SW Engineering CSC648/848 Spring 2020

The Garage

Group 06

Ray Rees Jr - Team Lead (rrees@mail.sfsu.edu)
Brad Peraza - Frontend Lead
Jiahong Zhan - Backend Lead
Joel Samaniego - Database Master
Mesoma Esonwune - Github Master
Roshni Varghese - Developer

Milestone 2
March 22, 2020

Revisions	
Milestone / Version	Date
Milestone 2 Version 1	3/31/2020

Table of Contents

Data Definitions V2

Data Name	Definition
Occupancy	A location on a map that has been created by a Host to offer Parking Spots for rent
• Parking Spot	The measure of what the app is selling. This indicates a place where the driver can park their car.
1. Available	a parking spot that is available for reservation.
2. Occupied	a parking spot that has been reserved by a different user.
3. Reserved	a parking spot that has been booked by the current user.
4. Blocked	a parking spot is blocked and cannot be occupied
Vehicle	Any automobile of a supported size. This can include different sub-categories that indicate the size of said vehicle.
• Categories	
1. Motorcycle	an automobile with two wheels.
2. Compact	a smaller automobile that will fit in tighter spaces. (Volkswagen Jetta)

3. Full Size	a standard size automobile. (Toyota Camry)
4. SUV	larger automobiles that require more space (Chevrolet Suburban)
Users	All users of The Garage. Each sub-classification is not mutually exclusive from others, meaning a user can be both a Host and a Guest.
• Host	a user that posts a parking spot for reservation by Guests. This user has privileges to create parking spot listings, edit parking spot listings, and report Guests.
• Guest	a user with a registered vehicle that is looking for a parking spot. This user has privileges to access the database to search for parking spots, report parking spots, and reserve parking spots.
• Admin	a user with moderation privileges. This user has the capability to take action on hosts, Guests, and parking spots. Given the nature of this role, this will likely only consist of Garage employees.
• Rating	the credibility of any registered user using The Garage.
Web Application	The visual presentation of The Garage.
• Listing	the visual representation of the parking spot posted by a host.
• Time Table	a visual representation of the reservation availability.

Functional Requirements V2

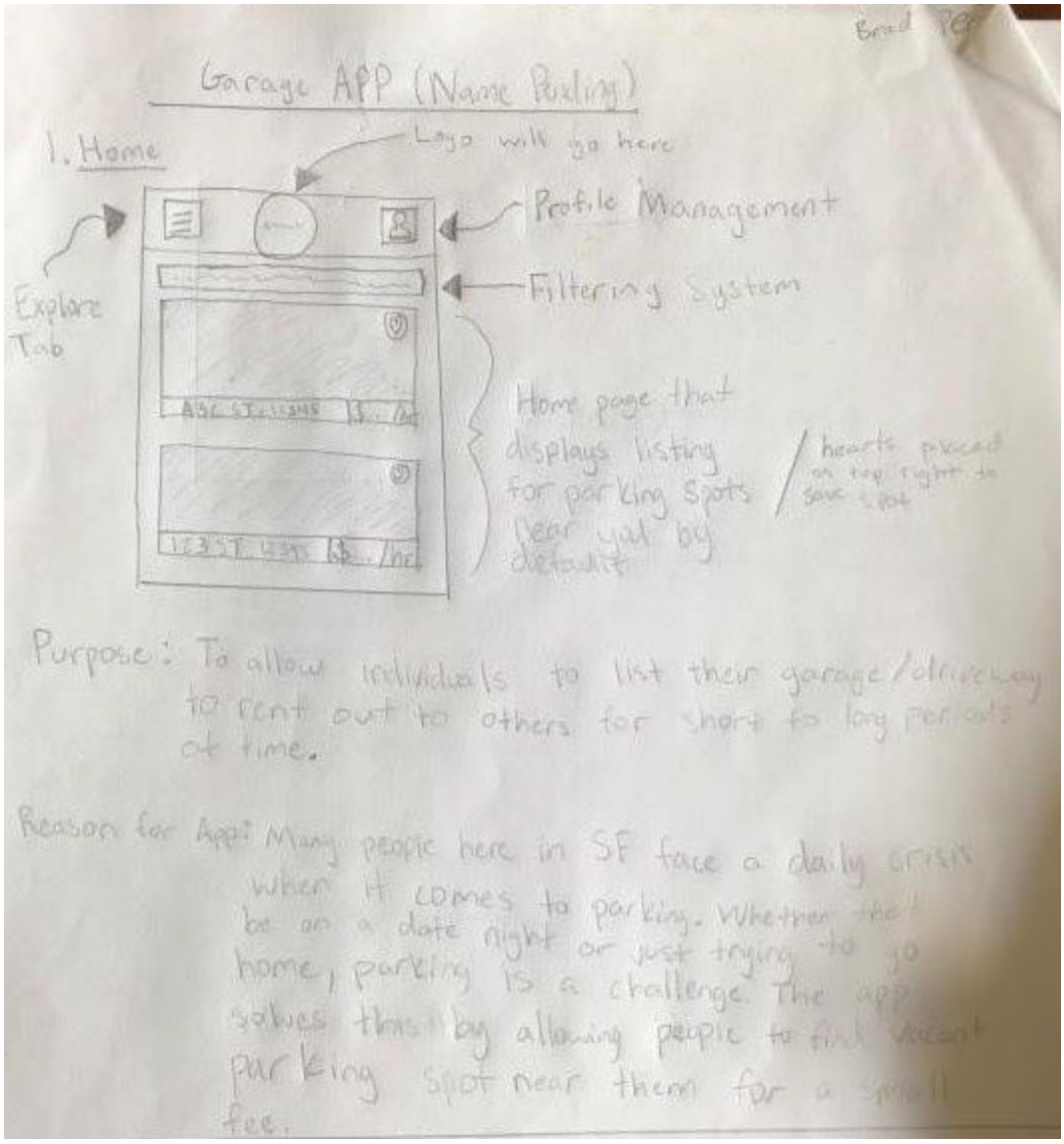
requirement	**Priority**
Users:	
1. Users shall be able to create an account.	**1**
2. The system shall enable the user to reset their password.	**3**
3. The system shall enable users to edit their own information.	**1**
4. Users shall be able to view additional information about the listing.	**1**
5. Users shall be able to search a location to view available spots nearby.	**1**
6. Users shall be able to reserve a parking spot	**1**
7. Users shall be able to filter the search result based on price range.	**2**
8. Users shall be able to filter the search result based on distance.	**2**
9. Users shall be able to filter the search result based on vehicle size.	**2**

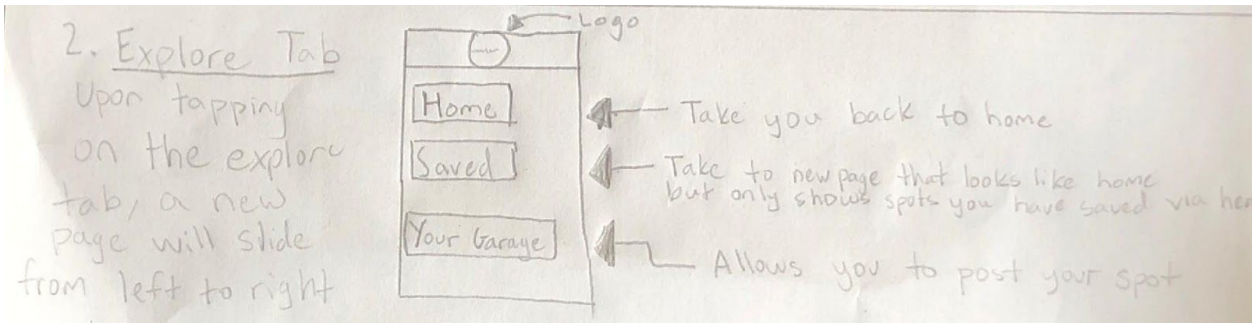
10. Users shall be able to sort listings by availability.	**1**
11. Users shall be able to filter the search result based on availability.	**2**
12. The system shall enable users to view a report of fees.	**3**
13. The system shall enable users to view a report of rental fee income.	**3**
14. The system shall send a Confirmation email when a user signs up.	**2**
Guests:	
15. Guests shall be able to cancel the parking spot booking.	**1**
16. Hosts shall be able to cancel the parking spot booking.	**1**
17. Guests and Hosts shall be able to rate each other.	**2**
18. Guests shall be able to review payment summaries.	**2**
19. Guests shall receive a confirmation email when they have Reserved a Parking Spot.	**1**
20. Guests shall be able to view previously booked spots.	**2**
21. Guests shall be able to interact with the parking spot's respective time table during reservation.	**1**

22. Guests shall be able to view the dates and times of previous bookings.	**1**
23. Guests shall be able to manage vehicles under their account	**1**
24. Guests shall be able to extend more time for parking.	**2**
25. Guests and Hosts shall be able to communicate with each other regarding the booked parking spot.	**3**
26. Guests shall receive an alert when the end time is approaching.	**2**
Hosts:	
27. Hosts shall be able to upload their own available parking spots.	**1**
28. The system shall be able to verify the parking spot Host's ID.	**1**
29. Hosts shall be able to adjust the pricing of their parking spot.	**1**
30. Host shall be able to change the availability status of a parking spot.	**1**
31. Hosts shall be able to remove their own listings from the application	**1**
32. Hosts shall be able to upload pictures of their parking spot listing.	**1**

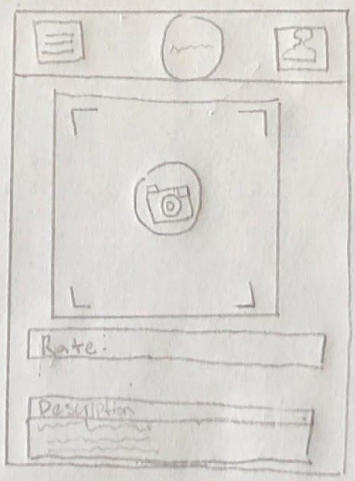
33. Hosts shall be able to rate Guests.	**2**
34. Hosts shall be able to set user viewing restrictions based on rating. e.g; A user with a 3.5 star rating looks for a parking spot, but cannot see Jim's listing, as he set his spot to only be visible to people with a 4 star rating or above.	**2**

UI Mockups and Storyboards





3. Your Garage



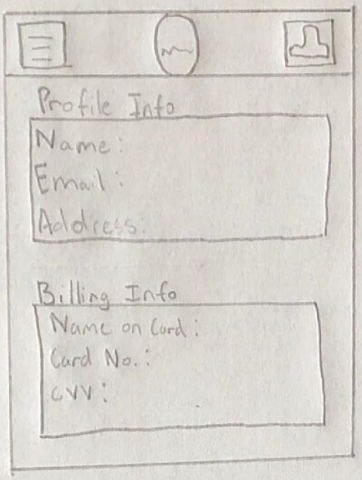
allows you to upload a photo of your spot

Set price of the spot

Write a small description of the spot

4. Profile Management

Upon pressing the Profile Management a new screen will load up.



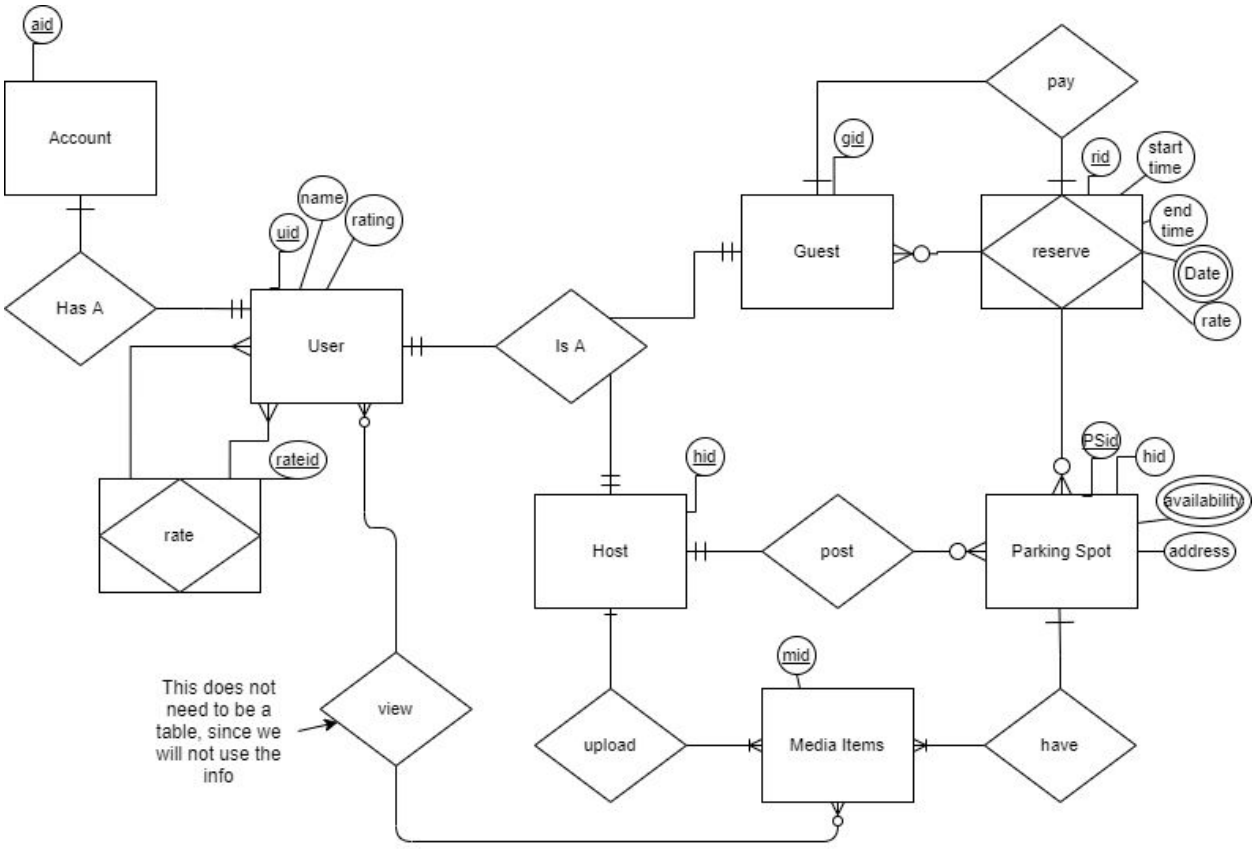
Update and manage both Profile / Personal info and billing info.

High Level Database Architecture and Organization

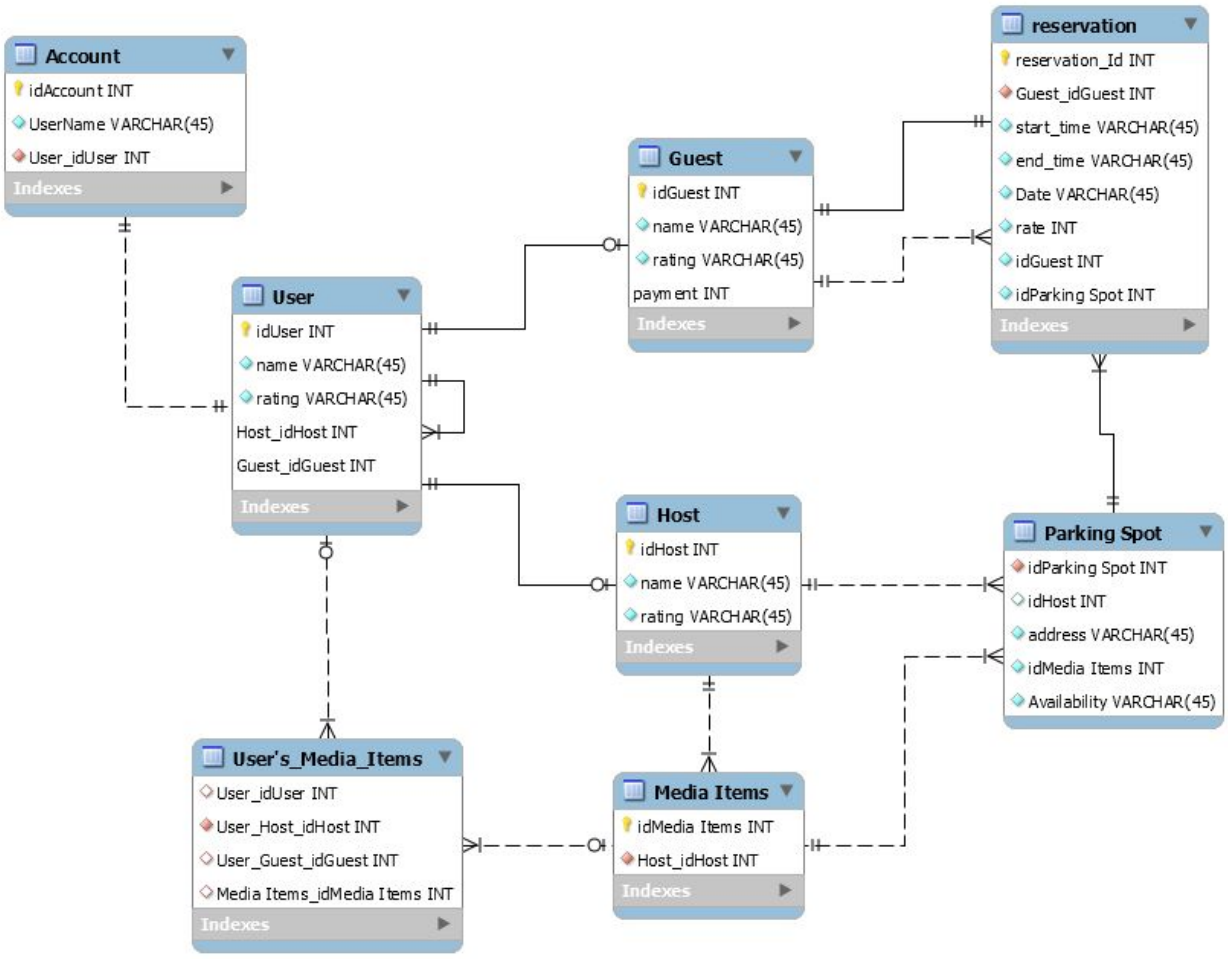
Business Rules:

- Hosts can post zero or many parking spots.
- Guests can reserve zero to many parking spots, and must pay after submission
- A user can have only one account
- Guests and Hosts are users
- User accounts can have only one rating
- Hosts can upload media items
- Parking Spots have media items attached
- Users can view any parking spot's media items

Entity Relationship diagram



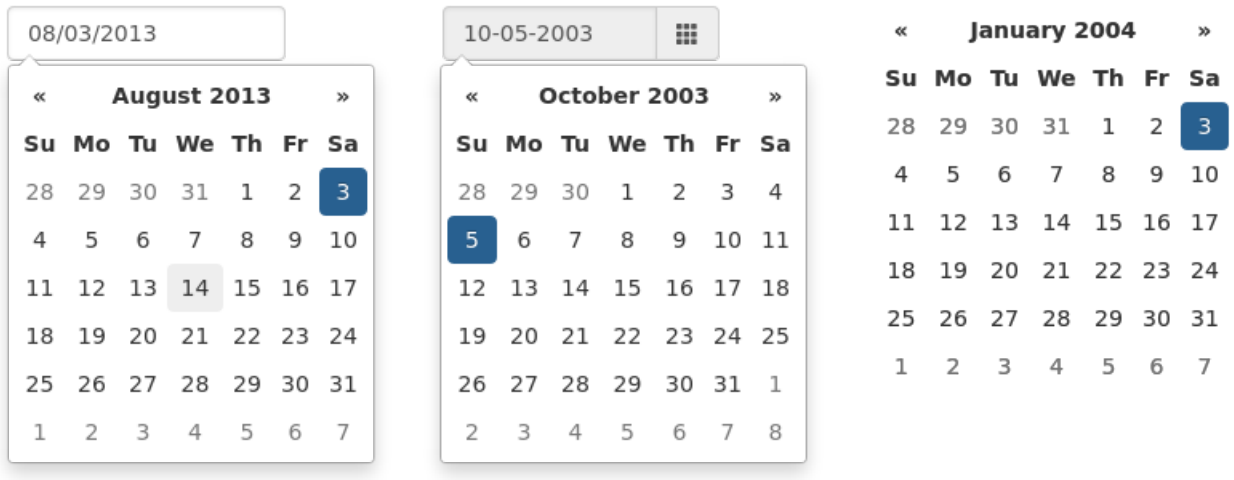
Database Model



In this project, we will be using mySQL because having a relational database will make searching easier for our customers. We do not expect a huge follower base right off the bat, so the less-scalable mySQL will work perfectly. In addition, our team is used to using mySQL, so this takes less work to develop.

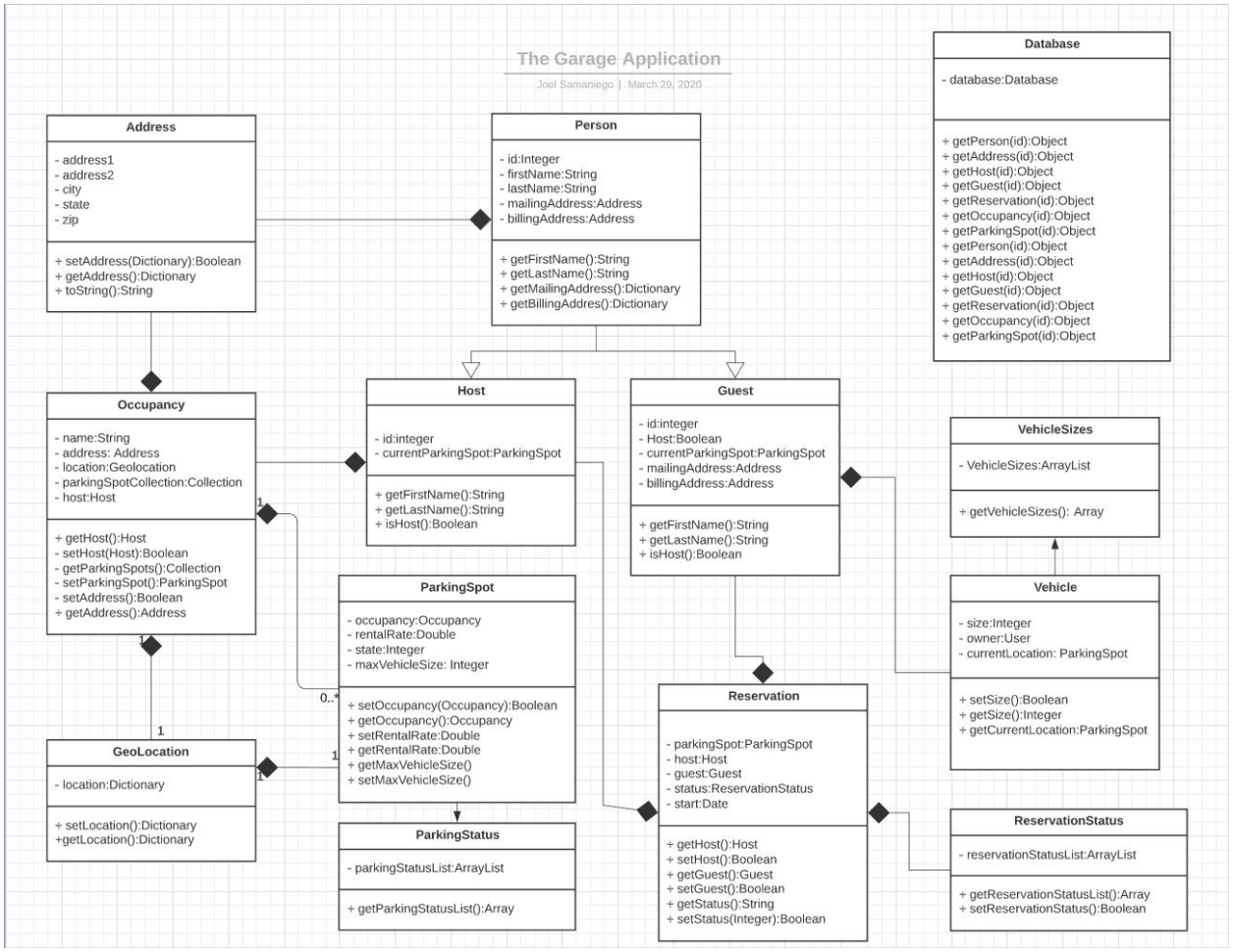
High Level APIs and Main Algorithms

We are planning on using the Bootstrap datepicker to select specific dates and times for booking a listing. For example, see image below.

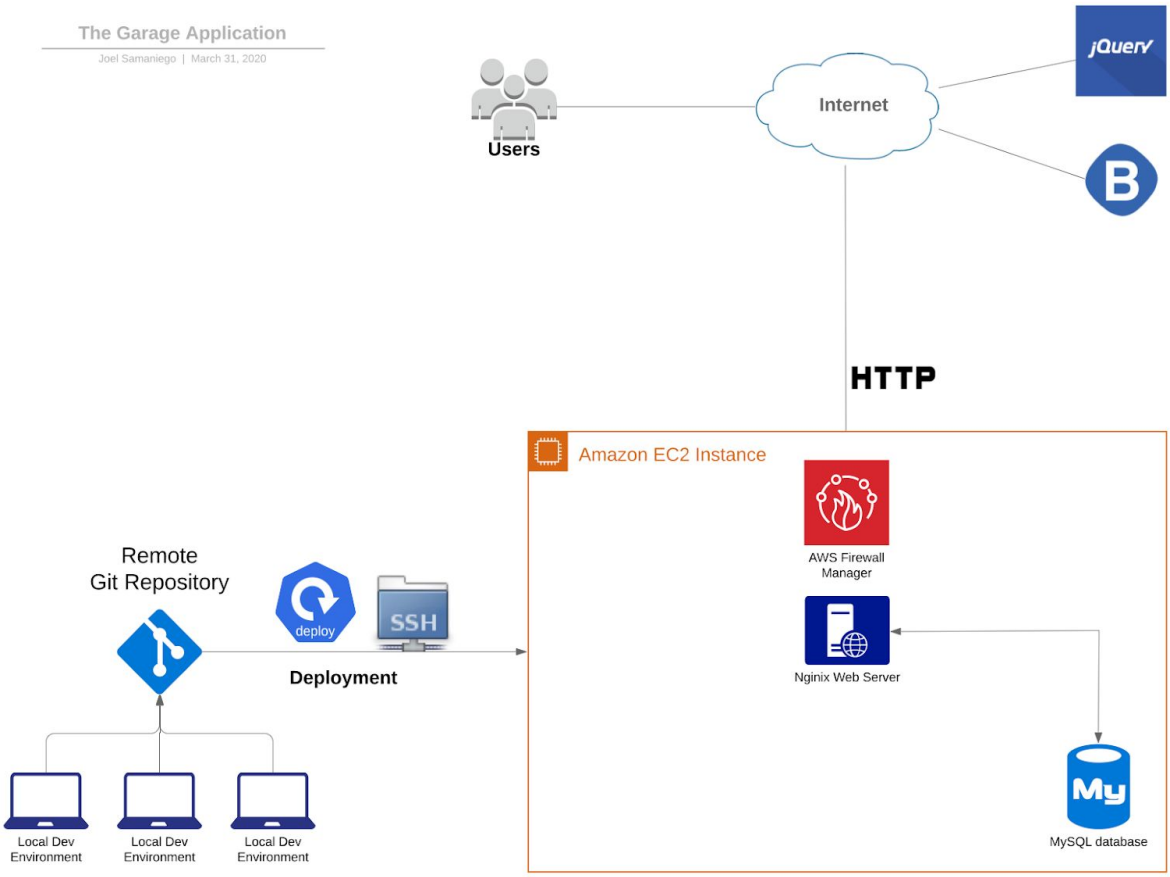


The main APIs used in our application would be bootstrap and jquery.

High Level UML Diagram



High Level Application Network and Deployment Diagram



Identify Actual Key Risks

Skills risks -

A lot of us do not have a ton of experience with mySQL. Since this is our database language of choice, it is a risk that is unavoidable.

We just need to practice as we implement the features promised from the milestones. Practice will mitigate this skill risk.

Schedule risks -

This is possibly the biggest risk, as the Covid-19 outbreak is still at large, making each and every one of us stuck at home.

We need to reassess our schedules and figure out a definite time for each of us to meet offline.

Technical risks -

We are not entirely sure how the database will handle each and every available time slot, but have some solid ideas on how to tackle it. If we plan to store each time slot in the database, the data usage will be much higher, potentially making thousands of entries for a single user.

We are planning to use some fancy maths and javascript APIs to make a workaround. At this point, however, we are still a bit unsure on how to fully solve it.

Teamwork risks -

Up until now, most of the work has been purely documentation, so there has been minimal coding involved, but our teamwork could use some brushing up on. There have been multiple cases of procrastination and poor communication, team lead included. Other times a team member read a document or request too fast, leading to loss of information, and misinterpretation.

Not being afraid to ask for help or advice on a task would lead to everyone being on the same page, and minimize the teamwork risk involved. Also asking others what their status on an issue that is blocking thiers will lead to better streamlining of workload.

Legal / Content risks -

Since we originally planned on making a system that verifies a person's ID, this would make quite a few legal problems if done incorrectly. In addition, we would be listing a spot which normally would count as private property.

We have the plan to research how Airbnb tackles this problem, and draw up a legal document similar to theirs. Other than that, we still are unsure of how to solve this problem.

Project Management

We managed M2 tasks by splitting up the workload as much as possible. All of this was by word of mouth and a small chart of who did what, so naturally it caused some issues with deadlines and responsibility. Having a formal task management system will hopefully iron out those issues. We plan to use something like Trello in the future to implement the features promised in the milestones.

Breakdown of Tasks -

UML Diagram	Joel	V1 completed
Network Diagram	Joel	V1 completed
Vertical Prototype	Brad, Roshni, Mesoma	In process
ERD and Database Model	Ray	V1 completed
Milestone 2 Document	Jiahong, Joel & Ray	V1 completed
UI Mockups	Brad	V1 completed
APIs and Main Algorithms	Ray	V1 completed?
Data Definitions V2		V1 completed