SW Engineering CSC648/848 Section 01 Fall 2017
HOMIEZ
Team 15 – International team

Mena Morkos (*menamorkos@gmail.com*)
Andrew Patterson
Norald Alejo
Benedikt Anselment

Milestone 1
10/03/2017

Version control

Date	Version
10/03/2017	Milestone 1 submission

Summary of Content

1. Executive Summary

As many are aware, finding the right house and/or apartment is just a pain. Because of this, the sales are lower and the housing market is very limited. The need is an area that allows real estate agents to list residential housing and grants potential buyers the ability to seek this housing based on their preferences or needs.

The solution? **HOMIEZ**! HOMIEZ is a real estate website that brings agents and those who are seeking homes together in one place. This website is easy-to-use, hassle-free, and easy-to-manage. It allows real estate agents to list houses and apartments, including their features such as number of bedrooms, the type of flooring and many more. At the same time, HOMIEZ empowers potential home owners to easily search homes and/or apartments based on zip code, price, and housing features. It also enables potential home owners to interact with real estate agents as easy as never seen before.

Some key advantages of this website are the ability to find homes for sale in a map. It also allows users to filter houses and apartments based on their preferred location, housing features and/or housing prices. Furthermore, it not only enables users to message real estate agents directly, it also improves the way of scheduling an viewing appointment by its innovative calendar integration.

HOMIEZ will help to greatly increase the housing sales due to the improved matching of agents with buyers. Therefore, this gives people the opportunity to find the home of their dreams.

The HOMIEZ Company is a small student startup from San Francisco State University lead by Mena Morkos. We are a mix of frontend and backend developers that have a wide variety of background in Computer Science and Engineering. We produce for our clients engaging user-friendly web application, thus helping companies to enjoy a flying start by bringing their brand to the internet.

2. Use Cases

2.1 Unregistered User:

- Skill Level
  - Beginner

- Who they are
  - Prospective home buyer or renter
  - First time visiting the site / Has no enquiries yet

- User tasks
  - Views and browses listings
  - Looks for listings on maps
  - Filters listings by a range from a zip code, by price, by number of rooms, by size, by furniture…

- User Errors
  - Usage of wrong keywords

- Frequency and importance
  - When someone is looking for a place to rent/live he/she is going to use the site very often and frequently (probably once a day)

- Scenario
  - An Unregistered User visiting the website for the first time. He / she looks for housing in a certain zip code. The Unregistered User finds a house he / she likes to view. The Unregistered User presses the contact button, a window to login respectively register appears.

2.2 Registered User

- Skill Level
  - Beginner

- Who they are
  - Prospective home buyer or renter
  - Coming back to the site after registering

- User tasks
  - Views and browses listings
  - Looks for listings on maps
  - Filters listings by a range from a zip code, by price, by number of rooms, by size, by furniture…
  - Managing enquiries

- User Errors
  - Usage of wrong keywords

- Frequency and importance
  - When someone is looking for a place to rent/live he/she is going to use the site very often and frequently (probably once a day)

- Scenario
  - The Registered User manages its enquiries by answering to messages or by scheduling a viewing with the unique calendar integration of HOMIEZ.

2.3 Real Estate agent

- Skill Level
  - Low to average

- Who they are
  - Professional real estate agent

- User tasks
  - Uploads and manages listings
  - Manages customer relationship

- User Errors
  - Incorrect listings regarding contact info, address, image…

- Frequency and importance
  - Very often, it is part of their daily job

- Scenario
  - A real estate agent that logs in to post a new listing. The agent can enter details like address and area as well as uploading images. The agent goes to its already existing listings to check how many times they were seen. Afterwards, he / she goes to his / her inbox to check new messages and follows up on old ones. In the end he / she checks the calendar to see if there are new viewing appointments.

2.4 Administrator

- Skill Level
  - Skilled // can be trained if needed

- Who they are
  - Maintenance of website

- User tasks
  - Manages listings
  - Manages users
  - Updates website information

- User Errors
  - Incorrect deletes listing or accounts

- Frequency and importance
  - Administrator makes sure that the website is well maintained and up-to-date, it is their daily job

- Scenario
  - The Administrator receives a message that a Registered User needs their password reset, so he / she resets the password for the Registered User. Afterwards, the Administrator receives a message about an inappropriate listing. The Administrator deletes the listing and maybe deletes the Real Estate Agent that posted it.

3. Data Definition

- ACTORS
  - Unregistered User
    - An Unregistered User looking for housing *(see Use Case 2.1)*

  - Registered User
    - A Registered User managing its enquiries *(see Use Case 2.2)*

  - Users
    - Term used for referring to "Unregistered User" and "Registered User" at the same time

  - Real Estate Agent
    - A Real Estate Agent managing its listings *(see Use Case 2.3)*

  - Administrator
    - An Administrator making sure that the website is well maintained and up-to-date *(see Use Case 2.4)*

- LISTINGS
  - Apartment
    - A listing at the website. Element contains the following attributes:
      - size in square feet
      - address
      - number of bedrooms
      - number of bathrooms
      - kitchen
      - living room
      - furnished
      - parking possibilities
      - for rent
        - lease
        - security deposit
        - monthly rent
      - for sale
        - price
        - fees
      - image *(optional)*
        - maximum file size
      - video *(optional)*
        - maximum file size

  - House
    - A listing at the website. Element extends "Apartment" and contains the following additional attributes:
      - number of floors
      - size of property

4. Initial list of functional requirements

4.1 Backend

1. Searching
    *   Search will query the database for listings matching desired criteria

2. Logging in
    *   Registered Users attempting to login will query the database to check for their proper credentials. Once verified, they will be able to access their account.

3. Store/Upload/Delete home listings
    *   Real Estate Agents will be able to store and upload information regarding home listings on the sql database
    *   Real Estate Agents will be able to delete their listings after posting
    *   Administrators will be able to delete listings

4. Store/Upload/Delete user information
    *   Users and Real Estate Agents will be able to create new accounts (upload)
    *   Registered Users and Real Estate Agents will have their accounts stored on the sql-database
    *   Administrators will be able to delete user information

5. Resetting passwords
    *   Administrators will be able to reset Registered User passwords

6. Messaging
    *   Registered Users and Real Estate Agents will be able to speak to each other via messaging by having it be sent to the database and back

7. Password encryption
    *   Passwords created from new accounts will be encrypted and stored in the sql-database

4.2 Frontend

1. Browsing
    *   Users will be able to view and access listings

2. Filtering
    *   Users will be able to screen listings when browsing

3. Messaging
    *   Real Estate Agents and Registered Users will be able to communicate with each other through messaging

4. Google Maps Integration
    *   Users will be able to view listings on a map based on the address of listings

5. Login
    *   Registered Users and Real Estate Managers will be able to login and to access their dashboard

6. Dashboards
    *   Real Estate Managers can manage their listings and their messages
    *   Registered Users can manage their requests and their messages

5. 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, 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 as to not confuse this with a real application).

6. Competitive analysis

	Map search	Browse	Viewing scheduling
HOMIEZ	++	++	++
REALTOR	-	++	-
HOTPADS	++	+	-
ZILLOW	++	+	+

++ market leadership; + feature available; - feature not available

Our USP is the combination of the advantages of our competitors. Whereas REALTOR is excelling in browsing, HOTPADS and ZILLOW both offer a convenient map-searching option. None of our competitors however, offers both at the same time. Moreover, REALTOR and HOTPADS only allow to contact the real estate agent. ZILLOW offers basic online viewing scheduling. OURPRODUCT however, will allow to schedule a tour without the annoying waiting for conformation by calendar integration.

7. High-level system architecture
7.1 The Code
Node

Node.js is a popular server side framework that uses javascript as its main language. Thanks to the flexibility of the javascript language, efficiency of chrome's v8 engine, and node's easy to use package management system, it is an extremely reliable and strong choice for our back end.

Express

Express is a mature and well supported Node.js framework which enables us to create apps and API's from the ground up. This framework is much more minimalistic than its peers (Hapi.js, for example) and requires more code to be written and tested. Although it does take more code, the community is much larger and can be used as an extremely helpful resource. In the end, we chose Express because it's well supported, has a large community, and is a mature framework.

https://expressjs.com/

Handlebars

Handlebars is a templating engine that looks like regular HTML, but enables us to embed handlebars expressions within it. Thanks to its extreme similarity to plain HTML and JavaScript it's very easy to learn and pick up for beginners, since it was created using all the community favourite parts of "Mustache", which is a templating engine it was based off. Handlebars separates generating HTML from the raw JS code, to ensure that the code remains as readable as possible. Not many of our team members have much experience with front-end, so ensuring the framework we use is easy to pickup was an extremely important deciding factor.

http://handlebarsjs.com/

Bootstrap

Bootstrap is a frontend framework with built in CSS and JavaScript code, which enable an easy to implement mobile first and responsive functionality with very little effort. Instead of having to create our own classes from scratch, from a navbar, search bar, and so on, it's included with bootstrap. In our efforts to make everyone's job as easy as possible, bootstrap decreases the burden of CSS knowledge from our front-end team.

http://getbootstrap.com/

<u>Less</u>

Less is a CSS Pre-processor that extends the CSS language to allow for variables, functions, mix-ins, and more. As a result, Less enables to us write clean and sustainable code.

http://lesscss.org/

http://getbootstrap.com/2.0.4/less.html

7.2 The APIs
Google Maps
The google maps API allows us to display and track locations of homes, as well as providing us an easy to implement platform to display location data.

https://developers.google.com/maps

Google Analytics
Google analytics is a free analytics service that stores website traffic. Google analytics is one of the most popular analytics platforms on the web due to its ease of use and high efficiency.

https://analytics.google.com/analytics/

7.3 The Tools
WebStorm
WebStorm will be the main IDE we use, as it includes "intelligent code completion, on-the-fly error detection, powerful navigation, and refactoring for JavaScript and stylesheet languages." Furthermore, everyone in the group has had to use JetBrains products for either CS413, CS667 or other serious projects. This already-had familiarity makes WebStorm the obvious developer environment for our group.

https://www.jetbrains.com/webstorm/

Sublime
Sublime is an extremely light text editor which will be used when making small edits and reads of files. Nearly matching the speed of built in text editors, it will be extremely useful in providing the team with easy file access without having to fully launch a large and bloated IDE like WebStorm.

https://www.sublimetext.com/3

MySQL Workbench
MySQL workbench enables us to access and view our database through a GUI, increasing productivity, accuracy, and sanity.

https://www.mysql.com/products/workbench/

Nginx
Nginx is an HTTP server and reverse proxy. Due to its stability, simple configuration, and high performance, it's an excellent choice to use Nginx with our servers to run our web application.

https://www.nginx.com/resources/wiki/

Github
Github is a version control platform, that uses git, the version control software. We chose this as our platform for version control due to every group member having familiarity in it.

https://github.com/

SSL
Https is crucial for guaranteeing security standards, and will be provided by a free license on let's encrypt.

https://letsencrypt.org/

7.4 Requirements for Supported Browsers

Every browser has their quirks, and with that, what they choose to and not support. To build

our application, these are the minimum requirements needed from our supported browsers.

CSS:
- All CSS Features

HTML 5:
- All HTML 5 Features

SVG:
- SVG (Basic support)

JS API:
- All JS API Features

JavaScript:
- All JS Features (Excluding Object.observe data binding)

http://caniuse.com/#index

7.5 Guaranteed Supported Browsers

Chrome
Versions supported:
- 60
- 61

Mozilla
Versions supported:
- 54
- 55

Safari
Versions supported:
- 10.1
- 11

8. Team

Mena Morkos   *- Team lead*

Andrew Patterson   *- Backend lead*

Norald Alejo   *- Frontend lead*

Benedikt Anselment   *- Frontend team member*

9. Checklist

- Team decided on basic means of communications
  - DONE

- Team found a time slot to meet outside of the class
  - ON TRACK

- Front and back end team leads chosen
  - DONE

- Github master chosen
  - DONE

- Team ready and able to use the chosen back and frontend frameworks
  - ON TRACK

- 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