SMART PARK

"RU Ready to Park Smart?"

Demo 1: User Documentation

Github: https://github.com/swetha-5689/SmartPark

Group 3

Disha Bailoor, Neha Nelson, Param Patel, Swetha Angara,
Nicholas Meegan, Thomas Murphy, Charles Owen, Jeffrey Samson,
Aniqa Rahim, Brian Ogbebor

Table of Contents

Customer Group User Documentation **3**

- Signup 3
- Login 3
- Reservations 4
- Billing 5
- Google Assistant 5

Manager Group User Documentation **6**

- Login 6
- Home Page 6
- Statistics Page 7
- Calendar Page 7
- Garage View 8
  - Reservations 8
  - Edit Layout 9
  - Overview 9
- Pricing Admin Panel 9
  - View/Edit Live Price Model 9
  - Compare Historic Model with Results 11

Elevator Group User Documentation **13**

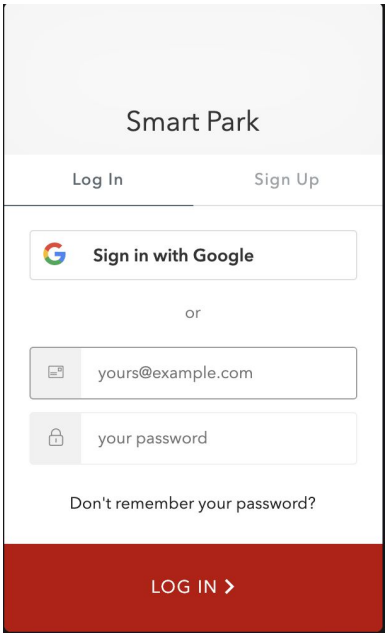
- Elevator Front End 13
- Simulation 14
- Spot UI 16

Customer Group User Documentation

Welcome to SmartPark! Here the user can login or create an account for the SmartPark website. To Run the front end proceed to this link: https://boiling-river-38785.herokuapp.com/

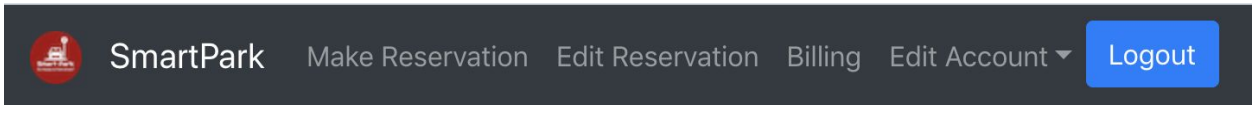
Signup

Put in an email and password to be able to login.



Login

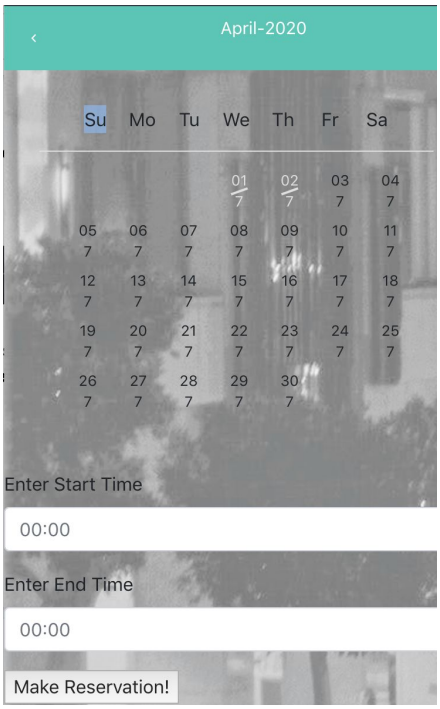
Once they are Logged in, the user can now access their account where they can make/edit reservations, edit their account in which they can add new vehicles, credit card information and change account password and pay off their bill.



Reservations

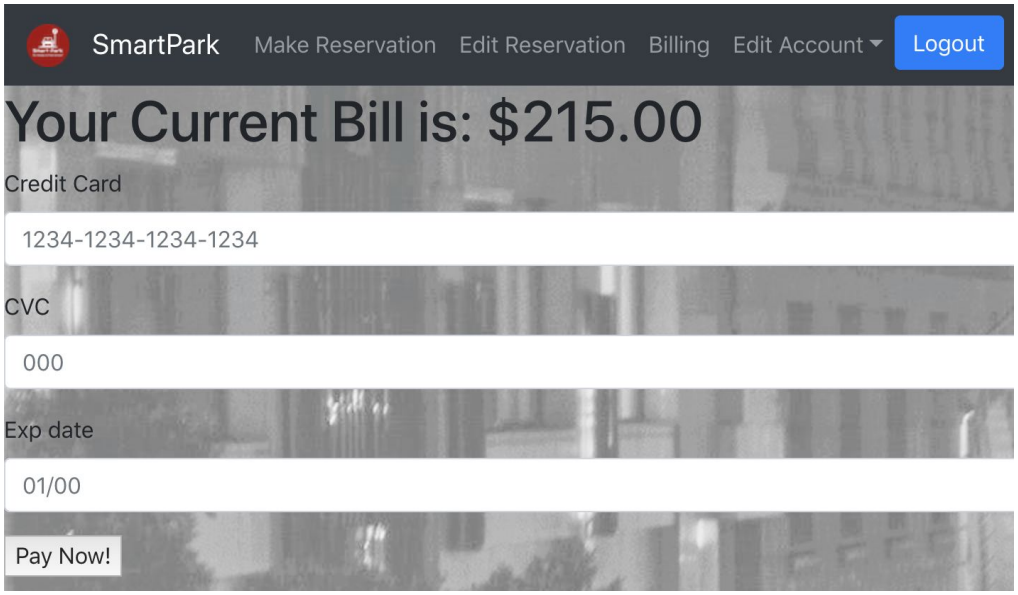
The User can click on the ‘Make a Reservation’ button. This page is not completed yet, however, the user can scroll through months and select a date. After the pages are integrated with the back end, the user should be able to choose a date, enter the start and end time to make a reservation that will all be saved

In order to Edit a reservation, currently this page is blank, but once completed User should be able to access the reservations they have created and be able to change the date and time.



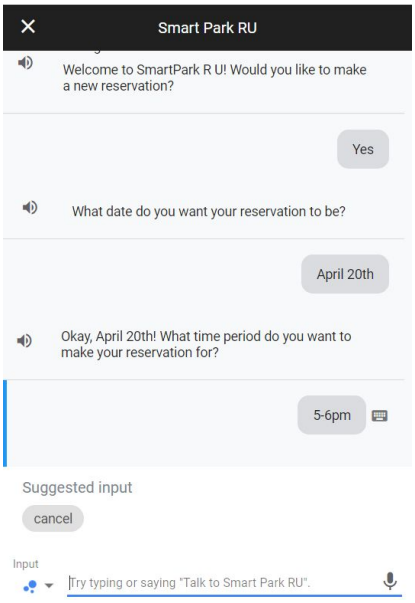
Billing

For the Billing page the user should be able to see the total amount owed. They can enter their credit card number CVC, and expiration date and click Pay Now. Once the amount owed is successfully paid the user should receive an email confirmation.



Google Assistant

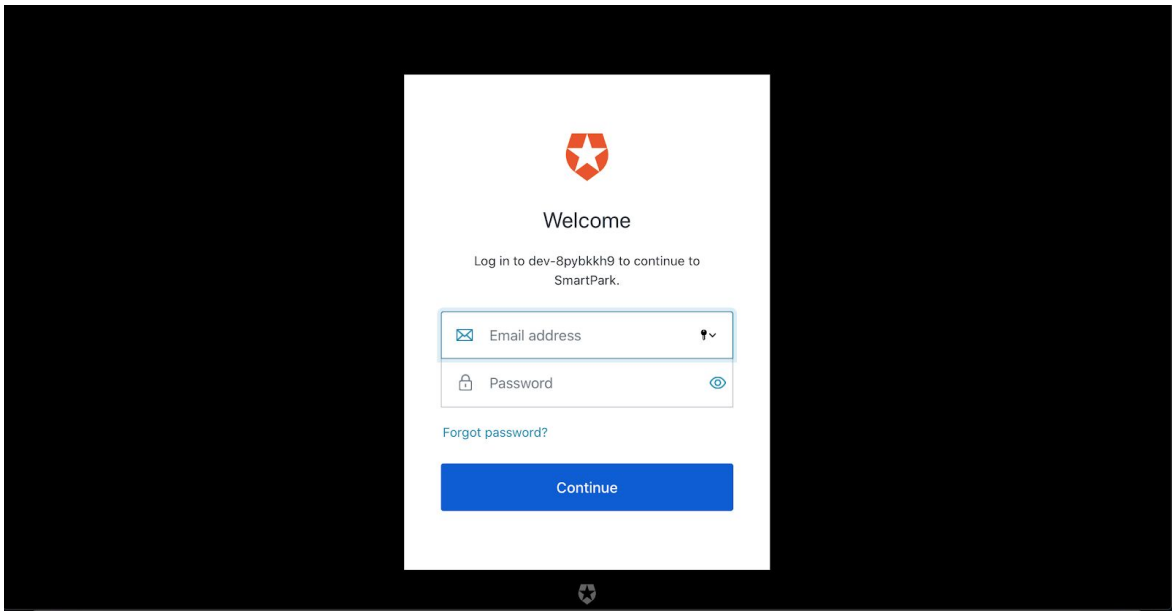
The User can create reservations hands-free using any Google Assistant smart device. The User tells SmartPark the date of their reservation and the duration of their reservation. SmartPark confirms your reservation by repeating it back to you.



Manager Group User Documentation

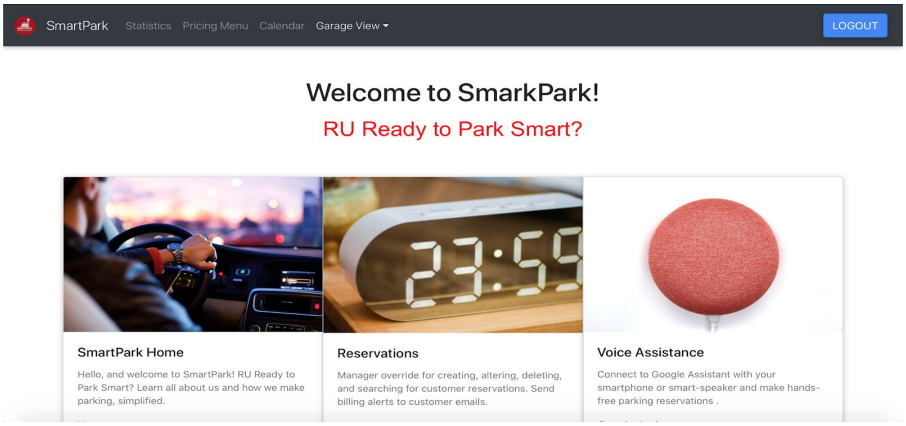
Login

First a manager is required to enter their login credentials.
Example:
Email: jeffrey@example.com
Password: SmartPark2020



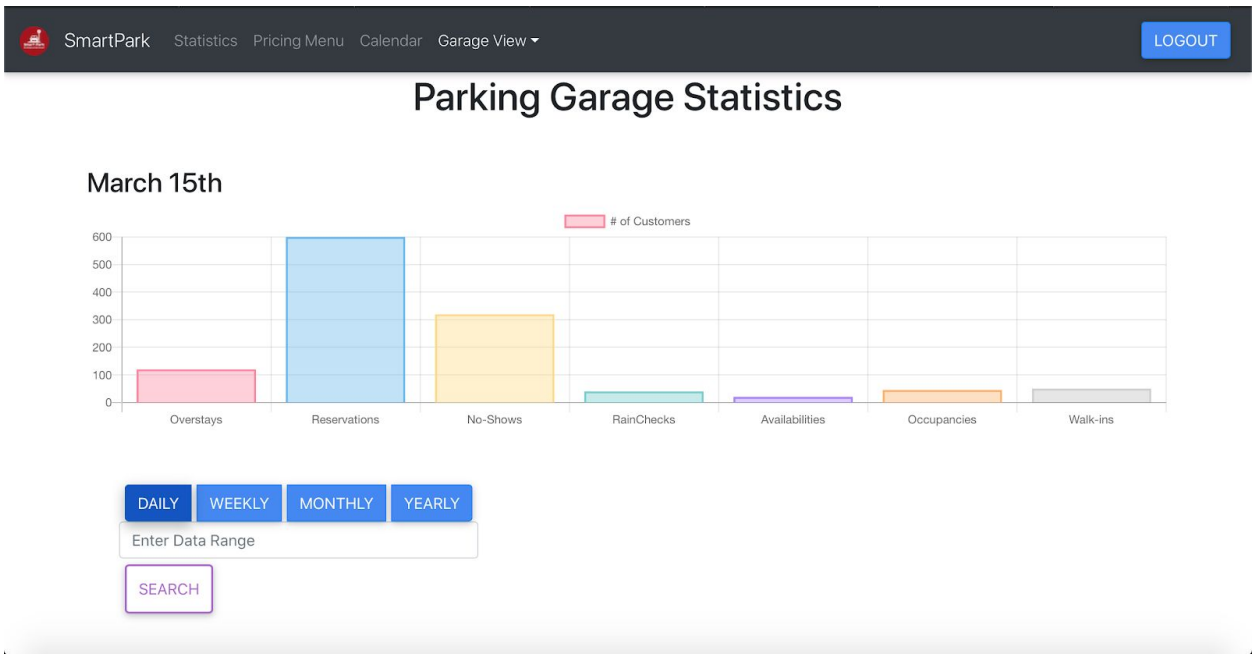
Home Page

Once a manager logs in they will be directed to the home page where they can go to other pages with other features.



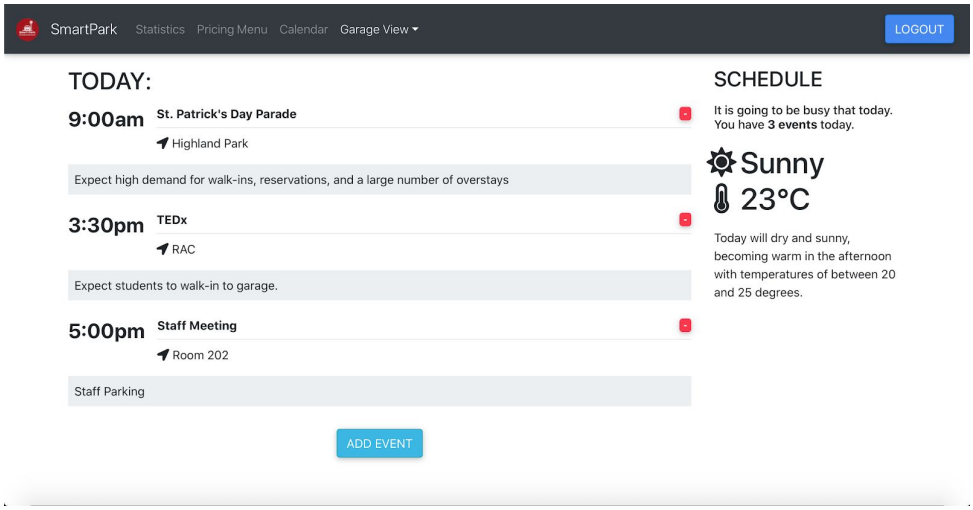
Statistics Page

This is our statistics page and here you can view business statistics on a daily, weekly, monthly, and yearly time frame. You can also search for a specific data range to get a more accurate and relevant business model.



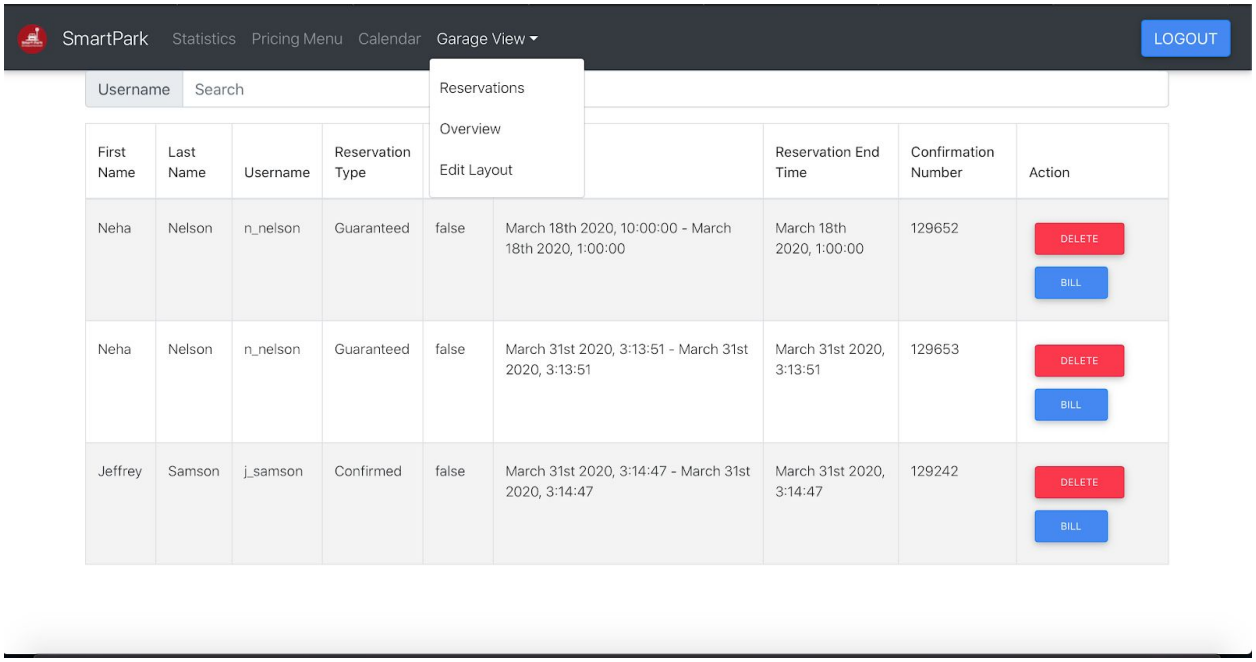
Calendar Page

Our calendar page can store events that would require parking and probably cause an influx of business. Also, you can have events related to your business such as staff meetings.



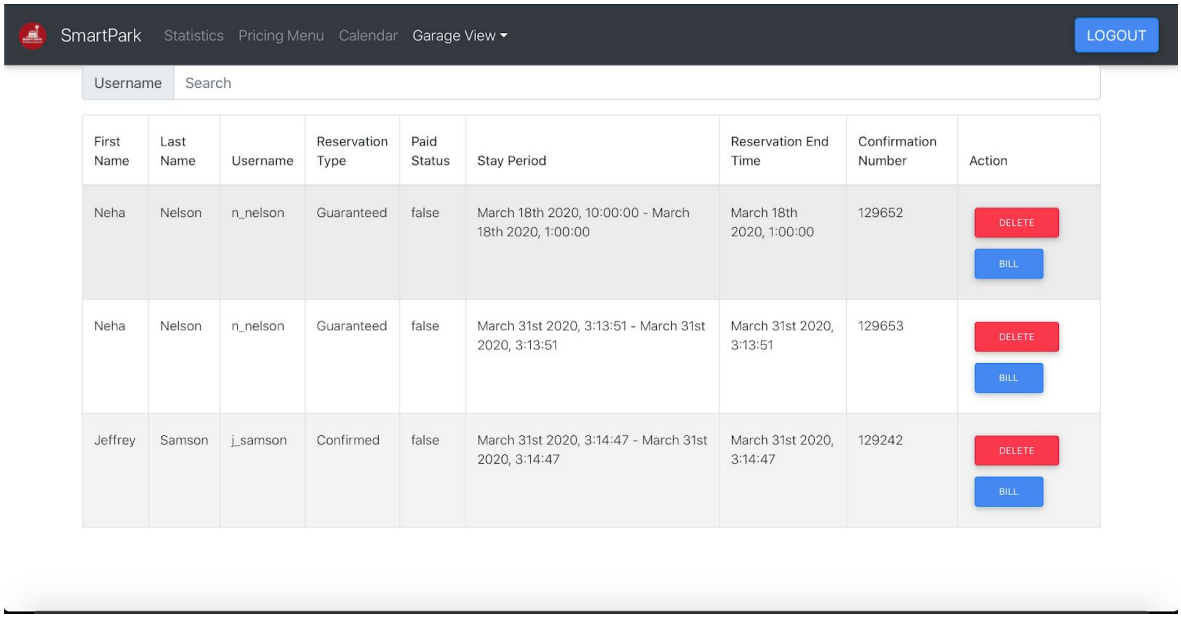
Garage View

The garage view is a drop down menu to see the garage in three different aspects: with regards to reservations, a general overview, and to edit the layout of your garage.



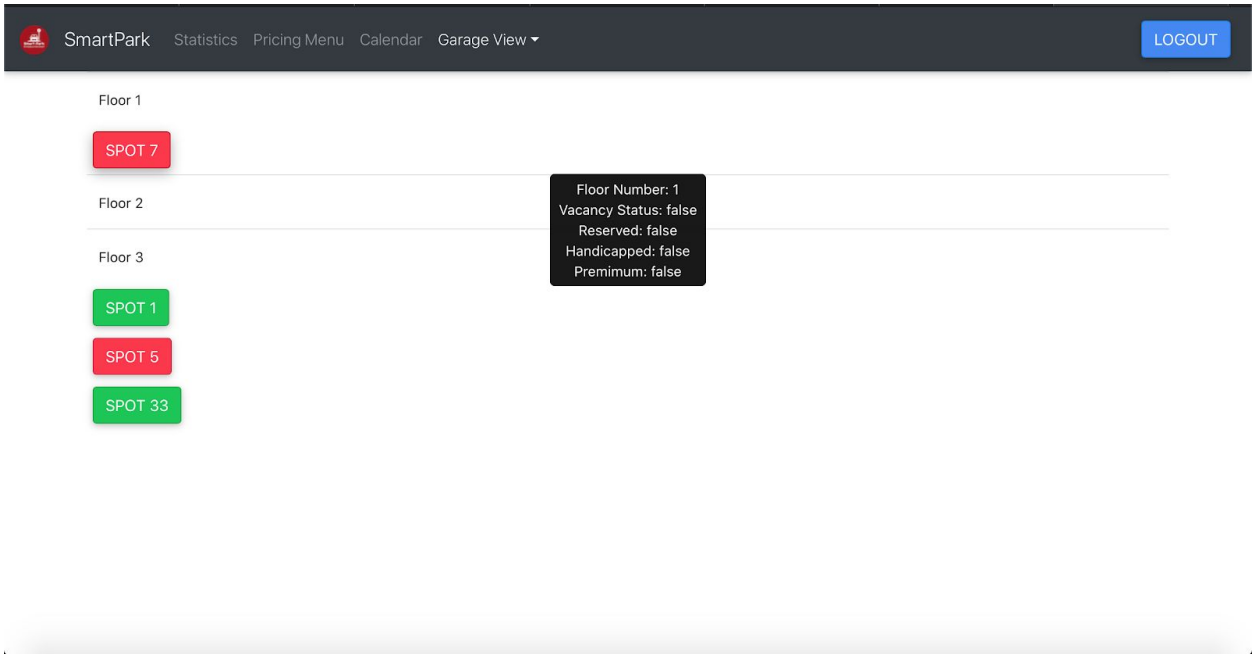
Reservations

The reservations page allows the manager to see the current scheduled reservations and customer details like name, username, type of reservations etc. Here we can physically bill or delete reservations.



Edit Layout

This page is not yet completed but if you hover over the spot number you can see the popup in the middle of the screen with all the spot information. After more development the users will be able to list their garage configurations and see their garage in real time.



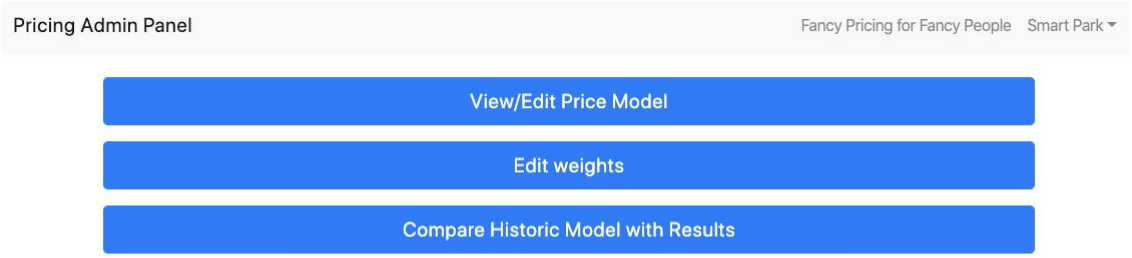
Overview

Under development.

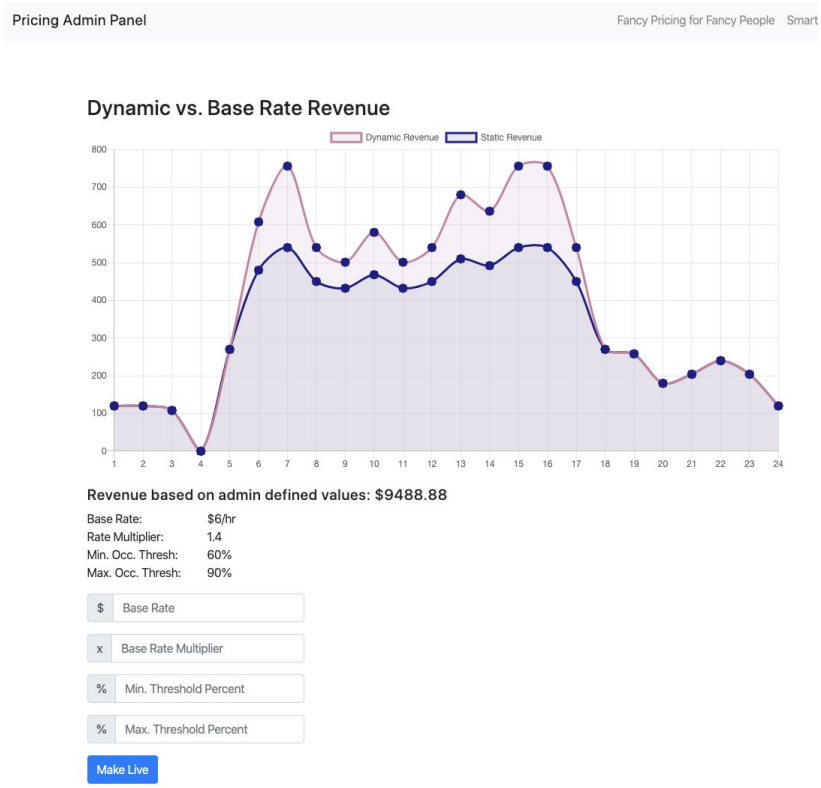
Pricing Admin Panel

View/Edit Live Price Model

Welcome to the Pricing Administration Panel. Here, you can update the Live Price Model for your garage. The price model is intended to be an easy to understand, easy to use function to incorporate dynamic pricing principles into your parking garage rates and fees.



Under the **View/Edit Live Price Model** option, you'll find access to the user defined fields you'll use to define pricing and a simple graph that displays projected revenue.



The premise of the model is that you, the administrator of the system, may enter your standard hourly parking base rate first. This is the value from which all subsequent inputs work off of. So, if the normal hourly rate for parking in your garage is $6/hr. That's the rate we'll start off entering in that field. You can always come back and change this value after you've experimented.

The base rate multiplier will determine the maximum rate that you would ever like to charge as an hourly rate. For instance; if your base rate is $6/hr and you would like to charge a maximum of $8.40/hr when the garage is starting to reach capacity, you would enter a multiplying factor of 1.4.

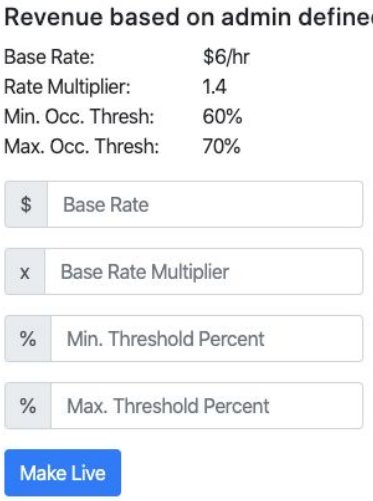
Ex: Base Rate * Multiplying Factor = Maximum Rate Charged
Ex: $6/hr. * 1.4 = $8.4/hr

Next, you can set the threshold values. The minimum threshold is the percentage of occupancy you want to wait for before the dynamic pricing "kicks in". For instance, if you don't want to charge a higher rate until the garage is at least 60% occupied. You would enter the value: 60, into this field.

The maximum occupancy threshold is the percentage of occupancy you would like the maximum rate applied to. For instance, if you want the maximum rate applied to those parking when the occupancy is 70% and above you would enter the value: 70, into this field.

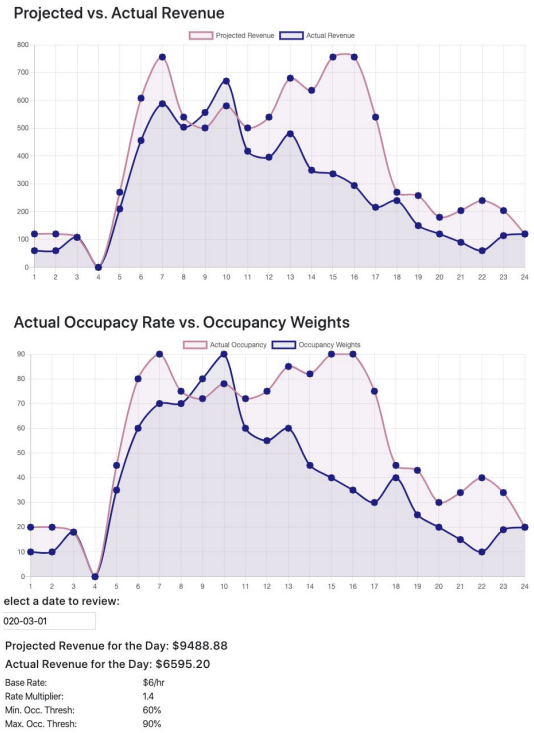
The rate will scale from base to maximum over the range of occupancy between the two values you input and stop scaling thereafter.

The described inputs are shown as an example below.



Compare Historic Model with Results

As the administrator you will want the ability to view the historic results of your price model. This feature is found by pressing the **Compare Historic Model with Results** button.



Under this portion of the Pricing Admin Panel you will find a simple “date picker”. Clicking on the date picker will bring up a calendar. On this calendar you can easily select the date of interest.

After selecting the date you’d like to see historic price model performance for the graphs and revenue fields will automatically populate. The first photo of this section is an example of what you will see.

The top graph presents you with an overlay of the projected revenue for the day in question along with the actual revenue the garage generated on that same date. This projection was created with the values you input for the price model. The graph charts revenue in dollars on the X axis and hour of the day on the Y axis.

The bottom graph presents you with an overlay of the actual occupancy percentages by hour and the assumed occupancy weights on the date in questions. The occupancy of the garage in percent is charted on the X axis, and the hour of the day is charted on the Y axis.

Elevator Group User Documentation

Elevator Front End

In this brief user documentation, a walkthrough of necessary components and how to use the SmartPark elevator system is gone into detail.

In order to begin using the front end of the elevator terminal, four devices need to be connected: the camera in the elevator, the back end software, the weight sensor hub (which connects all of the weight sensors together in one spot), and the license plate scanner in the elevator terminal. If these devices are not connected, a boot failure screen will display. To advance to the main elevator terminal, please ensure these devices are connected. (If you are using developer mode, press the respective buttons in order to advance the screen. See the “dev mode” or “demo mode” section of the documentation for more details.)

After the devices are properly connected, the main elevator screen is displayed. Once this screen is displayed, using the camera, the elevator terminal system begins searching for vehicles that are going to enter the elevator terminal. When a vehicle is found, the customer is prompted to enter the elevator. From there, the elevator terminal scans the license plate of the customer. If the license plate scan is successful, then the success is displayed and a reservation number is searched for through the database for that license plate number. If the reservation is found and the garage is not full, the customer’s spot is displayed on the screen and is taken to that floor. The elevator then returns to the ground level, ready to accept another customer.

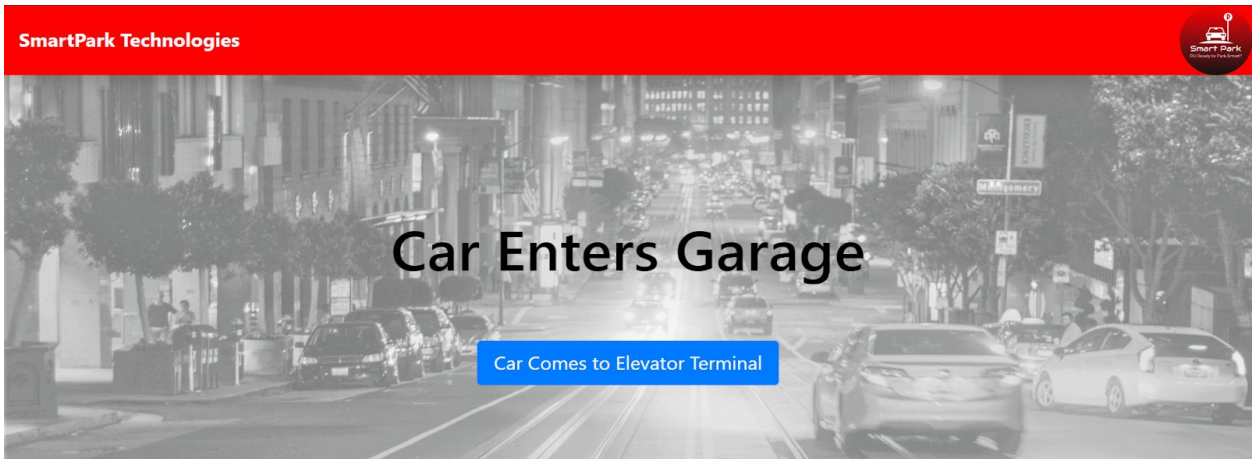
However, if the license plate could not be scanned, however, then the customer is asked whether they have a membership number or not. The membership number is a unique number given to the customer when the reservation was made. After being prompted, the customer has the choice to enter the membership number if they have one or exit the elevator to create an account on the walk-in terminal. A reservation is then searched for using the membership number.

These consist of a few of the branching paths that the elevator terminal is able to take. Each path can be explored in depth by using the elevator terminal’s built in demo mode. NOTE: In order to access the demo mode, a variable in the devMode.jsx file needs to be edited. This should only be done with authorized SmartPark staff, and most garage owners should not edit the code under any circumstances. The demo mode can be accessed through the SmartPark website for managers under the simulation tab.

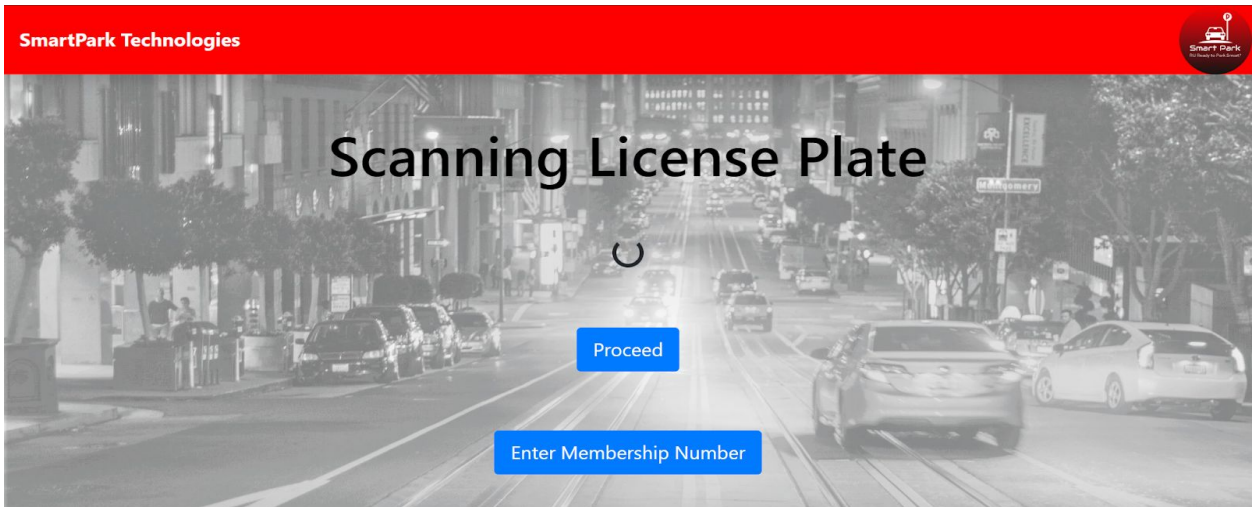
When demo mode is enabled, button inputs can be used in order to test the scenario that the garage owner wishes to see. Setting the buttons will simulate what would take place if a vehicle had entered the elevator under the given conditions. Moreover, plans to simulate multiple cars entering the elevator planned to be added in a future release.

Simulation

The simulation is mostly to outline the different scenarios that the elevator group has to account for when the customer approaches the elevator terminal.
So the home page looks like this:

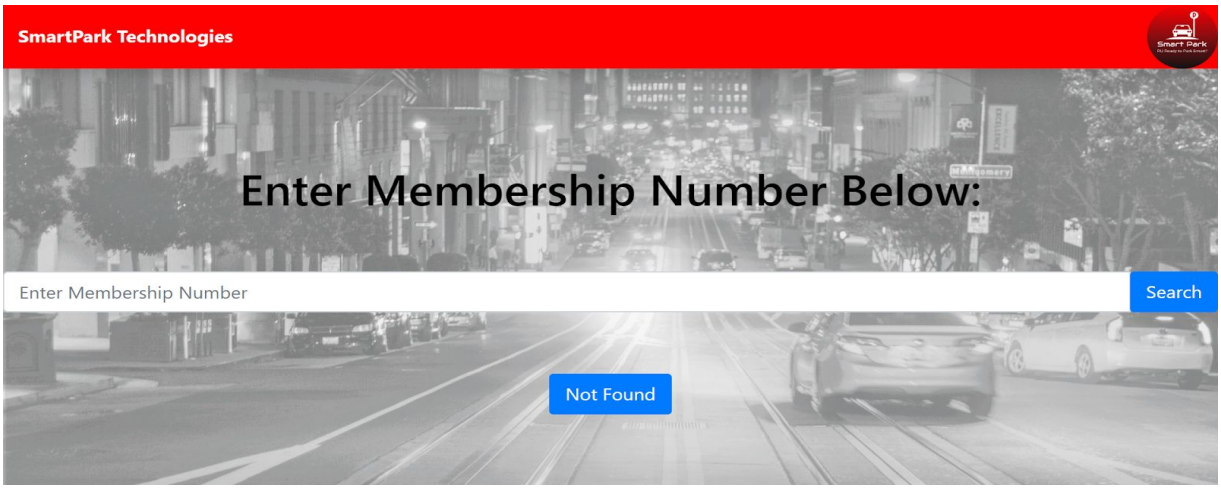


The interactive button on the home page simulates the scenario for when the user approaches the elevator terminal. Clicking that button leads us to the Scanned.js page:

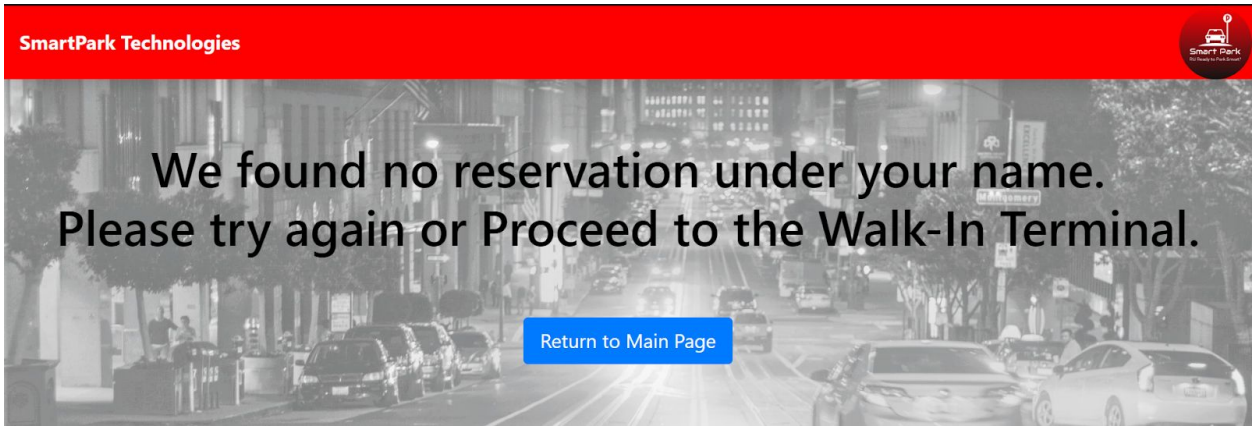


This page is simulated for the license plate scanner, when the Proceed is selected, it leads to the success page, when the spot is found. This is for when the license plate is

successfully scanned. If the license plate is not able to be scanned due to faulty license plate or a faulty scan, then it leads to this page:

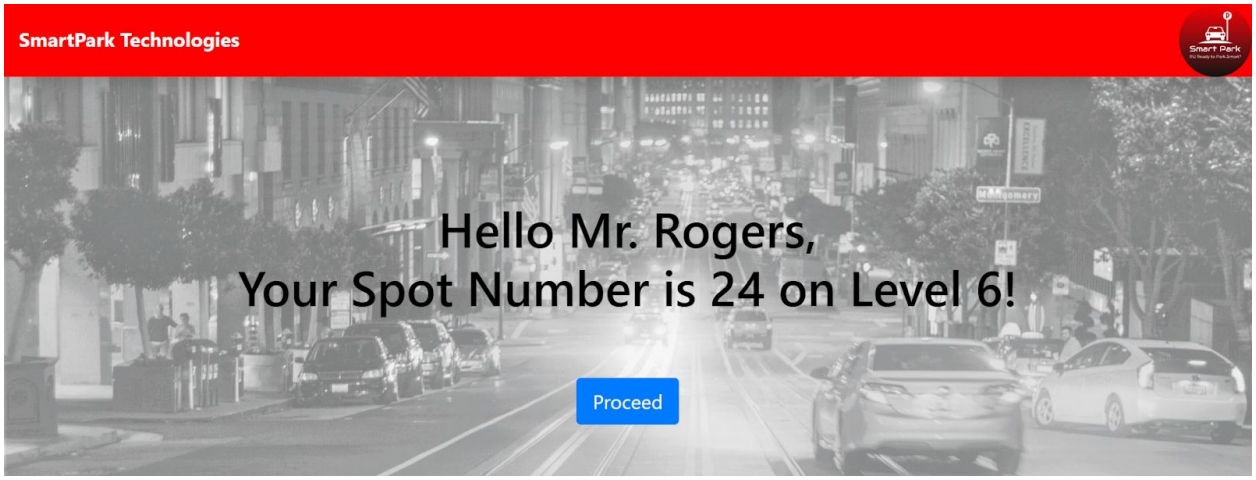


This is for the Not Scanned Scenario, when the user is asked to manually input the membership number in the Text Box and then when the Search button is pressed there is a backend search conducted on the database with the reservation information. Search leads to the success page. However, when the membership number is not found in the backend, and thereby not finding the reservation leads to this page:



This is the failure page which loops back to the main menu through the button, and informs the customer to leave the elevator so that they can be treated as a Walk-In customer.

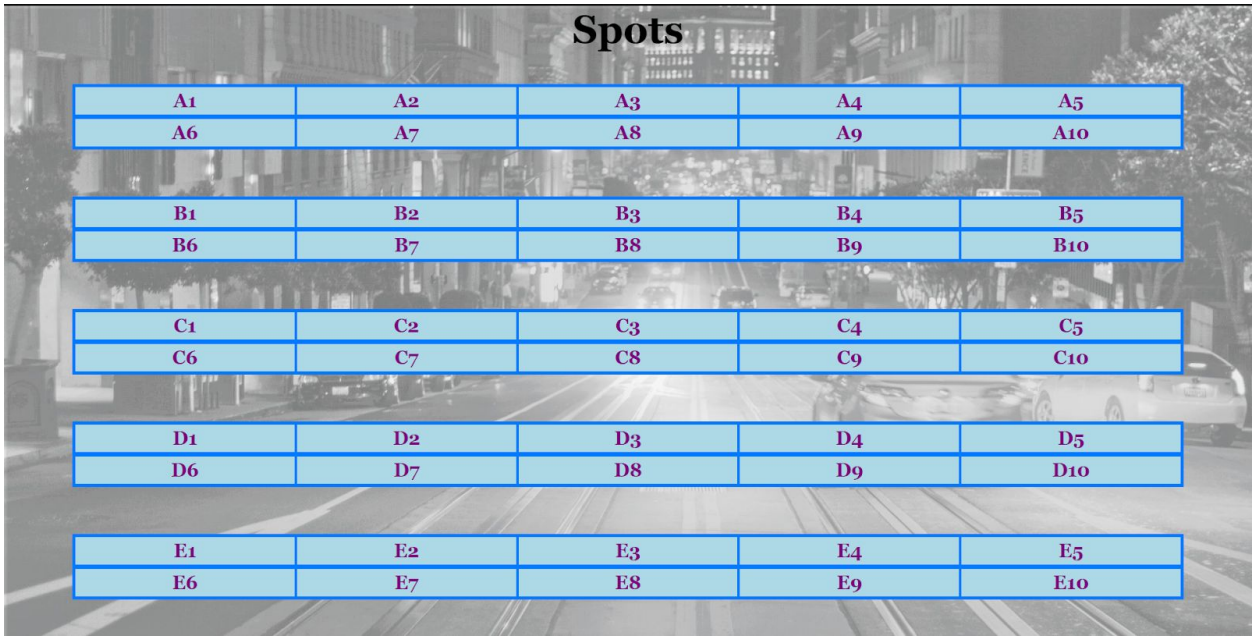
The success page looks something like this:



This page also loops back to the home page, through the proceed button.

Spot UI

This is the front end UI for the backend of the elevator group. It is the spots on the floor layout of the parking garage. We used a definite set of spots to model it in this UI; we will be able to change the number of floors and spots per floor in the next demo.



In this the boxes represent the spots on the floor. There is space between the rows of the spots to allow mobility for the users. The user and the manager can look at this layout to see what the floor plan garage looks like. This layout will also be used to figure out which spots are occupied and otherwise by the manager.