

SW Engineering CSC 648/848 Section 02 Fall 2017

Dream Home

Team 05

Milestone 1:

Use Cases, High Level Requirements, and Architecture

- Sophie Tait (stait@mail.sfsu.edu)
- Bravolly Pich
- James Hinds
- Supritha Amudhu
- Saengduean (Em) Calderaz
- Brendan Kelly
- Steve Cardenas

Date	Description
10/03/2017	created first draft of M1 for Professor Petkovic to review
10/04/2017	draft update
10/17/2017	revised, frozen

1. Executive Summary	2
3. Data Definition	5
4. Initial List of Functional Requirements	6
Website	6
Unregistered User	6
Registered User	6
Seller	6
Administrator	7
5. List of Non-Functional Requirements	8
6. Competitive Analysis	9
7. High-Level System Architecture	10
Platform	10
Frameworks	10
APIs	10
Tools	10
Supported Browsers	10
8. Team	11
9. Checklist	12

1. Executive Summary

Dream Home is a real estate website designed for individuals looking for their new dream home and real estate agents selling potential dream homes. Users will be able to browse, search, or filter for their potential new home. They can also contact the real estate agent directly from our website and manage contacted agents with the Dream Home Dashboard. Real estate agents will be able to post new listings for sale, manage their listings with their own dashboard, and communicate directly with buyers from their Seller dashboard.

Dream Home was created with usability in mind. We understand that searching online for a new home can be stressful and confusing so we've chosen to keep features to a minimum and straightforward. Users won't be bombarded with an overwhelming number of choices, just a few simple options. Dream Home also helps buyers keep track of all the real estate agents they have contacted with the Dream Home Dashboard. Users don't need to worry about losing contact information or forgetting which homes they were interested in since all their potential dream home information is stored on one easy-to-use website.

Real estate agents will also love their Seller Dashboard since it stores all potential buyers for each listing. Agents can message buyers directly from their Seller dashboard, so they can assure they are contacting the correct buyer about the correct home.

Dream Home will be competitive with other real estate websites since it will contain common features of other real estate websites, but Dream Home will be more user friendly. By eliminating unnecessary features that other real estate websites have, Dream Home will be easier for users to search for homes and contact sellers.

Our team of student software engineers is dedicated to creating a high quality real estate website for our users. Some of our team members have industry experience, but all of us have the passion and hard work it takes to make Dream Home the perfect place to find your new dream home.

2. Use Cases

1. **Unregistered User** – An unregistered user, Harry, wants to search for houses to purchase. He visits Dream Home, lands on the Home Page, and views a list of featured houses and a few advertisements in the Home Page. He has a specific location on his mind, so he types the zip code for that location in the search bar. He is able to view a listing of available houses. He tries to sort them in order of increasing price because he wants a smaller house. He filters the search results for one bed, one bath and tries to decide on a house. He is not sure of which one to buy, so he searches for more houses. Harry clicks on one house that interests him and views various images of the house. He finds a video that covers the interiors and exteriors of the house. He tries to change the details of the house, but he isn't able to edit the page contents. He tries to contact the seller for the house, but the application prompts him to register. So he enters his personal details and registers with the website. He then goes ahead and contacts the seller for the house.
2. **Registered User** – A registered user Ronald has decided that he wants to buy a house. He goes to the Home Page. He searches for a city name in the search bar and gets a series of listings displayed related to his search. He scrolls down and tries to decide on a house. He filters the listings based on the amenities for the house because he wants a two bed, two bath apartment. Based on the filtered listing, he clicks on one of the houses and sees images and a video about the house. He tries to change the details of the house, but he isn't able to edit the page contents. He reviews the details of the house and then contacts the seller for the house. He navigates back to the list page and clicks on one more house. He contacts the seller for the second house as well to get details about it. He then goes to the User Dashboard to view the list of houses for which he's contacted the sellers for. He logs out of the application.
3. **Seller** – A seller Hermione wants to post data about her house so that she can sell it to someone. She logs into her account where she already has her contact details stored. She clicks on a button to add a new listing. She then enters details of the house, adds a few images for the house. She enters a price for the house. Finally, she saves the details of the house and it gets displayed on her Seller Dashboard. In that page, she is also able to see a list of two other houses that she has already posted for sale. She logs out of her account in the website.
4. **Administrator** – The Administrator Dumbledore manages the data displayed and the list of users and sellers in the website. Dumbledore has access to the database of the application. He has access to the details of the houses listed by Sellers. He views one of the house's details and finds

that there is a duplicate listing and deletes the duplicate listing from the database. He then sees that there is inappropriate content posted about one other house. He removes the listing and bans the user for inappropriate content. He cannot alter the details on listings; he can only delete listings from the database and block users.

3. Data Definition

- **Unregistered User:** A user who does not have an account or is not currently logged in. They use the website to browse listings and to create an account.
- **Registered User:** A user who has an account and is currently logged in. They use the website to browse listings and contact sellers.
- **Administrator:** A user who has elevated privileges including access to the website database. They use the website to perform administrative tasks and to moderate listings and users.
- **Seller:** A user who is a real estate agent. They have an account and are logged in. They use the website to browse listings, post listings, and receive messages from registered users.
- **Listing:** The entity that represents a property for sale. Listings contain information on location, property amenities, number of bedrooms, number of bathrooms, price, and seller. Listings are posted by sellers and available to users. Listings may be moderated at the discretion of the administrator.
- **User Dashboard:** A page that includes all listings that the user has contacted the seller about.
- **Seller Dashboard:** A page that includes all listings that the user has posted. Seller Dashboard contains all users that have contacted the seller about a particular listing.

4. Initial List of Functional Requirements

1. Website

- a. Shall be able to browse listings
- b. Shall be able to search listings by address, city, or zipcode
- c. Shall display a house listing
 - i. Shall display the address
 - ii. Shall display the price
 - iii. Shall display images of the house
 - iv. Shall be able to have a description of the house
 - v. Shall display the location of the house in Google Maps

2. Unregistered User

- a. Shall be able to browse through the listings
- b. Shall be able to view house listing details
- c. Shall be able to search listing
- d. Shall be able to sort listings
- e. Shall be able to register an account

3. Registered User

- a. Shall be able to do a-d of Unregistered User
- b. Shall be able to login to their account
- c. Shall be able to log out of their account
- d. Shall be able to contact seller
- e. Shall be able to view their own Registered User Dashboard
 - i. Shall show sellers that they have been in contact with
 - 1. Shall be able to contact these sellers
 - ii. Shall show house listing that they are interested in
 - 1. Shall be able to view these house listings
 - iii. Shall be able to remove house listing that they are no longer interested in

4. Seller

- a. Shall be able to do a-d of Unregistered User
- b. Shall be able to login to their account
- c. Shall be able to log out of their account
- d. Shall be able to view their own Seller Dashboard
 - i. Shall show their own listings of houses that they had created

1. Shall be able to add a house listing
 2. Shall be able to remove their house listing
 3. Shall be able to edit their house listing
- ii. Shall be able to view Registered Users that had contacted them
 1. Shall be able to contact these Registered Users

5. Administrator

- a. Shall be able to access the database
- b. Shall be able to remove a house listing
- c. Shall be able to ban an account for inappropriate use
- d. Shall be able to view website analytics

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

	Search by address (city, zip)	Filter (Price, # of bedrooms, etc)	Registered users (username/password) w/ ability to contact seller	Registered sellers w/ ability to add and edit listings	Integration of google maps
Dream Home	+	+	+	+	++
Zillow	+	+	+	+	+(uses bing)
Trulia	+	+	+	+	++
Realtor	+	+	+	+	++

Dream Home aims to be competitive with the rest of the top real estate websites. However, We want to offer all of these features with a simple user interface that strives to be easier to use. With these improvements to the user interface we hope to gain customers by reducing the time spent on the website and making it more user friendly.

7. High-Level System Architecture

Platform

- Ubuntu 16.04.3 (Linux 4.10.0-35-generic x86_64)
- Google Cloud Platform
- NGINX Web Server 1.12.1

Frameworks

- Bootstrap 3.3.7
- jQuery 1.7.4
- NodeJS 8.4.0
- ExpressJS 4.15.4
- npm 3.5.2
- pm2 2.6.1
- Git 2.7.4
- MySQL 5.7.19

APIs

- Google Analytics
- Google Maps

Tools

- GitHub
- MySQL WorkBench

Supported Browsers

- Mozilla Firefox 55.0.3 and 56.0
- Chrome 60.0.3112 and 61.0.3163

8. Team

- Sophie Tait: team lead
- Bravolly Pich: front-end lead & GitHub master
- James Hinds: back-end lead
- Supritha Amudhu: back-end
- Saengduean (Em) Calderaz: back-end
- Brendan Kelly: front-end
- Steve Cardenas: front-end

9. 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
- Team lead ensured that all team members read the final M1 and agree/understand it before submission: DONE