SW Engineering CSC648/848 Spring 2019

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 1
February 27, 2020

Revisions	
Milestone / Version	Date
Milestone 1 Version 1	2/27/2020
Milestone 1 Version 2	03/21/2020

Table of Contents

Executive Summary

The Garage is an application that brings a revolutionary approach to the way we look for parking. One of the most frustrating tasks for many individuals, especially those that live in densely populated areas, is finding parking near your destination. Parking, whether that be at work, hitting the city for a night out, or in your neighborhood, can seem like a daunting task when there simply isn’t enough parking to go around. Both public and private sectors have taken advantage of this parking crisis with cities increasing the number of meters that rely on coins; an inconvenient form of payment for most, or privately-owned parking structures that charge outrageous prices.

The Garage aims to bring new resources to this need by offering a database of private parking spots, specifically privately-owned garages or driveways. Allowing users to go onto our web application, view all the spots that are listed near their destination, then select the parking spot that meets their needs (i.e. availability, vehicle size, and time). What makes our application unique is that we allow our users to post their garage or driveway onto our database, allowing our users to make extra money when they aren't using their spot. This allows us to continue to grow our inventory of parking spots while incentivizing users to continue to use the application.

Main Use Cases

Finding a Parking Spot

Unregistered User

James is a student at San Francisco State University. They commute to campus everyday and have a hard time finding parking near campus. They find The Garage Application website and decide to check it out. James goes into the Search feature to see what parking spots are available around campus. James finds a parking spot that is not too expensive and close enough to campus and selects that parking spot. The website asks if they want to create an account. James decides no and is taken to the next page where they are asked for their email address and asks how long they want to book the parking spot. The system checks to make sure that there is no conflict in the length of time and if there is no conflict asks for James' form of payment. Once James enters their payment method and clicks Book, the system will send a confirmation email to James saying that they have the booking.

Registered User

Melissa is also looking for parking spots near their destination. They go on The Garage Application website and select a parking spot. The website asks if they want to create an account. Melissa decides yes and inputs their email address and creates a password. They are taken to the next page where they are asked for how long they want to book the parking spot. The system checks to make sure that there is no conflict in the length of time and if there is no conflict asks for Melissa's form of payment. Once Melissa enters their payment method and clicks Book, the system will send a confirmation email to Melissa saying that they have the booking.

Registering a Parking Spot

Unregistered Host

Adam lives near San Francisco State University. They notice that a lot of students need to park and decide that they want to rent out their parking spot to people. They will go to The Garage Application. The website will ask if they have an account. Adam does not so they create an account on the website. Adam cannot rent out a parking space without an account.

Registered Host

Adam now has an account on The Garage Application. They will log into their account and select the Host version. The website will ask for the address they want to register, a photo of the parking spot, how many parking spots they can provide and how much they want to charge. Clicking next takes Adam to the main page.

User Registration

Beatriz is a business woman in SOMA and has two vacant parking spots in her building. With commercial real estate leases being so expensive, Beatriz is excited to capitalize on these parking spots. Beatriz decides to sign up with the Garage App to offer these two parking spots to professionals in her neighborhood.

Creating Parking Spots at an Occupancy

Beatriz lists the number of parking spots she wishes to offer on the Garage App. She's able to list her two vacant parking spots in her building. Guests will be able to see Beatriz's listing when they search for available parking spots in SOMA.

Configuring a Parking Spot

Mary, a Grad Student at SFSU rents a Studio that includes a small parking spot. Mary would still like to earn some extra income, so she uses the Garage App to offer her Parking Spot to Guests with small vehicles.

Configuring a Vehicle

James commutes to school in his Ford Focus. He can usually find street parking but he's often stuck having to park several blocks away. This increases the time it takes Jams to get to class. James' Professors have commented on his tardiness. James decides to use the Garage App to look for parking near school. James only needs a small parking spot, which he can configure in the Garage App. This allows James to connect with Hosts near school, which will help get him to class on time.

Viewing the Parking Spot Status

Mary is excited to see if she's had any hits on her listing on the Garage App. Mary uses the Garage App to see if anyone has made a reservation for her parking spot. When her spot has been reserved, she will see an update in her account.

Cancelling a Reservation

Beatriz has clients coming into her office next week, but she notices that her two parking spots have been reserved on the Garage App. Beatriz decides to cancel the reservations so that she has space for her clients when they come into her office. Thankfully the Garage App Guests will receive an email notification of the unfortunate need for the cancellation.

Reserving a Parking Spot

James is excited to use the Garage App. He's planning his commute to school next week and he decides to use the Garage App to reserve a parking stall. He searches for parking spots near his destination that are available on the day he'll be in class and that meet the size needs for his vehicle. He selects a spot that meets his needs from the search results and reserves the spot. James is relieved that he won't be late to class because he knows parking won't be an issue this time.

Changing the Status of a Parking Spot to Blocked

James arrives at the parking spot that he's reserved, only to find that it's blocked by a dumpster! James uses the Garage app to flag the parking spot as blocked. Fortunately, he's able to find a spot close by. Hopefully this delay won't make James tardy again.

Checking out of a Parking Spot

James has had quite a day. He's glad that he at least was able to find a spot nearby. It's now time to head home. James uses the Garage app to check out of the parking spot he reserved for his Ford Focus, and he's now heading home.

Requesting Payment

Mary has been able to rent out her parking spot and she is excited to see that extra income. She uses the Garage App to request payment from the users that rented her parking spot.

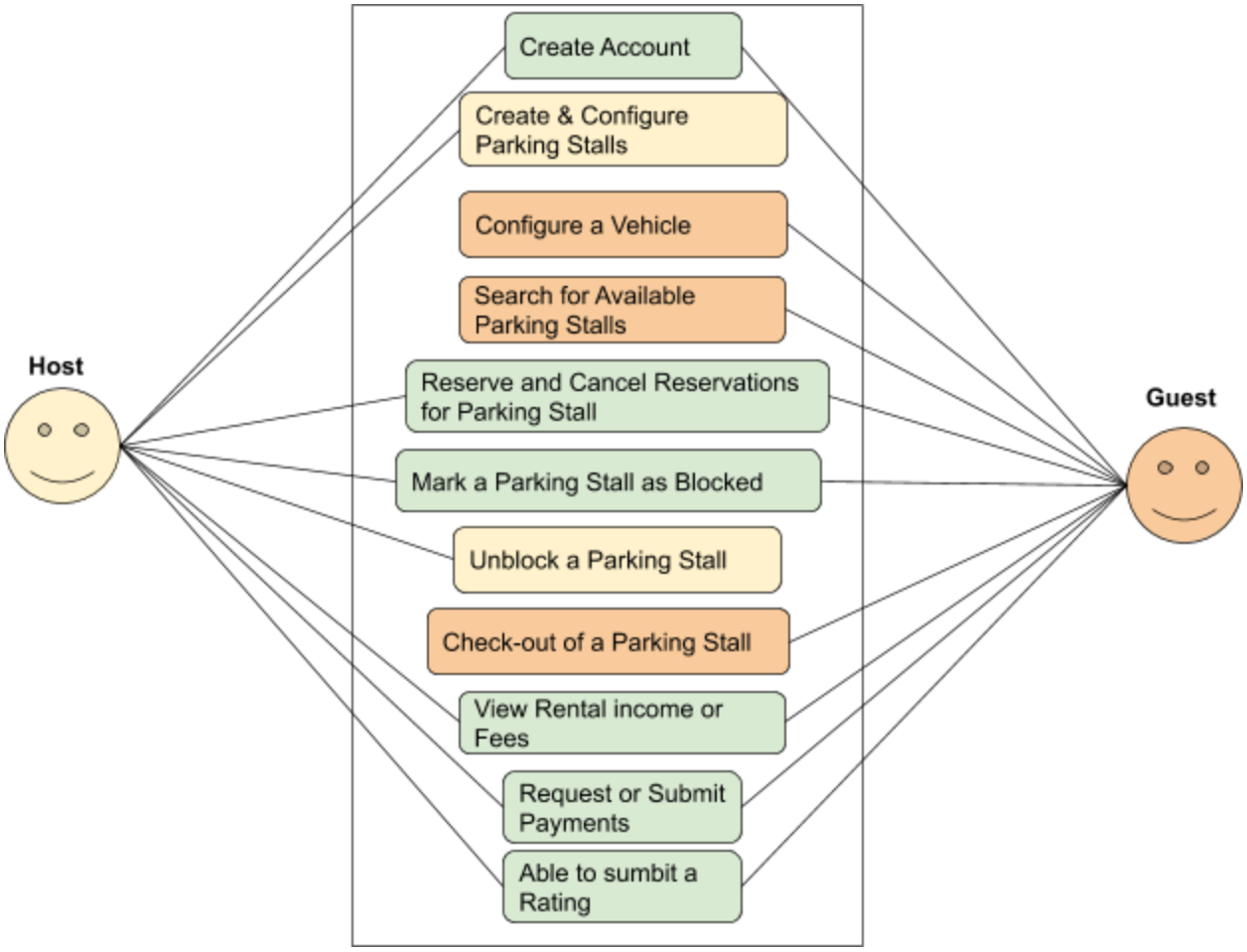
Submitting Payment

James is glad that he was able to find a solution to his parking problem and is no longer tardy to class. James uses the Garage app to submit payment for the parking spot reservations that he's used.

Submitting a Rating

James is really pleased with Mary's parking spot and the proximity to school. James uses the Garage App to give Mary a 5 star rating. He hopes this will help Mary find other tenants for the days that he's not traveling to school.

Use Case Diagram



Competitive Analysis

Feature / Company	Spot Hero	BestParking	ParkMe	The Garage
pricing	$13.00/hr	$30.00/hr	$10.00/hr	$10.00/hr
weakness	Mix public and private parking	Mix public and private parking	information is not updated in time	Low user base
strengths	There are mobile applications, users are more convenient	A large number of parking lots, Users can have more choices	Cheap price	Cheaper prices, more safety, and your parking space is closer to your destination
Social media	Google promotion	Google promotion	Google promotion	Instagram，Twitter.

Features

Feature / Company	Spot Hero	BestParking	ParkMe	The Garage
Location Search	++	+	+	+
Timeframe Search	+	+	+	+
Map Search Results	+	+	+	+
Social Sign-On	++	+	+	-

iOS App	+	+	+	-
Residential Parking	-	-	-	++
Commercial parking	+	+	+	+

+ Feature exists; ++ Superior ; - does not exist

As listed in the above chart, *The Garage* offers lower prices and more security than its competitors. At the same time, the parking spots we offer are similar to Airbnb's private rental rooms but more convenient than public parking lots. In other words, as long as there are people living near your destination, we can guarantee the highest parking convenience. An advantage over our competitors lies in private car parking which is safer than public parking. Our weakness is that we have a smaller user base than the other three sites. Therefore, early publicity and word of mouth is particularly important. But for this shortcoming, we have an advantage compared to other sites. The campus should be our potential service objects since our team members are SFSU students. Because of this, we fully understand that convenience and affordability are crucial to the success of a college student's career. Our competitors do not understand the struggle of finding a parking spot or risk being late to class by parking far away.

We connect with our users because we are our users.

Main Data Items and Entities

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.

Available

- a parking spot that is available for reservation.

Occupied

- a parking spot that has been reserved by a different user.

Reserved

- a parking spot that has been booked by the current user.

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

Motorcycle

- an automobile with two wheels.

Compact

- a smaller automobile that will fit in tighter spaces. (Volkswagen Jetta)

Full Size

- a standard size automobile. (Toyota Camry)

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

Users

1. The system shall send a Confirmation email when a user signs up.
2. The system shall enable users to view a report of rental fee income.
3. The system shall enable users to view a report of fees.
4. The system shall enable the user to reset their password.
5. Users shall be able to create an account.
6. Users shall be able to sort listings by price.
7. Users shall be able to sort listings by distance.
8. Users shall be able to sort listings by availability.
9. The system shall enable users to edit their own information.
10. Users shall be able to search a location to view available spots nearby.
11. Users shall be able to view additional information about the listing.
12. Users shall be able to reserve a parking spot
13. Users shall be able to filter the search result based on price range.
14. Users shall be able to filter the search result based on distance.
15. Users shall be able to filter the search result based on vehicle size.
16. Users shall be able to filter the search result based on availability.

Guests

17. Guests shall receive an confirmation email when they have Reserved a Parking Spot.
18. Guests shall be able to extend more time for parking.
19. Guests shall receive an alert when the end time is approaching.
20. Guests shall be able to interact with the parking spot's respective time table during reservation.
21. Guests shall be able to manage vehicles under their account
22. Guests shall be able to view previously booked spots.
23. Guests shall be able to view the dates and times of previous bookings.
24. Guests shall be able to review payment summaries.
25. Guests and Hosts shall be able to rate each other.
26. Guests shall be able to cancel the parking spot booking.
27. Hosts shall be able to cancel the parking spot booking.
28. Guests and Hosts shall be able to communicate with each other regarding the booked parking spot.

Hosts

29. Hosts shall be able to upload their own available parking spots.
30. The system shall be able to verify the parking spot Host's ID.
31. Hosts shall be able to upload pictures of their parking spot listing.
32. Hosts shall be able to adjust the pricing of their parking spot.
33. Host shall be able to change the availability status of a parking spot.
34. Hosts shall be able to remove their own listings from the application
35. Hosts shall be able to rate Guests.
36. 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.

Non-Functional Requirements

1. The system shall charge the user if they overstay.
2. The system will not process passwords without a set password strength.
3. The system will not allow the user to proceed with account creation until requirements are met
4. The system will prompt the user as to which requirements are not met during form filling
5. The system will not record the user's location if permission is not granted
6. The system will not record the user's payment information if permission is not granted
7. Users under 18 will not be able to reserve parking spots
8. When payment information submitted by the user is wrong, the system will hold the reserved parking spots for the user for 30 min.
9. All monetary amounts must be accurate to two decimal places
10. Password shall never be viewable at the point of entry or at any other time unless prompted
11. The system will not allow hosts to access users' payment methods.
12. The system will not allow guests to change hosts' fee amount.

Coding Standards

- Javascript shall be written in adherence to the Google Javascript Style Guide

System Requirements

Compatibility

The application shall be compatible with the following Operating Systems

- Windows X
- macOS
- Ubuntu (Linux)

Browser Support

- The application shall run on the latest version of modern desktop browsers
    - Chrome
    - Firefox
    - Safari

- Microsoft Edge

Performance Requirements

Error Rate

- The system shall maintain a daily error rate less than 1%

Availability

The system shall provide an up time no less than 98%

Response Time

- Should not exceed 1 second for UI interactions
- Should not exceed 5 seconds for query functions

Workload

Scenario	Daily Total	Pages	Think time
Account Sign up	100	Homepage	10 seconds
Login	200	Homepage	2 seconds
Search for parking spot	1000	Search page	5 seconds
Reserve Parking spot	200	Search page	0.5 seconds

Scalability

- The system shall be designed in order to allow for scalability to meet the increased demands placed on the system

Capacity

- The system shall have a capacity to manage all of the workload values

Storage Requirements

- The application environment shall provide 50gb of storage for database records

Security Requirements

- Communications shall use HTTP protocol
- The system shall authenticate users credentials for validity before granting access to user data
- DDoS protection shall be provided (AWS Shield)
- Sensitive information shall not be stored in Cookies
- Any Cookies used shall have an expiration date
- X XSS Protection shall be enabled

Marketing Requirements

- Social Media shall be used to promote the Garage App.
- Google Ads shall be used to target the Garage App to it’s target demographics

Privacy Requirements

- Usernames and passwords will be collected and stored for authentication purposes
- User emails will be collected and used to communicate with users
- User addresses will be collected in order to provide services at these locations

High Level System Architecture and Technologies Used

Server Host: Amazon Elastic Compute Cloud (Amazon EC2) t2.micro
Operating System: Ubuntu Server 18.04.3 LTS
Database: mySQL 14.14 Distrib 5.7.29 for Linux
Web Server: nginx 1.14.0
Server-Side Language: Javascript
Additional Technologies:

Javascript Library: jQuery 3.4.1
IDE: IntelliJ
Web Analytics: Amazon Kinesis Data Analytics

Team Member Responsibilities

Ray Rees Jr	Team Lead Document Master Front-end Team Member (Primary)
Brad Peraza	Frontend Lead
Jiahong Zhan	Backend Lead Front-end Team Member
Mesoma Esonwune	Github Master Back-end Team Member Front-end Team Member (Primary)
Roshni Varghese	Front-end Developer (Primary)
Joel Samaniego Campos	Database Master Backend Team Member (Primary) Front-end Team Member

Checklist

Status	Description
Done!	Team found a time slot to meet outside of the class
Done!	Github master chosen
Done!	Team decided and agreed together on using the listed SW tools and deployment server
Done!	Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing
Done!	Team lead ensured that all team members read the final M1 and agree/ understand it before submission
Done!	Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)