SW Engineering CSC 648/848, Fall 2017

Real Estate Website - California State Real Estate

Team 08 - Section 2 - global team

Team Lead: Falco Becker (falcobecker@freenet.de)
Members: Richard Robinson, Karan Gupta, Vidit Manglani

Milestone 2

26th October 2017

History Table

Date	Revisions and Feedback	Page
10/24/2017	First draft	
10/25/2017	Reworked after feedback	1 – 16
10/30/2017	2nd rework after feedback, frozen	3 – 8 & 16

Content:

1. Data Definitions V2 3
2. Functional Requirements V2 5
3. UI Mockups and Storyboard 6
4. High Level Architecture and Data Base Organization 9
    a. Data Base Organization 9
    b. Architecture 11
5. High Level UML-Diagrams 13
    a. Class Diagram 13
    b. Deployment Diagram 15
6. Key Risks 16

1 – Data Definition V2

Name	Description	Data
User	Any entity that accesses the website	
Buyer	Any entity that is interested in purchasing a property	Can be registered or unregistered -> inheritance
Unregistered Buyer	A buyer who has not registered with the site	
Registered Buyer	A buyer who has registered with the web site	● Buyer_ID ● First Name ● Last Name ● Address ● ZIP code ● State ● Country ● Contact Info ● Favorites ● Messages
Seller	Any entity that wants to sell multiple properties, needs to be registered	Can be a owner or a real estate agent -> inheritance
Owner	A private person who wants to sell a property, registered	● Seller_ID ● First Name ● Last Name ● Address ● ZIP code ● State ● Country ● Contact Info ● Description ● Favorites ● Postings ● Messages
Real Estate Agent	A professional seller of properties, registered	● Seller_ID ● First Name ● Last Name ● Address ● ZIP code ● State ● Country ● Company ● Contact Info ● Description ● Favorites ● Postings ● Messages

Property	A house that has been listed	● Property_ID ● Seller_ID ● Owner_ID ● Address ● ZIP code ● State ● Country ● Bedrooms ● Bathrooms ● Size ● Price ● Description ● Pictures
Administrator	A employee of the web-site who is able to maintain the system.	● Admin_ID ● Surname ● Family Name ● Messages ● Access-rights
Mailbox	A function, which enables Users to send and receive messages relating to a property	● MailBox_ID ● Owner ● Messages
Message	The messages which are send and received in a mailbox. The Property_ID can be seen as the subject of the message.	● Message_ID ● Contend ● Sender ● Receptor ● Prop_ID ● Date
Favorites	A property which was saved by a user for further investigation. It also can get a rating by the user.	Is a property -> inheritance ● Rating ● Notes
Dashboard	Dashboards can be displayed for registered users for several services.	● User_ID (Seller,Buyer, Admin) ● Mailbox ● Favorites ● Postings ● Description
Posting	A posted property by a owner or a agent	Is a property -> inheritance ● Rating

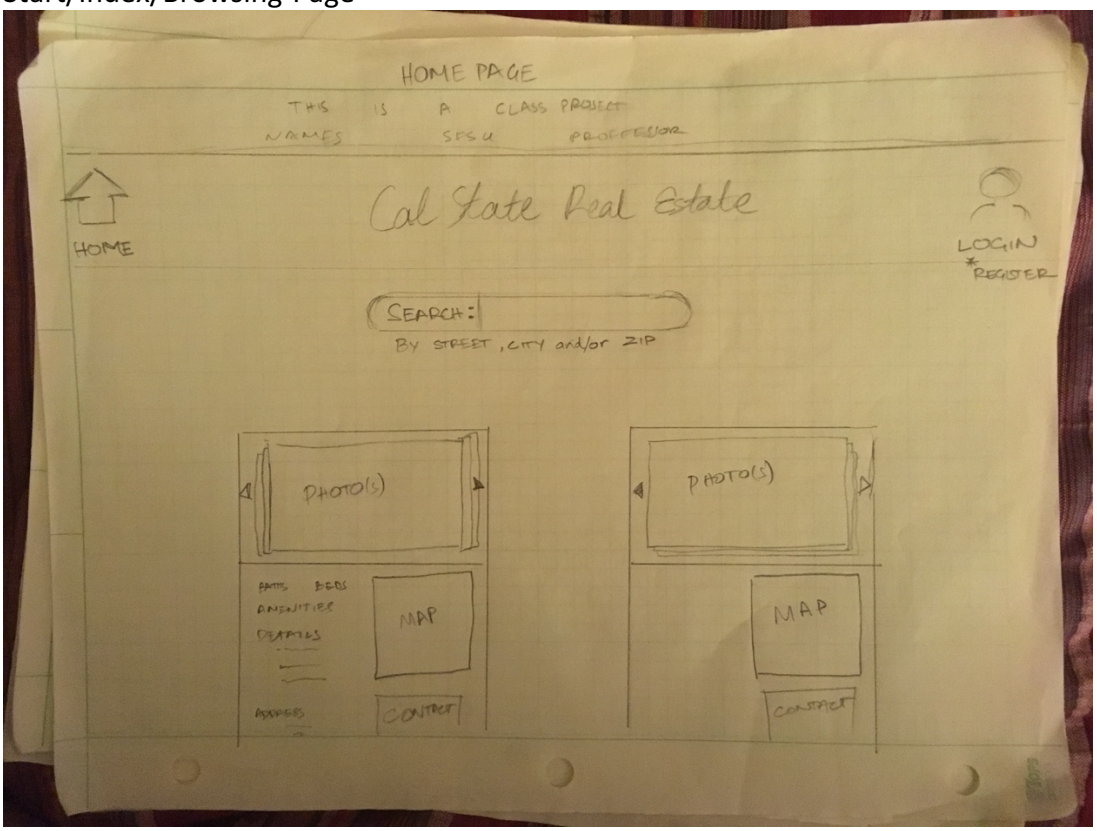
2 – Functional Requirements V2

Priority	Users	Requirement
1	Buyers, Sellers,	• **FR01 - Search Function**: Shall be implemented, in minimum, as a text based way to query the property database. • **FR02 - Filter Functions:** Users shall be able to filter their searches and browsing by price-scale and number of bed/bath-rooms. • **FR03 - Browse Page**: A place for buyers, and a separate place for sellers, to see related properties using keywords, categories or schema. • **FR07 - 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 • **FR11 - Contact Form to Seller** A user shall be able to get in contact with a seller with a contact form or mail.
1	Admin	• **FR12 - DB access and property management:** Admins shall be able to access the database and configure tables, relationships and the data. This shall be done with MySQL Workbench.
1	Seller	• **FR04 - Multiple property management:** Manage a database which handles all information about the offered properties.
2	Seller, registered Buyers	• **FR06 - Login System:** User gives a Username and Password, which shall be checked by the system to give proper access 3, Dashboards.
3	Buyers, Sellers,	• **FR08 - 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. • **FR09 - 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. • **FR10 - Bookmarking:** Registered Users shall be able to bookmark and remember properties, which shall be displayed in their dashboard.

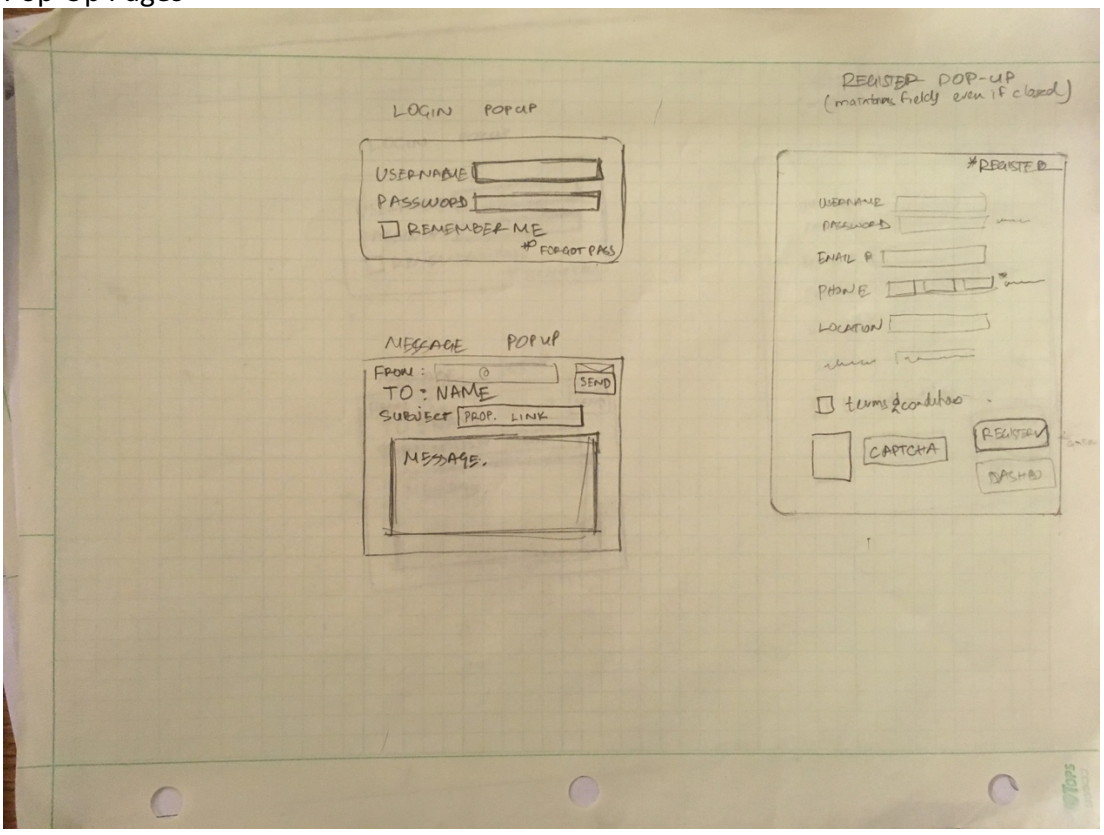
(1 is most important)

3 – UI Mockup and Storyboard

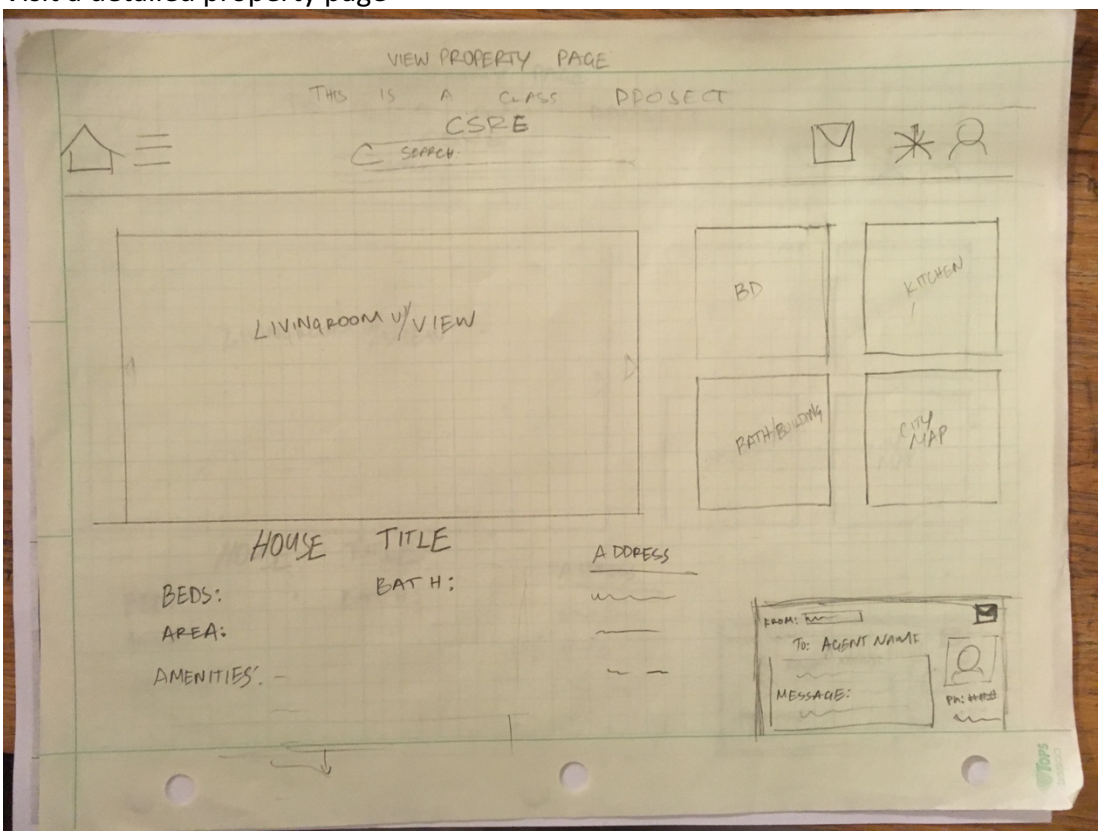
Start/Index/Browsing-Page



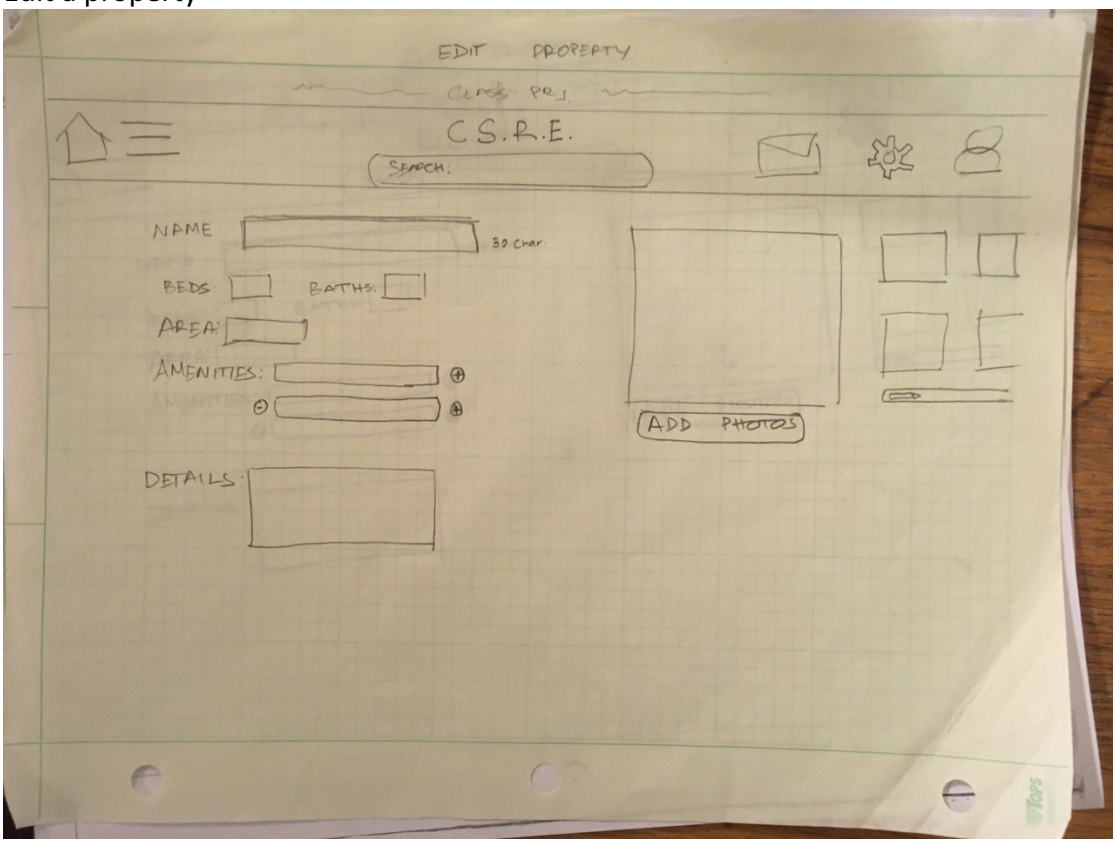
Pop-Up Pages



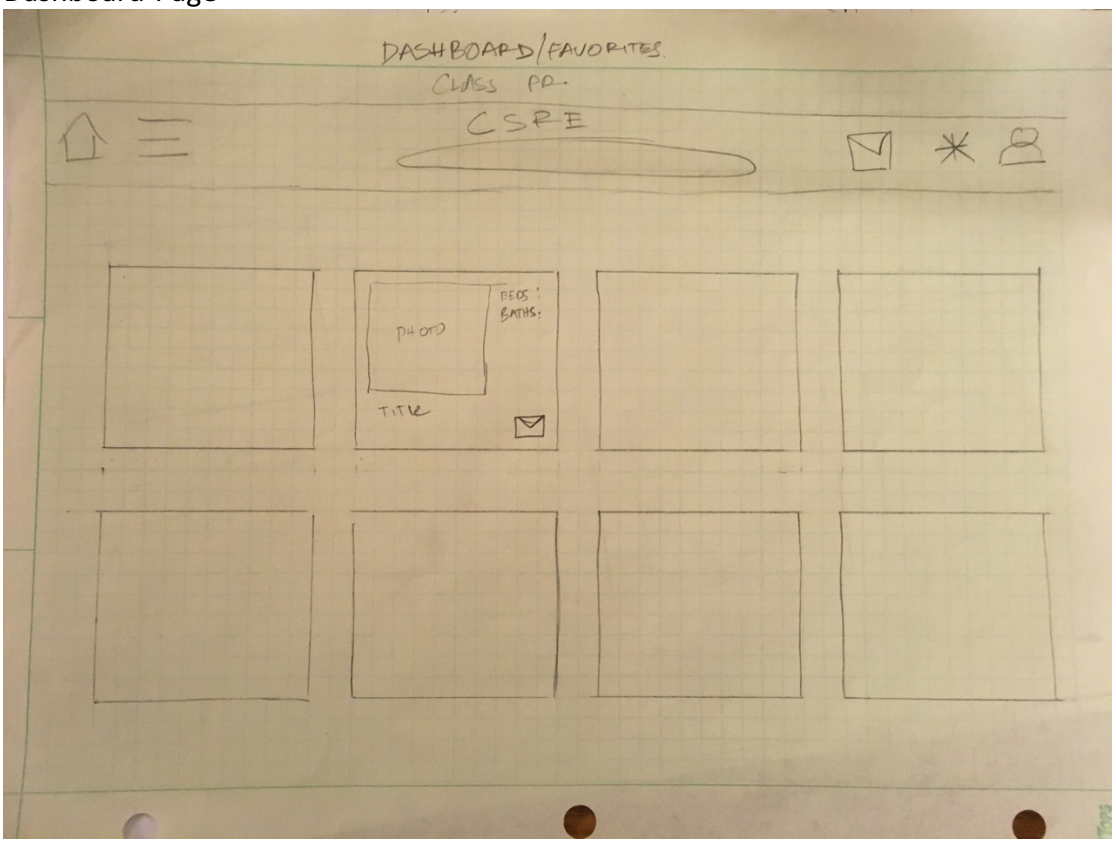
Visit a detailed property page



Edit a property



Dashboard-Page



The goal is, that a user is able to navigate quick and easily throw the webpage. A new user will be able to see featured properties at the start-page and type in search requests on the same page in the search bar.
The search-results shall be displayed under the search bar instead of the featured homes. The user can click on the results and shall get a page with all details (pictures, map, ect.) of the chosen property. There shall also be the opportunity to get in contact with the owner or seller of this property.
If the user is a registered user (buyer or seller) he should also have the option to bookmark properties.
Bookmarked properties, offered/posted properties and messages shall be displayed in a dashboard every registered user has. There are also accessible options for editing/deleting the own posts and user-profile.

4 – High Level Architecture and Data Base Organization

4a – Data Base Organization

Tables

Users		
UserID	Integer	pk
IsAgent	bit	
IsOwner	bit	
FirstName	char	
LastName	char	
EMail	char	
Phone	integer	
Address	char	
ZIPcode	integer	
State	char	
Country	char	
Company	char	
Description	char	

Property		
PropertyID	integer	pk
OwendBy	integer	fk
ManagedBy	integer	fk
Address	char	
ZIPcode	integer	
State	char	
Country	char	
Bedrooms	char	
Bathrooms	char	
Size	double	
Price	double	
Status	char	
Description	char	

Message		
MessageID	Integer	pk
UserID	integer	kf
Content	char	
Date	date	
PropertyID	integer	

Mailbox		
MailboxID	integer	pk
UserID	integer	fk

Favorties		
UserID	integer	fk
PropertyID	integer	fk
Rating	integer	
Notes	char	

Images		
ImageID	integer	pk
PropertyID	integer	
content	BLOB	

Relationships

not tables just overvie

owns	
PropertyID	Integer
UserID	Integer

manages	
PropertyID	Integer
UserID	Integer

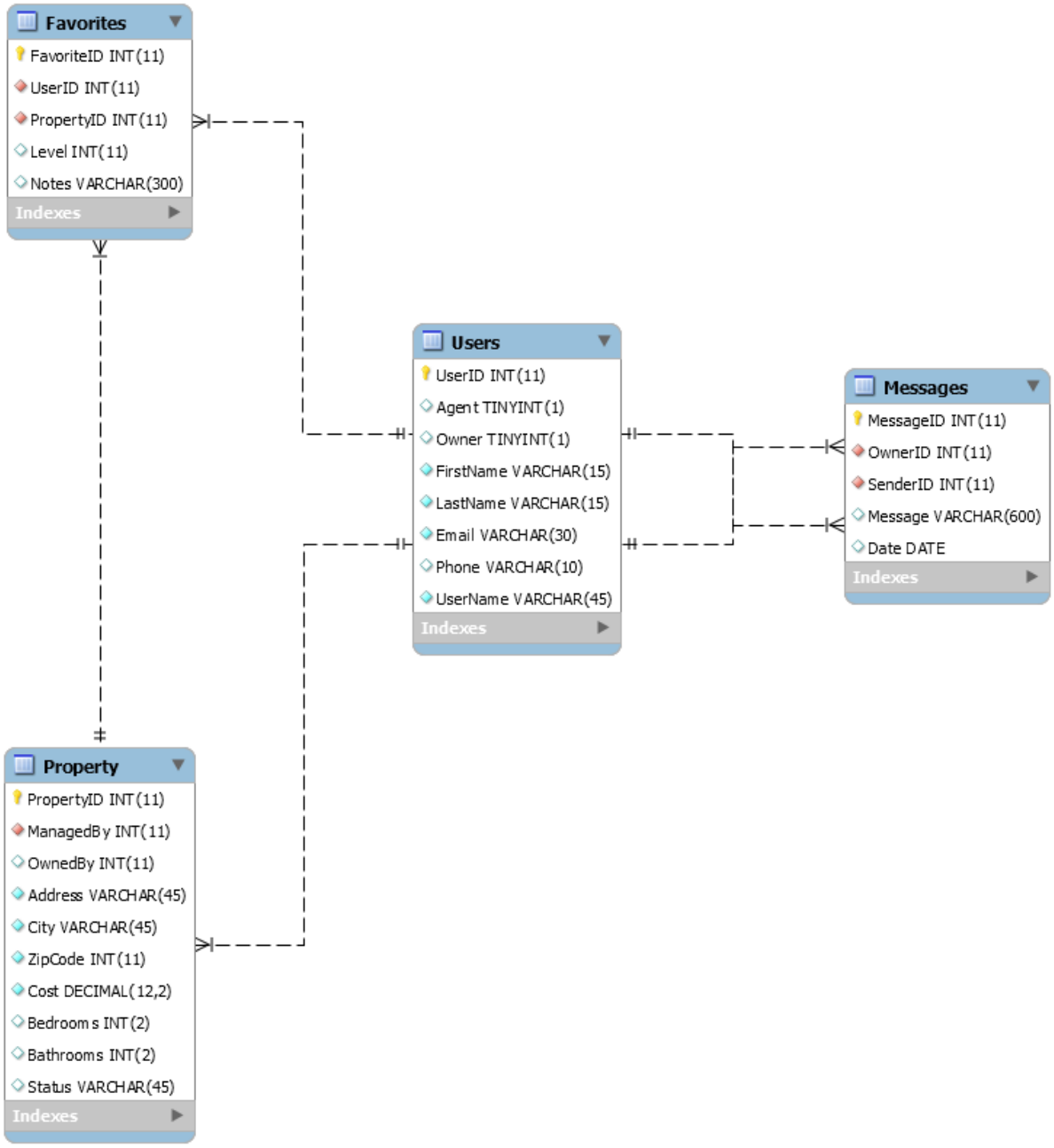
has image	
PropertyID	Integer
ImageID	Integer

has favorite	
UserID	Integer
PropertyID	Integer

send	
MessageID	Integer
UserID	Integer

receive	
MessageID	Integer
UserID	Integer

stores	
MessageID	Integer
MailboxID	Integer


Favorites
FavoriteID INT(11)
UserID INT(11)
PropertyID INT(11)
Level INT(11)
Notes VARCHAR(300)
Indexes
Users
UserID INT(11)
Agent TINYINT(1)
Owner TINYINT(1)
FirstName VARCHAR(15)
LastName VARCHAR(15)
Email VARCHAR(30)
Phone VARCHAR(10)
UserName VARCHAR(45)
Indexes
Messages
MessageID INT(11)
OwnerID INT(11)
SenderID INT(11)
Message VARCHAR(600)
Date DATE
Indexes
Property
PropertyID INT(11)
ManagedBy INT(11)
OwnedBy INT(11)
Address VARCHAR(45)
City VARCHAR(45)
ZipCode INT(11)
Cost DECIMAL(12,2)
Bedrooms INT(2)
Bathrooms INT(2)
Status VARCHAR(45)
Indexes

4b – 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's 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.

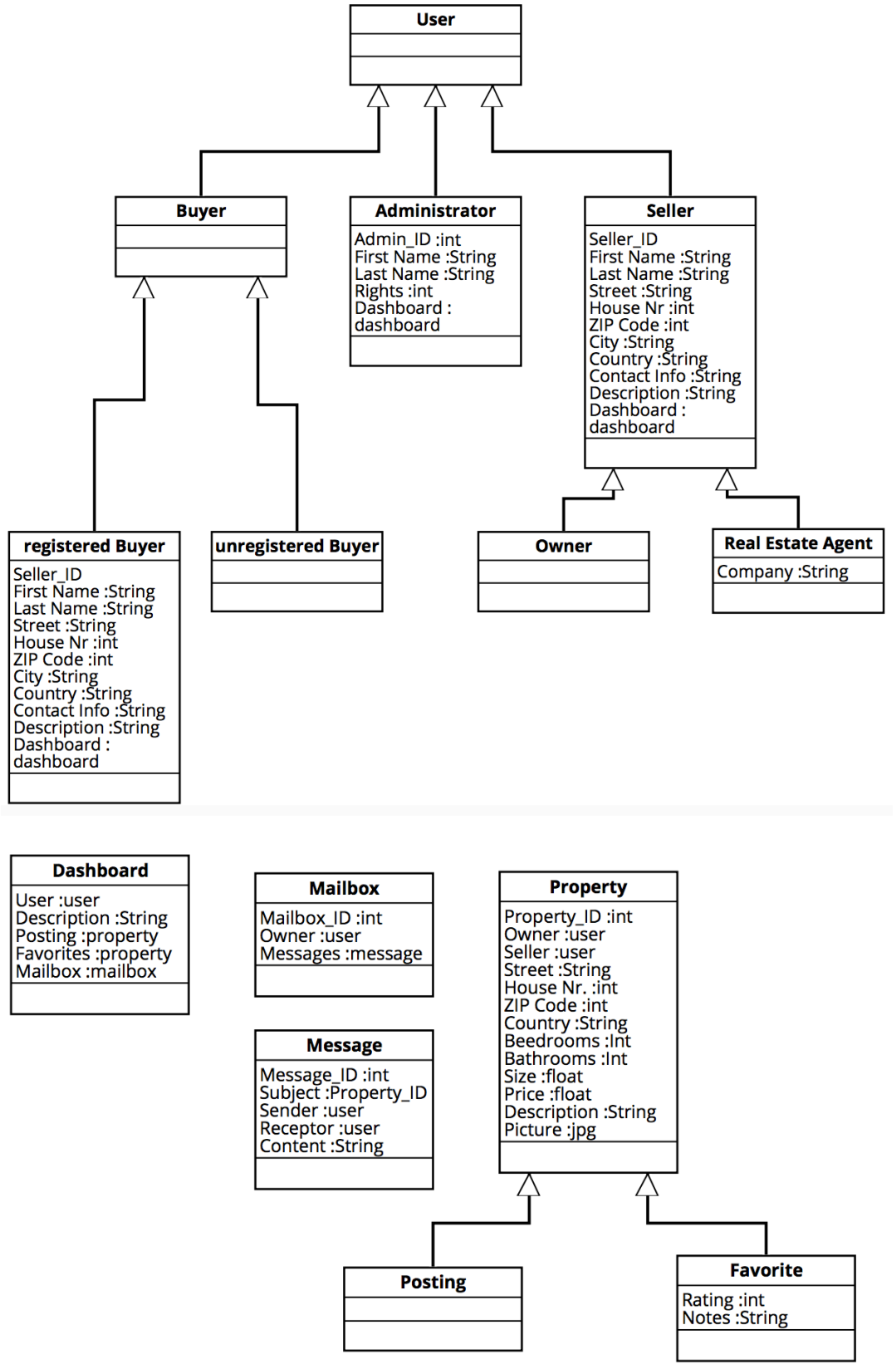
PM2
- Process manager for Node.js
- Allows for easy running, and monitoring of the server PM2 is the process manager that is running on our server.

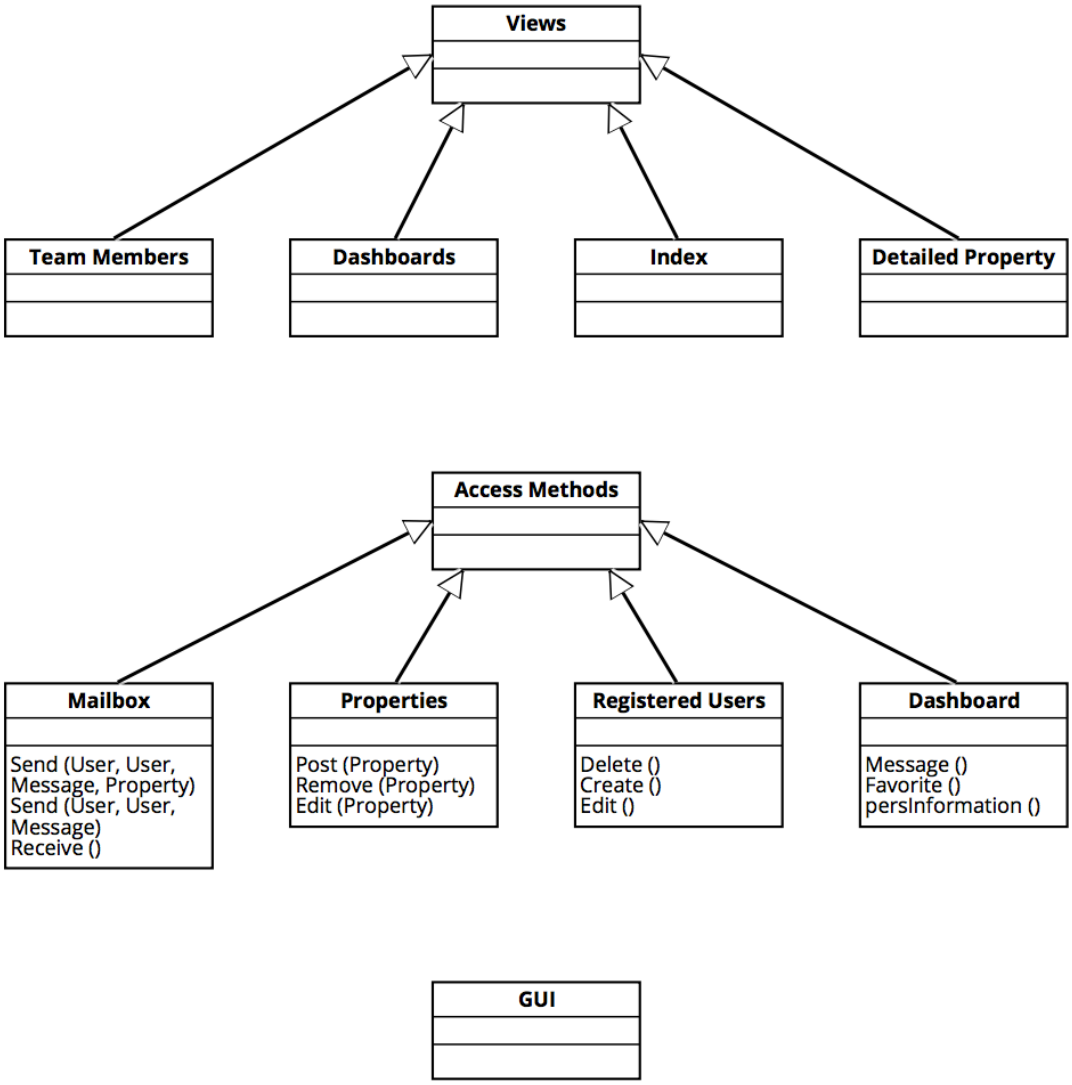
SASS
- Eases writing of CSS
- Can use variable options, allowing for components to be modified in the future

Supported Browsers (latest two versions)
- Firefox
- Chrome
- Safari

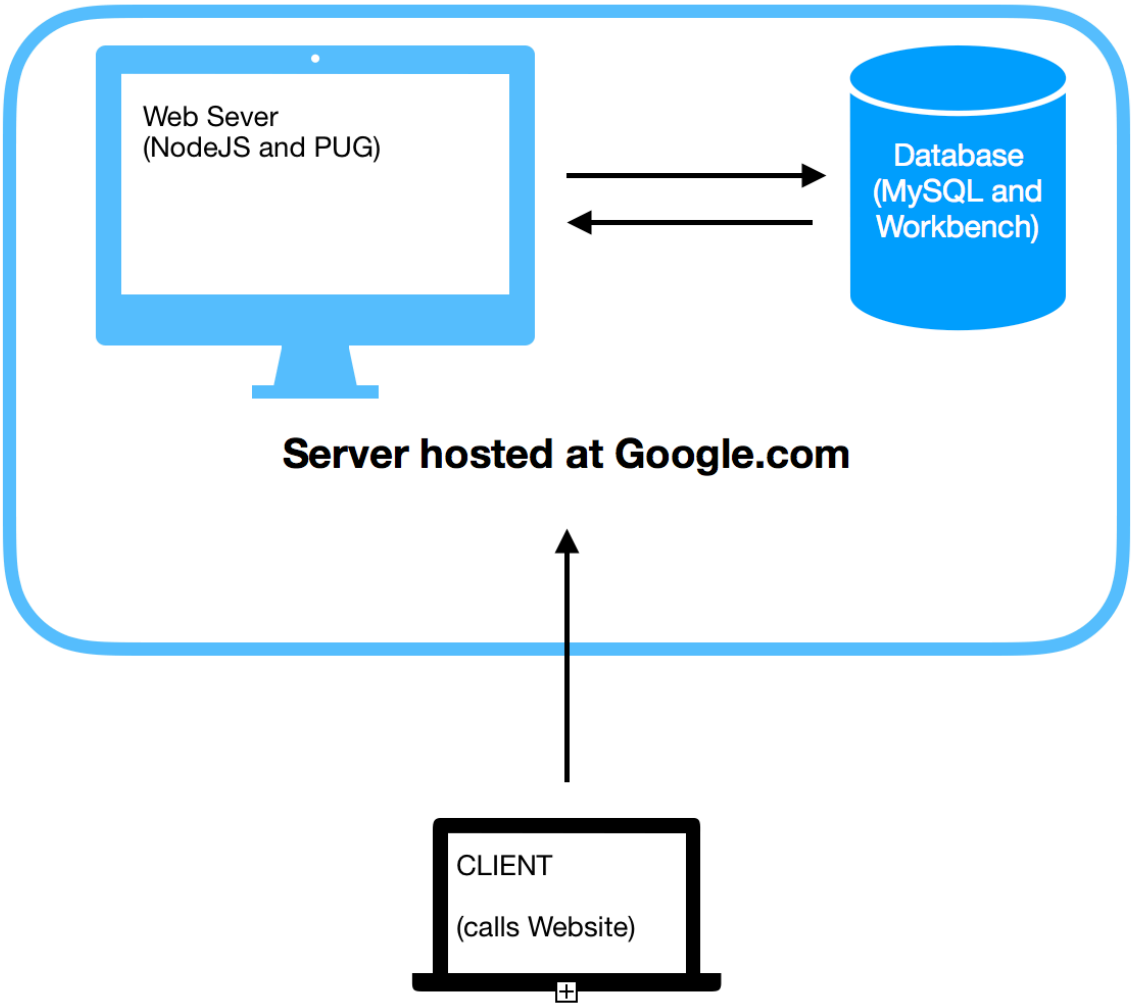
5 – High Level UML Diagrams

5a – Class Diagram




Views
Team Members
Dashboards
Index
Detailed Property
Access Methods
Mailbox
Send (User, User, Message, Property)
Send (User, User, Message)
Receive ()
Properties
Post (Property)
Remove (Property)
Edit (Property)
Registered Users
Delete ()
Create ()
Edit ()
Dashboard
Message ()
Favorite ()
persInformation ()
GUI

5b – Deployment Diagram



6 – Key Risks

Category	*Description*	*Solution*
Skill	Because the global team has just four (instead of seven) members, there is the possibility that the skillset of the members is less and resolving problems or bugs can be harder.	Good organization, work distribution, project documentation and research phases
Skill	In the time of studies, students are able to select courses from a wide list of possibilities. It can be possible, that no student selected a proper internet-application development class, so that there could be a lack of skills in the group.	Good organization, work distribution, project documentation and research phases
Schedule	This is a student project. There is a possibility that the final phase of development is going to happen in the same time, when much work is need to be done for other courses, so that deadlines could not be hold or solved.	Good organization, work distribution and good communication with the instructor, scope of project
Schedule	Because the global team has just four (instead of seven) members, the time for reaching the next milestone with the work that needs to be done for it can be to less.	Good organization, work distribution and good communication with the instructor
Technical	he hosted server could have not enough resources to run the nodeJS app.	Good research, good resource planning and distribution
Technical	The selected framework could possible not support required methods or possibilities to develop the required features.	Good research, good resource planning and distribution
Teamwork	Since this is a global project, there are high risks of communication, at least in terms of timezone-communication. Also can social and cultural differences appear.	Good and responsible communication
Teamwork	It is possible that members misunderstand each other in terms of work-distribution or personal communication, which can lead to strong arguments.	Good and responsible communication
Legal/content	It is possible on the website to post visual media, which can contain copyright issues.	Good review of the user content
Legal/content	It is possible that contend posted by users (properties) are not their legal property.	Good review of the user content