|  |
| --- |
| SW Engineering CSC648/848 Section 02 Fall 2017 |
| 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:   * Functional Requirements V2 has been rearranged to group by priority rather than user. * Home Page has been changed to give an explicit login button for Agent. * 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. * Details Page now has the search navbar under the main navbar. * Login/Register Page now has a “Forgot Password” link and has a place to try a captcha. * Agent’s My Listing Page has been renamed to Agent’s Dashboard. * 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. * Added a UML Deployment Diagram |
| 10/28/17 | FROZEN |

Table of Contents

[1. Data Definition V2 2](#_Toc496990820)

[2. Functional Requirements V2 4](#_Toc496990821)

[3. UI Mockups and Storyboards 5](#_Toc496990822)

[Mary’s (Agent) Use Case Flowchart 5](#_Toc496990823)

[Sarah’s (Seller) Use Case Flowchart 6](#_Toc496990824)

[Henry’s (Buyer) Use Case Flowchart 7](#_Toc496990825)

[Home Page 8](#_Toc496990826)

[Listing Results Page 9](#_Toc496990827)

[Details Page 10](#_Toc496990828)

[Login/Register Page 11](#_Toc496990829)

[Login Flowchart 12](#_Toc496990830)

[Agent’s Dashboard 13](#_Toc496990831)

[Buyer’s and Seller’s Dashboard (P2) 14](#_Toc496990832)

[4. High-level system architecture 15](#_Toc496990833)

[Architectural Design Pattern 15](#_Toc496990834)

[MVC Model 15](#_Toc496990835)

[Database Organization: 16](#_Toc496990836)

[Media Storage: 16](#_Toc496990837)

[Search/filter architecture and implementation: 16](#_Toc496990838)

[Non-trivial algorithm: 16](#_Toc496990839)

[5. High Level UML Diagrams 17](#_Toc496990840)

[UML Deployment Diagram 17](#_Toc496990841)

[6. Identify actual key risks for your project at this time 18](#_Toc496990842)

# Data Definition V2

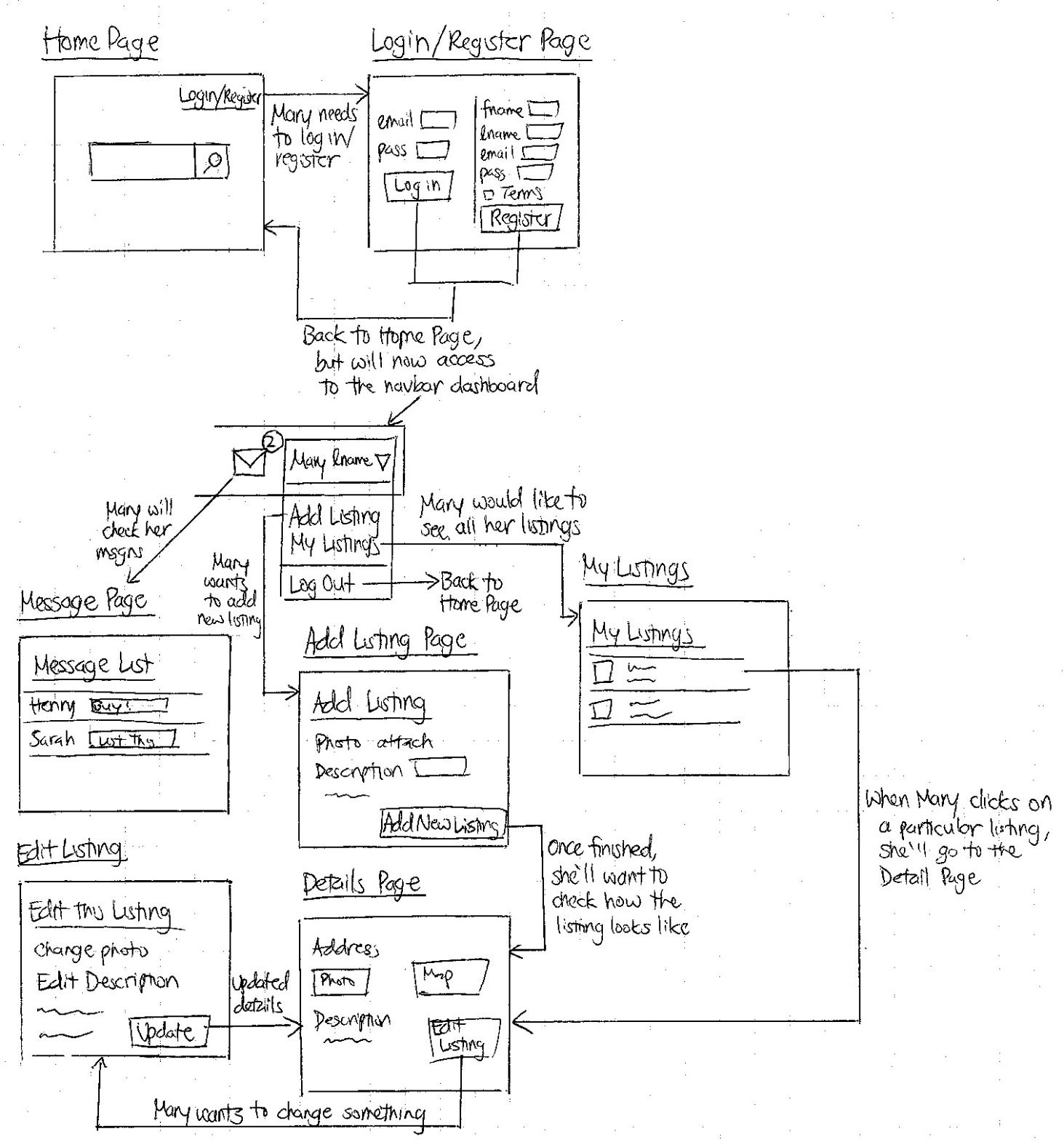
* **Unregistered User**: user/users who either do not have an account or have not logged in on the site.
  + Can view **listing** and use the **search bar**.
  + They cannot use the **messaging system** and do not have access to the **dashboard**.
    - To access these features, an unregistered user 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
  + Number of specific types of rooms
    - Bathrooms
    - Kitchens
    - Bedrooms
  + Number of floors
  + Real Estate Type
    - Condo
    - Mansion
    - House
    - Apartment
* **Listings**: a real estate of which only the **agent account** can post.
  + Price
  + Description
  + Photo(s) (min: 1, max: 3)
  + Real Estate Type (Condo, Apartment, Mansion, House)
  + 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
  + An **unregistered user** may search by city, zip code, and address

# Functional Requirements V2

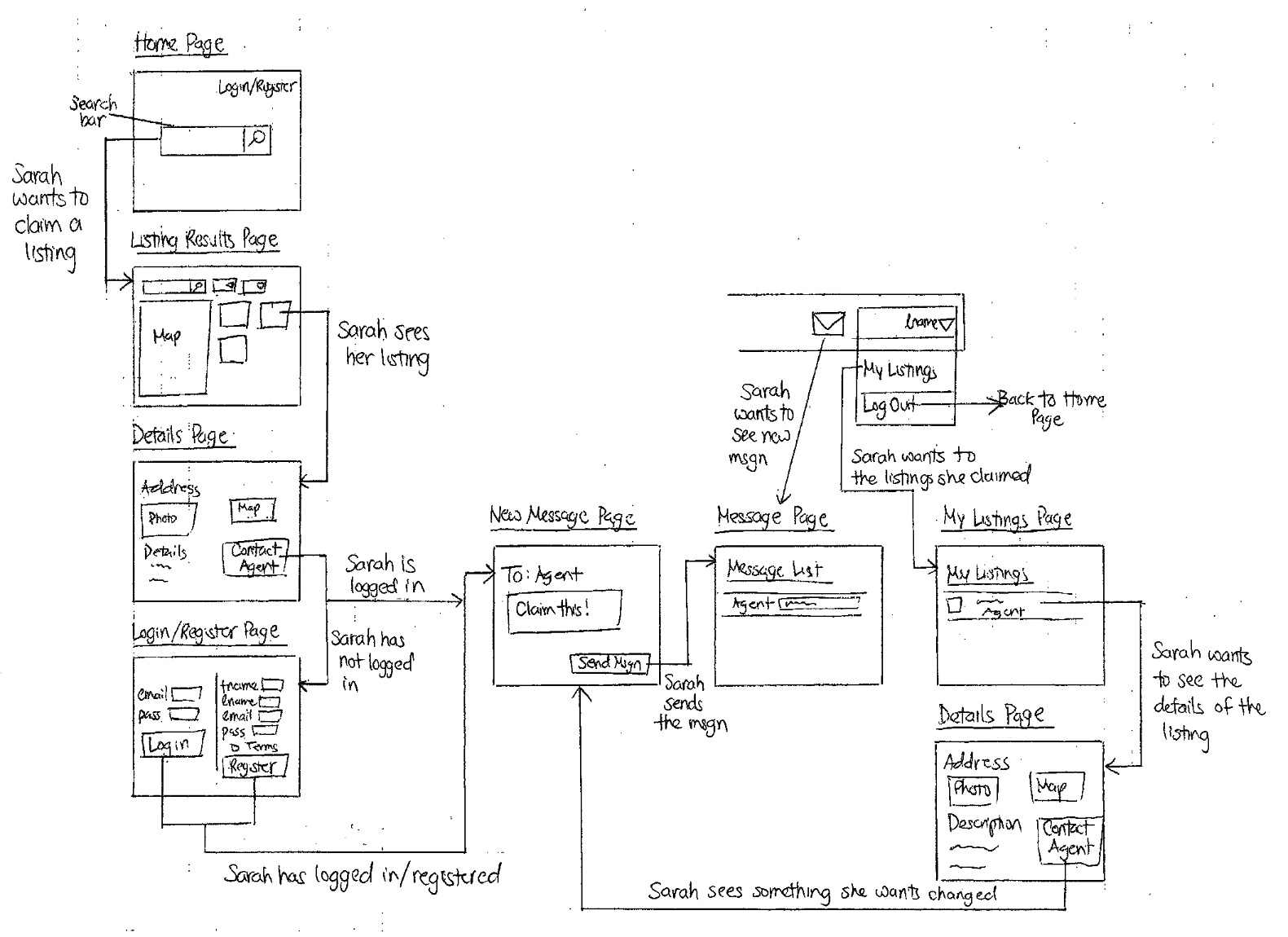
|  |  |  |
| --- | --- | --- |
| Priority Rank | Requirements | User |
| Priority 1 | 1. An **unregistered user** shall be able to search the website for **listings** via the **Search Bar** on the main landing page.    1. An **unregistered user** shall be able to search via cities. 2. An **unregistered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

# UI Mockups and Storyboards

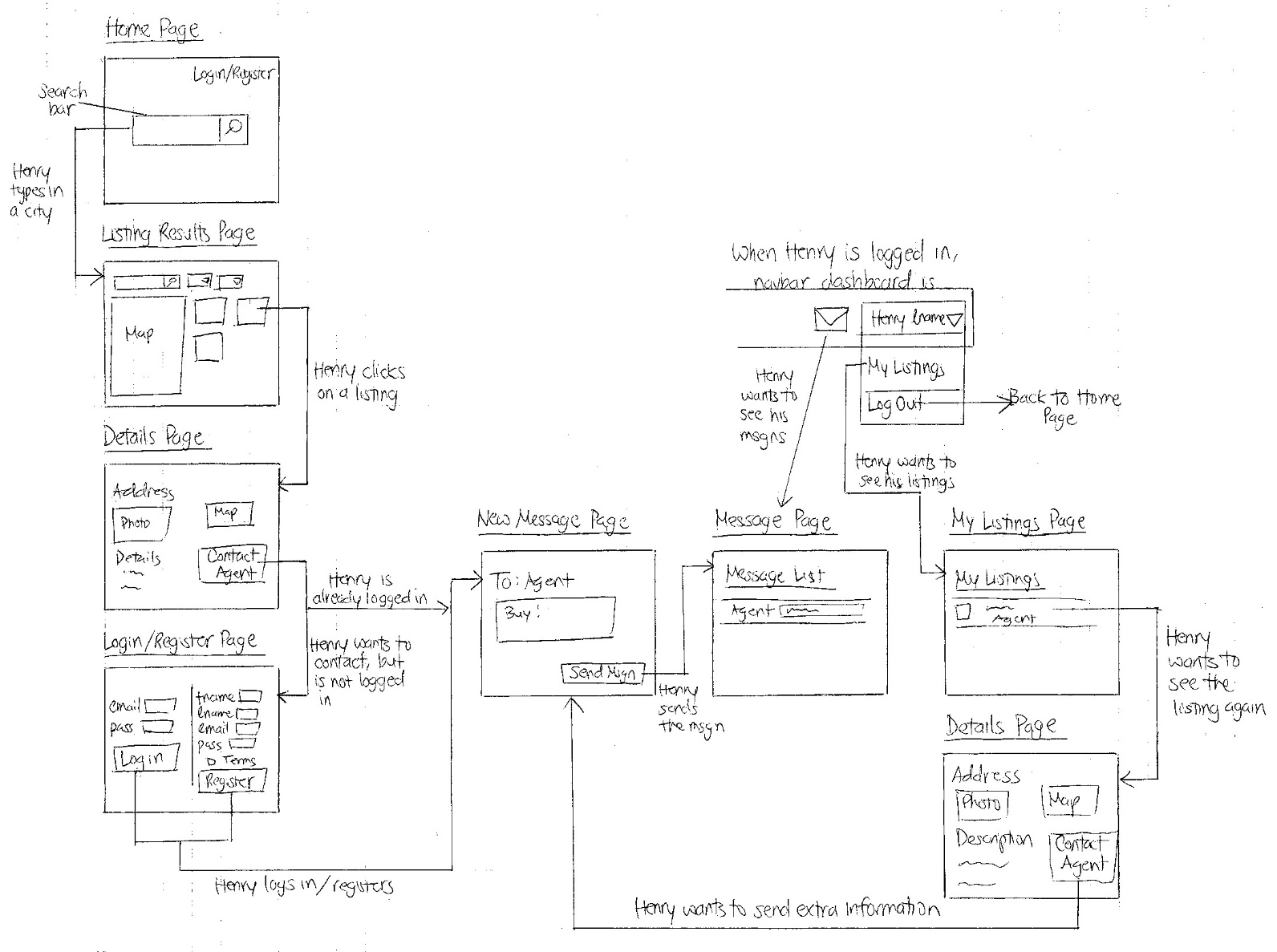
## Mary’s (Agent) Use Case Flowchart



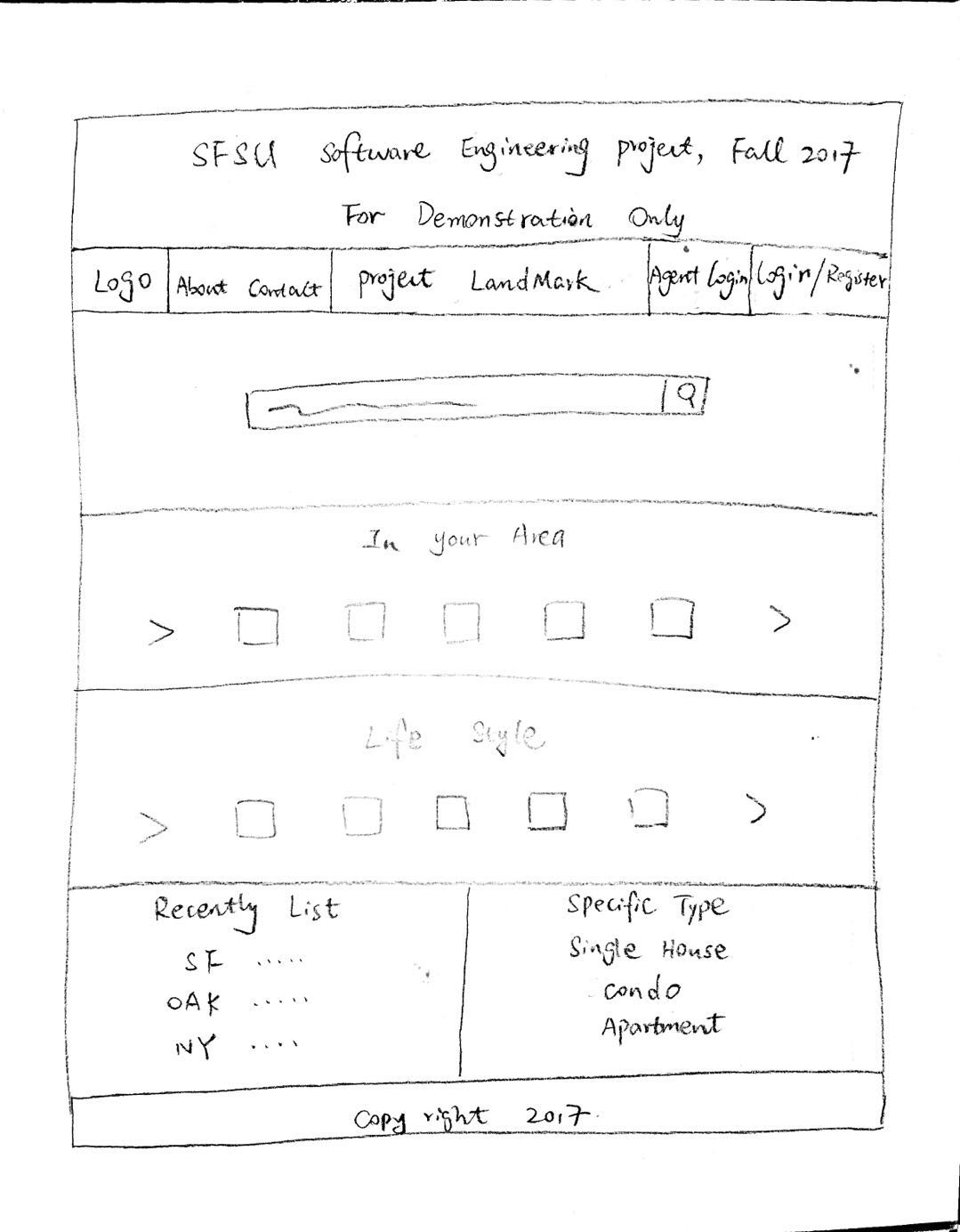
## Sarah’s (Seller) Use Case Flowchart



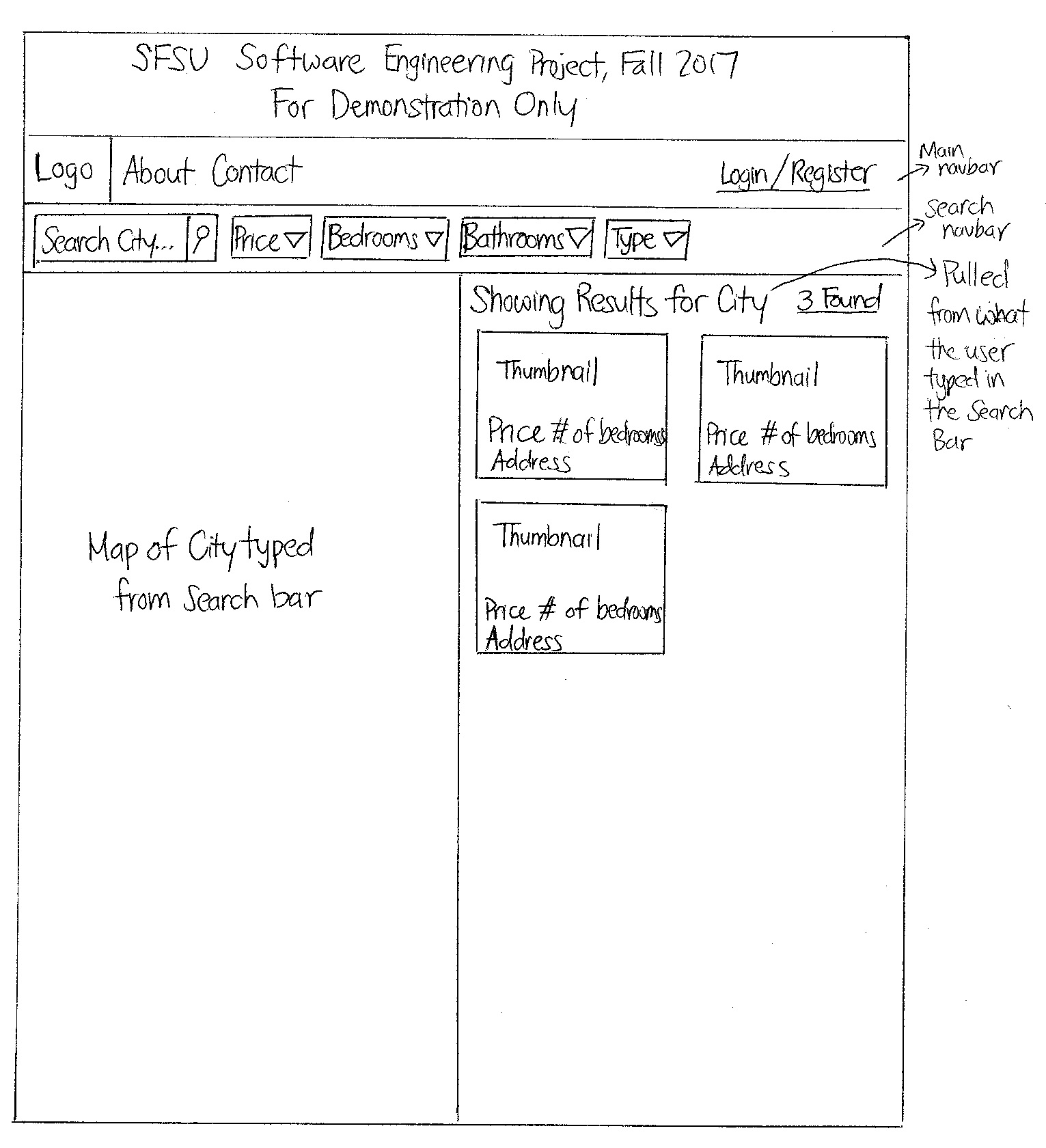
## Henry’s (Buyer) Use Case Flowchart



## Home Page



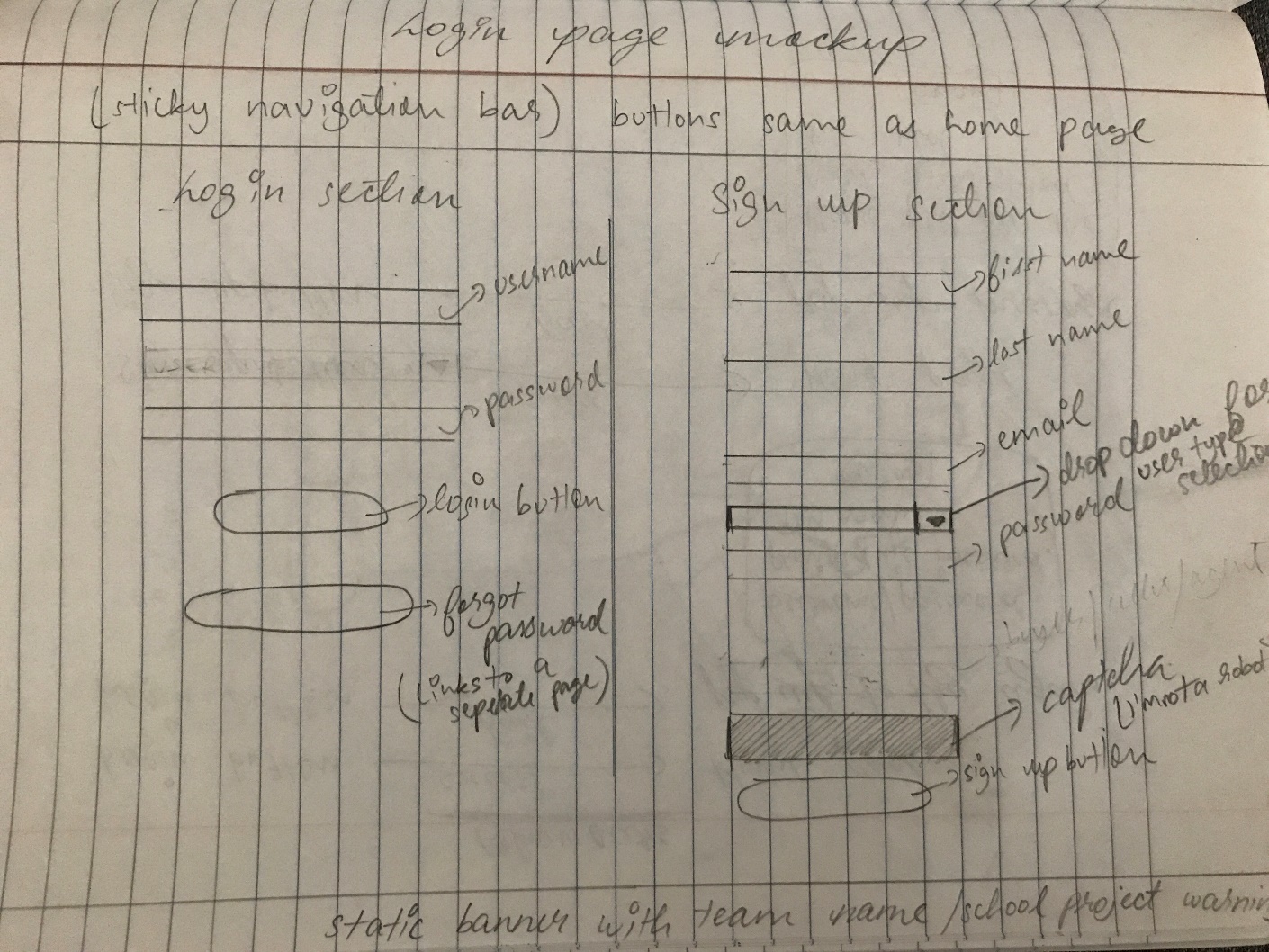
## Listing Results Page



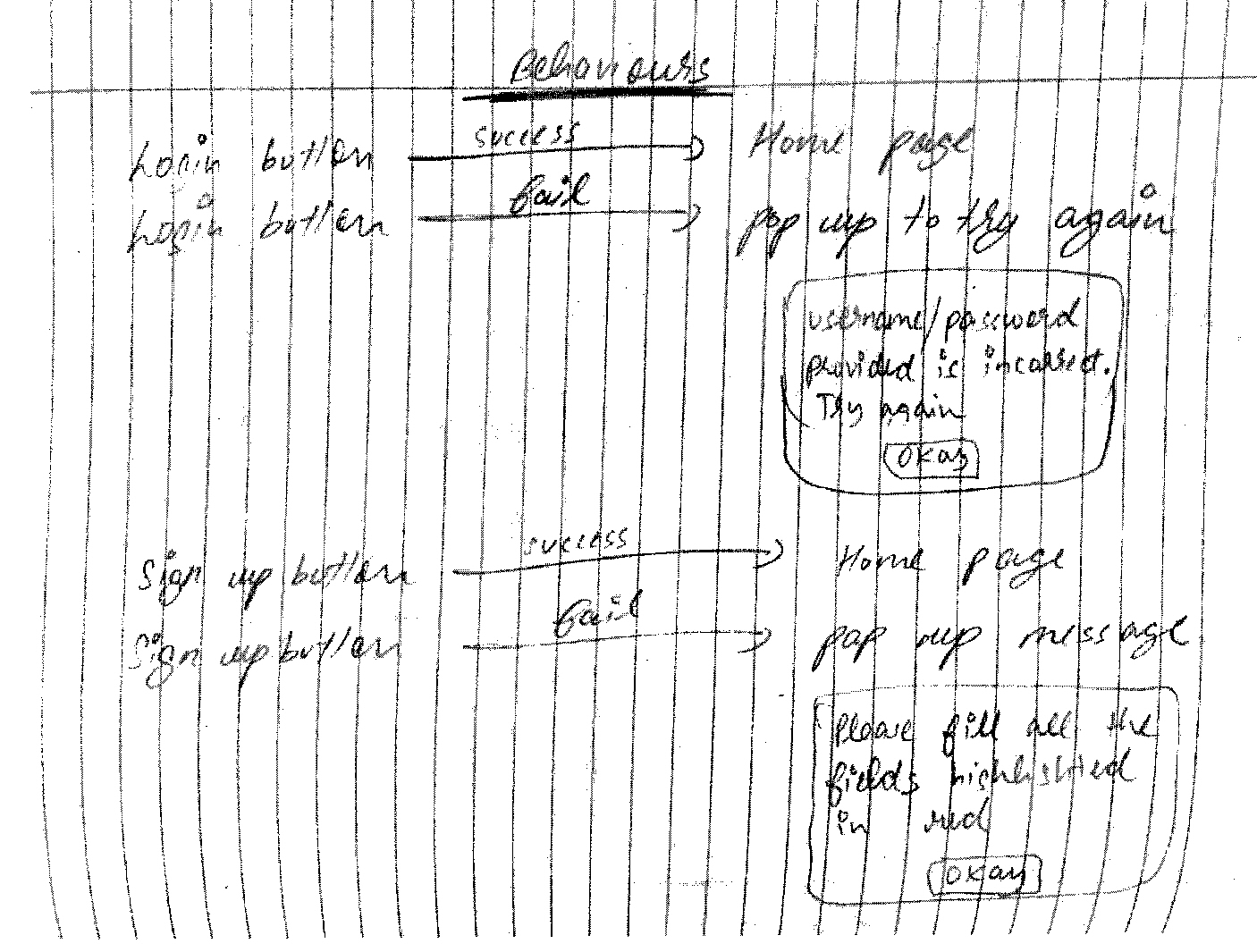
## Details Page



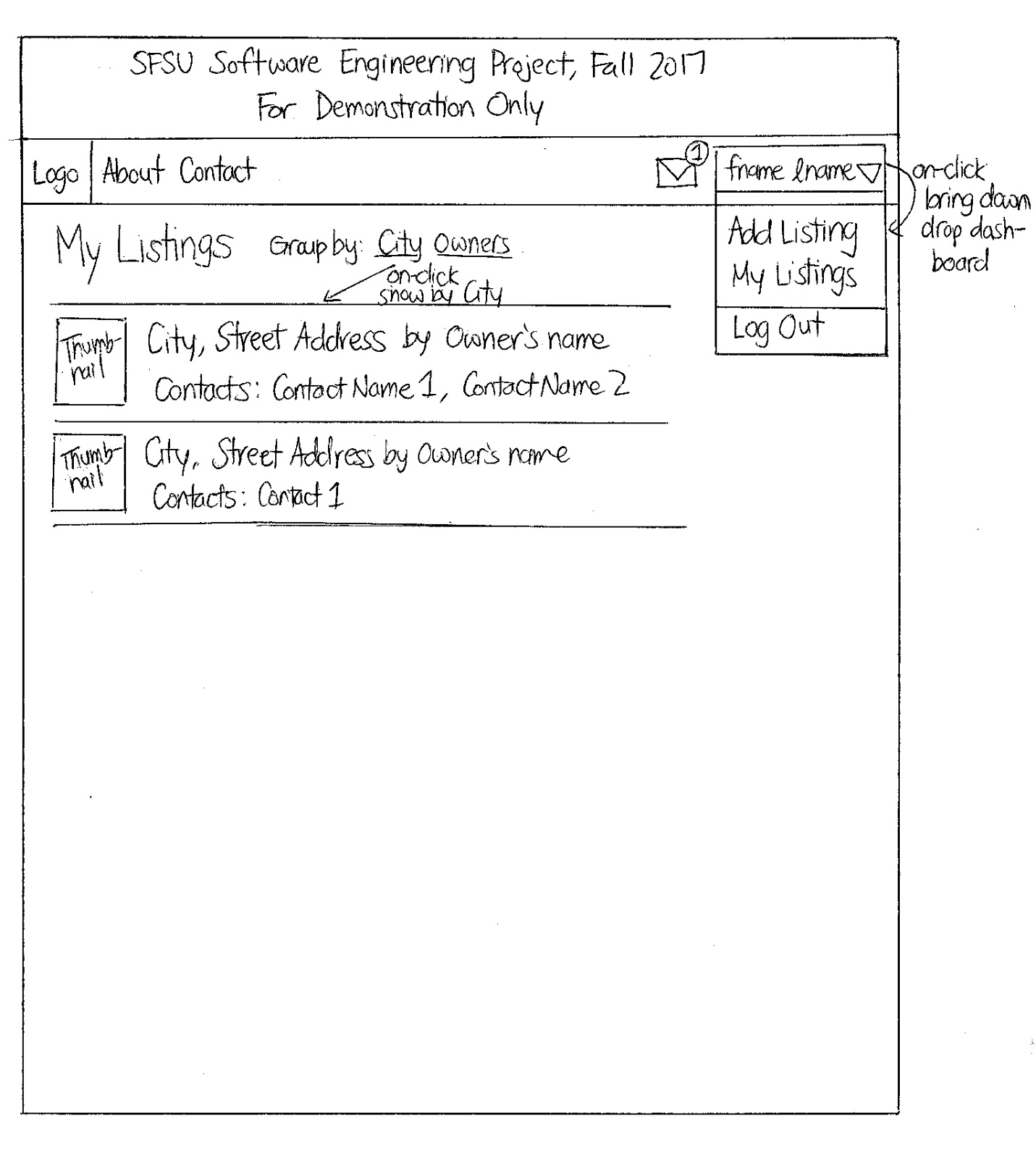
## Login/Register Page



## Login Flowchart

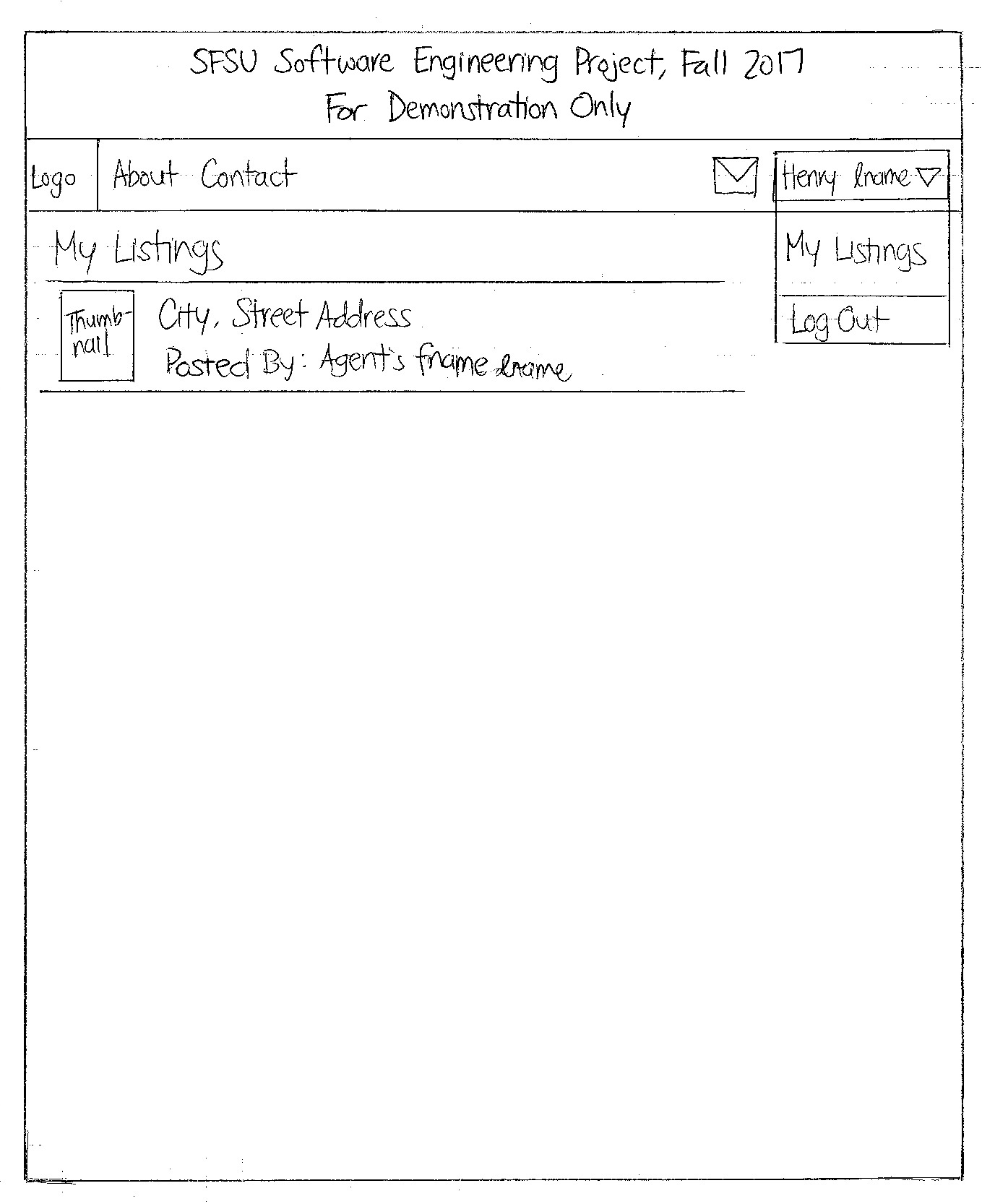


## Agent’s Dashboard



## 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.

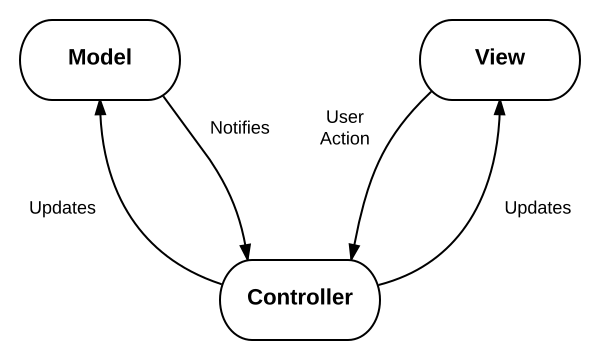


# High-level system architecture

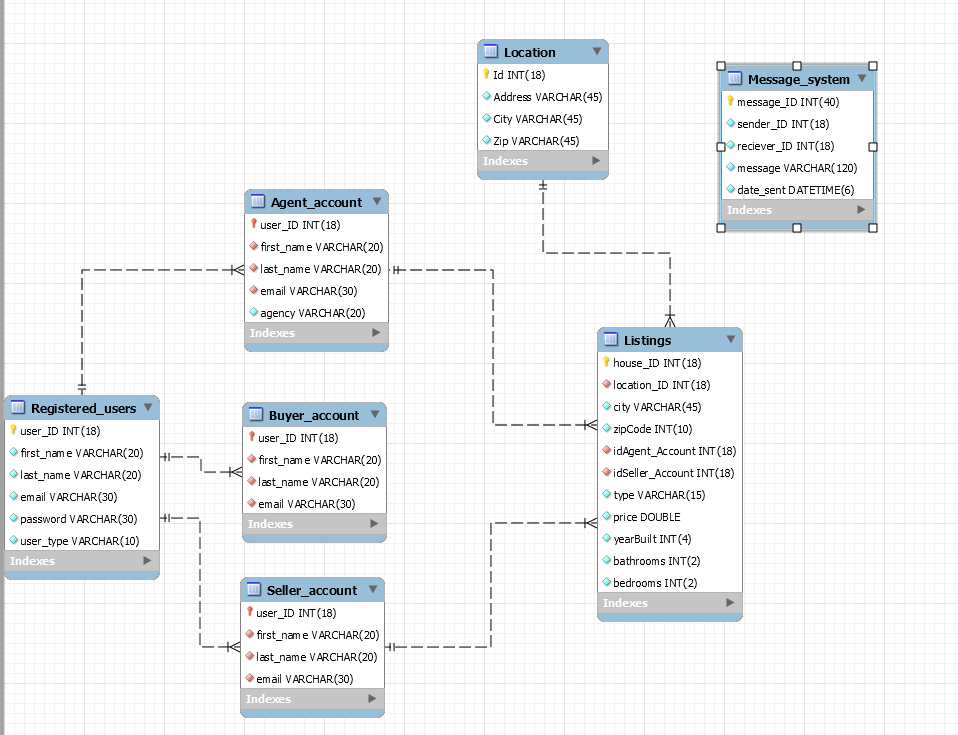
## Architectural Design Pattern

* 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.

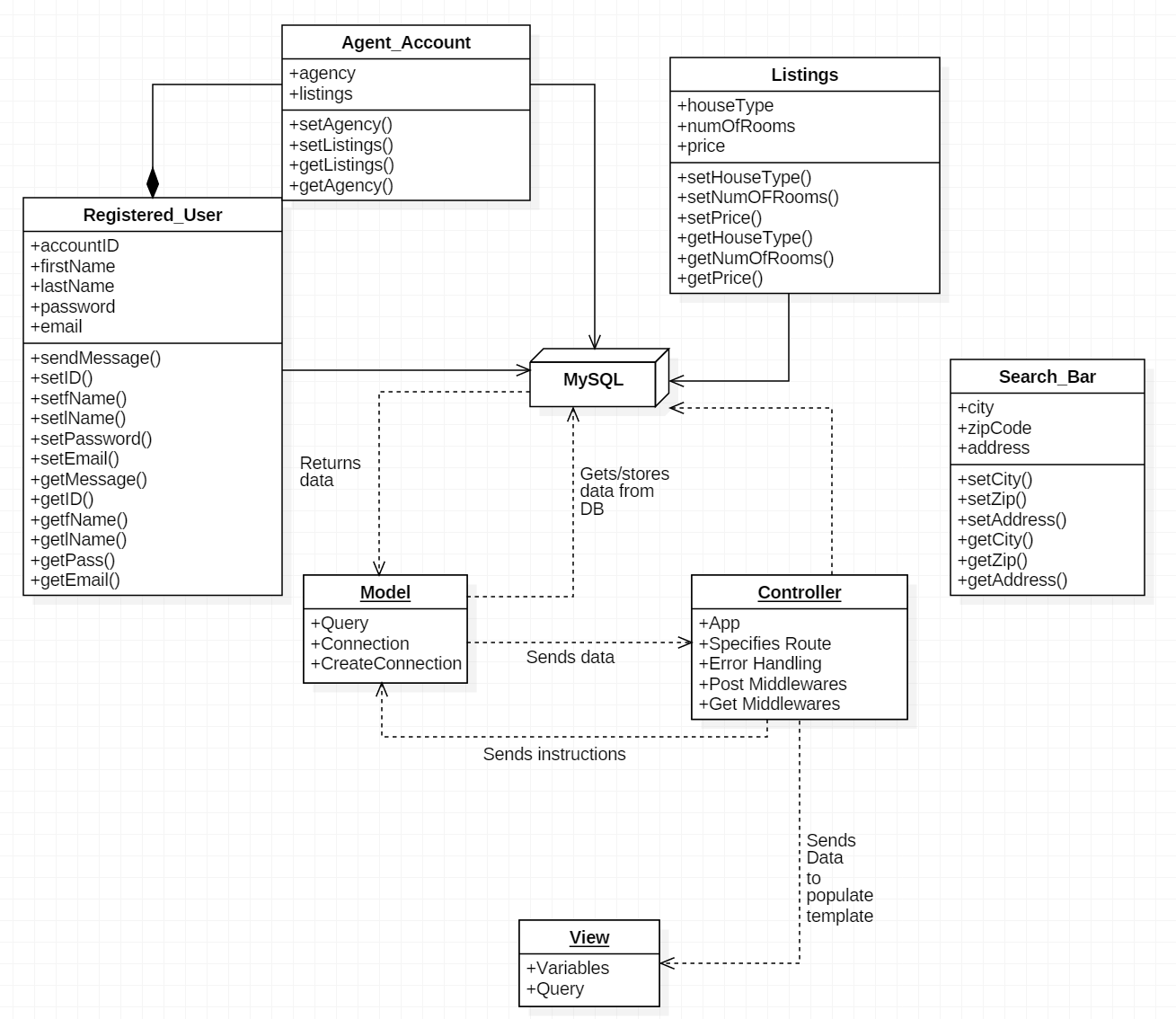
## Search/filter architecture and implementation:

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.

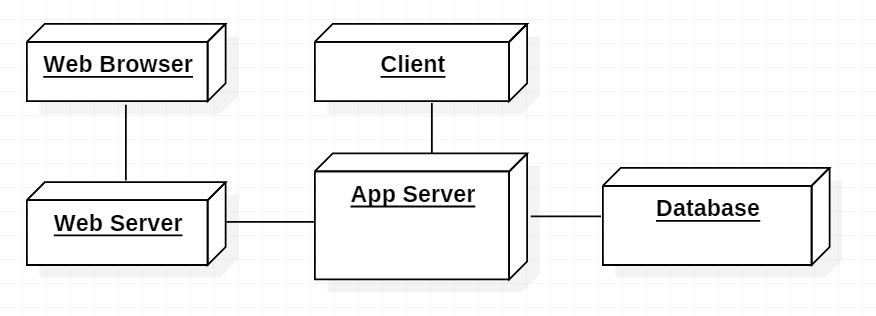
## Non-trivial algorithm:

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

# High Level UML Diagrams



## UML Deployment Diagram



# Identify actual key risks for your project at this time

|  |  |
| --- | --- |
| Skills | Unfamiliarity with Pug syntax.  **Resolution: Learn syntax** |
| Schedule | Deadlines are in danger of not being met.  **Resolution:**   1. **Re-prioritize website features.** 2. **Reduce scope of website features.** 3. **Re-delegate tasks.** |
| Technical | Image storage configuration  **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.  **Resolution: Tasks delegation and communication with teammates on their workload outside of CSC648.** |
| Legal/Content | To avoid privacy issues concerning addresses and private properties.  **Resolution: Use business addresses and stock photos.** |