

## CSC 648/848 Fall 2017

### Milestone 2: Use cases, High Level Requirements and Architecture



Franklin Henry Boswell, Team Lead (fboswell@mail.sfsu.edu)

Nehemya McCarter-Ribakoff, Backend Lead

Khanh Nguyen, Front End Lead

Evgeny Stukalov, Backend Dev

Kevin Lay, Backend Dev

Jesse Christiansen, Front End Dev

#### Revision History

Date	Revision
10/16/17	Rough draft submitted
10/27/17	M2 Submitted

## Table of Contents

Executive Summary.....	1
Use Cases.....	2
Data Definition .....	3
High Level List of Functional Requirements .....	4
List of non-functional requirements .....	5
Competitive Analysis .....	6
UI Mockups and Storyboards .....	7
High Level system architecture .....	8
High Level UML Diagrams .....	9
Key Risks.....	10

## Executive Summary

---

The real estate business is notably one of the most stable markets worldwide, post 2008, the housing market in San Francisco alone has seen a steady increase of 9.5% a year. There is a huge market readily available and here at HouseShop we have identified the key issues homeowners and homebuyers face while shopping real estate. We understand that not everyone has a luxury of having a real estate agent find them the right deal for them, and that not everyone has the opportunity to have bids electronically submitted online, but this is where HouseShop really stands out. Our product puts everyone at a even level, so no one has an advantage over someone else. This really assures our users that they are playing on a fair battleground.

We believe that real estate should not only be accessible to all types of people, but it should also be effortless to them. HouseShop extracts the hardships and burdens of buying or selling a house and throws them out the window. Our principal mission here is to combine simplicity to our customers while hiding the complex fixtures working simultaneously, to provide a truly wonderful experience to our customers.

HouseShop's solution to the competition is simple, we're offering a effortless experience that other real estate websites just can't compete with. Other websites are bulky and contain too many things that overwhelm their users. We can solve this problem by just offering the right amount. We've planned user testing around optimizing a simple interface that both homebuyers and homeowners can feel comfortable using. Real estate is a near 1 trillion dollar industry and no real estate website has captured the market completely yet. This is the perfect time for HouseShop because the market is so heavily saturated between buyers and sellers.

Our team here at HouseShop is highly experienced and qualified in building cutting edge websites that combine simplicity at the front, with complex algorithms running

simultaneously in the core. HouseShop users can trust that their experience will be unlike any other and that they are playing on a fair battleground when it comes to real estate.

## Use Cases

---

### Case 1. Non registered user

Phuong is a 45-year-old mother of three. All of her children have moved away for college and she is thinking of moving. She knows how to use a PC, but does not bother with that smartphone stuff. She finds HouseShop in an internet search and begins browsing listings. She wants to stay in the area, so she sorts **listings** by zip code. She also wants to save money for retirement, so she adds a max price **filter** to see what is in her price range. She sees several listings she likes, so she decides to save them and is prompted to create an **account** to save them to her **watchlist**. Once she signs up for an **account**, she is then able to save the **listings** as well as contact the **listing agents** through her **inbox** for more information.

### Case 2. Registered user (buyer)

Zach is a 23-year-old startup CEO. He lives and breathes tech, and he recently received a massive endowment from an angel investor. He is looking to move out of his hacker house into a nice million-dollar home. He logs into HouseShop and checks on his **watchlist** to see if any **listings** have changed. Two houses have gone down in price, and one has sold. This gives Zach a fear of missing out, a feeling he experiences regularly. He decides to pursue a home. He sends a message to the seller's **inbox** listed and schedules a tour.

### Case 3. Registered user (seller)

Maggie is a 27-year old musician living in a condo in New York city. She recently signed a record deal with a big music company in Los Angeles and decided to sell her current place to move there. She visits the HouseShop website to request and schedule a listing consultant with an agent. She first signs up for an account to enter basic personal information including the current condition of her condo such as address, **square footage**, and **number of bedrooms**. She will be then contacted by an agent to discuss in

more details of how she can proceed selling her place through her **inbox**. Maggie sets the price based on her needs and estimation from the **listing agent**. Then, she hires professional photographers to take photos of her condo so she can upload **images**. Maggie can view how other people are viewing her place and what the agent is doing to sell it through her **dashboard**. Once the agent messages Maggie with different offers, she then decides and select which buyer to sell to and close the deal.

#### **Case 4. Listing agent**

Mark is a **listing agent** at HouseShop. Mark logs onto his **account** as a listing agent to check his **inbox** for any messages from interested buyers or sellers. He can also view the **listing(s)** he posted to track the popularity of those properties on his **dashboard**. Once a user contacted Mark, he can reply to consult the seller or negotiate with the buyer. He can use the information provided by the seller to create and post a new **listing**.

#### **Case 5. Admin**

Jimela is the site **admin** and logs on to check how the site is doing. Jimela cannot buy or sell as an admin. She has a few priorities to check on: user reports of two kinds: **listings** that are inappropriate or suspicious, and **accounts** that have been flagged by other users as abusive or fraudulent. Jimela references the Terms of Use to determine if a report is valid. She may do any of the following for each report: warn the user via their contact info that they have violated the Terms of Use, delete the **listing**, and/or remove the user account.

## Data Definition

---

**Admin:** The site administrator; Individual who may remove listings, delete user accounts, and perform general site management

**Browse:** the ability to view listings posted on the site

**Contact:** A datafield a Customer may acquire to contact a Seller or Listing Agent.

**Customer:** Any person with a registered account, with the following fields:

1. **Name:** The Customer's full name.
2. **Email:** The Customer's email address.
3. **Password:** The Customer's password.

A customer may be a

1. **Listing Agent:** A Customer who is a verified real estate agent. An agent may both post listings and assist other users in selling their own listings
2. **User:** a Customer who is buying or selling. A user is not a Listing Agent or an Admin

**Dashboard:** A listing agent's homepage. The Dashboard contains:

1. Notifications
2. Listings owned by the Listing Agent

**Filter:** a flag one may place on results while browsing to display listings in order of: price, zip code, number of bedrooms, number of bathrooms, and/or square footage.

**Guest:** a person browsing the site without a registered account

**Image:** A picture of a Listing, Listing Agent, or Customer.

**Inbox:** A list of messages a Customer may receive from other users or the admin.

**Lazy Registration:** the ability to browse listings without creating an account

**Listing:** an ad posting which includes a:

1. **Price:** The value posted for a Listing.
2. **Address:** The address of a Listing.
3. **Zipcode:** The zipcode of a Listing.
4. **Description:** A brief overview of details of the Listing.
5. **Date Posted:** The Year, Month, and Day a Listing was created.

A Listing may be either:

1. A house, which will include

- 1.1. **Number of Bedrooms:** the number of bedrooms in a listing
- 1.2. **Number of Bathrooms:** the number of bathrooms in a listing
- 1.3. **Square Footage:** the square footage of a listing
2. A plot of land, which only includes Square Footage

**Notifications:** an alert of recent activity relevant to a user's interests; includes:

1. new messages in a Customer's Inbox
2. changes to Listings saved to a Customer's Watchlist

**Watchlist:** An element of a user's dashboard where they may save listings to track if they change in price or are sold

# High Level List of Functional Requirements

---

## PRIORITY 1

### Guests:

1. Guests shall be able to navigate through all of our offered pages intuitively (excluding account information ).
2. Guests shall be able to create an account, login, and manage their account.
  - 2.1. To create an account user must type a few basic information, such as user name, password, email and security question. User will be able to login into this account using the information initially entered as well as password recovery.

### Both Guests and Customers:

3. Both registered and guest buyers shall be able to find sellers contact information.
  - 3.1. Once listing is clicked on, contact information of the seller will be easily visible.

### Registered Users:

4. Registered users shall be able to contact an agent if he/she wishes to sell their home.
  - 4.1. Registered users can request the ability to contact an agent if they navigate to their manage account options.
5. Registered users shall be able to access an inbox to receive and send messages.
  - 5.1. There will be an inbox button next to the account login/register that allows registered users to send and receive messages.

### Listing Agents:

6. Listing Agents shall be able to access a dashboard.
  - 6.1 This dashboard will be the Listing Agent's account management page and will provide all the functionality the Listing Agent will need, including: inbox management, activity on each listing, contact for listings etc.

### Admin:

7. Admin shall have the ability to see flagged listings that may be in violation to the terms and services.
  - 7.1 Admin will receive notifications when a user flags a listing, these notifications will appear as messages in the admins inbox.
8. Admin shall be able to access the terms and services to refer back to them.
  - 8.1. The ability to quickly view the terms of services for the admin will be accessible.
9. Admin shall have the ability to delete or remove the user's account of a listing that violates the terms of services.



9.1. The admin will see an option to remove a listing. There will also be an option to remove a user's account and delete it from the database.

## **PRIORITY 2**

### **Both Guests and Customers:**

10. Both guest and Customers shall be able to sort listings based on a variety of categories.
  - 10.1. Sorting listings based on various categories ie. price, square footage, bedrooms, location and so on.

### **Registered Users:**

11. Registered users shall be able to navigate through all of our offered pages intuitively.
  - 11.1. For example an easy, intuitive way to navigate from home page, to listings, to user's account etc.

### **Registered Buyers:**

12. Registered buyers shall be able to save homes they wish to view again later and access a list of saved homes.
  - 12.1. If the user is signed in, he/she can save listings that they want to refer back to later. These saved listings will be accessible at any time.

### **Registered Sellers:**

13. Registered sellers shall be able to upload pictures of their home they wish to sell to attach to their listing.

## List of non-functional requirements

---

1. Application shall be developed and deployed using class provided deployment stack
2. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks must be explicitly approved by Anthony Souza on a case by case basis.
3. Application shall be hosted and deployed on Amazon Web Services as specified in the class
4. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Chrome.
5. Application shall have responsive UI code so it can be adequately rendered on mobile devices but no mobile native app is to be developed
6. Data shall be stored in the MySQL database on the class server in the team's account
7. Application shall provide real-estate images and optionally video
8. Maps showing real-estate location shall be required
9. Application shall be deployed from the team's account on AWS
10. No more than 50 concurrent users shall be accessing the application at any time
11. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
12. The language used shall be English.
13. Application shall be very easy to use and intuitive. No prior training shall be required to use the website.
14. Google analytics shall be added
15. Messaging between users shall be done only by class approved methods and not via e-mail clients in order to avoid issues of security with e-mail services.
16. Pay functionality (how to pay for goods and services) shall not be implemented.
17. Site security: basic best practices shall be applied (as covered in the class)
18. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
19. The website shall prominently display the following text on all pages *"SFSU Software Engineering Project, Fall 2017. For Demonstration Only"*. (Important so as to not confuse this with a real application).

## Competitive Analysis

---

Feature	Zillow	Trulia	Realtor	Our Product
Text Search	+	+	+	+
Browse	+	+	+	+
Video/Picture Previews	++	+	+	+
Dashboard	+	+	+	+
Register/Login	+	+	+	+
Save option	+	+	+	+
Filter options (i.e., by price, # of bedrooms, etc.)	++	++	++	++

+Feature exists; ++Superior feature; - Does not exist

Our planned product will have all listed features as our fellow competitors do. We intend to provide a more intuitive experience to our user. This will include a flushed out browsing and sorting option that allow the users to easily navigate to their ideal home. The ability to navigate to sign in and save homes will be as simple as needed and extremely user friendly, and quick to get started. Zillow's filter system offers to sort by newest, cheapest, and then a drop down menu of eight more options, trulia offers the user to sort by price, bedrooms, home types, if its for sale or not and a mores tab. Realtors offers the user to sort through mostly the same as the other two but on top of those, bathrooms and amenities as well.

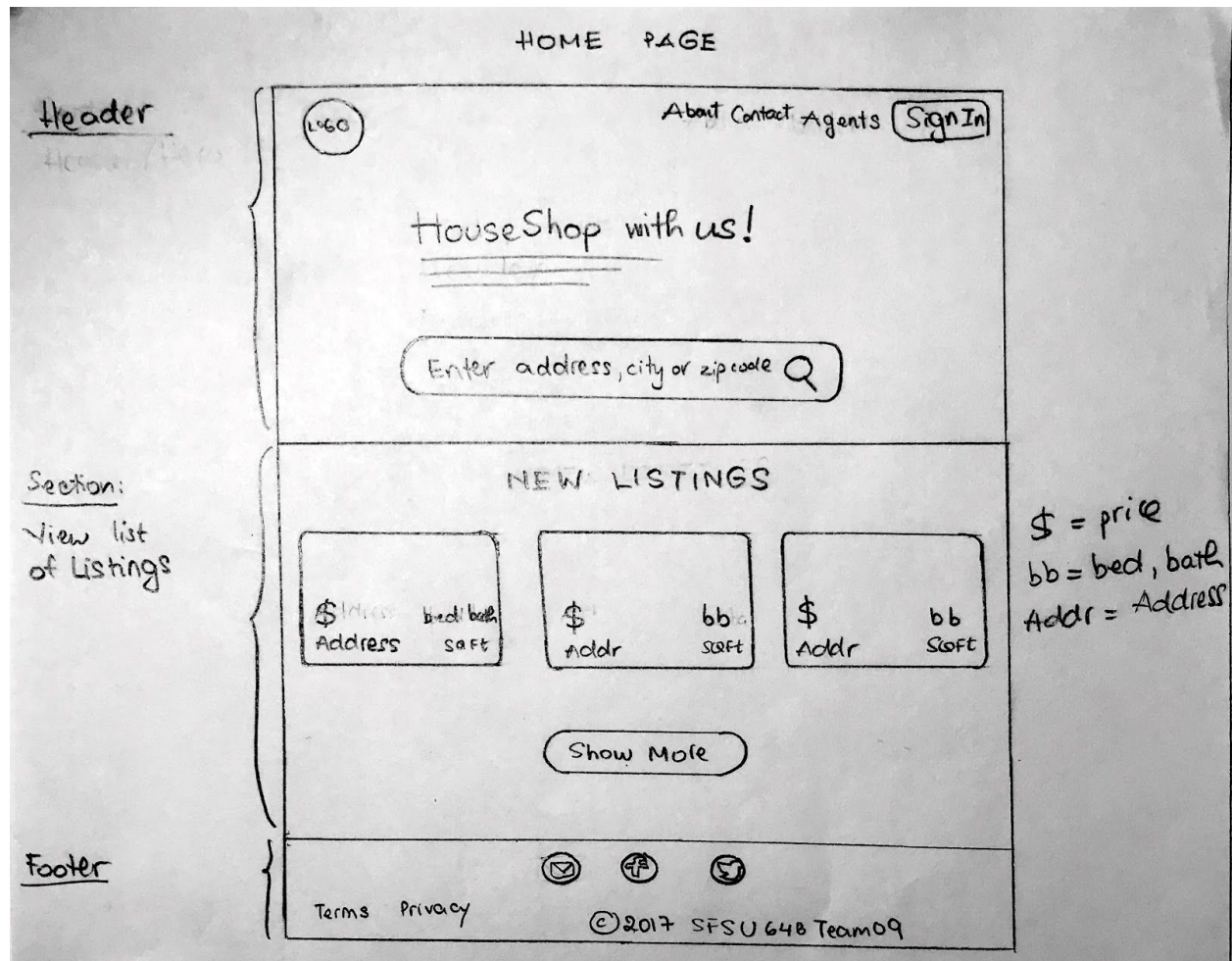
# UI Mockups and Storyboards

## HOME PAGE

Home page will have the search bar in the center of the page for users to search listings by querying attributes such as address, city, state, or zipcode.

There will also be a section below that briefly describes what HouseShop does and have a "Show All" button to display all the listings without specifying any attributes.

The navigation bar should be consistent throughout the website and include links to "About", "Contact", "Agent", and "Sign In" pages.

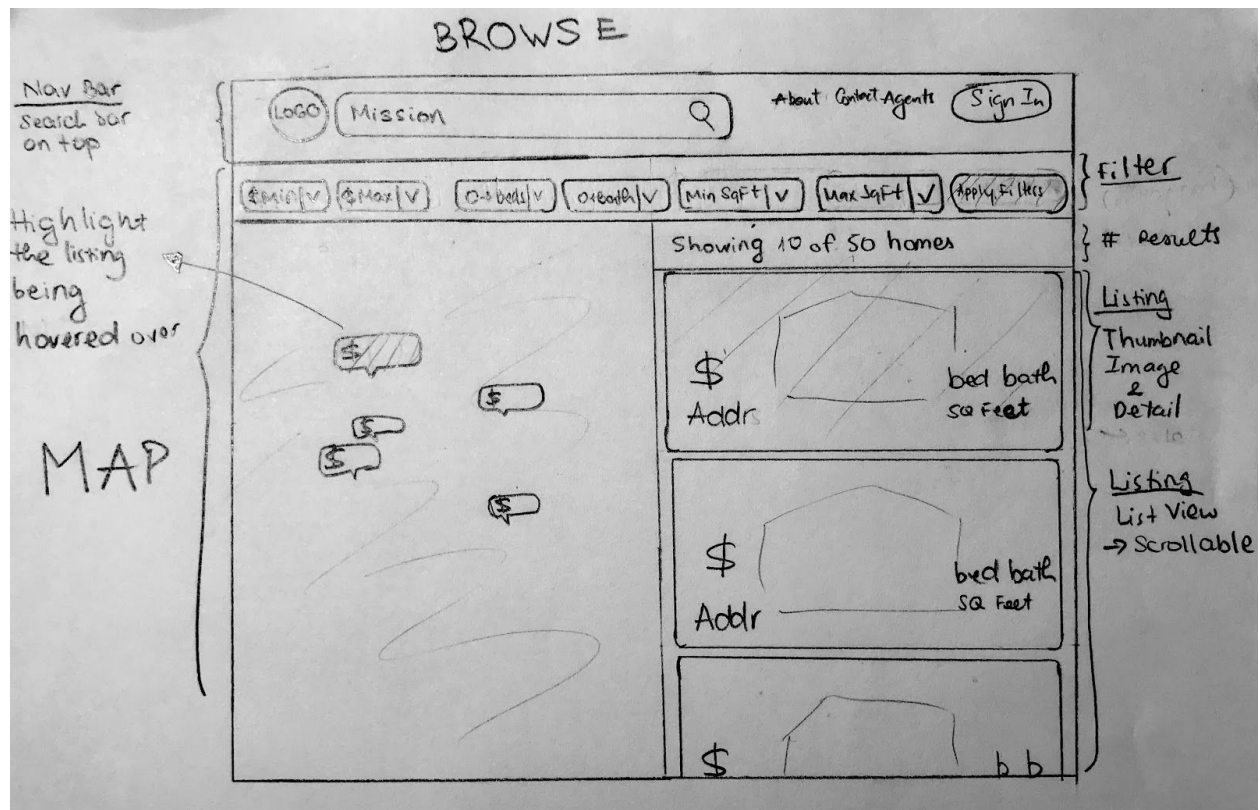


## BROWSING

Browse page will have the search bar fixed to the navigation bar for easy access.

Underneath, there will be a filter bar that users can select and apply different attributes such as number of bedrooms / bathrooms, price, and square footage to narrow down their listing query.

On the left side of the view, it should include a map which also pins the listing and its price. On the right side, it should show how many listing results match the queries and the listing thumbnails that match such queries.



## LISTING

Each listing will display images on the top right after the default navigation bar with search bar. Users can view images as they click on the right-arrow button which will display other images of the same listing.

The basic information section will display the address, price, features, and brief description of the listing. Also, on the right side, there will be a contact form to message Listing Agent easily if buyers are interested in learning or buying the property.

LISTING

nav bar

LOGO

Enter address, city, or zip code

Q

About Contact Agent!

Sign In

Images  
slideshow

>

BASIC  
INFO

Address

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\$ \_\_\_\_\_

Bed \_\_\_\_\_ Bath \_\_\_\_\_

Sq. Feet \_\_\_\_\_

Description

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

CONTACT AGENT

Name

\_\_\_\_\_

Phone

\_\_\_\_\_

Email

\_\_\_\_\_

Hello, [agent name],  
I am interested  
in [address]

CONTACT

Cancel

CONTACT  
FORM

auto  
populated  
from DB

SUGGESTION  
(priority 2-3)

SIMILAR LISTINGS

\$

Addr

BB

Sq Ft

\$

Addr

BB

Sq Ft

\$

Addr

BB

Sq Ft

MAP  
(priority 2-3)

Footer

Terms Privacy

© 2017

3



## SIGN IN/UP

The sign-in/sign-up form would be a pop-up on top of the current page. For the Sign-Up tab, there will be required text field for users to enter their information, security check such as Captcha, Terms of Services agreement checkbox, Sign-Up button, and Cancel option.

Similarly, for the Sign-In tab, users will also have to enter their username and password before they can click on the Sign-In button. Also, they can have the options to recover their password or Cancel signing in.

SIGN IN / UP

Sign Up →

WELCOME TO HOUSESHOP X

Sign In    Sign Up

(\*)

(\*)

(\*)

☒ ☒ ☒

☐ I agree to ...

SIGN UP

Cancel

Pop up modal for the sign in/up form

Sign In →

WELCOME TO HOUSESHOP X

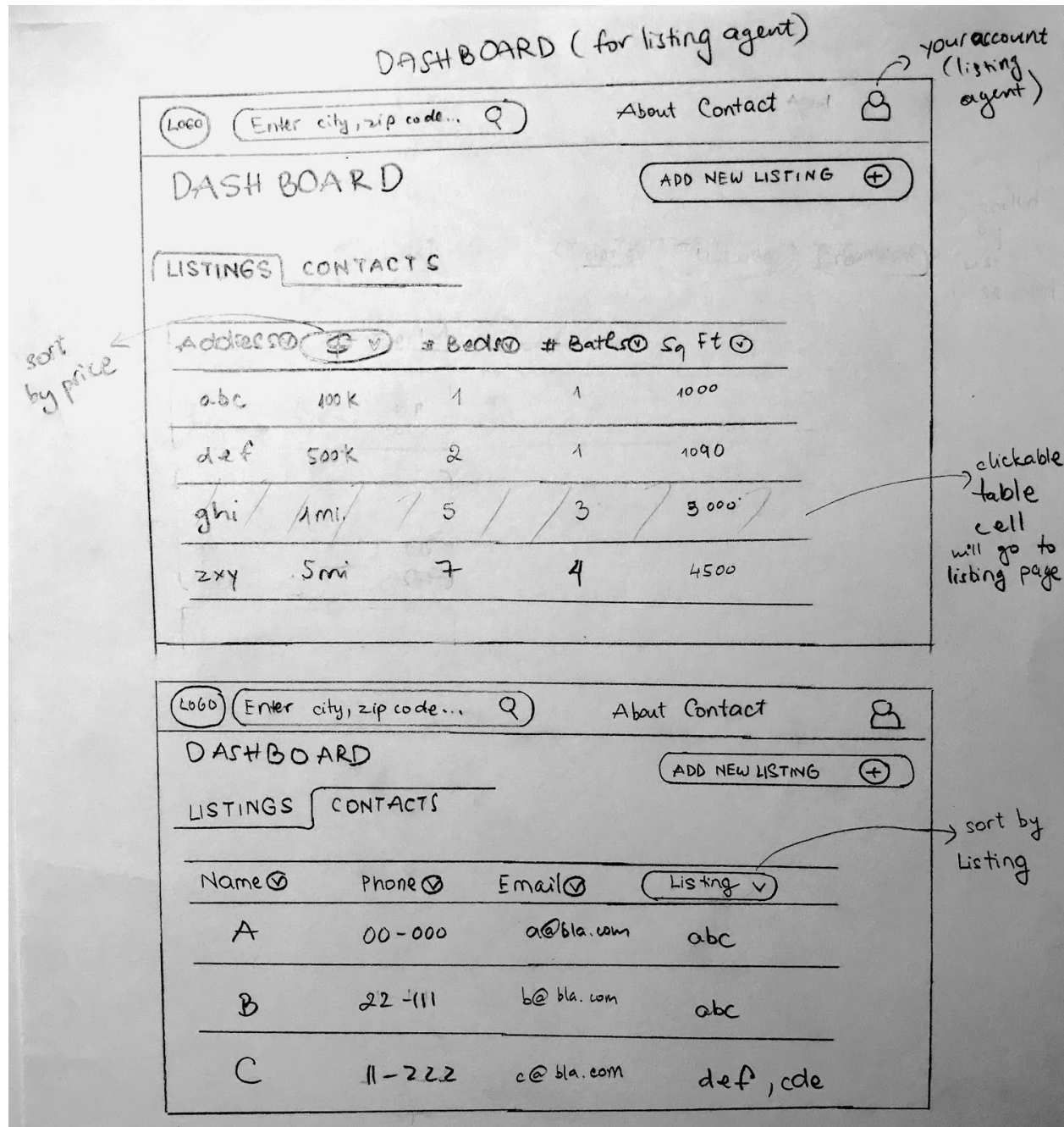
Sign In    Sign Up

SIGN IN

Cancel

## DASHBOARD (for Listing Agent)

The dashboard is designed specifically for the Listing Agent to track their listings for sale and contacts who show interests in their listings. There will be a button to Add New Listing and a table to display detailed information of listings and contacts. Listing Agents can also sort the table by specific features for easy grouping view.





## High-level system architecture

---

**HTTP server:** nginx (lightweight web server, also handles request routing)

**Front End layer:** Bootstrap, EJS (Javascript template engine for Node.js)

**Application layer:** Express (web framework for Node.js), Node.js (server-side Javascript runtime environment)

**Additional Node.js libraries:** mysql2 (MySQL client for Node.js), sequelize (object-relational mapping tool for Node.js)

**Node.js infrastructure:** npm (package manager for Node.js), PM2 (process manager for Node.js)

**Data layer:** MySQL (relational database management system)

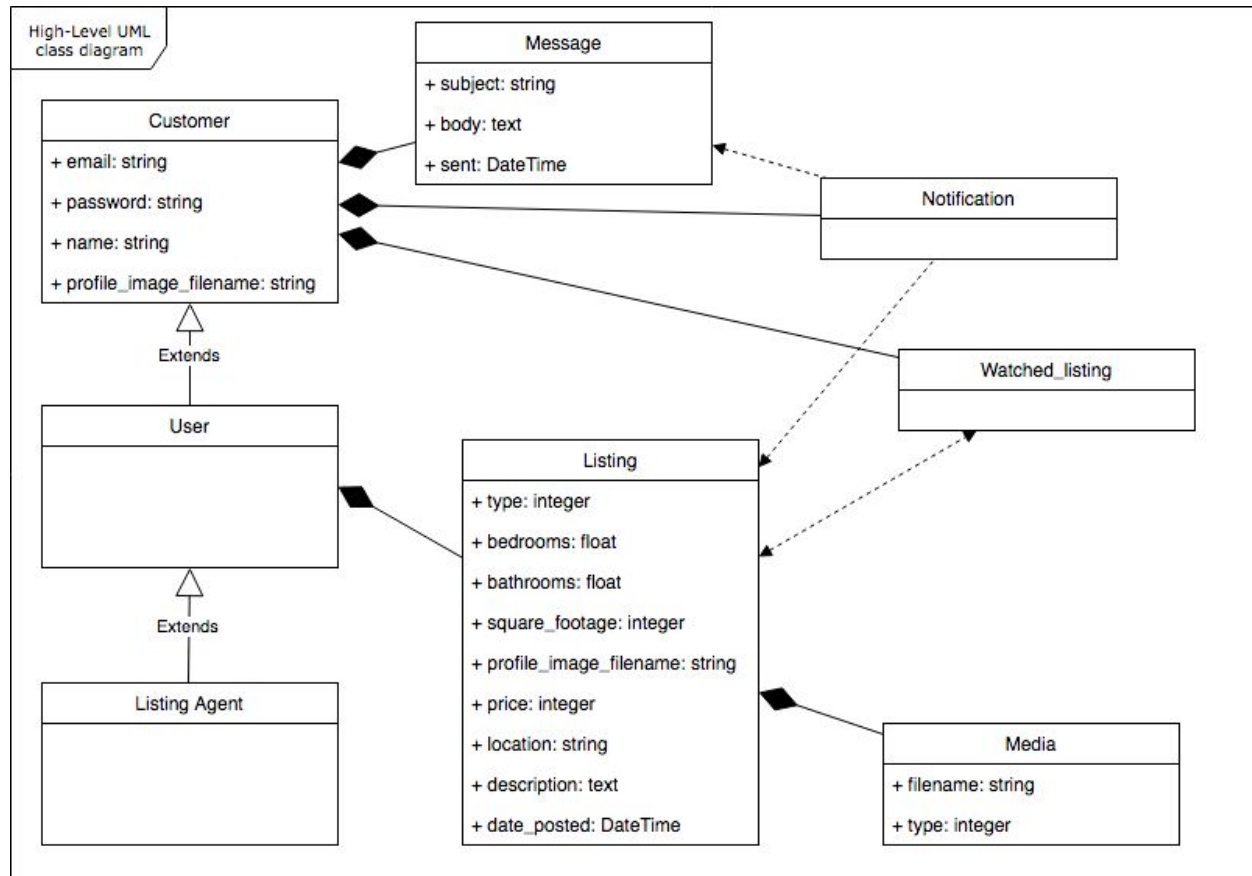
**Deployment platform:** Ubuntu (GNU/Linux distribution), GNU/Linux (Unix-like operating system), Google Compute Engine (virtual computing environment).

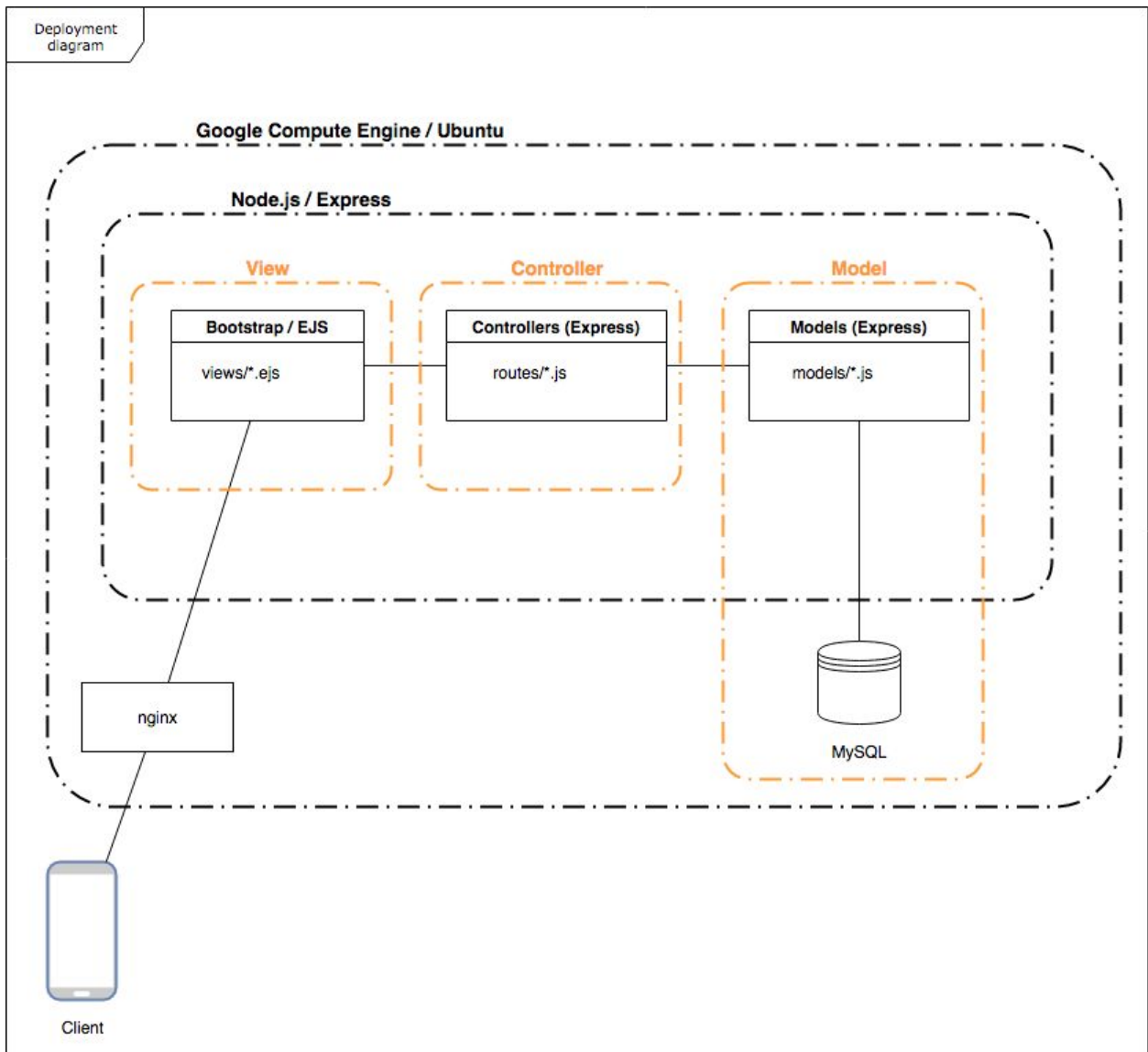
### **Supported browsers:**

Google Chrome (two latest versions)

Mozilla Firefox (two latest versions)

## High Level UML Diagrams





## Key Risks

---

### **Skill risks (do you have the right skills)**

May not be familiar with certain programs and front end / back end frameworks. This could drastically slow our progress and may jeopardise our team's ability to publish a complete product. For example we are using bootstrap for the front end, it will be essential to review the available documentation as we construct our site for best results. We will make sure we use all available resources, including our team members, if we have any questions or concerns or if need be the Chief Technology Officer. Using all available resources and keeping all team members up to date will be instrumental into overcoming these potential risks.

### **Schedule risks (can you make it given what you committed and the resources),**

So far we have rarely found times where every single member can meet aside from after the lecture period of the class on Mondays. The team frequently meets in smaller parts and keeps rest of team well informed of what happened during these meetings. There is nothing we can do to resolve this issue as scheduling between several students will naturally be very difficult. We have been successfully taking the required steps to minimize the damage this might cause. One way is to use slack to keep track of ideas and or progress that these smaller sub-division of team members make during any given meeting. It also helps that our front-end and back-ends tend to have schedules that tend to have aligned with each other thus far.

### **Technical risks (any technical unknowns to solve)**

There may be a risk of front-end and back-end having poor communication which may result to a lot of frustration and time consuming errors. Bootstrap, the front-end framework, may not communicate with the back end, express, framework. Making sure frontend and backend both communicate to work as one seamless website. Version controlling may cause merge errors or other issues especially merge conflicts or accidentally overriding someone's work. There also may be a risk of improperly pushing to the master branch with changes that may break the whole website. In order to solve some of these problems we have made a very clear protocol for how our version control will be handled within our team. Our team is very quick to respond to issues and questions regarding any of these risks which helps minimize error and every member has done their part to be as ready to help and clear with their changes as possible.

### **Teamwork risks (any issues related to teamwork);**

There is a risk of some team members not getting along with each other. As of right now there are no known teamwork risks.

**Legal/content risks (can you obtain content/SW you need legally with proper licensing, copyright).**

Users that use our site may wish to hold us liable for any event of misfortune. Running into legal issues if not in possession of the proper licensing is a risk factor that must be handled before launch. Ways we can solve these legal concerns is to obtain the proper licensing before the launch of our website. We need to make users agree to our terms and services under the condition that they have read and agree to it. This way we can avoid all accusations and blame for any misfortune. For the scope of this project we are assuming that all images uploaded are copyright-free and okay to use.

## Team 09

---

1. Franklin Henry Boswell, **Team Lead**
2. Khanh Nguyen, **Front-end Lead Developer**
3. Nemi McCarter-Ribakoff, **Backend Lead Developer**
4. Eugene Stukalov , **Backend Developer**
5. Jesse Christiansen, **Front-end Developer**
6. Kevin Lay, **Backend developer**

## Checklist

---

- Team decided on basic means of communications  
Done
- Team found a time slot to meet outside of the class  
Done
- Front and back end team leads chosen  
Done
- Github master chosen  
Done
- Team ready and able to use the chosen back and front end frameworks  
Done
- Skills of each team member defined and known to all  
Done
- Team lead ensured that all team members read the final M1 and agree/understand it before submission  
Done