Software Engineering CSC 648 Section 01 Fall 2017
Milestone 1

Team 14 (FriscoHousing)

John Lazzarini, Team Lead, jlazzari@mail.sfsu.edu
Jonah Manarang
Ivan Villamar
Eric Gumba
Prakash Gurung

10/01/2017	First version
10/03/2017	Revised after instructor's comment, frozen

1.Executive Summary

FriscoHousing is a real-estate browsing web application that facilitates the process of purchasing or advertising for a home. The current market for such applications are cluttered with overly ambitious start-ups and wayward has-beens; now is the perfect opportunity for a no-nonsense, what-you-see-is-what-you-get competitor to emerge onto the playing field.

Like Craigslist before it FriscoHousing will be a testament to the value of minimalism and purposeful design choices. A simple but intuitive user interface will enable buyers to browse through real-estate listings from a given area code. Realtors will be able to post properties to FriscoHousing's database by registering for a site-wide account.

Our simple premise is a great investment opportunity -- gone are the costs of unproductive innovation departments or expensive front end teams, which means more return on investment for our financiers. Maintenance of such a system will be inexpensive, considering there will be very little that a changing API can damage in a system with few dependencies. And just as our investors will benefit from our minimum-viable-product approach, so too will our customers. Fast page load and action response times will ease the activities of both buyers and sellers, and FriscoHousing's uncomplicated application architecture will be easy for an administrator to navigate and check.

FriscoHousing is being developed by a team of undergraduates from San Francisco State University -- their youth and inexperience will serve as a clever diversion to competitors, who will not expect their resourcefulness and ability to rapidly organize and prototype this application. FriscoHousing will appear from out of nowhere and disrupt the real estate market with a tried-and-true approach, leaving the big players scratching their heads and scrambling to regain their page visits.

2.Use Cases

1.Unregistered Home Buyer: Ralph is an **unregistered home buyer**. He is looking to buy a house but doesn't know where to start so he Googles house buying website and ends up on our website. When on our website Ralph he can browse the houses that have been posted, filter the houses by price, number of bedrooms and bathrooms, and can search for houses by using a zip code or address. However since Ralph is not a registered user he is unable to get the emails of realtors he would like to contact to get more information on a certain house. When Ralph tries to get contact information for a listing he likes he will be asked to register for an account. Ralph then creates an account using his email address and a password of his choice.

2.Registered Home Buyer: Jenna is a **registered home buyer** who has been on our website before. Jenna is able to do the things an unregistered home buyer can do but also has more perks. Jenna is able to see the contact information for the realtor of a house she is interested in. Jenna is also able to save houses she is interested in on her **favorites list** to keep for references later. She is also able to remove those houses from her personal list. Jenna is also able to store her information such as email and phone number on her profile.

3.Unregistered Realtor: Franklyn is an **unregistered realtor** who is looking to spread his listings on multiple websites. Franklyn is able to do things that an **unregistered home buyer** can do and that is all.

4.Registered Realtor: Sam is a **registered realtor** who has a couple of listings posted on our website. She is able to post **listings**, which will have an information on the address of the house, the price of the house, the amount of bedrooms and bathrooms, as well as a picture. Sam will also be able to edit her posted listings and delete her posted listings on her **relator dashboard**. She is also able to see which **registered home buyers** have saved her listings and look at their profile to get their contact information. Sam can also make a profile of her own which will have her contact information, all the houses she has listed on our website, and an about me area for things like websites and personal introductions.

5.System Administrator: Joel is the **system administrator** for our website. Joel is able to edit and remove a realtors post if it is inappropriate or spam. Joel is also able to ban any user who is posting spam and trolling on the website. Joel has access to things like usernames and passwords and can help recover accounts.

3.Data Definition

1. **Listing**: address, asking price, image, bedrooms, bathrooms, realtor
2. **Registered Home Buyer**: name, email, phone number, saved list, profile picture, username, password
3. **Registered Realtor**: name, email, phone number, website, profile picture, personal listings, username, password
4. **System Administrator**: list of all registered usernames and passwords, list of active listings
5. **Unregistered Home Buyer:** no stored data
6. **Unregistered Realtor:** no stored data
7. **Favorites List:** list of listings saved by a **registered home buyer**
8. **Realtor Dashboard:** list of listings created by a **registered realtor**

4.List of non-function 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, Safari, 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 cas to not confuse this with a real application).

5.List of functional requirements

1. **Users** shall be categorized according to **Unregistered Home Buyer, Registered Home Buyer, Unregistered Realtor, Registered Realtor,** and **System Administrator.**

Unregistered Homebuyers

2. **Unregistered Home Buyer** shall be able to browse the website.
3. All **Unregistered Home Buyer** shall be able to sort and filter the listing.
4. All **Unregistered Home Buyer** shall be able to signup and register for an account.

Registered Home Buyers

5. **Registered Home Buyers** shall be provided the same privileges as **Unregistered Home Buyers.**
6. **Registered Home Buyer** shall be able to see contact information for the realtors' listings.
7. **Registered Home Buyer** shall be able to contact the realtor.
8. **Registered Home Buyer** shall be able to contact the realtor via WWW site.
9. **Registered Home Buyer** shall not be able to contact the realtor using email.
10. **Registered Home Buyer** shall be able to save and remove houses from his/her personal list.
11. **Registered Home Buyer** shall be able to edit and store the personal information such as email and phone number.
12. **Registered Home Buyer** shall be provided a "dashboard" to manage contacts.

Unregistered Realtors

13. **Unregistered Realtor** shall be able to browse the houses that have been posted.
14. **Unregistered Realtor** shall be able to filter and sort the house listings.

Registered Realtors

15. **Registered Realtors** shall be provided the same privileges as **Unregistered Realtors.**
16. **Registered Realtor** shall be able to create post listings for house sales.
17. **Registered Realtor** shall be able to see which **Registered Home Buyer** have saved his/her listings.
18. **Registered Realtor** shall be able to contact the registered home buyers who have saved his/her listings.
19. **Registered Realtor** shall be provided a “dashboard” to manage sales and messaging.

System Administrator

20. **System Administrator** shall be able to edit and remove a realtor’s post.
21. **System Administrator** shall be able to ban any users.
22. **System Administrator** shall be able to access MySql WorkBench.

All Users

23. All **users** shall be able to browse the website without any login and registration.
24. All **users** shall be able to sort and filter the listing.
25. All **users** shall be able to signup and register for an account.

6.Competitive Analysis

++ = superior quality.
+ = contains feature.
- = does not contain feature.

Website	Search	Log-in	Filters	ease-of-navigation	mobile-first
Team 14	+	+	+	++	+
www.zillow.com	++	++	++	++	++
www.trulia.com	++	++	++	+	+
www.redfin.com	++	++	++	+	+
https://streeteasy.com/	+	+	+	-	-

Summary:
We plan on creating a minimum viable product for our website, which includes being able to register and login to the website. Below is a comparison of our planned features to that of other websites, such as zillow.com, trulia.com and so forth.

Login
Some websites contain features, which allows one to connect via social media applications like Facebook or google+. Our website will only have a basic login feature, where a user must enter their user ID and password to log in.

Mobile First
We plan on allowing our website to be easily navigable from a mobile browser thanks to frameworks such as bootstrap. Some of our competitors do not allow for ease of

navigation using mobile while other websites have not only developed from mobile-first principles, but also have their own mobile and iOS apps.

Ease of Navigation

Certain websites are cluttered with information, and also contain ads, which makes searching for houses slightly more confusing. Because our website does not contain as many features, our website should be easier to navigate than most other competitive websites.

Search

Some websites contain a search engine that lists the different housings on a map, and in different areas around the United States. Because our website focuses exclusively on San Francisco, we believe that it is not necessary to have search results for all the different areas of the world.

Filters

Some websites allow one to filter through the list of housing by price range, number of rooms, year built, size of the house and so forth. We believe that certain functions or “filters” will not be necessary for our final product.

7.High Level System Architecture

This section briefly describes the frameworks, APIs, tools and systems that will be used throughout the development of this project.

LNMN
Our application will be developed and deployed using a variant of the popular web platform LAMP software stack which consists of the following:

- Linux
    - Our application will be deployed on an Ubuntu Server, Version: 16.04
    - Hosted on Amazon Web Services (AWS), a cloud computing platform.
- NGINX
    - We will be serving our application through the NGINX web server to coordinate user requests for access to our application over the Internet.
- MySQL
    - The database management will be handled by MySQL to store and retrieve information about our users, listings, etc.
    - It will be used to easily display data when necessary.
- Node.js
    - Node.js is an event-driven, non-blocking I/O JavaScript runtime that allows many connections on our application to be handled concurrently.

Frameworks and APIs
In order to quickly and effectively develop our application, we will make use of the following frameworks to help in our development:

- Bootstrap
    - A web development framework that allows us to increase the speed of the development by providing a collection of templates and themes versus coding from scratch.
    - Supports responsive web design with its grid system thus making the application mobile-friendly.
    - Holds a big support community and is regularly maintained and updated.
- Express.js
    - Express is a popular lightweight web application Node.js framework that makes developing a node.js app much more convenient.

    - It saves developers a lot of time by providing a set of ready functions and modules as opposed to writing them from the start.
- Google Maps API
    - Provides us the capability to incorporate Google Maps into our application.
    - This will enable us to create a map view feature of nearby houses or the location of a specific real estate.

Supported Browsers

Our application will support and run with no problems in the following browsers:

- Google Chrome version: 61.0.3163
- Mozilla Firefox version: 56.0
- Microsoft Edge version: 40.15063
- Safari version: 11.0

Team

John Lazzarini, Team leader
Jonah Manarang, Back end lead
Ivan Villamar, Back end contributor
Eric Gumba, Front end lead
Prakash Gurung, Front end contributor

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**