# Project Landmark

## Milestone 2

Team 03 (Local)

**Jesse Gabriel (jgabrie2@mail.sfsu.edu),
Avneesh Setia, Huiliang Huang, Yingjing Chen, Jenny Ngo,
Matthew Berkman, David Lau
10/27/2017**

| | | |
|---|---|---|
| History Table | 10/26/17 | Rough Draft #1 Submitted for Review |
| | 10/27/17 | Final Draft Submitted addressing the following comments:<br>• Functional Requirements V2 has been rearranged to group by priority rather than user.<br>• Home Page has been changed to give an explicit login button for Agent.<br>• Listing Results Page now has a title for the page, shows the number of listings that appeared, and has addition information to display for each listing such as address and number of bedrooms.<br>• Details Page now has the search navbar under the main navbar.<br>• Login/Register Page now has a "Forgot Password" link and has a place to try a captcha.<br>• Agent's My Listing Page has been renamed to Agent's Dashboard.<br>• Buyer's and Seller's Listing Page has been noted as a P2 in the title and renamed to a Buyer's and Seller's Dashboard.<br>• Added a UML Deployment Diagram |
| | 10/28/17 | FROZEN |

# Table of Contents

# 1. Data Definition V2

- **Unregistered User**: user/users who either do not have an account or have not logged in on the site.
  - Can view **listing** and use the **search bar**.
  - They cannot use the **messaging system** and do not have access to the **dashboard**.
    - To access these features, an unregistered user must sign-in or register for an account.
- **Registered User**: is a user who has either a **buyer account**, **seller account**, or an **agent account**.
  - For all registered users, they will need to provide:
    - First Name and Last Name
    - Email
    - Password (max: 30 characters)
  - For an agent, along with the previous information they will also need to:
    - Specify if they are from one of these agencies: SFStateHomes, SJStateRealtor, or CSURealEstate.
- **Admin**: someone who can access all data and content.
  - Admin can modify the database through the MySQL Workbench.
- **Agent Account**: user/users that can view all **listings** and post it on the website.
  - Can modify their existing **listings**, use the **messaging system** to communicate with all available **registered user's** information, has access to their own personal **dashboard**.
  - Needs to login/register.
- **Buyer account**: user/users that can view all the available **listings** and agent's information.
  - Will can contact agents through a **messaging system** and has their own personal **dashboard**.
  - Needs to login/register.
- **Seller account**: user/users that can view all the available **listings** and agent's information.
  - Will can contact agents through a **messaging system** and has their own personal **dashboard**.
  - Needs to login/register.
- **Comparisons**: a list showing **listings** near location of the real estate the user is currently viewing.
- **Message System**: a form of communication. All registered users will be able to:
  - Send and receive messages. Message consists of:
    - Intended recipient's first and last name.
    - Message content (Max: 120 characters).
    - Date Sent
- **Dashboard**: a registered user interface that allows access to features such as:
  - Inbox
  - View registered **listings** (**Seller account**)
  - Add **Listings** (**Agent account**)
  - Log out
- **Filter**: Allows an **unregistered user** to narrow searches down by:
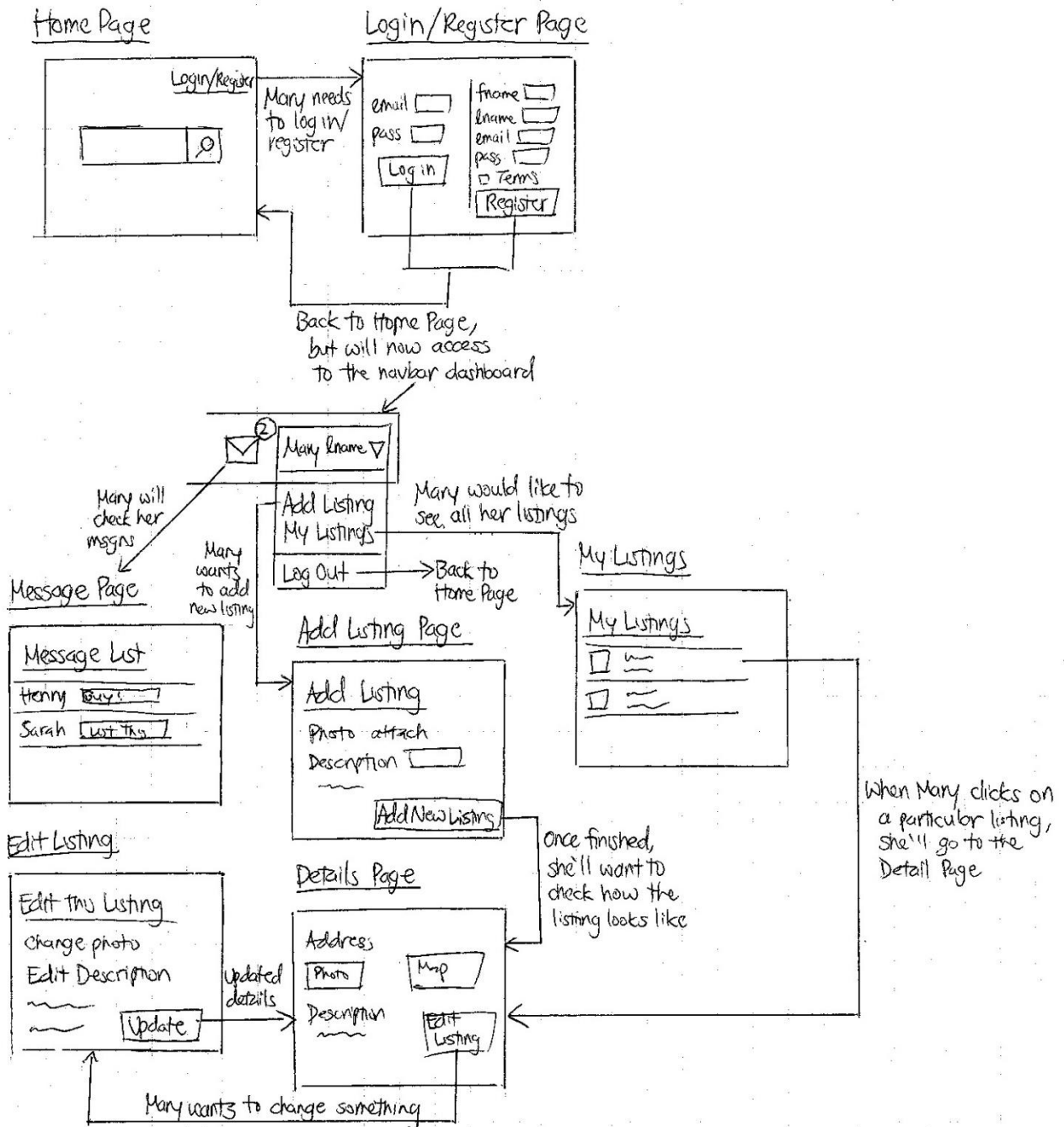  - Price range

- o Number of specific types of rooms
  - Bathrooms
  - Kitchens
  - Bedrooms
- o Number of floors
- o Real Estate Type
  - Condo
  - Mansion
  - House
  - Apartment
- **Listings**: a real estate of which only the **agent account** can post.
  - o Price
  - o Description
  - o Photo(s) (min: 1, max: 3)
  - o Real Estate Type (Condo, Apartment, Mansion, House)
  - o Keywords
    - Parking availability
    - Heating and cooling
    - Utilities
    - Number of rooms (specifying bathrooms, bedrooms, kitchens, living rooms)
- **Search Bar**: an application that allows users to find specific content on site
  - o An **unregistered user** may search by city, zip code, and address
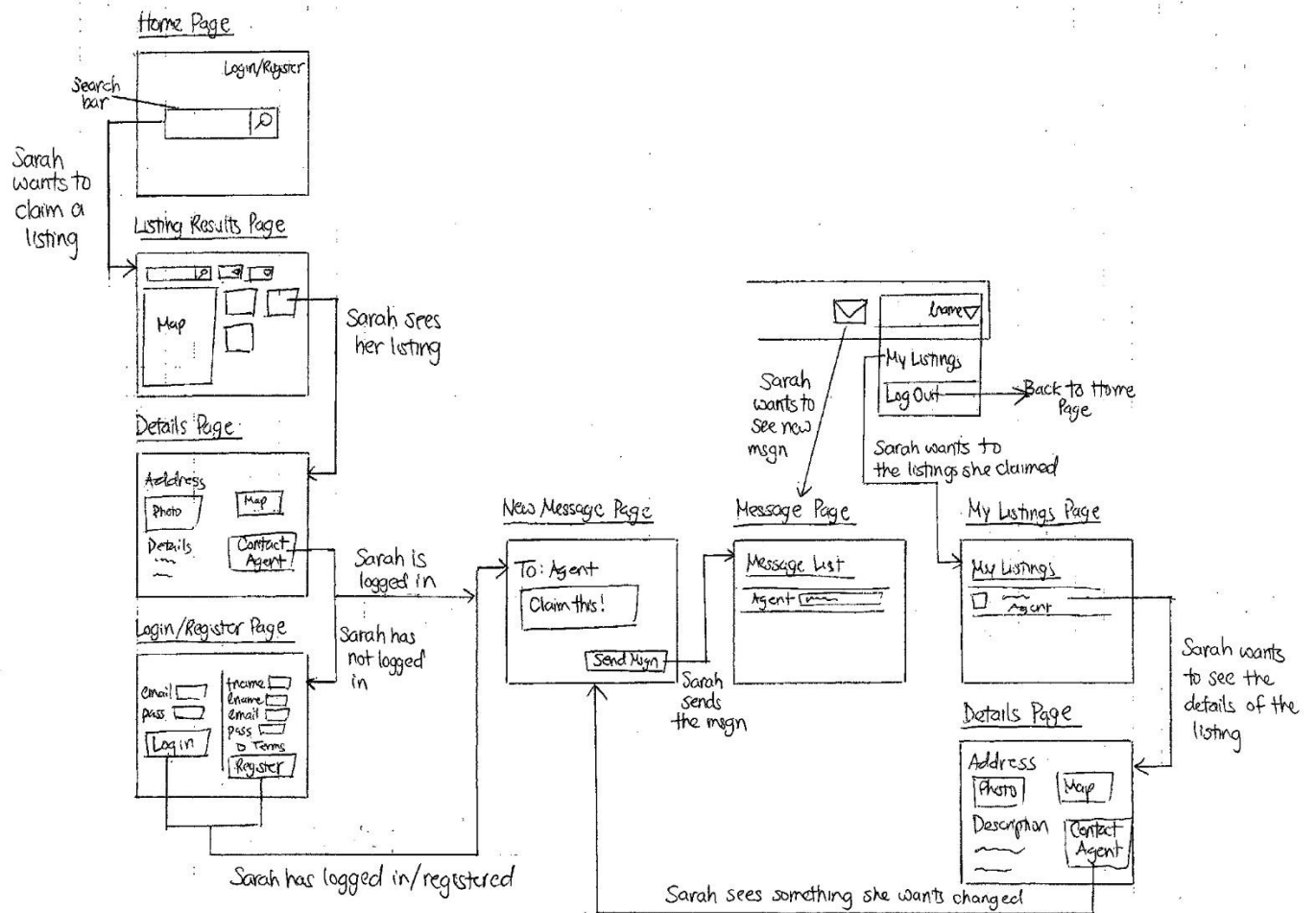
## 2. Functional Requirements V2

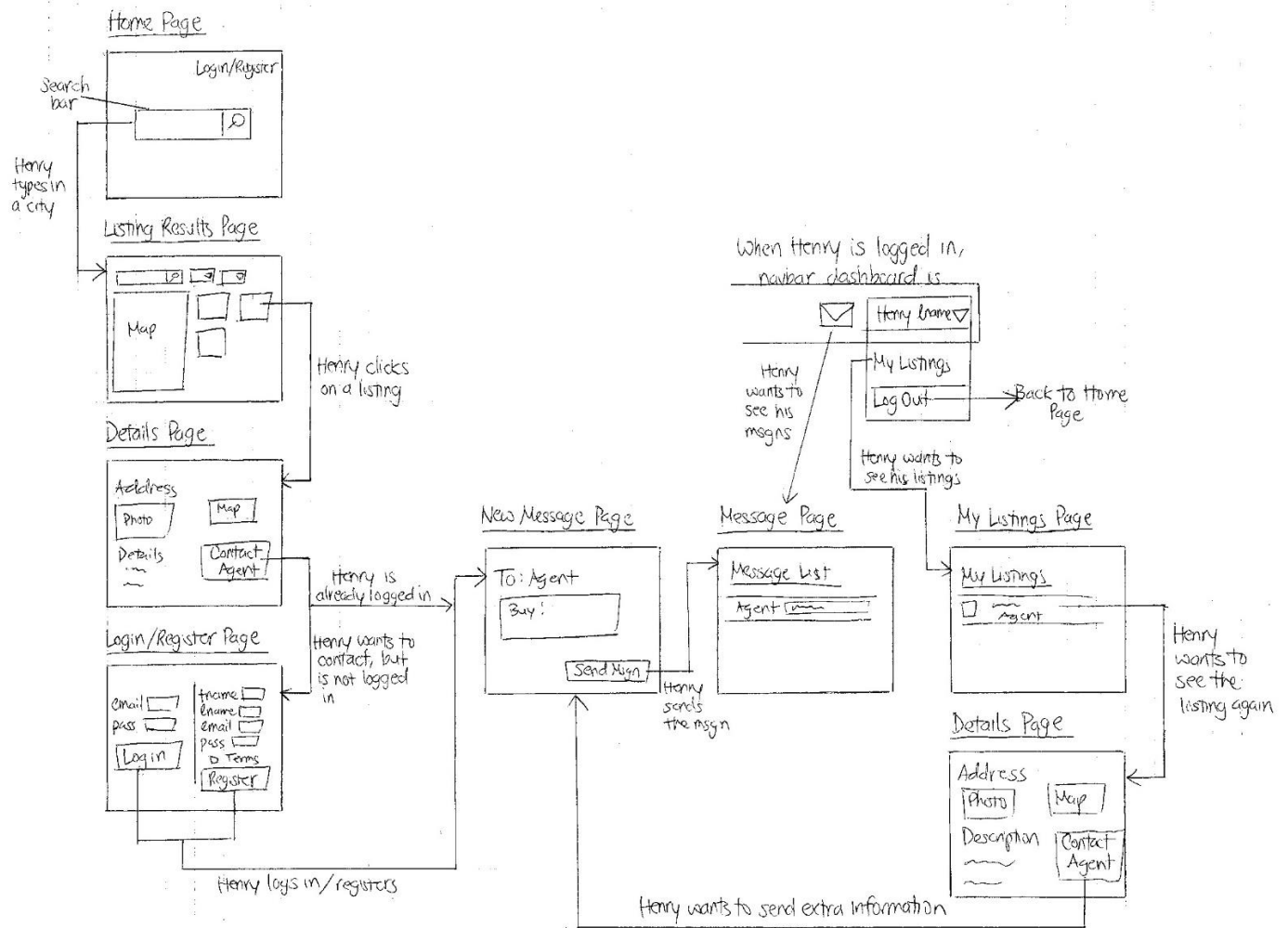| Priority Rank | Requirements | User |
|---|---|---|
| Priority 1 | 1. An **unregistered user** shall be able to search the website for **listings** via the **Search Bar** on the main landing page.<br>   1.1. An **unregistered user** shall be able to search via cities. | Unregistered User |
| | 2. An **unregistered user** shall be able to able to filter **listings**.<br>   2.1. **Filter** shall be able to filter by keywords such as price, bedrooms, bathrooms, and real estate type. | Unregistered User |
| | 3. An **unregistered user** shall be able to view multiple thumbnail images with the **listings'** price if search results appear.<br>   3.1. In the case the user enters a search query that gives no results, and then the user shall see top-priced listings instead. | Unregistered User |
| | 4. [New] An **unregistered user** shall be able to acquire a **buyer account**, **seller account**, or **agent account** through a registration process.<br>   4.1. All **unregistered users** shall enter their first and last name, email, password, and agree to a Terms of Service.<br>   4.2. An **unregistered user** shall be able to specify if they are a buyer/seller or an agent. | Unregistered User |
| | 5. A **buyer account** shall be able to contact agents via a **message system**.<br>6. A **seller account** shall be able to contact agents via a **message system**. | Buyer Account<br>Seller Account |
| | 7. An **agent account** shall be able to create and post new **listings** to the website. | Agent Account |
| | 8. An **agent account** shall be able to receive new messages from **seller accounts** and/or **buyer accounts** who have contacted them via the **dashboard**. | Agent Account |
| Priority 2 | 9. A **buyer account** shall be notified of new messages from their agent(s) via a **dashboard**. | Buyer Account |
| | 10. A **seller account** shall be notified of recent updates and changes to the **listing** their agent has listed via a **dashboard**. | Seller Account |
| | 11. An **agent account** shall be able to send messages back to **seller accounts** and/or **buyer accounts**. | Agent Account |
| Priority 3 | 12. An **unregistered user** shall be able to see **comparisons**. | Unregistered User |
| | 13. [New] An **unregistered user** shall need to verify they are a human via captcha during registration. | Unregistered User |
| | 14. An **unregistered user** shall be able to return to their search results after viewing a listing. | Unregistered User |
| | 15. [New] An **agent account** shall be able to edit existing **listings** they have created. | Agent Account |

## 3. UI Mockups and Storyboards

Mary's (Agent) Use Case Flowchart

## Sarah's (Seller) Use Case Flowchart

Henry's (Buyer) Use Case Flowchart

Home Page

Listing Results Page

SFSU Software Engineering Project, Fall 2017
For Demonstration Only

Logo | About Contact                    Login/Register    → Main navbar

[Search City... 🔍] [Price ▽] [Bedrooms ▽] [Bathrooms ▽] [Type ▽]    → Search navbar

Showing Results for City   3 Found    → Pulled from what the user typed in the Search Bar

Thumbnail

Price # of bedrooms
Address

Thumbnail

Price # of bedrooms
Address

Thumbnail

Price # of bedrooms
Address

Map of City typed
from Search bar

Details Page

Details Page.

top.

SFSU school project. Fall 2017.

| Logo | Project Land Mark | ✉ ① | user | ☰ |

| SF, CA .... 🔍 | House Type ▼ | Price Range ▼ | 0+ Bedroom ▼ | - - - - - |

pull down
- profile
- setting
- Help
- log out

press down
- 1 room
- 2 rooms
- 3 rooms
- ⋮

xx City, CA. 94xxx

Google map

pic1 | pic2 | pic3 | ---- | >

Details:
- owner: xxx
- Built: xxx
- Bedroom: 2.
- Livingroom: 2

Interesting?
Name: [        ]
phone: [        ]
Email: [        ]
[contact]   Help?

Vistor Mode

Rigister Mode

Interesting?
[contact] Help?

Some House you might interest?

🏠 1    >    🏠 2    >    🏠 3    ....
xxxx         xxx          xxx

Follow [F] [GH]

Copy Right 2017.

Login/Register Page



Login page mockup

(sticky navigation bar) buttons same as home page

login section         Sign up section

→ username         → first name

→ last name

→ password         → email

→ drop down for user type selection

→ login button         → password

→ forgot password
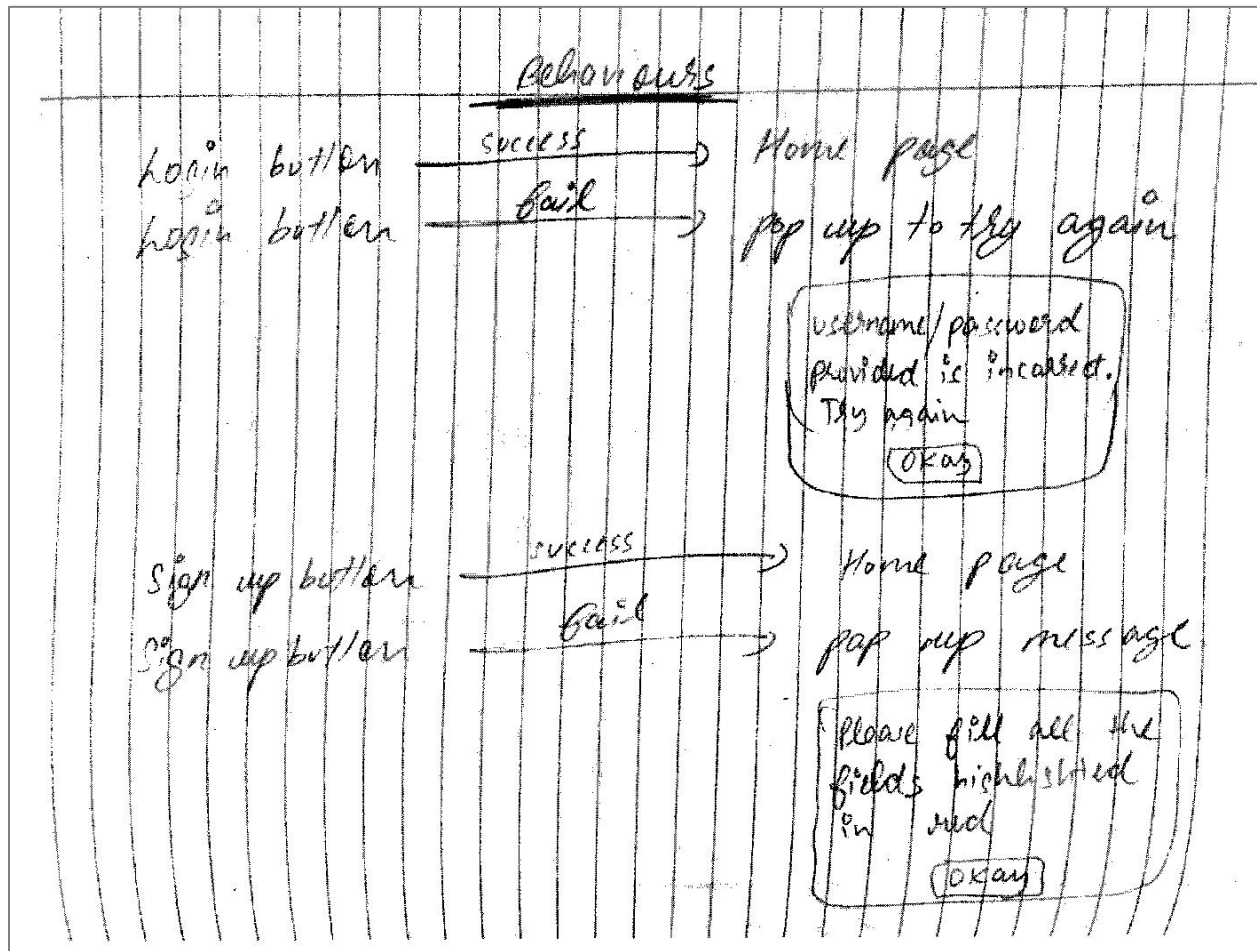
(links to a separate page)

→ captcha    (i'mnota robot)

→ sign up button

static banner with team name / school project warning

Login Flowchart

Agent's Dashboard

SFSU Software Engineering Project, Fall 2017
For Demonstration Only

| Logo | About Contact | | ✉① | fname lname ▽ |

on-click bring down drop dash-board

My Listings   Group by: City Owners

on-click show by City

| Thumb-nail | City, Street Address by Owner's name |
| | Contacts: Contact Name 1, Contact Name 2 |

| Thumb-nail | City, Street Address by Owner's name |
| | Contacts: Contact 1 |

Add Listing
My Listings
Log Out

## Buyer's and Seller's Dashboard (P2)

Both Henry's and Sarah's dashboard will be the same, but the only thing different would be the name that displays in the dashboard.

## 4. High-level system architecture

### Architectural Design Pattern

- Application will use the MVC (Model-View-Controller) Pattern to reduce coupling.
- A query file will be our model. The model will take data from the user or MySQL and manipulate that data to be viewed or stored. The model should validate any incoming/outgoing data constraints. The model will also notify the controller for updates in the database.
- Pug files will be our view for our application's server-side/client-side rendering. Pug is a template engine, and it receives data from the controller and sends it to the browser, filling out dynamic content for the user to view. This is where the client can interact with the server.
- ExpressJS will be our controller that processes a user's request and communicates with the model. The controller will also acquire data directly from model and send it back to the user. The controller will also send user data for model to store.

### MVC Model
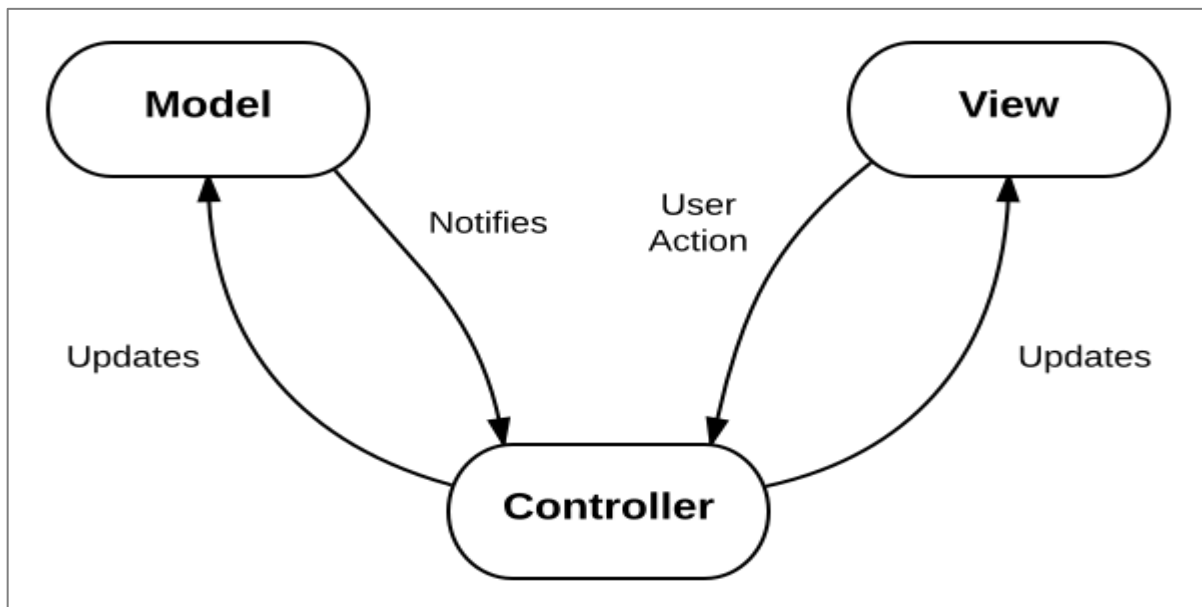
Database Organization:
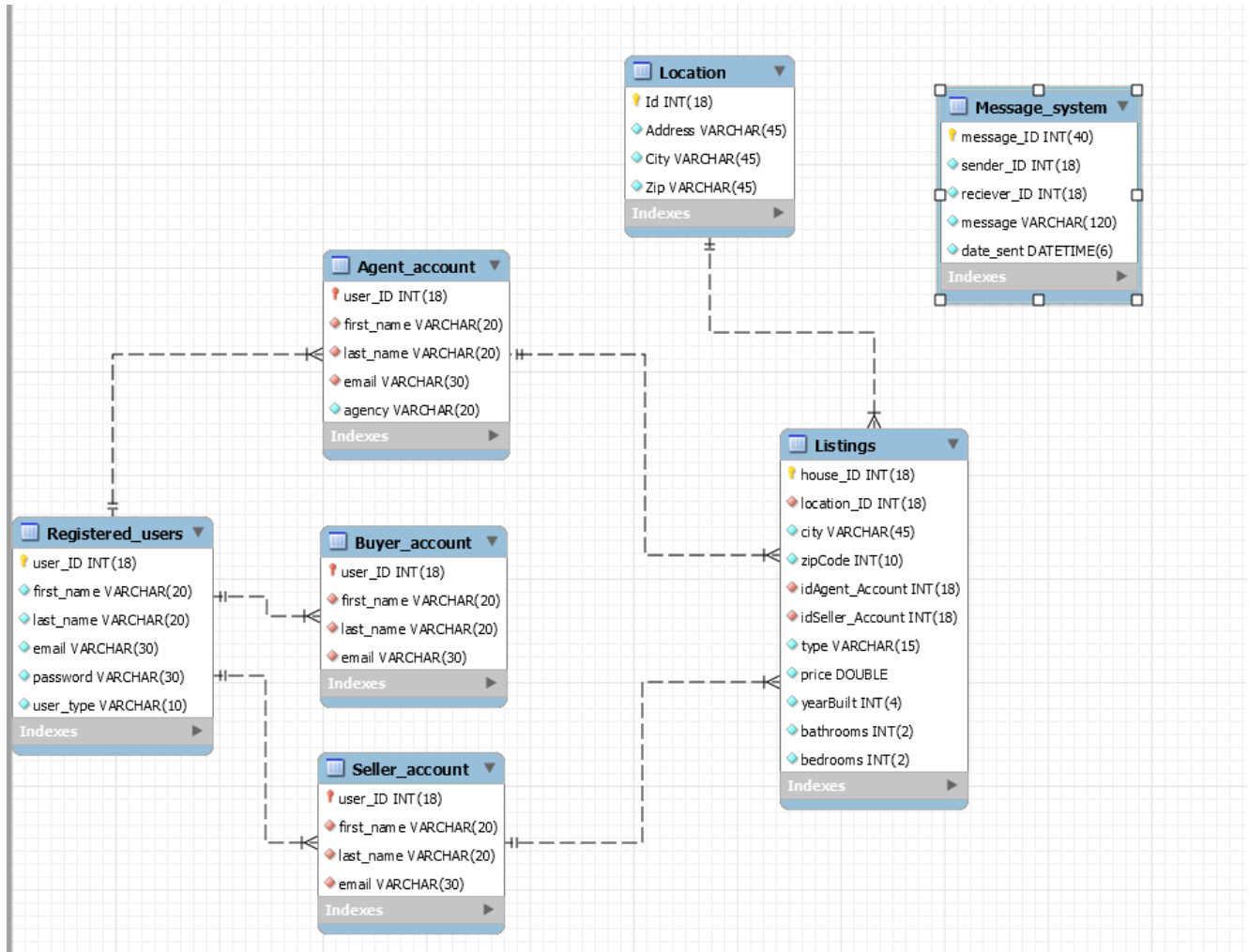


## Media Storage:

We will be using Cloudinary for media storage. Cloudinary is a platform that uses Amazon's Web Service for media storage for backend image hosting. Their API is specifically focused on efficient content delivery for websites.
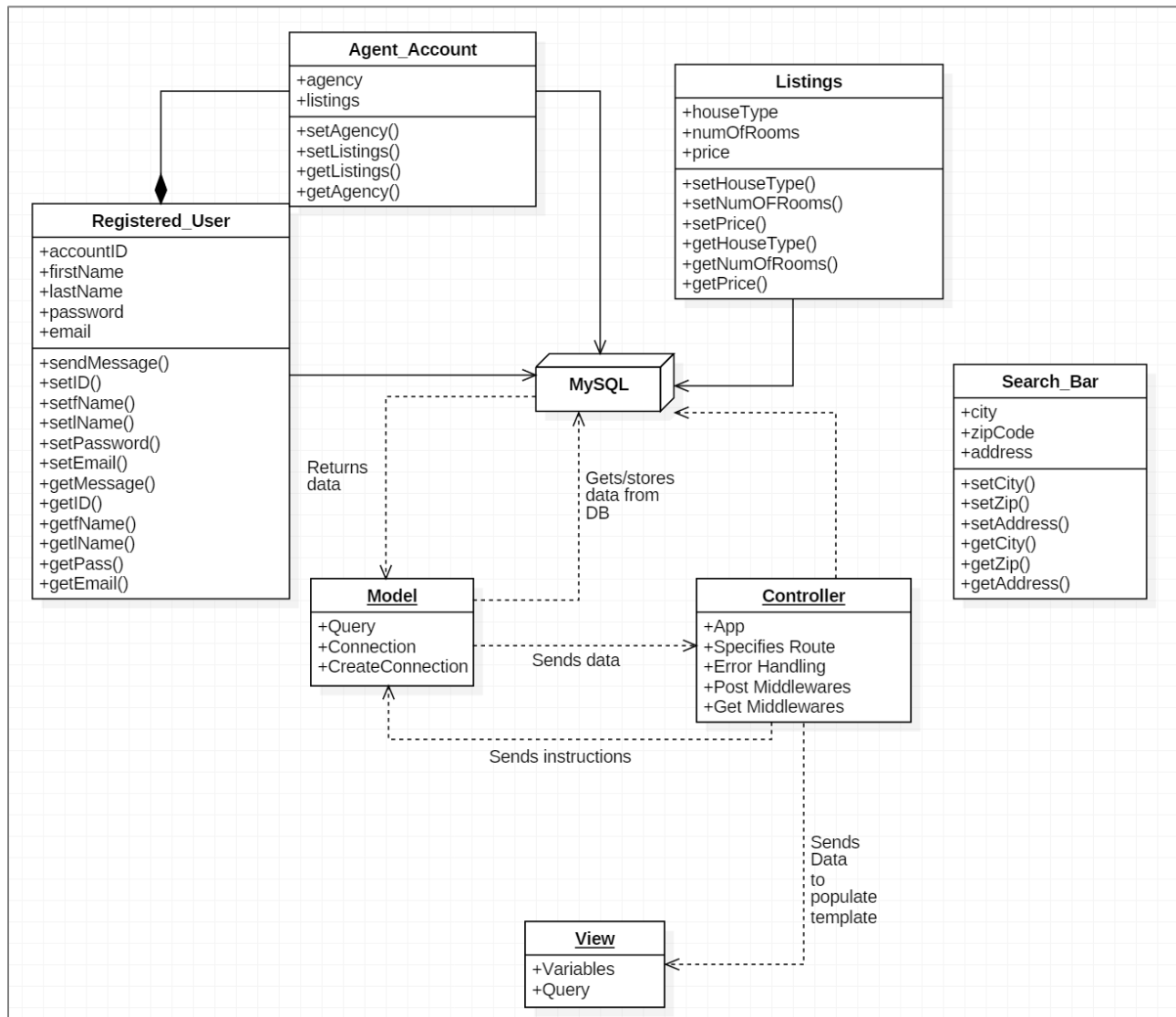
## Search/filter architecture and implementation:

The search algorithm will use Sequel's "LIKE" command to find any related characters from the query string that matches with the attributes city, address, and zip code.
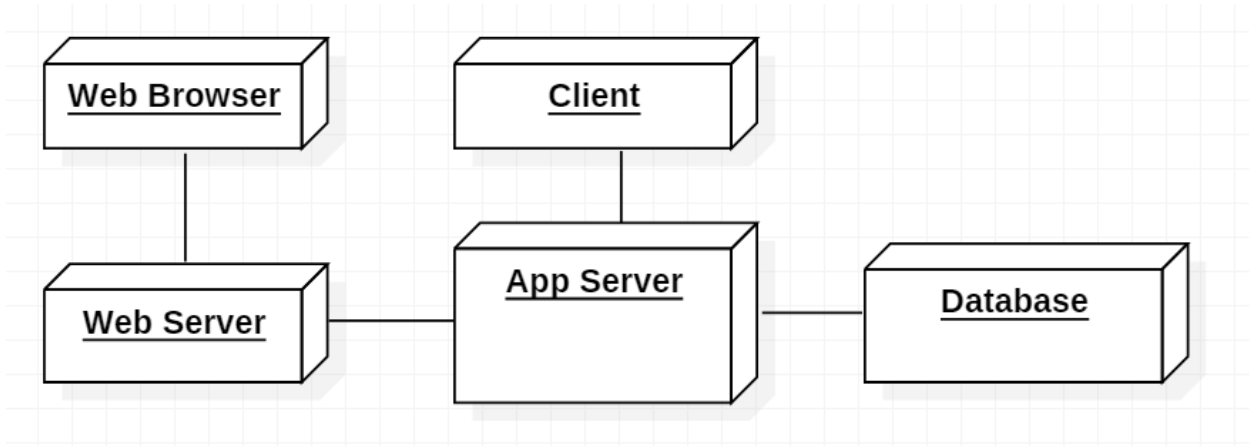
## Non-trivial algorithm:

We may integrate some machine learning APIs to return some suggested or related views to the user.

# 5. High Level UML Diagrams



## UML Deployment Diagram

## 6. Identify actual key risks for your project at this time

| | |
|---|---|
| Skills | Unfamiliarity with Pug syntax.<br>**Resolution: Learn syntax** |
| Schedule | Deadlines are in danger of not being met.<br>**Resolution:**<br>1. **Re-prioritize website features.**<br>2. **Reduce scope of website features.**<br>3. **Re-delegate tasks.** |
| Technical | Image storage configuration<br>**Resolution: Use a method that fits the scale of the project that is mindful of the schedule and skill limitations.** |
| Teamwork | A task requires more work than initially considered.<br>**Resolution: Tasks delegation and communication with teammates on their workload outside of CSC648.** |
| Legal/Content | To avoid privacy issues concerning addresses and private properties.<br>**Resolution: Use business addresses and stock photos.** |