Software Engineering CSC 648/848 Section 01 Fall 2017
Department of Computer Science
SAN FRANCISCO STATE UNIVERSITY

Milestone 2

Gator Realtor

Team 11 - Local

Alexander Tung : alextung94@gmail.com

Sukhjit Singh

Ilya Nemtsov

Nicholas Szeto

Ralph Acosta

Yangshan Huang

Date	Version	Description
10/27/2017	1.0	First draft

1. Data Definition

1. **Users**: A user can be Registered or Guest.
    a. **Registered User - Buyer**: any user who has created an account.
        i. **Email**: Email address that was used to register an account.
        ii. **Password**: secret number of characters that provides access to the account.
        iii. **Phone**: particular phone number that associated with account.
        iv. **Dashboard**: used by buyers to see messages.
    b. **Registered User - Agent**: Registered User who can create listings for homes and receive messages from other registered users.
        i. **Dashboard**: used by agent to see listings and messages.
    c. **Guest User**: any user who can browse the site for content (homes). This type of users *can not list homes*, but can reach out to any seller.
2. **Administrator (Admin)**: is a special User who have some privileges that allow him or her to remove illegal content posted by agent; or even ban agent for not following the site rules (Terms and Conditions).
3. **Listing Content**: a house/ apartment/ condo/ townhome for sale.
    a. Type of Home
    b. Year Built
    c. Number of Bedrooms
    d. Number of Bathrooms
    e. Heating / AC
    f. Lot Size
    g. Image: jpeg, png, jpg (size: 3.0 Mb)
    h. Location
    i. Price
    j. Additional details
        i. Parking
        ii. HOA
        iii. Appliances
        iv. Amenities
        v. Other

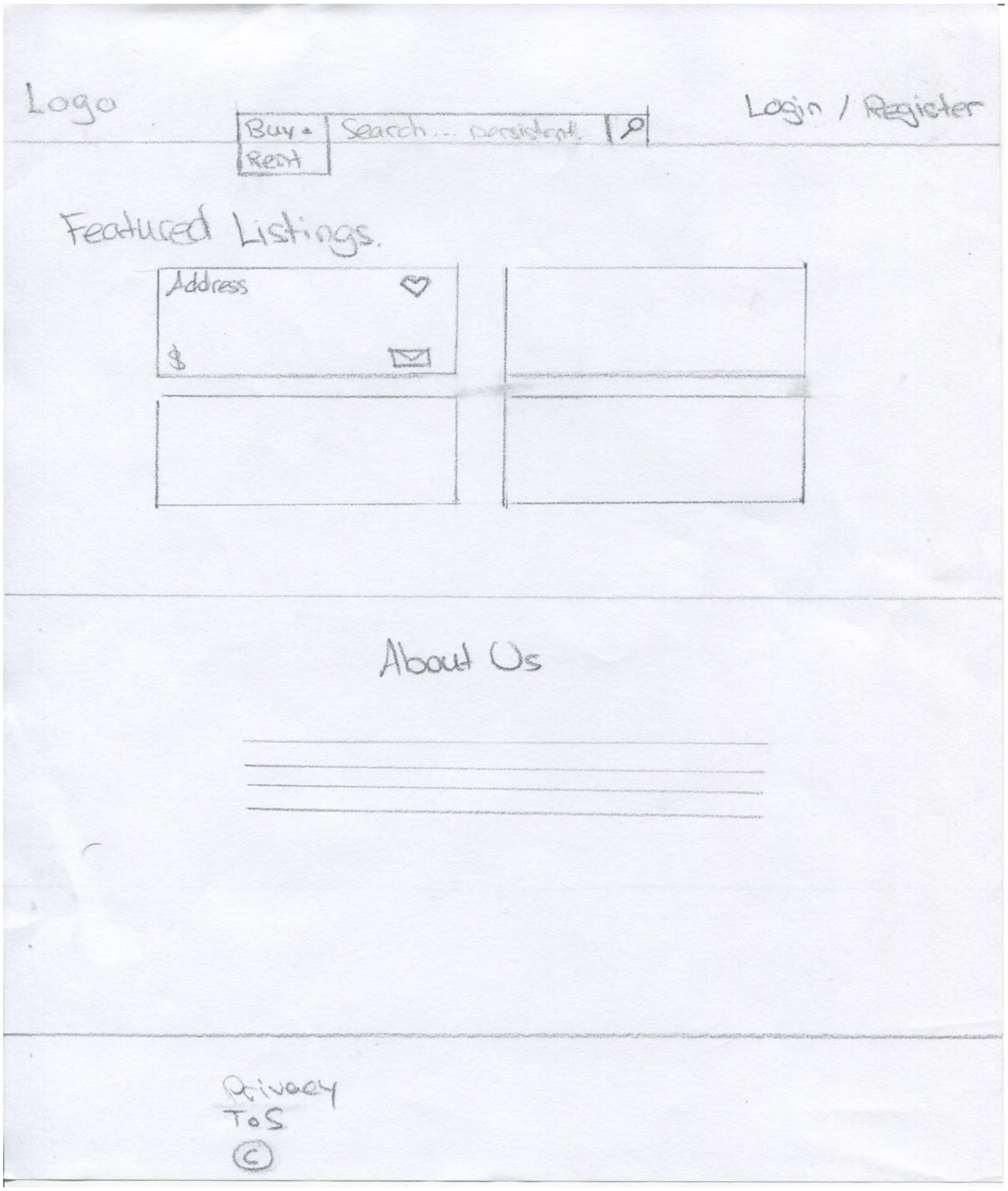
2. Functional Requirements

1. Guest users shall be able to register and login	Priority 2
2. Terms of service shall be present during registration	Priority 3
3. All users shall be able to browse for listings	Priority 1
4. All users shall be able to search listings	Priority 1
5. All users shall be able to filter listings	Priority 1
6. All users shall be able to sort listings	Priority 1
7. All users shall be able to see contact information	Priority 1
8. Buyers shall be able to contact agents	Priority 1
9. Agents shall be able to create listings	Priority 1
10. Agents shall be able to edit or remove listings	Priority 2
11. Agents shall be able to update their listings	Priority 3
12. All data listings shall display contact information	Priority 1
13. All data listings shall require at least one image	Priority 2
14. All data listings shall show location on map	Priority 2
15. Administrators shall be able to suspend accounts	Priority 3
16. Administrators shall be able to delete listings	Priority 3

3. UI Mockups

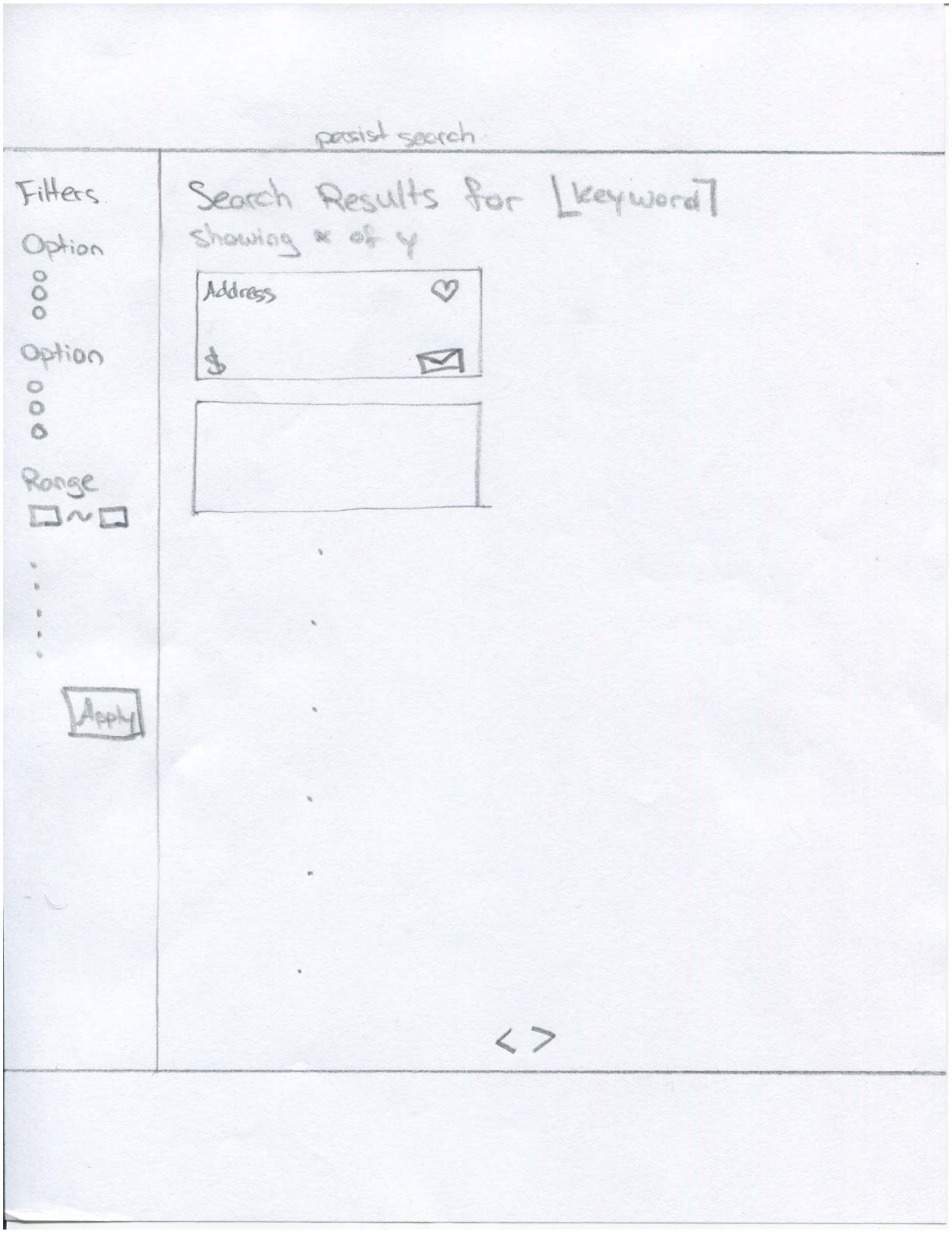
1.1 Guest User

Joey visits the Gator Realtors website for the first time, enters keywords into the search bar and clicks on the magnifying glass to process his search.



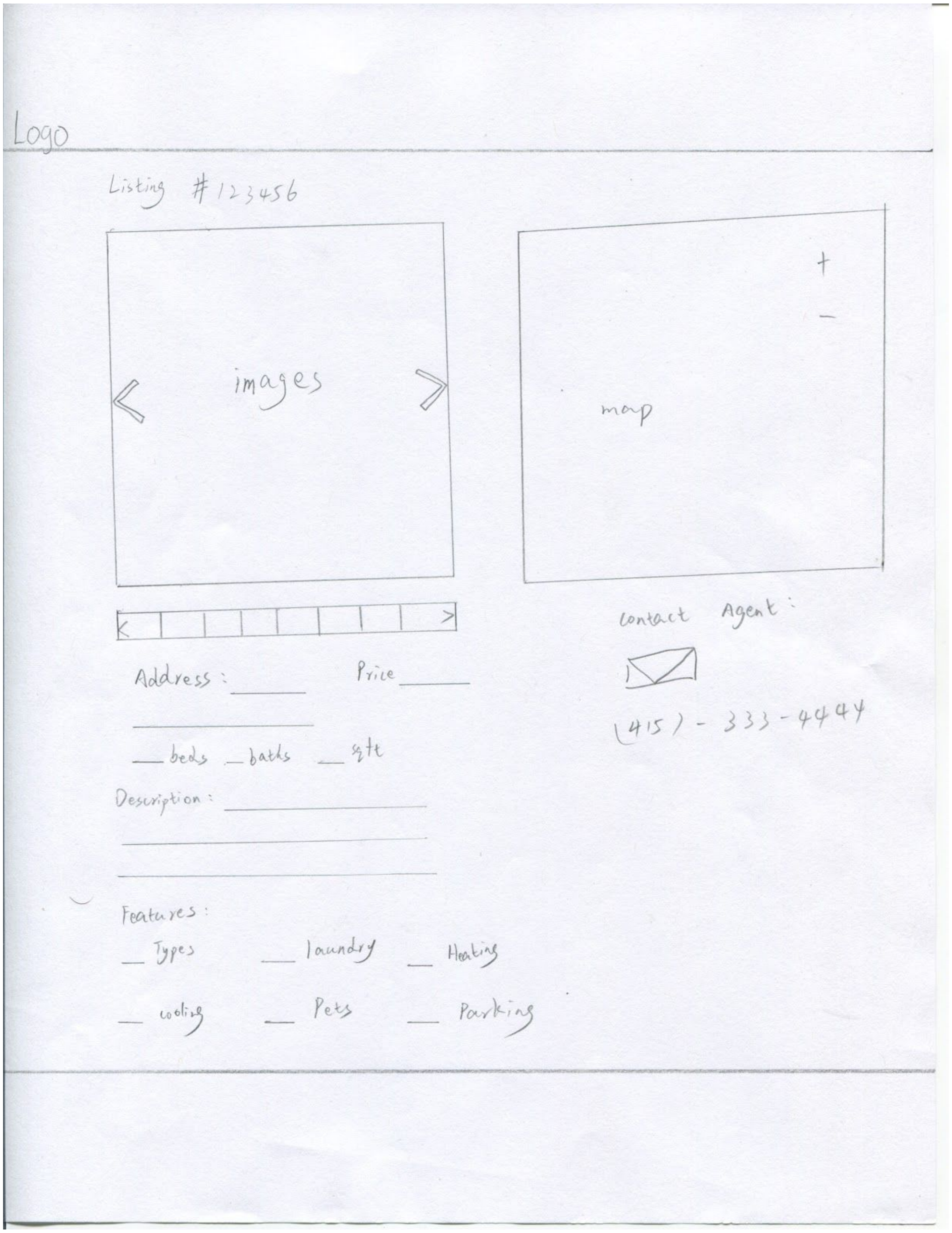
1.2 Search Results Page

Joey is now presented with another page that displays search results. He is able to choose filters to refine his results. He finds a home he is interested in he clicks on the result.



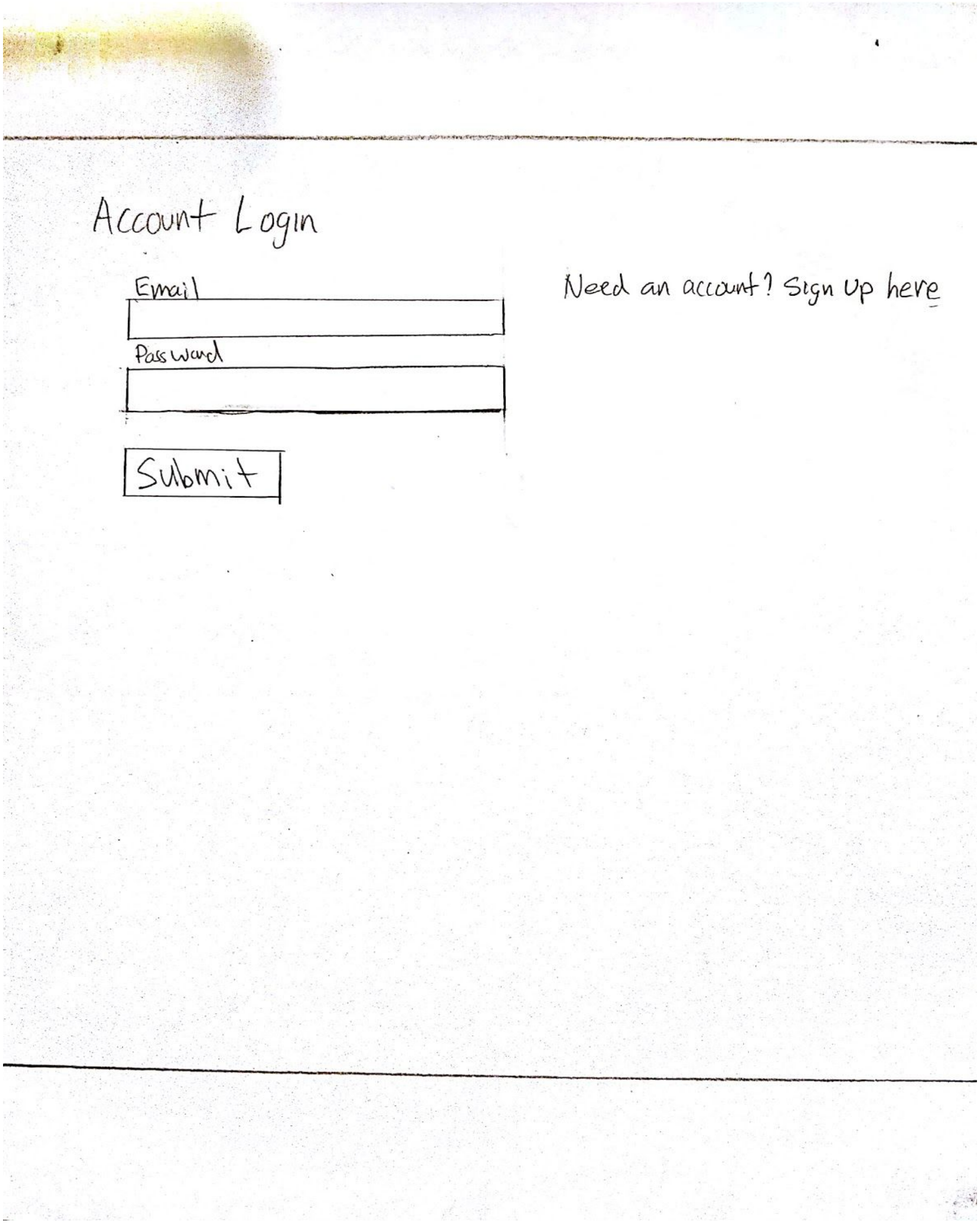
1.3 Listing page

Joey is now presented with the listing's detailed page and he can view different images and see the description of the home. He is interested in seeing this home in person so he clicks on the contact agent button.



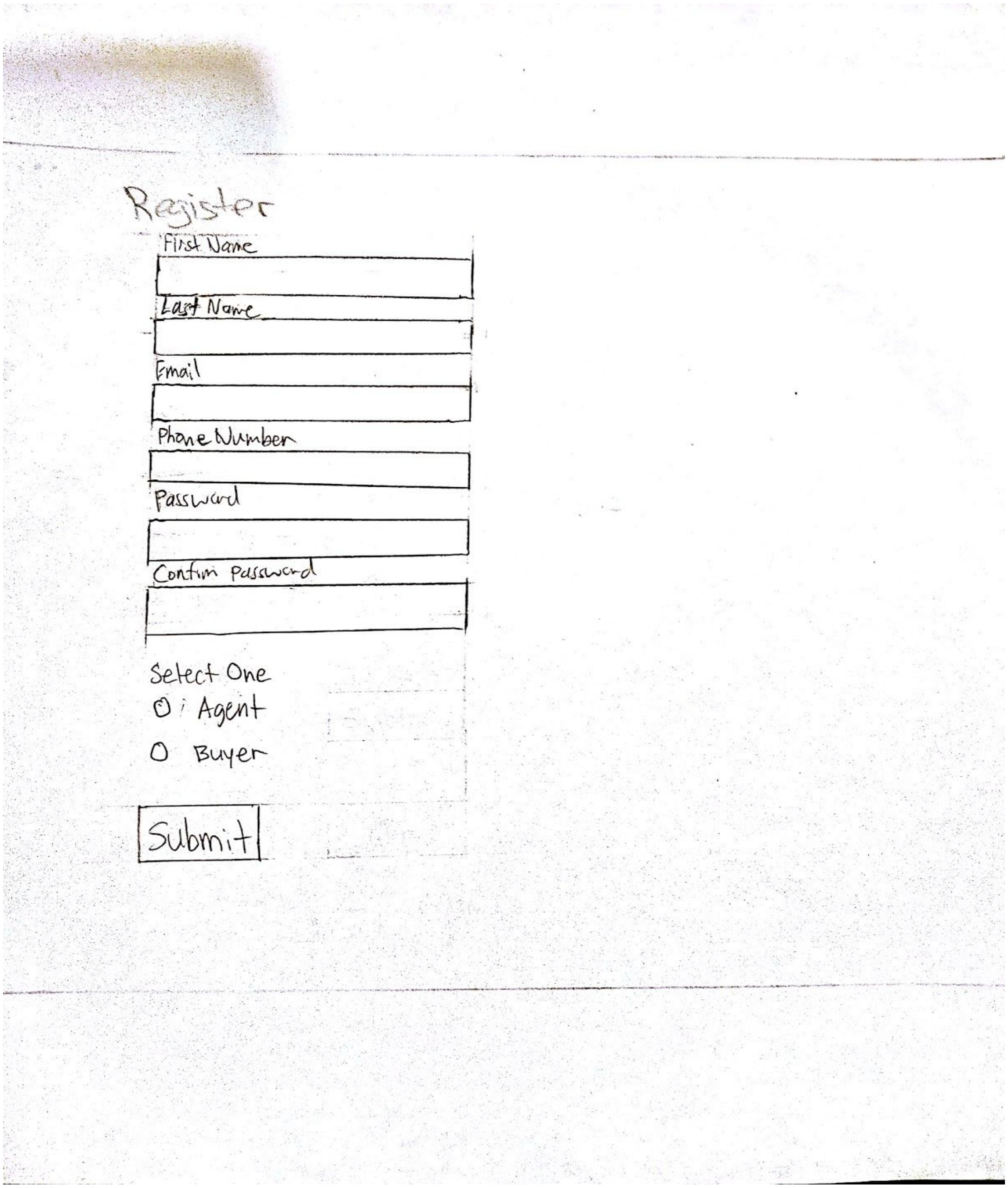
1.4 Login Page

Joey is now prompted to login to his account, but he doesn't have one. Joey then sees the message "Need an account? Sign up here," and clicks on the link. Once he registers he can contact the seller and ask for an appointment to look at the house.



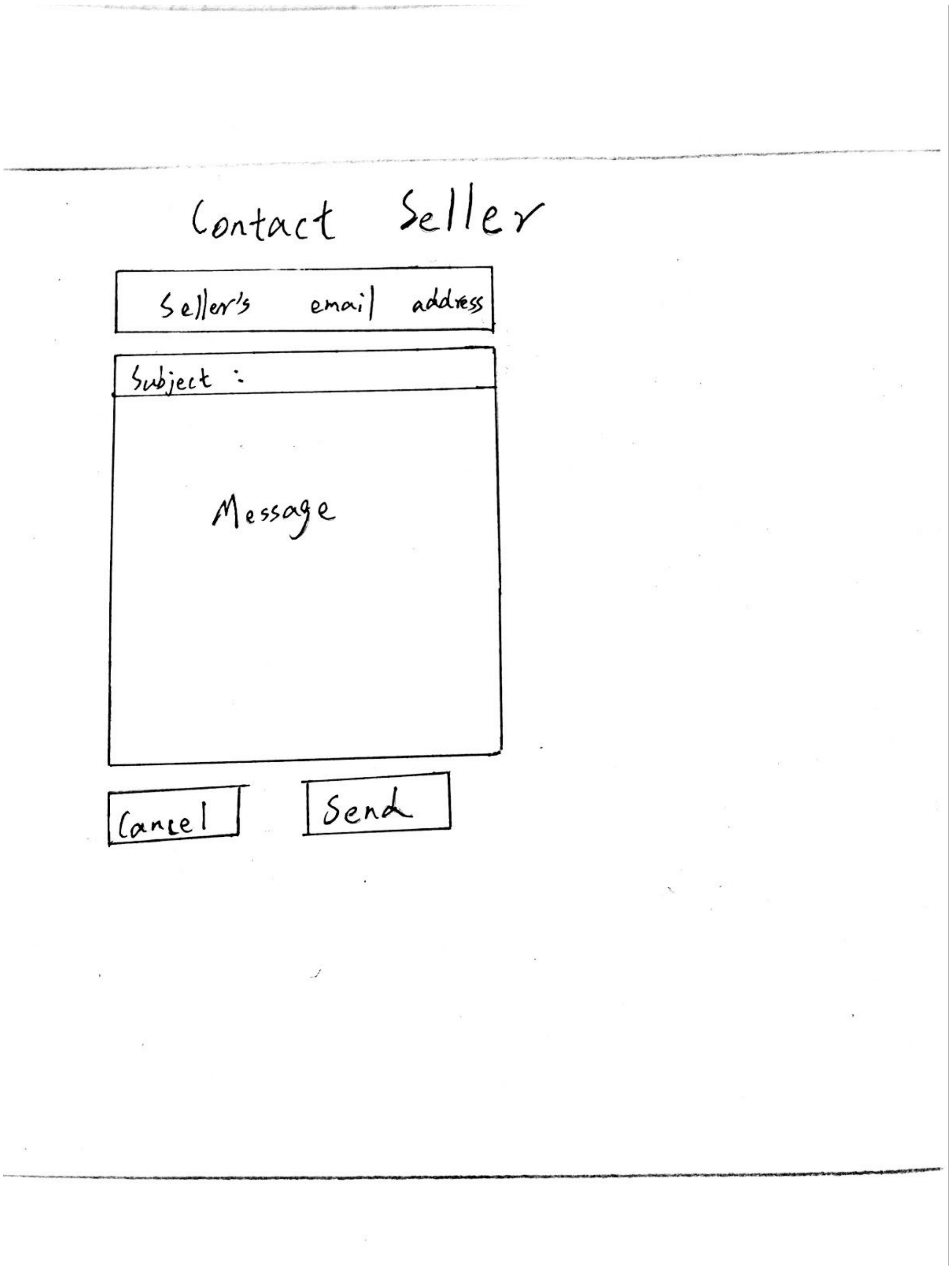
1.5 Register Page

Joey is now sent over to the register page, where he can input his credentials before proceeding to contact the seller.



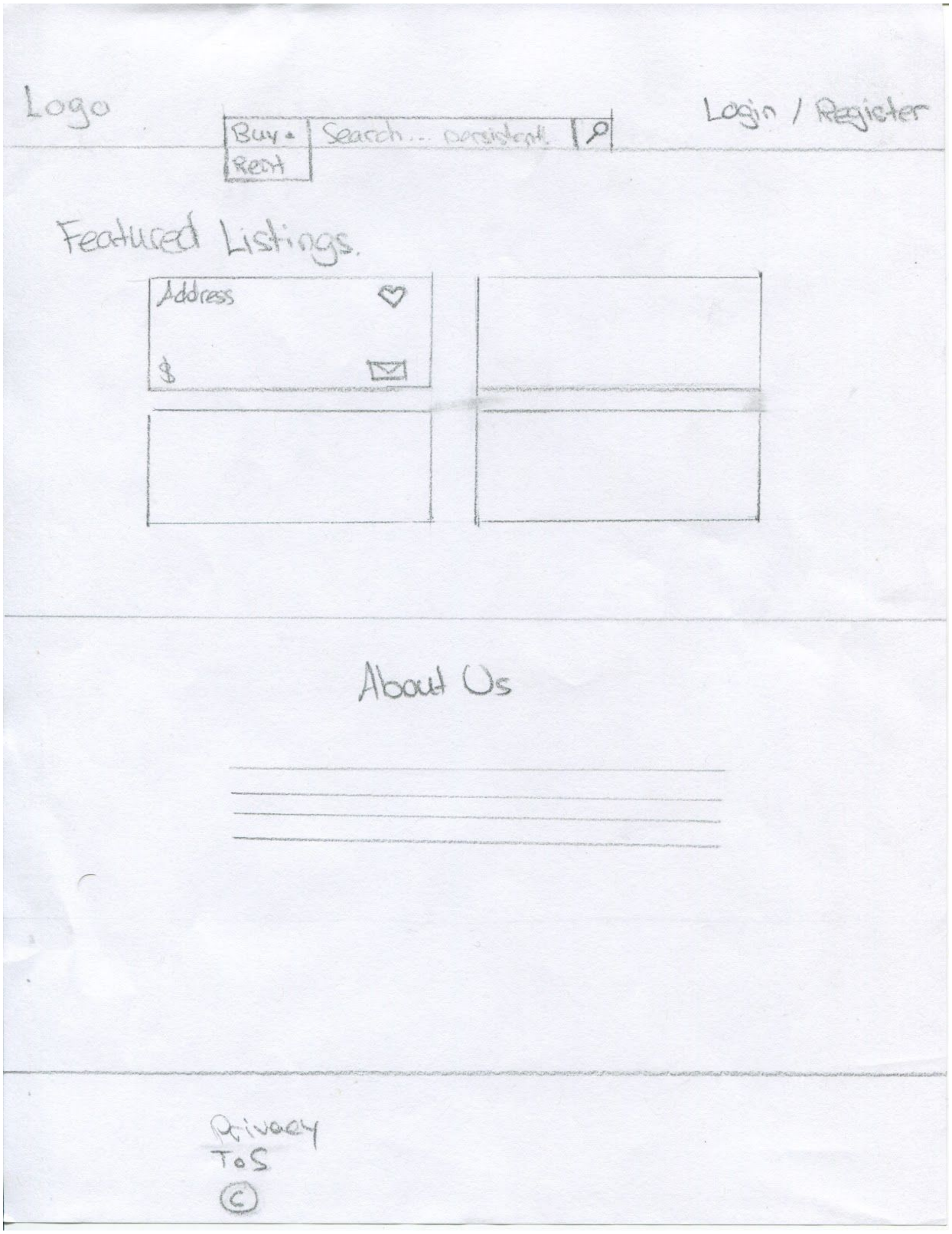
1.6 Contact Seller Page

After creating an account Joey is now allowed to contact the seller to see if he can schedule a meeting to look at the home.



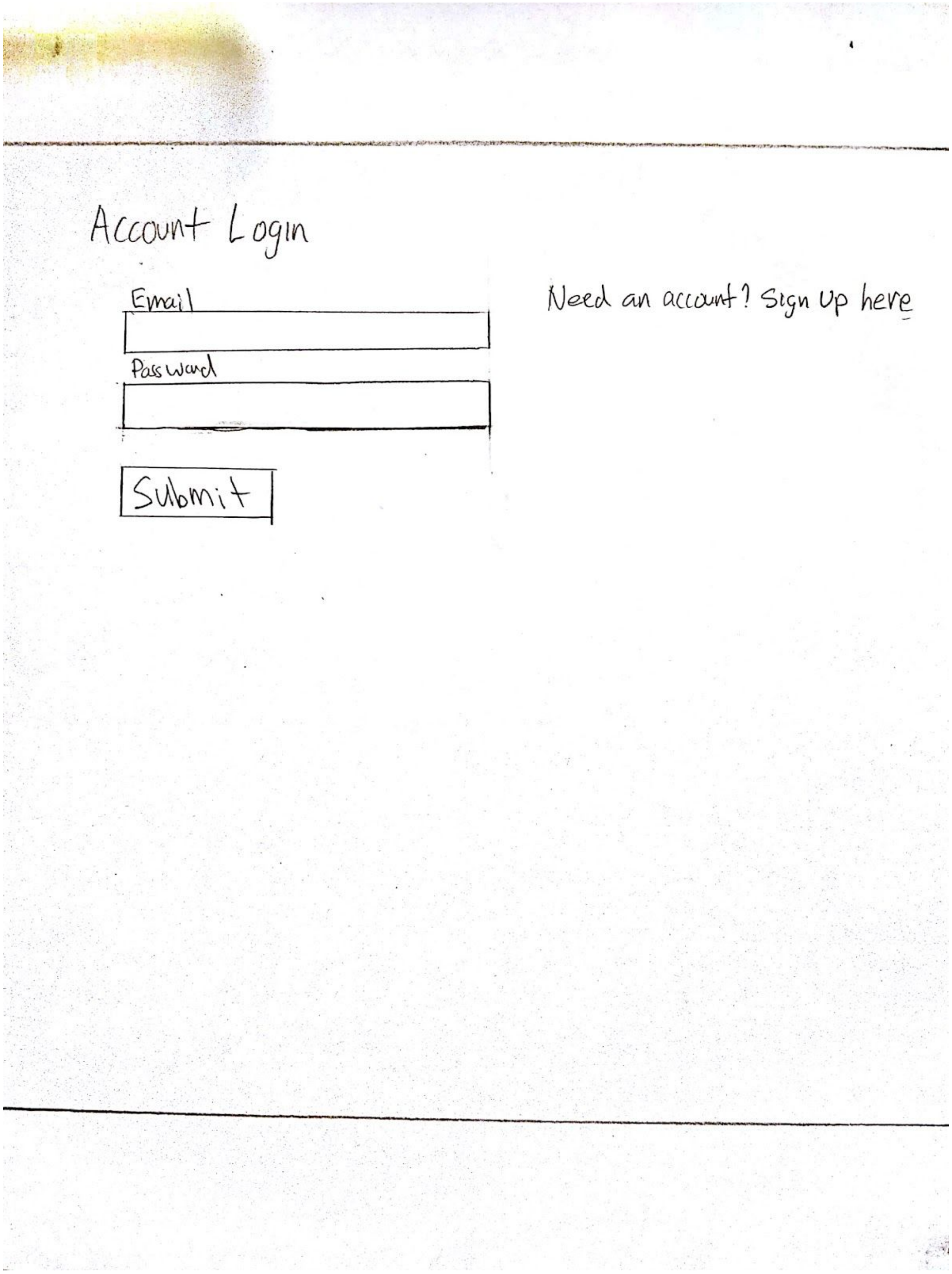
2.1 - Registered User - Buyer

Dawn is already a registered user and wants to continue her buying session. She goes to the Gator Realtors website and clicks in the login/register button.



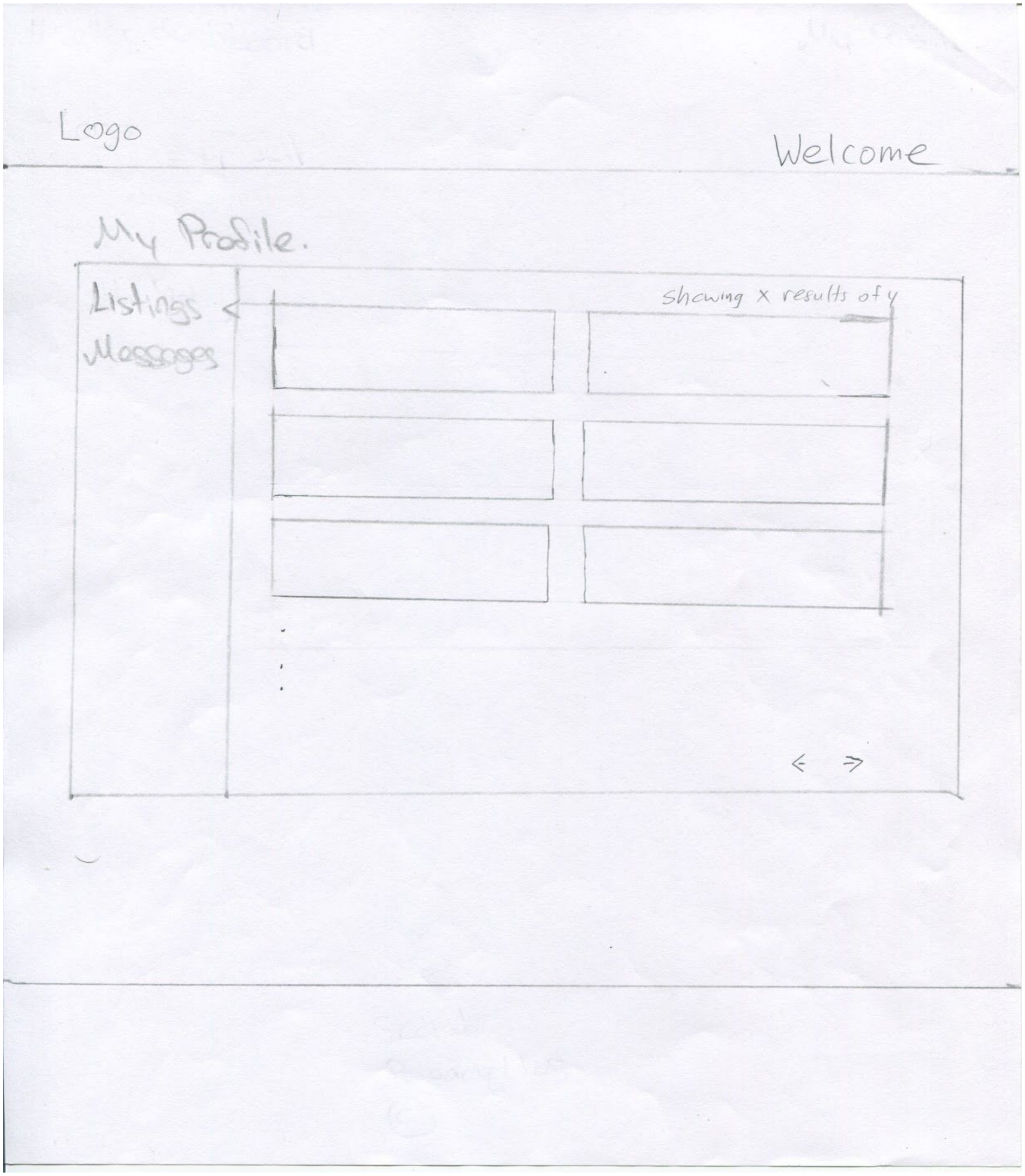
2.2 - Account Login Page

Dawn is now presented with the login screen where she will input her credentials.



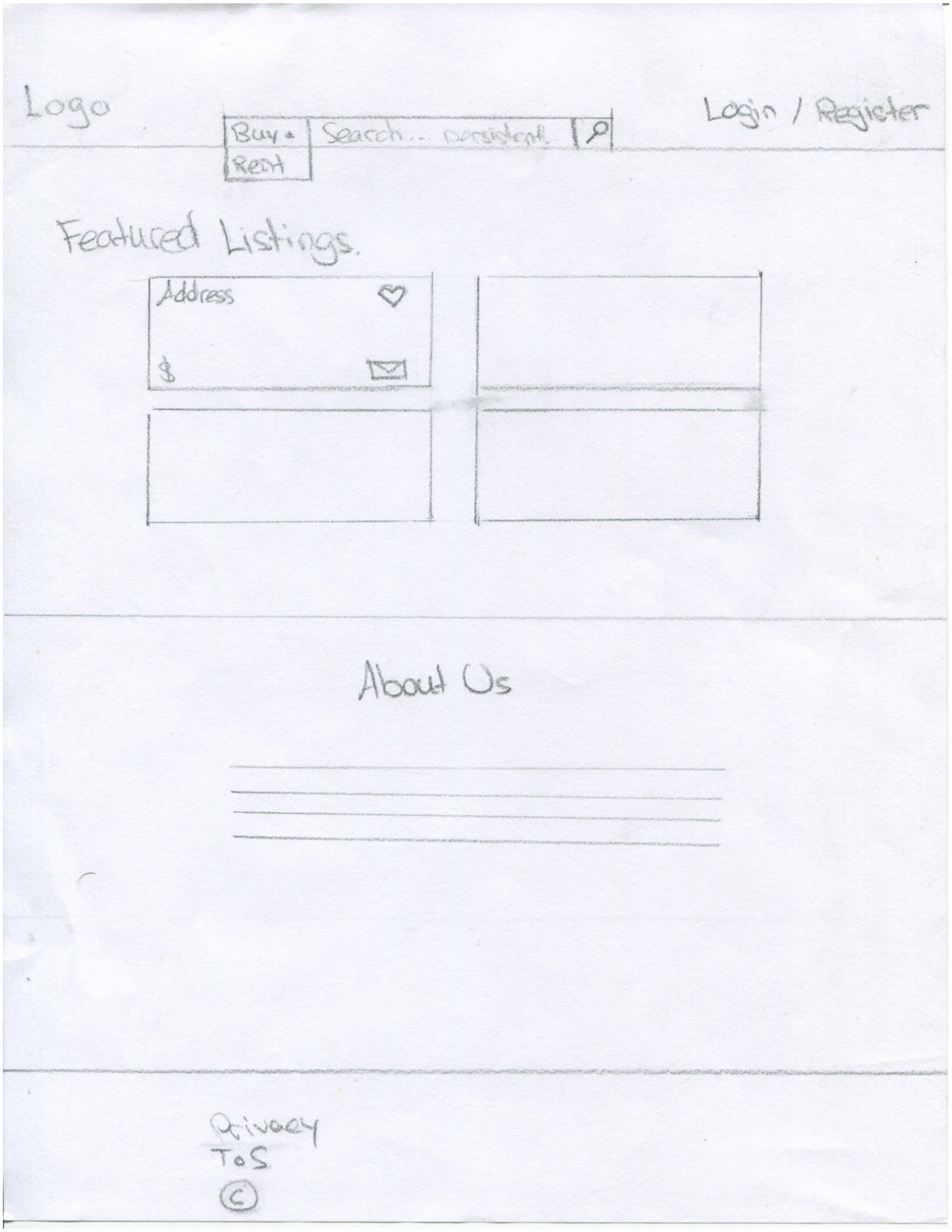
2.3 - Account Dashboard Page

Dawn’s screen now shows her dashboard where she can see her favorite listings and messages to sellers.



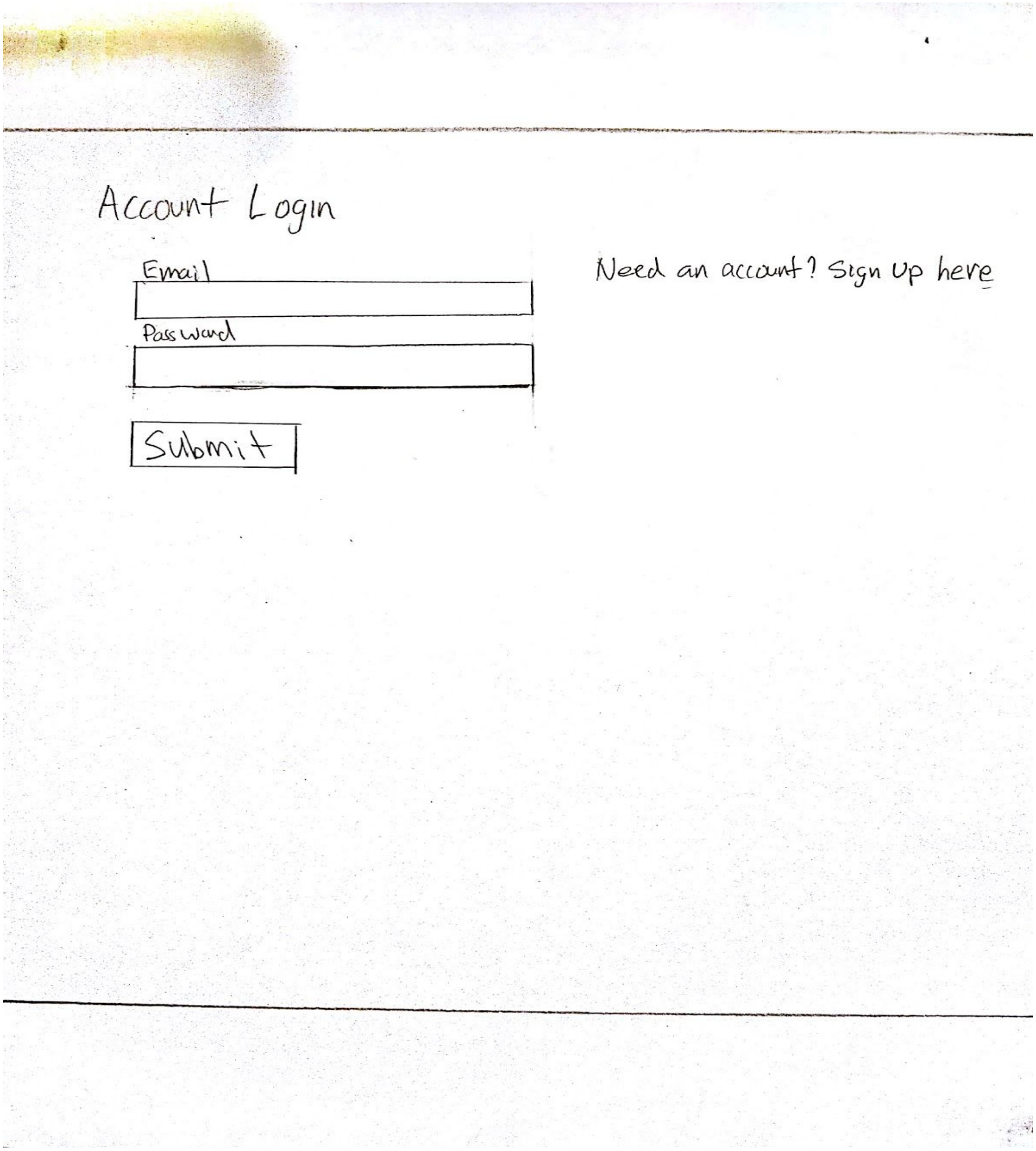
3.1 - Registered User - Agent

Gary is an agent and wants to post a new listing of a home he wants to sell. He goes to Gator Realtors web page and clicks on the login button.



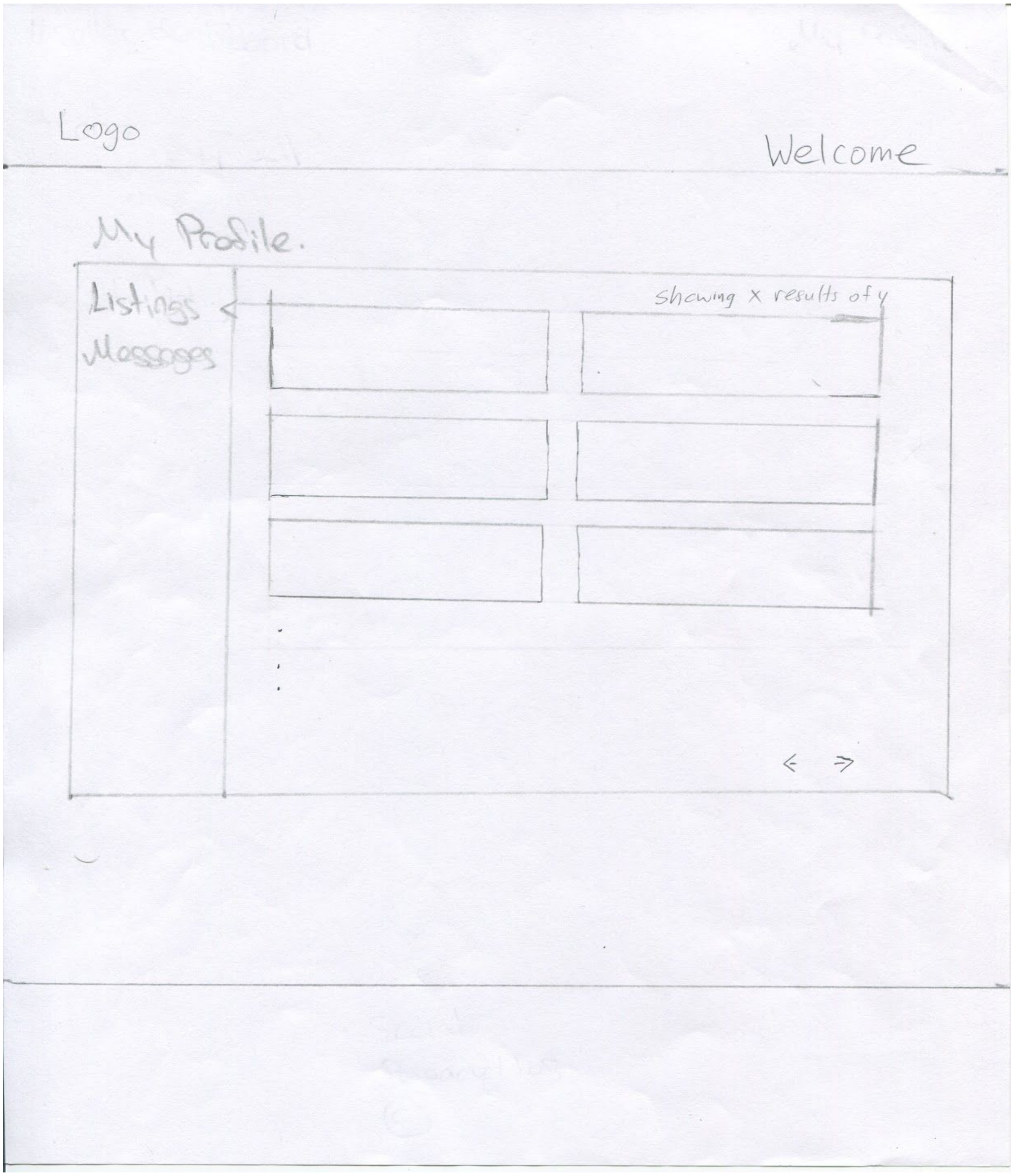
3.2 - Account Login Page

Gary is then redirected to the login page where he is able to input his credentials.



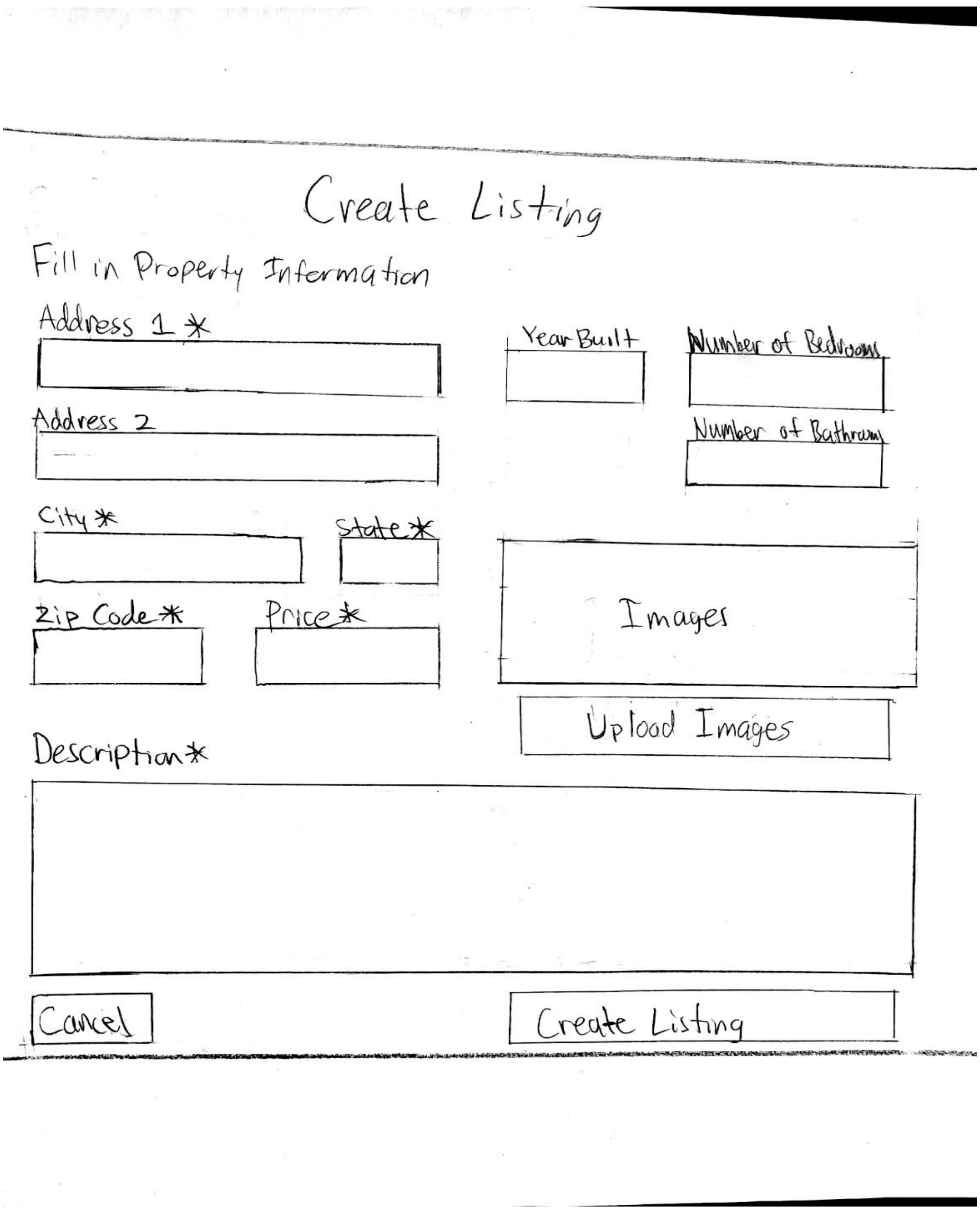
3.3 - Dashboard Page

Once Gary logged in he is directed to his dashboard where he is able to look at his current listings and messages. He clicks on listings and add new listing.



3.4 - Create Listings Page

Gary is then directed to the create listing page where he can input information and images about the home he wants to post.



4. High Level System Architecture, Database Organization

1. High Level Architecture:

The architecture of our web application will be using GCE (Google Compute Engine) for a server. We will be using a mySQL database connected to this server. We will be using node.js to manage the application's server along with express.js, a node.js web framework. These tools are all we need to build our back end. We will be organizing our application in MVC system architecture format. This will allow for modularity within our app which prefaces good scalability for future upgrades.

For our front end we will be using the handlebars.js in conjunction with bootstrap. We chose handlebars it is easy to work with, useful for templating, and most importantly works with bootstrap and handlebars.js. Bootstrap is included because it is extremely powerful and makes mobile responsive page designs easy.

For APIs will be using the required Google ones: analytics and maps. Analytics will be used to track user activity and maps will be used for geolocation services.

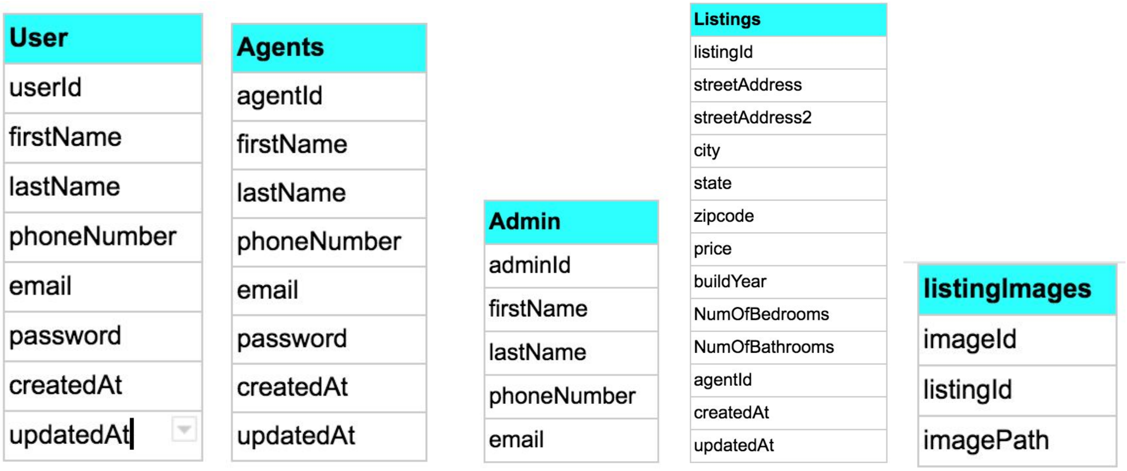
Project Overview:

1. **Technology Stack**
    a. *Node.js* - Application Server
    b. *Express.js* - Node.js Web Framework
    c. *MySQL* - Database Storage
    d. *Handlebars.js* - HTML Templating Engine
    e. *Bootstrap* - UI Component and Layout Library
2. **External APIs, Frameworks and Libraries**
    a. Google Maps API
    b. Google Analytics API
    c. *Sequelize.js* - Database mapper
    d. *Passport.js* - Express-compatible authentication middleware
3. **Deployment** - Google Compute Engine
    a. Great, reliable, and cheap
4. **Supported Browsers**
    a. Chrome: Version 60, 61
    b. Mozilla Firefox: Version 55, 56
    c. Safari: Version 10, 11
5. **Version Control**
    a. GitHub
6. **Additional Tools**
    a. WebStorm
    b. Workbench
    c. Sublime Text Editor

2. DB Organization:

Our main database schema is organized in five (possibly a sixth for messages) tables. These tables will contain data for Users, Agents, Listings, Admins, Images, and Messages.

Database Layout:



User
userId
firstName
lastName
phoneNumber
email
password
createdAt
updatedAt

Agents
agentId
firstName
lastName
phoneNumber
email
password
createdAt
updatedAt

Admin
adminId
firstName
lastName
phoneNumber
email

Listings
listingId
streetAddress
streetAddress2
city
state
zipcode
price
buildYear
NumOfBedrooms
NumOfBathrooms
agentId
createdAt
updatedAt

listingImages
imageId
listingId
imagePath

3. Media Storage:

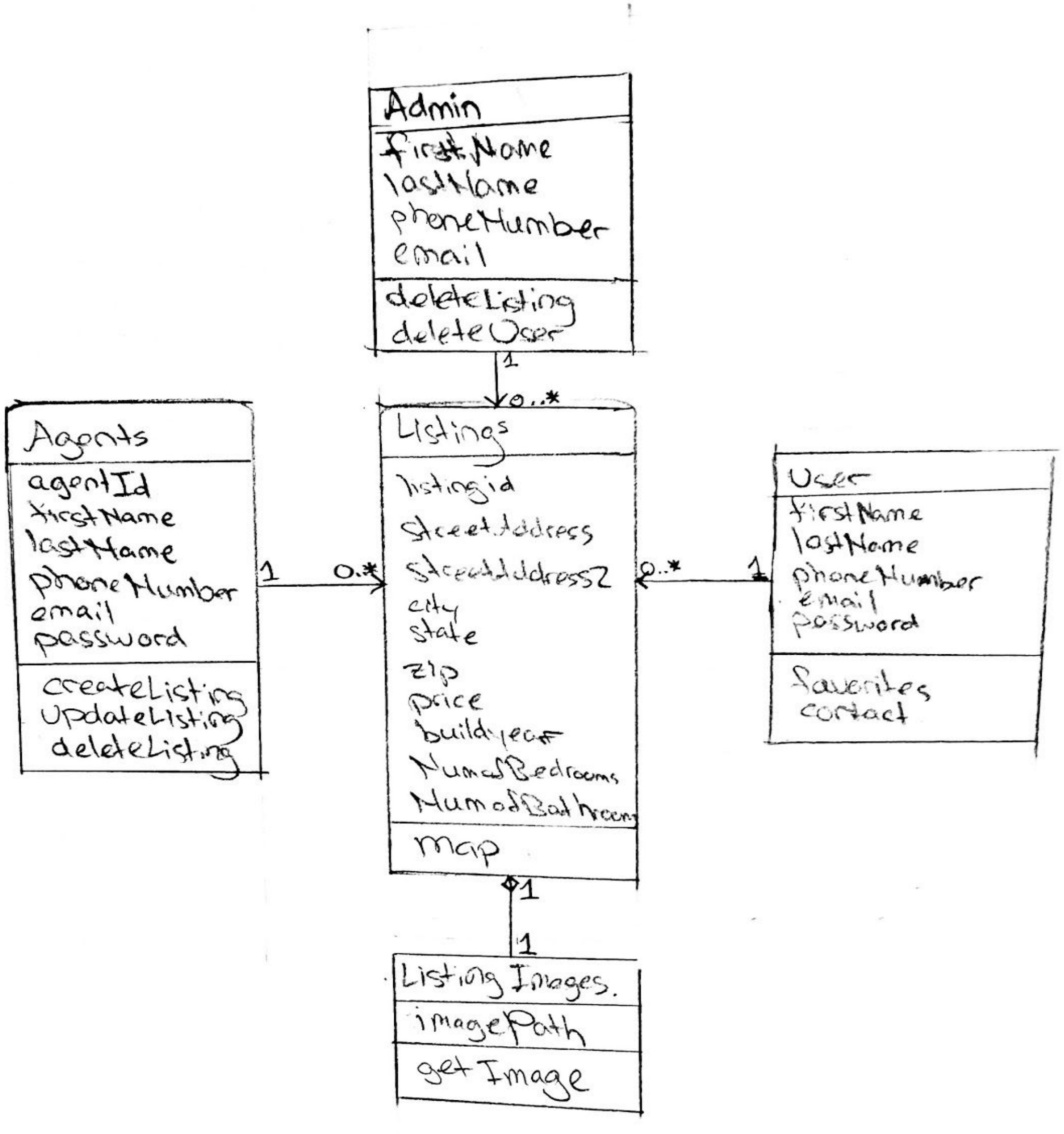
For media storage we decided to put our files in the file systems and store in the database the path to each file. We will be supporting jpg/jpeg and png for image files. To store the files onto the server we are using the Multer library. We are going to use a library to support thumbnails, most likely Sharp or NodeThumbnail.

4. Search/filter architecture and implementation:

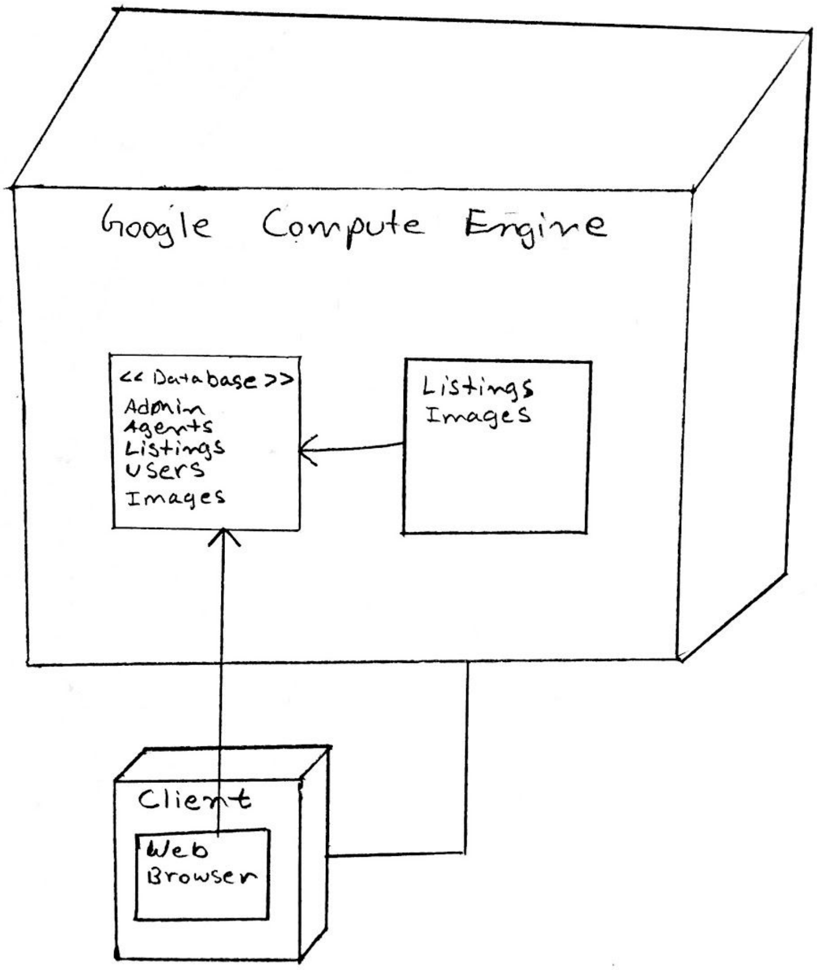
For our search algorithm we'll be using %like for the general, and ASC, DEC for the sorting. Our database will be able to be sorted by price, city, and state. Filtering will be available for data fields such as bedroom count, bathroom count, and type of home such as house or apartment.

5. High Level UML Diagrams

A. UML Class Diagrams



B. UML Deployment Diagrams



6. Key Risks

Type	**Risk**	**Solution**
Skills	Vast differences in skills from member to member in both frontend and backend teams. Most of us have no experience with node or javascript.	We find the best tutorials for us and have our smartest member make recommendations.
Schedule	We all have different class schedules which makes working together as a group difficult.	We described our schedules and found overlaps of free time to have in-person meetups.
Technical	The backend framework being used is express.js and therefore requires the frontend to learn how to use an accompanying framework which is handlebars.js	All of our frontend members are learning while working with handlebars.js
Teamwork	There is a lack of communication and a mismatch of communication between the front and back end.	We hold weekly meetings. It is much easier to communicate in person instead of online.
Legal/content risks	Accidentally using content that isn’t royalty free because we need assets such as images or addresses.	We’re going to go out and take our own photos and use mock addresses.

7. Team

Team Member	**Role**
Alexander Tung	Team Lead
Sukhjit Singh	Backend Lead
Nicholas Szeto	Frontend Lead
Ilya Nemtsov	Backend
Ralph Acosta	Backend
Yangshan Huang	Frontend