

Milestone 2

Gator Realtor

Team 11 - Local

Alexander Tung : alextung94@gmail.com

Sukhjot Singh

Ilya Nemtsov

Nicholas Szeto

Ralph Acosta

Yangshan Huang

Date	Version	Description
10/27/2017	1.0	First draft
11/15/2017	1.1	Revised draft

1. Data Definition

1. **Users:** A user can be Registered or Guest.
 - a. **Guest User:** any user who can browse and search the site for content (homes). This type of user can reach out to any seller.
 - b. **Registered User - Buyer:** any user who has created an account. This type of user has all permissions a guest user has and can contact agents.
 - 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.
 - c. **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.
 - d. **Administrator (Admin):** is a special User who have some privileges that allow him or her to remove illegal content posted by agent; and ban any user for not following the site rules (Terms and Conditions).
2. **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. Seller ID
 - vi. Date posted
 - vii. Other

2. Functional Requirements

Priority 1

- Guest users shall be able to register and login
- Terms of service shall be present during registration
- All users shall be able to browse for listings
- All users shall be able to search listings
- All users shall be able to filter listings
- All users shall be able to see contact information
- Buyers shall be able to contact agents
- Agents shall be able to create listings
- Agents shall be able to edit or remove listings
- All data listings shall display contact information
- All data listings shall require at least one image
- All data listings shall show location on map
- Administrators shall be able to suspend accounts
- Administrators shall be able to delete listings

Priority 2

- All users shall be able to sort listings
- Buyers shall be able to favorite listings
- Agents shall be able to upload multiple photos at once when creating listing
- Administrators shall be able to delete users

Priority 3

- All users shall be able to report problems with any listing
- Buyers shall be able to save search query
- Buyers shall be able to share listings
- All listings shall display other, comparable listings

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.

The mockup is a hand-drawn sketch of a real estate website interface. It is divided into four horizontal sections by thin lines.

- Header Section:** On the left is the text "Logo". In the center is a navigation menu with two buttons: "Buy" and "Rent". To the right of these buttons is a search bar containing the text "Search... persistent" and a magnifying glass icon. On the far right is the text "Login / Register".
- Featured Listings Section:** This section is titled "Featured Listings." on the left. It contains a table with two columns and two rows. The first row has a cell on the left with the text "Address" and a heart icon, and an empty cell on the right. The second row has a cell on the left with a dollar sign "\$" and an envelope icon, and another empty cell on the right.
- About Us Section:** This section is titled "About Us" in the center. Below the title are four horizontal lines, suggesting a text area or a list of items.
- Footer Section:** This section is at the bottom and contains three items: "Privacy", "TOS", and a copyright symbol "©".

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.

Filters

Option

Option

Range

~

Apply

persist search

Search Results for [keyword]

Showing x of y

Address

\$

<

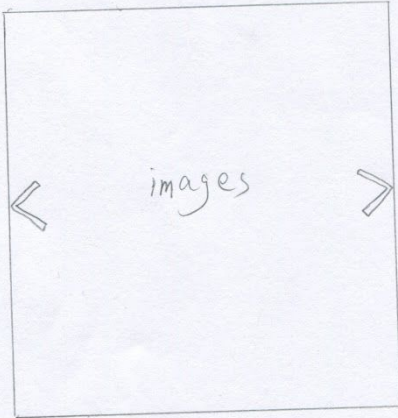
>

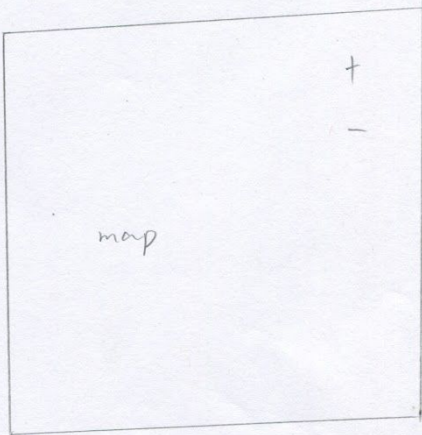
1.3 Listing page

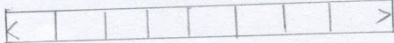
Joey is now presented with the listing's detailed page and he can view different images and see the description of the home. He is interested in seeing this home in person so he clicks on the contact agent button.

Logo

Listing #123456







Address: _____ Price _____

_____ beds _____ baths _____ sq ft


Description: _____

Features:

_____ Types _____ laundry _____ Heating

_____ cooling _____ Pets _____ Parking

Contact Agent:



(415) - 333-4444

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.

Account Login

Email

Password

Need an account? Sign up here

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.

The image shows a hand-drawn sketch of a registration form on a piece of paper. The form is titled "Register" in a handwritten font. Below the title, there are several input fields, each with a label to its left: "First Name", "Last Name", "Email", "Phone Number", "Password", and "Confirm Password". Each label is followed by a rectangular box representing the input field. Below these fields, there is a section labeled "Select One" with two radio button options: "Agent" and "Buyer". At the bottom of the form, there is a rectangular button labeled "Submit". The entire form is enclosed in a simple rectangular border.

Register

First Name

Last Name

Email

Phone Number

Password

Confirm Password

Select One

☐ Agent

☐ Buyer

Submit

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.

Contact Seller

Seller's email address		
Subject :		
Message		
<table border="1"><tr><td>Cancel</td><td>Send</td></tr></table>	Cancel	Send
Cancel	Send	

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.

The wireframe depicts a real estate website layout. At the top left is a 'Logo' placeholder. To its right is a navigation menu with 'Buy' and 'Rent' options, followed by a search bar containing the text 'Search... persistent' and a magnifying glass icon. Further right is a 'Login / Register' link. Below the navigation bar is a section titled 'Featured Listings.' which contains a 2x2 grid of listing cards. Each card has a header with 'Address' and a heart icon, and a body with a dollar sign '\$' and an envelope icon. Below the listings is a section titled 'About Us' with four horizontal lines for text. At the bottom left are links for 'Privacy' and 'ToS', and a copyright symbol '©'.

Logo

Buy • Search... persistent | 🔍

Rent

Login / Register

Featured Listings.

Address	♥
\$	✉

About Us

Privacy
ToS
©

2.2 - Account Login Page

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

Account Login

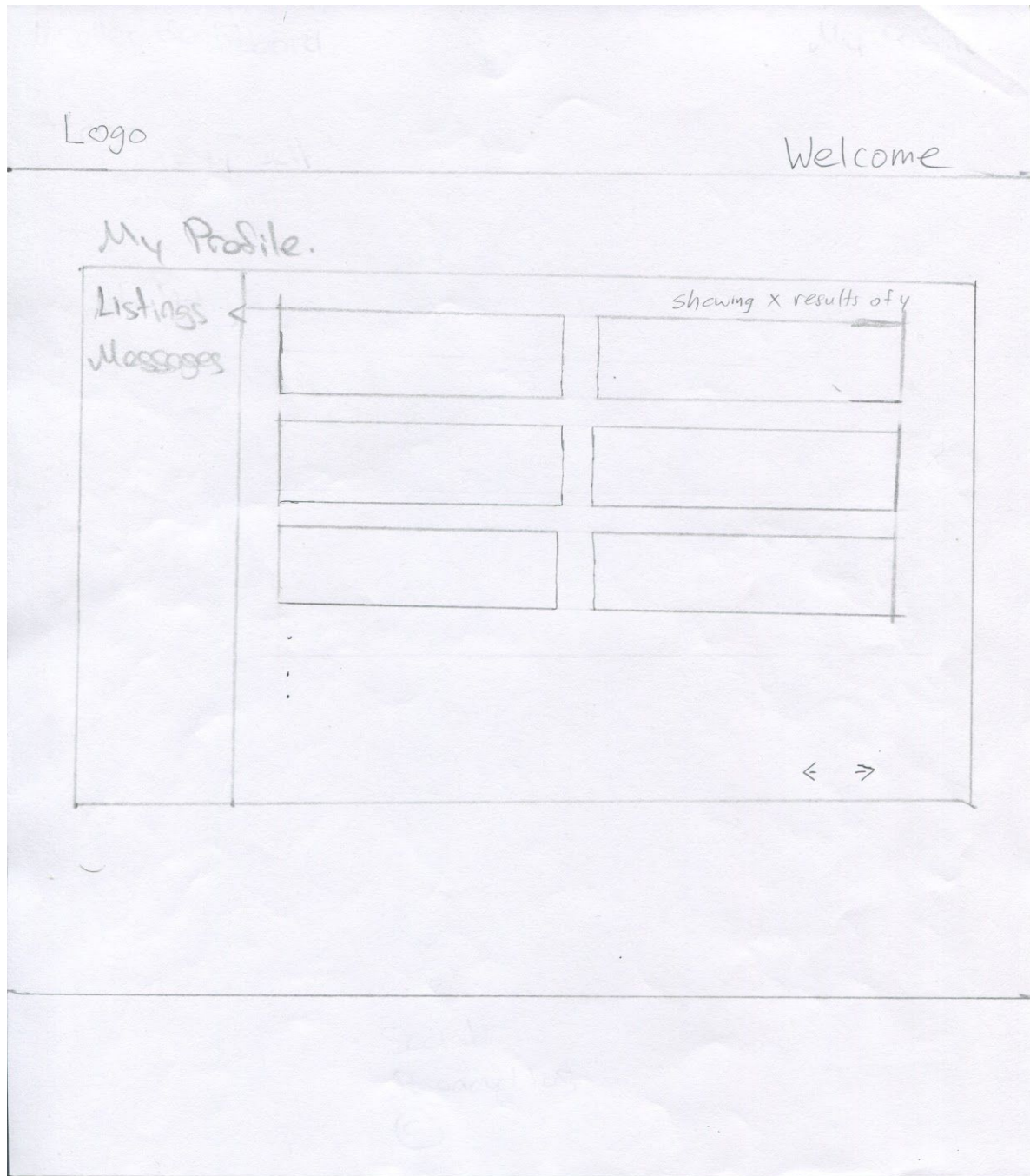
Email

Password

Need an account? [Sign up here](#)

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.

The wireframe depicts a web page layout with the following elements:

- Header:** A horizontal bar containing a "Logo" on the left, a navigation menu with "Buy" and "Rent" options, a search bar with the placeholder text "Search... persistent" and a magnifying glass icon, and a "Login / Register" link on the right.
- Featured Listings:** A section titled "Featured Listings." containing a grid of listing cards. Each card has a header with "Address" and a heart icon, and a body with a dollar sign "\$" and an envelope icon.
- About Us:** A section titled "About Us" with four horizontal lines for text.
- Footer:** A section containing links for "Privacy" and "TOS", and a copyright symbol "©".

3.2 - Account Login Page

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

Account Login

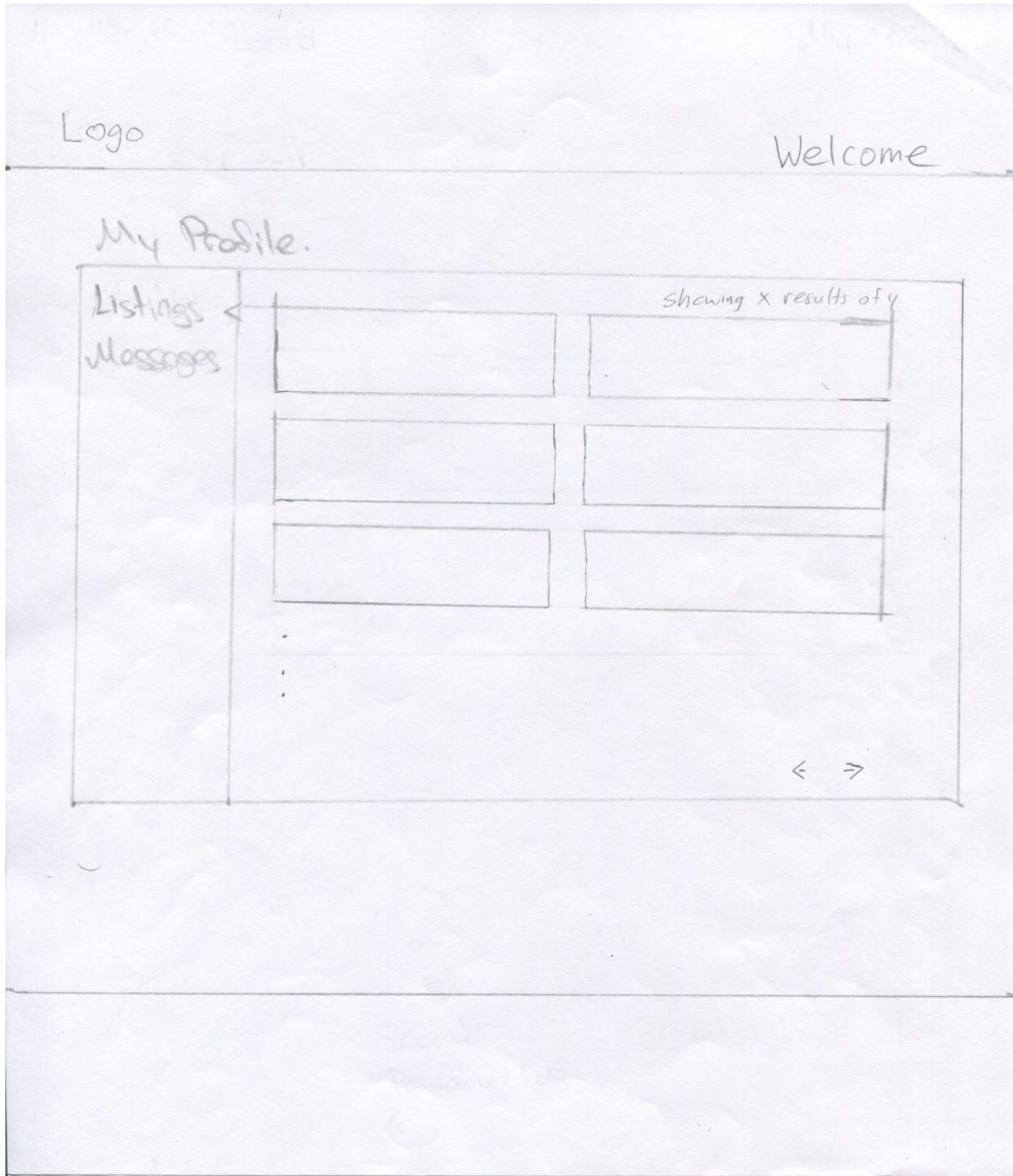
Email

Password

Need an account? [Sign up here](#)

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.

Create Listing

Fill in Property Information

Address 1 *	Year Built	Number of Bedrooms
<input type="text"/>	<input type="text"/>	<input type="text"/>
Address 2		Number of Bathrooms
<input type="text"/>		<input type="text"/>
City *	State *	<div>Images</div> <div>Upload Images</div>
<input type="text"/>	<input type="text"/>	
Zip Code *	Price *	
<input type="text"/>	<input type="text"/>	
Description *		
<input type="text"/>		
<input type="button" value="Cancel"/>	<input type="button" value="Create Listing"/>	

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	Agents	Admin	Listings	listingImages
userId	agentId	adminId	listingId	imageId
firstName	firstName	firstName	streetAddress	listingId
lastName	lastName	lastName	streetAddress2	imagePath
phoneNumber	phoneNumber	phoneNumber	city	
email	email	email	state	
password	password		zipCode	
createdAt	createdAt		price	
updatedAt	updatedAt		buildYear	
			NumOfBedrooms	
			NumOfBathrooms	
			agentId	
			createdAt	
			updatedAt	

3. Media Storage:

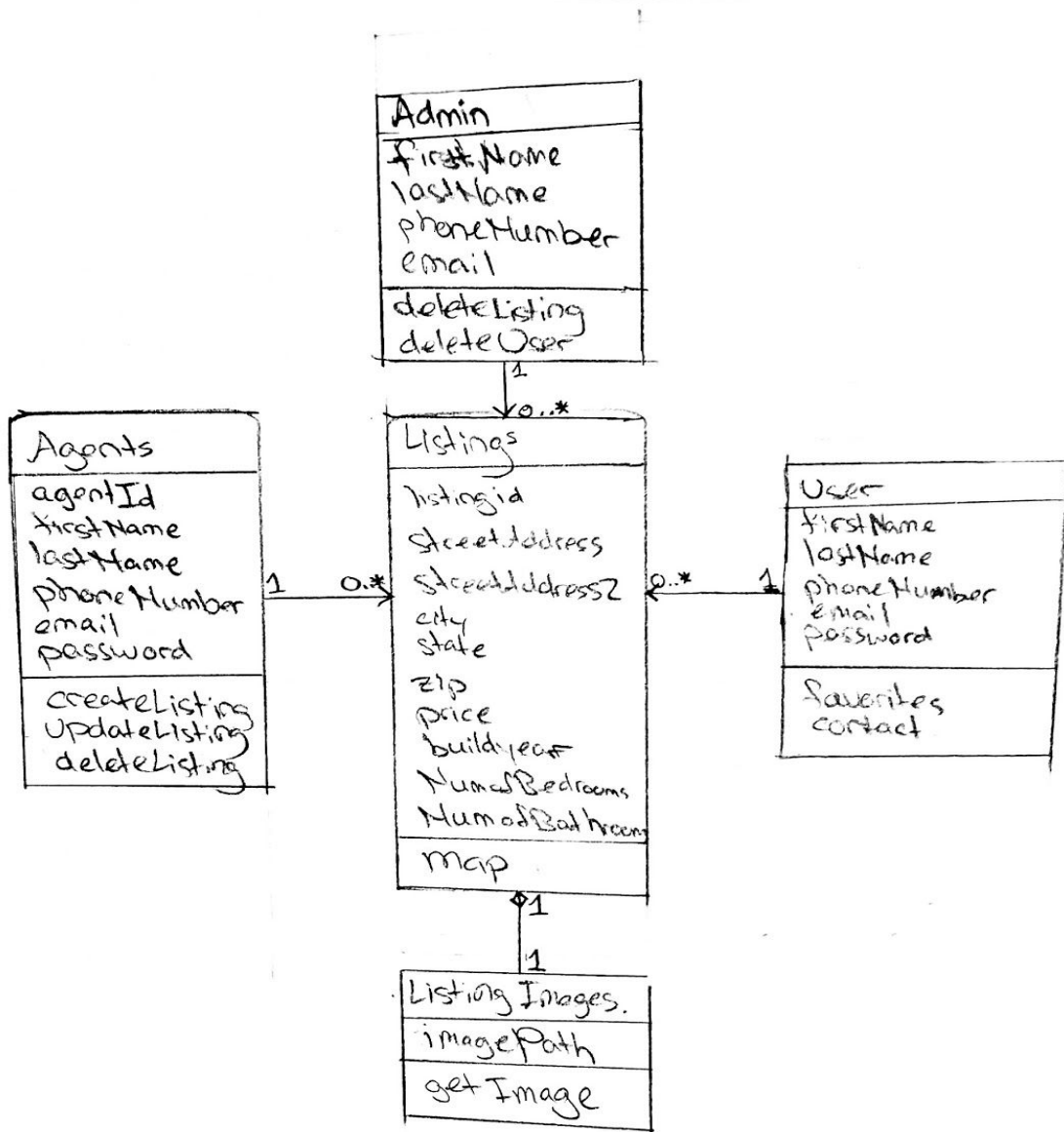
For media storage we decided to put our files in the file systems and store in the database the path to each file. We will be supporting jpg/jpeg and png for image files. To store the files onto the server we are using the Multer library. We are going to use a library to support thumbnails, most likely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

4. Search/filter architecture and implementation:

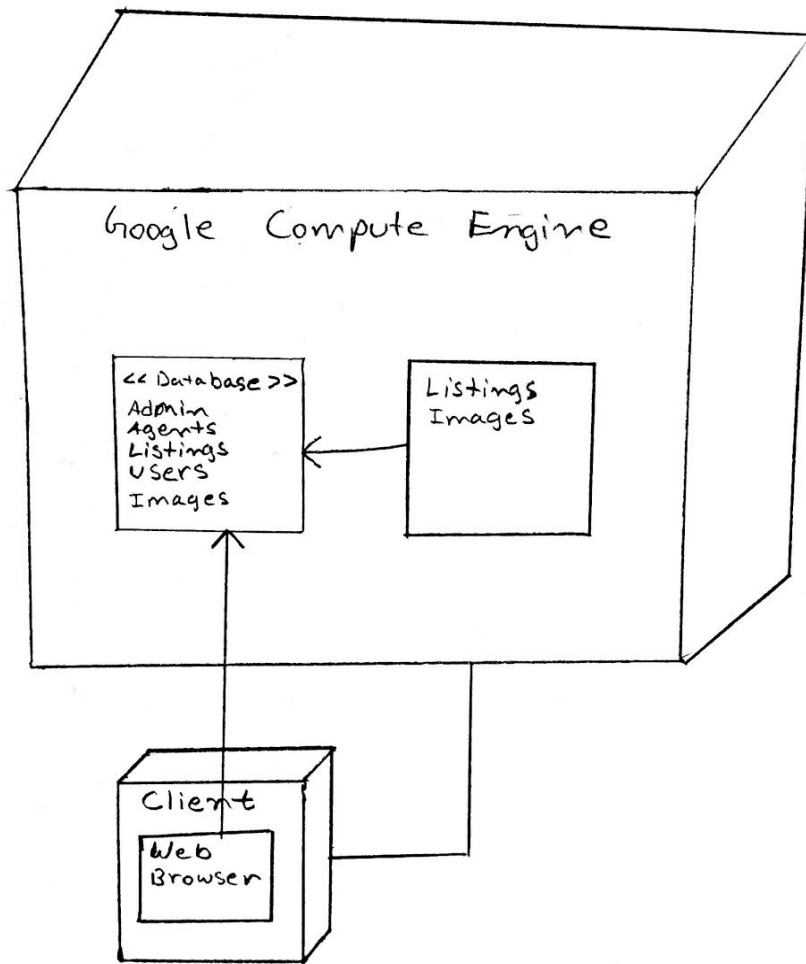
For our search algorithm we'll be using %like for the general, and ASC, DEC for the sorting. Our database will be able to be sorted by price, city, and state. Filtering will be available for data fields such as bedroom count, bathroom count, and type of home such as house or apartment.

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