

SW Engineering CSC648/848 Section 02 Fall 2017

Real Estate Website – Augment Realty

Team number: 08

Falco Becker falcobecker@freenet.de
Richard Robinson

Vidit Manglani
Karan Gupta

Milestone 1

October 04, 2017

History table

| DATE | REVISIONS AND FEEDBACK | PAGE NO. |
|------|------------------------|----------|
| | | |

Executive Summary

Have you ever visited a real estate website and ended up more confused than when you started. Does your current real estate portal try to do much and yet achieve nothing at all?

Augment realty

webportal [ôg'ment | rē(ə)ltē] make (real estate interaction) greater by adding to it; increase the probability of earnest interactions: *the real estate experience was augmented by effectively communicating with every right client.*

Used in contrast to deeply complicated interfaces that offer too much and do to little

'Augment Realty' A Real Estate Marketplace designed from the ground up to facilitate effective communication between buyer and seller.

Stands apart from its competitors by only delving in the important things, and doing them well. A simple interface that comforts the user, familiar like a fireplace. Yet detailed enough to be customizable to be the correct tool box for the job.

Offering only the essentials, search, see, compare, contact but in seamless and synergized interface. Appealing to essential ends of the webportal, home buyers and property agents. We recognize the differing requirements of the our two clients, neither of whom we can exist without, and we also recognize the ways in which the resemble each other, the points their interests intersect with the requirements of the real estate marketplace and we make sure do those well, and most importantly implement the differences in design using subtle changes which require little cost to implement and provide the required distinctions.

With an interface approved by real estate professionals in the United States and the Database designed by a team in Germany.. It enables the buyer to hone in on exactly what it is they are looking for and for the seller to work with a buyer, not a browser. Fluid in all the right places, with a modular design that is ready to change with the times and yet since it is built on the sound logic of the that is minimal and yet scalable

With a homepage that encourages buy-in from our *three* prospective clients: buyers, sellers and agents by displaying features that are inherently attractive to each sect of the differing perspectives and circumstances of our market(audience)

A specialised interface for each of the three clients that will consume our product.

Property buyers and shown a interface that suits their outlook—

Why is it seductive to buyers

It gives them the information they need, it helps them imagine their *future home*

What about it does the prospective seller feel that it will make their job easier

Does an organized interface appeal to single home sellers too? (multiple home sellers agents are just single home sellers with multiple homes to sell)

How is it profitable

It works

It allows for future/further implementation of monetization

It charges sellers for completed sales. And encourages them to complete sales.

Why is it unique

The messaging system ensures integrity of users

. An Identifiable Goal

USE CASES FOR MILSTONE ONE

Use Case Nr. 1: unregistered user searches

A unregistered user, which could be a future customer, visits the web site with the intention to find a proper property. He/she has a imagination which characteristics the property shall have.

At first he/she wants to search for properties in certain locations. So the unregistered user can search with a offered search-function on the index-site for either a proper name of a city or a zip code.

With this information the system can show every registered property-offer out of the database.

The system also offers a bunch of filter-options, such as filter by price-scale, distance and amount of bed/bath-rooms.

If the unregistered user is interested in an offer he/she can select it and will be leaded to a site with all the information needed, like google-maps location, price, bedrooms, bathrooms, owner, description and pricing.

Use Case Nr. 2: registered user wants to buy a bookmarked house

After a registered user found a proper offer with a suitable price he/she wants to buy it.

At first the user needs to log in on the index-site with username and password. After this the user will be led to his own user-information-dashboard with the bookmarked houses.

Those bookmarks can also be filtered by price-scale, distance and amount of bed/bath-rooms. After the wanted offer was found he/she can select it and will be led to the profile page of the house where the user can start the buying-transaction.

Use Case Nr. 3: owner offers house

If a owner of a property wants to sell his/her house on the website he/she needs to sign up at first. After the sign-up where the owner needs to give personal information the owner will be led to his/her dashboard.

This dashboard shall display all offered properties and incoming requests and buying requests.

For getting the new offer online, he/she needs to create a new offer. Here the owner needs to describe the property correctly with uploaded pictures, maybe a video-tour,

short description, price, location and the number of rooms. After the owner confirmed the information the offer can be uploaded to the database.

Use Case Nr. 4: real estate agent wants to sell

If a real estate agent wants to sell a property he/she needs to log in at first. After the log in on the index-site the real estate agent will be led to a proper dashboard. This shall display a history of all transactions, active offers and incoming requests.

The real estate agent can select the request which he/she is aiming to accept and start the selling-transaction.

Data Definition -

define main terms, data structures and “items” or “entities” at high or logical (not implementation) level (e.g. name, meaning, usage, and NOT how the data is stored in memory) so it is easier to refer to them in the document. Focus on key terms (main data elements, actors, types of users etc.) specific for this application and not on general, well know terms. These terms and their names must be used consistently from then on in all documents, user interface, in naming SE components and database elements etc. In later milestones you will add more implementation details for each item. You will later expand this section with more details

User

-Any entity that accesses the website

Buyer

-Any entity that is interested in purchasing a property

Unregistered Buyer: A buyer who has not registered with the site

Registered Buyer: A buyer who has registered with the web site

Seller

-Any entity that is interested in selling a property

Property

-A house that has been listed

Clients/Customers

Single HomeBuyer: Looking to *buy* one home

Multiple HouseBuyer: Looking to *invest* in houses

Multiple HouseSeller: *Agents*. Looking to manage sales.

Interface Routes

unregisteredUser: The route that convinces buyers to register

registeredBuyer: to enable buyers to search and bookmark properties

registeredSeller: to enable sellers to manage their listing(s)

Pages

Homepage: The page an unregistered user sees on opening the website. Once

Registered links to the Browse Page

Dashboard: The page reg'd buyers and sellers have access to where they can see their properties. *BuyerDash SellerDash*

Properties Page

BuyerPropertiesPage, SellerPropertiesPage

Property Page

BuyerPropertyPage, SellerPropertyPage

PropertyName:

PropertyAddress:

PropertyLatLong:

Elements

MessageBox:

SearchBox

BrowseArea

Initial list of functional requirements

– see class notes. This refers to high level functions you plan to develop to the best of your knowledge at this point. Focus on WHAT and not HOW. Keep the user in mind. Develop these functions to be consistent with use cases and requirements above. Number each requirement with unique numeric value and use these numbers consistently from then on. For each functional requirement use 1-3 line description

1. **Login System**
2. **Multiple property management**
3. **Dashboard Page:** Registered buyers and sellers shall be able to organize their properties and respective communications on a single page.
4. **Featured Properties:** A page area that shall display interesting properties on the Homepage, selected by a criteria such as popularity, and/or for paying registered sellers.
5. **Search Function:** Shall be implemented, in minimum, as a text based way to query the property database.
6. **Browse Page:** A place for buyers, and a separate place for sellers, to see related properties using keywords, categories or schema.
7. **Messaging System:** A way for buyers to express interest and inquire about certain properties and for sellers to manage each communication, even ones from unregistered users shall be integrated into the interface.
8. **Location on Map:** A map shall be visible on every property's opening page. With links to walking score, stores nearby and relevant tie-ins.

High-level non-functional specifications

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).

Competitive Analysis

Zillow:

- Buy
- Rent
- Sell
- Mortgage
- Find an Agent
- Home Design
- Zillow blogs and forums

Advantages over others:

- mildly animated, relaxing picture on main page
- sign in and join buttons are separate, more clarity
- forums for the particular website
- Design ideas
- Can find good school districts
- Word choice is amazing, for example zillow-specific terms that are catchy, and viewing a home is called a "walkthrough"

Trulia:

- Buy
- Rent
- Mortgage
- Sell
- News

Advantages Over Others:

- +Saved searches and homes is extremely easy to access.
- +Simple looking

Interesting Choices:

Search is always visible on main page, even when scrolling through featured houses.

Realtor.com :

- Buy
- Rent
- Sell
- Find Realtors
- Take out loans
- News
- Able to look at current home
- Feedback option
- Careers link
- Shows recently sold houses

Advantages Over Others:

- +Looks more modern, with news, your home estimate
- +Encourages lending by making it look easy

+Logging in and registering as a user is very clean and clear. The separation between different account types is likewise clear.

+Encourages action

Disadvantages:

+Does not look as smooth and sleek as the other two websites

Questionable Choices:

+Option bars at the top are split into two separate entities

+Option bar regions are twice as space consuming as other websites

Our implementation:

-Buy

-Sell

-Rent

-Featured houses

-Maps

-Searching

-Favoriting properties

Advantages Over Others:

-Unregistered buyers may request contact with sellers

High-level Architecture

HTML5

--The standard language for web applications. It is capable of being formatted. Objects, images, videos, and other media may be embedded.

--Usable on most platforms

HTML5 is the standard for web development, and as such this project should implement it.

JavaScript

--Supported by most modern web browsers

--Supports event-driven, functional, and imperative programming

--Can be run client or server side.

JavaScript is a standard for web development. It is extremely supported and widely used. JavaScript is diverse enough to develop anything from simple arithmetic operations to entire games.

CSS

--Eases formatting for HTML components

--Changes can be made in CSS files and applied automatically to all HTML files that implement the CSS files

--Allows for consistency, maintainability, and saves numerous headaches over using HTML without CSS

CSS is yet another of the standards for web development. It is essential for working efficiently; saving time from refactoring numerous files whenever a change in formatting desired.

NODE.JS (Express)

-- open source

--asynchronous input and output

--platform independent

--dynamic events can be run on server before sending to a user

Node allows for the running of dynamic components on a server smoothly and without interference with input or output.

PUG

--Templating language

--Cleaner and less cluttered than HTML, more readable.

--Is converted into HTML

Pug was chosen as a templating language because the syntax used for it is more readable than HTML. HTML contains a lot of clutter with the angled brackets and it can be time-consuming to sift through components. Pug should allow for more malleable and maintainable code due to it being easier to read.

Bootstrap

--Many prebuilt components for web formatting and scripting

--Allows for easy scalability on mobile devices

--Can break components down into grids, allowing for precise and scaled positioning with formats

--Eases resizing, when a window's size is changed the components will generally shift in a readable fashion

Bootstrap allows for easy scaling, supports grids, and makes navigation bars incredibly simple.

Dropdown bars, footers, carousels, forms, and many other components can also be easily accessed. There is a ton of templates that utilize bootstrap, many of which are usable by anyone. If there is something already implemented that does exactly what is needed there is no reason to reinvent that component.

SQL

--Query language

--Update, add to, delete, and retrieve data from databases

--Can structure tables such that items that do not satisfy prerequisites cannot be added

--May cascade on certain operations

SQL is an easy query language that has a lot of optimization that occurs behind the scenes. A user could make a request in an inefficient manner and it will be somewhat optimized. SQL is easy to learn, easy to apply, and extremely useful.

Google Maps

-May imbed google maps into a web-environment

-Simple to set up

Google maps will be extremely useful for the creation of a real-estate website.

Google Analytics

- Can be used to track web traffic
- Can report findings, which can later be interpreted or further analyzed

Workbench

- User interface for administrators/ database management.
 - Facilitates working with a database
 - May automatically fill and reuse certain queries.
- Workbench makes maintenance and usage of the database easier.

PM2

- Process manager for Node.js
 - Allows for easy running, and monitoring of the server
- PM2 is the process manager that is running on our server.

SASS

- Eases writing of CSS
- Can use variable options, allowing for components to be modified in the future

SUPPORTED BROWSERS

- Firefox
- Chrome
- Opera