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 4
May 9th, 2020

Revisions	
Milestone / Version	Date
Milestone 4 Version 1	
Milestone 3 Version 2	
Milestone 3 Version 1	4/23/2020
Milestone 2 Version 2	4/01/2020
Milestone 2 Version 1	3/31/2020

Table of Contents

Product Summary

Name of product: The Garage

Committed Functions

1. User	
1.1	Users shall be able to create an account.
1.2	Users shall be able to view additional information about the listing.
1.3	Users shall be able to book a parking spot.
1.4	Users shall be able to modify the search of parking spots.
1.5	Users shall be able to edit their account information
2. Guest	
2.1	Guests shall receive a confirmation email when they have booked a Parking Spot.
2.2	Guests shall be able to view the dates and times of previous bookings.
2.3	Guests shall be able to save spots that are marked as favorite.
2.4	Guests shall be able to view a page that lists all favorite parking spots.
3. Host	
3.1	Hosts shall be able to remove their own listings.
3.2	Hosts shall be able to upload pictures of their parking spot.
3.3	Hosts shall be able to change the availability status of their listing
3.4	Hosts shall be able to change the pricing of their parking spot.

What makes The Garage unique:

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.

Link to product:

http://ec2-18-144-3-40.us-west-1.compute.amazonaws.com/

Usability Test Plan

1. Users shall be able to search a location to view available spots nearby.

   a. **Test Objective:**

   The purpose of this test is to be able to determine the usability of the search function in The Garage. The ability to search for available parking spots is a fundamental part of our product. We want to make sure our users have a fluid experience when looking for a parking spot. Our goal is to take away from the stress many people feel when trying to find somewhere to park. An effective search function would help to ensure that users make The Garage their go-to resource for finding parking.

   b. **Test Description:**

   i. System Setup
   The user will need a browser that supports HTML5 and Javascript.

   ii. Starting Point
   Users will start this task at the homepage without an account.

   iii. Intended users
   This feature is meant for any user looking to rent a parking spot.

   iv. URL of the system
   http://ec2-18-144-3-40.us-west-1.compute.amazonaws.com/browseSpots.html

   v. What is measured
   We will measure the effectiveness of this task by checking the amount of time it takes to perform the search. We will also have the user fill out a questionnaire so we can evaluate their overall satisfaction with the task.

   c. **Usability Task Description:**

   The users will be tasked with implementing the search function on the Garage. They will have to navigate to the search area of the site, enter an address, and check for the results of available parking spots in that area.

Usability Test Table

Test/Use Case	% Completed	Errors	Comments
Search			

   d. **Questionnaire:**

I was able to find the search bar easily.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I was able to locate a specific listing.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The process of searching and getting results was done quickly.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

2. Hosts shall be able to post their own available parking spots.

   a. **Test Objective:**

      The purpose of this test is to evaluate the efficiency with which Hosts can post their parking spot to be rented out. Hosts play a vital role in our application. Without them, there would be no spots to book. By making the process of posting a spot easy and efficient, we can help ensure we make hosting through The Garage an enjoyable experience.

   b. **Test Description:**

      i. System Setup

         The user will need a browser that supports HTML5 and Javascript.

      ii. Starting Point

         The tester will have an account. The starting point for this task will be at the My Profile page.

      iii. Intended users

         This feature is intended for any user who wants to host a parking spot.

      iv. URL of the system

         http://ec2-18-144-3-40.us-west-1.compute.amazonaws.com/parking-spot/create/

      v. What is measured

         For this task we will measure the time taken for the Host to complete the process of uploading their post and the number of screens they have to navigate. We will have the user complete a questionnaire afterwards in order to gauge how they felt through each step.

   c. Usability Task Description:

      In order to host a spot, the user will have to create an account. For the user to become a Host, they will have to navigate to the page that contains the option to host a spot. They will be prompted to enter information about their parking spot, and then will finalize by confirming and uploading it.

Usability Test Table

Test/Use Case	% Completed	Errors	Comments
Post a spot			

d. Questionnaire:

I was able to find the Host a Spot page easily	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The required information needed from me to post a spot was conveyed clearly.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The process of posting a spot was simple.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I was satisfied with the process of posting a spot.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

3. Guests shall be able to reserve a parking spot.

   a. **Test Objective:**

      We are testing this function because we want to determine the efficiency with which guests can book a parking spot through The Garage. It is important that the user experiences a smooth process when booking a parking spot. Since this is one of the main functions our platform offers, we have to make sure users enjoy booking a spot. This will play a valuable part in bringing users back to our platform when they need to find somewhere to park.

   b. **Test Description:**

      i. System Setup

         The user will need a browser that supports HTML5 and Javascript.

ii. Starting Point

The guest will create an account before starting this task. The starting point for reserving a spot will be the Browse Spots page.

iii. Intended users

This feature is intended for a user who wants to book a parking spot.

iv. URL of the system

http://ec2-18-144-3-40.us-west-1.compute.amazonaws.com/parking-spot/001/

v. What is measured

We will evaluate the amount of time taken, number of clicks, and number of screens it takes to complete the process of booking a spot. We will also measure user satisfaction having the user fill out a questionnaire about their experience.

c. Usability Task Description:

The guest will be required to make an account in order to book a spot. They will have to select a spot, review its information, handle the payment process, and confirm their booking.

Usability Test Table

Test/Use Case	% Completed	Errors	Comments
Book a Spot			

d. Questionnaire:

I was able to find the Book option quickly.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The required information needed from me to book	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

a spot was conveyed clearly.					
The process of booking a spot was simple.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I was satisfied with the process of booking a spot.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

4. Hosts shall be able to change the availability status of their listing.
   a. **Test Objective:**
   This test focuses on the ease of use when it comes to changing the availability of a Host's parking spot. When a host goes out of town, changing whether Guests can use their parking spots should be quick and easy. This ensures that the host earns a little extra money on the side, while helping those who are stuck searching for parking.

   b. **Test Description:**
      i. System Setup
      The user will need a browser that supports HTML5 and Javascript.
      ii. Starting Point
      The user should create an account with us and post a parking spot before performing this test.
      iii. Intended Users
      The intended user is anyone hosting a parking spot with The Garage
      iv. URL of the system
      http://ec2-18-144-3-40.us-west-1.compute.amazonaws.com/moreSpecific.html
      v. What is measured
      Ease of use

   c. **Usability Task Description:**
   The User will be required to sign up for an account and post a parking spot. After this, they will need to navigate to their parking spots and

Usability Test Table

Test/Use Case	% Completed	Errors	Comments
View more info			

d. **Questionnaire:**

I was able to find the availability status quickly.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The process of changing availability was simple.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I was satisfied with the process of changing availability.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

5. Guests shall be able to view a page that lists all favorite spots.

   a. **Test Objective:**

   The objective of this task is to test how efficiently the user can navigate to the page containing a list of their favorite spots. Our top priority is to try and find ways to make the booking process easier for our users. By providing each user with their own personalized Favorites page, they can easily find their top picks for parking in the city and book them again.

   b. **Test Description:**

   i. System Setup

   The user will need a browser that supports HTML5 and Javascript.

   ii. Starting Point

   The user will have an account prior to starting this test. They will start at the Homepage.

   iii. Intended Users

   The intended users of this feature are Guests.

   iv. URL of the system

   1. http://ec2-18-144-3-40.us-west-1.compute.amazonaws.com/favorites/001

   v. What is measured

   In this test we will measure efficiency and overall user satisfaction.

   c. **Usability Task Description:**

The user’s task is to view the My Favorites page in the Garage. The user will start at the Homepage and navigate to the My Favorites page, where they can view favorited parking spots.

Usability Test Table

Test/Use Case	% Completed	Errors	Comments
Save a spot			

d. **Questionnaire:**

I was able to find the Favorites page quickly.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The process of viewing the Favorites page was simple.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I was satisfied with the process of viewing the Favorites page.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

QA Test Plan

Test Objective	**HW and SW Setup**	**Feature to be tested**
Verify that Guests who overstay their reservation will be incur additional charges	Portable or Desktop Computer running a modern version of Chrome	The system shall charge the user if they overstay.

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results
1	Overstay charge is applied to reservation	The reservation will reflect that overstay charges are applied when Guests occupy a parking spot for an extended period	Parking Spot Reservation for a fixed window of time	The reservation will show overstay charges for the extended period that the Guest occupied the Parking spot	FAIL
2	Overstay charges are applied to account	Overstay charges are reflect in the Guest's account	Reservation with overstay charges	The Guest account reflects the overstay charges for their reservation	FAIL
3	Overstay charge is applied to payment due	Overstay charges are reflected in the payments due for the Guest	Reservation with overstay charges	Guest account payment due includes overstay charges	FAIL

Test Objective	**HW and SW Setup**	**Feature to be tested**
Users who do not meet the password strength criteria are not permitted to create an account	Portable or Desktop Computer running a modern version of Chrome	The system will not process passwords without a set password strength.

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results
1	Short passwords will not be accepted	Short passwords will not be accepted	pass	The user is displayed an error message	FAIL
2	Simple passwords will not be accepted	Password that do not include a mix of alpha numeric values will not be accepted	password	The user is displayed an error message	FAIL

3	Common passwords will not be accepted	Passwords that contain english words will not be accepted	Secret123	The user is displayed an error message	FAIL

Test Objective	**HW and SW Setup**	**Feature to be tested**
Verify that an account cannot be created until all requirements are met	Portable or Desktop Computer running a modern version of Chrome	The system will not allow the user to proceed with account creation until requirements are met

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results
1	First Name is required	Verify that the First Name Field is required to create an account	null	An error message is displayed	FAIL
2	Last Name is required	Verify that the Last Name Field is required to create an account	null	An error message is displayed	FAIL
3	Password is required	Verify that the Password Field is required to create an account	null	An error message is displayed	FAIL

Test Objective	**HW and SW Setup**	**Feature to be tested**
Verify that the user is prompted to which requirements have not been submitted	Portable or Desktop Computer running a modern version of Chrome	The system will prompt the user as to which requirements are not met during form filling

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results
1	Error message is displayed when First Name is not populated	Verify that an error message is displayed which indicates that the First Name must be populated	null	An error message indicating that the First Name field is required is displayed	FAIL
2	Error message is displayed when Last Name is not populated	Verify that an error message is displayed which indicates that the last Name must be populated	null	An error message indicating that the Last Name field is required is displayed	FAIL
3	Error message is displayed when Password is not populated	Verify that an error message is displayed which indicates that the Password must be populated	null	An error message indicating that the Password field is required is displayed	FAIL

Test Objective	**HW and SW Setup**	**Feature to be tested**
Verify that the system will not record the location of the user who has not opted in to sharing their location	Portable or Desktop Computer running a modern version of Chrome	The system will not record the user's location if permission is not granted

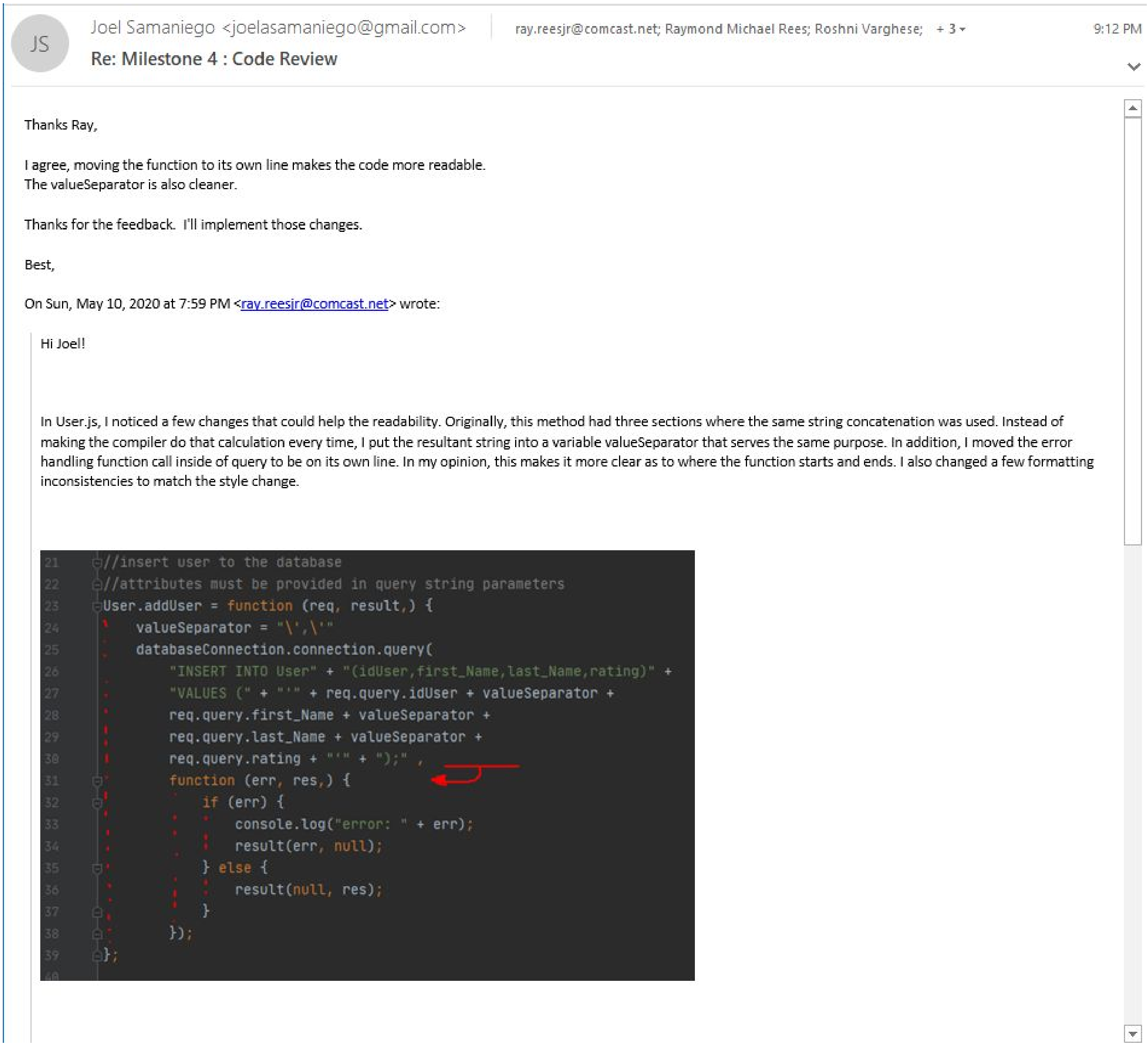
Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results
1	Search results will not display User Location unless opted in	Search option by location will not be available	null	Search by location is unavailable	FAIL

		to users who have not opted in			
2	Database will not record user location	Verify that the database does not record the user location	null	Database has no record of user location	FAIL
3	Account page will not display location unless opted in	The account page will not show the location for users who have not opted in	null	Account page location value is empty	FAIL

Code Review

```
const databaseConnection = require('../databaseConnection');

//User Object Constructor
let User = function () {
};

// mySQL request to retrieve user with matching ID
// userID is available in req.params property form the url
User.getUserWithId = function (req, result,) {
   databaseConnection.connection.query('SELECT * FROM User ' +
       'WHERE idUser = ' + req.params.userId, function (err, res) {
       if (err) {
           console.log("error: " + err);
           result(err, null);
       } else {
           result(null, res);
       }
   });
};

// insert user to the database
// attributes must be provided in query string parameters
User.addUser = function (req, result,) {
   databaseConnection.connection.query(
       "INSERT INTO User" +
       "(idUser,first_Name,last_Name,rating)" +
       "VALUES (" + "'" + req.query.idUser + "'" + "," + "'" +
       req.query.first_Name + "'" + "," + "'" +
       req.query.last_Name + "'" + "," + "'" +
       req.query.rating + "'" + ");", function (err, res) {
           if (err) {
               console.log("error: " + err);
               result(err, null);
           } else {
               result(null, res);
           }
       });
};

// mySQL request to delete user with matching ID
// userId is available in req.params property form the url
User.deleteUserById = function (req, result) {
   databaseConnection.connection.query('DELETE FROM User ' +
       'WHERE idUser = ' + req.params.userId + ";", function (err, res,) {
       if (err) {
           console.log("error: " + err);
           result(err, null);
       } else {
           result(null, res);
       }
   });
};

// exporting the User object for use in the App
module.exports = User;
```

Code review email:



JS Joel Samaniego <joelasamaniego@gmail.com> ray.reesjr@comcast.net; Raymond Michael Rees; Roshni Varghese; + 3 9:12 PM

Re: Milestone 4 : Code Review

Thanks Ray,

I agree, moving the function to its own line makes the code more readable.
The valueSeparator is also cleaner.

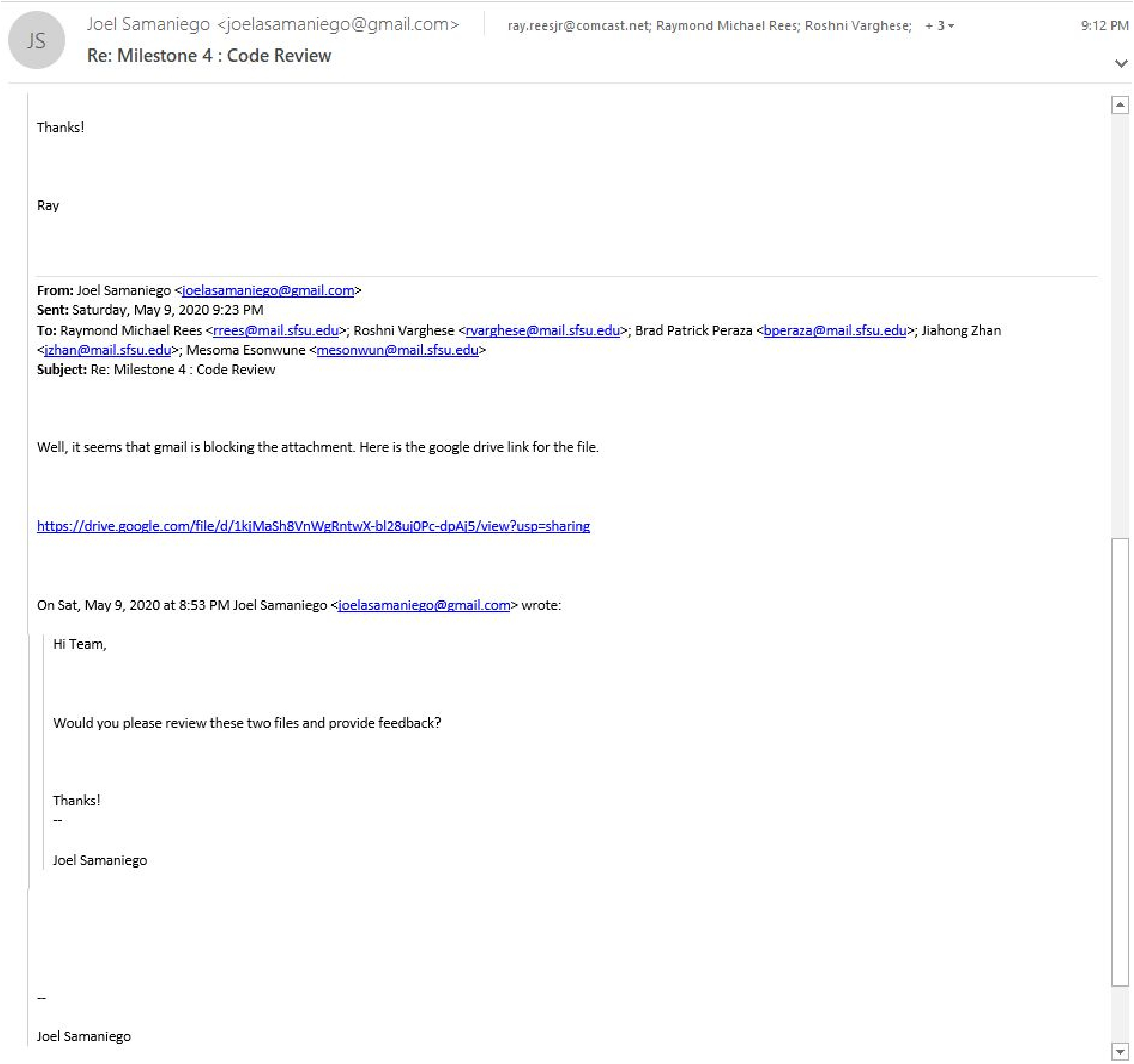
Thanks for the feedback. I'll implement those changes.

Best,

On Sun, May 10, 2020 at 7:59 PM <ray.reesjr@comcast.net> wrote:

> Hi Joel!
>
> In User.js, I noticed a few changes that could help the readability. Originally, this method had three sections where the same string concatenation was used. Instead of making the compiler do that calculation every time, I put the resultant string into a variable valueSeparator that serves the same purpose. In addition, I moved the error handling function call inside of query to be on its own line. In my opinion, this makes it more clear as to where the function starts and ends. I also changed a few formatting inconsistencies to match the style change.

```
//insert user to the database
//attributes must be provided in query string parameters
User.addUser = function (req, result,) {
    valueSeparator = "\',\'"
    databaseConnection.connection.query(
        "INSERT INTO User" + "(idUser,first_Name,last_Name,rating)" +
        "VALUES (" + "'" + req.query.idUser + valueSeparator +
        req.query.first_Name + valueSeparator +
        req.query.last_Name + valueSeparator +
        req.query.rating + "'" + ");" ,
        function (err, res,) {
            if (err) {
                console.log("error: " + err);
                result(err, null);
            } else {
                result(null, res);
            }
        });
};
```



Joel Samaniego <joelasamaniego@gmail.com>

ray.reesjr@comcast.net; Raymond Michael Rees; Roshni Varghese; + 3

9:12 PM

Re: Milestone 4 : Code Review

Thanks!

Ray

From: Joel Samaniego <joelasamaniego@gmail.com>
Sent: Saturday, May 9, 2020 9:23 PM
To: Raymond Michael Rees <rrees@mail.sfsu.edu>; Roshni Varghese <rvarghese@mail.sfsu.edu>; Brad Patrick Peraza <bperaza@mail.sfsu.edu>; Jiahong Zhan <jzhan@mail.sfsu.edu>; Mesoma Esonwune <mesonwun@mail.sfsu.edu>
Subject: Re: Milestone 4 : Code Review

Well, it seems that gmail is blocking the attachment. Here is the google drive link for the file.

https://drive.google.com/file/d/1kjMaSh8VnWgRntwX-bl28uj0Pc-dpAj5/view?usp=sharing

On Sat, May 9, 2020 at 8:53 PM Joel Samaniego <joelasamaniego@gmail.com> wrote:

> Hi Team,
>
> Would you please review these two files and provide feedback?
>
> Thanks!
> --
>
> Joel Samaniego

--

Joel Samaniego

Self-check on best practices for security

Self-check: Adherence to original Non-functional specs

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.
- Google Ads shall be used to target the Garage 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