Github Link: https://github.com/GioEpic123/GarageManager

1. Project Title and Authors
    a. **Team:** GarageMan
    b. Krimika Keemtee, Aaron Villasenor, Giovanni Quevedo, Aster Lee, Andrew De La Rosa

2. Preface

For our document, we expect this to be read by both our developers and our client(s) that are overseeing this project. Our version history will help show the current state of both our documents and our actual project, the use of decimal places will be used to show whenever we make any changes to an artifact, whether it be a major or minor change, major changes are shown by a change in the first number, and minor changes/revisions will be shown by a change in the second number.

Software version 1.0, Document Version 2.0

3. Introduction

Garage Man is a system designed to make garage maintenance painless. Users can reserve a spot at a Garage Man location using our website, or simply arrive at a site. The automated system will allow users to check their vehicle into the garage, and issue a virtual ticket.

The automated system also allows users to make future reservations, and pay for their parking time up front. Reservations are $10 an hour, with a $5 flat fee for creating the reservation. Users can cancel a reservation if they no longer need it and be refunded, however the $5 reservation fee will not be refunded.

When a user is finished enjoying their parking, they simply check out and Garage Man will bill their account pursuant to the parking rate, which is $10 per hour.

4. Architectural Change

We did not make a change in the architecture of the project.

5. Detailed Design Changes

1. **Change:** We had originally planned to allow the user to check out and cancel the tickets and reservations from the Homepage, but during the implementation we decided to move this feature to the Tickets Page.
    1.1. **Rationale:** We did this in order to make the Homepage feel less cluttered, and since our Tickets

page had tables for the active/inactive tickets and reservations, it made it easier for our user's to see exactly which ticket or reservation they would be checking out from or canceling.

2. **Change:** We were going to have the user insert their payment information alongside their car information when creating their account, as well as allow them to update this information if necessary in an account info page, but we ultimately decided not to implement this.
    2.1. **Rationale:** Due to the nature of this project, we did not think it was necessary to ask the user for their actual payment information, leading to us not asking for this information at account creations and leading to us not making the account information page.

3. **Change:** We had planned on displaying a countdown timer to show how long until a user's reservation had started but ultimately did not implement this.
    3.1. **Rationale:** Due to time constraints and wanting to get more important features finalized, we decided to not implement this, but if given more time, this countdown timer would have most likely been shown on the Homepage with the corresponding reservation.

4. **Change:** We wanted to have logic to activate a reservation when it's time rolled around, and deactivate it when it's duration ended to allow easier checking if a car should be parked at a given time. Instead, the values always stay the same as when they are written, or canceled/closed.
    4.1. **Rationale:** Due to the time constraint we could not implement this logic, as it required either a front end to change the tickets according to the time, or some back-end-server whose purpose would be to constantly update ticket data considering time.

6. Requirement Change

If you have changed any of the requirements since your assignment 1 and 2, explain the following:

a. <u>Payment Method</u>
    i. Does this change your design?

- It did not change the overall design of the project.

ii. If not, why not?
- Since we planned on having an external payment service, or some partitioned one, the communication would have been the same, thus our system would have functioned the same way.

iii. Has this new requirement been implemented? Or will be implemented in future?
- The requirement was not implemented due to the time constraint and difficulty. This will not be implemented in the future.

b. <u>Countdown Timer</u>

i. Does this change your design?
- Yes, it changed our design.

ii. If so, what should be changed and how?
- The main function for the timer was to let the user see a live countdown of how much time they had left for parking. Having this feature would have enabled us to make the reserved ticket active automatically. This also led to not having the buttons correctly formatted to cancel the active ticket within the active table rather than the inactive table.

iii. Has this new requirement been implemented? Or will be implemented in the future?
- No this requirement has not been implemented and it will be implemented within the future.

c. <u>User changes vehicle information instead of adding multiple vehicles</u>

i. Does this change your design?
- No, this did not change our design

ii. If not, why not?
- We are still storing a user's vehicle and the main point of having that information is still to validate what vehicle the user will be parking in when they go into the garage. Having the user enter and store multiple vehicles would make dealing with each user's vehicle troublesome, especially if we have a large number of users making reservations everyday.

iii. Has this new requirement been implemented? Or will be implemented in future?
- Yes, this requirement has been implemented.

d. <u>Reservation spot vacancy</u>

i. Does this change your design?
- Yes, not having this requirement has changed our

design.

ii. If so, what should be changed and how?
    - We should be able to let the user know if there are any open reservation spots in the parking garage. We could have done this by creating 10 spots in the database that each represented a single parking space and had a value of true, to represent open, and false, to represent closed. Then, the front end displays the open spots the users can choose from and closes the spot when they do.

iii. Has this new requirement been implemented? Or will be implemented in future?
    - This requirement was not implemented, but we plan on implementing it so that the web app can work more to a real parking website.