SW Engineering CSC648/848 Spring 2019

Housingator

Team 14 (Local)

Davis Hoang (Davis.hoang2015@gmail.com) (Team Lead / Github Master)

Front-End

Jonathan Fox (Front-End Lead)
Kevin Mitsuda
Jarrett Lee

Back-End

Hashim Jacobs (Back-End Lead)
Eric Chen
Shalaka Aigal

Milestone 2

3/22/2019

Revision Table		
03/22/2019	1.0	Initial Document
4/25/2019	2.0	Revised Document

Data Definitions:

- Unregistered User : The unregistered user will be able to browse through all the property listings but will not be allowed to post any listing.

- Registered User : Registered user will have the have the same rights as unregistered users. The registered user will have a user account on the website will be able to view property listings and can also post the listing provided they are logged in on the site. Registered Users can be both Landlord/Seller and Tenant.

- Landlord/Seller: A registered user who has a property to rent and can post the property on the website. Registered user can view property listing.

- Administrator : Users that have special privileges, and have the ability to remove posts from the site, remove items, issue warnings and bans from the site, and generally enforce the Code of Conduct for the site. Administrators also are responsible for helping users when needed.

- Property : The main data item is property listed for rent or sublease. The following are the types of property which can be listed on the site.
    - House : Individual house with 1 or more rooms to be rented.
    - Apartment: Apartment with 1 or more rooms to be rented.
    - Room: Single or shared room to rent.
    - Price: Price of the property
    - Description: The property will have a description of the house listing.

- Registration: The user will have to register in order to post. The registration details will collect the User's name, email, contact number will be collected from the users during registration.

- Message : Registered user can message landlords about house listings. Registered users will be able click a button that sends a message to the homeowner/landlord that they are interested renters.

- Dashboard : Registered users
    - Shall be able to see house listings
    - Shall be able to see their post
    - Shall be able to see messages about house inquires

Data Definitions (continue):

- Dashboard : Admin users
    - Shall be able to approve or deny posting
    - Shall be able to see all postings

Functional Requirements:

Priority 1:

Unregistered Users:

1.1 Users shall be able to view housing types on the home page.
1.2 Users shall be able to sort by house type, price and distance.
1.3 Users shall be able to see a photos and description of the house.
1.4 Users shall be able to sign up with email and password.
1.5 Users shall be prompted to sign up before contacting homeowners.

Registered Users:

2.1 Users shall have the same functionality as an unregistered user.
2.2 Users shall be able to login with their email and password.
2.3 Users shall be able to contact homeowners about housing inquiry.
2.4 Users shall be able to post listing of their housing type.

Admin:

3.1 Users shall have the same functionality as a registered user.
3.2 Users shall review the content and description of all uploaded house listings and approve it before going live.
3.3 Users shall be able to delete or block registered users.

Priority 2:

Users:

4.1 Unregistered Users shall be able to view a video of the house listing.
4.2 Registered Users shall be able to sort housing post by the number of allowed occupants.

Priority 3:

Users:

5.1 Users shall be able to use an Interactive Map to see the surrounding neighborhood.
5.2 Registered Users Shall be able to compare and contrast up to 5 at once posts.

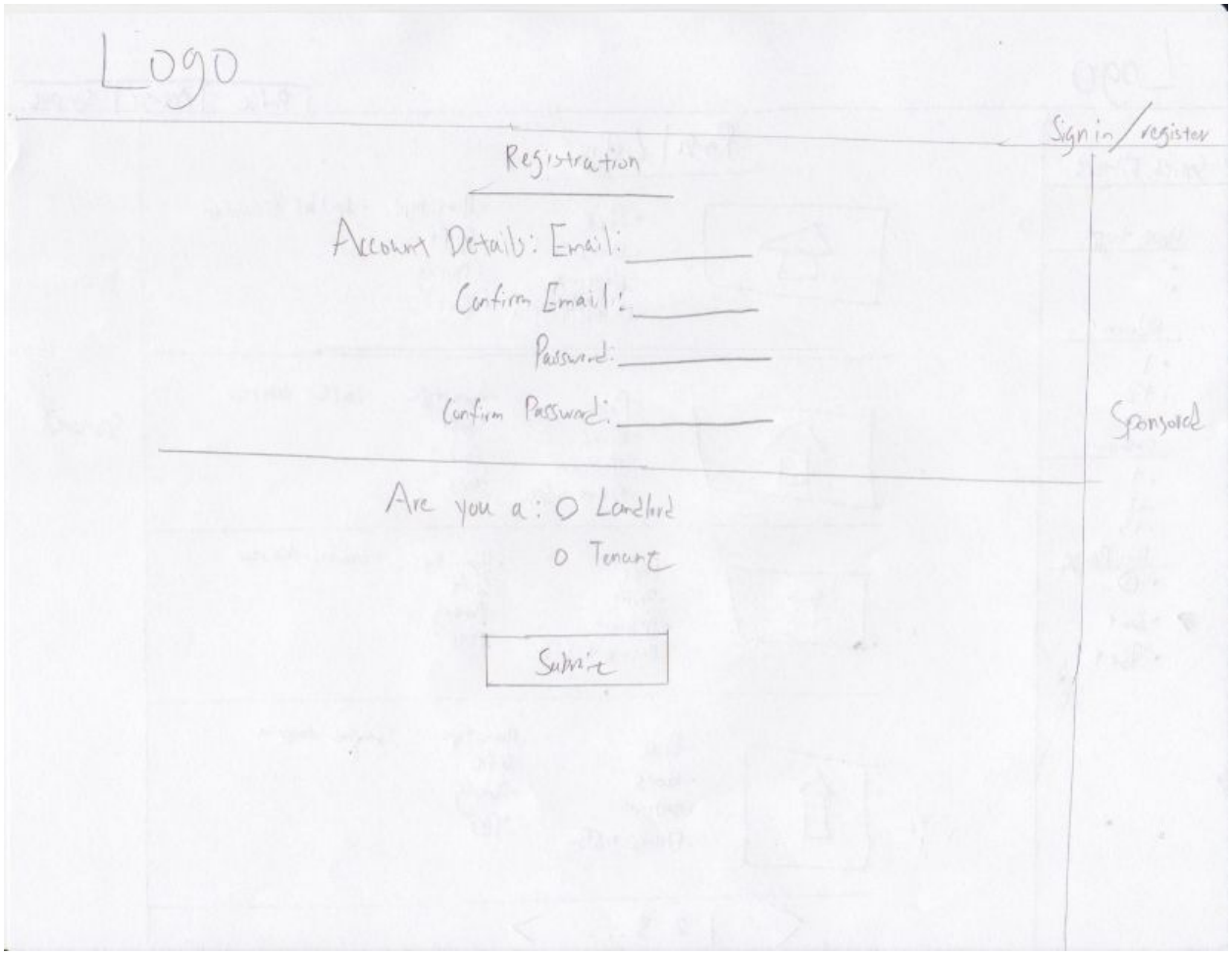
UI Mockups and Storyboards

Home Page Listings:

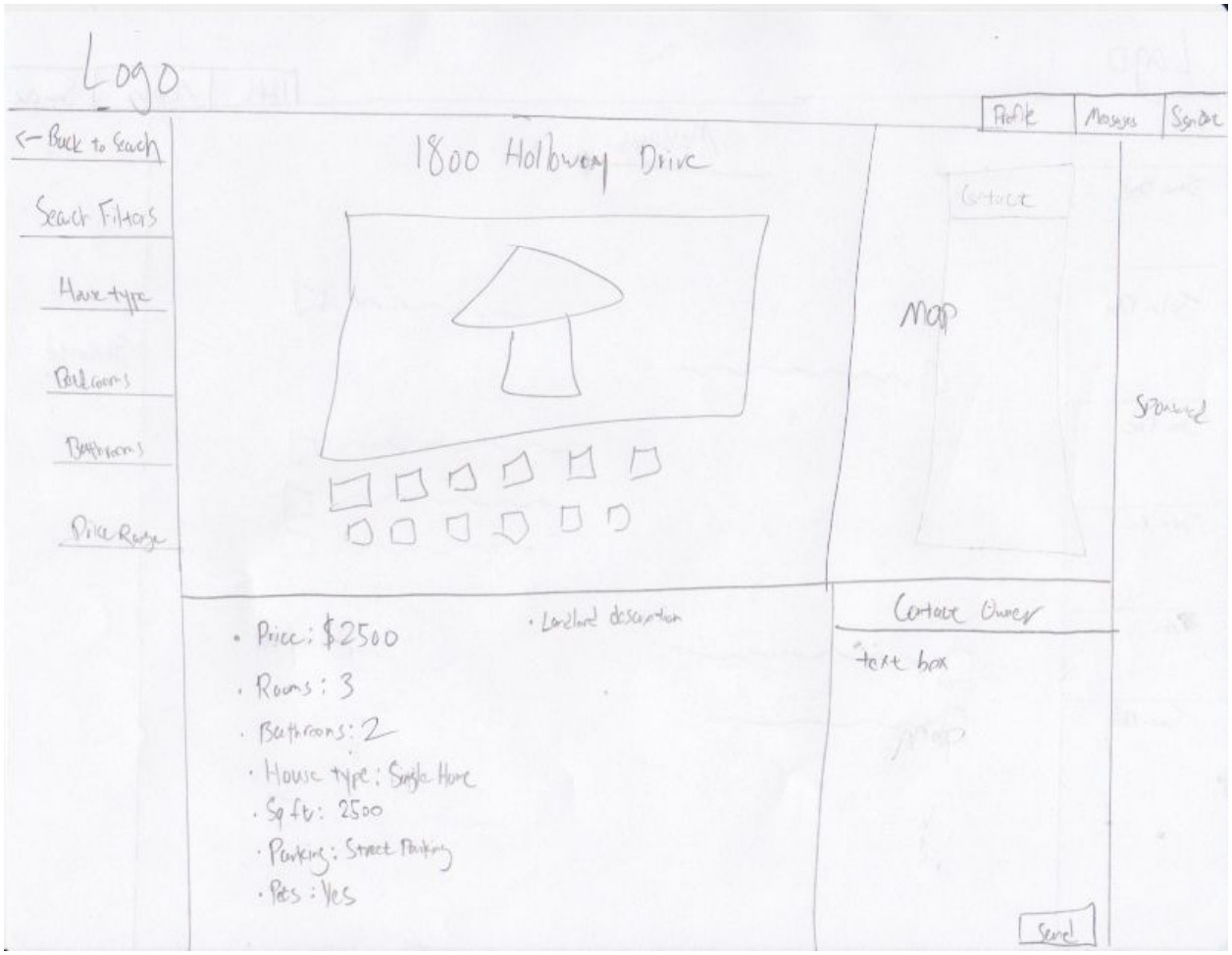
Use Case 1:

Joe is a college students. He is currently a incoming freshman who works part time at starbucks. Joe was planning on moving into the dorms but got waitlisted and is 100th on the waitlist. At this point it seems unlikely that joe will not find housing on campus. Joe decides to take action himself to look for a place before school starts. Joe does onto our app as an unregistered users and browses through available options. Joe finally finds a room he would like to rent out and would like to contact the homeowner. He tries to contact the homeowner and was prompted to register or log-in. Once joe has his own account he will be able to contact the homeowner.

Registration:



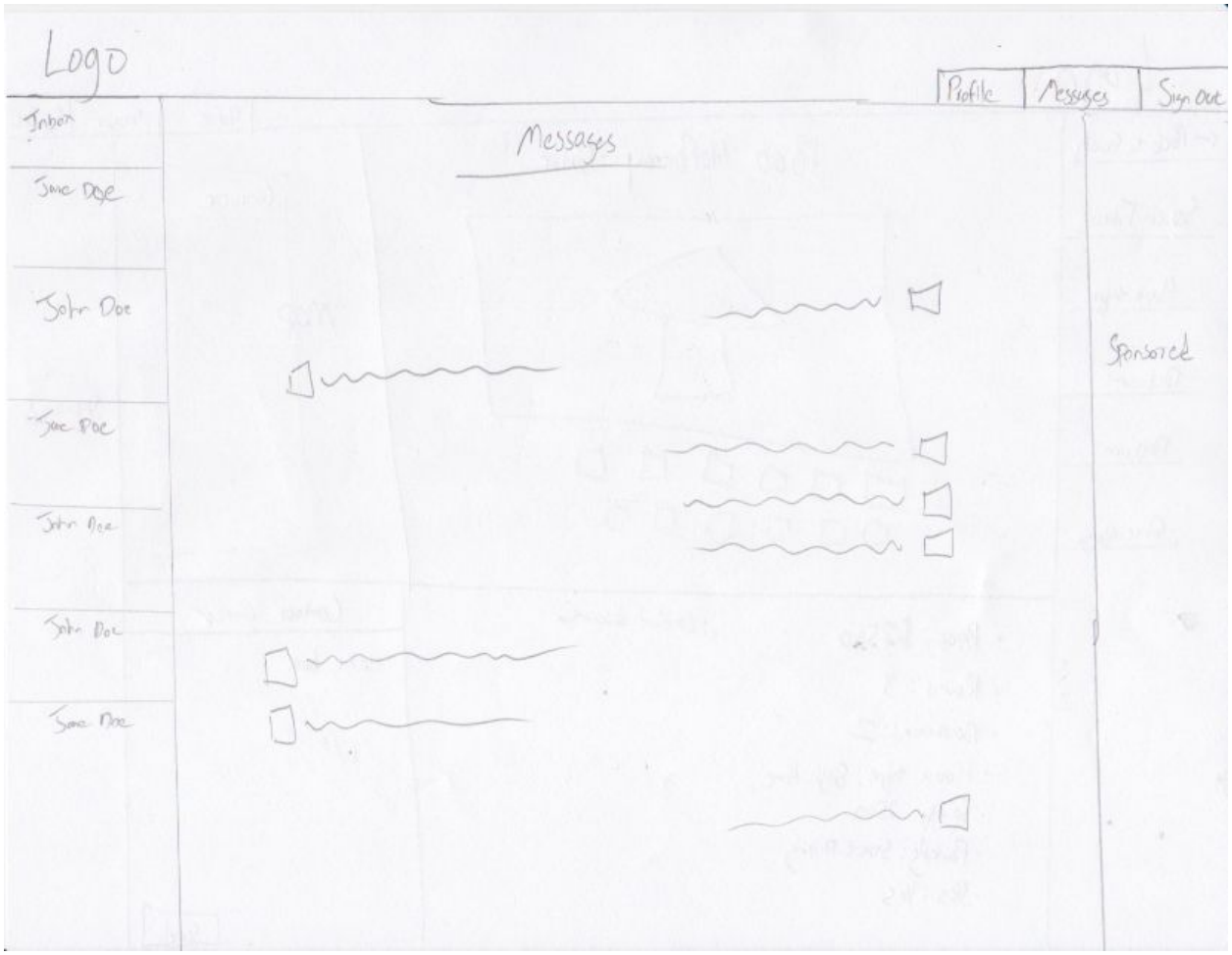
Detailed view of rental property:



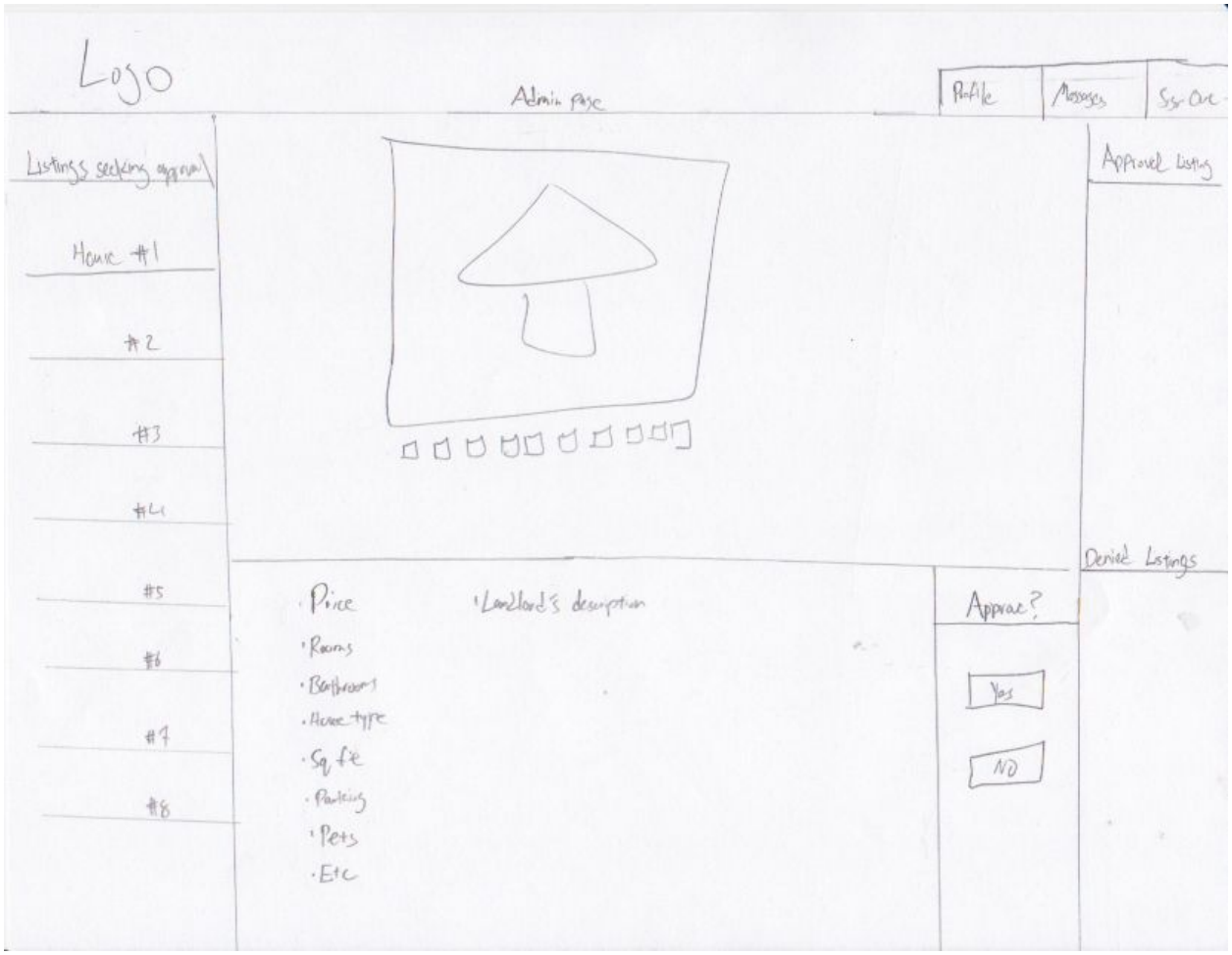
Use Case 2:

Adam is a homeowner. Adam is a 54 year old man who flips houses for a living. He would like to rent out one of his houses he just remodeled. Adam goes into SFSU house me website and registers. Once adam registers, he will then be able to post photos, videos and a description of his house to rent. After adam post a listing of his house he will be able to see any inquiry of registered users who are interested in his house listing on his dashboard.

Messages View:



Admin Page for approving listing:



Use Case 3:

Jasmin is an admin. When she starts her day at work, she opens the homepage of SFSU house me and logs into her account. She then navigates to the house approval section. She the reviews each house listing and its description before deciding to approve or decline the request. She will be able to review accounts to check to for integrity and ethics. If an account does not uphold standards of SFSU house me their account will be deleted or modified.

High level Architecture, Database Organization

· *DB organization*

PostGreSQL is the database which will be used for the database storage.
Following are the high level tables :

Table Users: userId:int (PK),
userName:String,
email:String,
password: String,
address: String

Table Property: postId:int (PK),
noOfRooms:int,
address:String,
zipcode:int
description:String,
price:double,
imagePath:String,
postDate: DateTime,
userId:int (FK references Users),
propertyId:int(FK references Property)

· *Media storage*:

- File System is the the media storage option . Any Image, Audio or Video files will be placed in a file system and be retrieved from there to display on the webpage.

· *Search/filter architecture and implementation*:

- The header section of our website will have a search box where the user can search for some listing by searching for some keywords. There will be filter criteria like a drop down for Property Type like House, Apartment, Room.

Database Organization (Continue)

The implementation for the filter section:

- By default none would be checked so we would display some results based on recent posting. When a filter is checked or chosen by the user, the database SQL query will be modified to put the WHERE clause for all the filter and the result will be sent to the frontend to be displayed.

The implementation for the search section:

- The SQL %Like query will be used to implement the search section by querying the database for the search keywords entered by the user.
- Our Homepage will include a search box where if a user just clicks on the search box, then all the listings will be displayed on /search. The user can search by some keywords such by the city, by zip code, etc., then only the specific listings according to the keyword will be displayed using %LIKE SQL query.

Your own APIs (if any): Describe and define at high level any major APIs that you will create other than standard ones provided by tools and frameworks you use
Google Maps Platform:

- Routes - Directions API: Provides directions for transit, biking, driving, and walking between locations.
- Routes - Distance Matrix API: Calculate travel times for multiple destinations.

Describe any significant non-trivial algorithm or process if any (like rating, ranking, automatic prioritizing of items etc.)

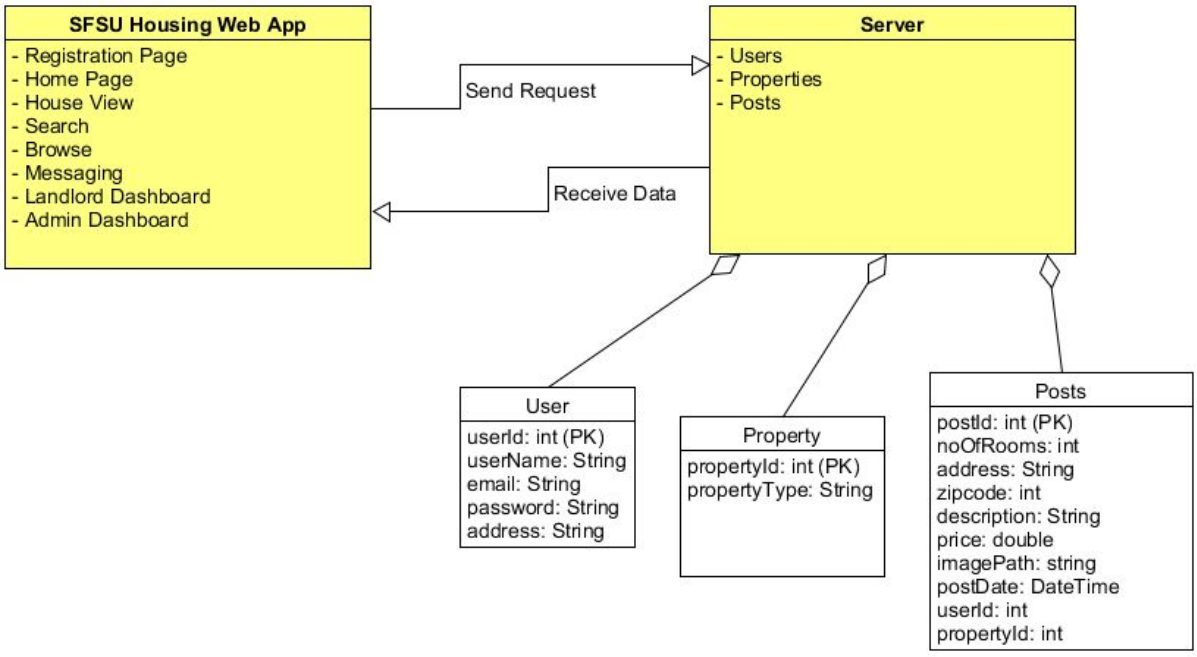
- No non-trivial algorithm for search and process expected at this particular time.

If you have changed SW tools and frameworks or added any new one please describe it. Any new SW or framework you will be using has to be approved by CTO in writing by this time.

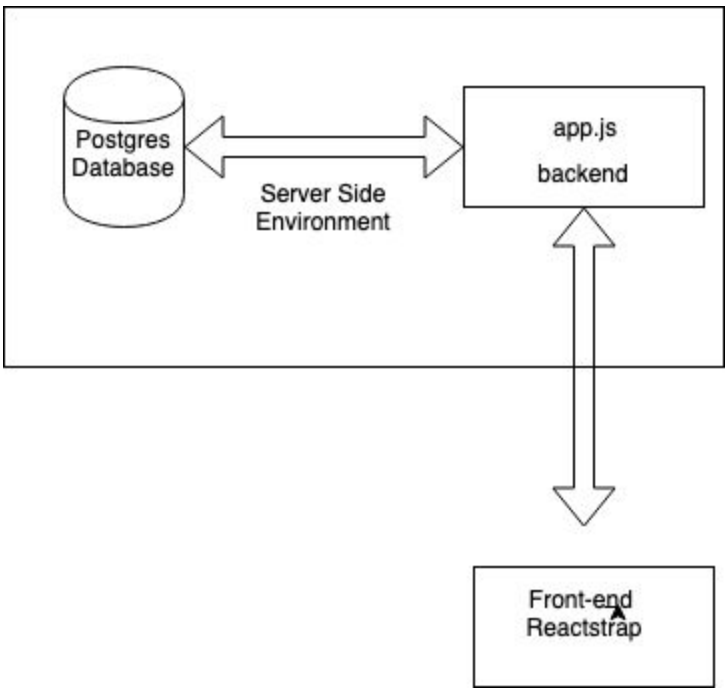
- We have changed the server service from AWS to Google App Engine.

UML Diagrams

Class UML:



Deployment Diagram :



Risk

Skills: Medium Risk. We are still learning many of the tool that we will be implementing. However, We are working together and sharing our knowledge of current tools to minimize our risk.

Schedule: No risk at the moment. We are currently working together inside and outside of class.

Technical: No risk at this time. We have switched servers from AWS to Google computing engine and have resolved technical issues.

Teamwork: No risk at this time. Teamwork is productive.

Legal: No risk at this time. We have not use any products that require legal or proper licensing/ copyright.

Project management

Our team will be managed using trello and google docs. In M2, our team used google documents to split up and assign specific task to each team member. As well, each team member was informed of the deadline of their assigned task. After assigning each task, each group members accepted the milestone 2 document via google docs and started working towards completing their task. Once each group member completed their assigned task, they will inform the document master to check their work. Moving forward onto milestone 3 to milestone 5 we will be using trello to assign task.