

**SW Engineering CSC648/848 Section 01 Fall 2017  
NAME APPLICATION  
Team 15**

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

Milestone 1  
10/XX/2017  
History Table

**Summary of Content**

1. **Executive Summary** **3**
2. **Use Cases** **4**
3. **Data Definition** **6**
4. **Initial list of functional requirements** **7**
5. **List of non-functional requirements** **8**
6. **Competitive analysis** **9**
7. **High-level system architecture** **10**
8. **Team** **11**
9. **Checklist** **11**

**1. Executive Summary**

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

The solution? [Product-Name]. [Product-Name] is a real estate website that brings real estate agents 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 its many features such as number of bedrooms, the type of flooring and many more. At the same time, it empowers potential home owners to easily search homes and/or apartments based on zip code, house price, and housing features. It also enables potential home owners to interact with real estate agents, bringing the seller and the buyer together.

Some key advantages of this website is 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. It enables users to message real estate agents directly.

This will greatly increase the housing market sales because of the increase in buyers. Not only that, this also gives people the opportunity to find the home of their dreams.

[Team-Name or Company] is a small student startup team from San Francisco State University lead by Mena Morkos. We are a mix of front-end and back-end developers that have a wide variety of background in Computer Science and Engineering. We produce for our clients engaging user-friendly web application. We aim to bring to life to companies by bringing their brand to the internet.

**2. Use Cases**

XXXX

**3. Data Definition**

* ACTORS
  + unreguser
    - unregistered user like described in section “use cases”
  + reguser
    - registered user like described in section “use cases”
  + agent
    - agent like described in section “use cases”
  + administrator
    - administrator like descriped in section “use cases”
* ITEMS
  + (room)?
  + apartment
    - A listing at the website. Element contains the following attributes:
      * size
      * adress
      * numberofbedrooms
      * numberofbathrooms
      * kitchen
      * livingroom
      * furnished
      * parkingpossibilities
      * forrent
        + lease
        + securitydeposit
        + monthlyrent
      * forsale
        + price
        + fees
  + house
    - A listing at the website. Element extends “apartment” and contains the following additional attributes
      * numberoffloors
      * sizeofproperty

**4. Initial list of functional requirements**

XXXX

**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 | Automated broker? |
| **OURPRODUCT** | ++ | ++ | ++ | + |
| **REALTOR** | - | ++ | - | - |
| **HOTPADS** | ++ | + | - | - |
| **ZILLOW** | ++ | + | + | - |

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**

**The Code**

**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 favorite 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 front end 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/>

**Less**

Less is a CSS Preprocessor 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>

**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.” Futhermore, everyone in the group has had to use JetBrains products for either 413, 667, and 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>

**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>

**Guaranteed Supported Browsers**

**Chrome**

Versions supported:

* 59
* 60
* 61

**Mozilla**

Versions supported:

* 54
* 55

**Opera**

Versions supported:

* 47

**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
  + ON TRACK
* 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
  + ON TRACK