ZippyPark

An Optimized Automated Parking Garage

14:332:452 Software Engineering

Project Report 1

Team 2

Samantha Moy

Atmika Ponnusamy

Samantha Cheng

Kylie Chow

Andrew Ko

Parth Patel

Shreya Patel

Nandita Shenoy

Piotr Zakrevski

Submitted February 23, 2020

CONTRIBUTION BREAKDOWN

The following table reflects the approximate contribution of each team member to each subsection of this report through brainstorming, researching, writing, or editing that portion. *All team members contributed equally.*

	Samantha M.	Atmika	Samantha C.	Kylie	Andrew	Parth	Shreya	Nandita	Piotr
Project Management*	90%	10%							
Customer Problem Statement	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%
System Requirements	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%
Functional Requirements	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%
User Interface Specification	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%
Domain Analysis	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%
Project Size Estimation	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%
Plan of Work	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%

Notes:

** "Project Management" refers to the activities required to assemble a cohesive report. These affairs include managerial tasks such as: organizing meetings, planning timelines, managing communications, and editing full reports. Project management also includes the activities required to enable teammates to write knowledgeably about their portion. These affairs include researching, setting up, and creating tutorials about the project systems and technologies for the education of the other teammates.*

Contents

CUSTOMER PROBLEM STATEMENT

Problem Statement

Parking garages are a modern development created to make parking easier and more convenient for drivers who do not want to circle the streets for open parking. However, the current system of parking garages has not been optimized to mitigate the full frustrations of parking for customers.

Not many of today's parking garages operate with the aid of computerized systems to manage garage occupancy. Instead they utilize a system where employees walk around to inspect the occupancy of individual spots. Those that do monitor occupancy with a sensor system simply keep count of vehicles that enter and exit. Thus, there exists the issue of searching for a vacant spot once a customer has entered the garage. For many customers, locating an unoccupied parking spot is a troublesome and tiring process. According to a report published in 2017 by INRIX Research, an analytics company that provides parking and traffic data, Americans spend on average 17 hours per year searching for parking [1]. Taking into account fuel and emissions, the resulting cost of this wasted time is calculated to be $345 per driver [1]. The grueling process of circling floors upon floors of a parking garage causes frustration for customers. Furthermore, garages nationwide currently utilize a parking system where customers pay after parking based on the length of their stay. This payment occurs during the process of leaving the garage, which leads to the issue of slowdown at the exit of the garage. Overall, under current parking garage systems, the process of entering and exiting the garage is slow and tiresome for customers. This is counterproductive of the purpose of parking garages which is to provide faster and more convenient parking.

These issues can be alleviated through a more streamlined process which can be provided by an automated parking garage system. By reducing the hassle of locating empty parking spots and eliminating the need to pay at departure, parking at garages will be made quicker and easier, lessening the burden on customers and increasing customer satisfaction.

Proposed Solution

The process of parking at a garage can be optimized in the following ways: providing customers with a designated spot so they do not need to waste time searching for vacancy and allowing customers to make reservations beforehand to expedite the payment process. Reservations will also allow for a more precise system of monitoring garage occupancy. This system will increase efficiency by lowering traffic congestion and the time associated with customers searching for a parking spot.

This project provides customers with a simple and streamlined process through an enhanced automated parking system that can be implemented in any garage. By means of a simple user interface provided on a smartphone application and/or webpage, customers can reserve a spot in the parking garage for an allotted time and upon arrival be assigned a designated spot. The system requires customers to set up a profile which will enhance the payment process by allowing for a prepaid method of payment. In summary, customers can register a profile and create, edit, and delete reservations on the website. Upon entry, a scanner will identify the car and customers with a reservation will be directed to a spot. The system also allows for walk-in customers, who must still have a registered profile to speed up the process of payment. The system will keep track of the occupancy status of specific spots to direct customers to the best available spot. Such a system will create an easy and flexible process for parking in a garage.

Existing projects have aimed to solve the issues of modern parking garage systems, through features such as dynamic pricing [2] and QR code implementation [3]. However, to create a new automated garaging system, these ideas have been improved and built upon with the goal of assisting both the customer and parking garage management in novel ways. These features, (1) a penalty-reward system, (2) handicap accessible parking, (3) surge pricing, and (4) QR code usage, are described as follows.

(1) A new functional feature is a penalty-reward system that will encourage customers to obey the regulations of the parking garage and reward preferred behavior. Good behavior will include general use of the parking garage, showing up to reservations on time, and making reservations rather than walking in. Conversely, bad behavior will include staying past a reservation. Customers will not be reprimanded for arriving late to or missing a reservation since they have already paid for the reservation and may use the spot as they would like. Each act of

good or bad behavior results in 10 points being given/taken away from a user's score. Customers may spend their points on awards, such as free parking days, discounted prices, or spots closer to exits. If a customer reaches -10 points, they will face a penalty, such as a fine, and their score will then be reset to zero. Repeated offenses may result in more severe penalties such as a restriction of walk-in privileges or loss of the ability to use the garage. This system aims to establish a larger number of frequent, loyal customers who consistently exhibit ideal behavior.

(2) A qualitative feature of our garage is accommodations for handicapped patrons. Due to the size of the garage, customers with physical disabilities may struggle to reach nearby exits. Our aim is to provide an extra option to help handicapped individuals without sacrificing user-interface simplicity. During registration, new customers will be able to request handicap arrangements so that their reserved spot will be closer to the entrance/exit. Their profiles will reflect this information, and future reservations will automatically reserve handicap parking spots. This feature guarantees the customer will always have accessible parking without the need for further action. In order to prevent abuse of this feature, customers will need to provide verification. They will need to enter their unique ID found on their handicap placard. This information will then be linked to the user's profile so that multiple profiles cannot share the same code. Ideally, the addition of handicap parking spots will improve customer loyalty without ruining the user experience. The handicap component is also a feature of the 2019 group's project, but we will be improving upon it by adding the verification system [4].

(3) An additional functional feature is a dynamic pricing model which determines parking-rates based on the time of day and the demand for walk-in reservations. A dynamic pricing system would allow owners of the parking garage to maximize their profits by charging a premium on the regular parking-rate when conditions permit. The factors this system specifically uses to determine price multipliers are the time of day—to raise prices during rush hour and other peak driving hours—and the volume of customers requesting parking spots. This pricing system would affect the rates for confirmed reservations, guaranteed reservations, and walk-in reservations uniquely. This component is a feature of the 2018 group that we will be improving on by introducing a new algorithm that will maximize profit [2].

To determine the dynamic parking rate, the minimum parking rate will be multiplied by a fee multiplier which will be predetermined for every hour. The multiplier will be particularly high during rush hours and weekends. Local traffic data may also be used to determine these rates. Customers who then elect to make confirmed reservations would pay the sum of the dynamic parking fees for their time slot. Long-term customers who make guaranteed reservations will have smaller multipliers applied to their parking rate to encourage more drivers to park long-term. Walk-in customers would pay the same price charged to confirmed reservation customers plus a “surge” fee. This “surge” fee is rate multiplier that increases as the frequency of drivers that request walk-in parking per hour increases. This policy will maximize profits for the parking spot because it will pick up on sporadic surges in demand for parking and charge drivers accordingly. The system will also reduce the overall parking fees for particularly low frequencies of walk-in requests to encourage more drivers to use the parking garage. By lowering fees, the number of customers at the parking garage is more likely to be maximized which in turn raises profits.

(4) Another qualitative feature we propose is a new QR code tool associated with every profile. The parking garage system can recognize the license plate of customers with a registered profile. However, for customers with rental cars (unrecognizable plate, but the driver has a profile) a QR code can save them a lot of time in the parking garage, thereby increasing efficiency.

When the customer creates an account, they will receive a QR code associated with their profile that they can either print out or download to their phone. In the instance where a registered customer has a rental car, and the garage does not recognize the plate, the customer may show the scanner their QR code. The garage will then be able to identify the customer and let them enter. The time that the customer saves, especially for those who frequently rent cars, can lead to increased customer satisfaction. A similar QR code feature was utilized by the 2019 group. However, their feature had customers scan a QR code presented by the garage for registration, while ours has the garage scan a user’s code for entry [4]. We improve upon this feature through a more user-friendly experience where the customer does not have an extra step of scanning a QR code in order to park.

Overall, a fully automated garaging system can eliminate the need for parking attendants and optimize the management of occupancy. Furthermore, it provides a more reliable toll system and user-friendly search system for parking spots. Such a parking system will surely increase customer satisfaction.

Glossary of Terms

- **Account:** Customers can access the garage app using their username and password. Each customer has their own account, where they can manage information and reservations.
- **Customer:** A person who interacts with the garage app to purchase its services.
- **Database:** A part of the overall server, where all of the customer account data and information is recorded and stored.
- **Elevator:** A lift for cars that will transport them to the appropriate floor. An elevator is located at both the entrance and exit.
- **Manager:** A person who interacts with the garage app to manage its services.
- **Map:** The overall map of the garage, includes each floor and displays the location of each spot.
- **Owner:** A person who the garage belongs to.
- **Point(s):** A reward for customers for using the parking garage services. Points may be rewarded, deducted, and spent by customers for rewards.
- **Profile:** A part of the account. Here, customer information is stored.
- **Reservation:** An arrangement to park in a spot during a fixed, future time-frame. It is guaranteed that customers will be able to park in a spot if a reservation is made.
- **Reward/penalty system:** A system where customers receive points for good behavior, such as making reservations, and lose points for bad behavior, such as leaving a reservation late. Points may be redeemed for rewards. A negative point balance may result in penalties.
- **Scanner:** A camera located at the entrance of the garage that can recognize and read license plates and QR codes.
- **Spot:** An individual parking space where one car can park at a time.
- **System:** The overall parking garage project; includes the physical garage, the app, the different sensors, and the software that binds them all together.
- **User:** Any person who interacts with the garage app.
- **Walk-in:** Customers who have not reserved a parking spot. Unlike reservations, walk-in spot availability is not guaranteed.
- **Weight Sensor:** Located at every parking spot, this sensor determines if a car is currently located at the given spot.

SYSTEM REQUIREMENTS

Enumerated Functional Requirements

Identifier	Priority	Requirement
REQ-1	5	The system shall allow the customer to create an account on the app.
REQ-2	4	The system shall allow the customer to edit their own information in their accounts.
REQ-3	5	The system shall allow the customer to make a reservation or cancel a reservation.
REQ-4	2	The system shall send the customer an email confirming their reservation.
REQ-5	4	The system shall display the vacant spot that the customer will park at when they arrive at the entrance.
REQ-6	3	The system shall read a car’s license plate when it approaches the entrance gate.
REQ-7	3	The system shall request a QR code if the license plate is unrecognized.
REQ-8	3	The system shall be able to scan the QR code.
REQ-9	4	The system shall mark the time the customer enters and exits the lot.
REQ-10	4	The system shall update the map when a customer arrives at a spot or leaves a spot.
REQ-11	2	The system shall automatically deduct fee from the user’s account when they make a reservation.

REQ-12	2	The system shall automatically deduct fee from the user's account when they leave if there is no reservation.
REQ-13	4	The system shall keep track of the occupancy of the walk-in section to determine if there are still available walk-in spots.
REQ-14	4	The system shall automatically update points when the customer makes a reservation.
REQ-15	4	The system shall automatically deduct points when the customer overstays their reservation.
REQ-16	2	The system should notify the customer with an email or text that they have 15 minutes remaining in their reservation.
REQ-17	4	The system shall allow the garage manager to view all the reservations.
REQ-18	3	The system shall allow the garage manager to edit or cancel any reservations.
REQ-19	2	The system shall allow the garage manager to be able to adjust the pricing system.
REQ-20	2	The system shall allow the garage manager to be able to adjust the point system.
REQ-21	2	The system shall allow the garage manager to create rewards for customers to purchase with points.
REQ-22	5	The system shall keep statistics of the number of reservations made, reservations cancelled, and number of cars entering the lot.
REQ-23	1	The system shall allow the garage manager to access and view the statistics of the parking lot.
REQ-24	5	The system shall save the customer's information in the database.

REQ-25	4	The system shall allow the garage manager to view the information in the database.
REQ-26	3	The system should allow the garage manager to edit the information in the database.
REQ-27	1	The system shall allow the garage manager to add, edit or delete a customer account

Enumerated Nonfunctional Requirements

Identifier	**Priority**	**Requirement**
REQ-28	3	The system will have access to the internet in order to connect with customers.
REQ-29	1	The system will have cameras to capture the license plate of a moving car.
REQ-30	2	The system will have weight sensors at each parking spot to determine a vacancy.
REQ-31	3	The system will allow for disability to be taken into account while assigning parking sports.
REQ-32	5	The system will store database information in a file system.
REQ-33	5	The system will use an app that customers can download to create accounts, make reservations, and access any information about the parking garage or their account.
REQ-34	1	The system will use an elevator system to guide cars towards open spots.

REQ-35	3	The system will use QR codes to handle customers who have unrecognized license plates.
REQ-36	1	The system will include a simulation to determine if the parking garage will work in real-world situations. It will cover the process of what a customer would typically go through while parking and how the system reacts to each action in that process.

User Interface Requirements

Identifier	Priority	Requirement
REQ-37	5	After first downloading the app, the customer will be required to create a username and password.
REQ-38	5	Upon opening the app, the customer will be required to login with their username and password (they may use facial/fingerprint recognition if they opt to do so).
REQ-39	4	On the account page, the customer will be able to click a button to go to reservation(s).
REQ-40	4	On the account page, the customer will be able to click a button to view their profile.
REQ-41	3	On the account page, the customer will be able to click a button to redeem points for rewards.
REQ-42	1	On the account page, the customer will be able to sign out of their account.
REQ-43	2	On the profile page, the customer will be able to click a button to edit account information.

REQ-44	3	On the profile page, the customer will be able to download their QR code.
REQ-45	3	On the profile page, the customer will be able to check a box for handicap accommodations and verify handicap information in a textbox (they may also take a picture to submit the alphanumeric code).
REQ-46	4	On the reservation page, the customer will see a calendar view that shows the number of available spots per time.
REQ-47	3	On the reservation page, the customer will be able to edit their reservations.
REQ-48	3	On the reservation page, the customer will be able to cancel reservations.
REQ-49	2	On the payment page, the customer will be able to save a credit card to their account.
REQ-50	4	Non Members will be able to create a reservation via guest login.
REQ-51	3	On the manager portal, managers will be able to see the number of available spots.
REQ-52	1	On the manager portal, managers will be able to click a button to check garage statistics over time.
REQ-53	3	On the manager portal, managers will be able to click a button to view all registered customers.
REQ-54	2	On the manager portal, managers will be able to click a button to change the parking price for a specified time period.
REQ-55	3	On the manager portal, managers will be able to edit the reward/penalty system.

Customer Interface Sketches

Several mockups demonstrating the user interface requirements are shown. See section, “User Interface Specification,” for full sets of graphics alongside use cases.

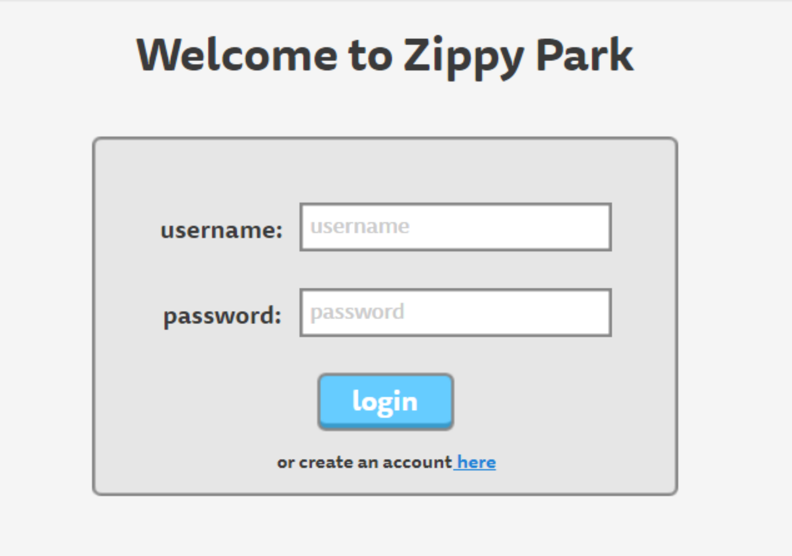


Figure 1. Customer login screen.

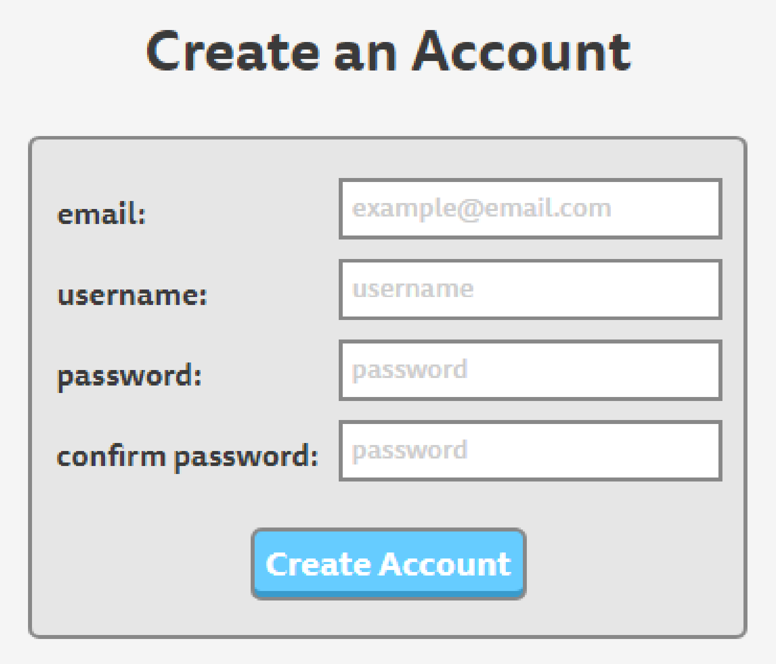


Figure 2. Customer account creation screen.

FUNCTIONAL REQUIREMENTS SPECIFICATION

Stakeholders

ZippyPark is intended to increase the efficiency of operation of parking garages, and therefore, maximize the profits. This system will be of interest to people and organization who own, operate, or maintain parking garages, as well as individuals who purchase the services of parking garages. Possible stakeholders include:

- Parking garage owners
- Parking garage managers
- Customers – reserved or walk-in

Actors and Goals

Actors are denoted as the following types:
I = initiating
S = supporting
O = offstage

Actor	Type	Goal	Use Case(s)
Customer, Garage Manager	I	To create, edit, or delete customer account by clicking the applicable buttons on the parking garage app	UC-1, UC-2, UC-3
Mobile App	S	To collect and display customer information	
Manager Portal	S	To display existing customer information and collect new or updated information	
Database	O	Stores updated customer information	
Customer, Garage Manager	I	To create, edit, or delete customer reservation by clicking on the applicable buttons on the parking garage app	UC-4, UC-5, UC-6
Mobile App	S	To collect and display reservation information (time, date)	

Manager Portal	S	To display existing customer reservations and collect new or updated reservation information	
Database	O	Stores updated reservation information	
Customer	I	To enter or exit the garage	UC-7, UC-8
Scanner	S	To scan and read the license plate of the customer’s car	
Database	O	Searches for license plate information to find what floor to send the car to, sends this information to elevator	
Elevator	S	To bring customer to correct floor	
Customer	I	To park or pull out of parking spot	UC-9
Database	O	Changes parking spot status depending on sensor output	
Garage Manager	I	To view parking garage statistics on the manager portal	UC-10
Manager Portal	S	To retrieve parking garage statistics from database and display on manager portal	
Database	O	Stores parking garage statistics	
Customer	I	To view points	UC-11
Garage Manager	I	To view and update customers’ points	
Mobile App	S	To display customer points as requested	
Manager Portal	S	To display existing customer points and collect information on updating points	
Database	O	Stores customer points	
Customer	I	To pay for parking reservation	UC-12
Mobile App	S	To collect customer payment information including credit card number and billing address	
Database	O	Stores customer payment information	

Use Cases

Casual Description

	Use Case	Description
UC-#1	Create Account	Utilizes REQ-1, REQ-22, REQ-23, and REQ-26. Allow new customers to register for the first time and allow managers to create a new customer account.
UC-#2	Edit Account	Utilizes REQ-2 and REQ-25, REQ-26. Allow current customers to update account information.
UC-#3	Delete Account	Utilizes REQ-26. Garage owners have ability to delete an account if needed
UC-#4	Create Reservation	Utilizes REQ-3, REQ-4, REQ-16, REQ-17 and REQ-21. Customers can create reservations which managers are able to view. Customers receive confirmation email upon creation.
UC-#5	Edit Reservation	Utilizes REQ-3 REQ-18, REQ-21 and REQ-25. Customers and managers can edit an existing reservation
UC-#6	Cancel Reservation	Utilizes REQ-3, REQ-18, REQ-19 and REQ-21. Customers and managers can cancel an existing reservation
UC-#7	Enter Garage	Utilizes REQ-5, REQ-6, REQ-7, REQ-8, REQ-9, and REQ-21. As the customer enters the garage it scans their license plate and determines if they have a reservation or are a walk-in and also displays the vacant spot available.
UC-#8	Exit Garage	Utilizes REQ-9. As the customer exits the garage it records the time and adds it to their reservation history
UC-#9	Update Spot Status	Utilizes REQ-10, REQ-11, REQ-21. Records in the database whether a parking spot is occupied or not.
UC-#10	Display Stats	Utilizes REQ-13, REQ-17, REQ-21, REQ-22, REQ-24 and REQ-25. Retrieves information from the database and displays updated trends to managers.
UC-#11	Points Management	Utilizes REQ-14, REQ-15, REQ-19, REQ-20. Customers can view their points, and managers can adjust the points reward scheme.
UC-#12	Payment	Utilizes REQ-12. If a customer has a reservation, the fee should be deducted prior to parking and if the customer is a

		walk-in then the fee should be deducted after they leave the garage. Also, if a reserved customer overstays the parking spot a late fee is deducted.

Use Case Diagram

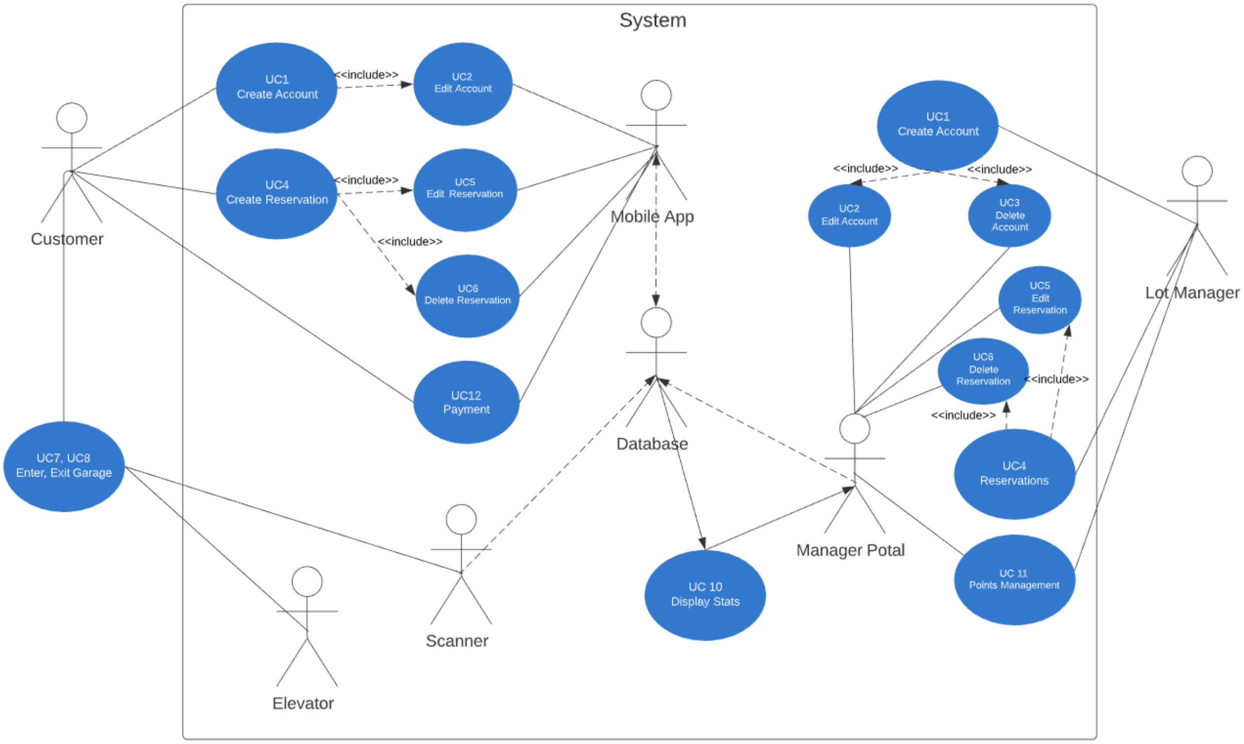
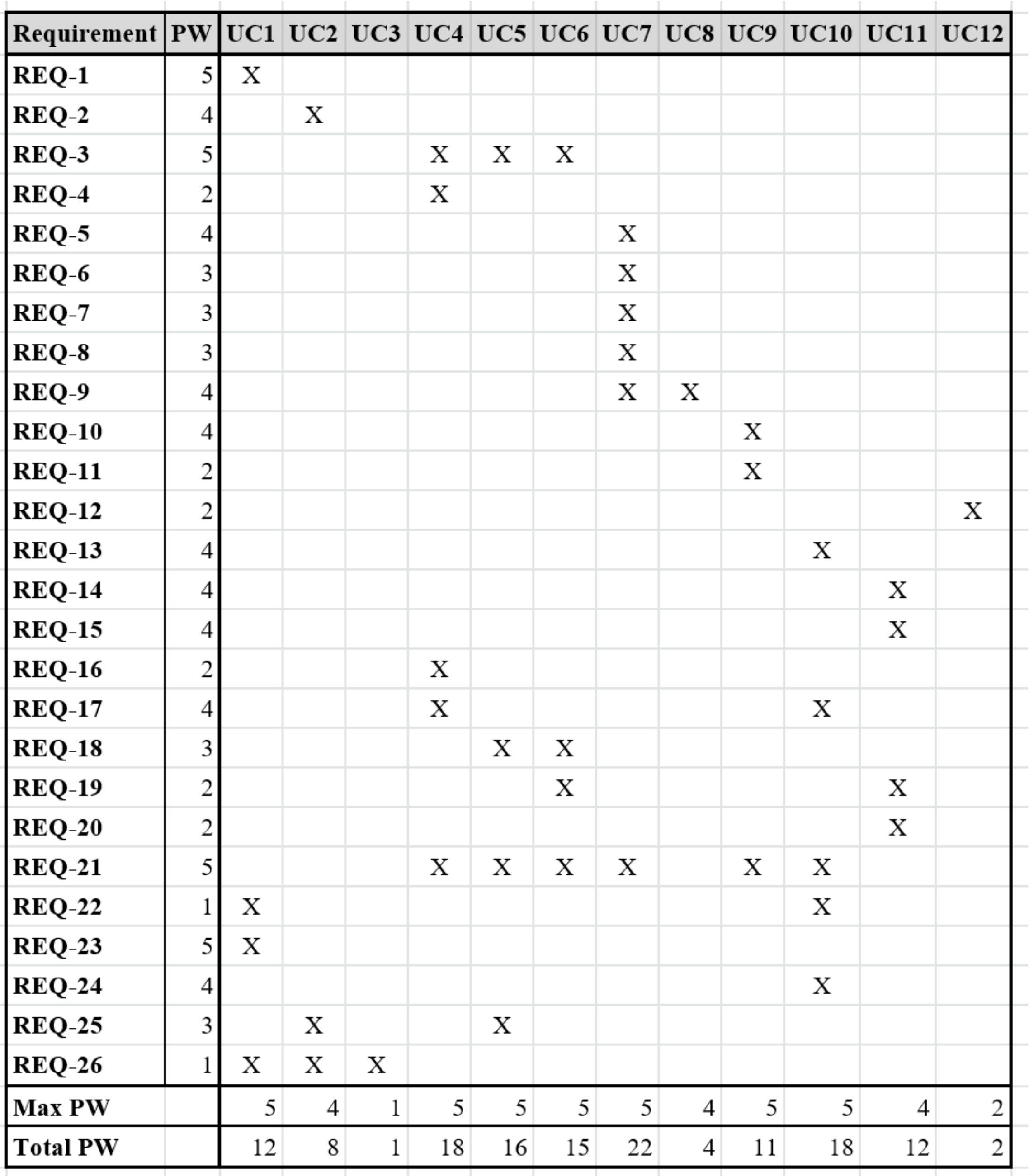


Figure 3. Diagram with all use cases.

Traceability Matrix

This matrix maps the system requirements to the use cases. The priority weight (PW) of each requirement is given on a scale from 1 to 5, with 5 being the highest priority.



Requirement	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12
REQ-1	5	X											
REQ-2	4		X										
REQ-3	5				X	X	X						
REQ-4	2				X								
REQ-5	4							X					
REQ-6	3							X					
REQ-7	3							X					
REQ-8	3							X					
REQ-9	4							X	X				
REQ-10	4									X			
REQ-11	2									X			
REQ-12	2												X
REQ-13	4										X		
REQ-14	4											X	
REQ-15	4											X	
REQ-16	2				X								
REQ-17	4				X						X		
REQ-18	3					X	X						
REQ-19	2						X					X	
REQ-20	2											X	
REQ-21	5				X	X	X	X		X	X		
REQ-22	1	X									X		
REQ-23	5	X											
REQ-24	4										X		
REQ-25	3		X			X							
REQ-26	1	X	X	X									
Max PW		5	4	1	5	5	5	5	4	5	5	4	2
Total PW		12	8	1	18	16	15	22	4	11	18	12	2

Fully-Dressed Descriptions

According to the traceability matrix, the use cases with the highest priority weight are UC-4, 5, 7, 10. The fully-dressed descriptions of each are given below.

Use Case: UC-4 Create Reservation

Related Requirements: REQ-3, REQ-4, REQ-16, REQ-17, REQ-21

Initiating Actors: Garage Customer

Actor’s Goals: To reserve a parking spot to use for a set amount of time

Participating Actors: ParkingGarageApp, InformationDatabase

Preconditions:

- The customer has an existing profile stored in the database
- The customer has logged in to the ParkingGarageApp
- The ParkingGarageApp displays a calendar for the current month as well as available time slots

Postconditions:

- The system reduced the number of available spots for the reserved time
- The new reservation is tied to the user’s account reservations

Flow of events for main success scenario:
→ 1.) Customer selects the “Reservations” item on the account page
← 2.) System displays the Reservations page with a calendar for the current month
→ 3.) Customer selects a date with start and end times
← 4.) System checks with the InformationDatabase to confirm availability
← 5.) System:
(a) saves the reservation to the customer’s account,
(b) reduces availability for the selected time, and
(c) signals to the customer that reservation was successful

Flow of events for Extensions (Alternate Scenarios):
4a. System finds time slot is invalid/filled
← 1.) System informs customer that the requested time is unavailable
→ 2.) Customer selects a valid date and time
← 3.) Same as Step 4 above

Use Case: UC-5 Edit Reservation

Related Requirements: REQ-3, REQ-18, REQ-21, REQ-25

Initiating Actors: Garage Customer, Garage Manager

Actor's Goals: To edit a reservation previously made with the parking garage

Participating Actors: ParkingGarageApp/Website, InformationDatabase

- The manager uses ParkingGarageWebsite and the customer ParkingGarageApp
    - In this use case, both have the same capabilities with different people accessing them

Preconditions:

- At least one reservation for the customer of interest must exist in the system

Postconditions:

- An edited version of the reservation will take the place of the original in the database
- The original reservation is wiped from the database

Flow of events for main success scenario:

→ 1.) User chooses option to edit a reservation of the ParkingGarageApp/Website

← 2.) ParkingGarageApp/Website communicates with InformationDatabase and displays all upcoming reservations under customer of interest in chronological order

→ 3.) User chooses the reservation they would like to edit and change the time of day, duration, or date of reservation

→ 4.) ParkingGarageApp/Website asks user if they would like to proceed as changes can't be undone

← 5.) User confirms and agrees to increased/decreased charges on the account of the customer of interest if the duration of the reservation was lengthened/shortened.

→ 6.) ParkingGarageApp/Website communicates with system to edit the reservation in the InformationDatabase

← 7.) ParkingGarageApp/Website sends a confirmation of completed editing

Use Case: UC-7 Enter Garage

Related Requirements: REQ-5, REQ-6, REQ-7, REQ-8, REQ-9, REQ-21

Initiating Actors: Garage Customer

Actor's Goals: To enter the garage and be directed to a parking spot

Participating Actors: Scanner, Elevator, Database

Preconditions:

- The customer has yet to enter the garage

Postconditions:

- The customer is directed either to the appropriate parking spot, or is asked to back out of the elevator

Flow of events for main success scenario:
→ 1.) Customer navigates onto the elevator
← 2.) Scanner scans the license plate of the customer's car
← 3.) License plate number is sent to and compared to existing reservations in database
← 4.) Database confirms whether the customer has a reservation or not
a) If customer has a reservation continue to step 6
b) If customer does not have a reservation continue to step 5
← 5.) The system determines there are available walk-in parking spots
← 6.) System assigns customer a spot and they drive to it

Flow of events for Extensions (Alternate Scenarios):
← 5a. The system determines if there are no more available walk-in parking spots
→ 1.) Customer backs out of elevator and exits the garage

Use Case: UC-10 Display Statistics

Related Requirements: REQ-13, REQ-17, REQ-21, REQ-22, REQ-24, REQ-25

Initiating Actors: Garage Manager

Actor's Goals: To pull and display updated parking trend data to parking managers.

Participating Actors: ParkingGarageWebsite, InformationDatabase

Preconditions:

- Garage Manager has access to an authorized account
- InformationDatabase has existing records to display

Postconditions:

- The ParkingGarageWebsite will display graphs and tables constructed from customer parking data

Flow of events for main success scenario:
→ 1.) Manager logs in using authorized account information and selects "View Stats"
← 2.) InformationDatabase checks credentials and provides parking garage records
← 3.) System compiles and sorts data into organized format
← 4.) ParkingGarageWebsite displays stats on-screen

System Sequence Diagrams

The system sequence diagrams for the fully-dressed descriptions are given below.

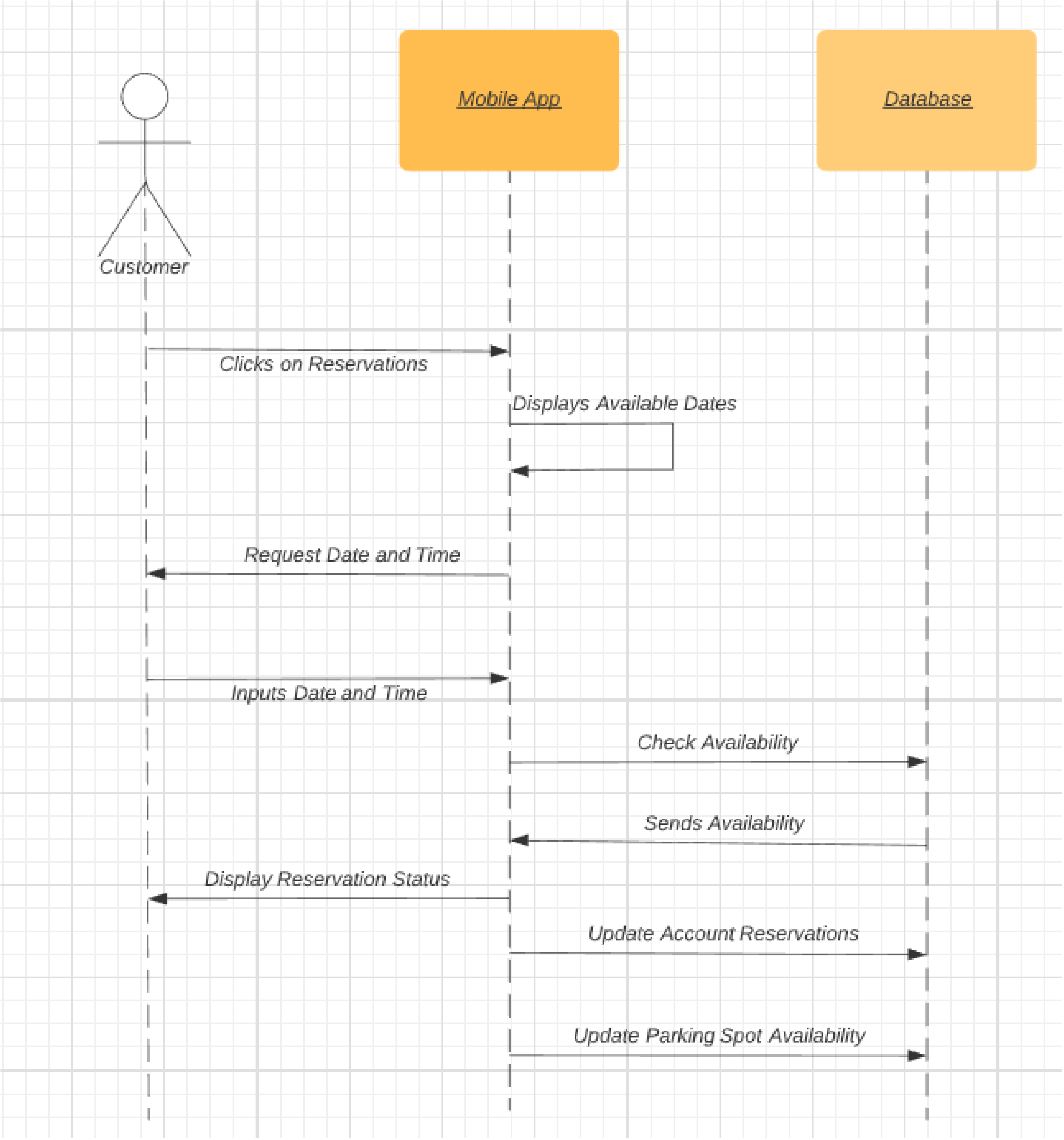


Figure 4. UC-4 Create Reservation.

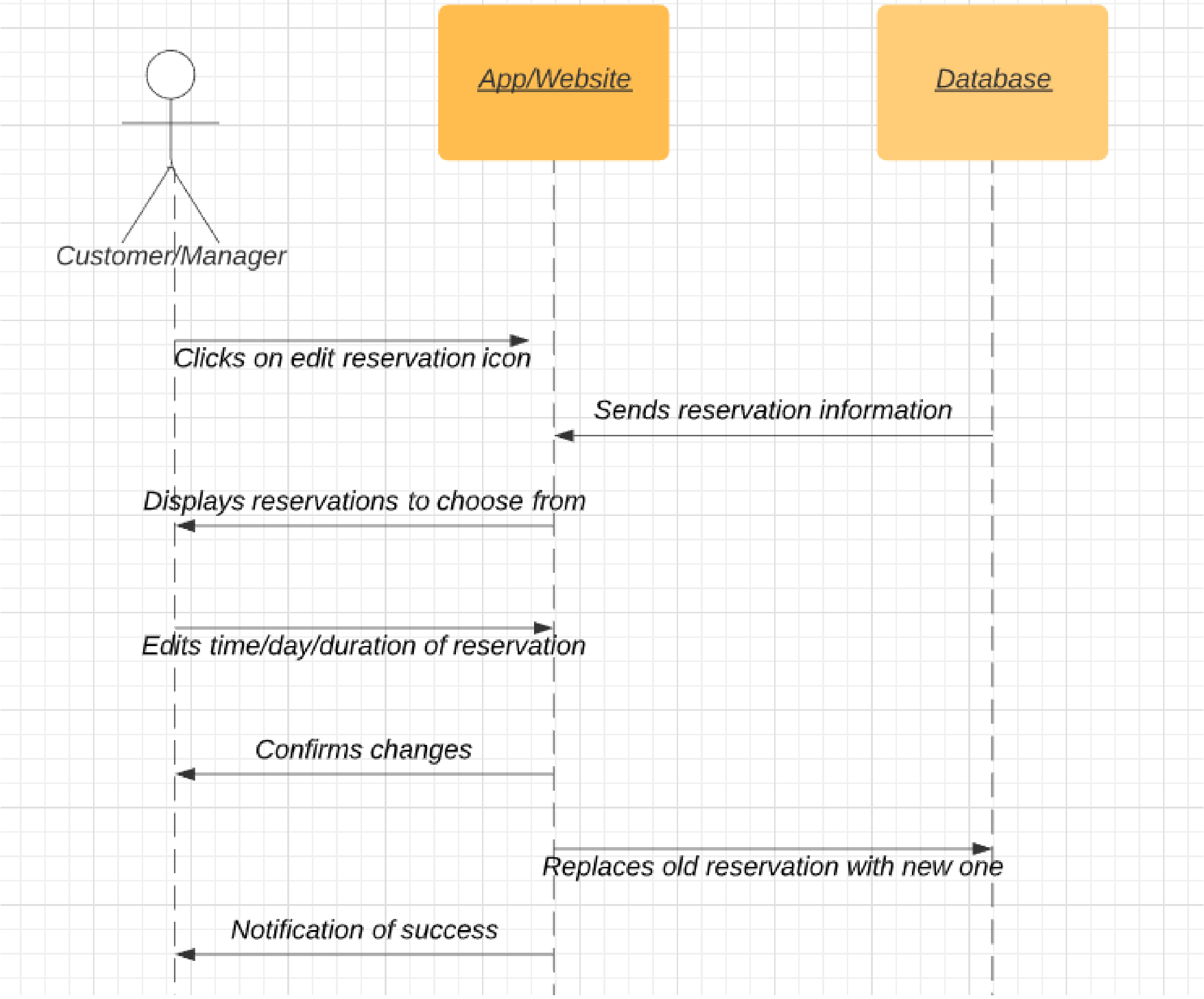

App/Website
Database
Customer/Manager
Clicks on edit reservation icon
Sends reservation information
Displays reservations to choose from
Edits time/day/duration of reservation
Confirms changes
Replaces old reservation with new one
Notification of success

Figure 5. UC-5. Edit Reservation.

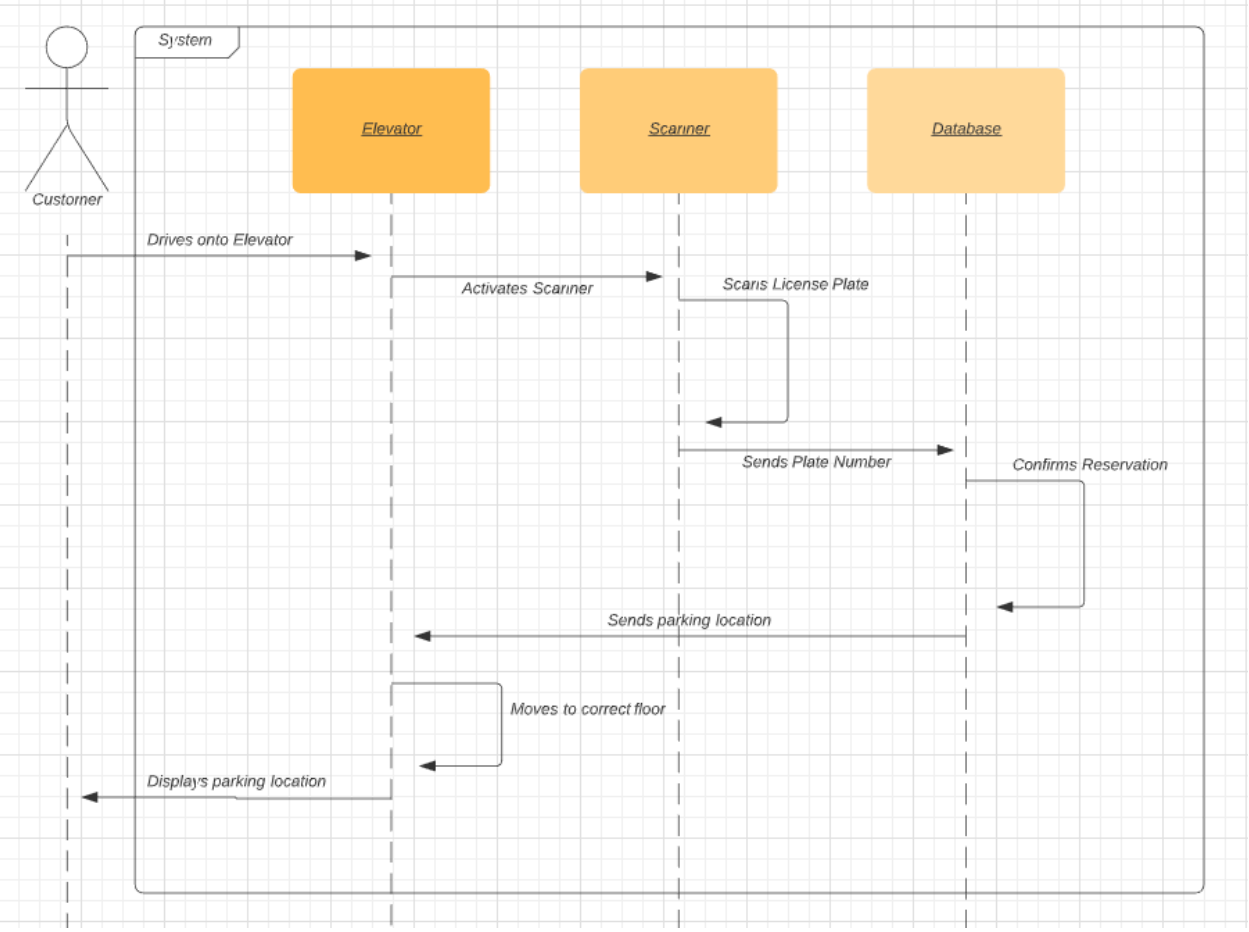


Figure 6. UC-7 Enter Garage.

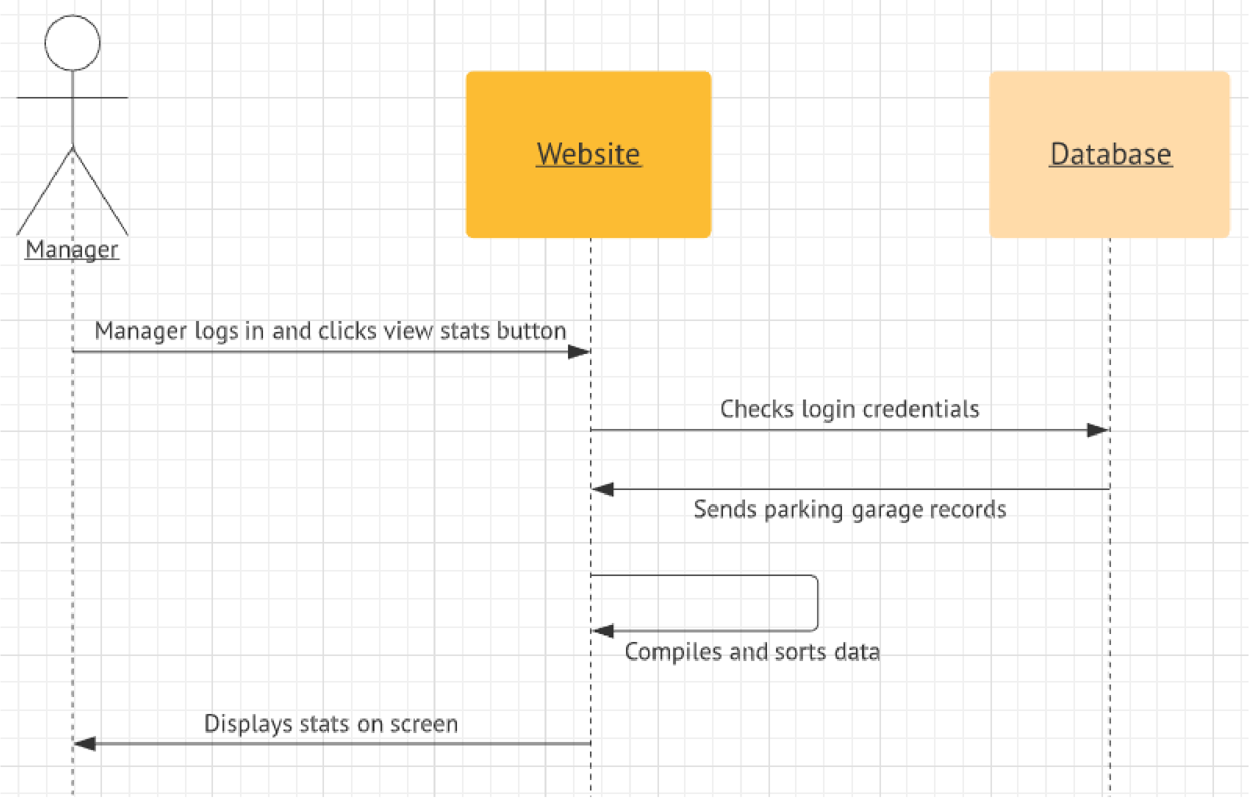


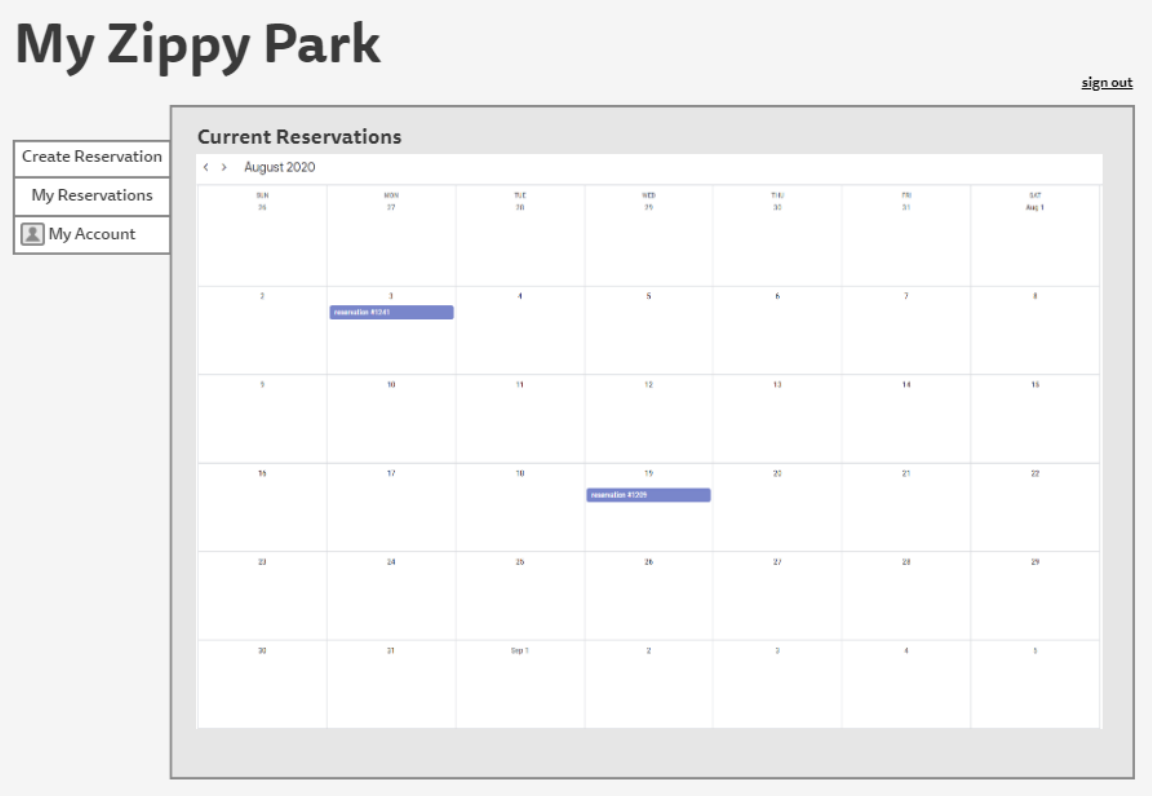
Figure 7. UC-10 Display Statistics.

USER INTERFACE SPECIFICATION

Preliminary Design

Our user interface design is composed of a website with multiple pages that each have their own functional purpose (to be described further in detail). The goal of our design is to provide an interface for users that is intuitive to navigate. The minimalist look and clarity of instruction both contribute to the user-friendliness of our website.

Shown below is the home page of Zippy Park's website. Displayed is a calendar showing current reservations. On the left side are tabs that link to the pages for creating a reservation ("Create Reservation"), viewing current reservations as a list ("My Reservations"), and viewing account details "My Account"). Customers may also log out of their account via the "sign out" button located at the top right of the page.



SUN	MON	TUE	WED	THU	FRI	SAT
26	27	28	29	30	31	Aug 1
2	3 reservation #1241	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19 reservation #1209	20	21	22
23	24	25	26	27	28	29
30	31	Sep 1	2	3	4	5

Figure 8. Reservation screen.

UC-1: Create Account

Upon entering the website, the customer will be prompted to login to their account on the welcome page. They can also register for an account by clicking a hyperlink that will lead them to an account creation page.

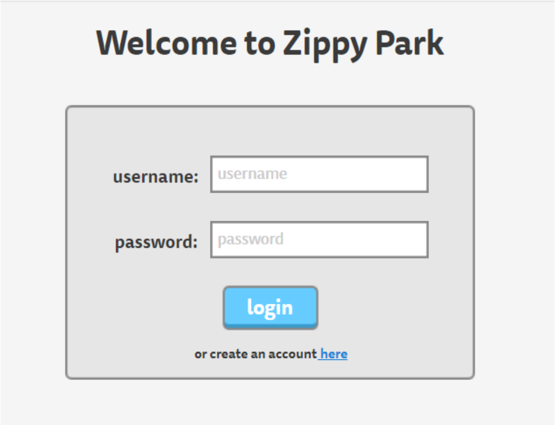


Figure 9. Customer login screen.

If the customer wishes to create an account, they must click on “here” under the login button. They will then be directed to the following page to create their account. In order to create an account, the customer will need to provide a valid email address, unique username, and unique password.

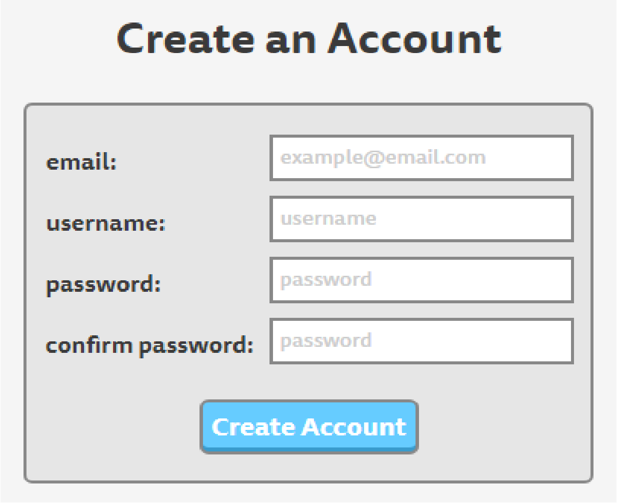
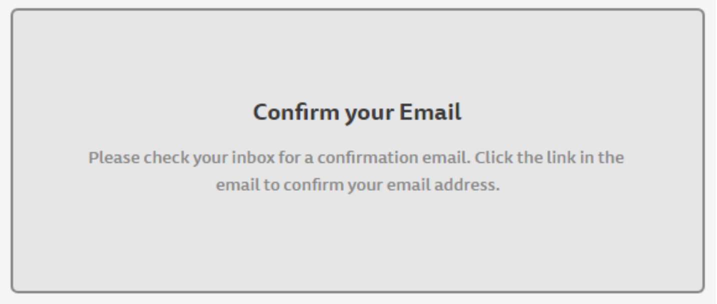


Figure 10. Customer account creation screen.

Once the customer inputs all the information and clicks the “create account” button, they will be directed to the following page asking for email confirmation.



Confirm your Email

Please check your inbox for a confirmation email. Click the link in the email to confirm your email address.

Figure 11. Confirmation screen.

The customer will then log into their email and confirm their account. They may then return to the website to log in.

UC-2: Edit Account

Once the customer logs in, if they have not yet registered personal information including a profile, car registration, and payment method, then they will not be able to create a reservation. To do so they must edit their account details. Customers may also use this page to edit their existing information.

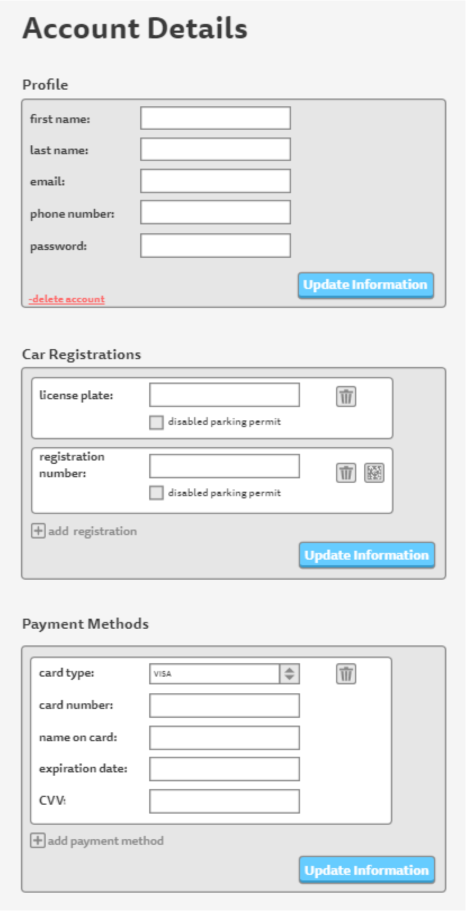


Figure 12. Customer information screen.

On this page, customers can update their profile, car registrations, and payment methods.

1. The profile includes fields for the user's first and last name, email, phone number, and password.
2. Multiple car registrations can be added to an account. Fields are provided for the required information, which is either the car's license plate number or, for rented cars, registration number. The option of marking a car as possessing a disabled parking permit is also available for each entry. Car registrations can be deleted using a button at the side of each entry. For cars that have a registration number, a QR code is provided and can be accessed with a button to the side of the entry.
3. Multiple payment methods can be added to an account. Fields are provided for the required information for each entry is the card type, card number, name on card,

expiration date, and CVV. Payment methods can be deleted using a button at the side of the entry.

For each section, once the customer is finished with updating their information, they must click the "update information" button to save their changes.

UC-3: Delete Account

The customer has the ability to delete their account. This feature is accessible on the "Account Details" page in the profile section.

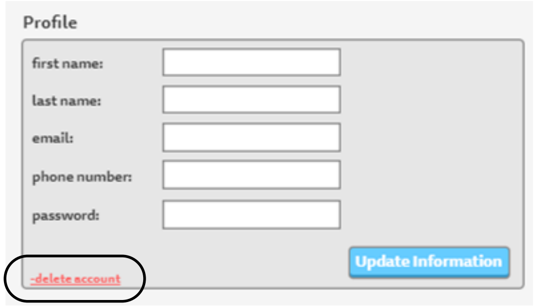
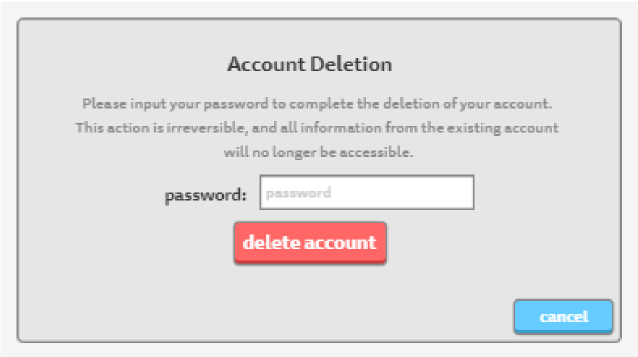


Figure 13. Delete account feature.

If a customer is to click this link, they will be directed to the following page asking for confirmation in deleting the account. It requires the customer to input their password to verify that the correct customer wants to delete their account. Once a customer inputs their password, they will click the "delete account" button. If a customer does not wish to continue with deleting their account, they may cancel the request and be directed back to the "Account Details" page through the "cancel" button.



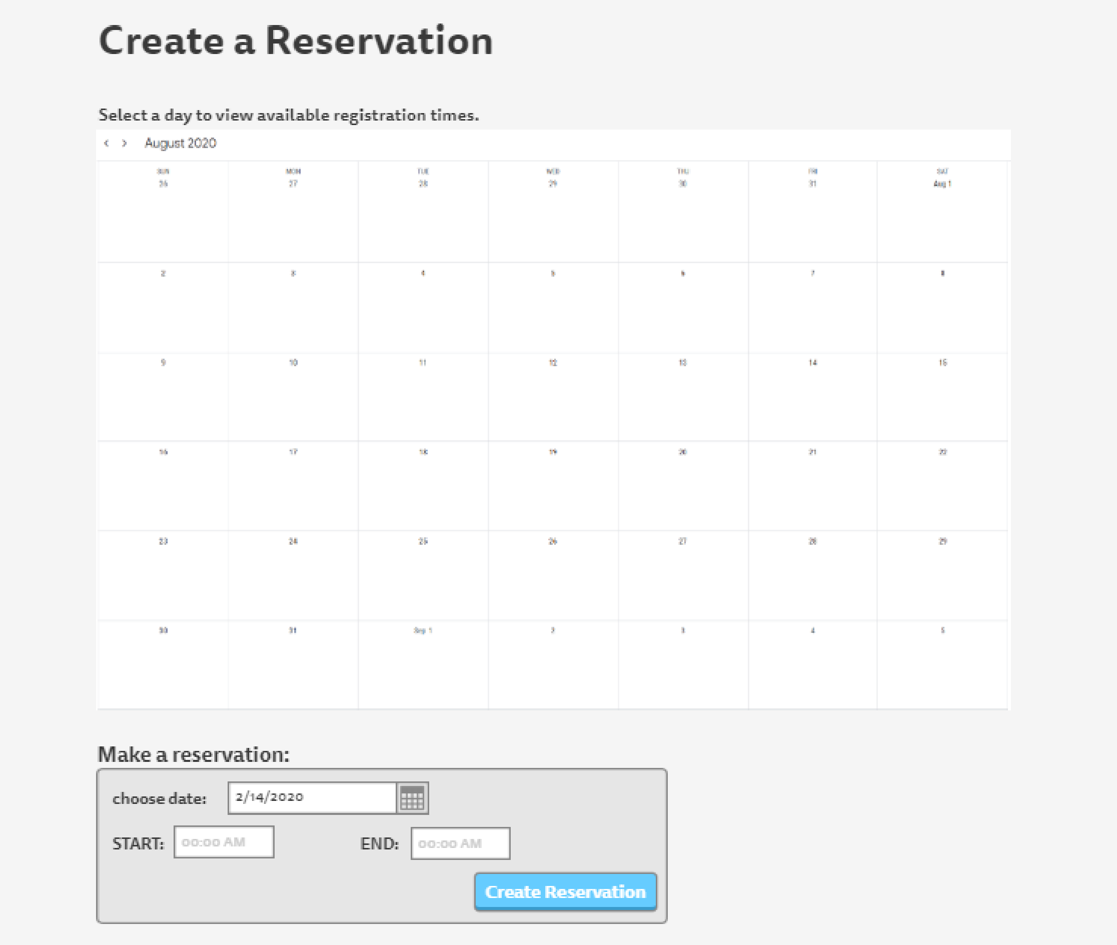
Account Deletion

Please input your password to complete the deletion of your account. This action is irreversible, and all information from the existing account will no longer be accessible.

Figure 14. Final delete account screen.

UC-4: Create Reservation

A customer can create a new reservation on the “Create a Reservation” page. On this page, the customer is provided with a calendar where they can navigate and select a date to see the available time slots. They can then make a reservation by choosing the date and inputting a start and end time for the reservation. They submit the reservation through the “create reservation” button.



Create a Reservation

Select a day to view available registration times.

‹ › August 2020

SUN	MON	TUE	WED	THU	FRI	SAT
26	27	28	29	30	31	Aug 1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	Sep 1	2	3	4	5

Make a reservation:

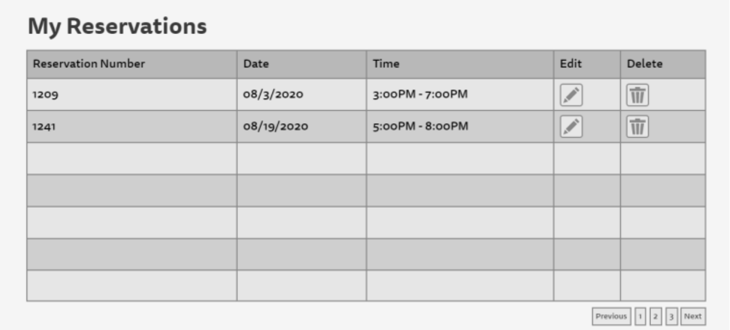
choose date: 2/14/2020

START: 00:00 AM END: 00:00 AM

Figure 15. Create reservation screen.

UC-5: Edit Reservation

The customer can view all their reservations on the “My Reservation” page. Through this page, they may also select a reservation to either edit or delete.

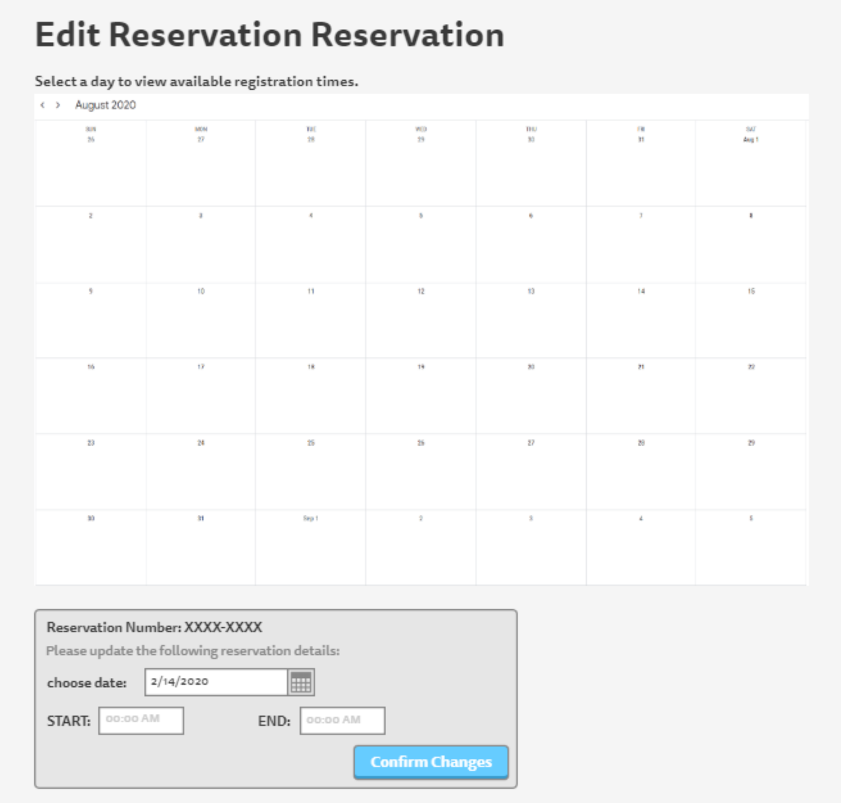


My Reservations

Reservation Number	Date	Time	Edit	Delete
1209	08/3/2020	3:00PM - 7:00PM	✎	🗑
1241	08/19/2020	5:00PM - 8:00PM	✎	🗑

Figure 16. Custome reservation record screen.

If the customer chooses to edit the reservation they will be brought to the following page. Similar to the page for creating a reservation, they will be provided with a calendar showing available reservation times. They can then edit their reservation and click the “Confirm Changes” button to save their new reservation information.



Edit Reservation Reservation

Select a day to view available registration times.

‹ › August 2020

SUN	MON	TUE	WED	THU	FRI	SAT
26	27	28	29	30	31	Aug 1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	Sep 1	2	3	4	5

Reservation Number: XXXX-XXXX

Please update the following reservation details:

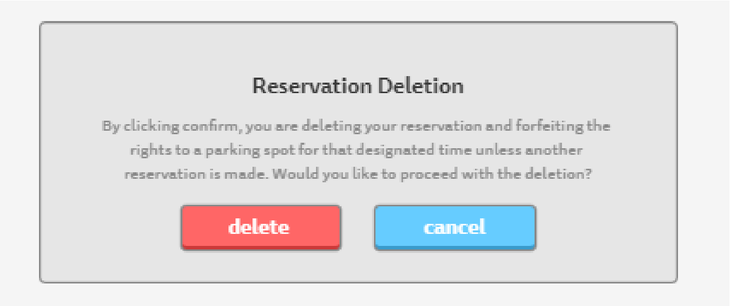
choose date: 2/14/2020

START: 00:00 AM END: 00:00 AM

Figure 17. Customer edit reservation screen.

UC-6: Cancel Reservation

From the “My Reservations” page, if the customer chooses to delete their reservation, they will be brought to the following confirmation page. They may click the “delete” button to officially cancel their reservation or the “cancel” button if they do not wish to delete the reservation.



Reservation Deletion

By clicking confirm, you are deleting your reservation and forfeiting the rights to a parking spot for that designated time unless another reservation is made. Would you like to proceed with the deletion?

Figure 18. Customer reservation cancellation screen.

User Effort Estimation

The purpose of our user interface design is to make interaction between the user and our applications as simple as possible. Thus, we created a design to keep the number of user mouse clicks and keystrokes to a minimum. The following are some test cases for each use case and the estimated user effort for each scenario. Each case considers that the user is already on the page on which they can complete the task (i.e. they do not have to navigate to this page).

1. Test Case Description: Create an Account

NAVIGATION: total 2 mouse clicks as follows

a. Click "here" hyperlink (from welcome page)

---*after completing data entry as shown below*---

b. Click "Create Account" button to finish

DATA ENTRY: total 1 mouse click and 3+ keystrokes as follows

a. Click cursor to "email" text field
b. Press the keys to enter their email address (indeterminate number of characters)
c. Press the "Tab" key to move to the next text field ("username")
d. Press the keys to enter their username (indeterminate number of characters)
e. Press the "Tab" key to move to the next text field ("password")
f. Press the keys to enter their username (indeterminate number of characters)
g. Press the "Tab" key to move to the next text field ("confirm password")
h. Press the keys to enter their username (indeterminate number of characters)

TOTAL: 3 mouse clicks and 3+ keystrokes

2. Test Case Description: Edit Account Details

The user wishes to change only the profile information in this scenario.

NAVIGATION: total 1 mouse click as follows

---*after completing data entry as shown below*---

a. Click "Update Information" button to finish

DATA ENTRY: total 1 mouse click and 14+ keystrokes as follows

a. Click cursor to "first name" text field
b. Press the keys to enter their first name (indeterminate number of characters)
c. Press the "Tab" key to move to the next text field ("last name")
d. Press the keys to enter their last name (indeterminate number of characters)
e. Press the "Tab" key to move to the next text field ("email")
f. Press the keys to enter their email (indeterminate number of characters)
g. Press the "Tab" key to move to the next text field ("phone number")
h. Press the keys to enter their phone number (10-digit phone number)

i. Press the “Tab” key to move to the next text field (“password”)
j. Press the keys to enter their password (indeterminate number of characters)

TOTAL: 2 mouse clicks and 14+ keystrokes

3. Test Case Description: Delete Account

NAVIGATION: total 2 mouse clicks as follows

a. Click “-delete account” hyperlink

---*after completing data entry as shown below*---

b. Click “delete account” button to finish

DATA ENTRY: total 1 mouse click and indeterminant keystrokes as follows

a. Click cursor to “password” text field
b. Press the keys to enter their password (indeterminate number of characters)

TOTAL: 3 mouse clicks and indeterminant keystrokes

4. Test Case Description: Create a Reservation

NAVIGATION: total 1 mouse click as follows

---*after completing data entry as shown below*---

a. Click “Create Reservation” button to finish

DATA ENTRY: total 1 mouse click and 22 keystrokes as follows

a. Click cursor to “choose date” text field
b. Press keys to enter date (8 keystrokes MM/DD/YYYY; no need to type “/”)
c. Press the “Tab” key to move to the next text field (“START”)
d. Press keys to enter time (6 keystrokes XX:XX am)
e. Press the “Tab” key to move to the next text field (“END”)
f. Press keys to enter time (6 keystrokes XX:XX pm)

TOTAL: 2 mouse clicks and 22 keystrokes

5. Test Case Description: Editing Reservation

The customer wishes to change only the start and end times of the reservation in this scenario.

NAVIGATION: total 1 mouse click as follows

---*after completing data entry as shown below*---

a. Click “Confirm Changes” button to finish

DATA ENTRY: total 1 mouse click and 13 keystrokes as follows

a. Click cursor to “START” text field
b. Press keys to enter time (6 keystrokes XX:XX am)
c. Press the “Tab” key to move to the next text field (“END”)
d. Press keys to enter time (6 keystrokes XX:XX pm)

TOTAL: 2 mouse clicks and 13 keystrokes

6. Test Case Description: Cancel Reservation (Profile Information)

The customer starts on the “My Reservations” page in this case.

NAVIGATION: total 2 mouse clicks as follows

a. Click delete button for corresponding reservation
b. Click “delete” button to confirm deletion to finish

DATA ENTRY: does not require any data entry

TOTAL: 2 mouse clicks

DOMAIN ANALYSIS

Domain Model

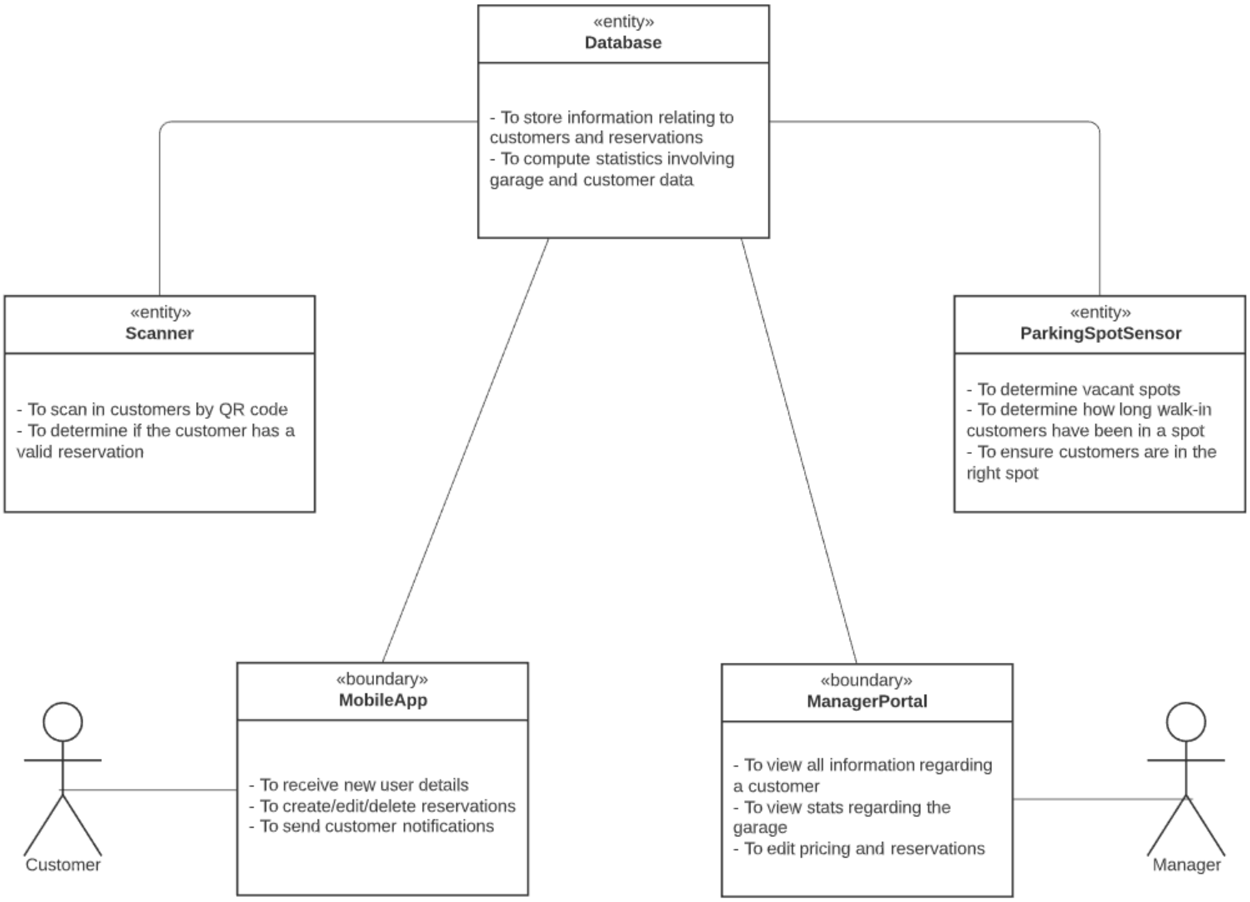

«entity»
Database
- To store information relating to customers and reservations
- To compute statistics involving garage and customer data
«entity»
Scanner
- To scan in customers by QR code
- To determine if the customer has a valid reservation
«entity»
ParkingSpotSensor
- To determine vacant spots
- To determine how long walk-in customers have been in a spot
- To ensure customers are in the right spot
«boundary»
MobileApp
- To receive new user details
- To create/edit/delete reservations
- To send customer notifications
«boundary»
ManagerPortal
- To view all information regarding a customer
- To view stats regarding the garage
- To edit pricing and reservations
Customer
Manager

Figure 19. Domain Model.

The domain model shown above was derived directly from the list of use cases and requirements for this project. This model displays how this complex project has been divided into a few concepts, based on the number of actors involved, to make it much simpler to understand. Information regarding the connections between each concept and certain uses cases, in addition to their priority weights, can be found in the traceability matrix below.

We chose to split up the inner workings of this project into five main concepts, which will all work together to simulate the procedure of our parking garage:

1. Mobile App: the first interaction our customers will have with our garage is creating a new account via the mobile app. The app will serve as their main interface for managing their reservations and viewing their profile.
2. Scanner: a system that will record which customer has entered and exited the garage, identifying the user by their license plate number. Upon entry, it will assign the customer a parking spot, and all information received by the scanner is sent to the database.
3. Parking Spot Sensor: one sensor mounted at each parking spot will determine whether that spot is currently occupied or not, and communicate this information with the database. This helps to ensure that the customers are parked in the correct spot, and whether they have arrived and departed from their spot on time.
4. Manager Portal: displays important statistics to managers, so they can view analytics on the performance of the garage. The portal serves as the main interface for managers to interact with the garage.
5. Database: stores all information regarding reservations, available spots, and customer data, and provides centralized access to all entities.

The Mobile App and Manager Portal are labeled as boundary concepts, because they will serve as the main interface for user interaction. All other concepts are labeled as entities, since information will be communicated between them with no need for user interaction.

Concept Definition

Responsibilities are denoted as the following types:
K = knowing
D = doing
N = neither

Responsibility Description	**Type**	**Concept Name**
To receive information on a new customer	K	Mobile App
To store information relating to customer and reservations	K	Database
To create/edit/delete reservations	D	Mobile App
To determine vacant spots	D	Parking Spot Sensor
To view all information regarding a customer	K	Manager Portal
To view stats regarding garage	K	Manager Portal
To compute data involving the garage/customers	D	Database
To determine how long walk-in customers have been in a spot	D	Parking Spot Sensor
To scan in customers by QR codes	D	Scanner
To determine if the customer has a valid reservation	D	Scanner
To edit reservation information	D	Mobile App
To edit pricing and reservations	D	Manager Portal
To send customer notifications	D	Mobile App
To ensure customers are in the right spots	D	Parking Spot Sensors

Association Definitions

Concept Pair	**Association Description**	**Association Name**
Mobile App ↔ database	App receives information from the customer regarding personal details, billing information, or reservations and sends this to the database to keep in records.	Store customer information

Manager portal ↔ database	Manager portal allows for the manager to view and edit information in the database regarding pricing, reservations, user stats, etc.	Exchange or change information
Scanner ↔ database	Scans in user and sends information regarding entry over to the database.	Scanning in
Parking spot sensors ↔ database	Updates database entries on which spots are available or occupied. Allows for the assigning of spots on user entry.	Assigning spots

Attribute Definitions

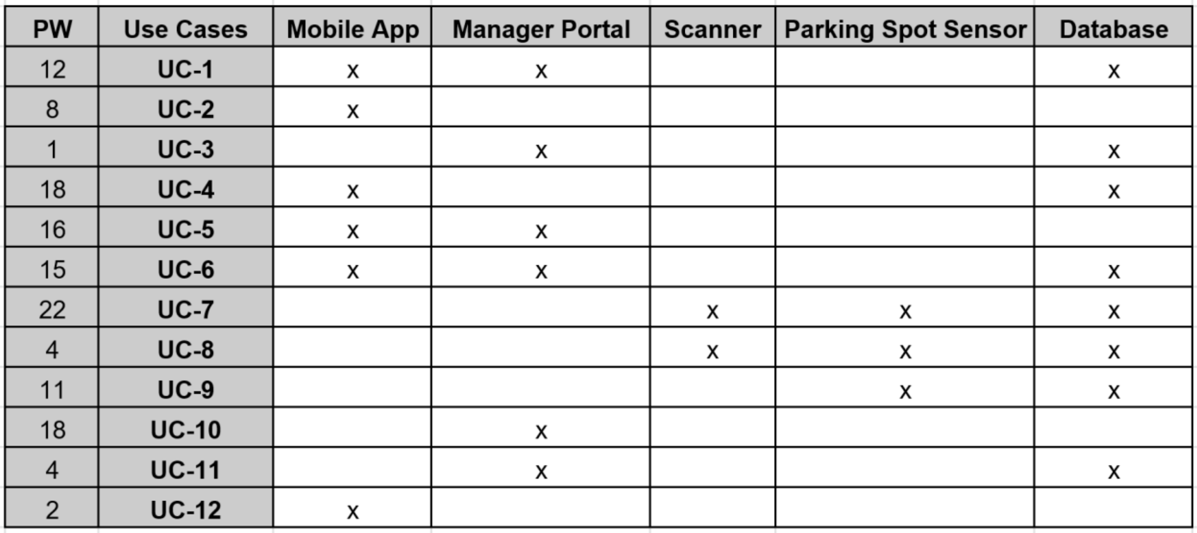
Concept	**Attributes**	**Attribute Description**
Database	Customer information table	A data structure that stores the customer's information such as name, license plate number, demographics, etc.
	Parking spots table	A data structure that stores the current information on the parking spots, to determine their occupancy status.
	Reservations table	A data structure that stores the information on upcoming and current reservations, including the customer's license plate number, and time of reservation.
	Walk-Ins table	A data structure that stores information on walk-in customers, such as the time the customer enters and leaves the garage.
	Records table	A data structure that stores the information on all reservations, existing and past, including whether the reservation was deleted or not.
Manager Portal	Login	The manager's username and password to verify their identity and access the portal.

	Available spots	The number of available parking spots in the garage, used to determine availability for walk-in customers
	Current reservations table view	A table that displays current reservations. This includes start and end times of each individual reservation and the customer they are under.
	Statistics table view	A table that displays garage statistics, including average reservation times and average reservation lengths.
	Customer table view	A table that displays the number of customers daily as well as individual customer information such as time of arrival, time of departure, and license plate numbers.
	Records table view	A table that displays past records. This includes past reservation dates and times, payment information (credit card information, name)
Mobile App - customer information	Login	The customer's username and password to verify their identity and access the profile.
	Name	The customer's first and last name for contact information.
	Email	The customer's email address for contact information.
	Phone number	The customer's phone number for contact information.
	Barcode	The customer's assigned barcode to identify the account.
	Registration number	The customer's assigned registration number to identify the account.
	License number	The customer's registered car's license number.
	Credit card	The customer's credit card information, including name, billing address, and CVV.
	Number of points	The customer's number of points collected.
	Date	The date of reservation.

Mobile app - reservation	Start time, end time	The start and end times of reservation, also used to determine payment charge by dynamic pricing model.
	Assigned spot	The parking spot number assigned to reservation.
	Payment charge	The dollar amount of cost of reservation.
Parking Spot Sensor	Empty	This detects if the parking spot is empty or filled (Boolean).
Scanner	Barcode	The barcode that the scanner reads and recognizes.

Traceability Matrix

This matrix maps the use cases to the domain concepts. The total priority weight (PW) of each requirement was calculated using the requirements-to-use-case traceability matrix. A higher number indicates a higher priority.



PW	Use Cases	Mobile App	Manager Portal	Scanner	Parking Spot Sensor	Database
12	UC-1	x	x			x
8	UC-2	x				
1	UC-3		x			x
18	UC-4	x				x
16	UC-5	x	x			
15	UC-6	x	x			x
22	UC-7			x	x	x
4	UC-8			x	x	x
11	UC-9				x	x
18	UC-10		x			
4	UC-11		x			x
2	UC-12	x				

]

System Operation Contracts

Operation	UC-4 Create Reservation
Preconditions	• The customer has an existing profile stored in the database • The customer has logged into the ParkingGarageApp • The ParkingGarageApp displays a calendar for the current month as well as available time slots
Postconditions	• The system reduced the number of available spots for the reserved time • The new reservation is tied to the user's account reservations

Operation	UC-5 Edit Reservation
Preconditions	• At least one reservation for the customer of interest must exist in the system
Postconditions	• An edited version of the reservation will take the place of the original in the database • The original reservation is wiped from the database

Operation	UC-7 Enter Garage
Preconditions	• The customer has yet to enter the garage
Postconditions	• The customer is directed either to the appropriate parking spot or is asked to back out of the elevator

Operation	UC-10 Display Statistics
Preconditions	• The Garage Manager has access to an authorized account • The InformationDatabase has existing records to display
Postconditions	• The ParkingGarageWebsite will display graphs and tables constructed from customer parking data

Mathematical Model

Dynamic Pricing Model

Research and calculations done by Fall 2018 Group 3 have determined that parking garages reach peak usage from 7:00 a.m. to 1:00 p.m [2]. This conclusion is made under the assumption that a single customer reserves only one spot during a span of time, disregarding group reservations [2]. This mathematics of this model is based off algorithms from the paper, "Dynamic Pricing for Reservation-Based Parking System [5]", and are modeled to work in a simple case. The model also accounts for customers that act with conditions described in the paper, "Optimal Dynamic Pricing of Inventories with Stochastic Demand Over Finite Horizons [4]." Their purpose is to find available parking rather than find the cheapest parking. However, this model considers only a single demand function, and would not be valid for special cases such as events that bring more drivers into the area.

Thus, we divide the parking into three intervals: regular-priced parking, surge-priced parking, and discounted parking. During hours where there is no peak or lull in parking garage usage, the price will be set at an arbitrary dollar amount per hour *x*, to be set by the manager of the garage. During peak hours when the demand for the parking garage spots is greater than its supply, then the price for this interval will be *x* multiplied by a factor of 1.6. This is based off of cost analysis from a previous study which states that an increase of less than a factor of 0.5 would not decrease the demand enough while a factor of 0.8 would cause there to be too much decreased demand [2]. A multiplier of 0.7 will be placed on *x* to calculate the price for discounted parking, which is applied during times when the parking garage does not expect as many customers in order to attract more customers. This factor was determined based on calculations that state a decrease by a factor of 0.5 would increase demand too greatly [2].

Considering data collected from the previous study [2], we set surge-priced parking to be from 7:00 a.m. to 1:00 p.m. Regular-priced parking will be from 12:00 a.m. to 3:00 a.m. and 4:00 p.m. to 12:00 a.m. Parking is charged at a discounted rate from 3:00 a.m. to 7:00 a.m. and 1:00 p.m. to 4:00 p.m.

Therefore, the parking price schedule, as determined, is follows (*x* being a fixed price set by the manager):

Start Time	End Time	Price ($/hr)
12:00 a.m.	3:00 a.m.	*x*
3:00 a.m.	7:00 a.m.	0.7*x*
7:00 a.m.	1:00 p.m.	1.6*x*
1:00 p.m.	4:00 p.m.	0.7*x*
4:00 p.m.	12:00 a.m.	*x*

Reward/Penalty System Model

This project utilizes a simple model for distributing and deducting points as a part of our reward and penalty system. The following is a table that dictates guidelines for rewarding, deducting, and exchanging points. Outlined are the action, points exchanged, and description of the action. The action depicts whether the points received were for a reward, penalty, rectification, or exchange. An addition symbol in front of the point value signifies that points are added to the registered account while the subtraction symbol signifies that points are deducted. The point values were arbitrarily decided based on the weight of each action; hence, showing up on time to a reservation gains a customer five points but having a reoccurring reservation would yield more points (15). The description of each action describes the conditions for the exchange of points.

Action	Points	Description
Reward	+10	Customer makes a reservation
Reward	+15	Customer makes a reoccurring reservation (for every 30 reservations)
Reward	+5	Customer shows up to reservation within 5 minutes of their start time
Penalty	-5	Customer stays longer than their reserved time
Rectification	-10	Customer cancels a reservation (to reconcile for the 10 points gained from making a reservation)
Exchange	-100	Credit to make a reservation for a 5-hour period
Exchange	-100	Credit to receive a preferred parking spot (closer to the entrance/exit)

Project Size Estimation

Unadjusted Actor Weight (UAW)

The complexity level was assigned based on the intricacy of the associated interface. The weight level was determined by the complexity, with higher values assigned to more complex levels.

Actor Name	Description of Relevant Characteristics	Complexity	Weight
Customer	The customer can interact with the system through a graphical user interface to manage their account and reserve parking spots.	Complex	3
Garage Manager	The garage manager can interact with the system through a graphical user interface to manage the garage as an authorized user.	Complex	3
Mobile App	Customers interact with the mobile app through a graphical user interface on an application downloaded to their mobile devices	Complex	3
Manager Portal	The garage manager can interact with the manager portal to edit customer accounts and adjust garage settings through text-based inputs.	Average	2
Database	Contains and updates customer information. Interacts with mobile app, manager portal, and scanner through text inputs.	Average	2
Scanner	Customers can interact with the scanner by having their QR code scanned upon entry. The scanner also records data in the information database.	Average	2
Elevator	The elevator interacts with the information database to lift customers to their designated parking level.	Simple	1

UAW = (1*SIMPLE) + (2*AVERAGE) + (3*COMPLEX)

= (1*1) + (2*3) + (3*3)

UAW = 16

Unadjusted Use Case Weight (UUCW)

The UUCW was calculated for the highest-priority use cases (fully-dressed). The category level is based on the number of steps in the main success scenario. A higher weight was assigned to higher category levels.

Use Case	**Description of Relevant Characteristics**	**Category**	**Weight**
Create Reservation	Average user interface (graphical). Three participating actors (Customer, Mobile App, Information Database). 5 steps for main success scenario.	Average	10
Edit Reservation	Complex user interface (graphical). Three participating actors (Customer, Mobile App, Information Database). 7 steps for main success scenario.	Complex	15
Enter Garage	Average user interface (graphical). Four participating actors (Customer, Scanner, Elevator, Information Database). 6 steps for main success scenario.	Average	10
Display Stats	Simple user interface (graphical). Three participating actors (Customer, Mobile App, Information Database). 4 steps for main success scenario.	Simple	5

UUCW = (5*SIMPLE) + (10*AVERAGE) + (15*COMPLEX)

= (5*1) + (10*2) + (15*1)

UUCW = 40

Technical Complexity Factor (TCF)

Technical factors were identified to estimate the impact on productivity by each requirement of the project. Each factor is assigned a complexity rating based on the perception of effort required to satisfy the requirement. Greater values correlate to higher perceived complexity levels. The weight is determined by the relative impact, with higher values assigned to more impactful items. The calculated factor is produced by multiplying the perceived complexity and weight.

Technical Factor	Description	Perceived Complexity	Weight	Calculated Factor
T1	Distributed system. Website and mobile app need to communicate with the database	3	2	6
T2	Customers expect a fast-performing parking system with minimal interaction.	4	1	4
T3	Customers and managers expect low response times and system stability.	4	1	4
T4	Dynamic pricing model will require moderate calculation. The system will use a simple formula.	2	1	2
T5	Design is reusable and can be implemented in other garages with similar structure.	3	1	3
T6	Installation is complex and would require installation of various software and hardware modules such as cameras and scanners at parking areas.	4	0.5	2
T7	Ease of use is extremely important. The reservation process should be simple and can be completed with minimal interaction.	5	0.5	2.5
T8	Not portable. Hardware systems and databases would be difficult to migrate.	0	2	0
T9	Dynamic pricing model will be easy to modify by the garage managers.	2	1	2
T10	The system should support many concurrent users both using the app and reserving spots	4	1	4
T11	The system will encrypt personal and transaction information.	2	1	2

T12	Third-party companies will not be permitted to use the system.	0	1	0
T13	No special training needed, system is automated and simple to use	0	1	0

TCF = 0.6 + 0.01*TOTAL_CALCULATED_FACTOR

= 0.6 + 0.01*31.5

TCF = 0.915

<u>Unadjusted Use Case Points (UUCP)</u>

UUCP = UAW + UUCW

= 16 + 40

UUCP = 56

<u>Environmental Complexity Factors (ECF)</u>

We assume that the environmental conditions are similar for all students and have been instructed to use ECF = 1.

ECF = 1

<u>Use Case Points (UCP)</u>

UCP = UUCP * TCF * ECF

= 56 * 0.915 * 1

UCP = 51.24

PROJECT MANAGEMENT

Plan of Work

After the first Software Engineering lecture, our roster was finalized and sent to Professor Marsic and Kartik Rattan. We agreed to appoint Samantha Moy as the project leader and Atmika Ponnusamy as the configuration manager. Mondays at 10:00am are reserved for full team meetings, scheduled as necessary. Sub-group meeting schedules are decided among members of each group. On Sundays before full-team meetings, Samantha M. will send a meeting agenda to the team. She records meeting notes on the agenda while leading the meeting. Members give updates on progress, discuss future work, and assign responsibilities. Our main form of communication is via Facebook Messenger. Samantha M. sends full-team reminders about meetings and deadlines through Messenger. We are beginning to transition to Slack to simplify communication as separate work on sub-projects begins.

Throughout the course of this semester, our team aims to collaborate efficiently by utilizing file sharing resources that are available to us. These resources, which include Google Drive and GitHub, will enable us to operate as one unit, while still allowing each member to make their own individual contributions to the project materials.

A Google Drive folder of resources will be updated throughout the semester. The folder includes contact information, calendars with deadlines, project resources, meeting agendas, and meeting minutes. This folder will serve as the primary tool for project management and will be overseen by Samantha M.

Team members who possess expertise regarding GitHub will serve as configuration managers. Each sub-group will appoint a configuration manager for their sub-project, and Atmika will serve as the overall configuration manager. These individuals will be responsible for ensuring that all code submitted to the project's central repository is cohesive and well written, both syntactically and semantically. Each team member will possess a local working directory, as well as their own public repository from which the code will enter a review process: the configuration manager will review the code, and if they decide it is acceptable it will be committed to the project's central repository. Otherwise, the code will be rejected with feedback in order to be improved. See the section, "Project Ownership," for specific information about the division of work.

Product Ownership

Together, our team agreed on the overall project business plan and shared infrastructure. Samantha M. and Atmika have researched and setup the shared infrastructure to lay a strong foundation for the team's sub-projects. Our team of nine members have been divided into three groups of three. Each group is mainly responsible for a single mini project with specific functional features and qualitative properties, listed below. Each group sub-topic encompasses several requirements from the requirements tables above. Of course, there may be inter-group collaboration if a member's specific expertise is ever required. Some members may also take on additional tasks as needed (see below).

Shared Infrastructure – Samantha Moy, Atmika Ponnusamy

- Rutgers ECS server
- MySQL database with 5 tables
    - Customer Info
    - Parking Spots
    - Reservations
    - Walk-Ins
    - Records
- 3 sub-projects will interact with the database (see the following)

Sub-Groups

1. **Customer Services** – Samantha Moy, Atmika Ponnusamy, Shreya Patel
   Technologies Used: Android Studio
    a. Create and edit a customer profile via the app.
    b. Create, edit, and cancel a customer reservation via the app.
    c. Reserve handicap spots after verifying their government-issued handicap ID.
    d. Reward points for making reservations.
2. **Arrival & Departure Operations** – Andrew Ko, Parth Patel, Piotr Zakrevski
   Technologies Used: JavaFX, Scene Builder, JDBC
    a. Handle walk-in cars.
    b. Update the vacancy status of spots as cars arrive/depart (on-time, early, or late).

    c. Allow rental cars to enter after scanning QR code to confirm their identity.
    d. Illegal usage
        i. Deduct points for staying past reservation
        ii. Redirecting customers whose spot is illegally occupied.

3. **Manager Tasks** – Kylie Chow, Samantha Cheng, Nandita Shenoy

   Technologies Used: Visual Studio, MEAN Stack

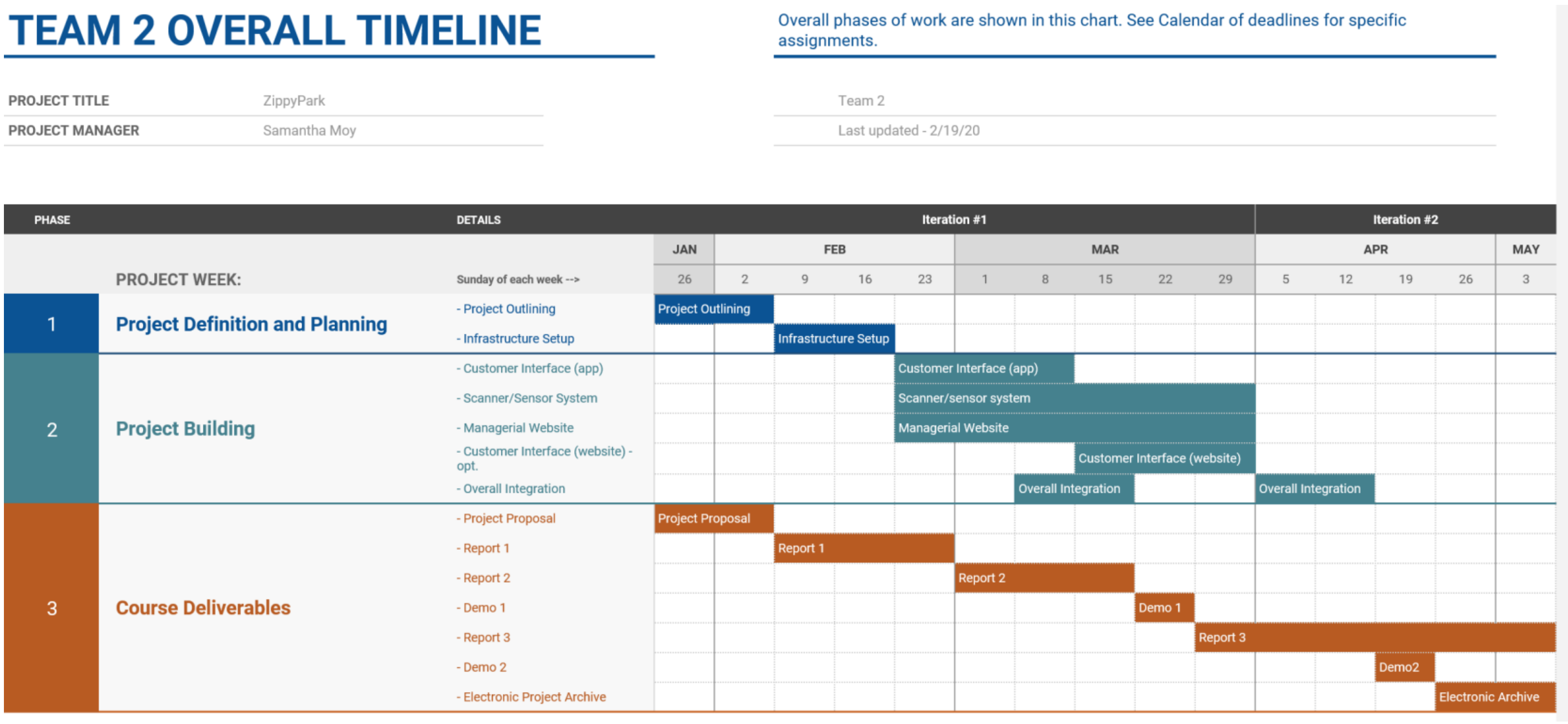
    a. Allow the manager to view all garage reservations through a special website.
    b. Allow the manager to adjust surge-pricing scheme through a special website.
    c. Allow the manager to adjust the rewards/penalty system through a special website.
    d. Allow the manager to view garage statistics.

Additional Tasks

1. Project Management – Samantha Moy
    a. Manage group deadlines and weekly tasks (set by professor and by the group).
    b. Update group calendars and documents.
    c. Run team meetings and write agendas.
    d. Assemble and submit weekly reports/deliverables.
2. Configuration Management – Atmika Ponnusamy
    a. Update project website.
    b. Manage GitHub account.
    c. Maintain good project practices when integrating group deliverables.
    d. Educate team on best coding practices and project infrastructure.

Timeline

Located in our shared group resources folder, we have a Gantt Chart that shows the high-level overview of this project. This spreadsheet is maintained by Samantha M.



TEAM 2 OVERALL TIMELINE

Overall phases of work are shown in this chart. See Calendar of deadlines for specific assignments.

PROJECT TITLE	ZippyPark
PROJECT MANAGER	Samantha Moy

Team 2

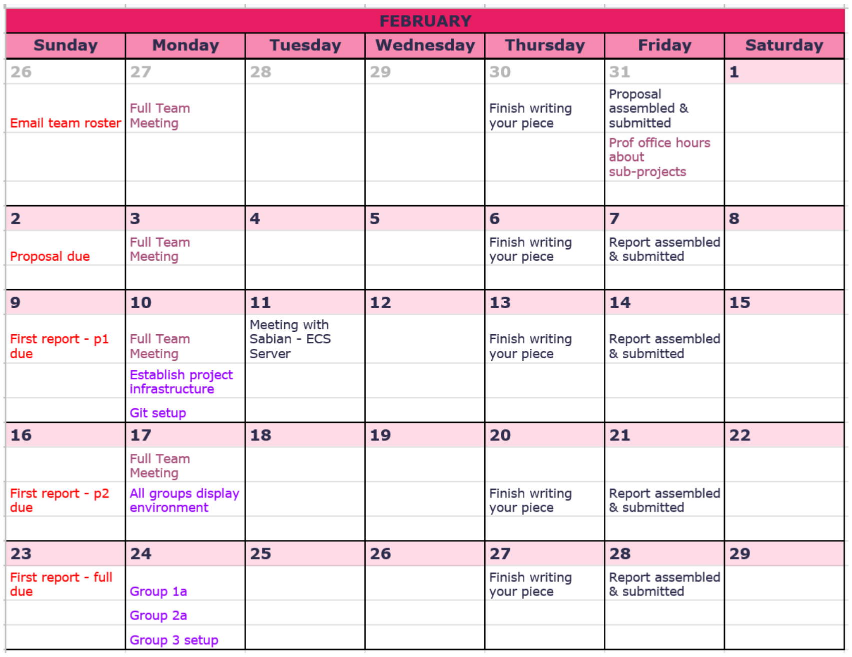
Last updated - 2/19/20

PHASE		DETAILS	Iteration #1										Iteration #2				
			JAN	FEB				MAR					APR				MAY
	PROJECT WEEK:	Sunday of each week -->	26	2	9	16	23	1	8	15	22	29	5	12	19	26	3
1	Project Definition and Planning	- Project Outlining	Project Outlining														
		- Infrastructure Setup			Infrastructure Setup												
2	Project Building	- Customer Interface (app)					Customer Interface (app)										
		- Scanner/Sensor System					Scanner/sensor system										
		- Managerial Website					Managerial Website										
		- Customer Interface (website) - opt.								Customer Interface (website)							
		- Overall Integration							Overall Integration				Overall Integration				
3	Course Deliverables	- Project Proposal	Project Proposal														
		- Report 1			Report 1												
		- Report 2						Report 2									
		- Demo 1									Demo 1						
		- Report 3										Report 3					
		- Demo 2													Demo2		
		- Electronic Project Archive														Electronic Archive	

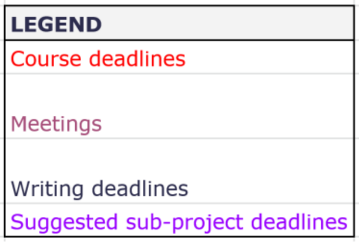
The same spreadsheet may also be viewed here: https://tinyurl.com/team2Gantt

Additionally, we have a calendar of deadlines that is maintained by Samantha M. This spreadsheet serves as a simple way to view more specific project deadlines and milestones.

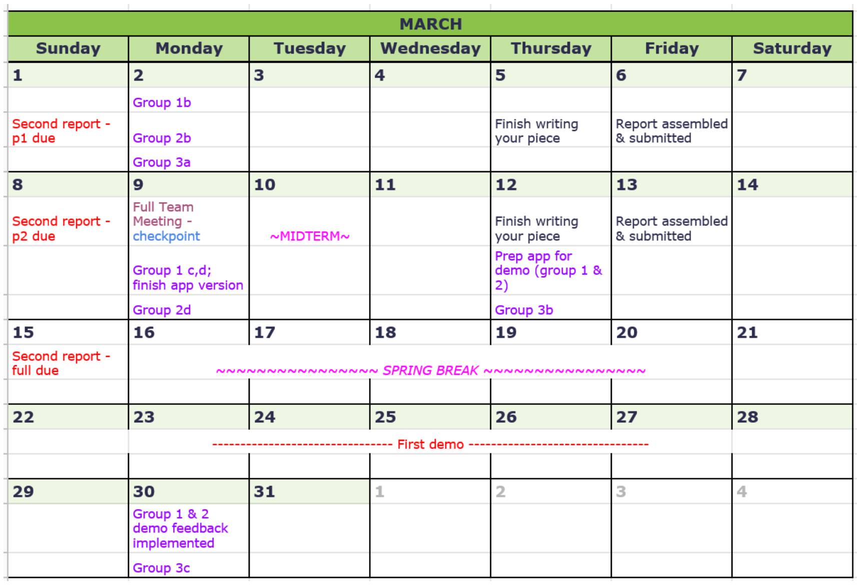
Note: "Group 2a" refers to the suggested deadline of subtask "a" of Group 2's work. However, sub-group deadlines (purple) are merely suggestions; sub-groups may set their own deadlines. All other dates (red, maroon, black) must be followed.



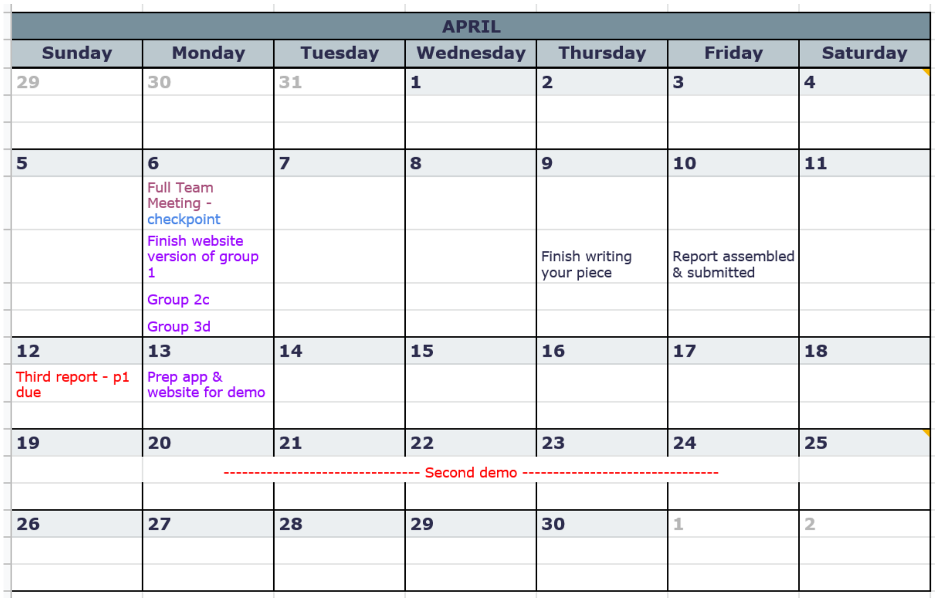
FEBRUARY						
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
26	27	28	29	30	31	1
Email team roster	Full Team Meeting			Finish writing your piece	Proposal assembled & submitted Prof office hours about sub-projects	
2	3	4	5	6	7	8
Proposal due	Full Team Meeting			Finish writing your piece	Report assembled & submitted	
9	10	11	12	13	14	15
First report - p1 due	Full Team Meeting Establish project infrastructure Git setup	Meeting with Sabian - ECS Server		Finish writing your piece	Report assembled & submitted	
16	17	18	19	20	21	22
First report - p2 due	Full Team Meeting All groups display environment			Finish writing your piece	Report assembled & submitted	
23	24	25	26	27	28	29
First report - full due	Group 1a Group 2a Group 3 setup			Finish writing your piece	Report assembled & submitted	



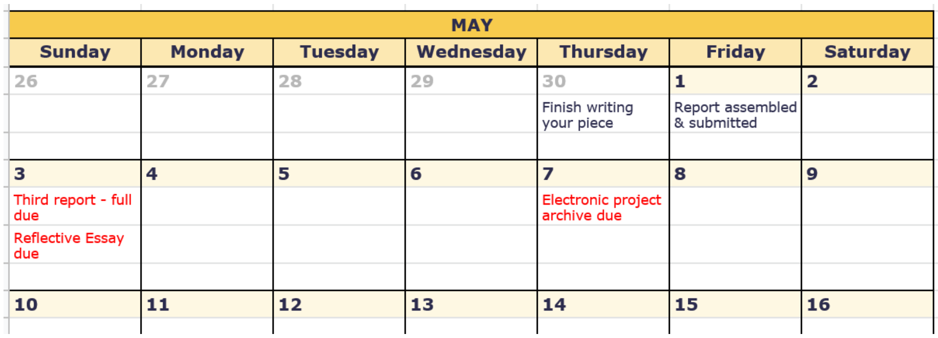
LEGEND
Course deadlines
Meetings
Writing deadlines
Suggested sub-project deadlines



MARCH						
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	2	3	4	5	6	7
Second report - p1 due	Group 1b Group 2b Group 3a			Finish writing your piece	Report assembled & submitted	
8	9	10	11	12	13	14
Second report - p2 due	Full Team Meeting - checkpoint Group 1 c,d; finish app version Group 2d	~MIDTERM~		Finish writing your piece Prep app for demo (group 1 & 2) Group 3b	Report assembled & submitted	
15	16	17	18	19	20	21
Second report - full due	~~~~~~~~~~~~~~~~ SPRING BREAK ~~~~~~~~~~~~~~~~					
22	23	24	25	26	27	28
	------------------------------- First demo -------------------------------					
29	30	31	1	2	3	4
	Group 1 & 2 demo feedback implemented Group 3c					



APRIL						
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
29	30	31	1	2	3	4
5	6 Full Team Meeting - checkpoint Finish website version of group 1 Group 2c Group 3d	7	8	9 Finish writing your piece	10 Report assembled & submitted	11
12 Third report - p1 due	13 Prep app & website for demo	14	15	16	17	18
19	20	21	22 ------------------------------- Second demo -------------------------------	23	24	25
26	27	28	29	30	1	2



MAY						
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
26	27	28	29	30 Finish writing your piece	1 Report assembled & submitted	2
3 Third report - full due Reflective Essay due	4	5	6	7 Electronic project archive due	8	9
10	11	12	13	14	15	16

The same spreadsheet may also be viewed here: https://tinyurl.com/team2calendar.

Future Work

Our team has held a full-team meeting every Monday since January 27th, for a total of four meetings. Over the last few weeks, we have thoroughly outlined our project, sub-teams, responsibilities, and timeline. We have also designed and implemented our shared infrastructure and team communication systems. As we transition into sub-team work, we will no longer meet weekly as a full-team. However, we will gather on pre-scheduled days for full-team check-ins, as well as when the need arises. We will also be in contact weekly when assembling the report portion that is due each week.

Initially, our team was apprehensive given our overall lack of software engineering experience. However, our team leaders have met with professors, advisors, and the Rutgers Engineer Computing Services (ECS) to gain insight into a project of this magnitude. The team has worked hard over the last few weeks to use this advice to establish a strong foundation for the project; we have a clear business plan, basic software infrastructure, and knowledge of the technologies required. We look forward to beginning sub-project work (divisions outlined in the previous section) and meeting again at our first checkpoint on March 9th.

REFERENCES

[1] Inrix. (2017, July 12). *Searching for Parking Costs Americans $73 Billion a Year*. Inrix Research. https://inrix.com/press-releases/parking-pain-us/

[2] Chen, C., Deng, C., Feng, X., Liao, S., Musale, S., Sakhuja., R. (2018, December 9). *Auto Park Parking Garage Automation*. Software Engineering Project: Parking Garage/Lot Automation. https://www.ece.rutgers.edu/~marsic/books/SE/projects/ParkingLot/2018f-g3-report3.pdf

[3] Choudhury, S., Ngo, T., Nguyen, D., Nguyen, K., Patel, N., Tran, L. (2019, December). *Blockchain and Docker Assisted Secure Automated Parking Garage System.* Software Engineering Project: Parking Garage/Lot Automation. https://www.ece.rutgers.edu/~marsic/books/SE/projects/ParkingLot/2019f-g4-report3.pdf

[4] Gallego, Guillermo, and Garrett Van Ryzin. "Optimal dynamic pricing of inventories with stochastic demand over finite horizons." *Management science* 40.8 (1994): 999-1020.

[5] Tian, Qiong, et al. "Dynamic pricing for reservation-based parking system: A revenue management method." *Transport Policy* 71 (2018): 36-44.