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 1 Version 1	2/27/2020
Milestone 1 Version 2	3/21/2020
Milestone 2 Version 1	3/31/2020
Milestone 2 Version 2	04/01/2020

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 1:	
1. Users:	
1.1	Users shall be able to create an account.
1.2	The system shall enable users to edit their own information.
1.3	Users shall be able to view additional information about the listing.
1.4	Users shall be able to search a location to view available spots nearby.
1.5	Users shall be able to reserve a parking spot
1.6	Users shall be able to sort listings by availability.
1.7	users shall be able to reserve parking spots by adding payment information.
1.8	Users shall be able to filter the search result based on price range.

1.9	Users shall be able to filter the search result based on availability.
1.10	Users shall be able to filter the search result based on vehicle size.
1.11	Users shall be able to filter the search result based on distance
2. Guests:	
2.1	Guests shall be able to cancel the parking spot booking.
2.2	Guests shall receive a confirmation email when they have Reserved a Parking Spot.
2.3	Guests shall be able to interact with the parking spot’s respective time table during reservation.
2.4	Guests shall be able to view the dates and times of previous bookings.
2.5	Guests shall be able to manage vehicles under their account
2.6	Guests shall be able to edit their payment information
2.7	Guests shall be able to leave the comment for the hosts when guests make a reservation.
2.8	Guests and Hosts shall be able to rate each other.

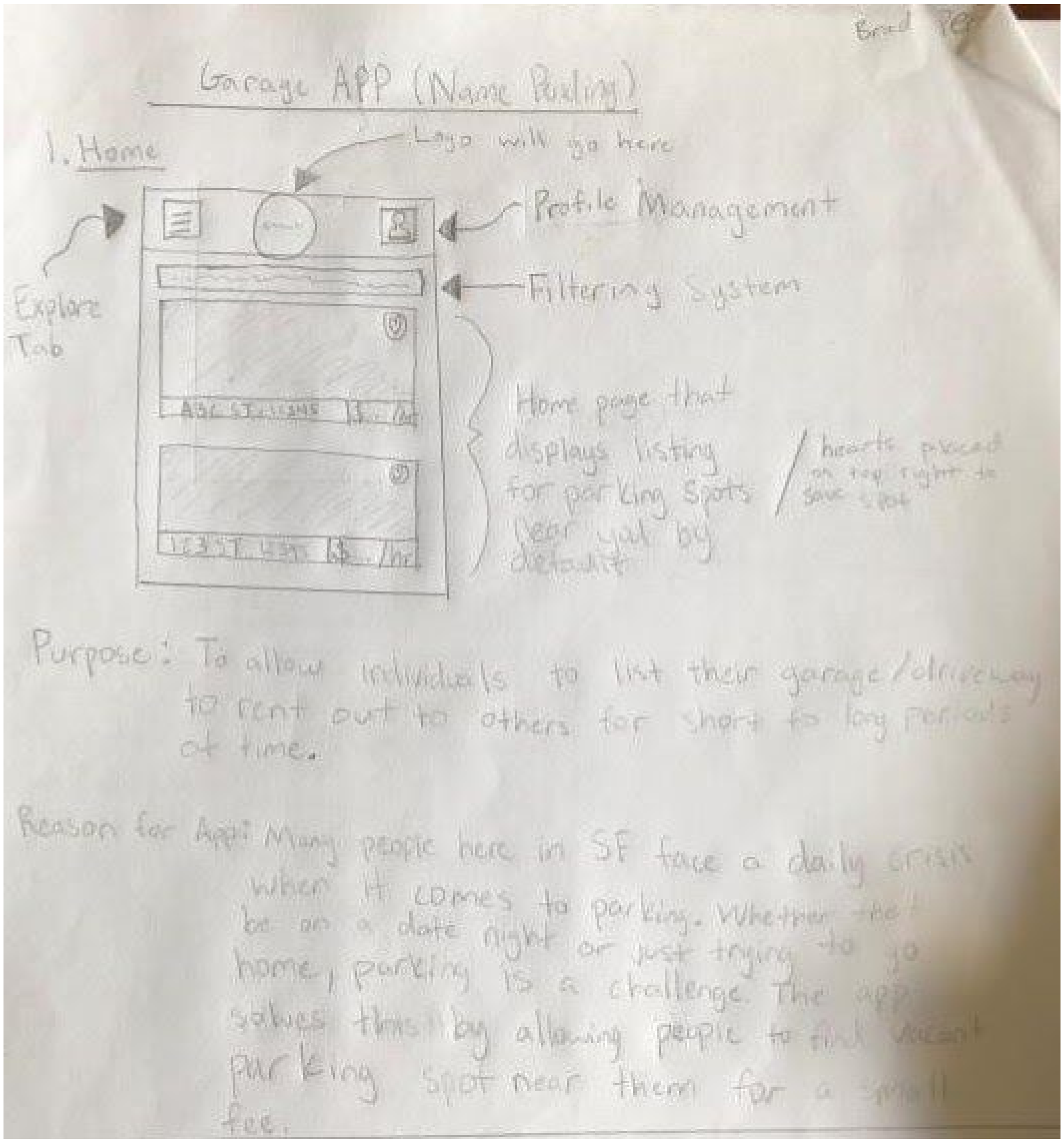
2.9	Guests shall be able to review payment summaries.
2.10	Guests shall be able to view previously booked spots.
2.11	Guests shall be able to extend more time for parking.
2.12	Guests shall receive an alert when the end time is approaching.
3. Hosts:	
3.1	Hosts shall be able to upload their own available parking spots.
3.2	The system shall be able to verify the parking spot Host's ID.
3.3	Hosts shall be able to adjust the pricing of their parking spot.
3.4	Host shall be able to change the availability status of a parking spot.
3.5	Hosts shall be able to remove their own listings from the application
3.6	Hosts shall be able to upload pictures of their parking spot listing.
3.7	Hosts shall be able to cancel the parking spot booking.
3.8	Before a reservation can take effect, hosts must confirm it.

3.9	Hosts shall be able to rate Guests.
3.10	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.
Priority 2:	
4. Users:	
4.1	Users shall be able to receive an email when they signed up
4.2	Users shall be able to contact the customer support
4.3	Users shall be able to save the parking spots as a mark, such as my favorite, my office or my home.
4.4	The system shall be able to edit/remove any listing.
4.5	The system shall be able to block or delete users' accounts, if for some reason, such as breach of user agreement.
4.6	The system shall be able to handle refunds or bill changes requirements.
5. Guests:	
5.1	Guests shall be able to rebook their parking spots in the history.

5.2	Guests shall be able to apply for refund transactions.
5.3	If guests encounter any security issues in the parking spots, they shall be able to contact the system immediately for help.
6. Hosts:	
6.1	Hosts shall be able to change the reservation time if the time that guests choose doesn't work for them.
6.2	Hosts can leave a comment for cancel reservation for the customer to see
6.3	If a guest's vehicle has any problem, hosts shall be able to send an emergency message to the guest and system.
Priority 3:	
7. Users:	
7.1	The system shall enable the user to reset their password.
7.2	The system shall enable users to view a report of fees.
7.3	The system shall enable users to view a report of rental fee income.
7.4	Users are warned by system when their ratings are too low.
7.5	Users shall be able to sort by security.

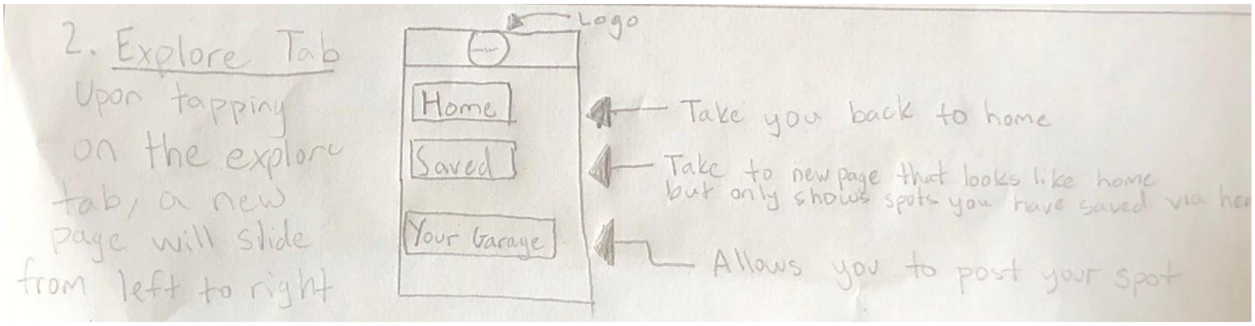
8. Guests:	
8.1	Guests and Hosts shall be able to communicate with each other regarding the booked parking spot.
8.2	Guests shall be able to subscribe to the same parking spots for a longer period of time. For example, subscribe to one-year parking spots from 9am to 5pm each weekday.
8.3	Guests shall be able to sort the collected parking spots according to the distance, rating, price and security.
9. Hosts:	
9.1	Hosts shall be able to use the money from the parking spots as collateral for his rental of other parking spots.
9.2	Hosts shall be able to edit the parking spots for one time rental or subscribe rental.
9.3	Hosts shall be able to add restrictions to their parking spots, such as prohibiting guests with ratings below 1 from renting their spot.

UI Mockups and Storyboards


Brad Pe
Garage APP (Name Pending)
1. Home
Logo will go here
Profile Management
Explore Tab
Filtering System
ABC ST. ...SMS $ /hr
123 ST. ...SS $ /hr
Home page that displays listing for parking spots near you by default
hearts placed on top right to save spot

Purpose: To allow individuals to list their garage/driveway to rent out to others for short to long periods of time.

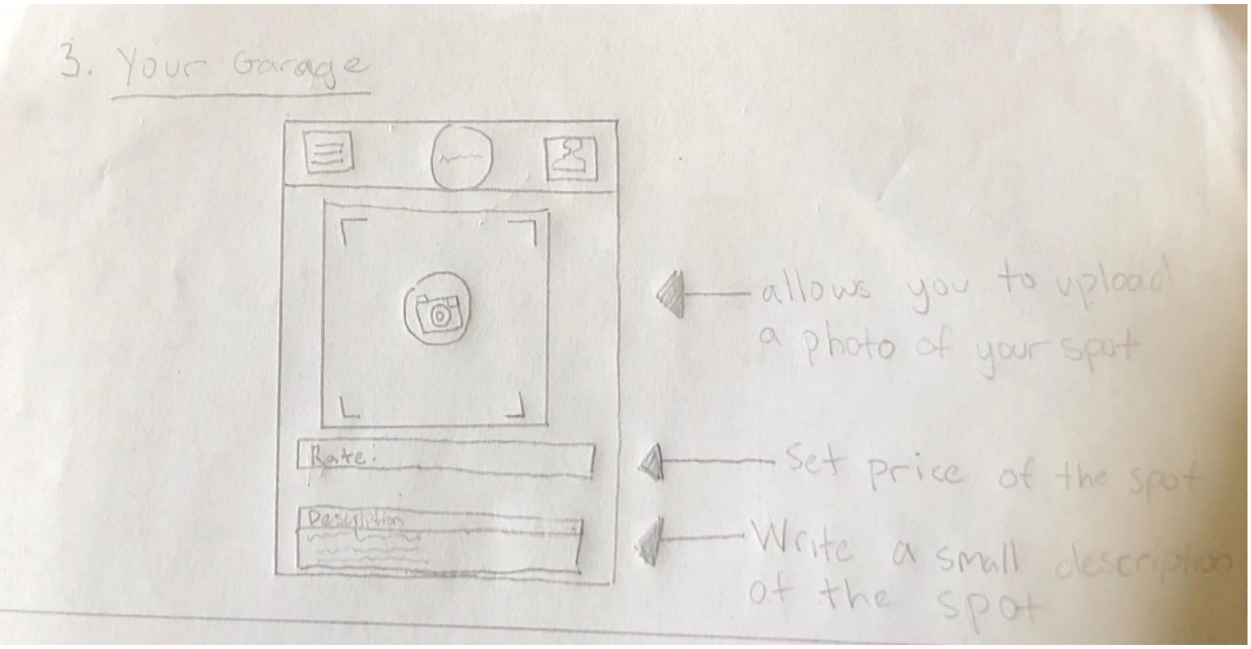
Reason for App: Many people here in SF face a daily crisis when it comes to parking. Whether the be on a date night or just trying to go home, parking is a challenge. The app solves this by allowing people to find vacant parking spot near them for a small fee.


2. Explore Tab

Upon tapping on the explore tab, a new page will slide from left to right

Logo
Home
Saved
Your Garage
Take you back to home
Take to new page that looks like home but only shows spots you have saved via hea
Allows you to post your spot

3. Your Garage

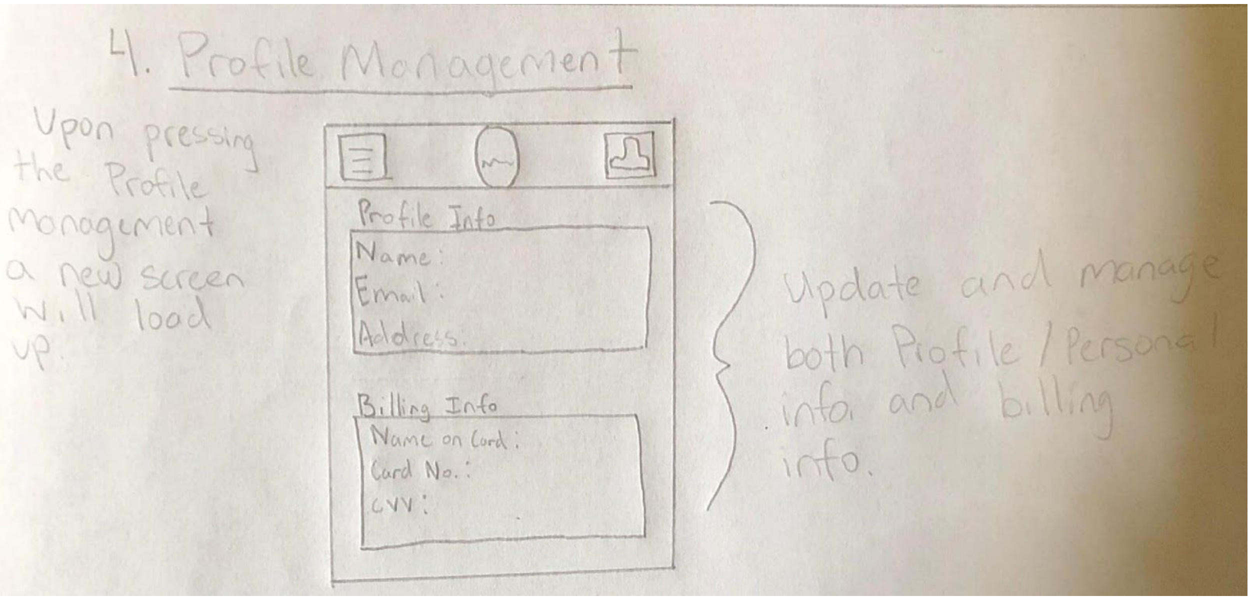


allows you to upload a photo of your spot

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:

1. Unregistered users must create one account into the website for reserving or posting parking spots.
2. Hosts can post more than one parking spot at the same account.
3. Guests cannot reserve more than two parking spots at the same time, and must pay after submission.
4. A user only logs into one account at a time. If they want to switch accounts, they will be logged out of the original account automatically.
5. Users can change state to either guest or host.
6. Users must have the Contact information.
7. All users must have ID information if they want to be a host.
8. User accounts must and can have only one rating. The default rating for all users is the maximum.
9. Hosts can upload many parking spot media items.

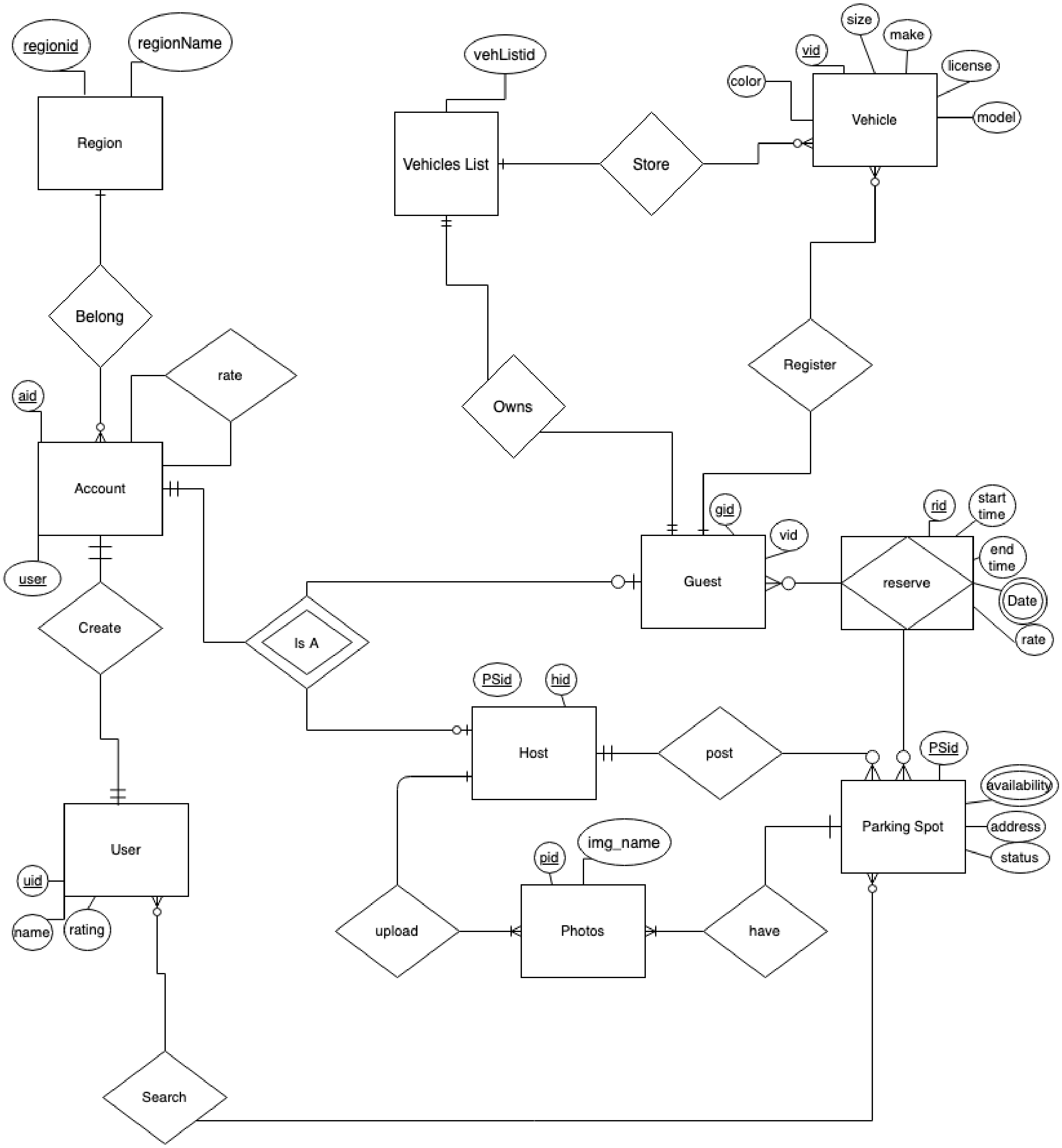
Entities Description and Attributes:

1. **Occupancy** - A location on a map that has been created by a Host to offer Parking Spots for rent
    a. **Parking Spot** - The measure of what the app is selling. This indicates a place where the driver can park their car.
        i. **Available** - a parking spot that is available for reservation.
        ii. **Occupied** - a parking spot that has been reserved by a different user.
        iii. **Reserved** - a parking spot that has been booked by the current user.
        iv. **Blocked** - a parking spot is blocked and cannot be occupied
2. **Vehicle** - Any automobile of a supported size. This can include different sub-categories that indicate the size of said vehicle.
    a. **Motorcycle** - an automobile with two wheels.
    b. **Compact** - a smaller automobile that will fit in tighter spaces. (Volkswagen Jetta)
    c. **Full Size** - a standard size automobile. (Toyota Camry)
    d. **SUV** - larger automobiles that require more space (Chevrolet Suburban)
3. **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.
    a. **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.

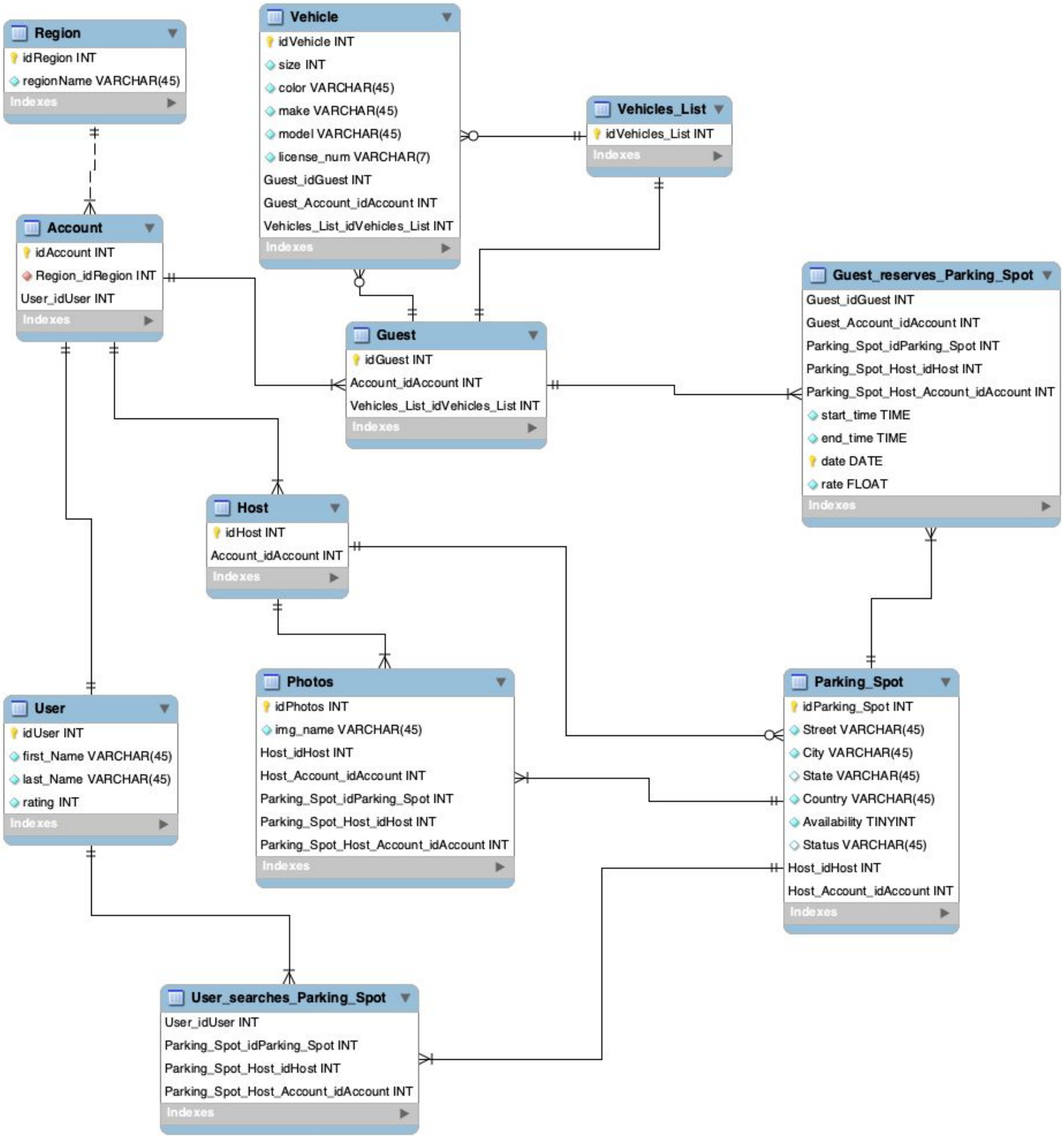
   b. **Guest** - a user with a registered vehicle that is looking for a parking spot. This user has privileges to access the database to search for parking spots, report parking spots, and reserve parking spots.
   c. **Admin** - a user with moderation privileges. This user has the capability to take action on hosts, Guests, and parking spots. Given the nature of this role, this will likely only consist of *Garage* employees.
   d. **Rating** - the credibility of any registered user using *The Garage*.

4. **Web Application** - The visual presentation of The Garage.
   a. **Listing** - the visual representation of the parking spot posted by a host.
   b. **Time Table** - a visual representation of the reservation availability.

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

Database Query API

This API will provide GET, POST, PUT, DELETE methods for the database entities that are outlined in the ERD. This API will be consumed via HTTP request by the Front End, in order to provide functionality to our site.

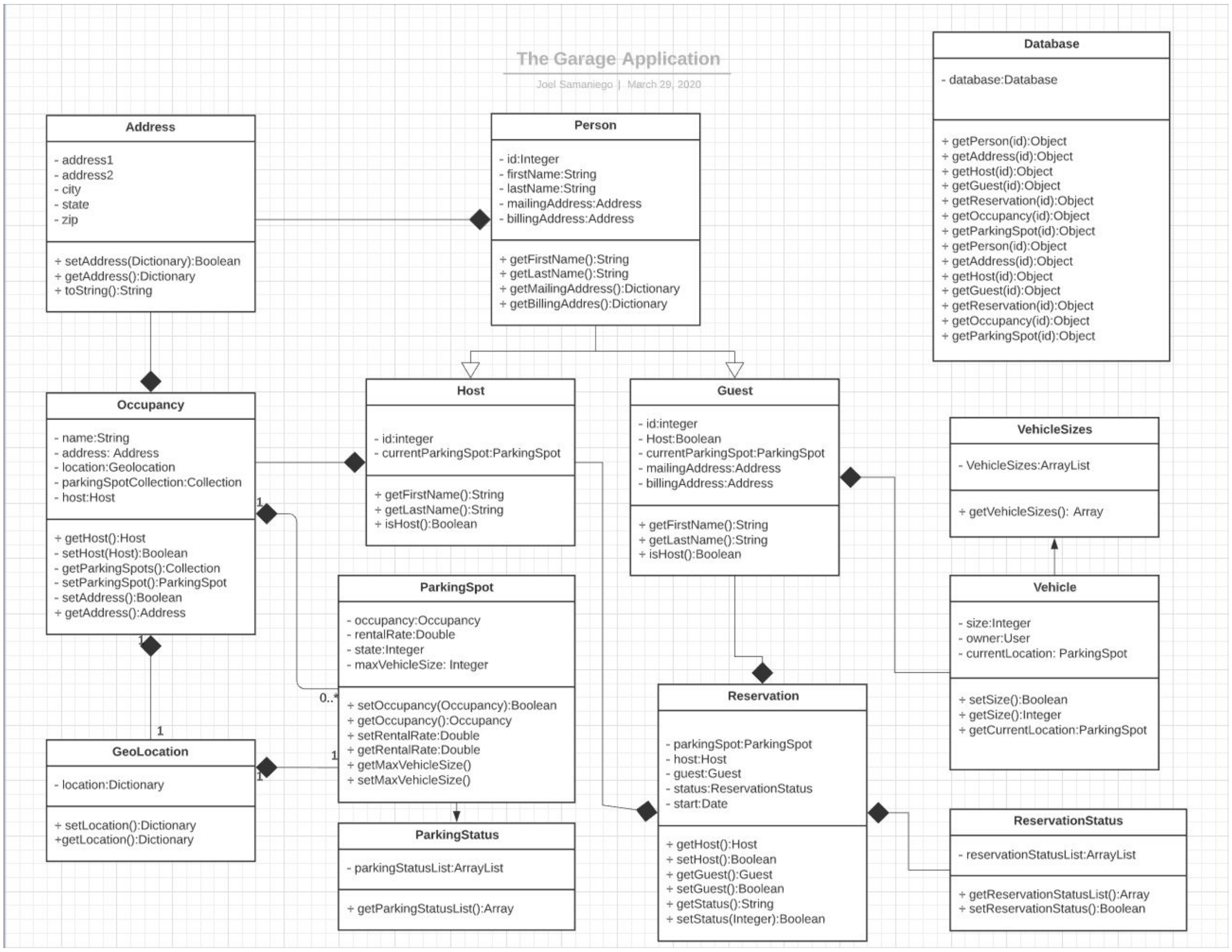
Some examples of the API endpoints are:

/parking-spot/
/reservation/
/host/
/guest/

Recommended Parking Spots Algorithm

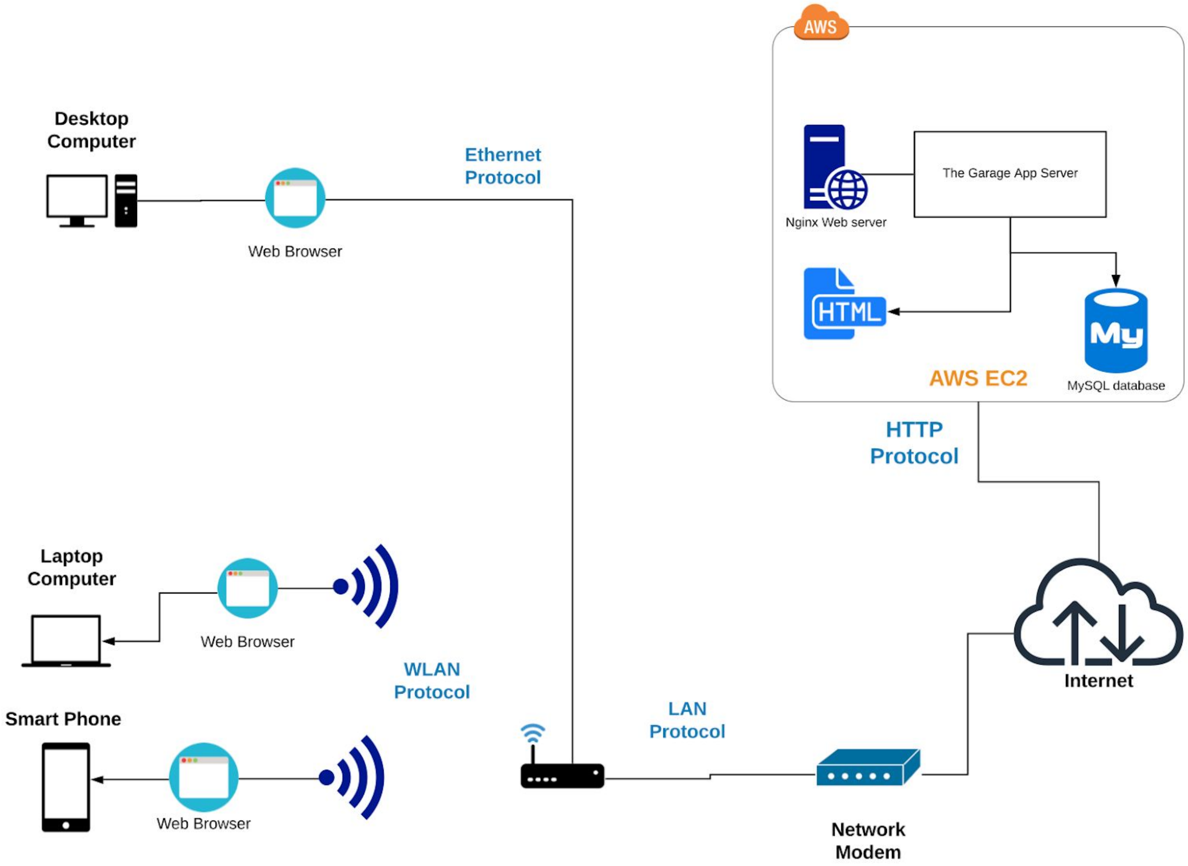
Retrieving available Parking Spots will require that we compare all Parking Spots that fall within the defined range of the desired location, which meet the vehicle size for the Guests needs, selecting only for parking spots that aren’t already allocated to a Reservation during the desired timeframe and listing them based on their proximity to the desired location, secondly by their review rankings.

High Level UML Diagram

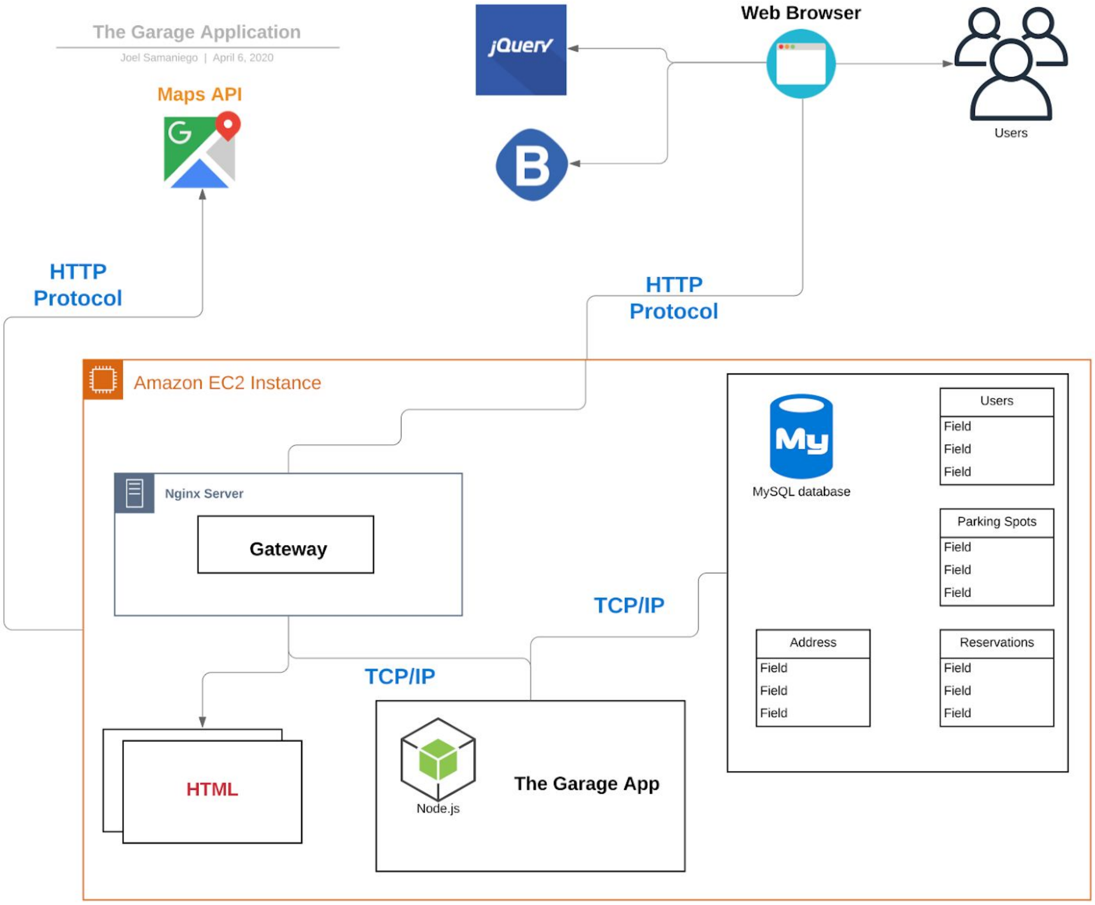


High Level Application Network and Deployment Diagram

Network Diagram



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 theirs 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	In process
ERD and Database Model	Ray, Mesoma	V2 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