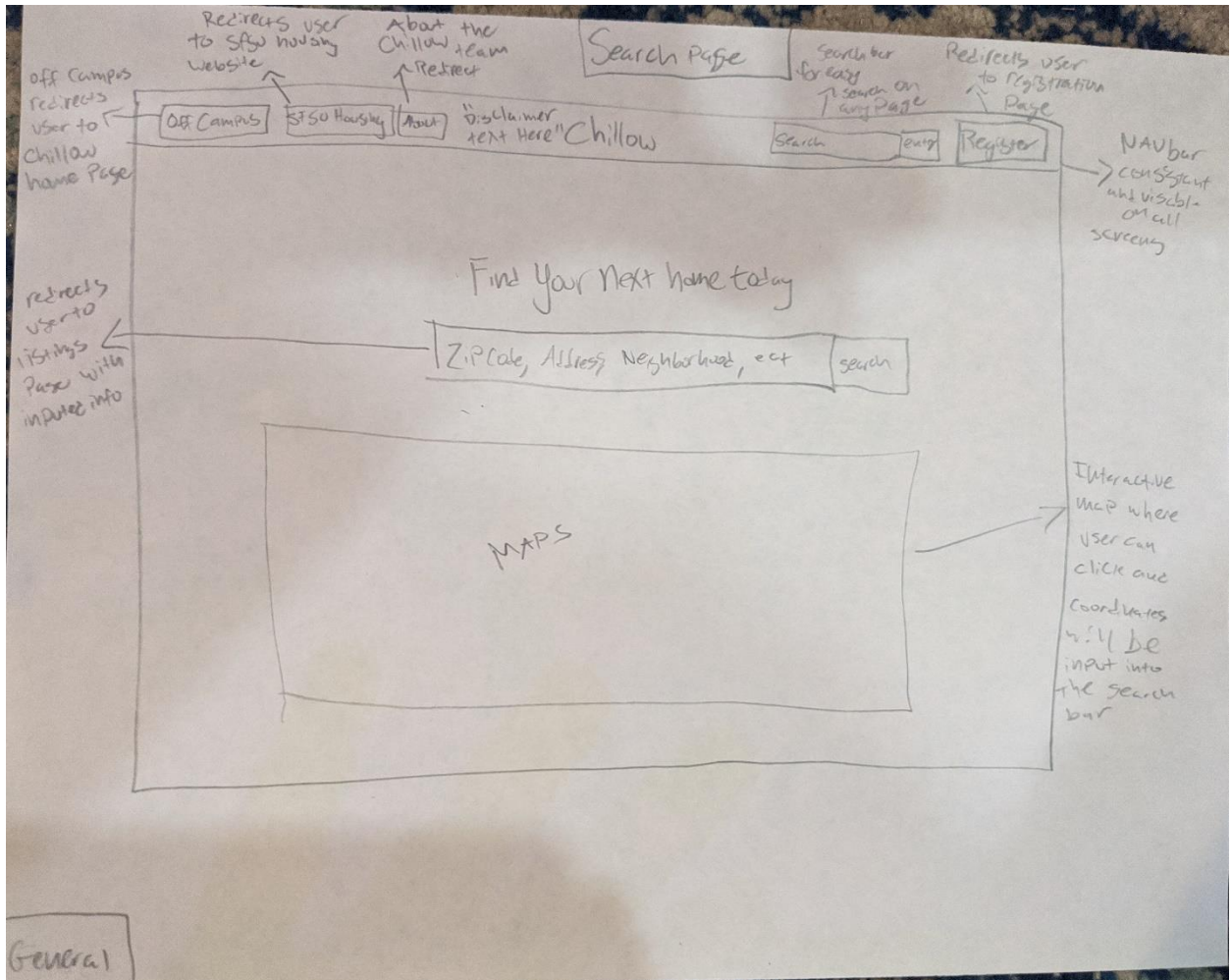


This is the about page as seen by Landlord



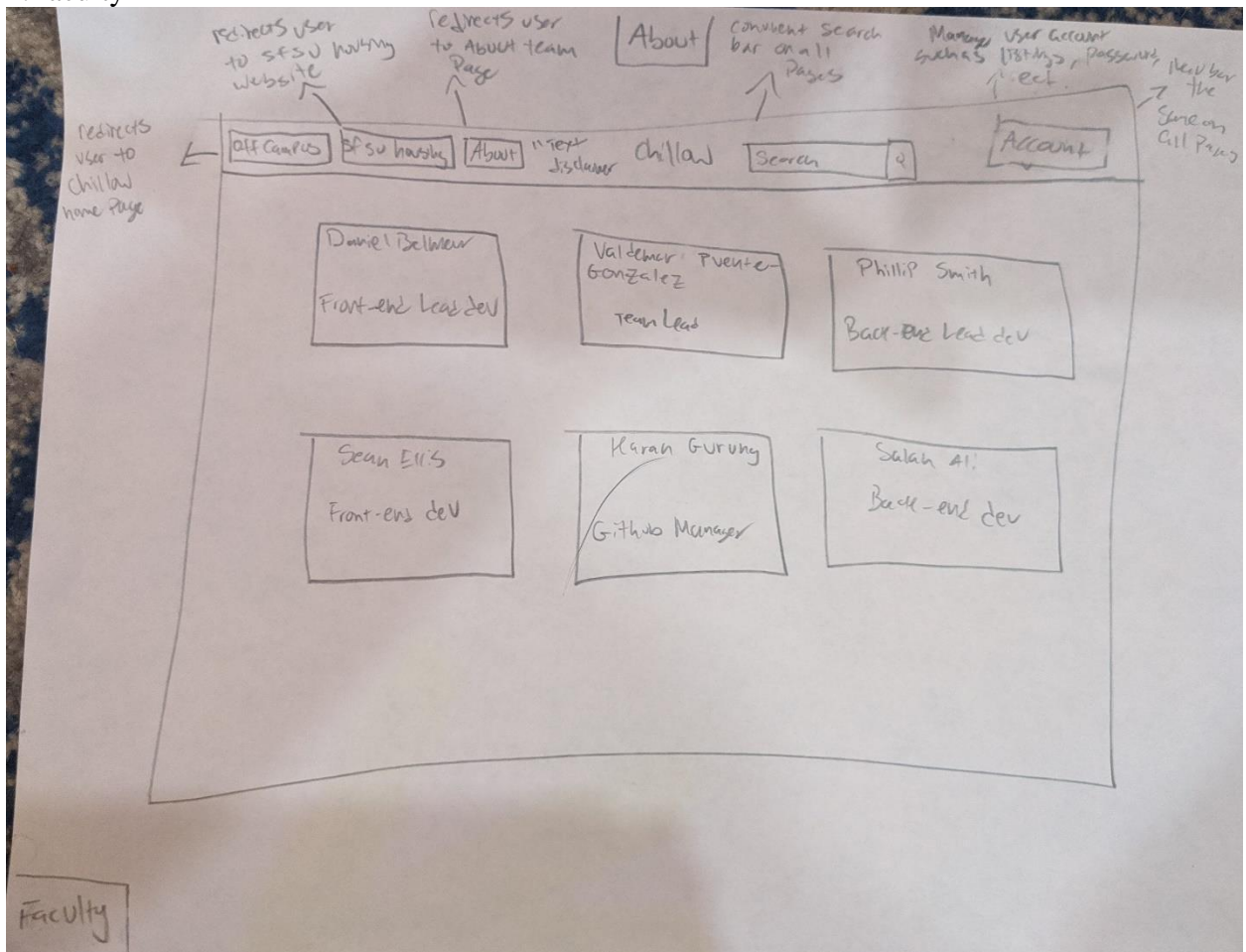
This is the listings page as seen by Landlord



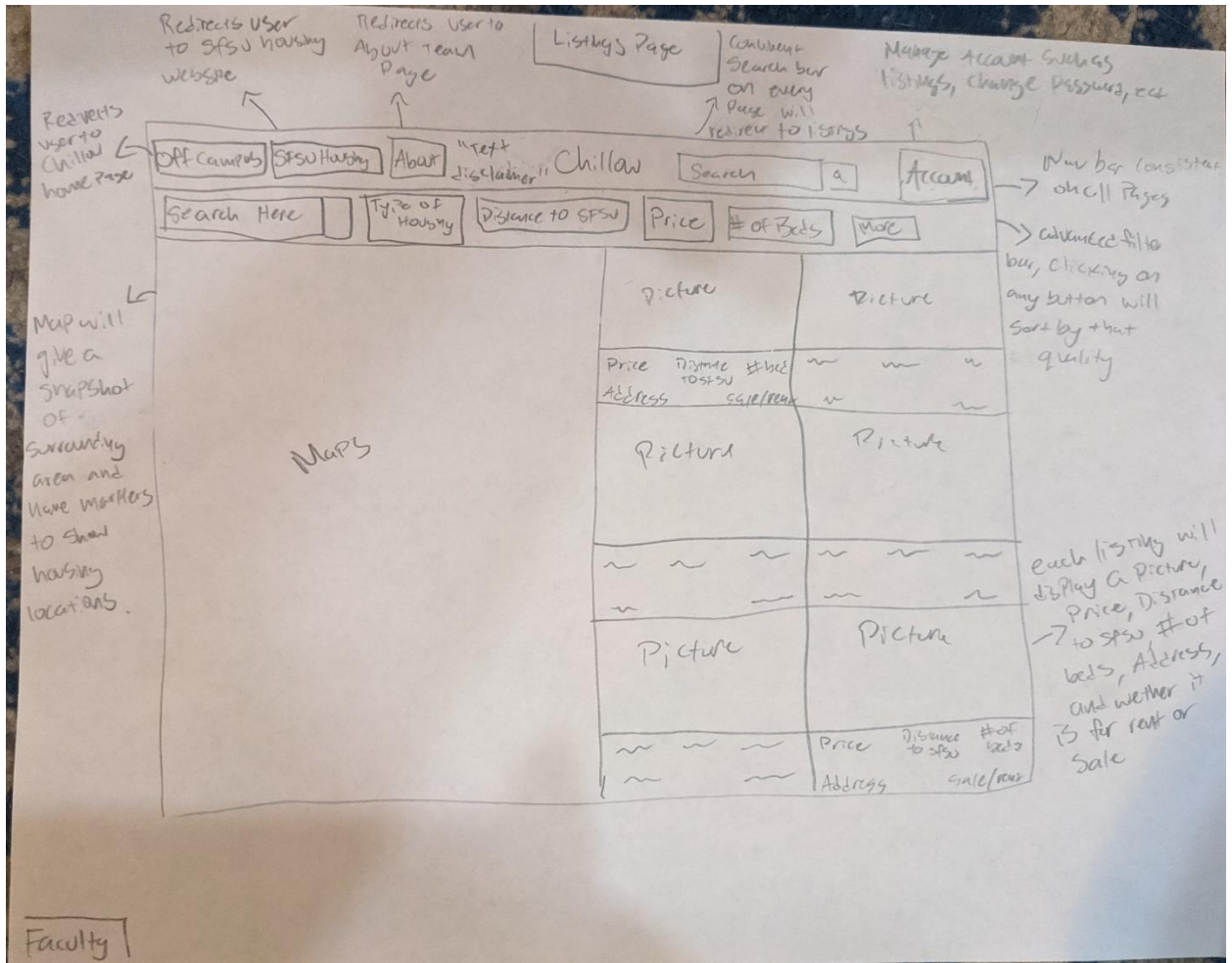


This is the homepage as seen by Landlord

#### 4. Faculty

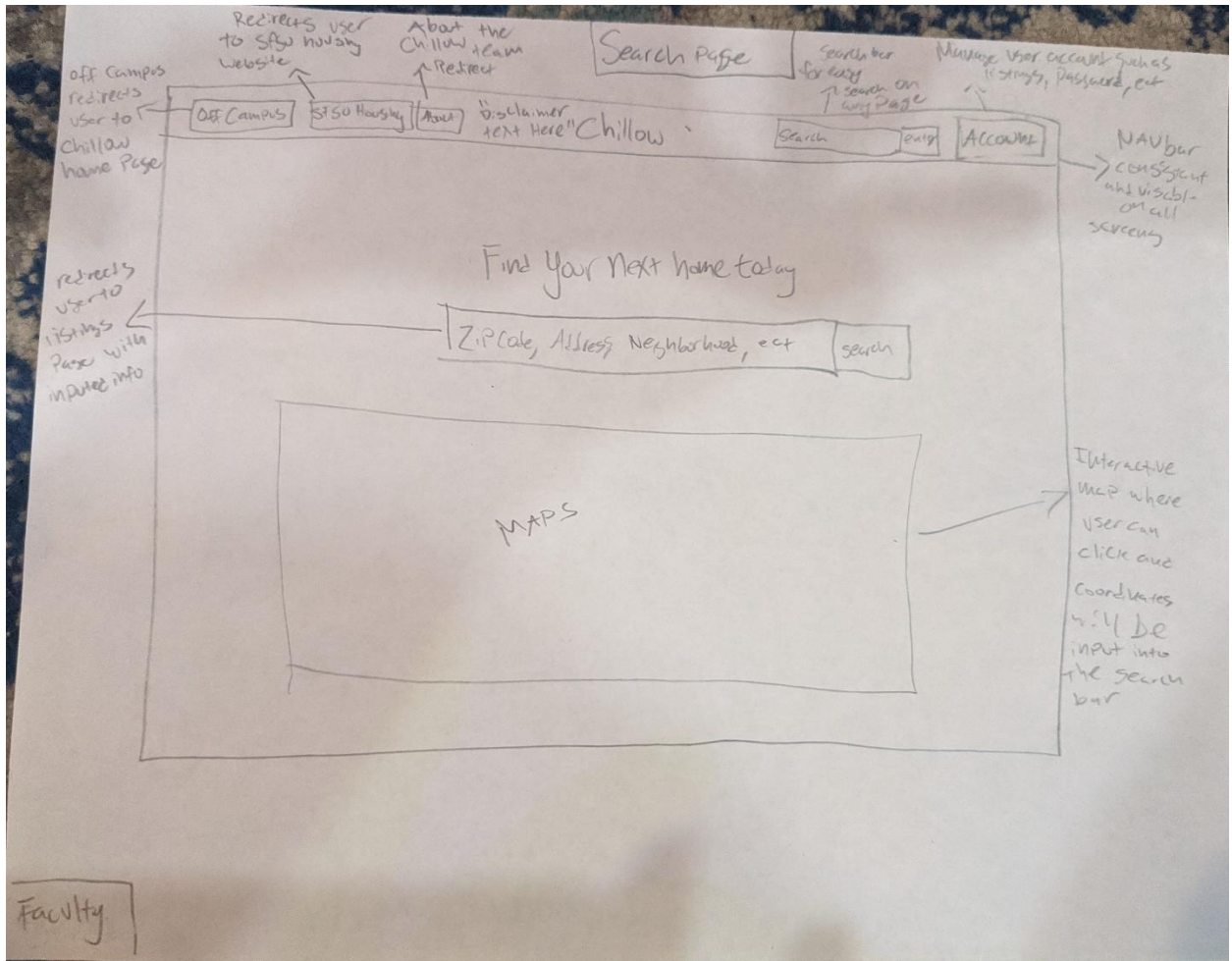


This is the about page as seen by Faculty Member



This is the listings as seen by Faculty Member





This is the homepage as seen by Faculty Member

---

## ***HIGH Level Architecture, Database, Organization***

---

### **DB Organization:**

Our database consists of the following schema:

- listings
  - listing\_id
  - lister\_uid
  - street\_addr
  - city
  - zipcode
  - num\_beds
  - num\_baths
  - square\_ft
  - parking
  - pets
  - description
- messages
  - msg\_id
  - sender\_id
  - recipient\_id
  - msg\_text
- users
  - uid
  - first\_name
  - last\_name
  - email
  - password
  - sfsu\_verified

### **Media Storage:**

For our media storage we are using an S3 bucket. This is a AWS cloud platform that allows us to create a storage location off of our git branches and work via the cloud

### **Search/Filter Architecture and Implementation:**

The search algorithm we use utilizes the built in %LIKE operator in SQL to search through the element such as street address, city, and zip code. Then it returns all matches of each instance.

### **APIs:**

In our APIs we are working on a Restful API using MySQL wrapped in Sequelize. The RESTful API will allow us to define the constraints that we will implement in creating the architecture of our webservice.

---

## ***Project Key Risks***

---

### **Skills Risks:**

With most of us coming from diverse backgrounds of programming. Our skill levels vary with each programming language we know.

To help this I have assigned a advanced level programmer to our less experienced programmers. This way there is the Mentor Mentee relationship and they can grow from learning and teaching one and other in fields they may feel more comfortable in.

### **Schedule Risks:**

As some of us have other classes, work, and commitments this poses a problem to the normal workflow environment of a real software engineering company.

Ways to combat this is our group chat and Trello to assign tasks and ensure that they are being completed in a timely manner.

### **Technical Issues:**

As students we are moving into unknown territories of coding and we may want to accomplish things in our program that may be out of our technical skill level.

Ways to solve this issue is to find programs and codes that have a lot of support. This is key so when we get stuck on something technical we can find the answer to a problem that someone may have had before.

**Legal/Content Risks:**

There may come the issue of paying for content that we want on our site. This is something our team will try to avoid at all costs.

If this becomes an issue we will look or ask for an alternative method to include on our webpage.

Another issue is that we must put together listings on our page which requires addresses and photos and there is the risk of using real address and photos. This can become problematic as our site is a housing site and a housing site without photos to show in the listing is as good as garbage.

---

## ***Project Management***

---

With the setup of my team it was very easy to assign them to their specific tasks. The assignments for Milestone 2 went down the line. The front-end team and I are tasked with doing the documentation and mockup storyboards. The back-end team and git master were tasked with the creation of the DB entries, vertical prototype, and media storage. Going forward this will be the way the team will be split with any team that may finish its part able to help the other teams at that team leader's discretion. By doing it this way it helps ensure that not too many people are helping if it is unnecessary.

For team organization we went with Trello as a place to broadcast all the tasks with out the fear of them getting lost in a thread like chat messenger. This makes it easier to assign tasks to a team and then divide them up by the individuals in the team.