**Project Report**

**CMPE 202 Fall 2020**

**Professor: Gopinath Vinodh**

**Team 5**

**Group members:**

**Chanip Chong**

**Parshwa Gandhi**

**Jacob Rodriguez**

**TABLE OF CONTENTS**

**introduction 2**

**tools 2**

**requirements 2**

**functionalities 4**

**architecture 5**

**wireframes 7**

**uml diagrams 8**

**challenging and solutions 12**

**agile development 12**

**areas of contributions 13**

**project urls 14**

**reference 14**

## **Introduction**

This project is to build a housing website that offers users to rent, sell, or buy houses, in the meanwhile, protecting their sensitive information and ensuring the website is running 7/24. The requirements are given by professor Gopinath Vinodh. Our group implements both functional and non-functional requirements from scratch with frameworks Django and Bootstrap. The time for this project will be about two months, the deadline will be 2020/12/03.

## **Tools**

* MongoDB as database
* Django for backend
* Bootstrap for frontend (CSS/ javascript)
* AWS (server)
* Slack (communication)
* Zoom (Meetings)
* Google drive (documentation share)
* Github (version control)

## **Requirements**

1.Register/Login: Available for all roles.

2.Approve/Remove users: The administrator can approve, reject or remove any users if the

need arises.

3.Search: All types of roles can search for Sale or rental listings -based on zip code or street address, and/or other attributes –price range, sqft., #bedrooms, #bathrooms, carpet vs wooden flooring, home type (apartment/townhome/attached Single Family Home or Detached Single), open/closed parking, other amenities, Year built, etc.

4.Save as Favorites: Users can save Favorite searches and Favorite Homes

5.Sell:

* (Realtor or Seller)Upload details of the home to be listed ;
* You could list multiple homes if you are a realtor(acting on behalf of multiple sellers) ;
* Update status or other details of listing(s)Schedule open houses;
* Review buyers applications and approve/reject;
* Remove listing

6.Buy: (Buyer or Realtor) Apply with an offer for the home –an email sent to the Seller/seller’s realtor

7.Rent out(Realtor or Landlord): Add a new listing, Upload details of a home, Update lease terms, Availability date, Security deposit, schedule visits, Review renter’s application and approve/reject, update listing details, remove the listing

8.Rent(Renter): Apply for the lease, including credit score, employment information-an email sent to the Landlord/landlord’s realtor

9.Browser requirement: The application must be accessible from the google chrome browser on a laptop or mobile device. Mobile App is not expected.

10.The system should provide secure multi-user access, assuring correct concurrent behavior. The system should maintain suitable authorization information and validate access. User authentication should be implemented (by checking user id and password).

11.The system must have an easy-to-use user interface (UI) with screens designed for each part of the system’s functionality and suitable for different roles (Admin, User, Realtor)

B 12. The system should use a persistent data store(Relational or non-relational or both)

## **Functionalities**

1. authentication(sign up/ login)

2. administrator can approve, reject or remove any users

3. Searching filter

4. Save an item as a favorite

5. Upload details of the home

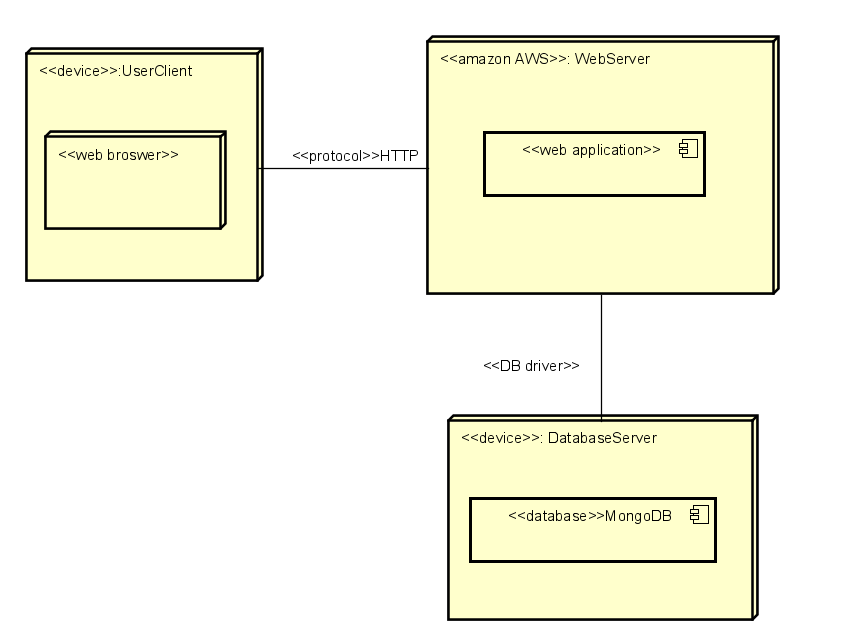
6. Update the status or other details of listing(s)Schedule open houses;

7. Review buyers applications and approve/reject;

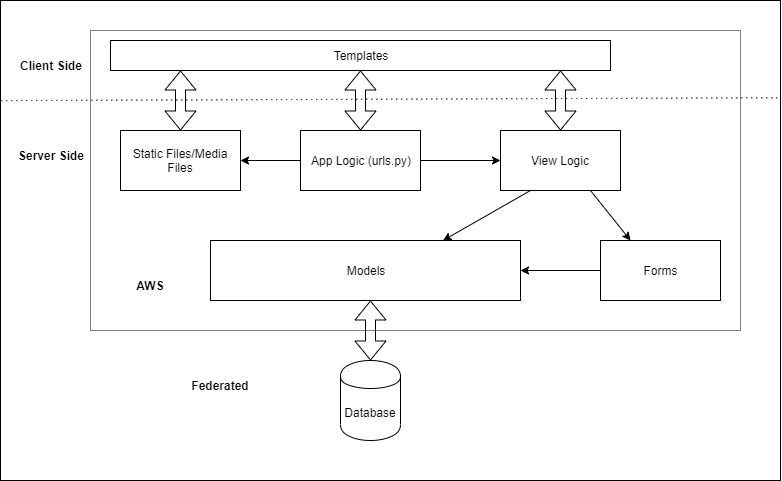
8. Remove a listing

9. Apply with an offer

## **Architecture:**

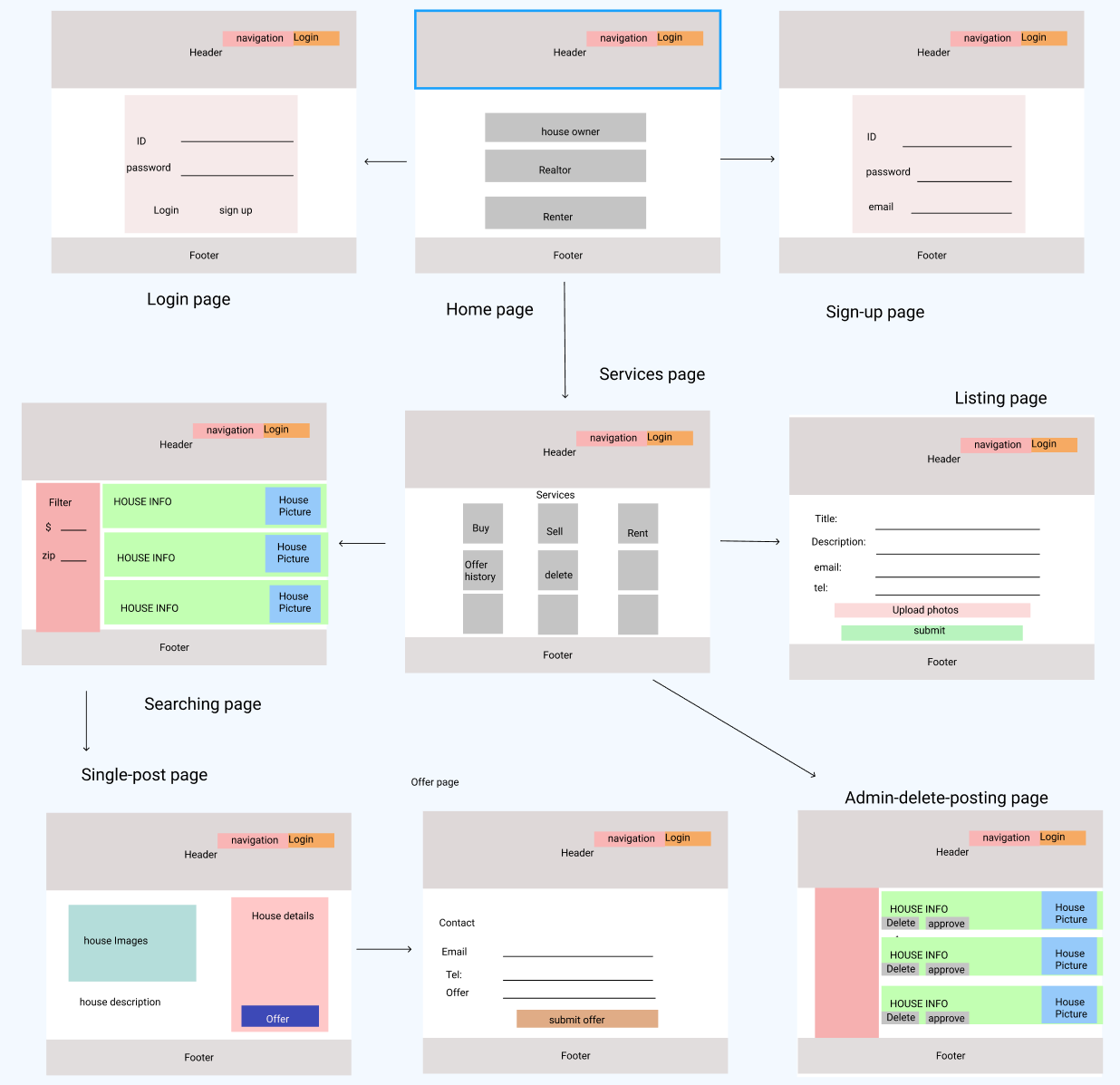


### **Dependency’s Diagram**



## 

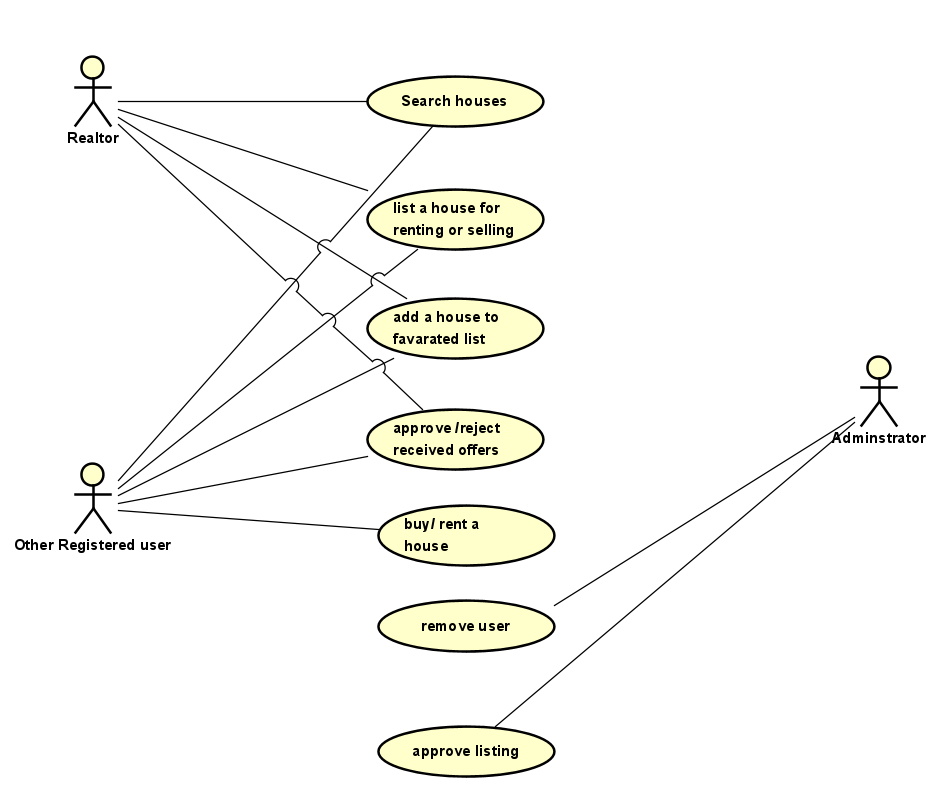
## **WireFrames**



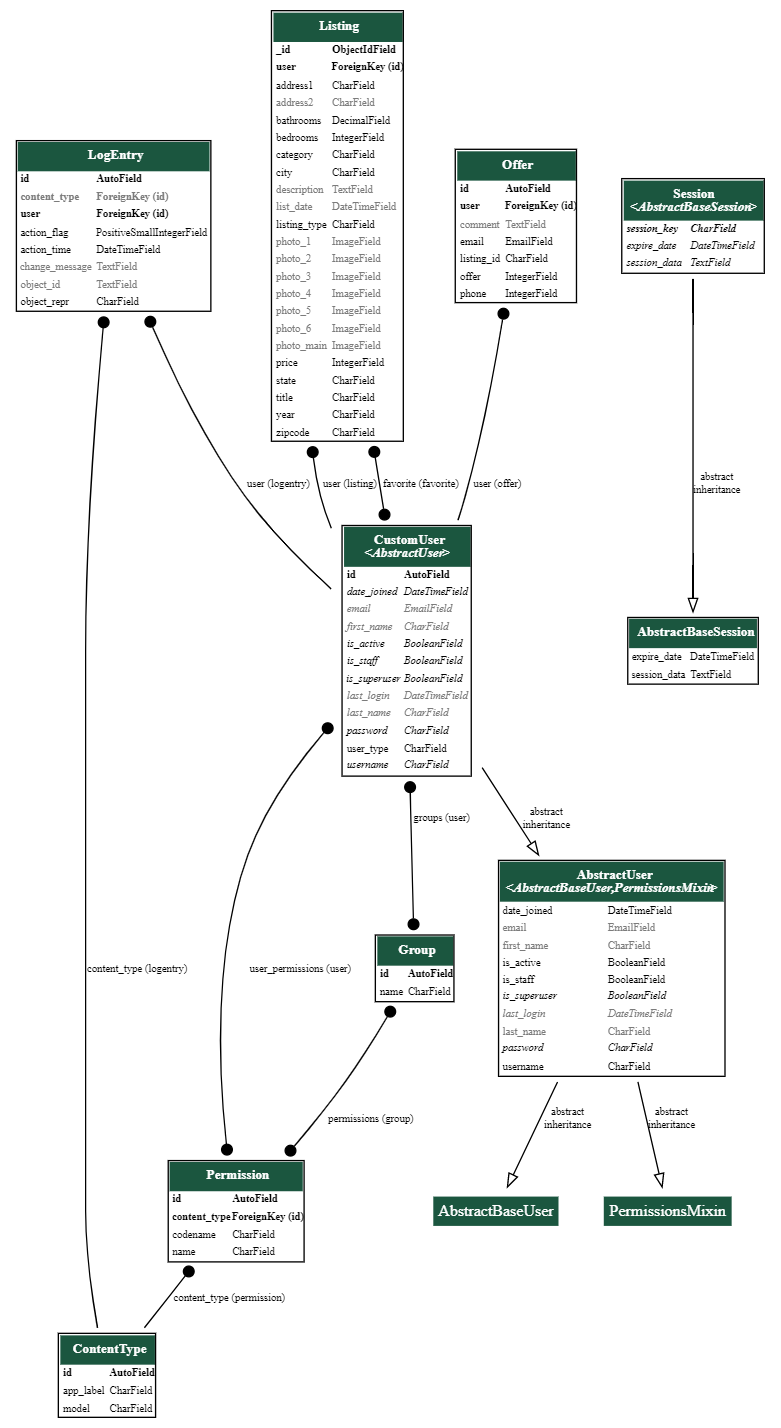
## 

## **UML diagrams**

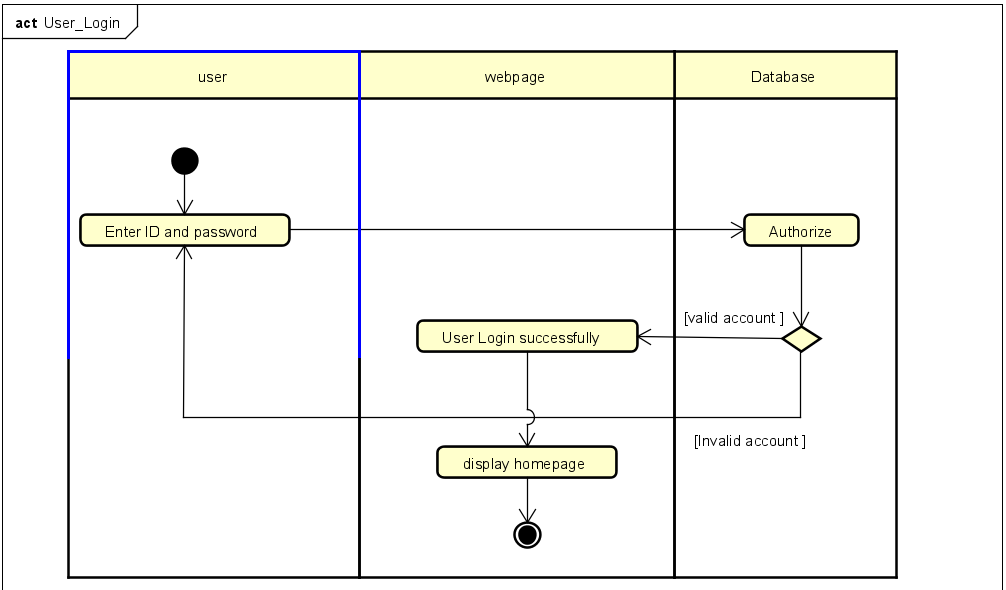
**Use case diagram**



**Class diagram:**

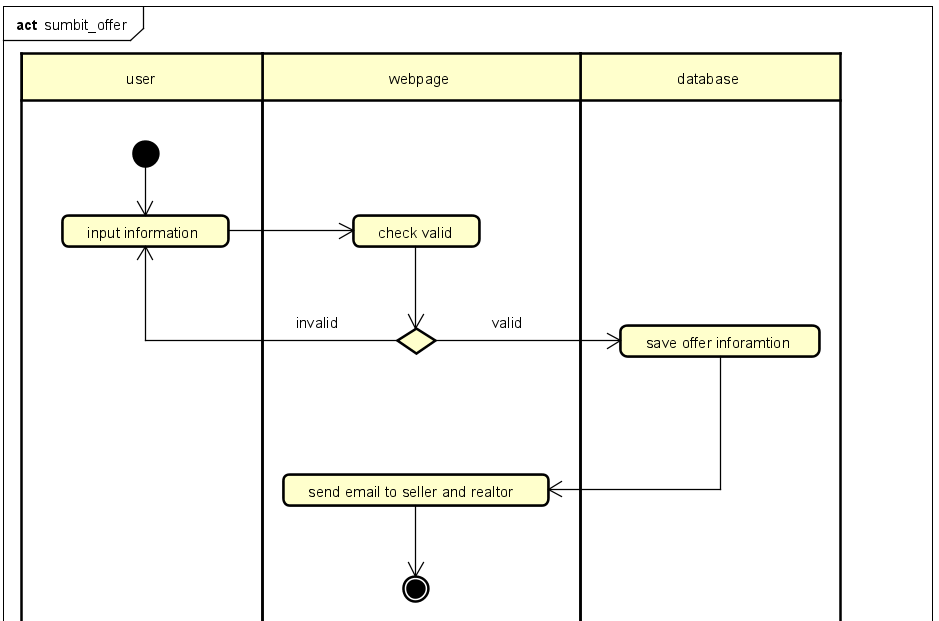
****

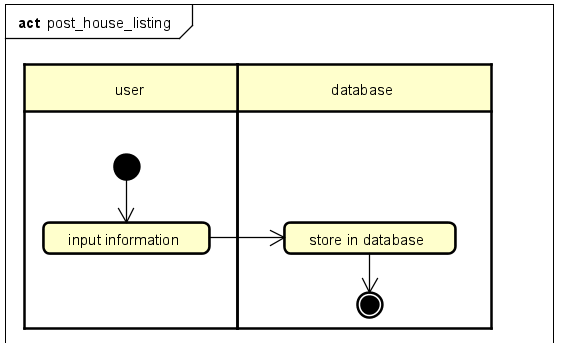
**Activity diagrams**



## 

## 





## **Challenges and Solutions**

1. We don’t have front-end development experience before, therefore, we use the bootstrap framework to build the HTML, CSS, and javascript. The advantages are quick and easy.
2. The overall architectural design was originally meant to be a React-Django REST-Framework so that the frontend team could make progress without knowing the implementation details of the backend, however, it was decided that we were not going to do React and instead just serve up static HTML files. The solution to this change was to instead run Django with its original functionality as serving templates to the client. This allowed us to keep the static HTML files as is and expand functionality on top of that as needed.
3. The use of the Djongo database engine for connecting to a MongoDB cluster proved to difficult at times since it is a relatively new engine type within Django and thus had little supporting documentation. To overcome this, our backend team worked collaboratively to sift through what documentation was available and combined our knowledge to gain a more in-depth understanding of the system as a whole.
4. Working on dependent features tended to be an issue towards the end of our creation process. For example, one member couldn’t complete the saving of a Listing to favorites until the other member was able to display a single Listing. To overcome these potential blocks, we mapped out key variable parameters that would be needed to accomplish the blocked-waiting feature and continued to work in parallel. When the initial feature was completed, only minor edits needed to be made to the existing code of the blocked-waiting feature.

## 

## **Agile development**

Our group decided to use 4 sprints(2 weeks) to finish this project. We maintain a scrump report and scrump task board on Github for everyone to see the progress. We use Slack to communicate with each other during this special time, also, We have a zoom conference for every sprint to discuss what is going to be done for that sprint. We decided to assign 7 hours/ week to develop the websites. The first 2 sprints will be preparation and design, and the rest of the time will implement both frontend and backend. Our backlog Item will be based on this schedule.

## 

## **Areas of Contributions**

Chanip Chong:

* Documentation (introduction, tools, requirements, functionalities, architecture ,wireframes, UML diagrams)
* front-end( 10 web pages of total 11 pages), except posting.html

Parshwa Gandhi:

* Research for the database to be used
* Features functionality
  + Send an email to the listing owner
  + View details of a single posting
* Creating posting.html page
* Documentation (Project Journal, Project Board)

Jacob Rodriguez:

* Initial creation of Django application
* Migration of Front-end files to Django app and configuration so that they could be served as templates
* Setup of MongoDB database instance as used by Django
* Feature functionality (New user, login authentication, Listing posting (Rent/Sell), Searching listings, Favorite listing, View favorites, View offers, Admin site)
* Deployment on AWS
* Documentation ( System design diagrams, README, Project Journal, Project Board, Burn-down chart)

## 

## 

## **Project URLs**

**GitHub Repo**

[**https://github.com/gopinathsjsu/fa20-cmpe-202-sec-03-team-project-team-5**](https://github.com/gopinathsjsu/fa20-cmpe-202-sec-03-team-project-team-5)

**Project Board**

[**https://github.com/gopinathsjsu/fa20-cmpe-202-sec-03-team-project-team-5/projects/1**](https://github.com/gopinathsjsu/fa20-cmpe-202-sec-03-team-project-team-5/projects/1)

**Project Journal**

[**https://github.com/gopinathsjsu/fa20-cmpe-202-sec-03-team-project-team-5/blob/master/Project%20Journal**](https://github.com/gopinathsjsu/fa20-cmpe-202-sec-03-team-project-team-5/blob/master/Project%20Journal)

**Google Sprint Task Sheet & burndown chart**

[**https://docs.google.com/spreadsheets/d/1hmaxy0rC4w1bO9iUSOJgVA3cTYgKkh3Lv9J3Y84oYuU/edit#gid=1401709108**](https://docs.google.com/spreadsheets/d/1hmaxy0rC4w1bO9iUSOJgVA3cTYgKkh3Lv9J3Y84oYuU/edit#gid=1401709108)

**Clearer pictures of diagrams**

[**https://github.com/gopinathsjsu/fa20-cmpe-202-sec-03-team-project-team-5/tree/master/Diagrams**](https://github.com/gopinathsjsu/fa20-cmpe-202-sec-03-team-project-team-5/tree/master/Diagrams)

## **Reference**

[www.Zillow.com](http://www.zillow.com)