Essos
Real Estate Web Application

Milestone 2
November 3, 2017

Team 06
Alberto Mancini, Team Lead | amancini@mail.sfsu.edu
Julian Morrisette
Mayank Sachdeva
Jirat Parkeenvincha
Felix Chin
Shenliang Wang

Revisions

October 27, 2017	Initial Draft
November 3, 2017	Revised after Instructor Feedback; Frozen

Table of Contents

Data Definitions

Unregistered Guest: Does not require registration or login. These users can browse listings using search and listing filters. They may access all listing information and financial tools, but are unable to save listings to bookmarks or contact an agent.

Registered User: Requires registration and login. These users have access to all search and financial tool features. They may save listings into their bookmarks, and can contact the Agent in-site from a listing. Registration requires the following information:
- Email
- Name (First and Last -optional)
- Password

Agent: Requires registration and login. These users have access to all search and financial tool features. They are granted access to an Agent dashboard in which they may post any number of real estate listings. Registration requires the following information:
- Name (First and Last)
- Email
- Contact number
- Password

Admin: Requires authentication. Admins have access to full site databases via MySQL Workbench. They shall delete accounts and/or listings as per Essos's Terms of Use.

Listing: Real estate listing posted by an Agent account. These listings generate pages for properties that are up for sale through Esso. Listings contain the following information:
- Title
- Address (City, State, Zipcode)
- sqft
- Number of bedrooms and bathrooms
- Pictures (minimum of 2. 1 inside, 1 outside)max 10
- Price ($)
- Type (house, apartment, etc.)
- Description

Financial Tools: Financial widgets that offer different cost estimations for prospective buyers.

Bookmarks: Registered Users may save viewed listings for later viewing.

Agent Dashboard: Agent landing page with options to post, view, and edit listings; and reply to listing inquires.

Functional Requirements

Priority 1 - MUST HAVE

Unregistered Guests
1. Application shall display new, featured, and pending sale listings.
2. Application shall display listing locations on Google Maps.
3. Application shall allow for listing search by city name or zip code.
4. Application shall allow search result filtering by number of bedrooms, number of bathrooms, and price.
6. Application shall allow guests to register for a User or Agent account.

Registered Users
7. Users shall register with their full name, email, and desired password.
8. Users shall be able to contact Agents from the listing page with a pre-filled description.
9. Users shall be able to view Agent responses to listing inquiries made from their account.

Agents
11. Agent accounts shall have access to an Agent Dashboard that provides the options to view messages, post listings, and edit existing listings.
12. Listings shall only be posted by registered Agent accounts.
13. Listings shall contain required information: address, square footage, number of bedrooms, number of bathrooms, asking price, property type, a minimum of 2 images, and a general description.
14. Listing inquiries shall be viewable by the posting Agent account.

Admins
15. Admins shall be able to view all database entities via Workbench.
16. Admins shall be able to delete listings and/or accounts, if necessary.

Priority 2 - MED

Unregistered Guests
    4.1. Application shall also allow search result filtering by sqft and HOA fee.

Registered Users
    10. Users shall have the option to save viewed listings into a collection of Bookmarks.

Agents
    17. Agents shall be able to edit their listings once posted.


Priority 3 - LOW

Unregistered Guests
    4.2. Application shall also allow search result filtering by distance to nearest mass transit node.
    5. Application shall offer a set of financial tools that guests may interact with to receive cost estimates, etc.
    18. Application shall display an average Agent rating on listing pages.

Registered Users
    19. Users can rate Agents from listings 1 to 5 stars.

UI Mockups and Storyboards

Landing Page
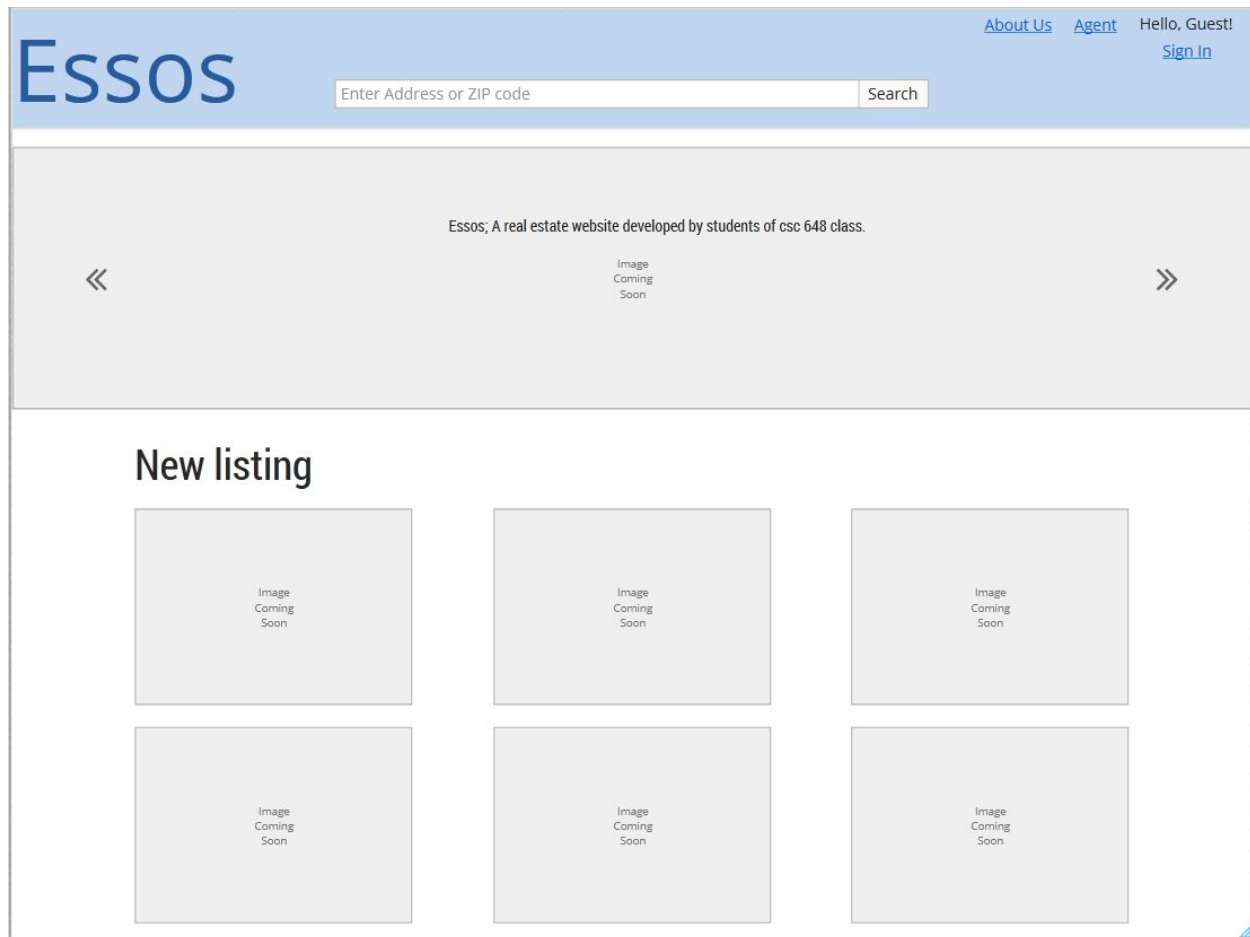
Use cases

- An **unregistered user** can browse through listing using the search tool and click Sign In to create a new account.

- A **registered user** can do further more once they are logged in by clicking Sign In in the top-right corner of the page.

- An **agent** can click Agent in the top right corner of the page to login/signup and begin posting.

Registration/login page

- **Unregistered users** and **unregistered agents** can sign up by filling the following sign up form. The new account type can be selected as Agent/User by using the toggle button right above the form.

- **Registered users** and **registered agents** can just click Sign In and use their credentials to log in.

Search result page
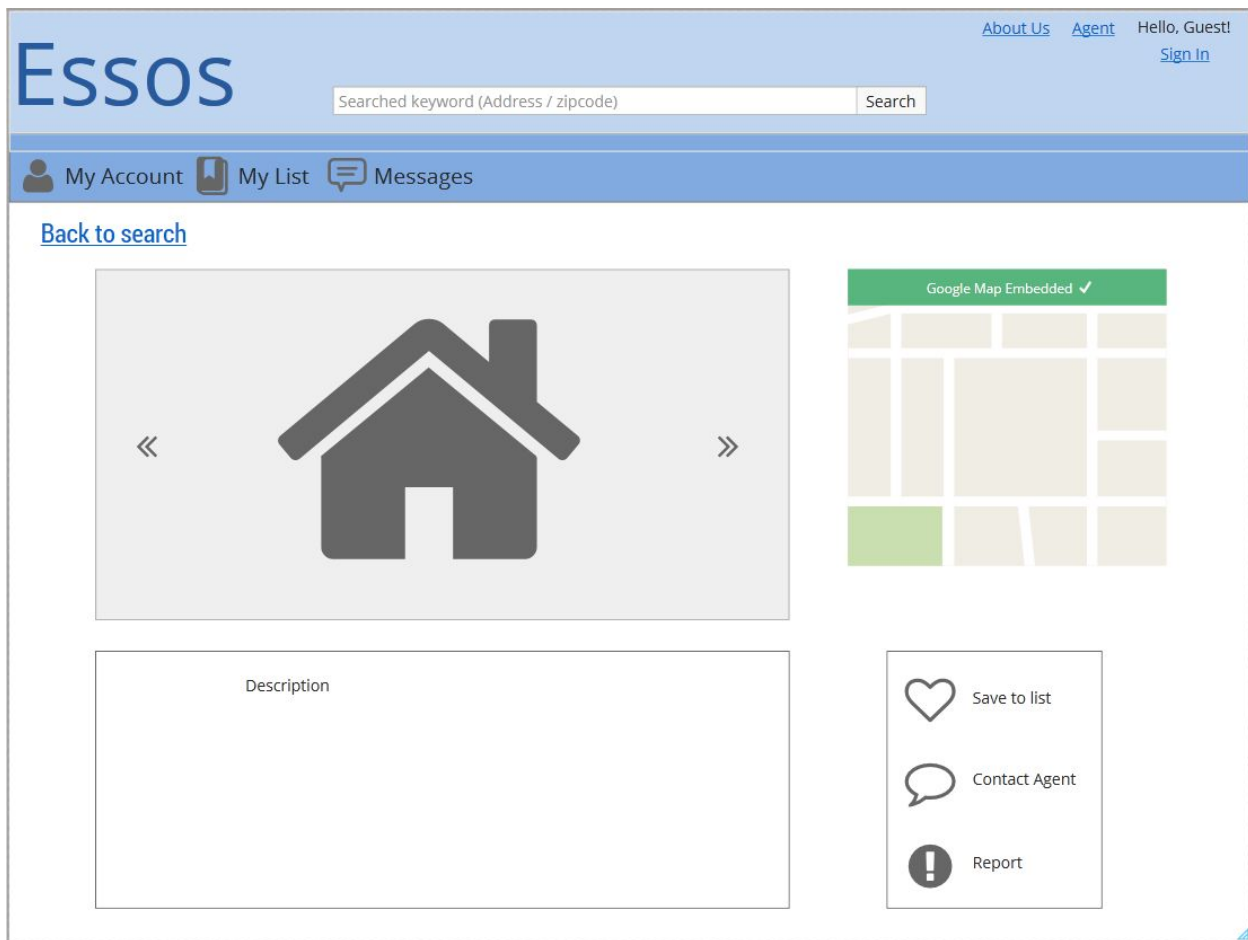

- **All users and agents** can search through listings and use filters based on their preferences. They can further open individual properties posted.

Individual listing page

- **Registered users** can use all available functionality (Save to list, contact agent, report) on this page. They can also access Messages to view their communications with agents, My List to view/edit the listings they have saved and My Account to view/edit their account details.

- **Unregistered users** cannot use any functionality on this page nor can they contact the agent who posted about the property. They may only view the posts and details. In order to use functionality, they must sign up or login by clicking Sign In in top right corner.

Agent listing page

- **Agents** reach this page when they click New Post from their dashboard. They may add photos and details about a property before they hit Submit to post.

User Dashboard
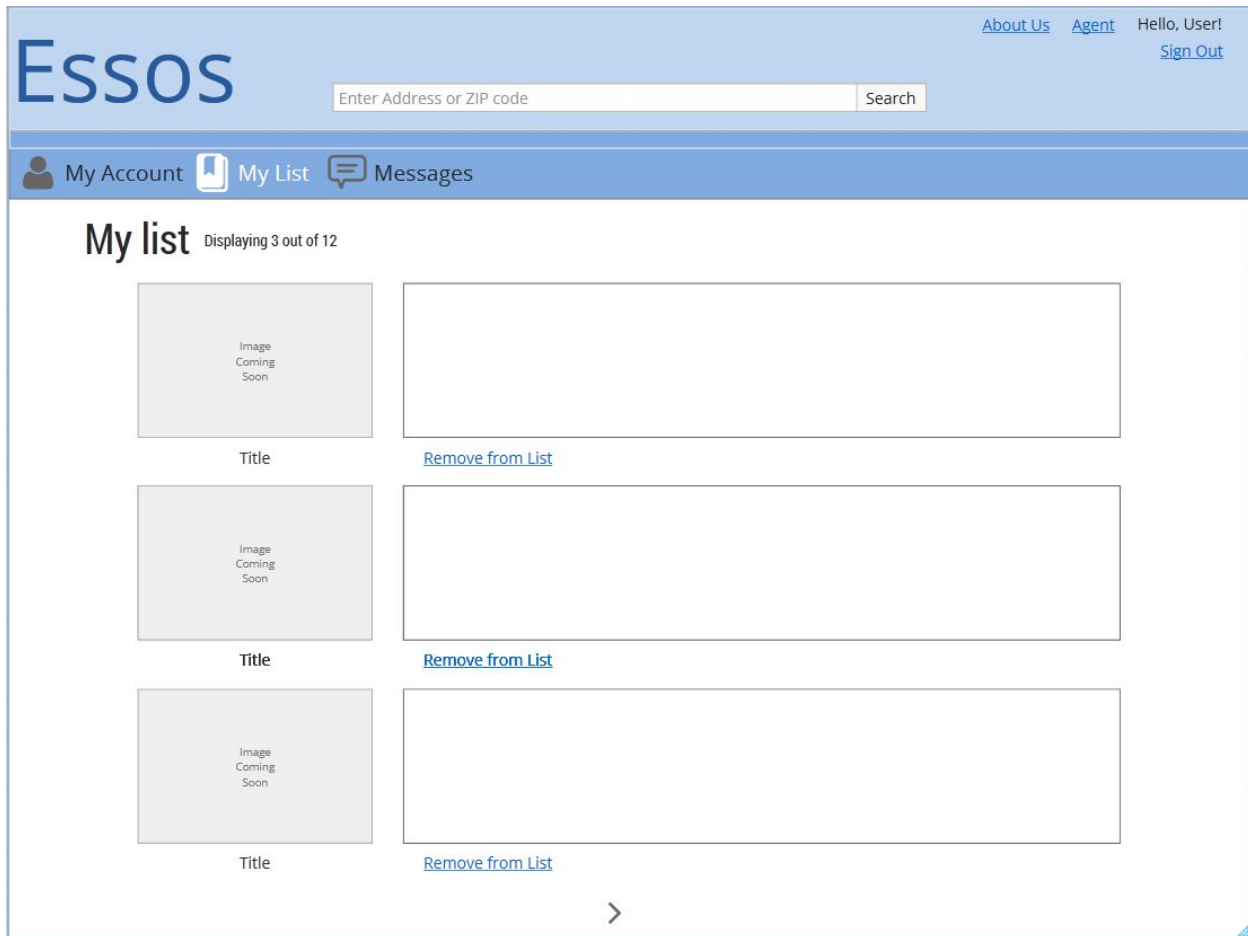
- **Only Registered users** can access this page from their dashboard. They may view or remove the properties that they saved in the past. They may open Messages and My Account as well from this page.

- **Agents** have an almost identical page accessible from their dashboard (discussed on next page). They also share the same functionality of viewing/removing the posts that they saved in the past except they have access to two additional features which we'll cover in Agent Dashboard.

Agent Dashboard
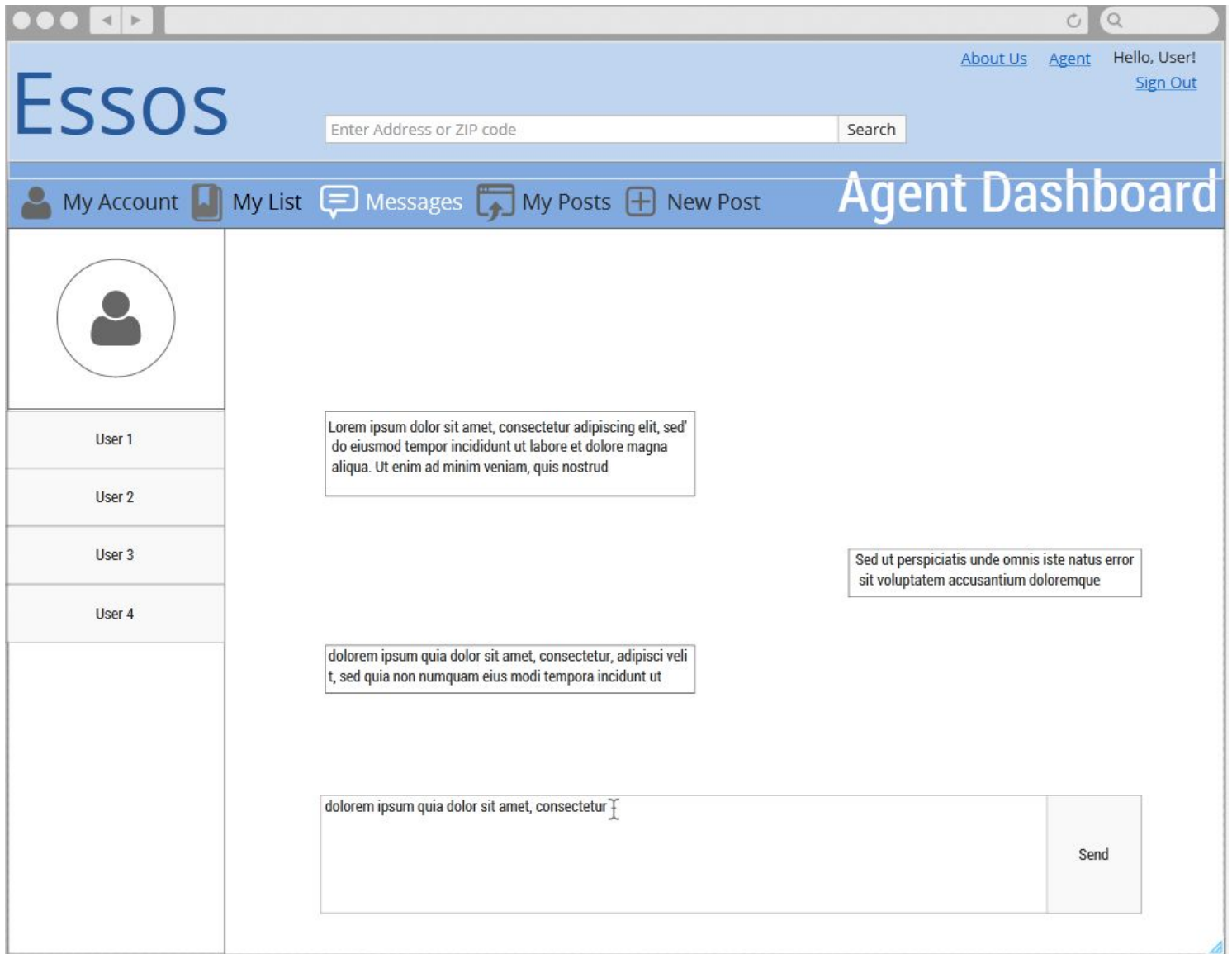
- **Only agents** can access this page this page from their dashboard. They may view their communications with users who have shown interest in their property. They may open My List to view their saved properties and My Account to view their account details. Unlike registered users, agents can utilize two additional exclusive features – My Posts to see the properties they have already posted and New Post to post a new property for sale.

- **Registered users** have a similar dashboard and functionality (as discussed on the previous page). The only difference is that users aren't allowed to post properties.

High-Level Architecture

Our project will follow the general MVC pattern established by the Express framework.
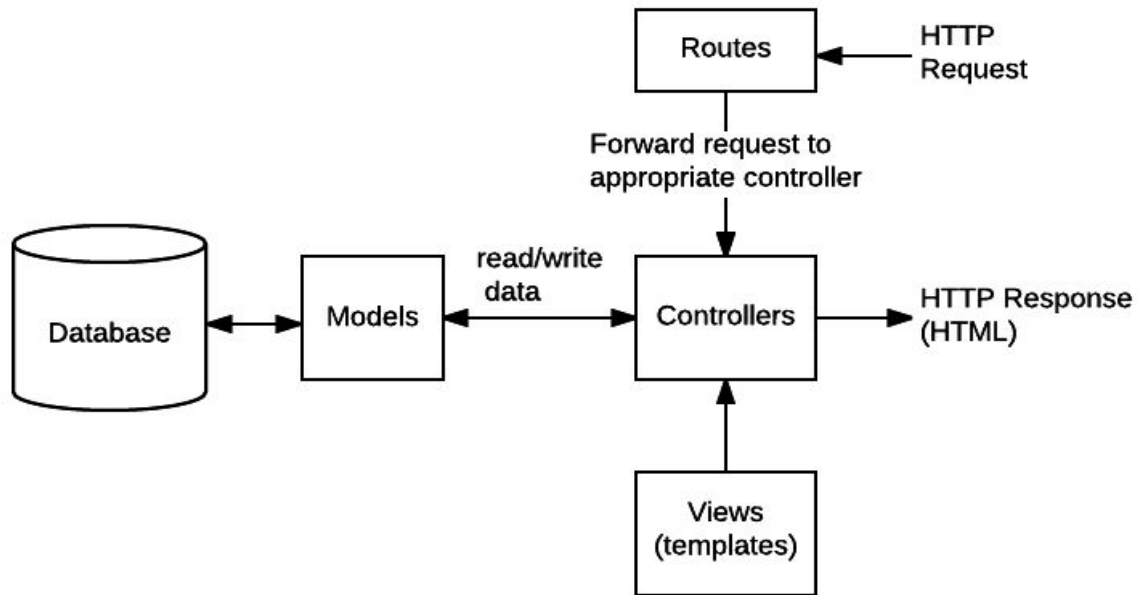
The diagram above illustrates the MVC pattern we will adhere to using Nodejs/Express when developing the Essos real estate web app. The Controller portion of our project will be the routing where a HTTP request will be made. Once a request has been made the controller will request the Model, represented by our MySQL Database and the information stored therein, and passes that Model to the Views portion of our project. The Views, in this case the Jade templating engine will then use the Model to form the HTTP response that the controller will send onward to the client to display on their screen.

Database Organization

Core Database Schema

listing	agent	user	password
-id (PK)	-agent_id (PK)	-user_id (PK)	-id (PK)
-type	-name	-name	-password
-sqrft	-email	-email	-user_id(FK)
-price	-phone	-phone	
-city			
-zipcode			
-image			
-description			
-agent_id (FK)			

Media Storage

Images are in Jpeg format and range in size from 1MB to 8Mb. Every image will have administrative information, along with the ability to connect with and contact the agent. Furthermore, technical information on what kind of medium was used to capture the media as well as descriptive data such as names or specific details pertaining to who/what was being photographed. All of our medias are in color and will be stored in a *file system*.

Search/Filter Architecture

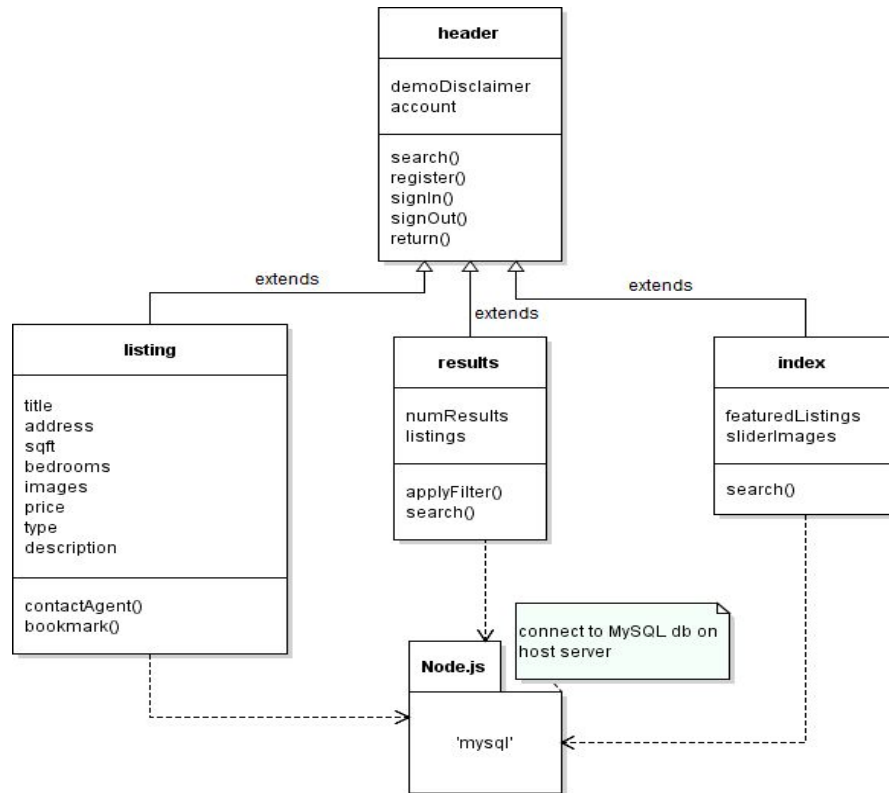Users have the ability to use search terms relevant to cities, zip code, and price, to name a few. Our search algorithm finds all items in our database that a best percent-like relevant to be search. If search fails, it displays featured items in our database.
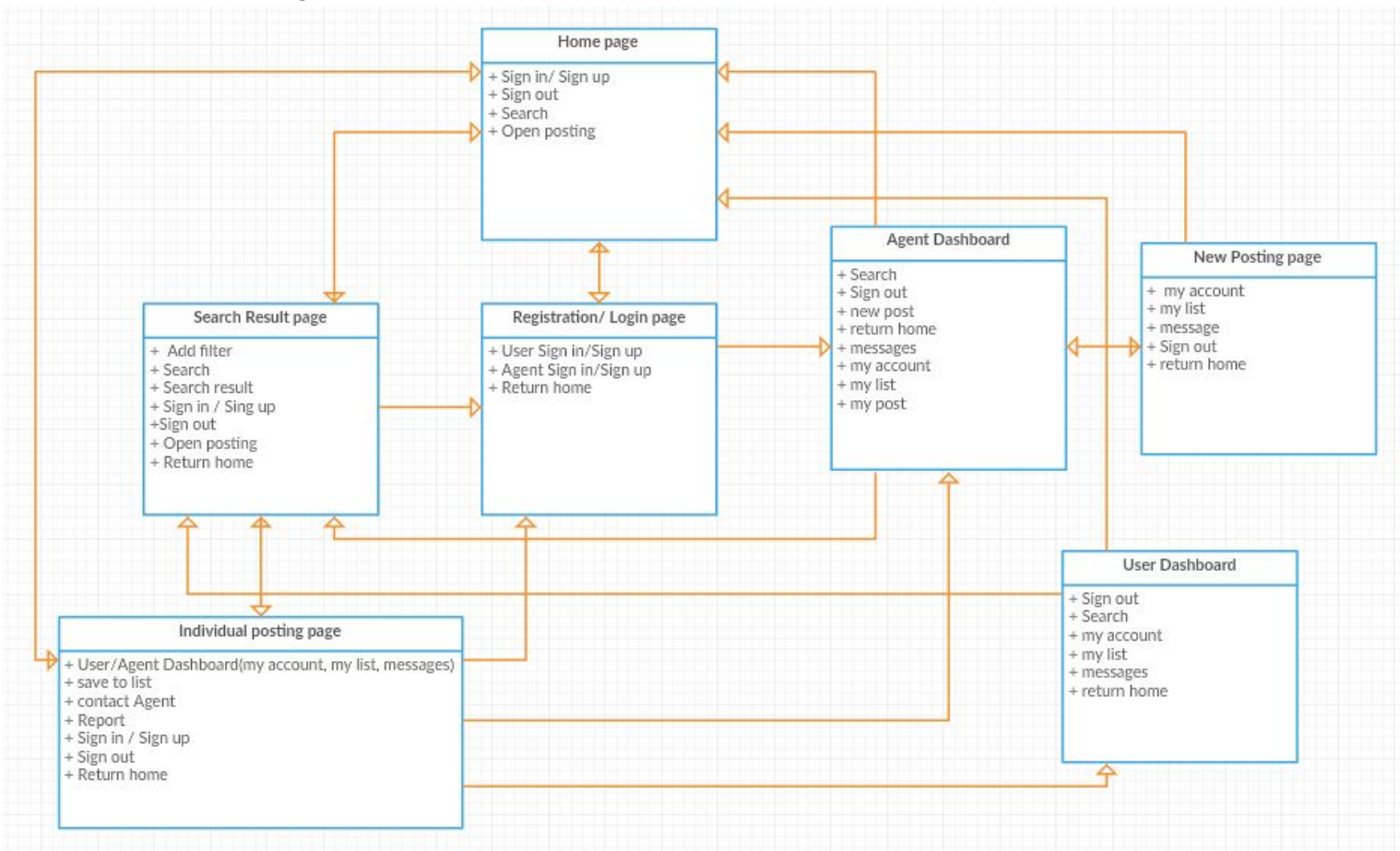
Filters shall be implemented by taking filtering input from the UI to extend the already present search query and use it to re-query the database.

UML Diagrams

Class Diagram - High Level Search Functionality

header

demoDisclaimer
account

search()
register()
signIn()
signOut()
return()

extends · extends · extends

listing

title
address
sqft
bedrooms
images
price
type
description

contactAgent()
bookmark()

results

numResults
listings

applyFilter()
search()

index

featuredListings
sliderImages

search()

connect to MySQL db on host server

Node.js

'mysql'

View Flow Diagram

Home page

+ Sign in/ Sign up
+ Sign out
+ Search
+ Open posting

Search Result page

+ Add filter
+ Search
+ Search result
+ Sign in / Sing up
+Sign out
+ Open posting
+ Return home

Registration/ Login page

+ User Sign in/Sign up
+ Agent Sign in/Sign up
+ Return home

Agent Dashboard

+ Search
+ Sign out
+ new post
+ return home
+ messages
+ my account
+ my list
+ my post

New Posting page

+ my account
+ my list
+ message
+ Sign out
+ return home

Individual posting page

+ User/Agent Dashboard(my account, my list, messages)
+ save to list
+ contact Agent
+ Report
+ Sign in / Sign up
+ Sign out
+ Return home

User Dashboard

+ Sign out
+ Search
+ my account
+ my list
+ messages
+ return home

Component and Deployment Diagram
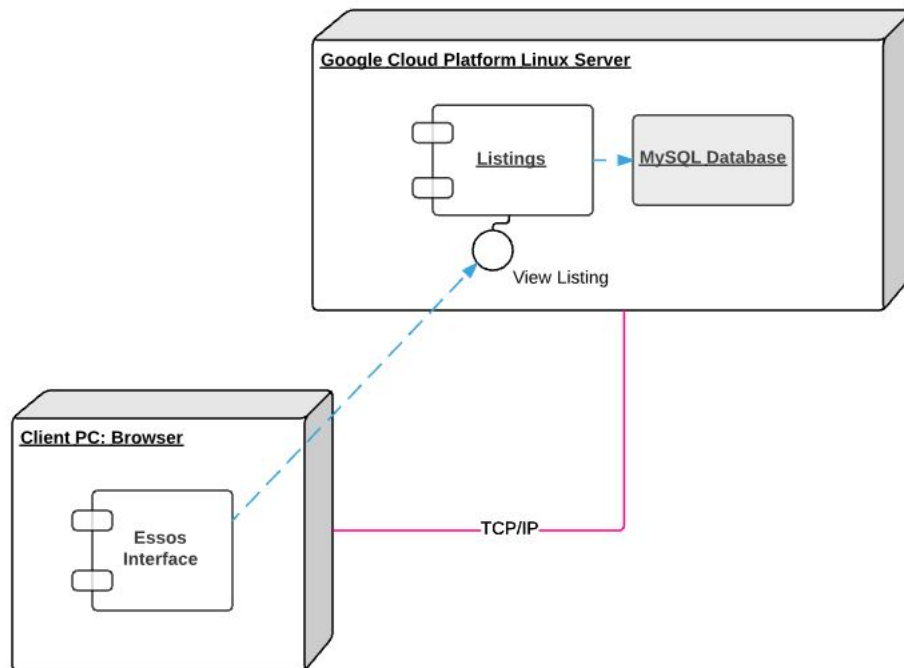
Project Risk Assessment

Upon review of the organization of the Essos real estate web application and its development team, the following risks have been identified:

- *Skills risks*: the majority of the team is unfamiliar with the given development stack (node.js/express) for the project, and may need accommodate time to gather and learn from various resources

- *Schedule risks*: as the development team is comprised of senior level undergraduate students, scheduling conflicts may emerge due to individual projects or jobs

- *Legal/content risks*: this project requires the use of media (uploaded images) which may potentially infringe upon the rights of its owners

In order to resolve the identified risks, SCRUM-style meetings will be held between the team lead and the rest of the team weekly either online or in person to discuss any issues with the application development. At scheduled extended meetings, we will periodically review the coding procedures that need to be done, and research the methods of accomplishing them within the chosen framework and development stack. Code reviews will be conducted within the smaller teams to resolve any coding issues. The back-end team will work together to document a vertical prototype and educate the rest of the team.

During the weekly SCRUM-style meetings, we discuss availabilities for the week and plan for any scheduling conflicts that may arise with any of the members. To ensure deadlines are met, tasks are assigned with soft deadlines depending on the availabilities discussed. An extended weekly meeting is typically scheduled to discuss issues in depth at these meetings. We will keep the scope of the project to a minimal P1 set as development continues on.

For demo purposes, our project will use images of properties. In order to avoid any legal issues, all images must be reviewed by team leads before pushed to the live project site. The sources will be checked for the distributive rights of the images - only commercial-free (and no attribution required) images will be approved for demo use.