

SW Engineering CSC 648/848 Fall 2014

“Listed”

Local Group: 04

| | |
|--------------|--|
| Lead: | Dhruvkumar Joshi joshi@mail.sfsu.edu |
| | Gilbert Szeto gszeto@mail.sfsu.edu |
| | Kumari Sweta ksweta@mail.sfsu.edu |
| | Jordan Guinn jguinn@mail.sfsu.edu |
| | Yefa Qi kevinqiyf@hotmail.com |

Milestone 2

History Table:

| Date | Comments | Version |
|----------|---|---------|
| 10/15/14 | First initial writeup by the team | 1.0 |
| 11/21/14 | Updated application per specs from Professor Petkovic | 1.1 |

Table of Contents

| | |
|---|----|
| (1) <i><u>Executive Summary</u></i> | 3 |
| (2) <i><u>Data Definition/Glossary</u></i> | 4 |
| (3) <i><u>Use Cases</u></i> | 5 |
| (4) <i><u>Functional Specs</u></i> | 6 |
| (5) <i><u>Non-Functional Specs</u></i> | 8 |
| (6) <i><u>UI Mockups and Storyboards</u></i> | 10 |
| (7) <i><u>High level Architecture and Database Organization</u></i> | 19 |
| (8) <i><u>High Level UML Diagrams</u></i> | 26 |
| (9) <i><u>Key Risk</u></i> | 28 |
| (10) <i><u>Team Organization</u></i> | 29 |

Bold Letters: Words will be found in Glossary/Definitions.

1. Executive Summary:

“Listed” is a real estate web application built for a particular real estate company, allowing them to manage their listed properties. This application allows the real estate company to post the contents of residential listings, which will later be seen by potential future homeowners. These future homeowners will be able to contact the realtor once they pick a home.

Buyers will be able to browse all property listings on the site or search for properties based on various attributes like zip code, city, etc. They will also be able to contact a specific realtor if they come across a desirable property. It also allows the current homeowner to contact the realtor with information about the home, which the realtor will then post online.

In addition, buyers will be able to determine the exact location of each property with the assistance of Maps. They will have access to a picture of the home, as well as a general description of the home itself and its surrounding neighborhood. Buyers will also have the option of connecting their Listed account to another social network account.

This web application is the real estate application for the experienced and inexperienced, with functionality for both. The website can be easily navigated by those without experience looking for homes. It displays the location of the properties being looked at, allowing the user to determine if the neighborhood they’d be living in fits their lifestyle. It provides contact information for multiple realtors, more than capable of assisting with the process of buying property.

Realtors will also get substantial usage out of this web application. By using various social media accounts, realtors are able to access more information about the client. This will help the realtor stay in the loop. Lastly, one of the more important features of the website is the “Open House” Calendar. This will let users keep track of what homes they would like to see over the course of several days.

“Listed” is a well designed interface for both the client and the realtor. With the integration of Maps, Social Media, and Open House Calendar, “Listed” will be the future of property listing.

2. Glossary/Data Dictionary:

Buyer:

Any person who is interested in purchasing a property and pays money to the seller to buy it. Can be either registered or unregistered. He needs to login and register in order to access the online service. Once he registers, his information is stored in database. He is financially capable of buying the property. His information may be used for background checking.

Seller/Owner:

Any person interested in selling real property. He is the owner of property and has legal rights to sell the property. Needs to register. All the information about the seller is stored in database.

Realtor:

A real estate agent who is a member of the National Association of Realtors. Realtors conduct their business and activities in accordance with a strict Code of Ethics. He can assist customers in both the buying and selling of their homes. He takes care of all legal work of client, required by local government. Also, assists his clients in mortgage or loan related processes.

Unregistered user:

Can access all listings. He gets to see minimal information on the website. Does not need to login/register. Information about unregistered user would not be stored in database.

Registered User:

Can access all listings as well. Can also access open house times, specify attributes of listings they would be interested in, contact realtors about given listings and connect directly to their various social media accounts. Needs to login/register. He should accept terms and conditions mentioned by the website.

Public listing/content

These listings can be viewed by any user(registered or unregistered). It will have at least one image of property. The listing contains information about the property, for e.g its price, location, size and walk score.

Administrator:

A person who controls the activities/posting on the website, content management system, managing access rights of different users of a website, monitoring security configuration. He can access all contents and delete inappropriate content, but can not modify the content. He will be also responsible to manage the database. He needs to login to perform above actions.

Walk Score:

It is a walkability index that assigns a numerical walkability score to any address. It means the extent to which the location is friendly to the presence of people living, shopping, commuting, visiting, enjoying or spending time in an area.

Open House:

Available exclusively to registered users, an open house allows buyers to physically observe the listing during specific times provided by the realtor.

3. Use Cases:

Owner of Real Estate Company

David runs a **Real Estate** Company in Northern California. His company is quite successful, so he wants to make a brand new website for it. He wants it to be powerful enough to manage all his listings day in and day out.

As an Owner, he wants to take his real estate business to next level. He wants to allow his customers to access the realtor-managed listings online.

Buyer

John is a math teacher at San Francisco State University, looking to purchase a property before he gets married. Since he works everyday, he's keeping his options open: a studio, apartment or full-size home are all possibilities for him.

John goes to our website and searches for properties near SFSU by entering a zip code in the search box. As an **unregistered user**, he can sort the listing by price and number of bedrooms.

Shawn is a car dealer. He is trying to invest in the property. As an investment he chooses to buy house. He want to see the house personally. As **registered user** he will be able to see the **Open House calendar** of the current month.

Ashley is a Doctor at UCSF. She is looking for a house in San Francisco. Ashley is looking for her dream home, so she visit the website and look at all the listings in San Francisco. Unfortunately, she did not like any of them. So, as a **registered user**, she can choose to receive emails regarding any new listings that match her criteria.

Seller

Yash is trying to invest in a business and is looking to sell his property. He is having a difficult time locating a website that best fits that needs when he comes across our website. He quickly becomes a **registered user** using some **basic information** like his name, email, and phone number. He is made explicitly aware that all of this information will be private and not shared with anyone **without his permission**. After becoming a **registered user**, he is able to connect with the **realtors**.

Admin

Dhruv is the **administrator** of the website. He is looking over the business and security aspects of the site. He has the ability to accept or reject property pictures that are uploaded by the **registered user**. He has complete control over the website's listings, content management system and monitor security configuration.

House-Flipper

Jordan is a wealthy entrepreneur who flips houses for a living. He is looking for a website that allows him to use social networking sites to improve his business. As a **registered user**, he will be able to connect his social networking sites in order to increase his prominence in business by the realtor.

4. Functional Requirements:

Must Have (Priority 1)

1. Listings shall contain the following:
 - a.) Picture(s) of the property
 - b.) Property description (address, city, zip code, price)
2. Unregistered users shall be able to browse all listings on the website.
3. Unregistered users shall be able to search listings by:
 - a.) Desired zip code

- b.) Desired city
- 5. Unregistered users shall be able to create a personal profile on the website, thereby becoming registered users.
 - a.) Profiles shall require full name & email address to be created.
- 6. Registered users shall be able to specifically describe attributes of homes they find interesting. Attributes shall include:
 - a.) Number of bedrooms
 - b.) Price
 - c.) Square footage
- 7. Registered users shall be able to contact realtors about properties they're seriously considering.
- 8. A search engine shall be available within the context of the home page.
- 9. Sellers shall be able to provide realtors with information about their home.
- 10. Realtors shall be able to post listings.
- 11. Realtors shall be able to modify and delete listings.
 - a.) Realtors shall only have access to their specific listings.
- 12. Admin will have comprehensive access to all listings and security details, excluding users' passwords.
 - a.) Admin shall have authority to edit or remove listings if content is deemed inappropriate in any way.
- 13. All users can read the website's privacy policy.

Desired (Priority 2)

- 4. Unregistered users shall be able to determine house locations with Maps.
 - a.) Specific house shall be highlighted on map, with surrounding neighborhood viewable as well.
- 14. Registered users shall have access to open house times for listings of

- their choosing.
15. Using the Open House Calendar, realtors shall be able to specify times for their listings to be physically viewable.
 - a.) Each listing may have more than one time posted.

Opportunistic (Priority 3)

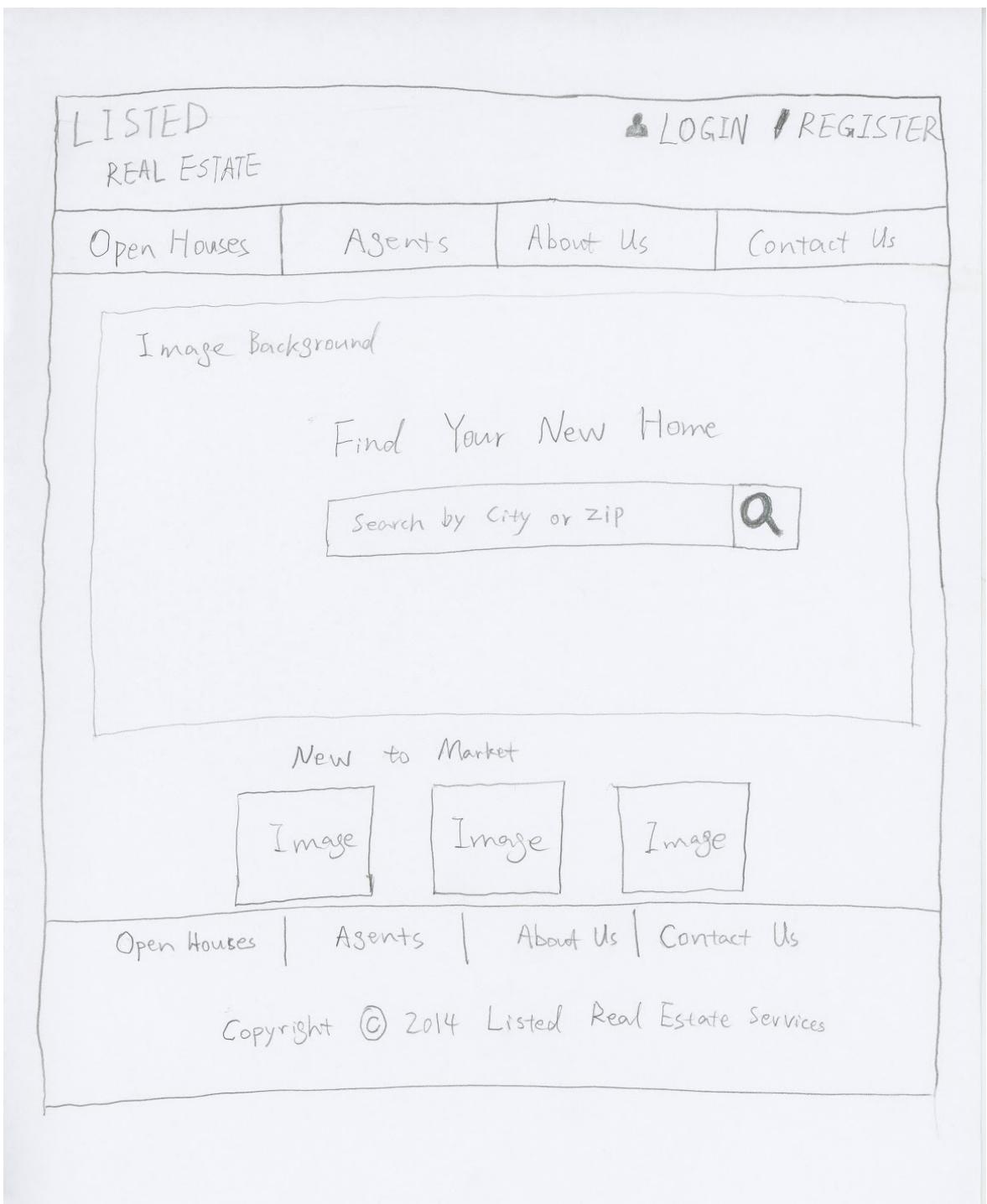
16. Unregistered users shall have access to rating of each house's location.
 - a.) Rating combines quality of school district, house's Walk Score and the year the house was built.
17. Unregistered users shall be able to search for listings by address.

5. Non-Functional Requirements:

1. Application shall be developed using class provided LAMP stack.
2. Application shall implement MVC architecture, with multiple views dependent on the user.
3. Application shall be viewable in standard desktop/laptop/mobile browsers, and shall render properly on the two latest versions of all major browsers: Mozilla, Chrome and IE.
4. Application shall render well on portable devices using the same source code as for desktop & laptop browsers..
5. Application shall be deployed on Amazon Web Services as specified in class.
6. Data shall be stored in the database on the class server in the team's account, pictures included.
7. Application shall have content access permissions for three separate roles: admin, realtor and user (both registered and unregistered).
8. No more than 50 concurrent users shall be accessing the application at any time.
9. Application shall have an availability of 90% (0.90).
10. Response time of a given page shall be no more than 3 seconds.

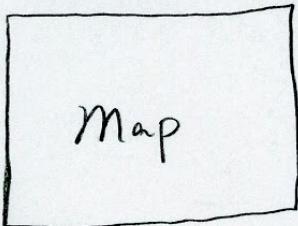
11. Application shall implement 128-bit encryption to protect users' passwords.
12. Privacy of users shall be protected and all privacy policies shall be appropriately communicated to the users.
 - a.) Privacy policy shall explain how Listed shall not solicit users' personal information to various third parties without explicitly requesting the users' permission.
13. The language used shall be English, & English exclusively.
14. Application shall be very easy to use and intuitive. No prior training shall be required to use the website.
15. Google analytics shall be added for major site functions.
 - a.) Google Maps API shall be utilized for displaying listings.
16. The website shall prominently display the following text on all pages "SFSU/FAU/Fulda Software Engineering Project, Fall 2014. For Demonstration Only"

6. UI Mockups and Storyboards (high level):



Contact Us

Our Locations



Map

Office Address

Phone #

Hours

How may we assist you?

First Name:

Last Name

Phone

Email:

Preferred Contact Method: No Preference

Question:

SUBMIT

Login Page

A hand-drawn sketch of a login interface. At the top left is a "Login" button. In the center is a "Sign in with LinkedIn" button with the LinkedIn logo. Below it is a "Email or Username" input field and a "Password" input field. To the right of the "Email or Username" field is the text "- Or -". At the bottom left is a "Sign In" button. At the bottom right are links for "Create new account" and "Forgot Password".

Forgot Password

A hand-drawn sketch of a forgot password page. At the top left is a "Forgot Password" button. Below it is the text "Recover Password: Enter the email that you used to create the account and we will send you your password". There is an "Email" input field and a "Email My Password" button at the bottom.

Agent

Find an agent



Name
Phone #
Email



Name
Phone #
Email



Name
Phone #
Email

About Us

About Listed

All information about
Listed

Register

Register Now!

Create an account to get email alerts, save favorite
properties and more.



Sign in with LinkedIn

— Or —

First Name:

Last Name:

Email:

User name:

Password:

I Accept All Rules & Privacy Policies

Create Account

Cancel

Property Search Results

Showing 1—10 out of 123

Sort by

City or ZIP

Property Type ▼

+ Beds ▼

- Min Price ▼

+ Max Price ▼

Open Home



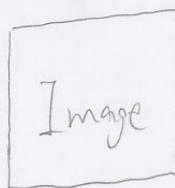
Minimum Info

More details



Minimum Info

More details



Minimum Info

More details

<< Previous | 1 2 3 ... Next >>

click
↓

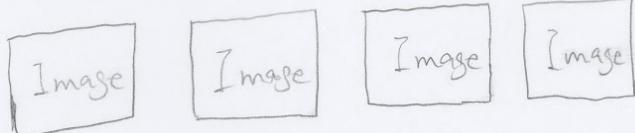
For registered users, it will link to
Property Info page

For new users, it will link to
Register page.

Property Info

Address of the house

\$ 1,234,567



Description/ Details



Street Map



Contact Us Now to Order

Contact Us

Open Houses

Listed

Login

Open houses

December 2014

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-------|-------|-----|-----|-----|-----|-------|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | Image |
| Image | Image | 30 | 31 | | | |

Buyers click these images will link to
the "Property Info" page.

7. High level Architecture and Database Organization:

The basis for our system architecture is the widely used LAMP stack. The stack is an aggregation of 4 categories of software necessary for creating a website. A Linux-based operating system, Ubuntu, is the one we will be using. Apache will be used for web serving documents and receiving/responding to requests. MySQL is the database management system used to manage records for our website. PHP is the programming language used to generate HTML code for web serving.

For front-end development we decided to go with jQuery for a JavaScript library and Bootstrap for HTML and CSS templates. This will provide us with a foundation to quickly create a functioning website.

We are also considering Google Maps API for property lookup, LinkedIn login APIs from various social media sites, and Walk score API to calculate the rating.

Our application shall be supported by the three most frequently used web browsers: Firefox, Chrome, and Internet Explorer.

LAMP Stack

- Linux Kernel/Ubuntu v.12.04.3
- Apache Webserver v2.2.22
- MySQL Database Management v14.14
- PHP Server-side scripting v5.3.10

PHP MVC Skeleton:

- php-mvc

JavaScript Library for Front-End Scripting

- jQuery v1.11.1

Front-End Templates

- Bootstrap v3.2.0

APIs

- Google Maps Embed
- LinkedIn Login API
- Walk score API

Supported Browsers

- Latest: Firefox/Chrome/Internet Explorer

Model classes:

- User
 - CRUD methods : createUser(), deleteUser(), updateUser(), getUser(), setRoles(), getRoles(), changeRoles(), showPraferances()
- Preferences
 - updatePraferances()
- Property
 - CRUD methods (create, retrieve, update, delete)
 - searchByName(), searchByPrice(), searchByAddress(), search()
- Authorization
 - getPassword(), updatePassword(), deletePassword(), validatePassword()
- Address
 - CRUD methods
- Listing
 - CRUD methods
 - searchListing(),getProperty()

The database will be used for following purposes:

1. To store the user information. User can be registered user, unregistered user, administrator, realtor, seller, and buyer.
2. To store the property information.
3. To store the listing information.
4. To store the image information. The image can be of property and user.
5. To store the address information. The address can be of property and realtor.

The user table will be generated by system when the user signs up. The information about the unregistered user will not be stored in database. The user table information will be used by system to validate his credentials and allow him to login. Information about the property and listing will be created by the real estate company and user can see the information. The address of a particular property and realtor can be retrieved from the address table.

All the above mentioned information is stored in the database so that it is available for information retrieval and modification.

The database consists of following eight tables :

User:

| Attributes | Data Types | Description |
|------------|--------------|---|
| Id | Bigint | Unique Id of user |
| Type | Enum | Registered user, administrator, realtor |
| Name | Varchar(50) | Name of user |
| Email | Varchar(100) | email id of user |
| Image_id | Bigint | id of user's image |
| Password | Varchar(255) | encrypted password |
| Address_id | Bigint | Address of the user |

Property :

| Attributes | Data Type | Description |
|-----------------|-------------|--|
| Id | BigInt | Unique id of property |
| Name | Varchar(50) | Name of property |
| Price | Float(20) | Price for property |
| School District | Boolean | Whether the school exists in the same district as that of property |
| Size | Float(20) | Size of property in terms of square feet |
| Unit type | Varchar(20) | Apartment, condo, townhouse |
| Num_room | Int(20) | Number of rooms |
| Date Built | Datetime | Date when property was constructed |
| Latitude | Float(30) | Latitude of GPS location |
| Longitude | Float(30) | Longitude of GPS location |
| Address_id | Bigint | Unique Id of address |

Address:

| Attributes | Data Types | Description |
|----------------|-------------|---|
| Id | Bigint | Unique Id of address |
| Street address | Varchar(50) | Street address of property or realtor |
| Apt_number | Int(10) | Apartment number of property or realtor |
| Zipcode | Int(20) | Zipcode of property or realtor |
| City | Varchar(30) | City of property or realtor |

Listing:

| Attributes | Data type | Description |
|----------------|-----------|----------------------------|
| Id | Bigint | Unique id of listing |
| property_id | Bigint | id of property |
| Validity | Datetime | Expiration date of listing |
| Special _offer | Float(10) | Discount offer |

Image:

| Attributes | Data type | Description |
|------------|-------------|------------------------------|
| Id | Bigint | Id of image |
| Name | Varchar(30) | Name of the image |
| Type | Enum | Thumbnail, regular |
| Image | Blob | stores image in form of blob |

User_like:

| Attributes | Data type | Description |
|------------|-----------|---------------|
| User_id | Bigint | id of user |
| Listing_id | Bigint | id of listing |

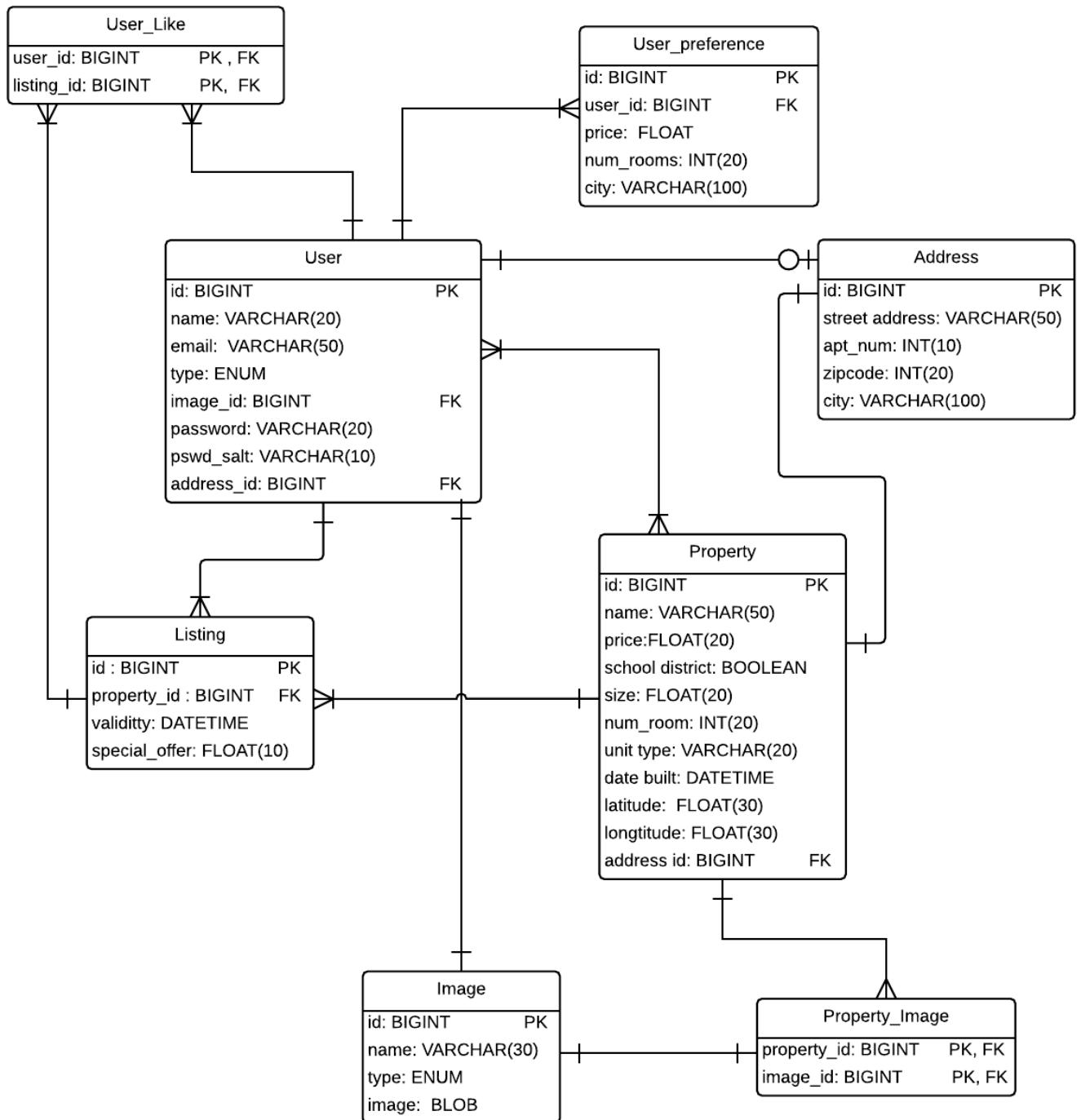
User_preference:

| Attributes | Data type | Description |
|------------|--------------|-----------------------|
| Id | Bigint | Id of user preference |
| User_id | Bigint | Id of user |
| Price | Float(20) | Price of property |
| Num_room | Integer(20) | Number of rooms |
| City | Varchar(100) | City of property |

Property_image:

| Attributes | Data type | Description |
|-------------|-----------|----------------|
| property_id | Bigint | id of property |
| image_id | Bigint | id of image |

Database Schema:



Search architecture :

1. Attributes that would be considered for search:

- City
- Zipcode

2. Attributes that would be considered for sorting:

- Price
- Size
- Date built

Based on the attributes selected, application will form the SQL query and fetch the data from the database. This would reduce the code complexity.

The result will be displayed in the descending order of the rating of selected property.

Rating algorithm:

The attributes to be considered for rating a property and their weightage:

- School district (60 %)
- Walk score (25 %)
- Date built (15 %)

Also, the rating will be based on 1 to 10 scale.

Formula used:

$$\text{Overall Rating} = (\text{School district rating} * 0.60) + (\text{Walk score rating} * 0.25) + (\text{Date built rating} * 0.15)$$

Images:

All the images related to the property and user profile will be stored in the database as Blob. Web-app related images, icons, and graphics will be stored in the file system.

property images meta data:

Name: Name of the image

Format: jpeg and jpg.

Image size:

- Thumbnail : 50 x 50
- Regular image : 600 x 450

User profile images meta data:

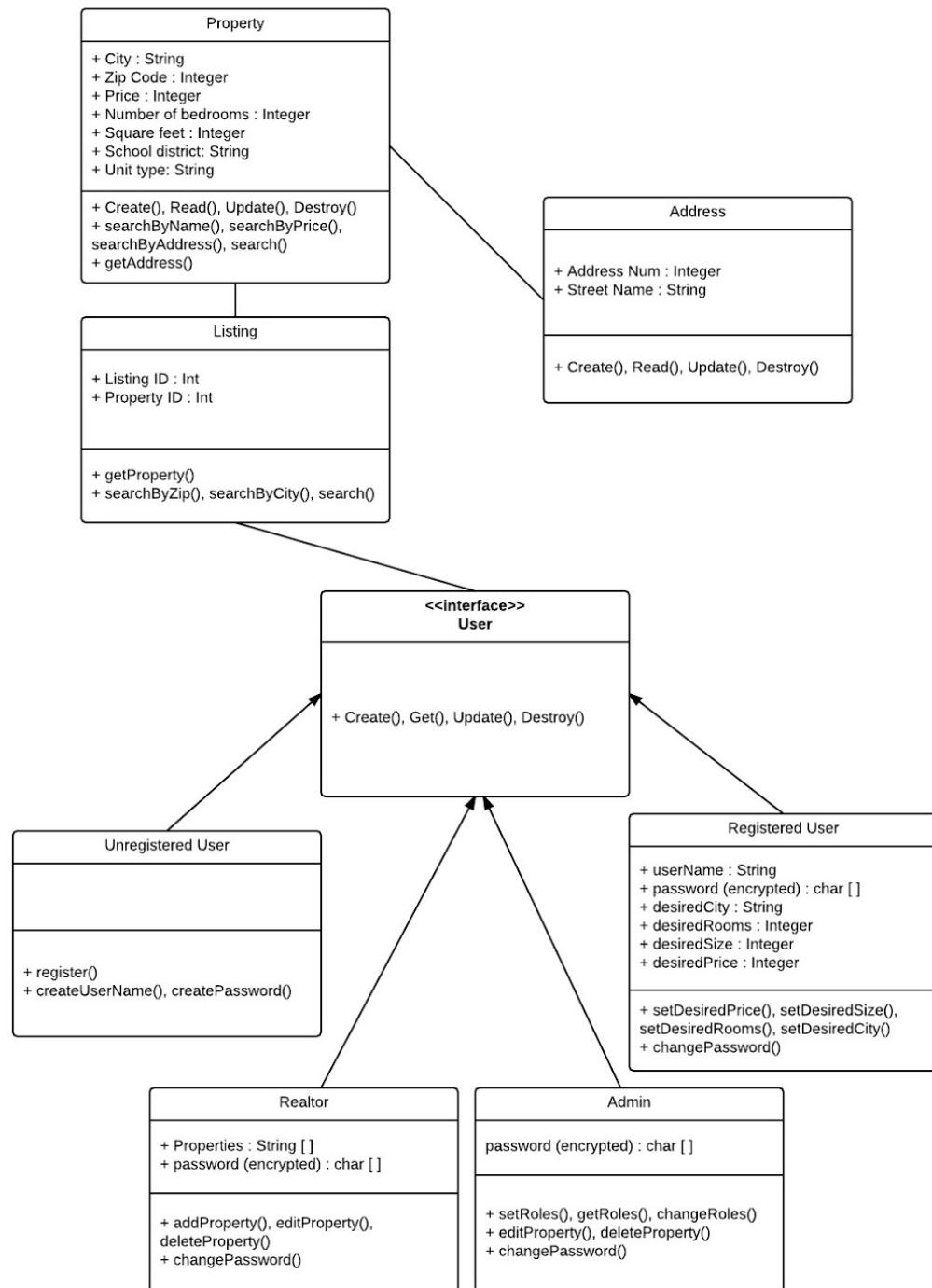
Name: UserProfile_id

Format: jpeg and jpg.

Image size:

- Thumbnail : 50 x 50

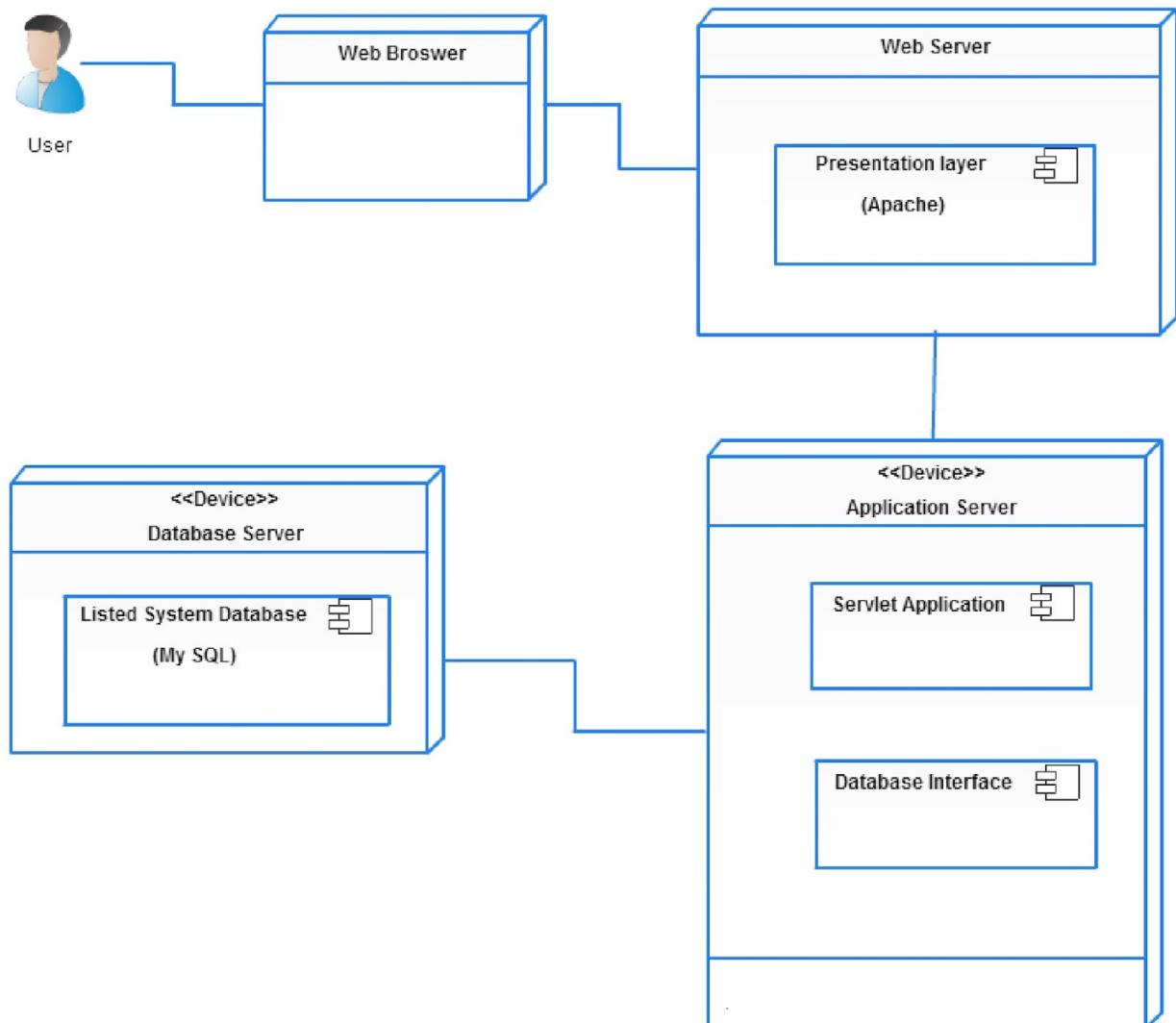
8. High Level UML Diagrams:



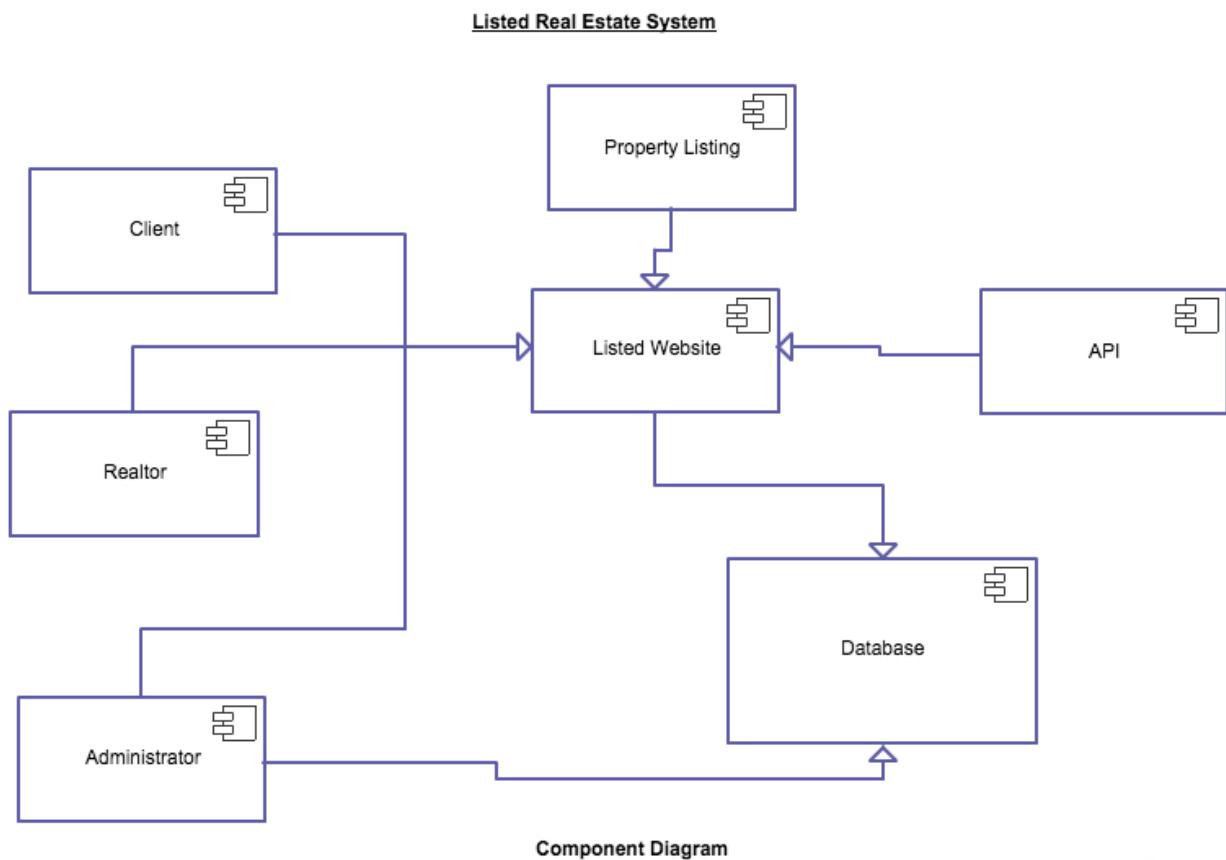
Component and deployment diagrams

(1) Deployment Diagram:

Listed Real Estate System



(2) Component Diagram:



9. Key Risks:

Skills risk:

There is a major risk of front-end development, as we could definitely use someone with solid experience in that concentration. But Dhruv and Jordan are more than willing to make up for this lack of experience with plenty of hard work.

Schedule risk:

Since the project is due in 8 weeks and most of our team is about to graduate, scheduling meeting times with all team members has proven to be quite difficult. To solve this issue we are working remotely as well as within groups to increase productivity.

Technical risk:

We are facing a minor issue with php mvc structure integration in our application. To resolve this, we are currently boosting our efforts in learning MVC structure.

Teamwork risk:

We do not see any teamwork risk in the future. Everyone is very passionate about their part of the project and ready to help the rest of the team in learning these new technologies.

Legal risk:

We are using “FREE”, “GPL”, “MIT Licensed” and “OPEN SOURCE” products in our Project. We will use the LinkedIn, Google Maps, and Walk Score APIs, which are free.

10. Team Organization:

We are a team of five passionate and hard working engineers.

1. Dhruvkumar Joshi:

As team leader, Dhruvkumar will be working on various parts of the project, from team building to deployment. However, he will focus mainly on front-end and adding multiple APIs.

2. Gilbert Szeto:

Gilbert has a great understanding of back-end programming, so he will be in charge of back-end programing in PHP. He will also be the SVN repository admin.

3. Kumari Sweta:

Kumari is overseeing all of the project’s database management, but she will also help the team with QA testing.

4. Jordan Guinn:

Jordan is very passionate about learning front-end and also helping the team with his knowledge of documentation and formal presentation.

5. Yefa Qi:

Yefa will contribute to the back-end development and will also assist Kumari with his expertise in QA testing.