

**SW Engineering CSC 648/848**  
**Section 02**  
**Fall 2017**

# HouseHunter

**Team 02 (Local)**

Udara Gunawardena, **Team Lead** | [ugunawar@mail.sfsu.edu](mailto:ugunawar@mail.sfsu.edu)

Vipul V Karanjkar

Aishwarya Laturkar

Mohan Maharjan

Larry Jiang

Savan Patel

**Milestone 2**

| Submission Date | Feedback Given |
|-----------------|----------------|
| 10/27/2017      |                |



## 1. Data Definitions V2:

- **Unregistered User:** An unregistered user shall have full access to the main parts of the website that involves browsing homes.
- **Registered User:** Users who have an account, and are logged into it, that are using the site. Registered users can be either an agent or buyer.
  - **Agent Account:** Users who have signed up as an agent, shall have access to the sell page, where they shall be able to build a listing and post to the website. They shall also have full access to the property browsing tool just as guest and buyer accounts. These users will have the most access to the website.
  - **Buyer Account:** Users who have signed up as a buyer will have access to everything EXCEPT the sell page. They add on two key points of functionality, ability to favorite listings and the ability to contact the agent selling any particular listing.
- **Administrator:** Users that have special privileges, and have the ability to remove posts from the site, remove items, issue warnings and bans from the site, and generally enforce the Code of Conduct for the site. Administrators also are responsible for helping users when needed.
- **Property:** The main product of the website, our home listings. It shall contain information about any given home that is currently for sale. It shall also contain the home address, price, the selling agent, and then the granular descriptions of the home.

- **Property Description:** Property descriptions shall include items like
  1. Home Type (Apt, Condo, etc.)
  2. Square footage, Lot size
  3. Number of Bedrooms/Bathrooms
  4. Year built
  5. Days on the market
  6. Agent Description
- **Images:** All images on the website shall be accessible by all types of users, from unregistered users to administrators. All images posted should be required to have no copyright for free usage. All images on the website should be posted by registered users with proper rights aka accounts of the agent type.

## 2. Functional requirements V2:

1. Non-registered user shall be able to search the property by using zipcode or city name. *Priority: 1*
  - 1.1. Search is the main tool of the website and will live towards the upper center of each page to make sure the user is able to quickly start another search.
2. Non-registered user shall be able to browse the entire website's home listings. *Priority: 1*
3. Registered Agents shall be able to post property listings for sale and respond to registered buyers. *Priority: 1*
  - 3.1. Posting properties will be handled by the sell page, this is only accessible by registered agents.

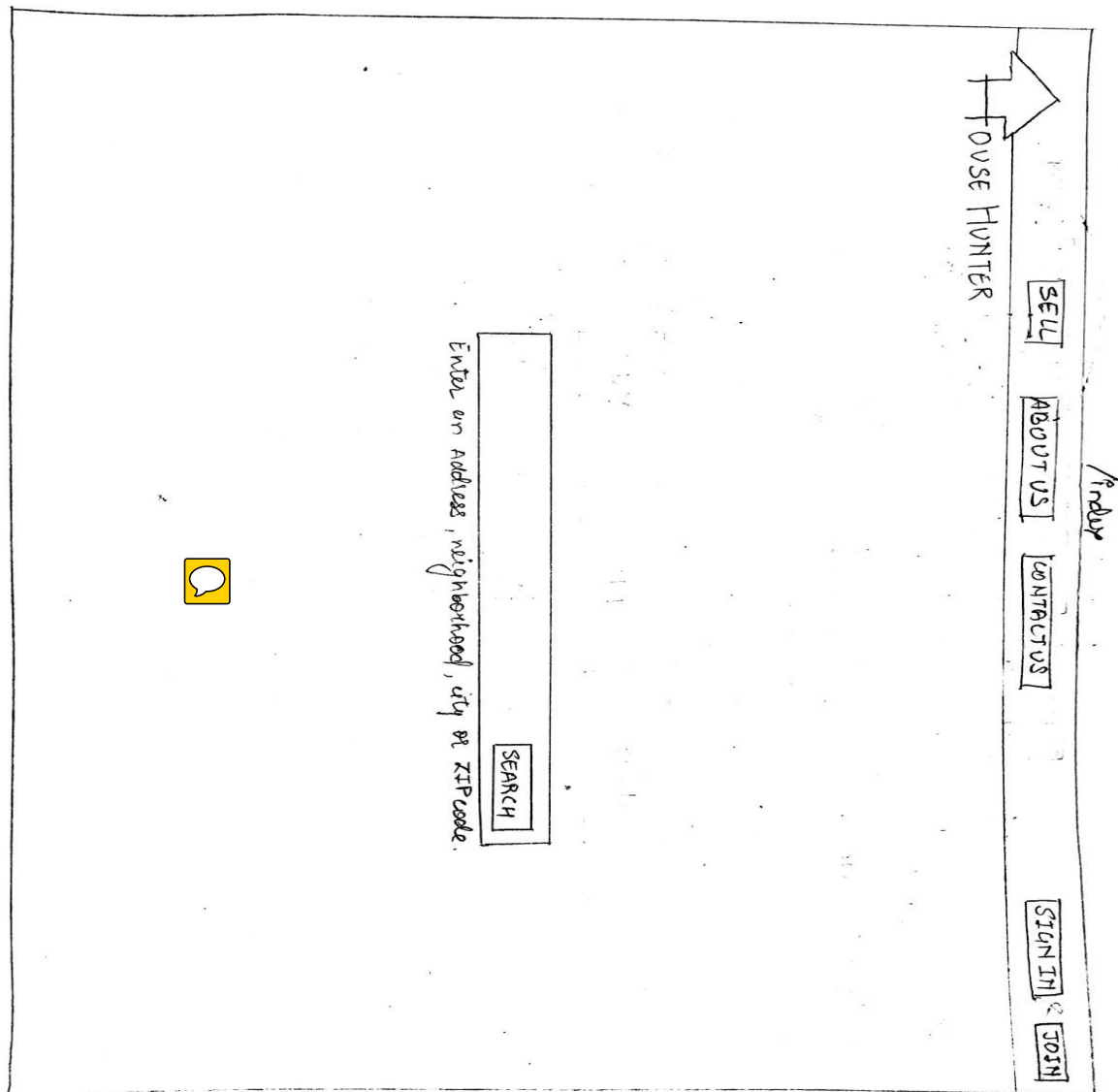
4. Registered buyers shall be able to contact property agent for any listing.   
*Priority: 1*
  - 4.1. The key functionality of the website lays in its ability to connect buys to the agent who's selling homes.
  - 4.2. In an effort to eliminate spam we've decided to implement a system where only registered users can use the contact agent system.
5. Admin shall be able control property listings and user accounts. *Priority: 2*
  - 5.1. Full access to the website's backend will be granted to the admin, giving them full power to delete, modify, add, users or listings.
6. Both Non-registered users and Registered users shall be able to see the location of searched property on maps API. *Priority: 2*
7. Registered buyers shall be able to favorite or flag property listings. *Priority: 3* 
  - 7.1. Favorites shall be able to be accessed in a list form so registered buyers can quickly see key properties.
  - 7.2. Flagged properties will reflect on the administrator's side where he or she can decide whether to take action on a listing.
8. Both Non-registered users and Registered users shall be able to add filters onto their search in order to limit their search to their needs. *Priority: 2*
9. Both Non-registered users and Registered users shall be able to contact us through the contact page, us being the developers behind the website.  
*Priority: 2*

### 3. UI Mockups and Storyboards(high level only):




#### 1. **Landing Page** ( */index* ) :

The home page of our website. When all users enter our website they will see a simple webpage with a search bar for searching and browsing property listings. At the top, a navigation bar with options to contact agents, sign in/out, register, and to contact website admins.



## 2. Registration page (/register):

When browsing our website while not signed in the navigation bar at the top right will have an option for unregistered users to join our website. When the join button is pressed, a registration page will appear for the user to complete information for an account. If the registering user decides to cancel their registration they may click on the cancel button which would bring them back to their previously viewed page.



</register>

FIRSTNAME  \*

LASTNAME  \*

EMAIL ID  \*

CONFIRM  \*

EMAIL ID  \*

PHONE  \*

PASSWORD  \*

CONFIRM  \*

PASSWORD  \*

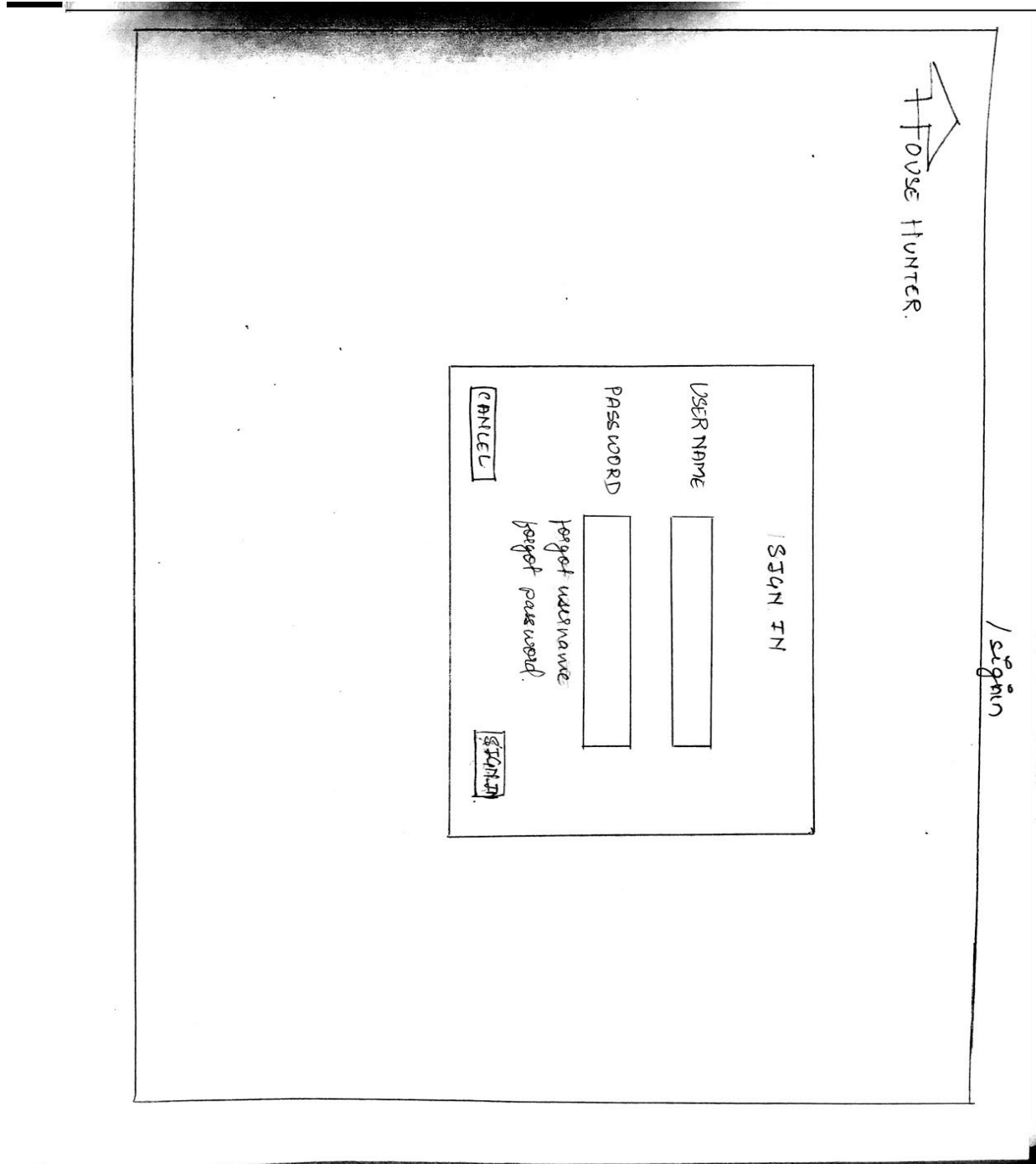
☐ BUVER ACCOUNT OR ☐ I AGREE

☐ BELEUR ACCOUNT

\* REQUIRED

### 3. Sign-In page ( /signin):

A registered user who is signed-out may sign-in at anytime when browsing our website. The sign-in button will be on the navigation bar at the top right corner of all webpages.

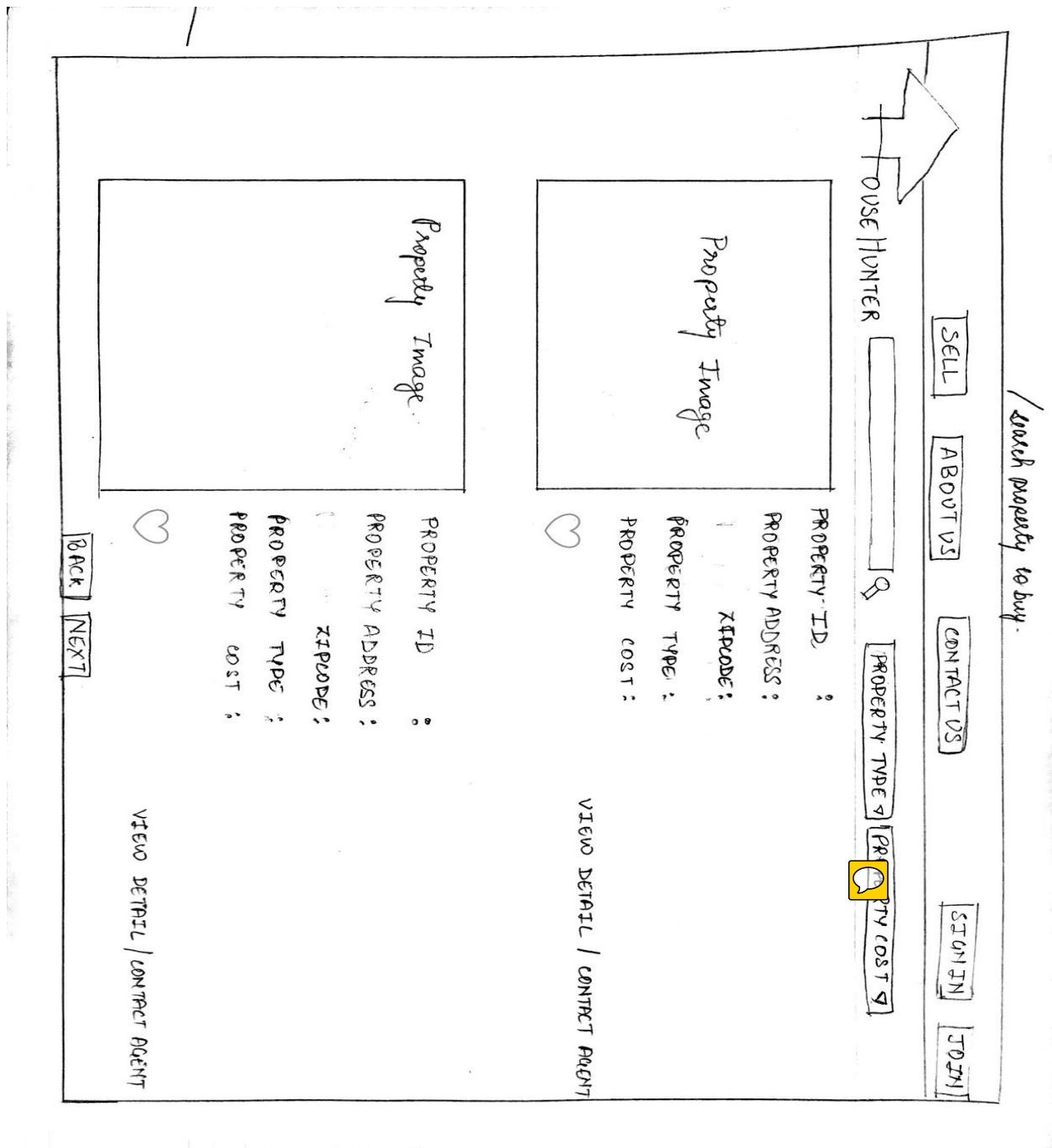


#### 4. Listing Page (/search) :

Any user may search and browse our property listings. When browsing through the property listings they can see a list of properties with information and an image of the property for users to view. While in this page the user have many



options to choose from, view detail, contact agent, and go to the previous or next page listing. Registered users may contact an agent here.



##### 5. Property detail ( /view page of a property):


If the user decides to view more detail of a chosen property listing they will come to this page. The property details will be displayed for the user which also

### 6. Contact Agent ( /contact agent) :

Hand-drawn wireframe of a 'House Hunter' web application. The layout includes a top navigation bar with links for 'SELL', 'ABOUT US', 'CONTACT US', 'MY PROFILE', and 'USERNAME'. A large central area is labeled 'HOUSE HUNTER' with a double arrow icon and the text 'PROPERTY IMAGES'. To the right of this is a 'SEARCH' button. Below the search bar is a 'MAP' section with the text 'GOOGLE MAP FOR THE SELECTED PROPERTY'. On the left side, there is a 'CANCEL' button and a 'CONTACT AGENT' button. The bottom of the page has a dashed line with the text 'LOG OUT'.

If a registered user decides to contact an agent on a specific property listing they will be prompt to this page. Agents will be notified when all required fields are completed and when the user presses the send button.

*/contact agent*

 HOME HUNTER

(Auto fill)

FIRST NAME:

LAST NAME:

EMAIL ID:

PHONE:

PROPERTY ID:

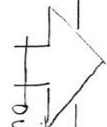
QUERY:

## 7. Selling (/sell a property):

In order to sell a property on our website the user must be registered as an agent. A registered agent may post property listings. Registered agents have the options to add images and input property information and description for displaying on the property listings page.



*/sell a property*

 HOME HUNTER

Add image

Add image

Add image

PROPERTY ADDRESS

ZIP CODE

PROPERTY TYPE

PROPERTY COST

PROPERTY DESCRIPTION

FIRST NAME

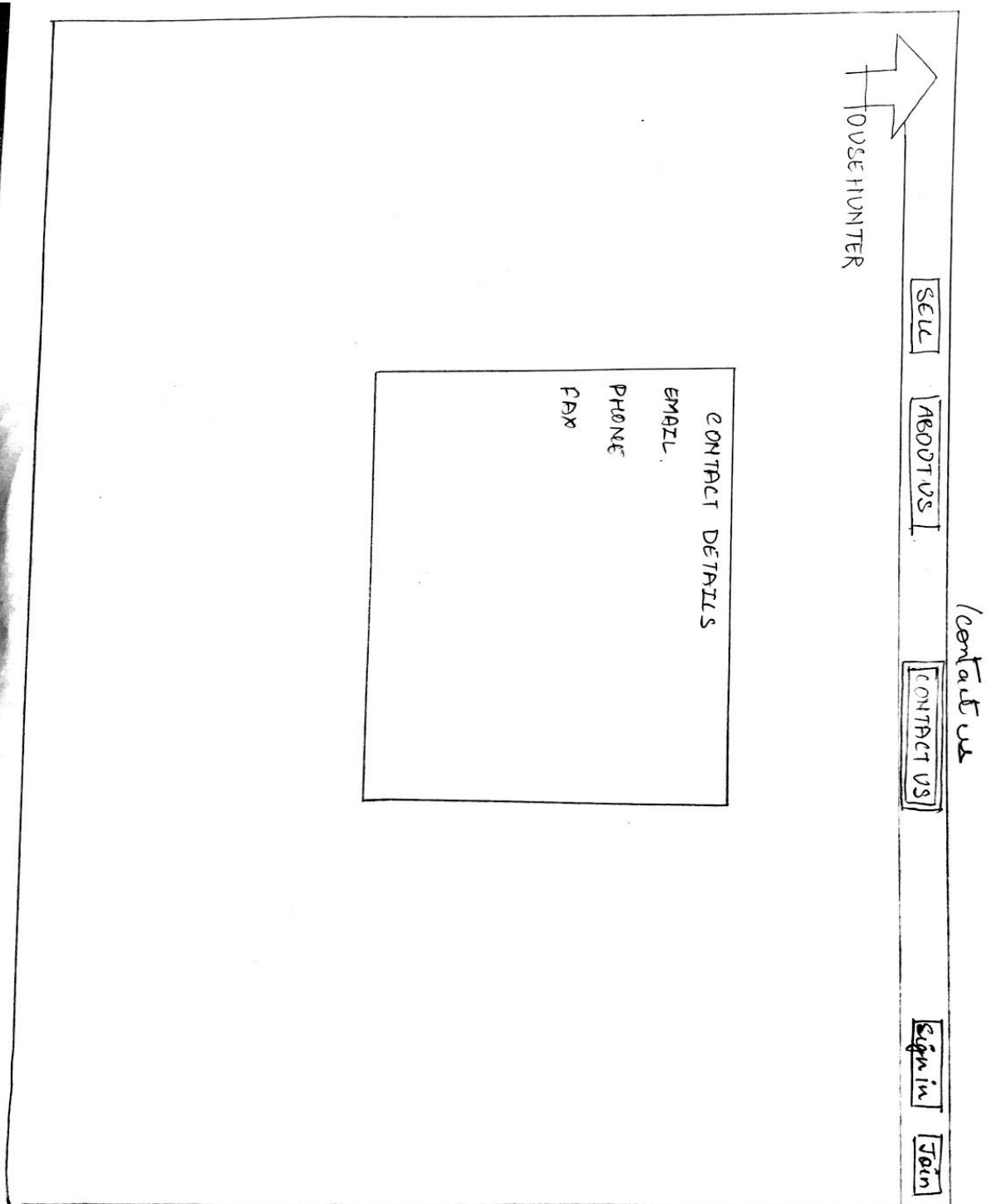
LAST NAME

EMAIL

PHONE

## 8. Contact ( /contact us) :

Our contact page will display emails and phone for users to contact us



## 4. High level Architecture, Database Organization:

**1. High level Architecture of the code:** We will be using MVC structure wherein the Model will hold the logic to get the listings according to the keyword entered in the search box if there exists the same keyword in the listing table. Controllers will contain several routes for search(/search), about me pages(/about), etc. And Views will contain the HTML pages which will be displayed on UI.

The Buyers and agent table both contain basic information about them such as their name, email address, phone number, profile picture, and website url for realtors. As for the Listings table, it holds informations about the actual listings like the images, addresses, the number of bedrooms and bathrooms, the price, and a foreign key that connects each listing to a specific agent. Our Favorites table includes 2 foreign keys to connect with the Buyers and agent table that will allow us to keep track of which listings a specific buyer has their eye on.

The Admin table stores the username and password of every agents, buyers and registered users with their respective foreign keys associated with them.

- /Home
  - Search
  - Sign In
  - Join
  - Sell
  - About Us
  - Contact Us
- /Sign In
  - Username
  - Password
  - Forgot username
  - Forgot password
- /register
  - Firstname

- Lastname
- Username
- Email
- Confirm email
- Phone
- Password
- Confirm Password
- Join as a buyer or Join as a Seller
- I agree

- /Listing


- Property Id
- Property Image
- Property address
- Property zipcode
- Property cost
- Property type
- Favourite
- Contact Agent
- View Details


- /View details of a property

- Property Id
- Property Images
- Property address
- Property zipcode
- Property cost
- Property type
- Property map
- Contact Agent

- /Contact Agent
  - Firstname(Autofill)
  - Lastname(Autofill)
  - Username(Autofill)
  - Email(Autofill)
  - Phone(Autofill)
  - Property Id(Autofill)
  - Query
  
- /Sell
  - Add Property Image
  - Add Property address
  - Add Property zipcode
  - Add Property cost
  - Add Property type
  - Firstname(Autofill)
  - Lastname(Autofill)
  - Username(Autofill)
  - Email(Autofill)
  - Phone(Autofill)

## 2. DB organization:

- Items (itemid:integer, username::string, name string. Price :double.
- Description:string, category: string, condition ; string, image:string.
- Users (username:string, password:string. Firstname:string, lastname :string, email: string, zipcode: int.
- Password will be encrypted.

**3. Media storage:** We are trying to use DB BLOBS at present, but if can't figure that out before the vertical prototype's deadline, then will switch to  file system. If we use a file system, then the link to the image will be stored in the database. Standard images file will be accepted.(accept="image\*"). Thumbnails will be



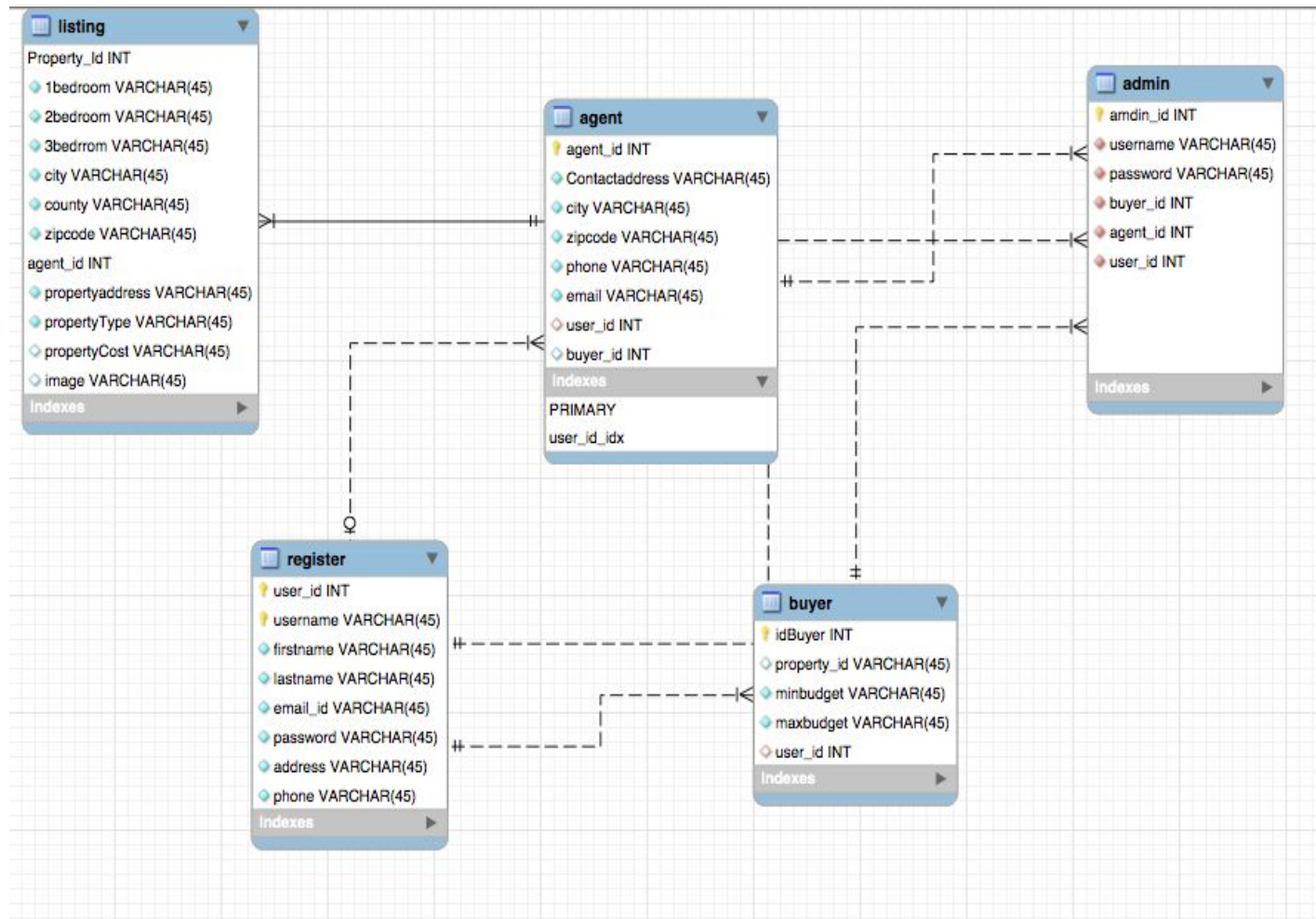
created on upload of the image, scaled down to around 200 x 200 pixels, maintaining aspects ratio.

**4. Search/filter architecture and implementation:** Our Homepage will include a search box where if a user just clicks on the search box, then all the listings will be displayed on /search. The user can search by some keywords such by the city, by zip code, etc., then only the specific listings according to the keyword will be displayed using %LIKE SQL query.

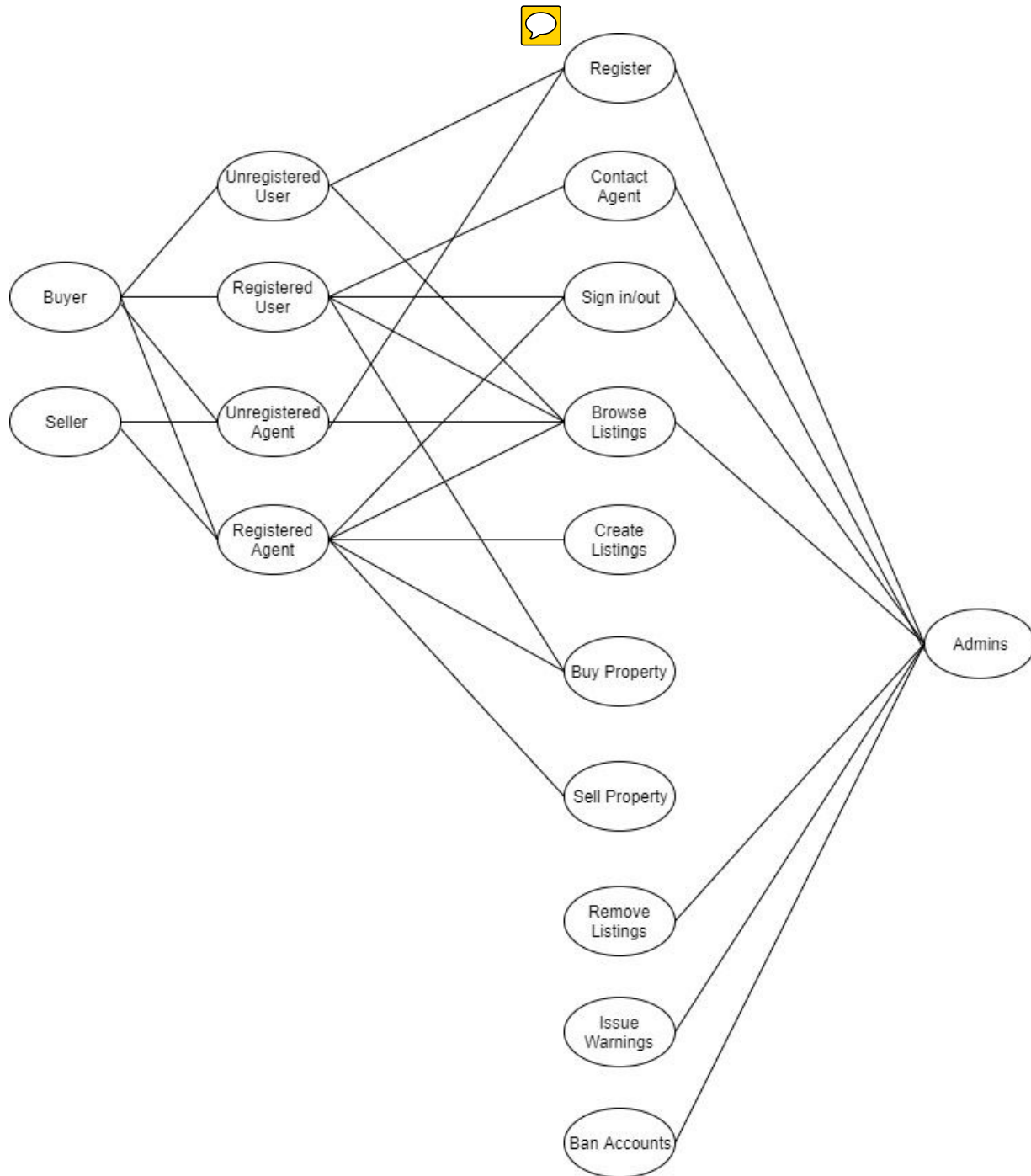
**5. Our own API:** One of our Real Estate website's API is /home. This will be our homepage and also default starting page(on /). This API will include a horizontal navigation bar which will include a logo of our website that will redirect to our homepage(/home). The nav bar will also include some tabs like About Us(/about) which will lead to team member details, Sell where an agent will post his advertisement to sell a property, and also a tab to Sign In(for Registered users) and Join(for new users who want to join). This whole navigation bar will be available on every page. Also, the important part of this API will be a search text box. This search box will allow a user to search the property according to a keyword such as a city name, state, or zip code.

## 5. High level UML Diagrams:


### 1.High level UML class diagrams:




## 2. UML Component and deployment Diagrams:



## 6. Key Risks:

- **Skills Risks:** Our team is experienced in development, however if we had to identify weaknesses it would be the fact that we have never actually developed a full website like this. This creates an interesting yet challenging environment for us to work in.
  - Because of this we have had to put in more effort to learn web technologies like node.js in order to deploy a useable product.
- **Schedule Risks:** This startup has 5 milestones throughout each of which have very specific requirements and specific deadlines. These are designed to ensure proper development is done throughout the cycle. We shouldn't have any issues adhering to these deadlines as our team is very driven.
  - Proper communication is key to making sure all deadlines are met with a product that is concise. We will be making sure all team members exercise full open communication .
- **Technical Risks:** The main technical risks lie in the fact that we are deploying the website on a remote amazon  web server. Meanwhile we have multiple group members to manage, all pushing through to git. There are many moving parts involved.
  - Outlining the workflow prior to starting work has been a key in ensuring we are all on the same page.
  - Our prior knowledge in git has helped ensure we aren't running into merge conflicts.
- **Teamwork Risks:** Our team works very well together, streamlined communication between backend, frontend and leads have helped ensure everyone is on the same page.

- Leverage of communication services such as whatsapp and slack have ensured this.
- **Legal/Content Risks:** Legal risks vary depending on deployment. Assuming it stays deployed on the AWS in private setting, we have full control over what images the site hosts. Thus in that case there shouldn't be any legal risks. However if we assume the site gets deployed publically, we could run into risks of users uploading photos/videos that aren't theirs.
  - Ways we shall fix this issues include  the following:
  - Given the first scenario, we will have a disclaimer posted on every page indicating that the website is purely a college project for a software class.
  - In the second scenario we would append a user agreement that ensures all photos uploaded are of their ownership, and any copyrighted material will be of their own responsibility.