Parking Garage/Lot Automation

Software Engineering Second Report

Group 14

Tingcong Jiang, Chenyu Cao, Shijie Xu, Yiran Tan, Zhuohuan Li, Buyuan Lin, Samuel Cho, Christopher Cheng

https://cc1539.github.io/SE-project/

May 5th, 2019

Contribution Matrix

All Equal

	Tingcong Jiang	Chenyu Cao	Shijie Xu	Yiran Tan	Zhuohuan Li	Buyan Lin	Samuel Cho	Christopher Cheng
Section Project Management	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 1	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 2	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 3	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 4	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 5	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 6	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 7	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 8	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 9	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 10	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 11	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 12	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%

Section 13	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%
Section 14	12.5%	12.5%	12.5 %	12.5 %	12.5%	12.5%	12.5%	12.5%

Table of Contents

Summary of Changes

Section 1: Deleted description related with LED hardware. Added discussion of existing parking design.

Section 2: Added

Section 3: Revised UC 6,7,14,20,21

Section 4: Removed stakeholders from project; Removed emergency alert; Removed notification; Removed Ad-hoc parking & payment; Adjustment to Entering The Garage

Section 5: Use case updated

Section 6: Domain Model is in the report

Section 7: Changed diagram of Use Case 3

Section 8: OCL style added

Section 9: Camera spec added; Sensor spec added; Motor spec added; Platform spec added

Section 10: No changes

Section 11: Added worst case scenario efforts from the user and the picture is from our final UI design which we demonstrated in DEMO 2.

Section 12: Updated tests

Section 13: Updated milestone and updated future plan before the next demo.

Section 14: Added car recognition API.

Section 1: Customer Problem Statement

As consumers, we face a major problem when dealing with parking garages today — one that stems from a lack of modern management strategies regarding parking lots in decently populated areas. In another word, it is not common to see such a parking garage that integrating reservation, convenient entering and payment together. Our society's current, deprecated system provides only an inefficient and inelegant solution to the high-density, simultaneous parking of many vehicles in a single parking garage. Thanks to the absence of proper and effective communication between the parking garage owners and us customers, these ineffective methods are not ameliorated or improved upon, but instead continue to proliferate. For commuters like us and those across the globe, who are plagued by the constant concern of successfully and quickly finding a parking space in busy metropolitan areas, the ultimate effect is a significant amount of wasted time and effort. The current design has many drawbacks such as lack of the integration of convenient parking and smart management. Many existing convenient parking garage needs many manual work to manage. Also, many existing design with very easy payment and smart management cost customers' time. Therefore, it is very meaningful to design a garage which can automatically record users information and spots status and also saves both customers and managers' time at the same time.

As the global population skyrockets (Roser and Ortiz-Ospina, 2019) and we move into a fast-paced age of new and improved transportation technology, it is disappointing, to say the least, to see how little progress has been made in the area of parking lot management to accommodate the increasing popularity of owning a private vehicle (Hedges & Company, 2019). The issue is only exacerbated by establishments that believe they can address the problem by simply making the parking lot bigger (Rasmussen, 2016), when a more modern solution would be to employ novel techniques such as giving indication to customers about the availability of space and decreasing wait times by eliminating cumbersome payment options. As commuter students, our team members must find a parking spot on campus every working day of the week, and therefore encounter this problem almost every day.

Although there are couple of huge parking lots located in different campus, we have to spend a lot of time driving through all the rows in the parking lot to find a vacant parking space. Sometimes, to save the time finding a vacant parking space, well park directly at a spot that is far away from the campus. However, we have to walk a lot and we might even miss those spots that's closer to the campus.

What if we already know the location of the vacant parking spot when we enter the parking lot? This will save the overall time for all the commuters. The problem described above is the reason we chose this project. The potential treatment is included as well. Besides the parking spot finding function, we are also interested in implementing an automatic payment function for those commercial parking lots.

As customers, we would suggest implementing these four paradigms: quantification, reservation, automation, and virtualization. If properly and intelligently implemented using modern technologies, each of these should be more than able to alleviate one or more of the aforementioned issues.

The first paradigm, "quantification", encompasses the idea of managing a garage by separating it into different sections and subsections and "quantifying" the information that describes each section in an intuitive, easy-to-visualize way. One can then view this information in real-time in the form of an online parking garage map. Utilizing this, customers and administrators alike can quickly gauge the state of the lot and make informed decisions. Additional features, such as the ability to sort all cars by their information (e.g. entry time, exit time, car make, car model, and car plate) can be implemented to facilitate ergonomics for the management team.

The second paradigm, "reservation", encompasses the idea of time management. Through reservation, customers who have a plan and know ahead of time when and how long they will leave their car in a parking lot, like us commuter students who have a well-defined class schedule, are given the option to take the initiative and apply for reservations. To simplify operation of the lot, the system should be implemented so that what customers can reserve is not a specific spot, but *a* spot. Of course, if this idea applied uniformly to the entire lot, the problem of our parking space being far from your destination would still remain. Therefore, we re-apply the concept of "quantification" and assert these rules within particular sections. On the same vein, the owner should be given the option to designate a reservable section and a "normal", un-reservable section to alleviate the effects of overbooking.

The third paradigm, "automation", encompasses the idea of lessening the burden of the customer. For customers like us, who will consistently visit the same lot, something like an account system would be of great benefit. After the initial account creation and setup, any and all payment should occur automatically, with no further intervention by the customer or owner. This would put an end to the hassle of paying at a machine or booth, which can easily become a source of congestion when there is an influx of customers. To promote the usage of this account system, clients will be prompted to create an account with a QR code as they enter the gate, or perhaps, as they walk into the establishment. This account will be tied to the customer's license plate and bank account/credit card. Through this account, the customer can be systematically billed for the correct amount according to the length of their accommodation when they exit. As a result, a cashier is no longer required, and this particular cost of labor can be eliminated.

The fourth paradigm, "virtualization", encompasses the idea of synchronicity between the physical and the conceptual. Using the online interface described earlier, customers can remotely view the data displayed by the websites which will display the map of the parking garage and status of the spots. Based on the map, the user will be able to locate their spot when they return. As a customer, I know that more often than not, I don't immediately know exactly which row and which spot in that row that I park in. Therefore, only spots that are deemed occupied by sensors should be able to be marked, so as to combat human error and prevent erroneous entries to the system. Optionally, customers may save information about their car make and model to their account for further identification purposes. On the administrative side, the backend management software that can help the owner maintain and oversee the garage should be included. Through this interface, various values and properties, such as lot pricing, can be viewed and adjusted on the fly and perhaps according to an algorithm of choice.

Section 2: Glossary of Terms

- Account - Can be created by any customer. The account will have all the useful information the customer entered and enable more functionalities, such as checking their location of the car.
- Backend Management Software - A software designed for the garage owner (or the manager). It allows the manager to monitor the garage and functions such as changing the price. The software is connected to the database so any change in the database will be reflected on this software.
- Database - All the data are stored in the database and encrypted for security concern.
- Gate - A system that's been installed at both the entrance and the exit of the garate. The gate consist of IR sensor, a License Plate Scanner and a pole that will open when the scanner done scanning the car plate.
- IR Sensor - Infrared Radiation Sensor. We use that to monitor if a spot is occupied or not. And also at the entrance and exit, we use the IR sensor to detect if there is a car approaching the gate.
- License Plate Scanner - A camera that's connected with the database and website. It will capture the car's license plate and send the information to the database.
- Occupied Spot - A parking spot that's currently been used. It will be reflected on the website so that it's not available to reserve.
- Parking Spot Number - Each spot in the parking is assigned with a unique number, this is easier for the backend to monitor the whole status of the parking garage.
- QR Code - A unique QR code displayed at the entrance. Every customer can scan that using their phone. It will lead to a registration page. However, they are not required to register.
- Registration - The process for a customer to have an account.
- Reservation - A function that will allow the registered user to reserve a parking spot in the garage in a certain period of time.

- Vacant Spot - A parking spot that has not been taken and is available for reservation.
- Website - Our website has the portals that leads to registration page, the garage map and the parking status checking page for those registered customers.

Section 3: System Requirements

Functional Features

Identifier	**Priority**	**Requirement**
REQ-1	10	The system shall allow a user account to make a reservation or cancel a reservation.
REQ-2	8	The system shall send text or email confirmation for either reservation success or failure
REQ-3	10	The system shall read a car's license plate when it approaches the entrance gate
REQ-4	10	The system shall recognize existing accounts from previously scanned license plates
REQ-5	8	The system shall provide payment methods for walk-in customers
REQ-6	9	The system shall mark the time a customer entered the lot with no time limits
[redacted]	[redacted]	[redacted]
REQ-8	6	The system shall display on a map which spaces are occupied or not
REQ-9	4	The system shall show customers the spot they occupied.
REQ-10	5	The system shall guide users to special spots for their special needs (electric car, disabled parking)

REQ-11	7	The system shall update the map when a car leaves or fills a spot
REQ-12	6	The system shall mark the time when a customer leaves the lot
REQ-13	4	The system shall notify a customer who has been parked for too long
REQ-14	10	The system shall automatically deduct the parking fee from the customer's account
REQ-15	9	The system shall allow the lot owner to view the information in the database
REQ-16	2	The system shall allow the lot owner to edit the information in the database
REQ-17	3	The system shall allow the lot owner to set the maximum number of spots that are allowed to be reserved
REQ-18	6	The system shall allow the lot owner to change the price of parking at the lot
[redacted]	[redacted]	[redacted]
REQ-20	3	The system should offer special needs when registered users make reservations. Disabled parking, spot with charging pile for electric car owners, etc...
REQ-21	5	The system should let the user to cancel their reservation if they do not want to park here anymore or they want a different type of parking spots.
REQ-22	6	The system shall allow customers to view and edit their own information
REQ-23	7	The system should be able to send a message or email to the desired phone number or email address
REQ-24	7	The system should be able to detect any unexpected situation

REQ-25	7	The system should be able to follow a predetermined solution to solve any unexpected situation
REQ-26	10	The system shall allow a user to create an account on the website.
REQ-27	10	The system shall save the user's information in the database.

Non-Functional Features

Identifier	**Priority**	**Requirement**
REQ-28	10	The system shall have access to the internet
REQ-29	8	The system shall have access to one or more SIM cards to facilitate automatic texts
REQ-30	10	The system shall include two or more cameras capable of clearly capturing the license plate of a car moving at moderate speed
REQ-31	10	The system shall store database information in the form of a filesystem.
REQ-32	8	The system shall be capable of interfacing with an optional, manual payment booth
[redacted]	[redacted]	[redacted]
REQ-34	6	The system shall translate information about the state of the lot into a visual form

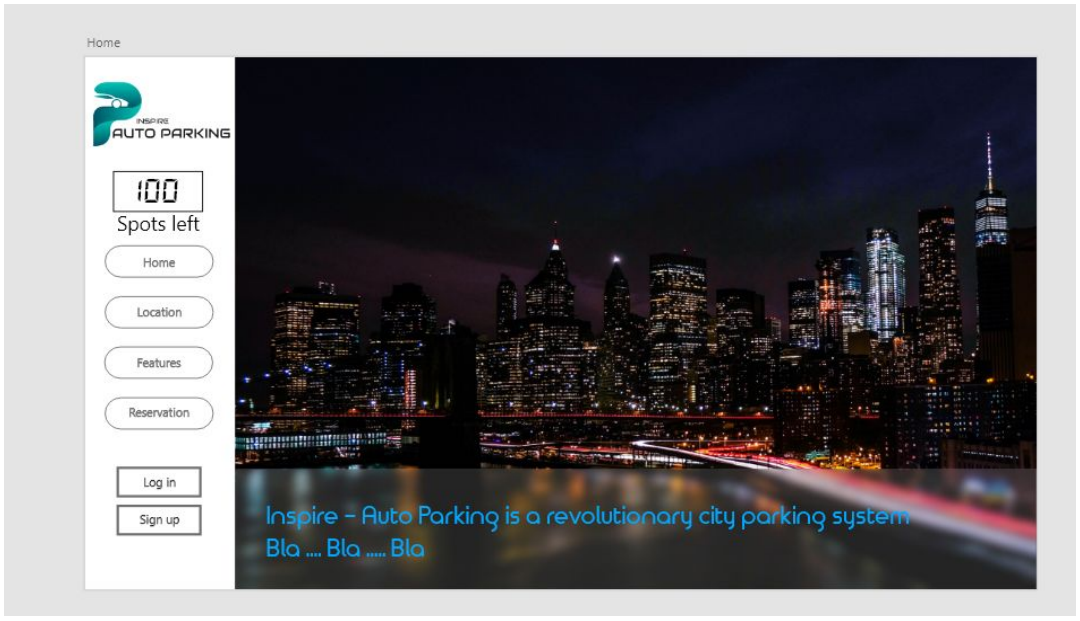
REQ-35	5	The system shall be able to identify special-needs customers through information on the account
REQ-36	7	The system shall interface with proximity sensors.
REQ-37	4	The system shall allow an administrator to configure its behavior
REQ-38	10	The system shall be able to automatically make transactions online.
REQ-39	5	The system shall be able to differentiate between a registered car and an unregistered one.

On-Screen Requirements

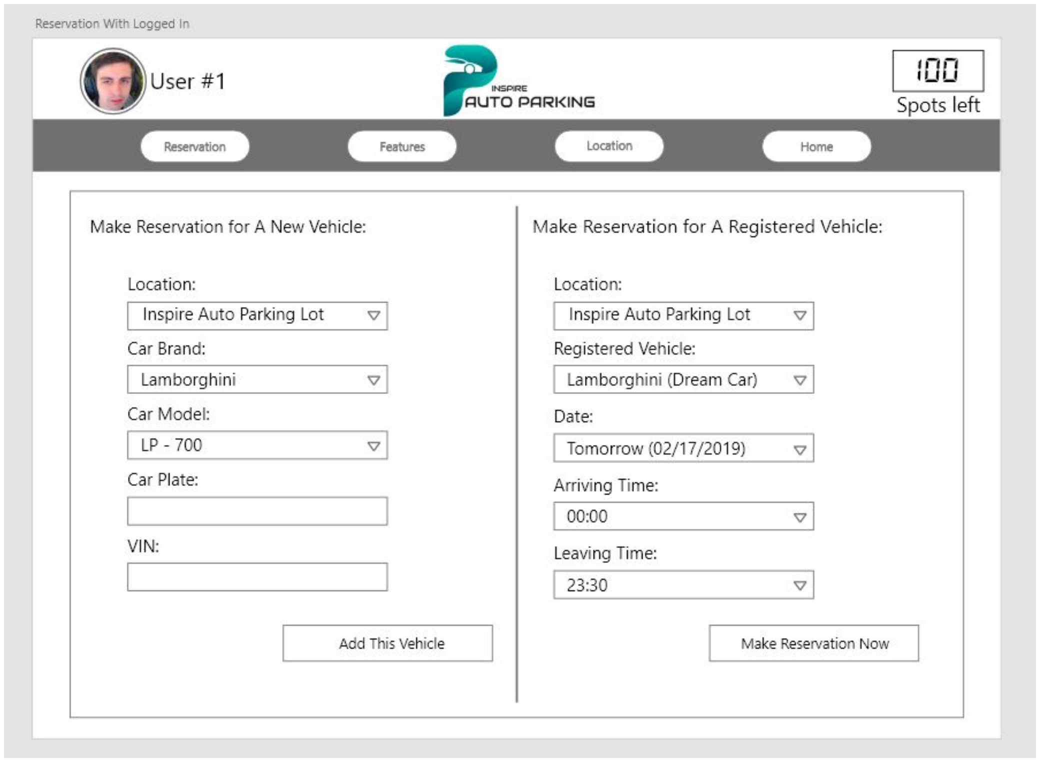
Identifier	**Priority**	**Description**
REQ-40	9	On homepage, allow unregistered users to sign up with clicking sign up button and inputting username and password
REQ-41	9	On homepage, allow registered users to login with clicking login button and inputting their username and password
REQ-42	6	Show users the number of available spots
REQ-43	5	Allow users clicking button to see the garage location
REQ-44	2	Allow users clicking button to see the unique features of garage
REQ-45	6	Allow users clicking button to make a reservation
REQ-46	4	Allow users clicking button to access homepage
REQ-47	7	Allow logged in users to access account page
REQ-48	4	On account page, users are able to click a button to edit personal information

REQ-49	4	On account page, users are able to click a button to make and cancel reservation
REQ-50	6	On account page, users are able to click a button to access balance and payment
REQ-51	4	On account page, users are able to click a button to check their parking status
REQ-50	4	On payment page, users are able to make a deposit with credit card or linking paypal account
REQ-51	3	On reservation page, users are able to input calendar date and time for reservations, or cancel the reservation with double check button
REQ-52	6	On parking status page, users are able to see the parking period, current parking fee and spots location

Home Page UI



Reservation Page With User Logged In



Make Reservation for A New Vehicle:

Location: Inspire Auto Parking Lot

Car Brand: Lamborghini

Car Model: LP - 700

Car Plate:

VIN:

Add This Vehicle

Make Reservation for A Registered Vehicle:

Location: Inspire Auto Parking Lot

Registered Vehicle: Lamborghini (Dream Car)

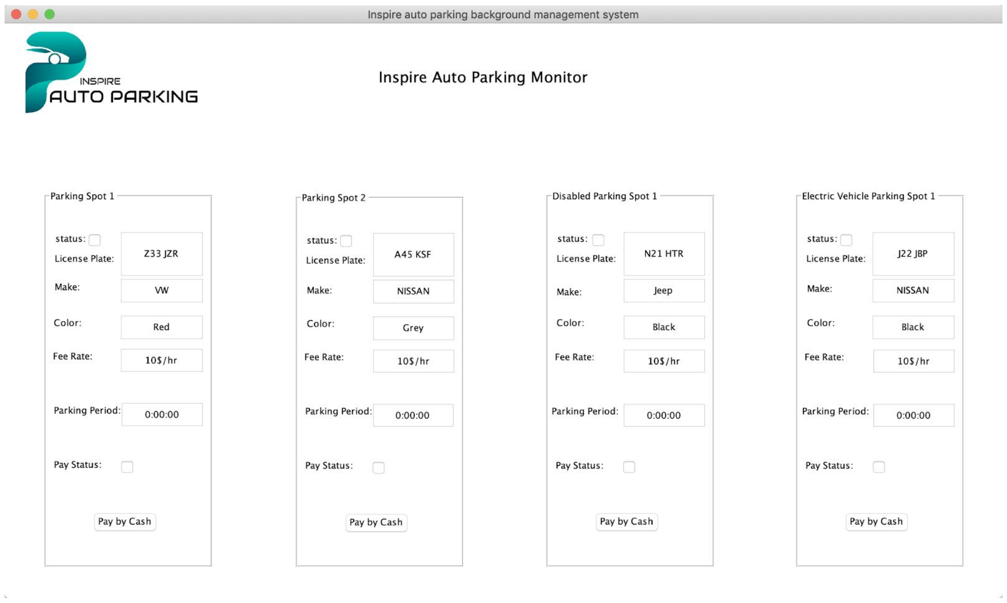
Date: Tomorrow (02/17/2019)

Arriving Time: 00:00

Leaving Time: 23:30

Make Reservation Now

Background management Toolset UI Preview


Inspire auto parking background management system

INSPIRE AUTO PARKING

Inspire Auto Parking Monitor

	Parking Spot 1	Parking Spot 2	Disabled Parking Spot 1	Electric Vehicle Parking Spot 1
status:	☐	☐	☐	☐
License Plate:	Z33 JZR	A45 KSF	N21 HTR	J22 JBP
Make:	VW	NISSAN	Jeep	NISSAN
Color:	Red	Grey	Black	Black
Fee Rate:	10$/hr	10$/hr	10$/hr	10$/hr
Parking Period:	0:00:00	0:00:00	0:00:00	0:00:00
Pay Status:	☐	☐	☐	☐

Section 4: Functional Requirements Specification

a) Stakeholders

Parking Garage owners

Business owners

Parking Customers

b) Actors/Goals

Actors	Type	Goal
Customer	Initiating	Park in the garage
Owner	Initiating	Set reservation limit and pricing
Owner	Supporting	Assist users with any issues
License Plate Camera	Supporting	Read license plates of incoming and exiting vehicles

Proximity sensor	Supporting	Detect when a car has occupied a space
Database	Offstage	Store user information
Parking User Interface	Supporting	Show the customers how many available parking spots left and allow them to reserve for spots before they arriving
Customer	Initiating	Reserve a parking spot online
Customer	Initiating	Know how many available spots left online
Customer	Initiating	Know the time when they arrive and how much time they have parked
Customer	Initiating	Can cancel the reservation after they have reserved for spots successfully
Customer	Initiating	Pay the parking fee online

c) Use Cases

i) Casual Description

UC#1 : Registration

Utilizing REQ-22, REQ-23, REQ-26, REQ-27 the system allows a customer to register for an account by submitting relevant information to an online form. Users must provide at minimum a license plate number, an e-mail address, and a bank account/credit card number.

UC#2 : Reservation

Utilizing REQ-1, REQ-2, REQ-8, REQ-18, REQ-20, and REQ-21, the system allows a customer to reserve a parking spot by specifying, at minimum, a timeframe to our online reservation page. In general, users may not reserve specific spots, but rather, *a* spot in a certain section of the parking lot. If at a later point in time the customer wishes to cancel a reservation, he or she may do so on the reservation confirmation page.

UC#3 : Entering the garage

Utilizing REQ-3, REQ-6, REQ-7,and REQ-8, A driver can enter the garage in their car and the camera will automatically scan his/her car's plate and model. The system will show their information and check which spot he/she should go. System will show a unique QR code for customer to scan, and direct to corresponding spots.

UC#4 : Status Checking

Utilizing REQ-6, REQ-9, REQ-12, REQ-13, REQ-15, and REQ-23, the system shows a customer the state of the parking lot, their car's position, and other relevant information about his or her current stay in the lot on an online visual interface.

UC#5 : Online Payment

Utilizing REQ-14, REQ-15, REQ-16, and REQ-18, the system automatically determines when a car has left the parking lot and charges the account associated with the observed license plate by the appropriate amount to the customer's PayPal

UC#6 : Cancel Reservation

Utilizing REQ-1, REQ-2, and REQ-23, the system allows the customer to cancel a reservation that they made. Based on REQ-2 and REQ-23, user can get their reservation info and reservation status. With the account the user created, they can login to website to cancel reservation.

UC#7 : Price Management

Utilizing REQ-18, the system allows an owner or administrator to adjust lot pricing by using an online form and submitting relevant information.

UC#8 : Reserved Parking

Utilizing REQ-3, REQ-4, REQ-6, REQ-7, REQ-8, and REQ-10, the system automatically determines whether entering customers have a reservation or not, and if so, what type of reservation they have. It then guides them through the parking lot accordingly.

UC#9 : Entering parking spot

Utilizing REQ-8, REQ-11. The customers go to their optimized spots and park there. The sensor will detect the cars has been parked and sensor lights are on which show that the spots are occupied.

UC#10 : Leaving garage

Utilizing REQ-3, REQ-11, REQ-12 The customers end parking and drive their car leave the garage.

ii) Use Case Diagram

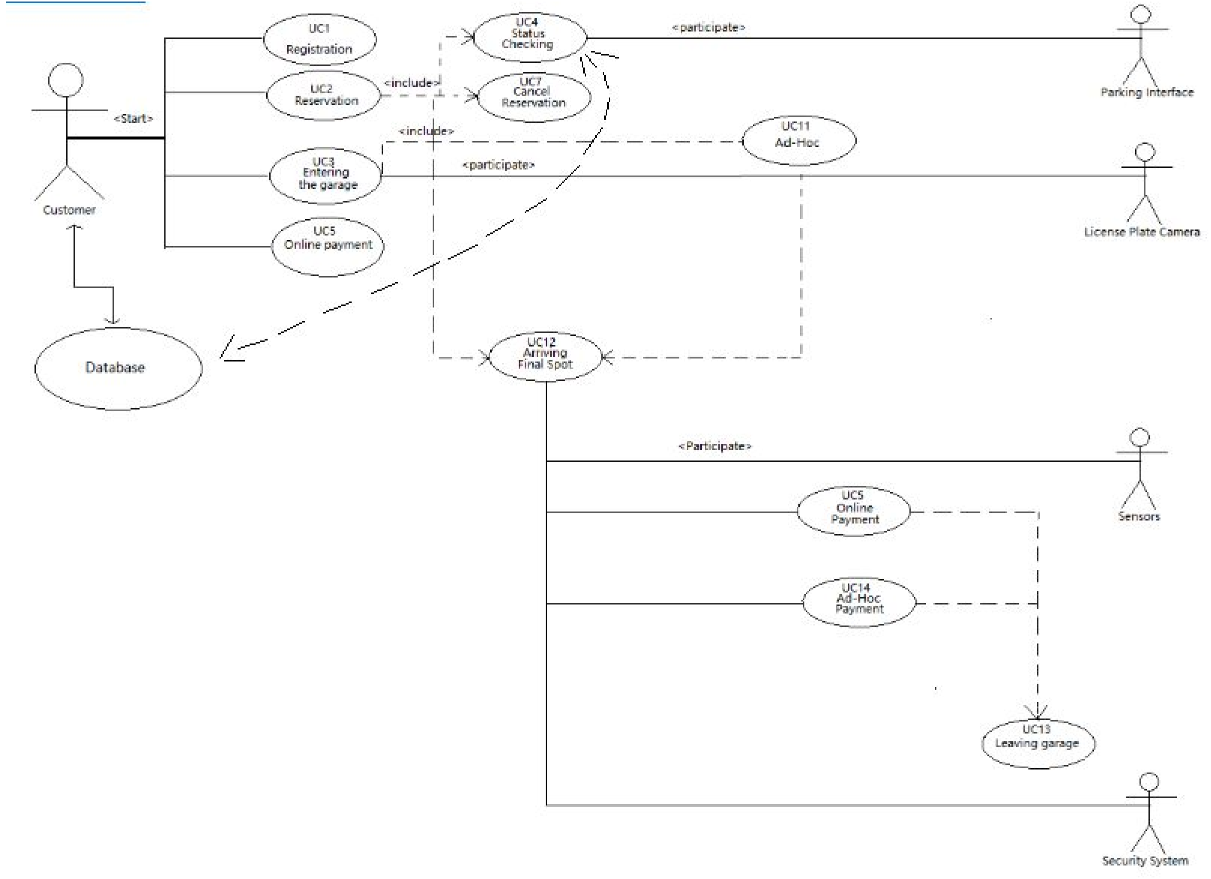


Figure 1: Customer use case

iii) Traceability Matrix

Requirement	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10

REQ1	10		✔				✔				
REQ2	8		✔				✔				
REQ3	10			✔					✔		
REQ4	10								✔		
REQ5	8					✔					
REQ6	9			✔	✔				✔		
REQ7	5			✔					✔		
REQ8	6		✔	✔				✔	✔	✔	
REQ9	4				✔						
REQ10	5								✔		
REQ11	7									✔	✔
REQ12	6				✔						✔
REQ13	4				✔						
REQ14	10					✔					
REQ15	10				✔	✔					
REQ16	9					✔					
REQ17	2										
REQ18	3		✔			✔					
REQ19	6										
REQ20	5		✔								
REQ21	3		✔								
REQ22	5	✔									
REQ23	6	✔			✔		✔				
REQ24	7										
REQ25	7										
REQ26	10	✔									
REQ27	10	✔									
Max PW		10	10	10	10	10	10	6	10	7	7
Total PW		31	35	30	39	40	24	6	45	13	13

iv) Fully-Dressed Description

Use Case UC-1: Registration
Related Requirements: REQ-22, REQ-23, REQ-26, REQ-27 Initiating Actor: Customer Actor’s Goal: Using the website to register for an account. Information stored in the database Participating Actors: Customer, Database, Garage Manager Preconditions: *A functional database *Graphical User Interface Postconditions: The customer’s account will be stored in database Flow of Events for Main Success Scenario: $->$ 1. The customer goes to the website and choose to sign in. $<-$ 2. The system returns a registration page. $->$ 3. The customer enter its email address, username and password. $<-$ 4. The system delivers the username and email to check if they are occupied in database a. If any username and email occupied, system returns to step 3 and notify customer to revise. b. If no username and email occupied, system continue to step 5. $<-$ 5. The system show a message to customer that registration successed.

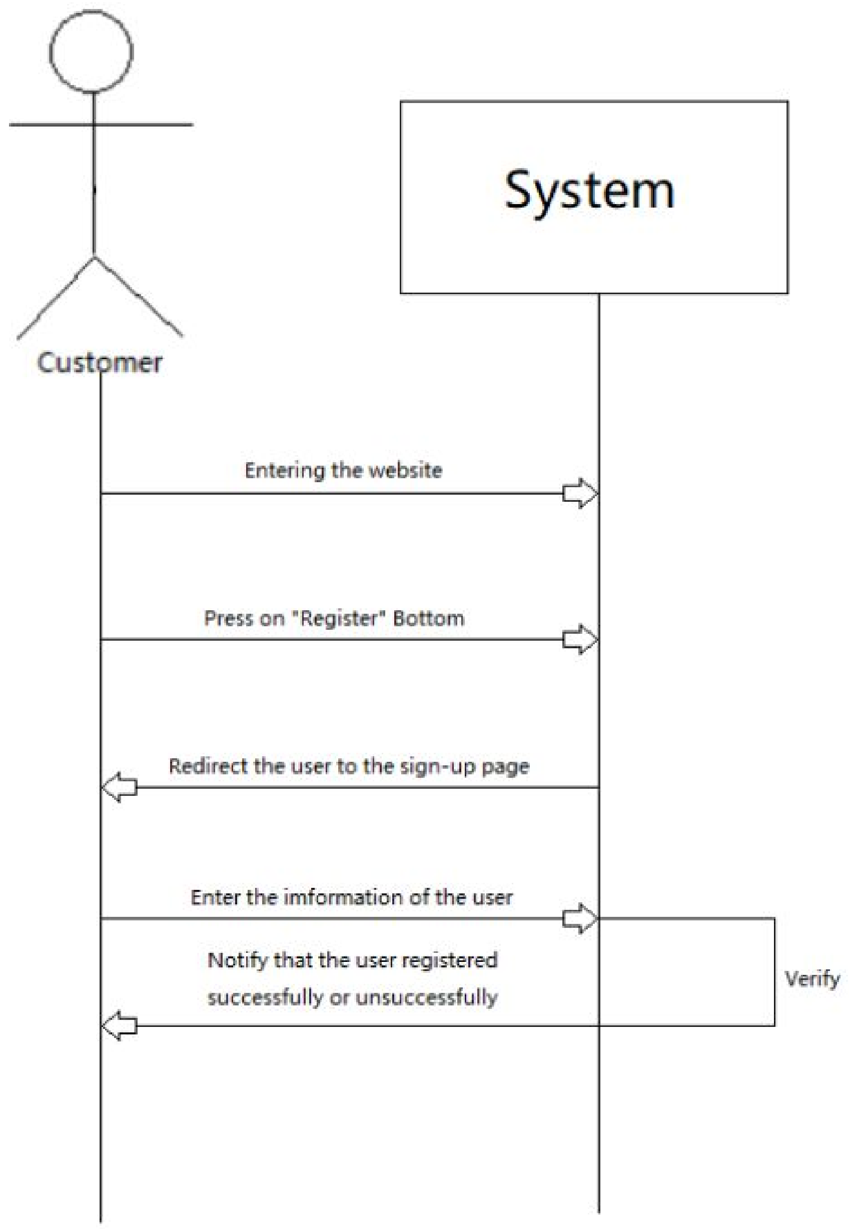

System
Customer
Entering the website
Press on "Register" Bottom
Redirect the user to the sign-up page
Enter the imformation of the user
Notify that the user registered successfully or unsuccessfully
Verify

Figure 2: Register Process

Use Case UC-2: Reservation
Related Requirements: REQ-1, REQ-2, REQ-8, REQ-18, REQ-20, and REQ-21 Initiating Actor: Customer Actor’s Goal: Make a reservation for a parking spot Participating Actors: Customer, Garage Website, Database Preconditions: *An account with sufficient information for reservation *Website with garage status Postconditions: Database updates and marks specific spot as occupied Flow of Events for Main Success Scenario: − > 1. The customer goes to the website and choose to reservation < − 2. The system check current cookie a. If customer is logged in, continue to step 4. b. If customer is not logged in, continue to step 3. < − 3. The system returns a login page, when customer finishes logged in, continue to step 4. < − 4. The system returns a reservation page. − > 5. The customer enters calendar time period and specific spots to reservation data fields. < − 6. The system checks the database of reservation informations. a. If there are available spots in selected time period, system will returns a UI page that indicating available spots for customer to choose, continue to step 7. b. If there are no available spots in selected time period, system will pop a message that no available spots, back to step 5. − > 7. The customer selects a parking spot. < − 8. The system pop a message that reservation successed, send text and email confirmation to customer; database updates.

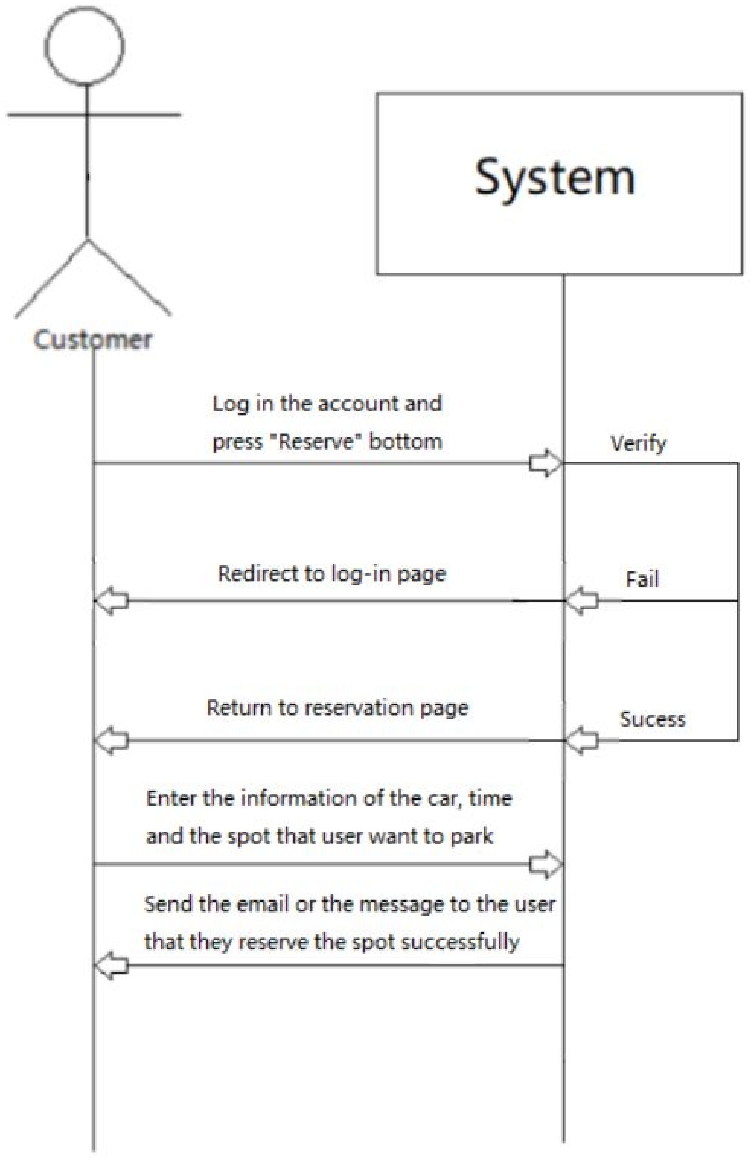

System
Customer
Log in the account and press "Reserve" bottom
Verify
Redirect to log-in page
Fail
Return to reservation page
Sucess
Enter the information of the car, time and the spot that user want to park
Send the email or the message to the user that they reserve the spot successfully

Figure 3 :Reservation Process

Use Case UC-3: Entering the garage
Related Requirement: REQ-3, REQ-6, REQ-7,and REQ-8 Initiating Actor: Customer Actor’s Goal: Park in to the desired spot Participating Actors: Customer, License Plate Camera, Database, Parking spot sensor Preconditions: *A reservation to guarantee a parking spot. *N/A for non ad-hoc customer Postcondition: *If a new vehicle has parked, the database will be updated. Marking the spot as occupied. Flow of Event for Main Success Scenario: $->$ 1. The customer drives to the garage gate. $<-$ 2. The camera will scan the plate of the vehicle. a. If the customer has a reservation, it will be guided to the reserved parking spot. b. If the customer has no reservation, system continue to step 3. $<-$ 3. The system requests information from database to check the availability of parking spots. a. If there is available spots, system continue to step 4. b. If there is no available spots, system continue to step 5. $<-$ 4. The system will show the customer a unique QR code to scan. $<-$ 5. The system will show no available spots, customer has to leave. $->$ 6. The customer drives in or leaving garage.

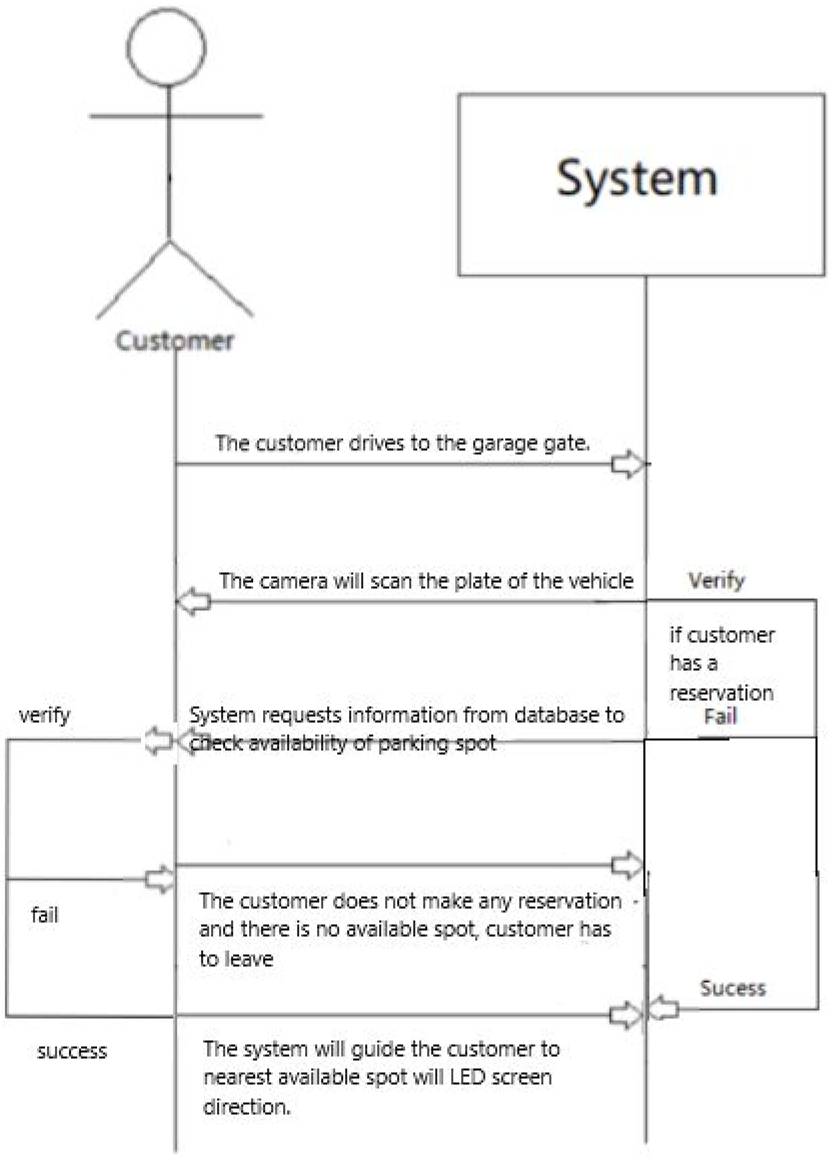
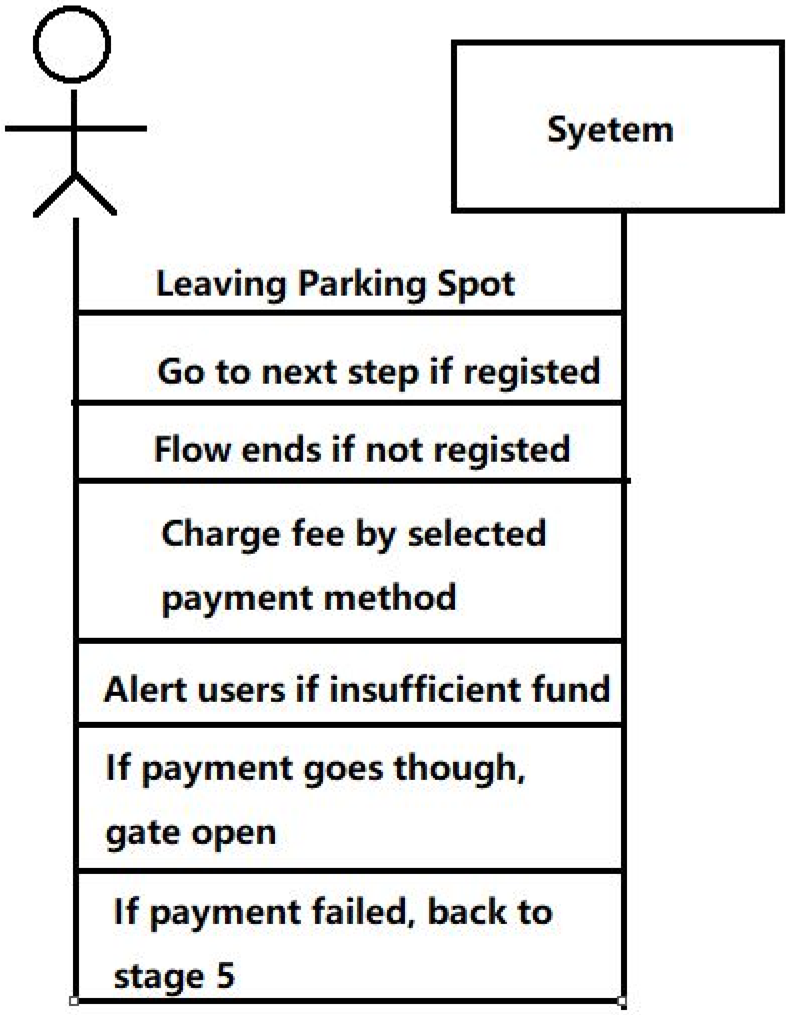

System
Customer
The customer drives to the garage gate.
The camera will scan the plate of the vehicle
Verify
if customer has a reservation
Fail
verify
System requests information from database to check availability of parking spot
fail
The customer does not make any reservation and there is no available spot, customer has to leave
Sucess
success
The system will guide the customer to nearest available spot will LED screen direction.

Figure 4 :Entering garage process

Use Case UC-4: Online Payment
Related Requirement: REQ-14, REQ-15, REQ-16, and REQ-18 Initiating Actor: Customer Actor’s Goal: Pay for their parking fee Participating Actors: Customer, Database, License Plate Camera, Owner Preconditions: *The registered customer about to leave garage. Postconditions: Gate will open to let customer leave. Flow of Event for Main Success Scenario: − > 1. The customer drives its car leaving parking spot. < − 2. Sensor indicate system that customer is leaving, system check a. If the customer is a registered customer, continue to step 3. b. If the customer is not a registered customer, flow ends. < − 3. The system check the account balance and payment linking for valid payment a. If the customer has valid payment method, continue to step 4. b. If the customer doesn’t have valid payment method, continue to step 5. < − 4. The system will charge the parking fee from customer’s valid payment method. < − 5. The system will use a screen, text and email to notify customer that insufficient funds. − > 6. Customer drives to main gate. a. If the customer has paid, continue to step 7. b. If the customer hasn’t paid, returns to step 5. < − 7. Security system will open the gate.


Syetem
Leaving Parking Spot
Go to next step if registed
Flow ends if not registed
Charge fee by selected payment method
Alert users if insufficient fund
If payment goes though, gate open
If payment failed, back to stage 5

Full Description Case 4

User Interface Specification

outline:
*user enters the website - can login or sign-up:
sign-up : requires a license plate number & valid e-mail address

> newcomers are prompted to create an account via a dynamically generated QR code which will not only link the user to the sign-up page but pre-fill the license plate number.
> account is in use, but once the new user parks, they are prompted to enter a valid Email address and create a password.
> after verification, user legit owns the account
> a card reader is at the exit; if at this point the user hadn't yet specified a credit card #, they must swipe, after which the data will automatically be saved to their account for them

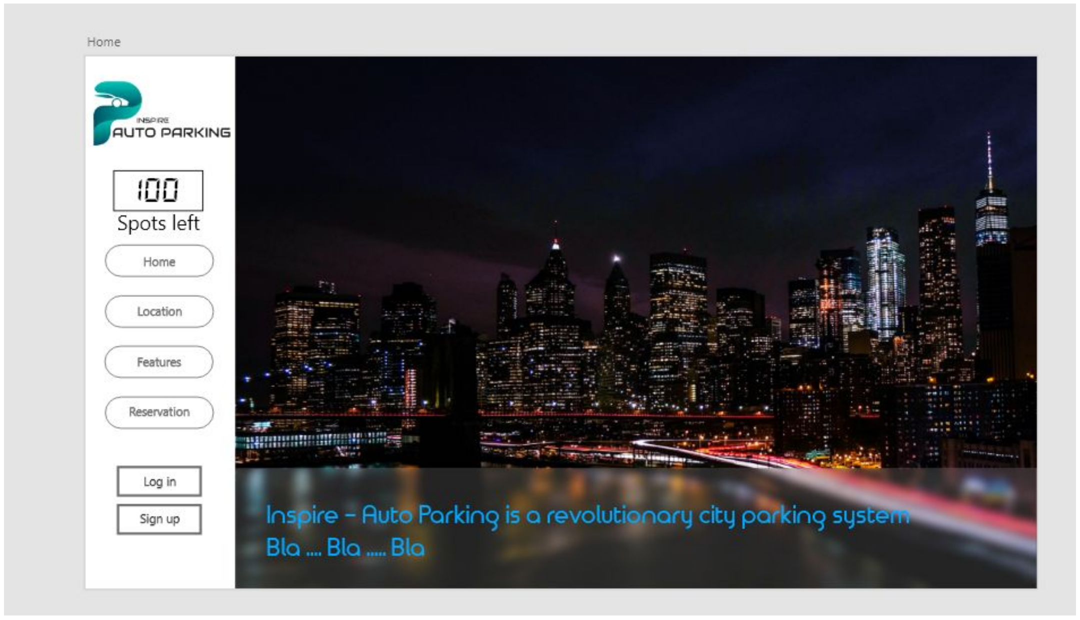
those who have already registered don't have to do anything but drive in and then drive out

one click to confirm license plate number
user parks
one click to select parking spot
form entry : e-mail
form entry : password
form entry : password again
one click to press ok
if not (form entry : credit/whatever card number) then
> swipe card reader on exit

a logged in user may modify information in their account from a single form
then a save button at the bottom applies the changes

User Website Preview

Home Page UI



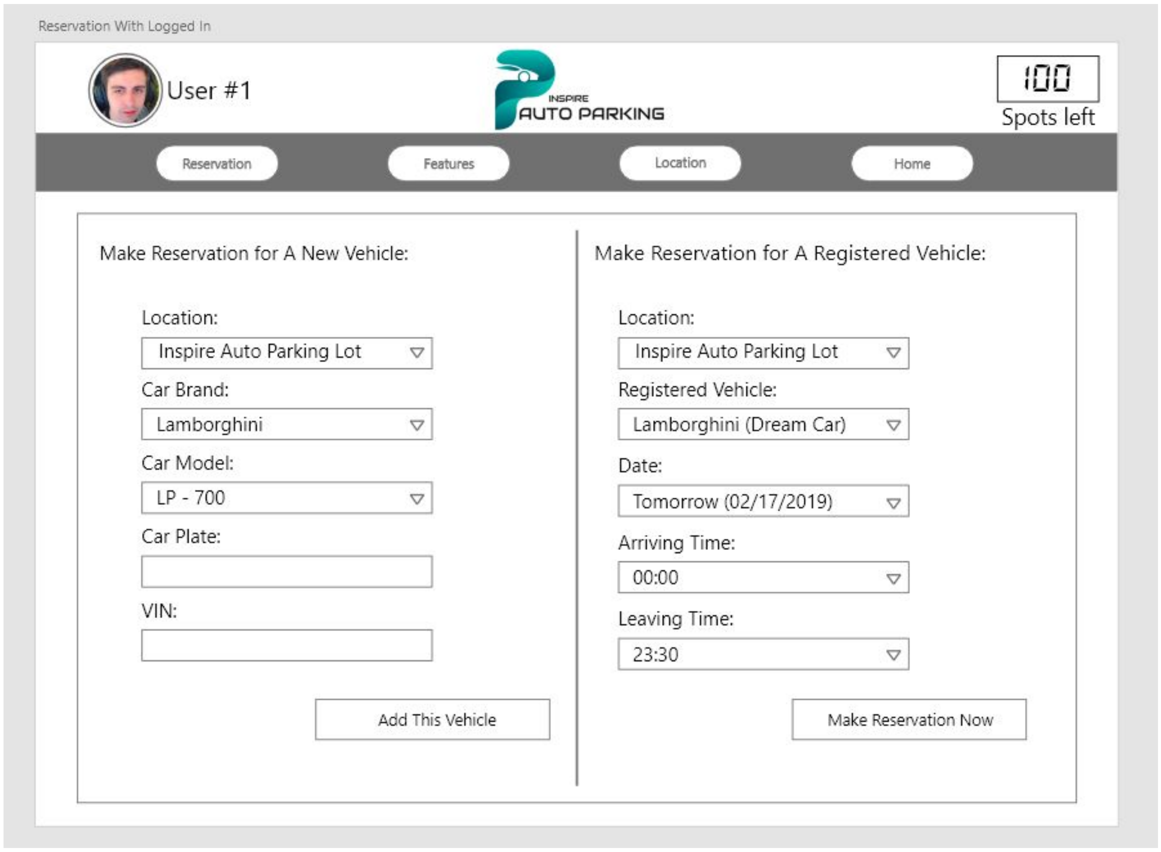
In the home page, depending on the user’s choice of action, there will be 1 mouse click for navigation and 0 keystroke.

1 mouse click, 1/1 navigation, 0/1 clerical data entry.

0 keystroke,

‘

Reservation Page With User Logged In


Reservation With Logged In

User #1

100 Spots left

Make Reservation for A New Vehicle:

Location: Inspire Auto Parking Lot

Car Brand: Lamborghini

Car Model: LP - 700

Car Plate:

VIN:

Make Reservation for A Registered Vehicle:

Location: Inspire Auto Parking Lot

Registered Vehicle: Lamborghini (Dream Car)

Date: Tomorrow (02/17/2019)

Arriving Time: 00:00

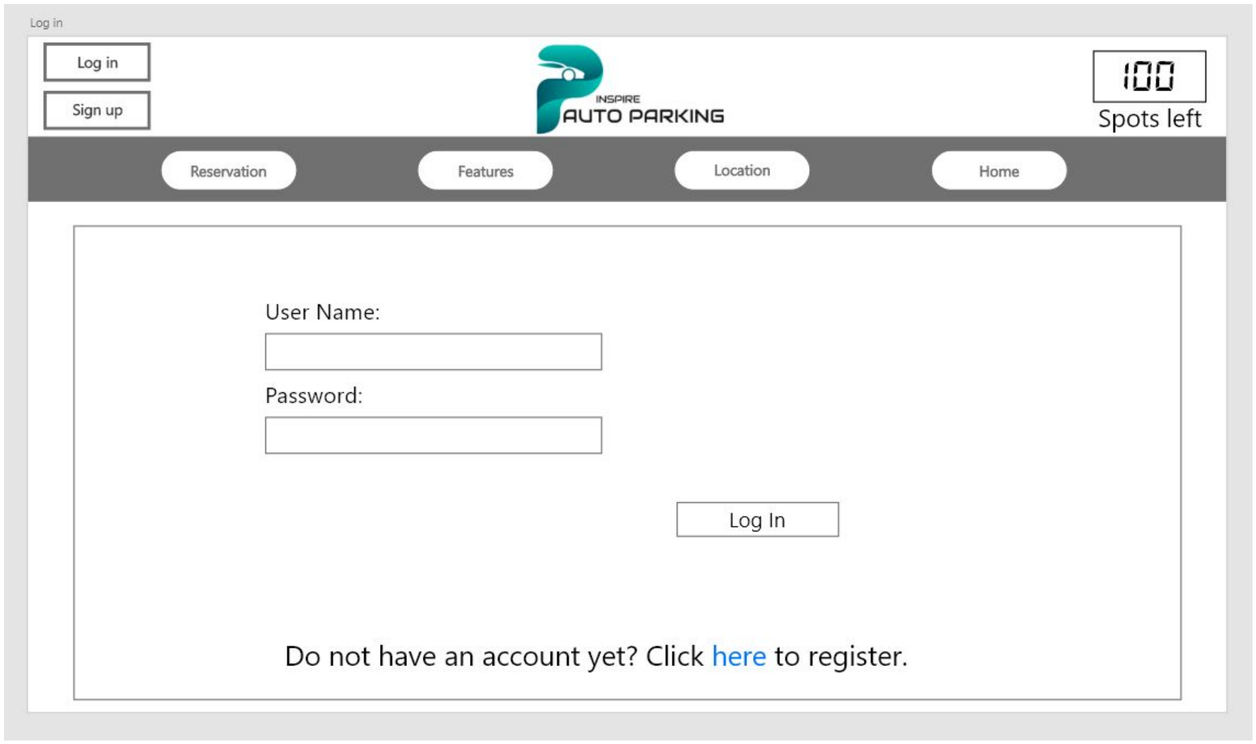
Leaving Time: 23:30

In this page, the user will perform 16 mouse clicks since they have to click on the arrow to show all the detailed information of each block and another click to choose one option. 8/16 of the mouse clicks will be clerical data entry and the rest will be for navigation. And 17 keystrokes will be needed for the VIN number, at least 6 keystrokes will be needed for the Car plate (based on car plate in NJ). 23/23 of the keystrokes are clerical data entry.

16 mouse clicks. 8/16 clerical data entry, 8/16 navigation.

23 keystrokes. 23/23 clerical data entry, 0/23 navigation.

Login Page

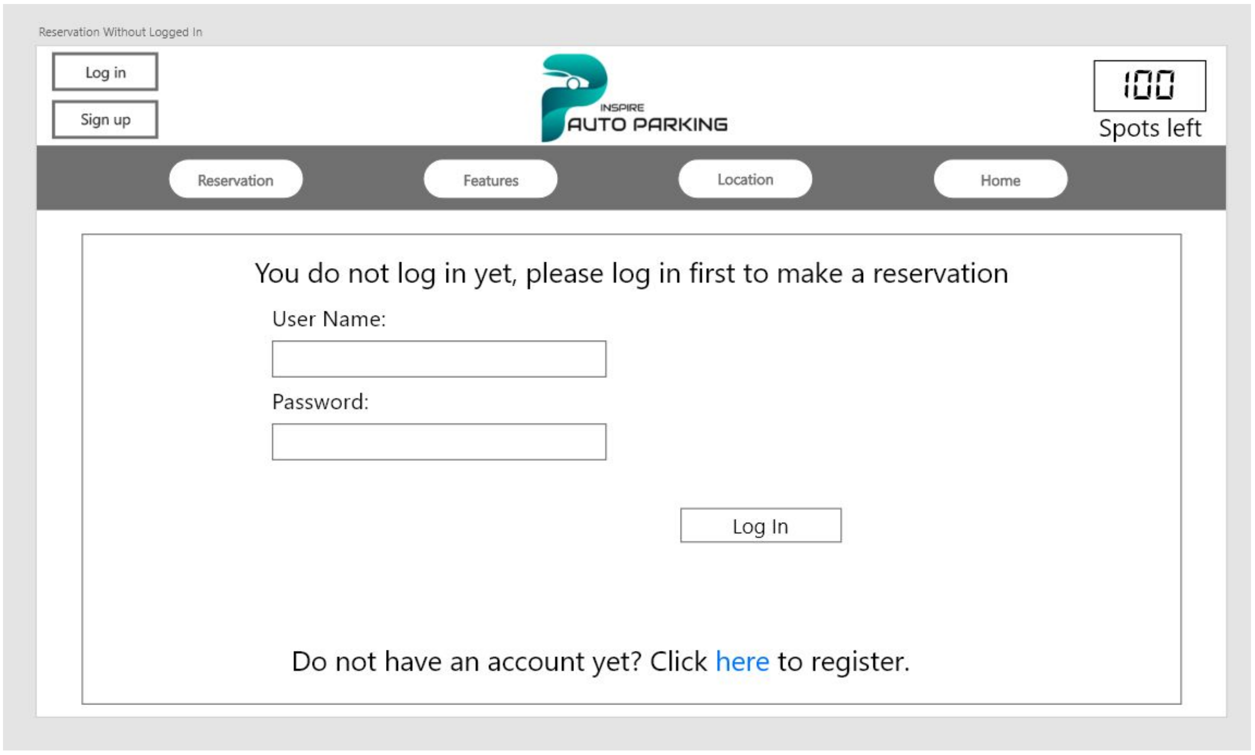


In this page, 1 mouse click is needed for navigation, keystroke base on their username and password.

1 mouse click. 0/1 clerical data entry. 1/1 navigation.

n Keystroke. n/n Clerical data entry. 0/n navigation.

Reservation Page Without User Logged In



You do not log in yet, please log in first to make a reservation

User Name:

Password:

Log In

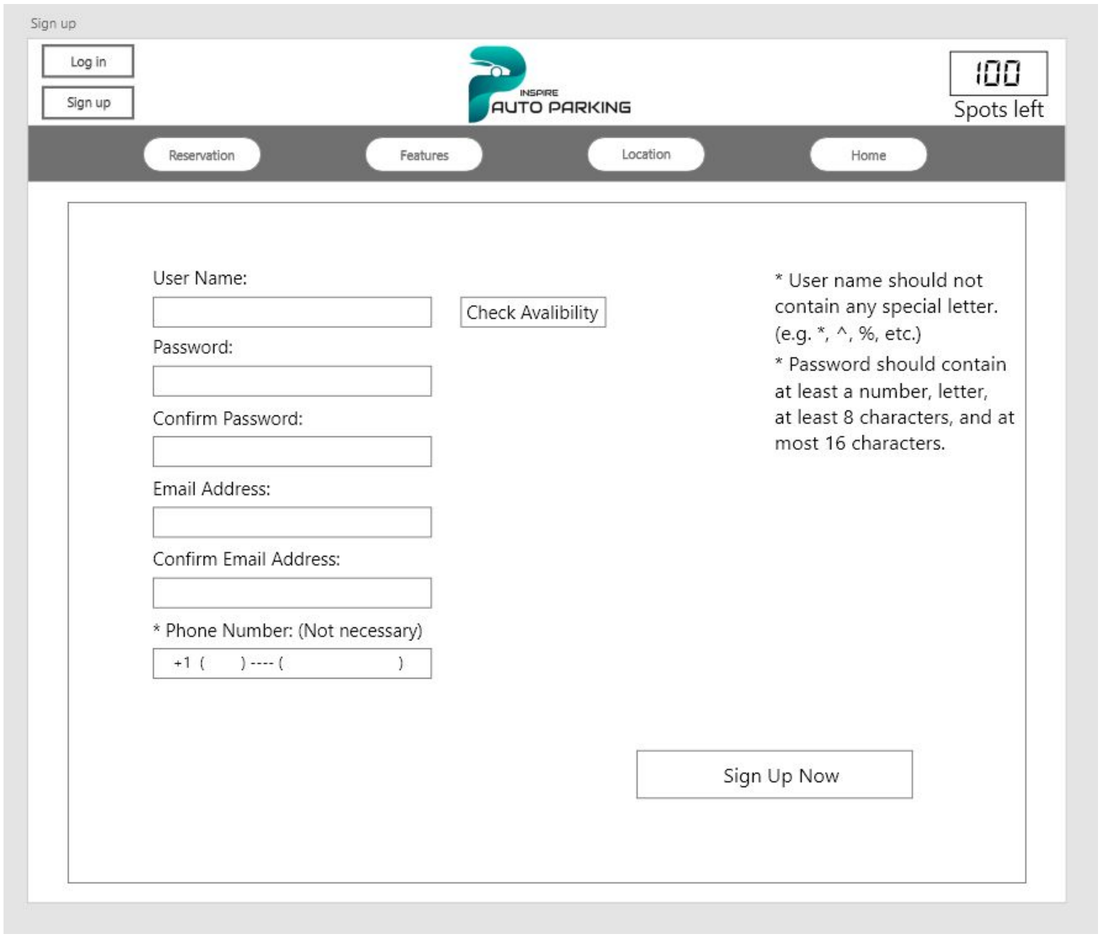
Do not have an account yet? Click here to register.

In this page, 1 mouse click is needed for navigation, keystroke base on their username and password.

1 mouse click. 0/1 clerical data entry. 1/1 navigation.

n Keystroke. n/n Clerical data entry. 0/n Navigation.

Sign Up Page



User Name:

Password:

Confirm Password:

Email Address:

Confirm Email Address:

* Phone Number: (Not necessary)

+1 ( ) ---- ( )

* User name should not contain any special letter. (e.g. *, ^, %, etc.)

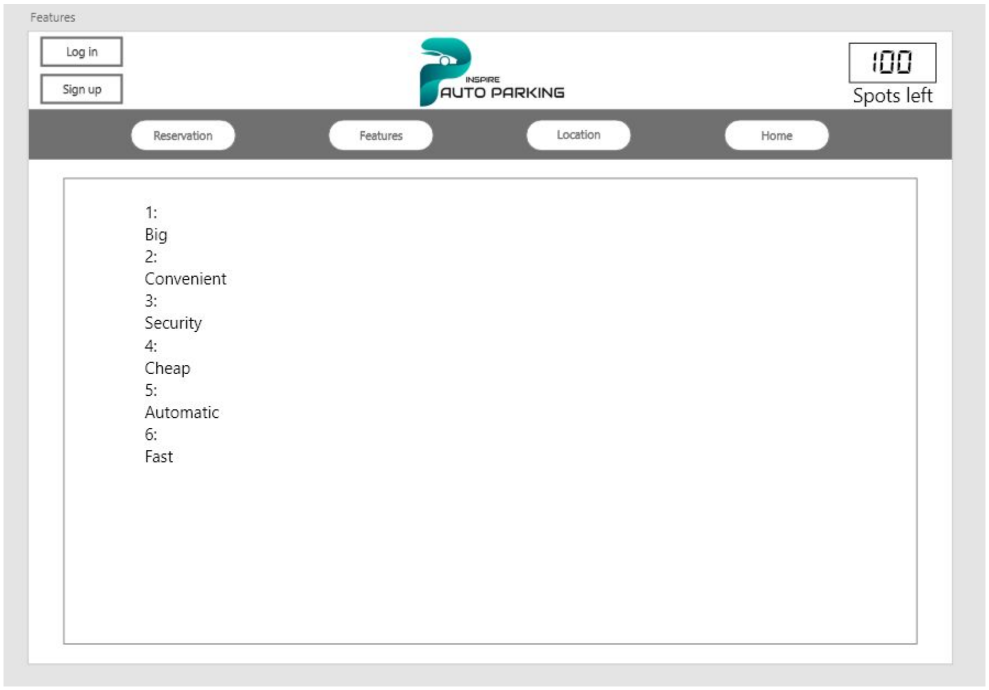
* Password should contain at least a number, letter, at least 8 characters, and at most 16 characters.

In this page, there will be 2 mouse clicks.1 of them is clerical data entry to check availability. The other one is for navigation. Keystroke counts based on user's chose.

2 mouse clicks, 1/2 clerical data entry. 1/2 navigation.

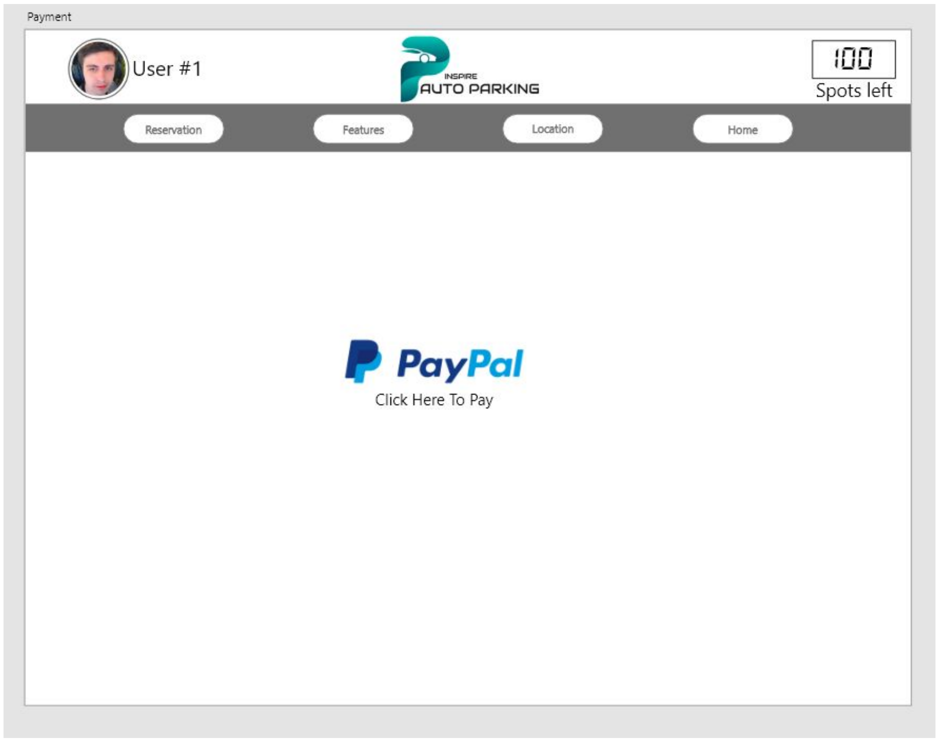
n key strokes. n/n clerical data entry. 0/n navigation.

Features Page

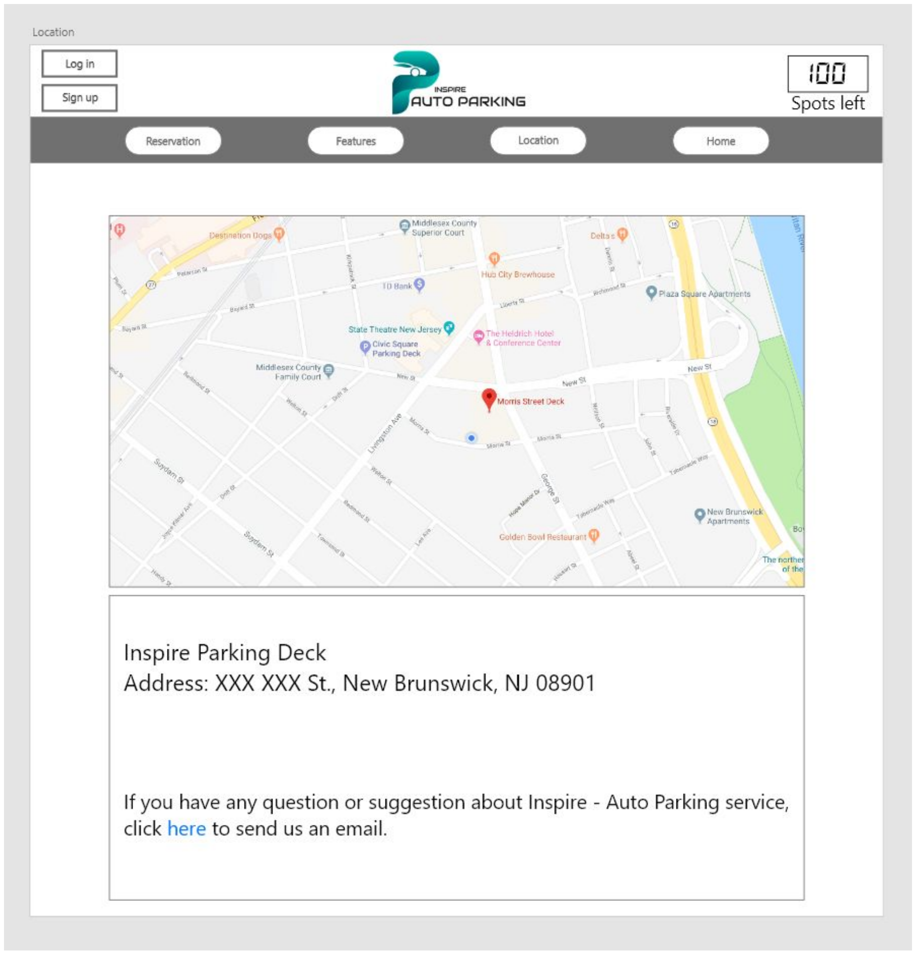


1:
Big
2:
Convenient
3:
Security
4:
Cheap
5:
Automatic
6:
Fast

Payment Page



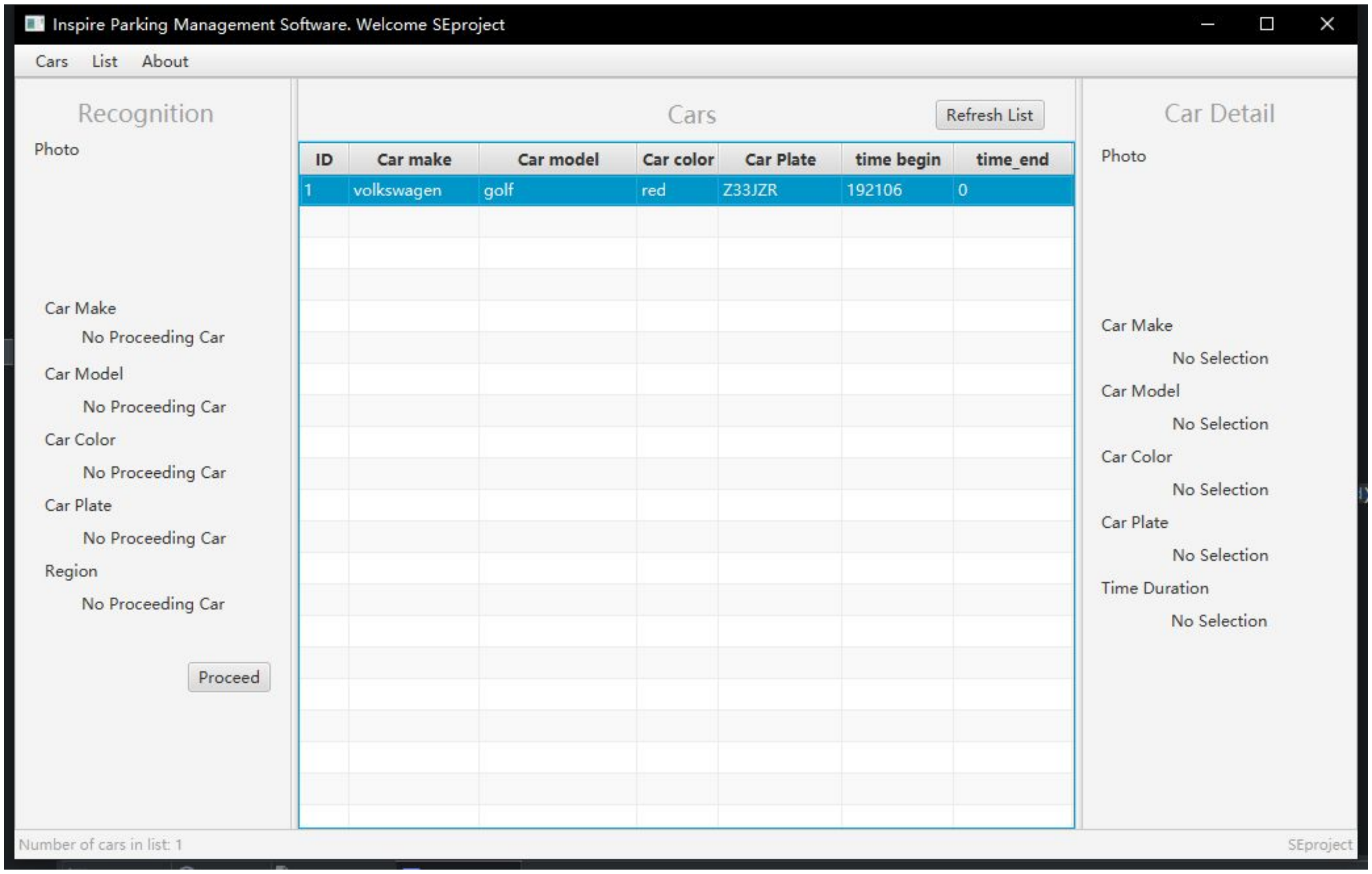
Location Page



Inspire Parking Deck
Address: XXX XXX St., New Brunswick, NJ 08901

If you have any question or suggestion about Inspire - Auto Parking service, click here to send us an email.

Background management Toolset UI Preview


Inspire Parking Management Software. Welcome SEproject
Cars
List
About
Recognition
Photo
Cars
Refresh List
Car Detail
Proceed
Number of cars in list: 1
SEproject

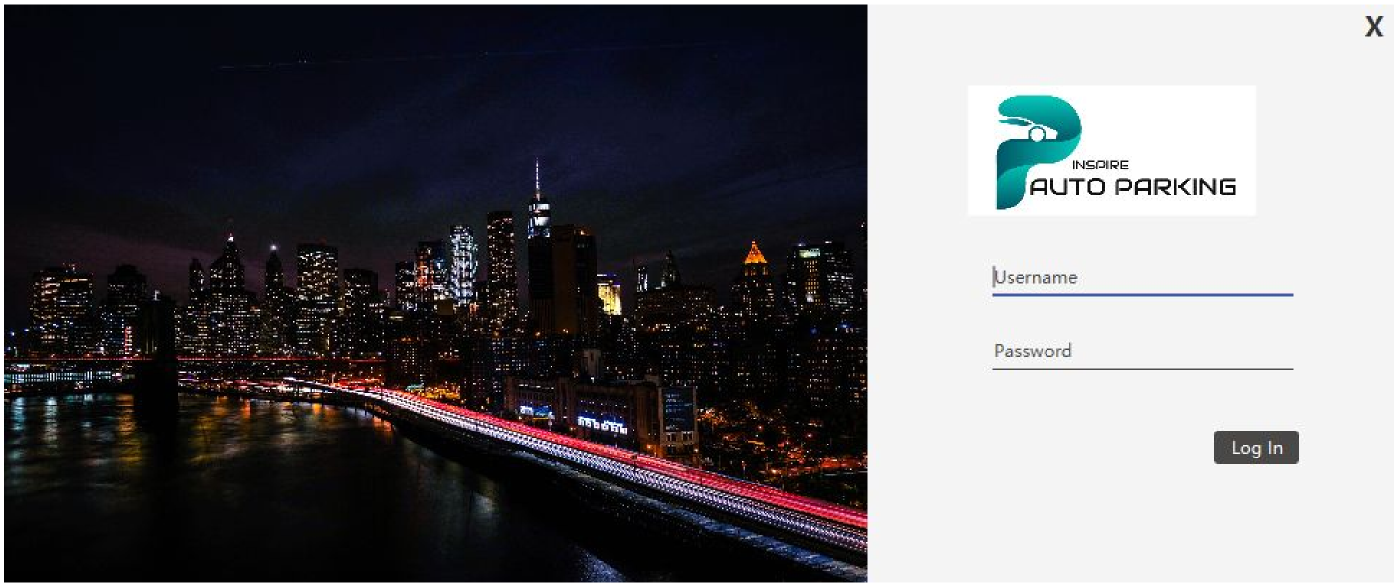
ID	Car make	Car model	Car color	Car Plate	time begin	time_end
1	volkswagen	golf	red	Z33JZR	192106	0

Car Make
No Proceeding Car
Car Model
No Proceeding Car
Car Color
No Proceeding Car
Car Plate
No Proceeding Car
Region
No Proceeding Car

Car Make
No Selection
Car Model
No Selection
Car Color
No Selection
Car Plate
No Selection
Time Duration
No Selection

Inspire auto parking system background management toolset

Inspire auto parking system background management toolset Login Preview



Section 5: Project size estimation based on Use Case Points.

Use Case Classification
UUCP: 6(15)+5(10)+3(5) = 155

Use Case	Description of relevant Characteristics	Complexity	Weight
UC-1 Registration	Simple website GUI. 5 steps for the main success scenario. 2 participating actors	Average	10
UC-2 Reservation	Complex website GUI. 8 steps for the main success scenario. 3 participating actors	Complex	15
UC-3 Entering Garage	No UI. 6 steps for the main success scenario. 3 participating actors	Average	10
UC-4 Status Check	No UI. 4 steps for the main success scenario. 3 participating actors	Average	10
UC-5 Online Payment	Complex website GUI. 6 steps for the main success scenario. 2 participating actors	Complex	15
UC-6 Cancel Reserve	Simple website GUI. 5 steps for the main success scenario. 3 participating actors	Average	10
UC-7 Price Manage	Simple backend GUI. 4 steps for the main success scenario. 2 participating actors	Simple	5
UC-8 Reserved Park	Display interface; 4 participating actors; 4 steps success scenario.	Complex	15
UC-9 Enter spot	No interface needed; 3 participating actors; 3 steps success scenario.	Complex	15
UC-10 Leaving Garage	No interface needed; 3 participating actors; 3 steps success scenario.	Complex	15

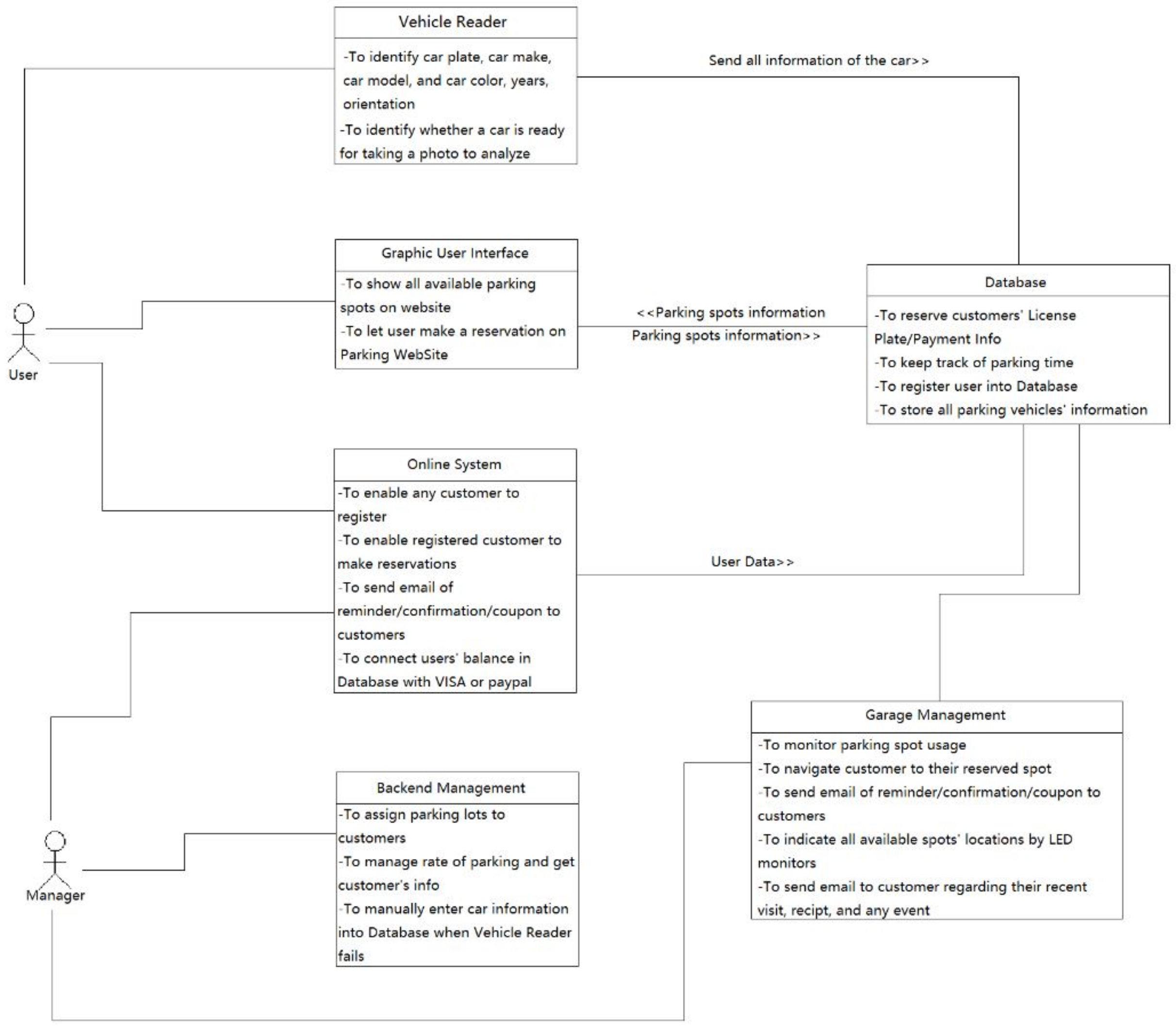
Technical Complexity Factors (TCFs)

Technical Factor	Description	Weight	Perceived Complexity	Factor (W*PC)
T1	Distributed, Web-based system	2	3	6
T2	User expect good performance but nothing exceptional	1	3	3

T3	End-user expects efficiency	1	3	3
T4	Internal processing is relatively complex	1	3	3
T5	No requirement for reusability	1	0	0
T6	Ease of install is moderately important	0.5	3	1.5
T7	Ease of use is very important	0.5	5	2.5
T8	No portability requirement, system is web-based	2	0	0
T9	Adding payment methods into the system	3	2	6
T10	Concurrent use is required	1	4	4
T11	Security is a significant concern	1	5	5
T12	Third parties included	1	3	3
T13	No unique training needs	1	0	0
			Technical Factor Total:	37

Section 6: Domain Analysis

Domain Model Diagram


Vehicle Reader
-To identify car plate, car make, car model, and car color, years, orientation
-To identify whether a car is ready for taking a photo to analyze
Send all information of the car>>
Graphic User Interface
-To show all available parking spots on website
-To let user make a reservation on Parking WebSite
<<Parking spots information
Parking spots information>>
Database
-To reserve customers' License Plate/Payment Info
-To keep track of parking time
-To register user into Database
-To store all parking vehicles' information
User
Online System
-To enable any customer to register
-To enable registered customer to make reservations
-To send email of reminder/confirmation/coupon to customers
-To connect users' balance in Database with VISA or paypal
User Data>>
Garage Management
-To monitor parking spot usage
-To navigate customer to their reserved spot
-To send email of reminder/confirmation/coupon to customers
-To indicate all available spots' locations by LED monitors
-To send email to customer regarding their recent visit, recipt, and any event
Backend Management
-To assign parking lots to customers
-To manage rate of parking and get customer's info
-To manually enter car information into Database when Vehicle Reader fails
Manager

Domain Model Derivation

The domain model above was derived from the use cases and the requirements. We categorize all the functional requirements into different concepts (database, online system..etc). With only a few concepts, it's easy for us to show how all the participating actors are connected.

As the customer approaches the garage gate, the pressure sensor will be triggered, causing the camera to take a picture of the car plate. After that, the picture will be transmitted to the plate recognizing API we are using. After that, the system will determine whether the customer has reservation or not. Then the customer will be guided to their reserved spot or empty spot for ad-hoc customers. After parked, that specific spot will be marked as occupied and will be updated to the database. That's the end of the parking process. The customer will be able to check their parking time and current rate during their parking. Also, all the information is available to the garage manager all the time.

All the concepts are interacting with the database and the database can only be accessed by the manager to ensure the security.

I. Concept definitions

Responsibility Description	Type	Concept
To reserve customers' License Plate/Payment/Balance/Vehicle Info	N	Database
To identify car plate, car make, car model, and car color	D	Vehicle Reader
To identify whether a car is ready for taking a photo to analyze	K	Vehicle Reader
To monitor parking spot usage	K	Garage Management
To enable any customer to register	D	Online System
To enable registered customer to make reservations	D	Online System
To register user into Database	K	Database
To navigate customer to their reserved spot	D	Garage Management
To show all available parking spots on website	K	Online System
To send email of reminder/confirmation/coupon to customers	D	Garage Management
To assign parking lots to customers	D	Backend Management
To manage rate of parking and get customer's info	D	Backend Management
To manually enter car information into Database when Vehicle Reader fails	D	Backend Management
To register user into Database	D	Database
To manually enter car information into Database when Vehicle Reader fails	D	Online System
To indicate all available spots' locations by the virtual map on website	K	Garage Management
To connect users' balance in Database with VISA or paypal	N	Online System
To send email to customer regarding their recent visit, receipt, and any event	D	Garage Management

II. Association definitions

Concept Pair	Association Description	Association Name
Vehicle Reader ←→	Vehicle reader identifies car plate, model and color,	Store car info

Database	sending the information to the database. Database stores these information for future use	
Online System ←→ Database	Online System collects the user data and send them to database. Online System also retrieves parking spots information from database for users to make reservation.	Store and Exchange Informations
Garage Management ←→ Database	Garage management retrieves parking information from database. Garage management system may interact with customer through phone screen.	Get Parking Info
Backend Management ←→ Database	Backend Management control and edit Database	Modifying data
Security System ←→ Garage Management	Security System requests information from Garage Management and determine necessary actions	Security Checking

III. Attribute definitions

Concept	**Attributes**	**Attribute Description**
Login request	Customer	Used to determine the identity of customer and determine the rate of parking
	Manager	Used to determine if user have the authority to backend management page
Preprocessor	License Plate Recognition	Get results of LPR system from API; need to specify every information from API and transmit info to backend management
Management	Database	List of customer info for make reservation/payment
	Manage Toolset	UI which hold useful tools for manager to manipulate the rate of parking
Payment Request	Website	Used to initiate a online payment
Notification	E-mail	Contact customers to notify them for confirmation and discounts

IV. Traceability matrix — show how your use cases map to your domain concepts.

PW	**Use Cases**	**Vehicle Reader**	**Online System**	**Database**	**Garage Management**	**Backend Management**
10	UC-1		✔	✔		

10	UC-2		✔	✔	✔	
9	UC-3	✔		✔	✔	
8	UC-4		✔	✔	✔	
8	UC-5		✔	✔		
4	UC-7		✔	✔	✔	
5	UC-8			✔		✔
7	UC-10	✔	✔	✔	✔	
7	UC-12	✔		✔	✔	
7	UC-13	✔		✔	✔	

System Operation Contracts:

UC-1:Registration
- Preconditions:
  - Manager needs a functional database
  - The customer is not in the manager's database
  - The customer needs to enter a valid username, email address
- Postconditions:
  - The customer's account will be stored in database

UC-2:Reservation
- Preconditions:
  - The customer needs to have an account with sufficient information for reservation
  - The manager needs to offer the website with garage status
- Postconditions:
  - Database updates and marks specific spot as occupied

UC-3: Entering the garage
- Preconditions:
  - A reservation to guarantee a parking spot.
  - N/A for non ad-hoc customer
- Postconditions:
  - If a new vehicle has parked, the database will be updated. Marking the spot as occupied.

UC-4: Online Payment
- Preconditions:
  - The registered customer about to leave garage.
- Postconditions:
  - Gate will open to let customer leave.

Section 7: Interaction Diagrams

Use Case 1: Registration

The registration is the most important idea since our system features (reservation, status checking, and etc.) are entirely based on registered account. We provide unregistered customers the services that nothing different from regular parking lot. So, to ensure that those unique features can be experienced by our registered customers, we need to make sure that our customers can easily use our system, by successful registration at first. For our website, it has the responsibility that allows customers to modify their accounts, such as registration, login, make reservations, and etc. Here it comes the first things first, registration.

For use case 1 Registration, the user who does not have an account for our website needs to enter our website first and click on the Register bottom to create an account. By entering the basic information of the user and the user's car, the user can create an account successfully and the user will be notified by our system that the account has been created successfully or not. User's information will be stored in our database, also the database will show the data stored successfully or not. Database can access the data that input by the user or the system to make a better communication between the user and the manger. Then the user can log in his account and make a reservation which is use case 2.

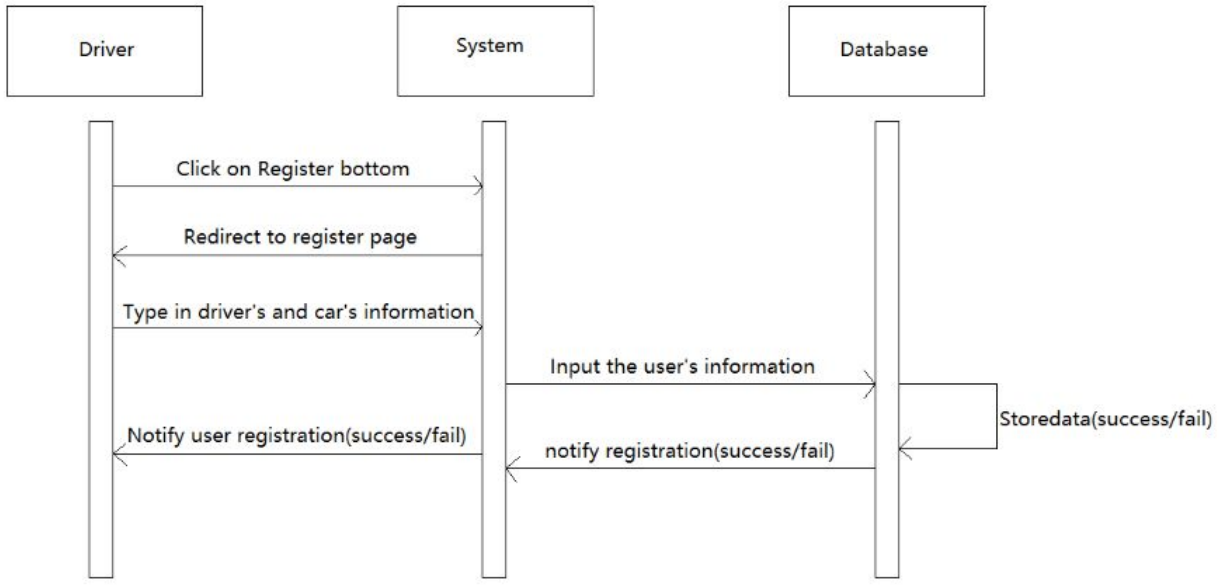

Driver
System
Database
Click on Register bottom
Redirect to register page
Type in driver's and car's information
Input the user's information
Storedata(success/fail)
Notify user registration(success/fail)
notify registration(success/fail)

Figure 1: Sequence Diagrams for Use Case 1.

Use Case 2: Reservation

The assignment of responsibilities of each role is based on privacy and safety. All parking details and users information is unknown to the website. The only data is known to the website is the result of each test, and the website takes action based on each test's result. In other words, the website is a mediator. Since the parking information is changed rapidly, the parking spots information is not stored in the database for space and time concern. The detailed procedure of reservation is stated below.

User sends a page request by clicking the reservation button. By examining cookie data in the user's browser, the website will either return a reservation page or login page depending on whether the user is logged in or not. In the case of that user is not logged in, the user needs to log in or register an account in the returned registration page. If the user chooses to create an account, they will send their data to the website with a UserInfo container. The website will call an availability test to the Database. If the test passes, the user will be redirected to the reservation page, and the database will store user info into. If the test failed, the website will send the registration page to the user. Once the user is in the reservation page, they will send their reservation preference within a ResInfo container to the website. The website will call an availability test to the database. The database will request all available parking spots at that specific time from Garage. If the test fails, the website returns the reservation page to the user. If the test passes, the website returns a reservation success page, and send all reservation detail to the user via email or text based on the user's preference.

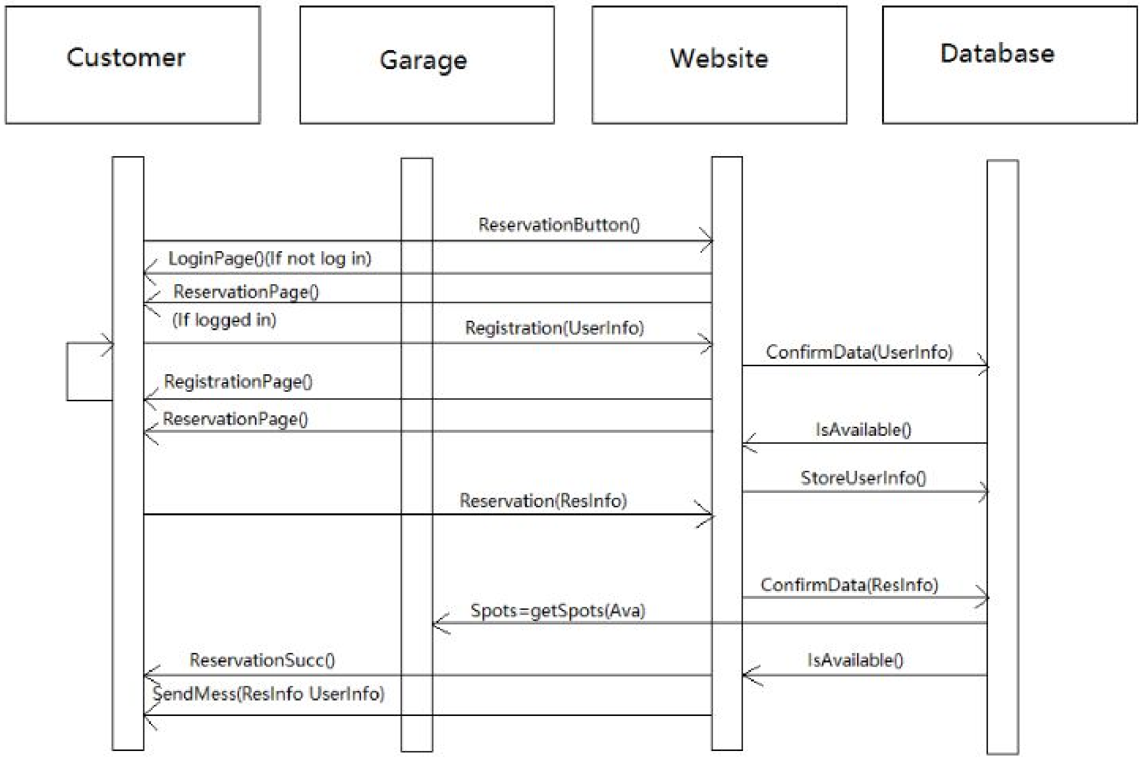
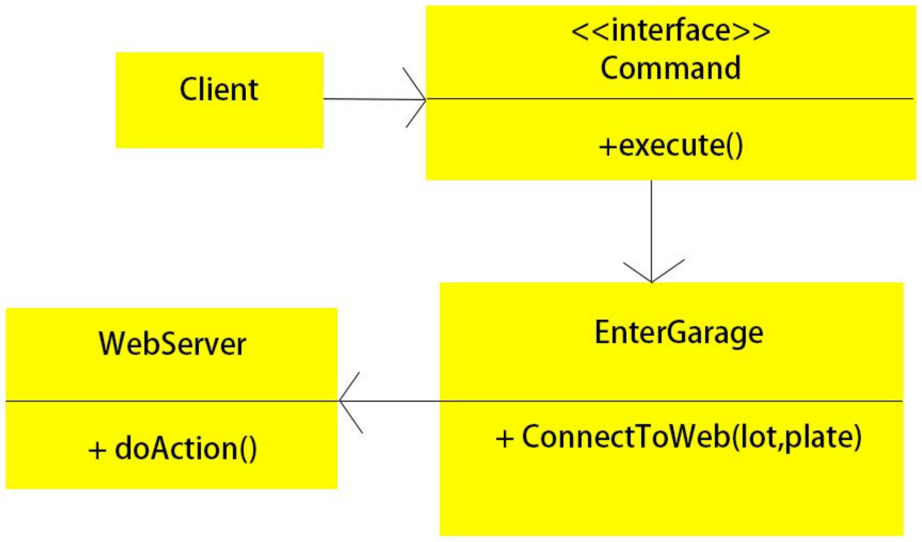


Figure 2: Sequence Diagrams for Use Case 2.

Use Case 3: Entering the garage

The reason we pick it as our fully dressed use case is because of its significance. This is the use case that providing basic functions as parking garage, while also very different from regular parking lots. The goal of our design (garage service) should be intuitive, fast and convenient. Therefore, it should be responsible for the basic parking function for unregistered customers, and the intuitive parking experience for our reserved customers.

For use case 3, The user drives to the gate of parking lot. The the License Plate Recognize System Scan the car plate. System first check customer if they have reservation. If they have reservation, the screen will directly guide him to his reserved parking lot and Gate will open. If the customer does not have reservation, system will connect to database to search if any available spots. Database will feedback the searching results. If garage still have available parking spot, the screen will show customer a unique QR code for registration, and guide the customer to available spots. Else, parking spot is insufficient, gate will not open and the customer has to leave garage. After customer successfully park their car, system will mark the spot as occupied and update information on database.



This design utilizes command design pattern. Our client, Android APP, only ask the web server to log car information when an car is entering the garage. In other words, our Android APP only decides timing of when a function should be run. The reason of why we use this design pattern is that the web server developers are different developers from software development team.

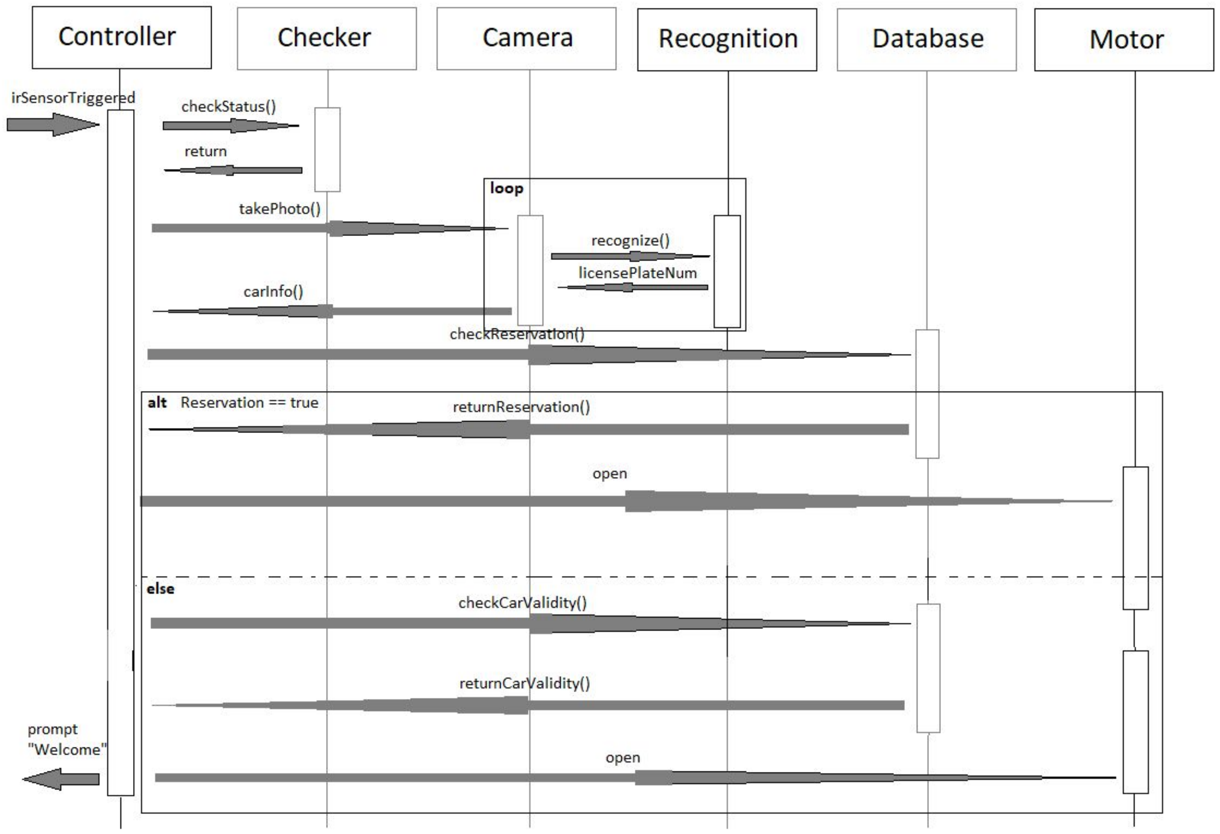


Figure 3: Sequence Diagrams for Use Case 3: Entering the garage. (**Pub-Sub diagram**)

Use Case 4: Online Payment

Our principle of designing this garage is to save as much time for the customers as possible. The automatic payment method is how we are going to achieve that. For registered customer, the parking fee will be automatically deducted from their account. In this use case, the data base plays the most important role. Entering and leaving the garage will trigger the camera, sending a signal to the database to start the time counting. When the customer leaves the garage, certain fee will be deducted from their balance. The receipt and rates will be available. When it comes to payment, security is our first concern. All the payment information are stored in the database and can only be altered by the customer, and can only be accessed by the database. The manager cannot alter or change the payment information without the customer's permission.

For use case 4 Online Payment, the customer will trigger the camera when leaving. The camera will get the license plate information and check if the customer is registered. If not then it will notify the user to use a valid payment method. If the user is registered the system will make the payment for the customer automatically. If the payment is done successfully, the gate will open. If not it will notify the customer to use a valid payment method. After that, the database will update the customers' information.

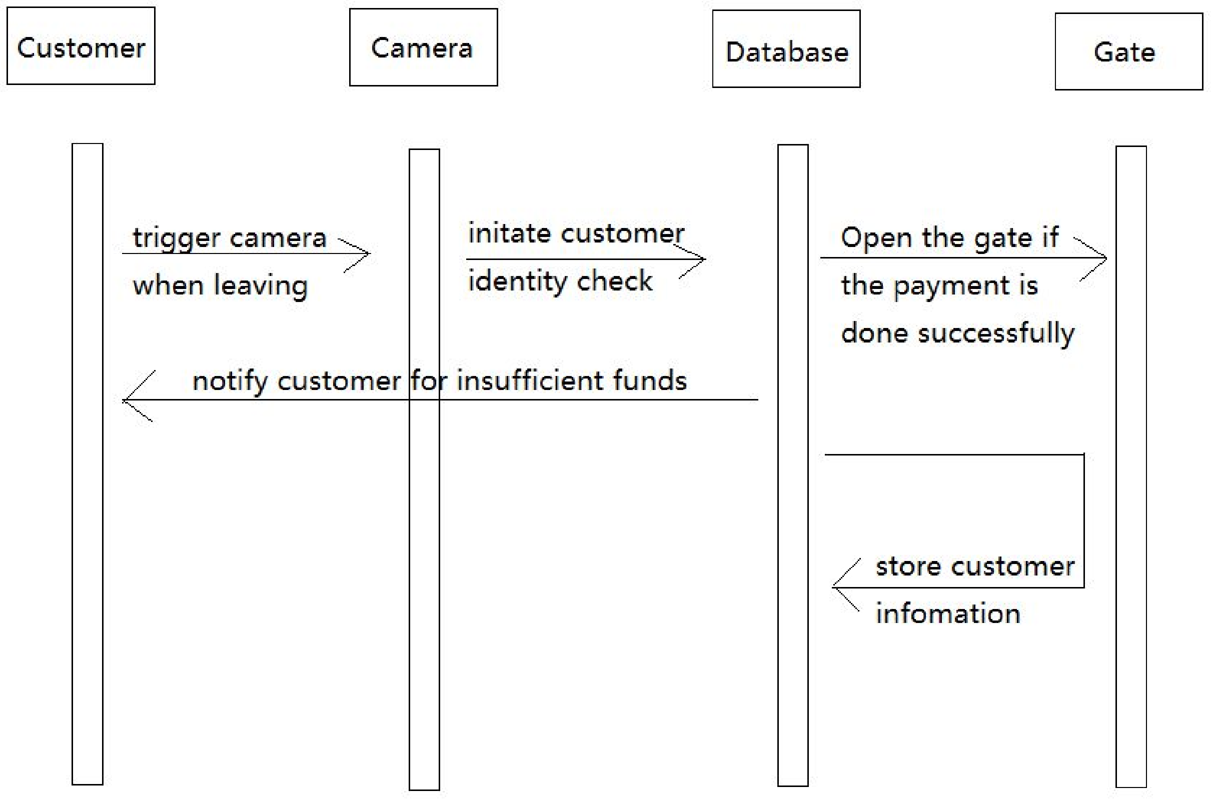
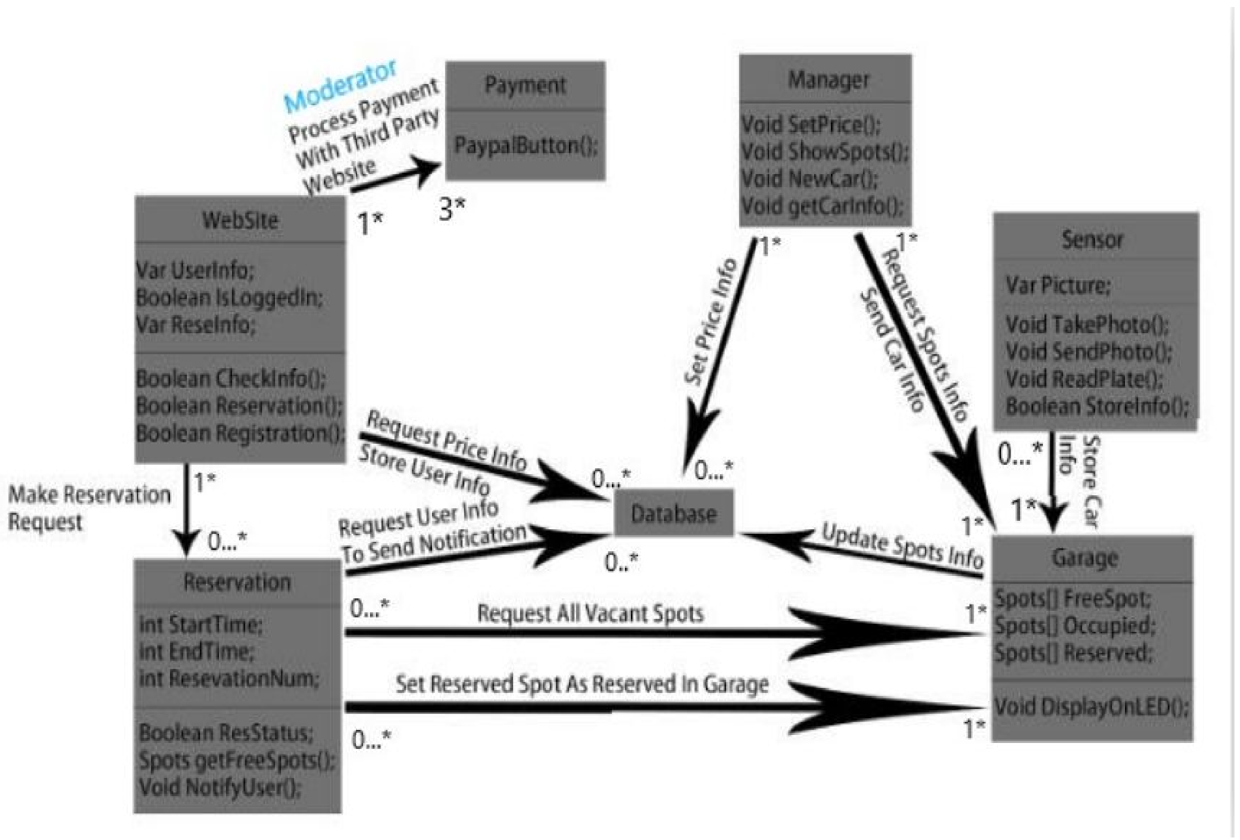

Customer
Camera
Database
Gate
trigger camera when leaving
initate customer identity check
Open the gate if the payment is done successfully
notify customer for insufficient funds
store customer infomation

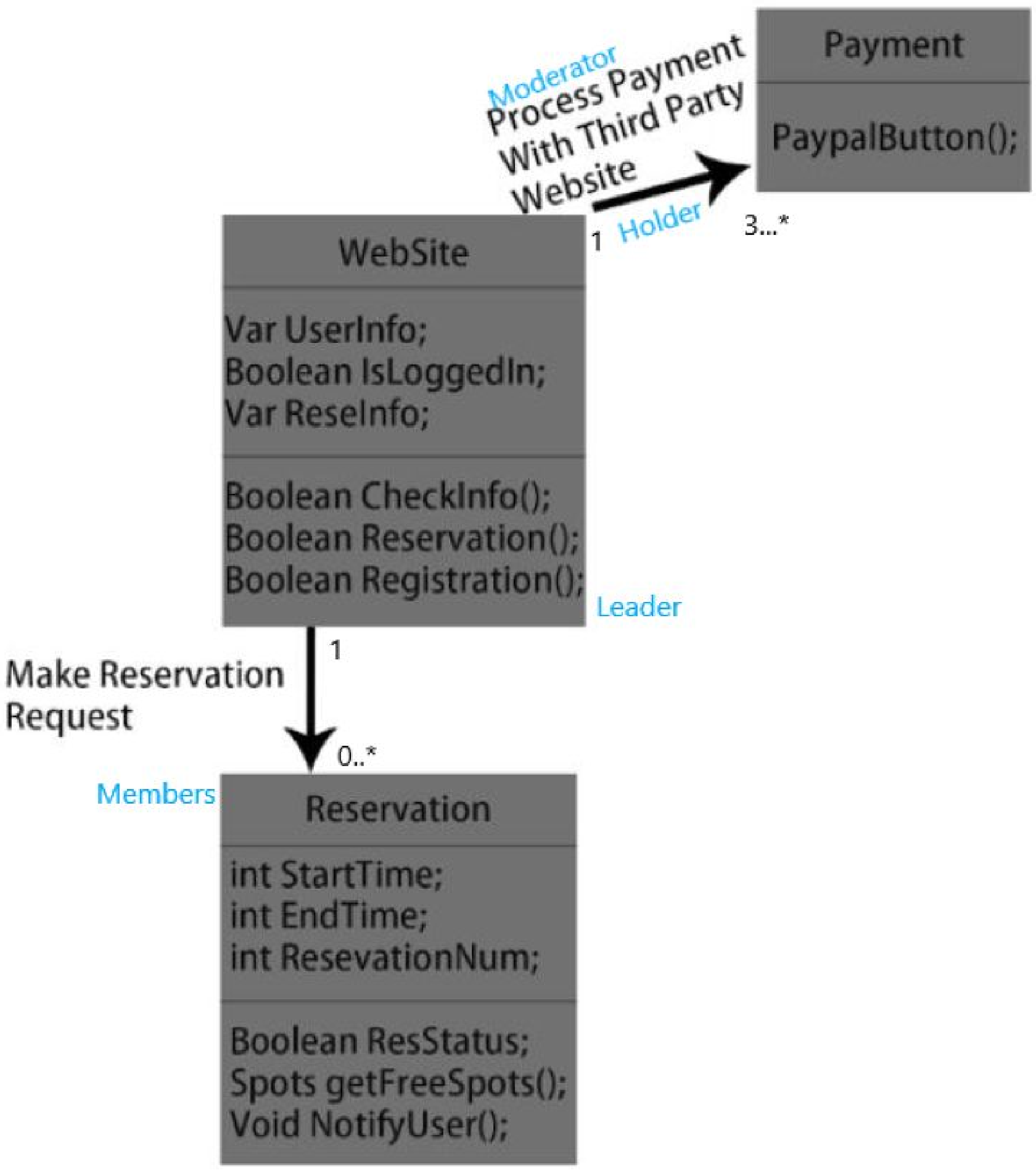
Figure 4: Sequence Diagrams for Use Case 4.

Section 8: Class Diagram and Interface Specification

a) Class Diagram

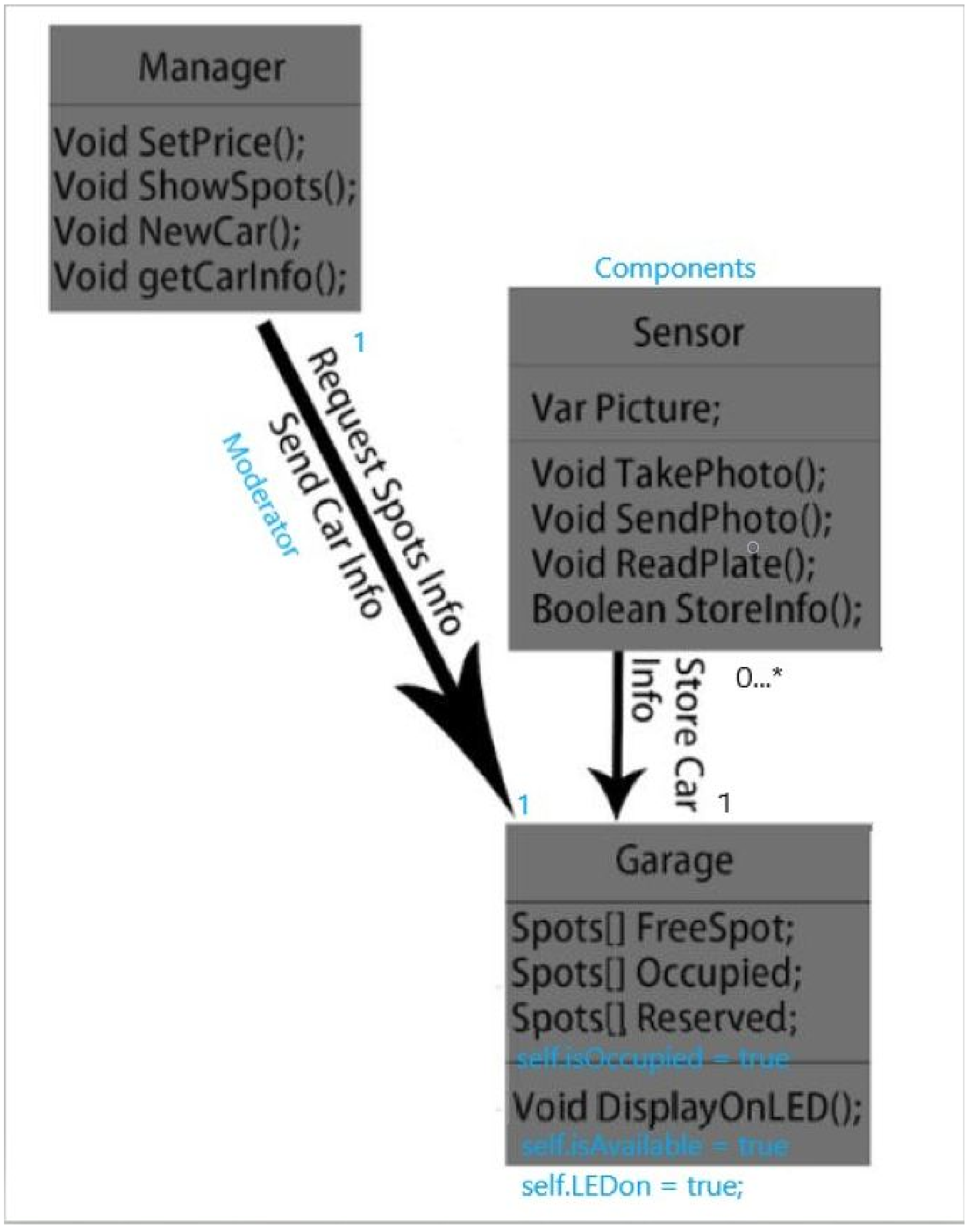


Subgroup #1: Payment, WebSite, & Reservation



This subgroup encompasses the frontend features of our system and defines our customer-facing functionality. The “WebSite” class interfaces with the “Payment” and “Reservation” classes, and allows customers access to the main features of our system.

Subgroup #2: Manager, Sensor, & Garage



This subgroup encompasses the backend features of our system and defines functionality available to administrators and parking lot owners. This group takes information from hardware to maintain and provide an updated representation of the parking lot.

b) Data Types and Operation Signatures

1. Manager

   (1) Attributes
      (a) String email: customer's email address for register.
      (b) String first_name: customer's first name
      (c) String last_name: customer's last name
      (d) String phone_number: customer's phone number
      (e) String account_number: customer's account number for login
      (f) String password: customer's customized password
      (g) Int customer_age: customer's age
      (h) String creditcard_number: customer's credit card number
      (i) String expire_date: expire date of customer's credit card
      (j) String CVV: three digits of CVV code from customer's credit card

   (2) Operation:
      (a) Register(): Creates a new user account and put the information into the database.
      (b) addCar(): Creates a car linked to an user account and put car's information into the database.
      (c) addCreditCard(): Creates a credit card number linked to an user account and put card's information into the database.
      (d) Reserve(): Creates a reservation linked to an user account and put reservation's information into the database.

The manager should have control over all the accounts and their information.

2.Camera and sensor

   (1) Attributes
      (a) Var picture: picture of customer's vehicle when the vehicle is entering the garage.
      (b) Boolean occupied: Tell whether a spot is occupied or not

   (2) Operation:

(a) CheckVehicleInfo(): check the vehicle's information with the the owner's information from database

(b) ScanVehiclePlate(): Scan the vehicle plate with picture photoed by the camera

(c) SenseOccupied(): Sense whether a car is on a spot

The camera sensor system should be able to recognize cars and determine if spaces have been occupied.

3.Timing System

(1) Attributes:

(a) Int total_parking_time: total time after a vehicle entering the garage

(2) Operation:

(a) Counting_time(): count the time after a vehicle entering the garage

The timing system should record how long a user has been in the garage based on their enter and exit times.

4.Ad-hoc Parking

(1) Attributes:

(a) String Vehicle_Plate: ad_hoc's vehicle plate

(b) Int spot_Num: the spot number of the customer's vehicle parked

(2) Operation:

(a) SelectParkingSpot(): select the parking spot with spot id and spot number.

Allow walk in customers to use the parking lot.

5.Reserved Parking

(1) Attributes:

(a) Int reservation_start_time: the start time of the reservation
(b) Int reservation_end_time: the end time of the reservation
(c) Int reservation_num: the reservation number generated when the customer make a reservation
(d) Int spotNum: spot number of the spot reserved by the customer
(e) String user_account: customer's account number

(2) Operation:

(a) confirmReservation(): send successful reservation notification
(b) pick_time(): let the customers to pick the time of reservation

Allow users to reserve a spot in the garage.

6.Garage

(1) Attributes:

(a) Int num_of_vacant_spots: number of vacant spots
(b) Int num_of_occupied_spots: number of occupied spots
(c) Int num_of_reserved_spots: number of reserved spots

(2) Operation:

(a) Direct_Vehicle(): direct the entering vehicles to their optimized spots.

The garage should keep track of the status of each spot to help users find an empty spot.

7.Website

(1) Attributes:

(a) String Username : used by a customer to login to the website.
(b) String Password : customer's account password linked with account number in the database
(c) Integer AccountID : Uniquely identifies a customer using a number.
(d) String SessionID : Identifies a customer's login session.
(e) String AccountType : Determines an account's privileges.

(f) String EmailAddress : The customer's email address.
(g) String HomeAddress : The customer's mailing address.
(h) String LicensePlateNum : The customer's car's license plate number.

(2) Operation:

(a) send_information(): send customer's information to database
(b) receive_information(): receive information from database

The website will allow users to create an account and use it to look at their parking information.

c) Traceability Matrix

1. Database

(a) Responsibilities

(i) Store customer and reservation information

(ii) Store price information

(iii) Store parking spots information

2. Payment

(a) Responsibilities

(i) Process payment via a third party API

(b) Classes

(i) PayPalButton();

3. Manager

(a) Responsibility

(i) Set parking price

(ii) Add new car into database

(iii) Show all spots and car information to administrator

(b) Classes

(i) SetPrice();

(ii) ShowSpots();

(iii) NewCar();

(iv) getCarInfo();

4. Website

(a) Responsibilities
(i) Acquire customer information
(ii) Acquire reservation information
(iii) Showing garage information

(b) Classes
(i) requestPriceInfo()
(ii) storeUserInfo()
(iii) processPayment()
(iv) makeReservation()

5. Sensor

(a) Responsibilities
(i) Take photo when pressure sensor is triggered
(ii) Send photo to manager software
(iii) Send photo to car plate reader

(b) Classes
(i) TakePhoto();
(ii) SendPhoto();
(iii) ReadPlate();

(iv) StoreInfo();

6. Garage

(a) Responsibility
   (i) Show all essential information on phone screen
(b) Classes
   (i) DisplayOnScreen();

7. Reservation

(a) Responsibility
   (i) Retrieve all available spots from Database
   (ii) Send notification to user regarding reservation status
(b) Classes
   (i) getFreeSpots();
   (ii) NotifyUser();

	Domain Concepts						
Classes	Database	Payment	Manager	Website	Sensor	Garage	Reservation
Customer Registration	✔	✔		✔			
Customer Login/out	✔			✔			
Garage Gate	✔				✔	✔	✔
License Plate Scanner	✔				✔	✔	✔
Notification System	✔		✔				
Security System	✔		✔		✔	✔	
Spots sensor	✔				✔	✔	

Database Managing	✔		✔			✔	
Database Connection	✔	✔	✔	✔	✔	✔	✔

Section 9: System Architecture

a) Architectural Styles

The Inspired Auto Parking System will follow a Master-Slave pattern. This pattern consists of two parties: the master and the slaves.The master component distributes work to each slave component and slave components return results to the master component. The Backend management system is our Master component. Every simple task distributed by the Backend management system will be handled by each slave component (like the camera, parking sensor). The Master will manage and distribute tasks to the slaves and the slaves send feedback results. There will be a lot of independent features which will be accomplished simultaneously so the bus is very important. The bus will connect each slave component to the master component. As the data sharing and component connection, the bandwidth of the bus is critical. All data (including customer's information, account info, parking status) are stored on our SHA-256 encrypted database on our server. So our server is our Master and all other sensors or monitors etc. are slave components.

Our service lives on the Internet which can be accessed through Inspire Auto Parking's website. Parking lot owners can register as administrators by providing parking lot proof-of-ownership and a valid key to access the backend management toolset through the website or application. Therefore, our working style is a Service-Oriented Architecture (SOA).

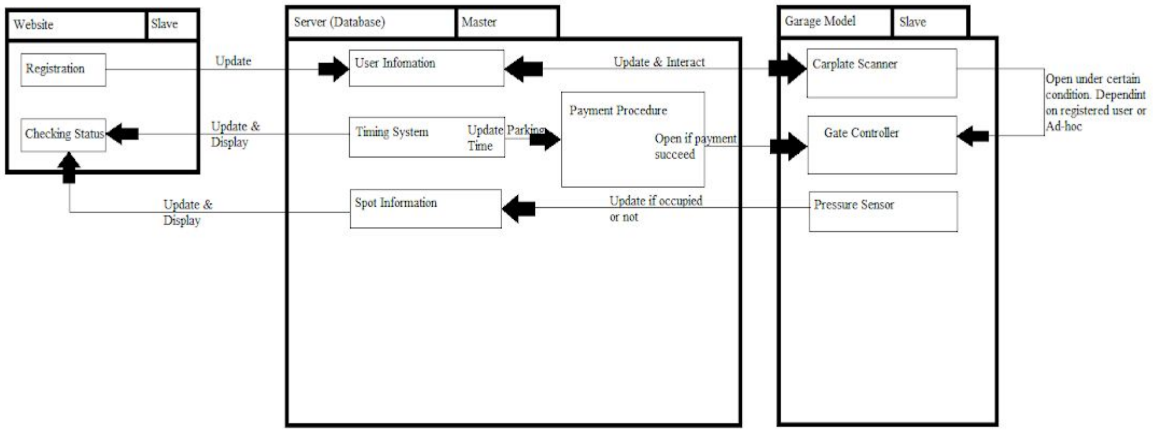
To reduce energy using and cost, server will take charge primary component functional. Even though make every slave component solve some simple task will reduce server load but this will increase cost. Our goal is to be economic. So we choose to use algorithm to optimize stability and reliability.

b) Identifying Subsystems

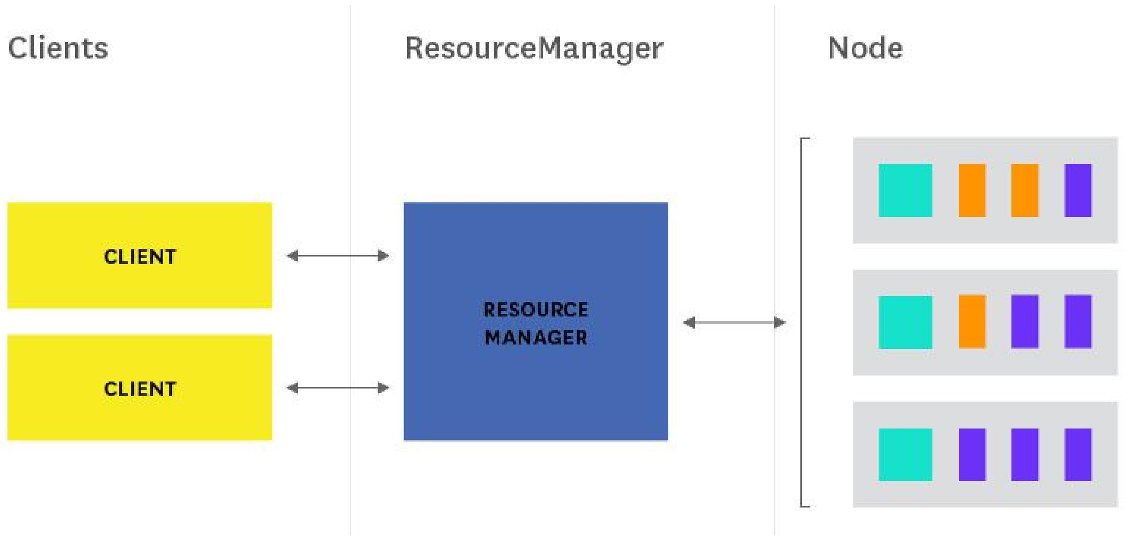
In our "Master-Slave" structure, there will be three main subsystems: The server (database); the website (User interface) and the hardwares (the garage demo model).

The "Master" in this structure is the server, both the website and the hardware will take order and report to the servers.

c) Mapping Subsystems to Hardware



The master server should be able to collect data correctly from multiple devices. The hardware includes a car plate scanner and a pressure sensor. Beside those hardware devices, the server should also be able to interact with the website, providing the desired and correct data to the website for the user to check their status.



d) Persistent Data Storage

A database will be used to store the information of the users and spots in our system. The users' payment/contact/car information will be stored. Also, the current status of spots will be always updated and stored for the reservation. Moreover, the history of spots will be recorded(Time,Occupied/Reserved or not,User,Payment)

e) Network Protocol

We decided to use TCP/IP as our network Protocol because TCP/IP is wide-used mature protocol. Our server will use ECS which is safer and convenient and costless. Our website will obey TCP/IP protocol which is a universal Internet protocol so that different computer could access to our website. Our website will be a simple HTML page which will accomplish several "manage" behavior.

f) Global Control Flow

Execution orderliness:

The system is both event driven and procedure driven. The sensor is waiting for the "Car Enter" event to wake the sleeping camera to take a picture and recognize the license plate which means the system is event driven. Moreover, the system is procedure driven because after recognizing the license plate, the system will check the users' information and decide the next step.

Time dependency:

The system has a reservation and payment feature. Therefore, we need a timer to determine the cost and update the status of the spots. Also, the system will notify the user based on real time to make sure they will not park overtime unintentionally.

Concurrency:

All parked cars share the same server. All updates and commands transfers simultaneously. Our service is completely web-based. Any "threads", if any, are handled by Apache, MySQL, PHP, or any other server-side services that already exist and not by the developer.

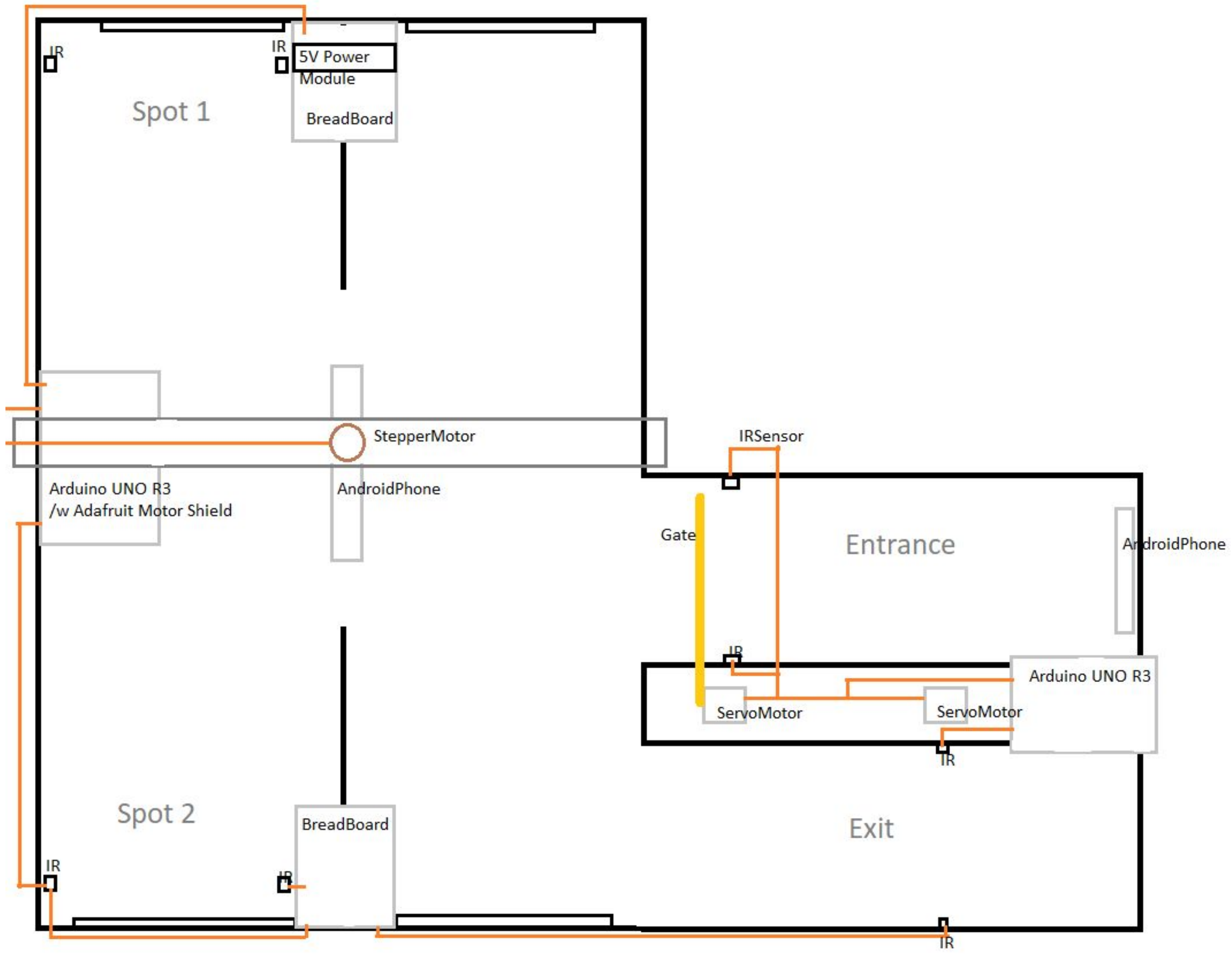
g) Hardware Requirements

Currently, our system has 4 servo motors, 1 step motor, 2 IR sensor, 1 pressure sensor. The pressure sensor is used to detect whether there is a car in front of the garage gate, and we used two servo motors to build a rotation machine to control camera

Android phone’s camera, used for capturing license plate picture

Micro Servo, used for control camera rotation and the pole at the entrance

IR Sensor, used for identifying spots occupation and indicating the camera to take photo



1. Website Host Server (Planning to use ECS)
    - 1GB RAM
    - 10GB Disk Space
    - Firewall Applicable
    - 100Mbps Connection Speed
2. Database
    - 4GB RAM
    - 64GB Disk Space
    - 1Mbps Wired Connection through USB2.0+
3. Garage Hardware
    - LattePanda
    - Arduino UNO R3
    - Force Sensitive Resistor Square
    - ArduCAM Mini 5MP OV5642PLUS
    - IR Break Beam Sensor - 5mm LEDs
    - Micro Servo - High Powered, High Torque Metal Gear
    - Adafruit Motor/Stepper/Servo Shield for Arduino v2 Kit

Section 10: Algorithms and Data Structures

Algorithms

In order to implement Use Case #3, "Entering the Garage", we must select a shortest-path algorithm that allows use to quickly determine the most efficient path from the entrance to an available space. Each parking lot will be modeled by an undirected graph, where each edge begins with a certain weight depending on the length of the path segment it represents. These weights may be modified accordingly to reduce congestion as traffic increases. Since the graph contains rings, we will implement Dijkstra's algorithm, which deals with rings sufficiently well. Since our edge weights can mutate as traffic changes, we cannot use a precomputed table of paths for each parking space.

Data Structures

Our system uses a multidimensional array to represent each space. We chose to use a multidimensional array because parking lots are often spatially organized in a grid-like manner. In the case that a parking lot is more unconventionally designed (e.g. it is shaped like an L or like a circle) we believe that the speed advantage of accessing an array outweighs the losses incurred by the space wasted by unused entries in the multidimensional array.

Our system utilizes an undirected graph to represent the paths that can be taken around the parking lot. We chose this representation over alternatives such as trees because a graph is more versatile and easily lends itself to representing multidimensional information. We chose to make the graph undirected, since this reflects that fact that a car could be moving in either direction along a particular path segment.

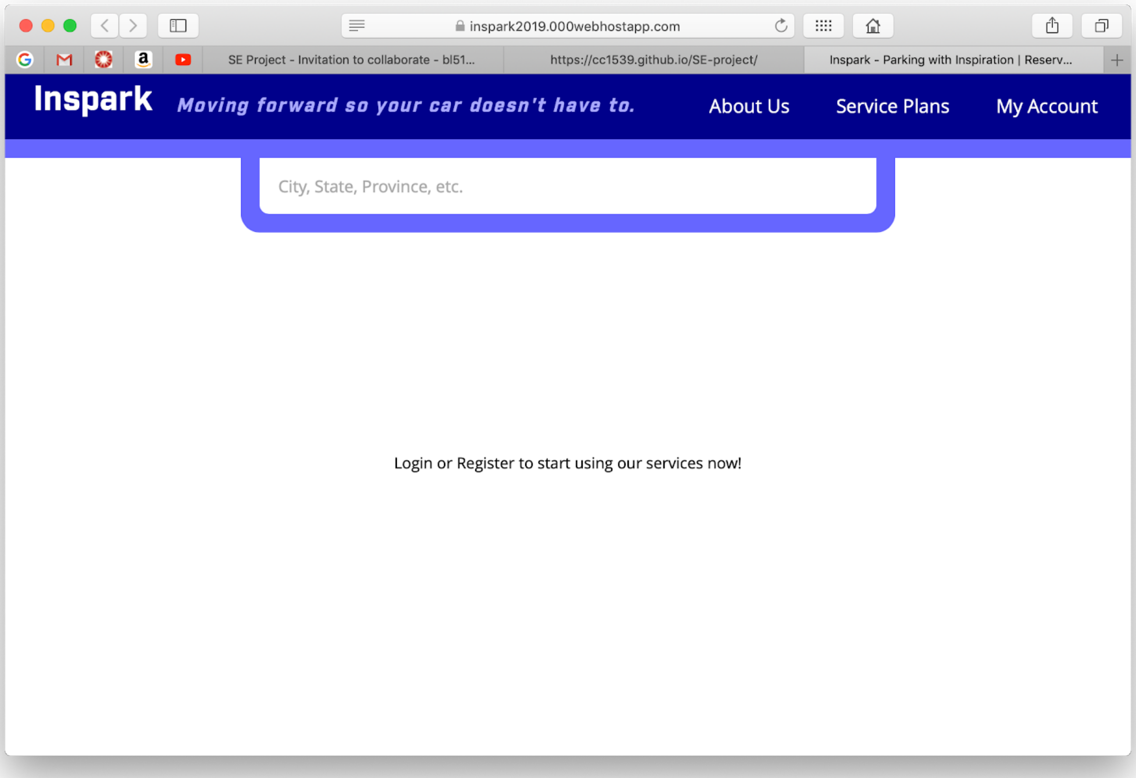
Section 11: User Interface Design and Implementation

Our initial user interface designs depict a website composed of various separate pages, each reserved for their own special function. Our current design and implementation instead consists of an easily accessible page that itself offers access to the full functionality of our service to the customer. This reduces the amount of clicks taken to access a certain feature and the time taken to become adjusted to the structure of a website, thus decreasing overall user effort. To organize our features and help customers compartmentalize them in their minds, we modularize our user page in the form of "tabs" that help a customer quickly access the desired feature.

Our website is designed to be highly streamlined and efficiency-oriented. We take pointers from what seems to be a modern trend of minimalist UIs, which are defined by simple blocks of solid color and only the occasional outline. This increases the "ease-of-use" of our website and further reduces user effort.

UC-1 Register

Homepage:

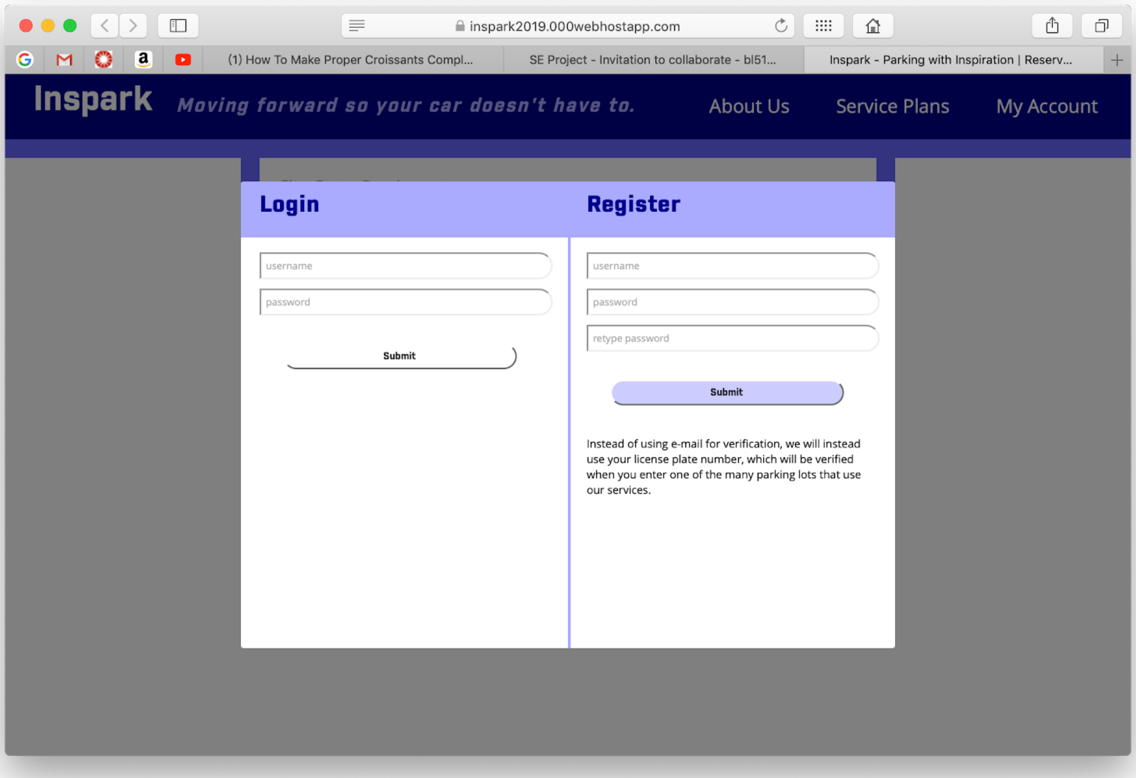

inspark2019.000webhostapp.com
SE Project - Invitation to collaborate - bl51...
https://cc1539.github.io/SE-project/
Inspark - Parking with Inspiration | Reserv...
Inspark
Moving forward so your car doesn't have to.
About Us
Service Plans
My Account
City, State, Province, etc.

Login or Register to start using our services now!

On the homepage, the customer can choose to use different functions. The "About us" and "Service Plan" button leads to lower part of the homepage. And if the customer wants to log in, they can click on the "My Account" button.

On the home page, in the worst case scenario, the user have to click 1 time.

After clicking on the "My Account" button, a pop up window will show up:

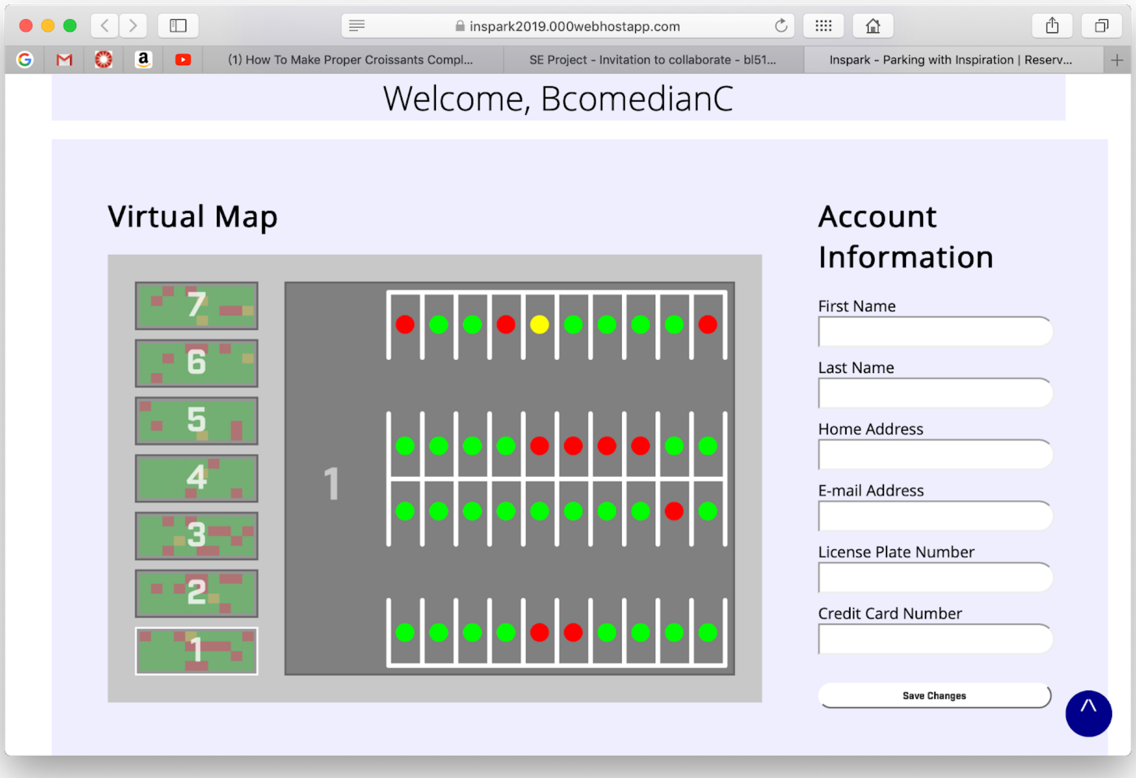

inspark2019.000webhostapp.com
(1) How To Make Proper Croissants Compl...
SE Project - Invitation to collaborate - bl51...
Inspark - Parking with Inspiration | Reserv...
Inspark
Moving forward so your car doesn't have to.
About Us
Service Plans
My Account
Login
username
password
Submit
Register
username
password
retype password
Submit

Instead of using e-mail for verification, we will instead use your license plate number, which will be verified when you enter one of the many parking lots that use our services.

The customer can choose to login or register for those unregistered customers.

Worst case: 1 click, keystrokes depending on the customer's choice of username and password.

After logging in:


inspark2019.000webhostapp.com
(1) How To Make Proper Croissants Compl...
SE Project - Invitation to collaborate - bl51...
Inspark - Parking with Inspiration | Reserv...
Welcome, BcomedianC
Virtual Map
7
6
5
4
3
2
1
1
Account Information
First Name
Last Name
Home Address
E-mail Address
License Plate Number
Credit Card Number
Save Changes
^

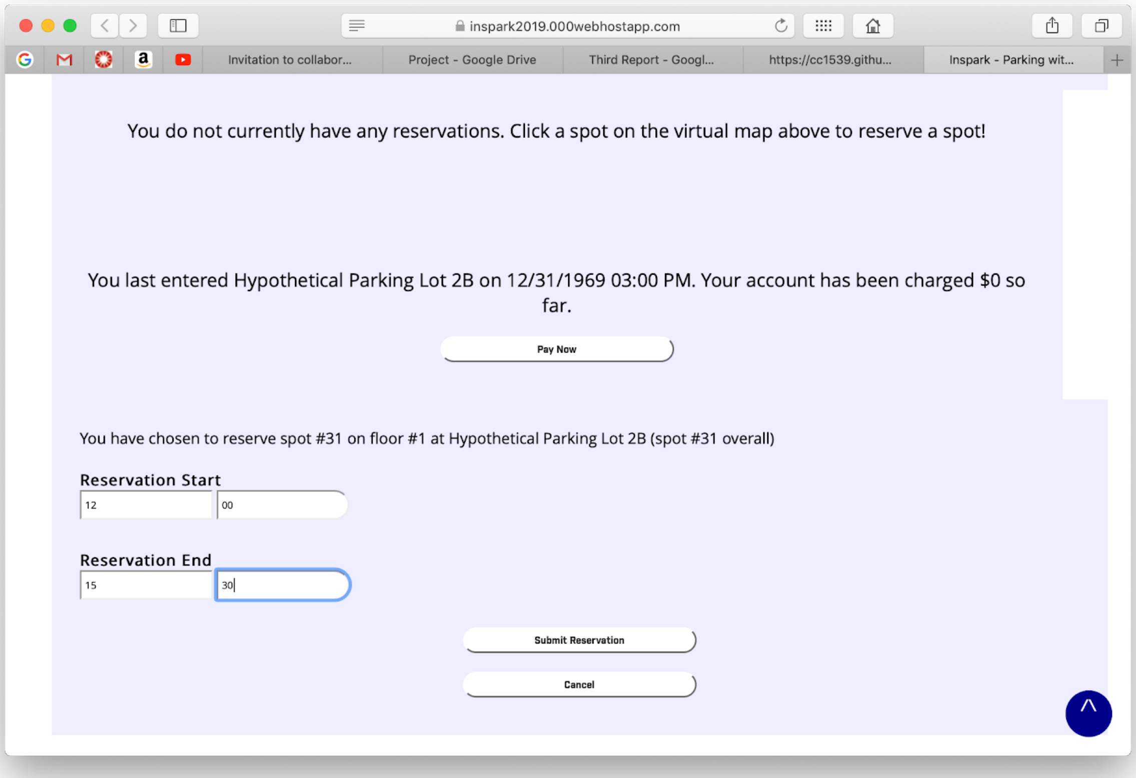
At this page, the customer can fill up their user information and click "Save Change" button after done typing.

The worst case scenario: 1 click and keystrokes based on the user's information.

<u>UC2- Reservation</u>

After logging in, the user will be displayed a virtual map of the whole garage, where they can click on vacant spots (green spots) to reserve.

Worst case scenario: 1 click; 0 keystrokes.



You do not currently have any reservations. Click a spot on the virtual map above to reserve a spot!

You last entered Hypothetical Parking Lot 2B on 12/31/1969 03:00 PM. Your account has been charged $0 so far.

You have chosen to reserve spot #31 on floor #1 at Hypothetical Parking Lot 2B (spot #31 overall)

Reservation Start
12 | 00

Reservation End
15 | 30

After clicking on the spot, the customer will be lead to this page. At this page the user can type in the time slot. After typing the time, click on Submit. If customers wish to cancel the reservation, they can click on the cancel button.

Worst case sceneral: 1 click, 8 keystrokes to type in the time.

Section 12: Design of Tests

For our first demo, we will only implement and test the use cases most critical to the functioning of our service, which happen to be our first five use cases:

UC#1 - Registration

A user should be able to register for an account by providing a unique username, password, and a valid e-mail address. If an account already exists with the given username or e-mail address, registration should fail. Otherwise, the account is created and the user should be able to login using the same credentials.

Test coverage: For the user name, we will proceed this test with three different inputs which are new username, duplicate user name, and invalid user name. For the password, we will proceed this test with two kinds of inputs which are valid password, and invalid password. For email address, we will proceed this test with the 29 test cases provided by the cited website.

UC#2 - Reservation

A user should be able to submit a reservation for a particular parking lot. An ordinary user should only be able to reserve one spot ahead of time; should they attempt to reserve a different spot, they will be asked whether or not they want replace their last reservation with their current one.

Test coverage: The accuracy of the reservation time and user name will be covered. Also, users cannot reserve a spot already be reserved at the same period. Moreover, the old reservation will be cancelled if the user successfully reserve another spot.

UC#3 - Entering the Garage

A user should be able to enter the garage when there is available spot in the garage or the user has a valid reservation for that specific period of time. The garage door should be opened when the user's vehicle is stopped at specific spot for scanner to run validity test, and the user's vehicle passes the validity test. If the user's vehicle did not pass the validity test, the garage gate should be remain closed. The screen should display the QR code after the vehicle passed the validity test.

Test coverage: We will simulate a small parking lot with four parking spots, and 1 car model. For the entering garage, the car should orient to a different angle from the camera. There are three situation we have considered. The first situation is when the garage is empty. The second situation is when there is a car with no reservation entering a full garage. The third situation is where there is a car with reservation entering a full occupied garage.

UC#4 - Status Checking

A user should be able to see his/her parking status, including the current parking time, parking spots and parking fees. If the user is currently not parking, then the user should see nothing with status checking. When the user leaves the spot, the sensor should notify the database and change the status checking site with updated information.

Test coverage: Parking status, parking time parking spot and fees of customer will be tested through data change in database. Status check interface will be tested through different customer accounts. Also, the parking sensor will be tested using car models.

UC#5 - Online Payment

Users should be able to set up online payment options for their accounts. User should be able to choose between Paypal and Credit Card for online payment. If the user don't want to link any account, they should be able to make deposits to their balance as online payment funds.

Test coverage: The amount of the funds should be accurate after payment finished or successfully recharged. The online payment should not charge the same order twice. Also, the payment should be done for the user whose account the charge is tied to.

7.3 Integration testing strategy We will act as a normal user, and follow the procedures below.

Test Case ID	Test Case Objective	Test Case Description	Expected Result
1	Register an account	Enter User name and Password to create an account.	For Invalid input, the account will not be successfully created. And the user will be notified. For valid input, the account

			will be created successfully and Customer Information will be stored in database
2	Reservation	Select a period of time on the reservation website.	Get a notification regarding reservation from website.
3	Enter garage	Drive the car to the front gate of the garage and enter.	The gate should open once the car arrives the front gate with no other problem.
4	Status Checking	After parking at the reserved spot, user wants to check parking records on the phone.	The website will show the reservation information and a text message will be sent to the user’s phone.
5	Online Payment	Pay for current parking fee or add balance into account	Current parking fee successfully paid. Customer Status show paid and gate opened. If customer choose to add balance into account, then database should update the new balance and account should present the accurate balance.

Section 13: History of Work, Current Status, and Future Work

13.1 History of Work, and Current Status

History of Work:

For now, we have the user interface for the users to create an account and make a reservation on our website. For creating an account, the users need to enter their official names and their license plates. We also have a backend associated with the database to store the users' statistics and it allows the manager of the system to edit all the data stored in the database. For the hardware, we have a simple arduino model with a force sensor to detect whether there is a car entering our garage. When the sensor is triggered by the gravity of the car, the camera will take a photo of the car from the back in the same time. We have a implemented API from online can recognize the car model, car plate, car color, car make and district of the car. The system also put the data into the database and show them to the manager.

Current Status:

Every component of our project is individually working properly except the photo module. To fix this, our hardware team and software team is working on using smartphone camera to capture car photos. Software team has implemented all basic functionalities of backend software. Software team also fixed some minor bugs found from demo 1. Hardware team has 1 step motors, 3 servos motors, 4 IR sensors, and 1 distance sensor to implement automatically capturing the photo of car on parking spot.

The front-end website is currently capable of displaying a real-time virtual map that reflects the state of a particular parking lot. The same page also facilitates the submission of reservations and/or updated account information. In fact, all information and functionality available to a logged-in user is shown on a single page in a concise format. The hardware interfaces with certain PHP modules on the website's backend to continuously update the information shown on this page. It also sends information to a "qrcode" page that shows what the display at the entrance of a parking lot would look like.

The site has been updated for mobile and now displays correctly, which diminishes the need for a native app.

13.2 Future Work

In the future, we will focus on the test of the whole use case. We will make sure different users (registered/unregistered;reserved/unreserved;normal/electrical car) can have the whole parking experience from entering the garage to paying the bill and leave. In addition, we will improve our reservation. Users will be notified when they successfully make a reservation and payment. Also, they will be notified of their parking time.

Major Accomplishments

- We were able to connect our management software with the online car recognition API we were using, so that we can implement the “entering the garage” use case smoothly.
- We are planning to set up camera that will monitor the whole garage model operated by IR sensor and electric motor.
- The website is informative and easy to use.

History of Work, Current Status, and Future Work
Instead of the section *Plan of Work* have the section *History of Work* which documents how the actual milestones and deadlines evolved. Compare these against the milestones as planned in Reports #1 and #2.
Also summarize (as a bulleted list) your key accomplishments in this project.
Discuss the possible directions for future work on this project.

By submitting the first report, it means we are almost done for the conceptual part. From now on, the major concern is how to accomplish all the requirements. All the work are assigned to subgroups so that we can work on different aspects simultaneously. There are four major parts: Database constructing, GUI design, Website and a fully functional garage model for DEMO.

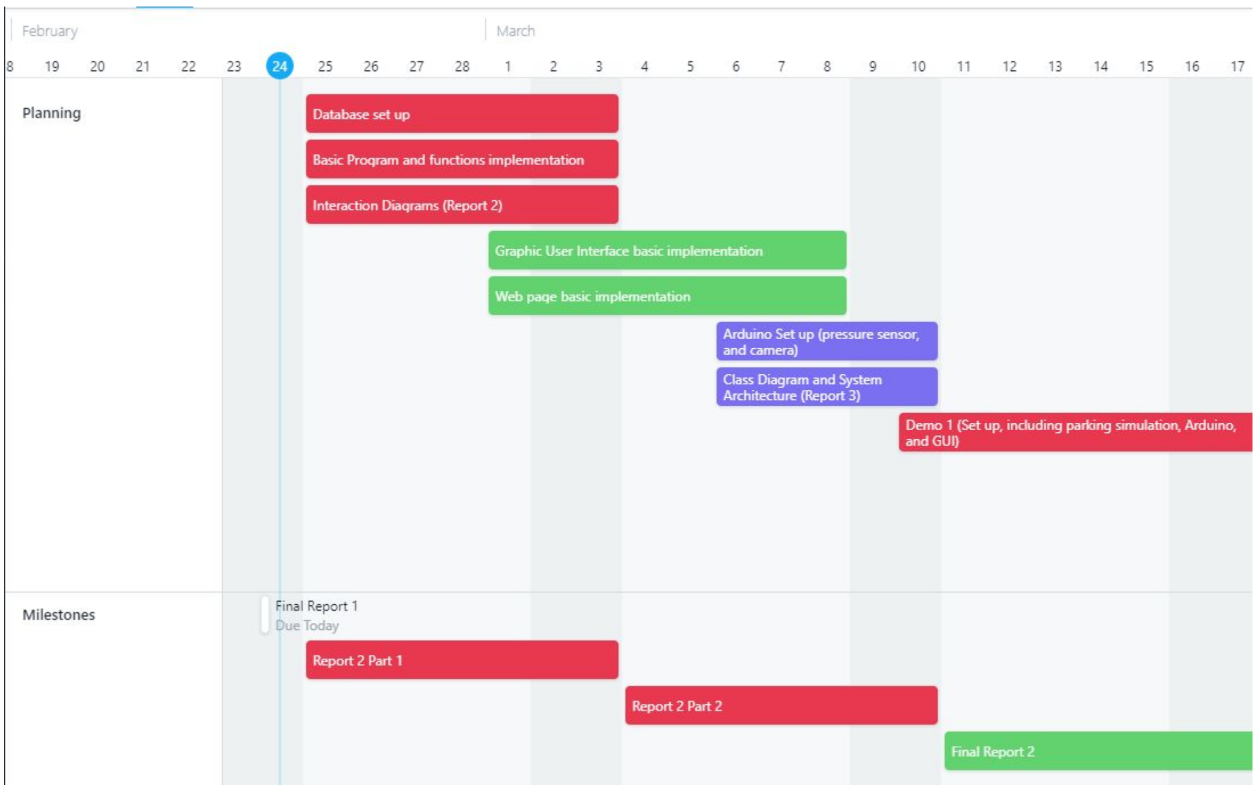
Database: Use MySQL to accomplish database.

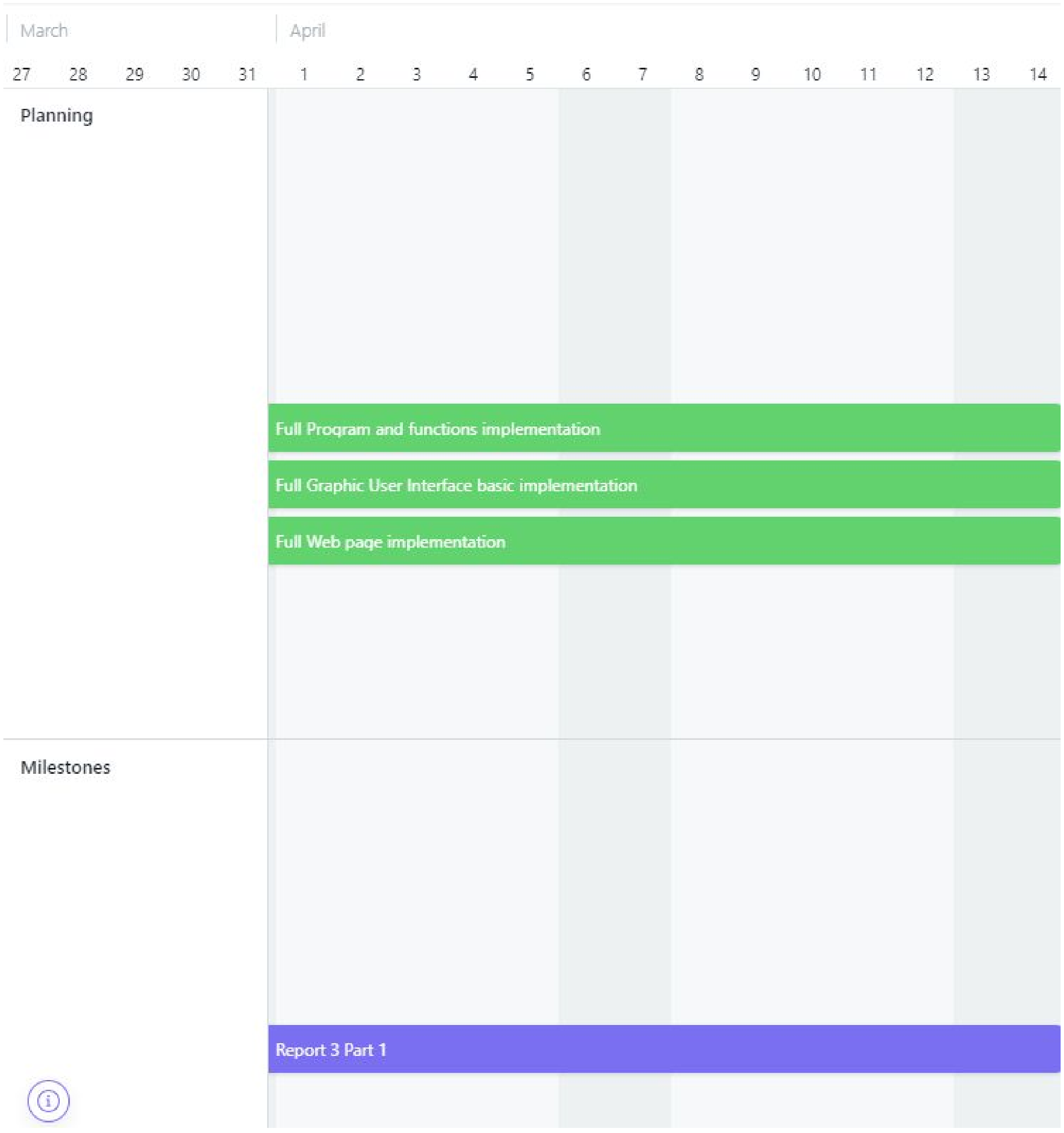
GUI design: Implement the previous design with JAVAFX (via. SceneBuilder).

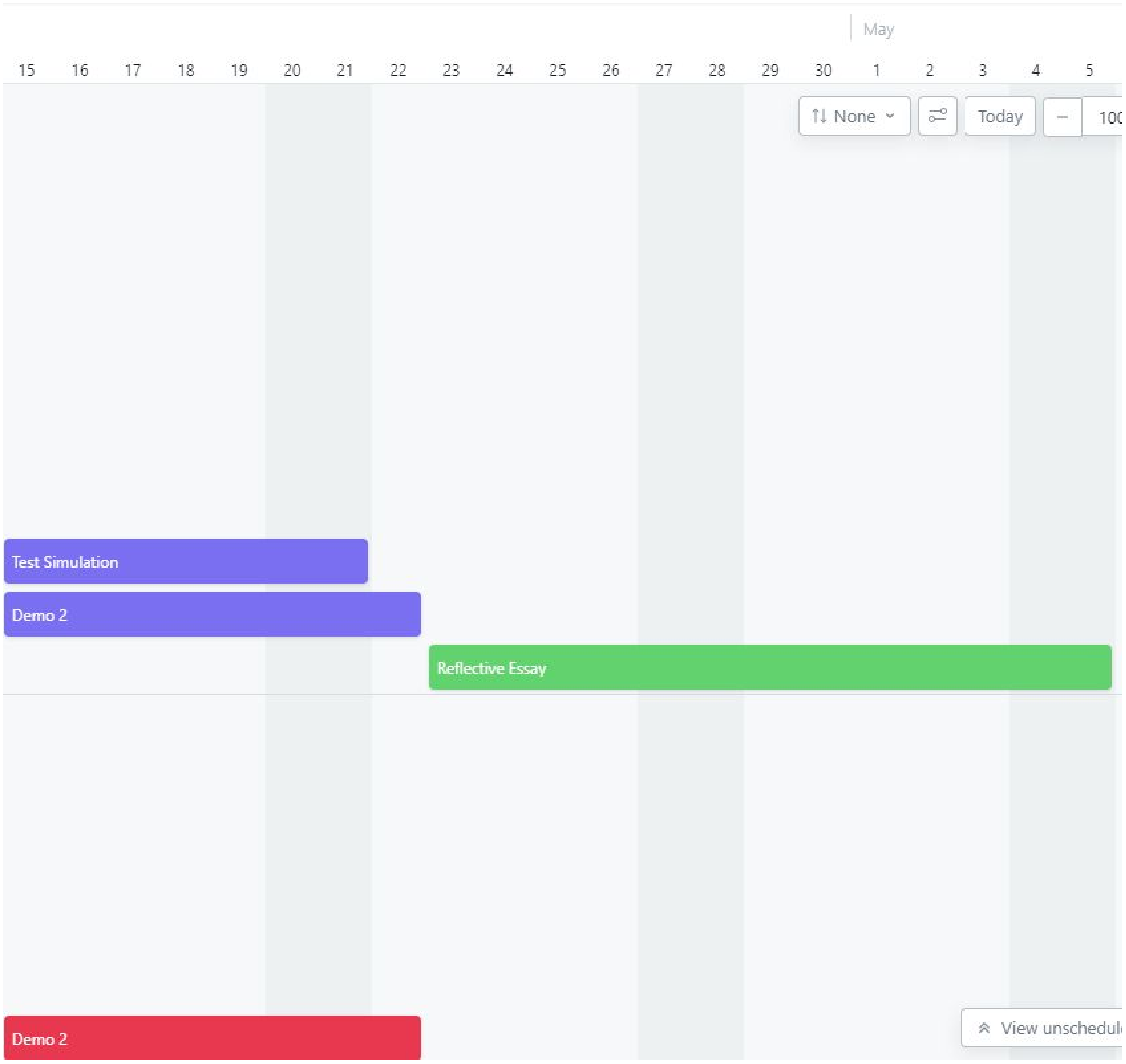
Website: Done with user interface designing and implementing.

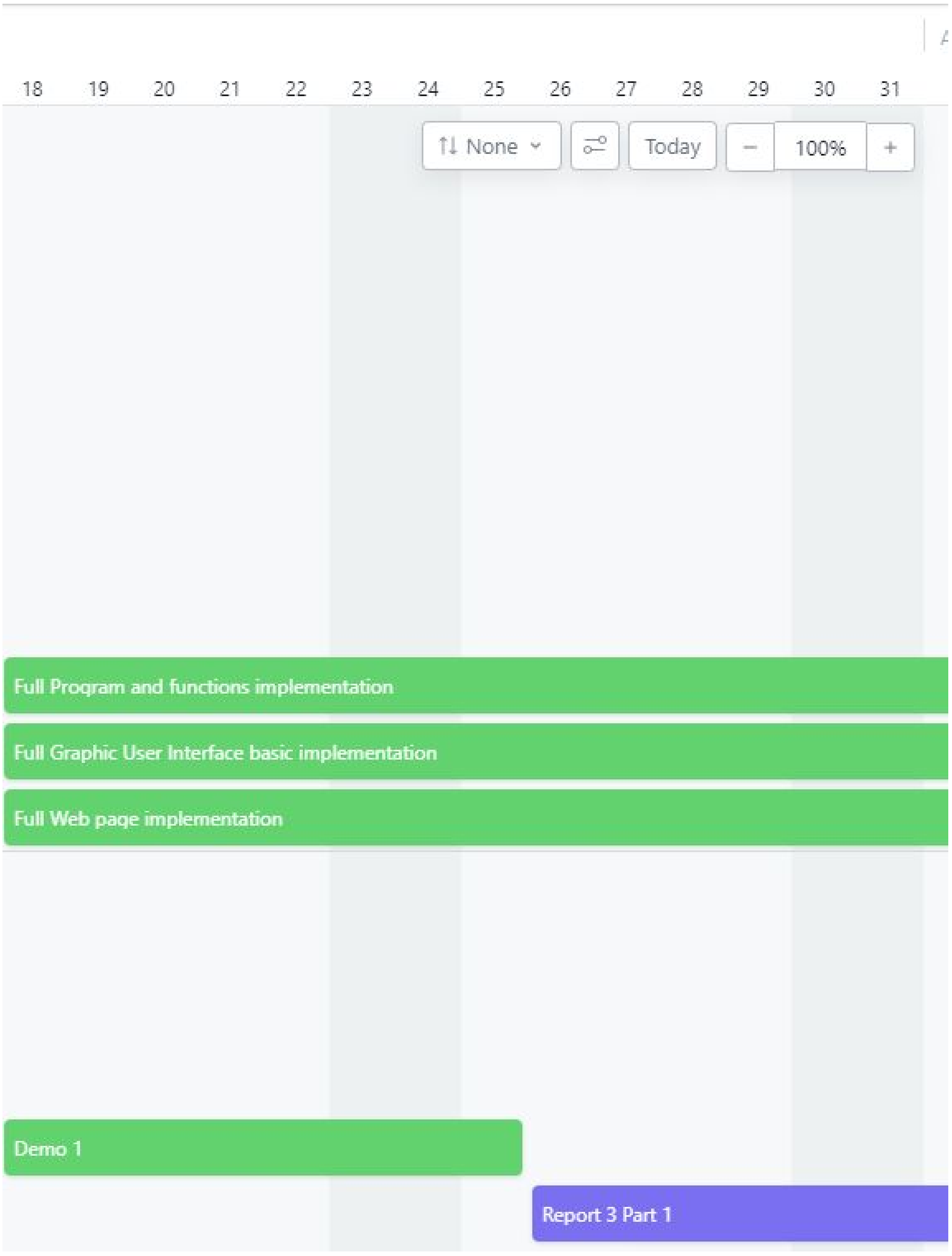
Model for DEMO: Fully functional sensor and camera setup.

After achieving the short term goal, we will start to focusing on the connection and compatibility between each parts. Improve the compatibility as well as the design and functionality of each part. We’ll check the progress every week and set the goal for next week. For the demo #2, we’ll show several whole use cases walkthrough.


February
March
8
19
20
21
22
23
24
25
26
27
28
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
Planning
Database set up
Basic Program and functions implementation
Interaction Diagrams (Report 2)
Graphic User Interface basic implementation
Web page basic implementation
Arduino Set up (pressure sensor, and camera)
Class Diagram and System Architecture (Report 3)
Demo 1 (Set up, including parking simulation, Arduino, and GUI)
Milestones
Final Report 1
Due Today
Report 2 Part 1
Report 2 Part 2
Final Report 2


March
April
27
28
29
30
31
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Planning
Full Program and functions implementation
Full Graphic User Interface basic implementation
Full Web page implementation
Milestones
Report 3 Part 1


May
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
1
2
3
4
5
None
Today
100
Test Simulation
Demo 2
Reflective Essay
Demo 2
View unschedul


18 19 20 21 22 23 24 25 26 27 28 29 30 31
None
Today
100%
Full Program and functions implementation
Full Graphic User Interface basic implementation
Full Web page implementation
Demo 1
Report 3 Part 1

Project Management

Subgroup #1 - (Software & Communication)

Members:
- Christopher Cheng
- Samuel Cho

Contributed Functionality:
- Parking Spot Reservation
- Parking Lot Visualization

Contributed Qualitative Property
- Develop clean and easy-to-use online UI

Subgroup #2 - (Direction and Account System)

Members:
- Zhuohuan Li
- Tingcong Jiang

Contributed Functionality:
- Account-User Information Binding
- Customer Account Registration

Subgroup #3 - (Hardware + Arduino)

Members:
- Buyuan Lin
- Yiran Tan

Contributed Functionality:
- Parking Lot Status Acquisition
- Parking Lot Management

Subgroup #4 - (Test & Development)

Members:
- Chenyu Cao
- Shijie Xu

Contributed Functionality:
- Online Payment
- Data Submission

Occasionally, a group meeting will be held to enable effective discussion between subgroups and allow team members to showcase their progress. For now, there is no fixed time or location for these meetings; they are simply organized when the majority of the team feels that one is needed, or would be convenient. When this happens, the time and place of the meeting is discussed on the GroupMe social platform and/or through one or more of Google’s services (e.g. Google Docs, GMail).

The general procedure for written reports begins with an initial draft wherein we generally disregard grammar and sentence structure in favor of getting critical details and information into the document as quickly as possible. This way, team members can spend more time brainstorming and improving each other’s ideas instead of getting stuck deliberating on what diction to use and how to word a particular sentence. Only after all team members agree that what’s written in the document properly expresses the state of our project do we then format everything into proper English.

Each subgroup consists of two team members who are capable of taking over for the other in the case that one, for any reason, is unable to finish his assigned contribution. In the case that both members from any one subgroup are unable to deliver, the subgroup in question should express this to the rest of the team as soon as possible through one or more of the previously established avenues of communication. One or more subgroups can then take over the duties of the defunct subgroup until it recovers.

First we need to procure the materials needed to initialize the project within the next two weeks. On the hardware side, the most practical piece of equipment that comes to mind is an Arduino board, which we can connect to various sensors (for example, distance sensors to determine vacancy), cameras, and displays. Two plate-reading cameras will be configured to log entry and exit times. On the software side, we are planning to implement a plate recognition algorithm and set up a database for the organization of user account information.

To test our system, we are planning to construct a simulation parking lot model. A program will simulate a virtual parking lot, complete with virtual customers and their virtual cars, and interpret information from simulated sensors. The filtered information can be sent to our online services to test the UI and other parts of our system. In order to observe how our system responds to different situations, we can apply various situations to our virtual parking lot. For example, we can induce a huge influx of cars to see how our system handles the sudden traffic. Once the system appears to work for the most part in our virtual environment, we may then proceed to use actual hardware to confirm that the system actually works. This is an important step in our process, since real-life sensors usually give fuzzy data that can be difficult for software to properly interpret. In other words, the virtual parking lot gives us an indication of how well our system manages information from various sensors, and our hardware experiments tell us how well our system can obtain that information in the first place.

Section 14: References

Roser, M. and Ortiz-Ospina, E. (2019). *World Population Growth*. [online] Our World in Data. Available at: https://ourworldindata.org/world-population-growth [Accessed 6 Feb. 2019].

Hedges & Company. (2019). *US VIO Vehicle Registration Data 2018, Fast Quote on Car Data*. [online] Available at: https://hedgescompany.com/automotive-market-research-statistics/auto-mailing-lists-and-marketing/ [Accessed 7 Feb. 2019].

Rasmussen, I. (2016). *A Major Parking Garage Problem*. [online] Strong Towns. Available at: https://www.strongtowns.org/journal/2016/11/21/major-parking-garage-problem [Accessed 8 Feb. 2019].

OPS-COM, Inc.(2019) *Definition of LPR System.* [online]. Available at: https://ops-com.com/parking-security-platform/license-plate-recognition/