

First Report

Software Engineering

Group 2

Mehul Vora, Sajeel Ahmad, Chris Steinert,
Omar Ouf, Alex Sanchez, Kerry Liu, Prit Modi
<https://github.com/Rutgers-SE/GalaxyGarage>

February 21, 2016



Individual Contribution Breakdown

Alex Sanchez

- Management of Overleaf, an online LaTeX formatting application - Helped transition collaborative Google Drive documents to final LaTeX documents
- Worked on "Plan of Work" table
- Worked on the Customer Statement Report - Glossary of Terms
- Worked on System Requirements - Functional and Nonfunctional
- System Requirements- On-Screen Appearance Requirements - Created initial hand-sketched screen mock-ups of the user interface (all possible user-interface scenarios)
- Completed Stakeholders, in addition to Actors and Goals
- Worked on Use Cases - Causal Descriptions and created the Use Case Diagrams
- Created the Domain model
- Wrote Concept Definitions

Mehul Vora

- Project Management
- Organized group meetings, assigned team members responsibilities, and is keeping track of deadlines
- Proof-reading the collated report to ensure consistent flow, proper layout, grammar, and proper completion of report requirements
- Worked on the Customer Statement Report - Problem Domain and Proposed Solution
- Worked on System Requirements - Functional and Nonfunctional
- Assisted in completing the Stakeholders along with Actors and Goals
- Worked on Use Cases - Causal Descriptions and created the Traceability Matrix
- Worked on Use Cases - System Sequence Diagrams - Created System Sequence Diagrams for Use Case 1 and Use Case 2
- Created Traceability Matrix for Domain Model
- Contributed to Attribute Definitions

Kerry Liu

- Creation and management of the Google Drive (Repository)
- Creation and management of Overleaf, an online LaTeX formatting application - Transitioned collaborative Google Drive documents to final LaTeX documents
- Worked on "Plan of Work" table
- Worked on System Requirements - Functional and Nonfunctional
- Completed all of the On-Screen Requirements using graphic-heavy application
- Worked on Use Cases - Causal Descriptions
- User Interface Specification - Preliminary Design - Created all screen mock-ups using 100 percent free open source graphic intensive / heavy software
- Created References page, in addition to listing references used
- Wrote Association Definitions

Sajeel Ahmad

- Worked on the Customer Statement Report - Proposed Solution
- Worked on Plan of Work
- Worked on the Customer Statement Report - Glossary of Terms
- Worked on System Requirements - Functional and Nonfunctional
- Worked on Use Cases - Causal Descriptions
- Worked on Use Case - Fully-Dressed Description for Use Case 6
- Worked on Use Cases - System Sequence Diagrams - Created System Sequence Diagrams for Use Case 6
- Used graphic-heavy applications to create various versions of garage logo
- Contributed to Attribute Definitions

Chris Steinert

- Worked on the Customer Statement Report - Problem Domain
- Worked on System Requirements - Functional and Nonfunctional
- Worked on Use Cases - Causal Descriptions
- Worked on Use Cases - Fully-Dressed Description - Use Case 3

- Worked on User Effort Estimation
- Wrote System Operations Contracts

Omar Ouf

- Worked on the Customer Statement Report - Proposed Solution
- Worked on System Requirements - Functional and Nonfunctional
- Worked on Use Cases - Causal Descriptions
- Wrote Use Cases - Fully-Dressed Descriptions for Use Cases 1 and 2
- Wrote System Operations Contracts

Prit Modi

- Revised Functional Requirements for spelling errors
- Assigned corresponding functional requirements for 2-3 Use Case Descriptions
- Contributed to Attribute Definitions

Contents

1	Customer Statement of Requirements	7
1.1	Problem Statement	7
1.2	Proposed Solution	8
1.3	Glossary of Terms	10
2	System Requirements	12
2.1	Functional Requirements	12
2.2	Non-Functional Requirements	12
2.3	On-Screen Appearance Requirements	13
3	Functional Requirements Specification	16
3.1	Stakeholders	16
3.2	Actors and Goals	16
4	Use Cases	16
4.1	Casual Description	16
4.2	Use Case Diagram	18
4.3	Traceability Matrix	19
4.4	Fully - Dressed Descriptions	20
4.5	System Sequence Diagrams	23
4.5.1	System Sequence Diagram for UC-1: Registration	23
4.5.2	System Sequence Diagram for UC-2: Reservation	24
4.5.3	System Sequence Diagram for UC-7: Park	25
5	User Interface Specification	26
5.1	Preliminary Design	26
5.2	User Effort Estimation	28
6	Domain Analysis	29

6.1	Domain Model	29
6.1.1	Concept Definitions	29
6.1.2	Association Definitions	30
6.1.3	Attribute Definitions	31
6.1.4	Traceability Matrix	34
6.2	System Operation Contracts	34
7	Plan of Work	37
8	References	38

1 Customer Statement of Requirements

1.1 Problem Statement

In today's modern age, with technology rapidly increasing its presence in our everyday lives, the parking garage has failed to stay consistent with innovation. The traditional, common parking garage in our society operates with an inadequate system that utilizes the familiar ticket method, which has become outdated and inefficient. As a result, this implementation is known for causing congestion upon entering and leaving the parking garage. In addition, utilization of tickets has produced many inefficiencies and has become troublesome. Thus, the most common cases are depicted when a customer loses their ticket. As a result, this causes the customer to be unable to leave the parking garage without requiring tedious assistance. Furthermore, tickets require customers to pay at a special terminal and/or at an exit terminal, which is not only sluggish but also an inconvenience to the customer.

Without any automation in the parking garage, it is impossible to determine which spots are vacant or populated without having an employee(s) self-managing the occupancy of the entire garage. As a consequence, this dramatically takes a toll on maximizing the parking garage's total profit for vacant parking spots will fail to be utilized. In addition, another complication customers encounter is that they have no way of distinguishing where available spots are in the parking garage. This causes customers to roam tediously throughout the garage until they locate a vacant spot. Hence, this produces congestion, and aids in the list of problems which make the traditional garage inefficient.

The customer has taken note of these inefficiencies and has asked us to design and implement a software system to better maximize their profits. This software will assign users a parking spot and allow for reservations to better maximize space and decrease inefficiencies. This software system should also decrease the amount of employees needed to run the garage, potentially making it autonomous. Increased security features, including cameras for surveillance, were also suggested to be implemented for safety measures.

1.2 Proposed Solution

To build a garage system that facilitates all of the requirements, a solution should be created with the customer in mind. It should be developed taking into account all of the concerns mentioned above and allow for superior operation of parking garages in order to handle customers' parking needs efficiently as well as simplify the process for users. It should be a seamless and easy system that both the owner and the customer will benefit from using.

To improve the efficiency of the automated system it was decided to implement QR codes to minimize the check-in time and additionally, to add a bonus security aspect to the system. Moreover, QR codes account for customers' online reservations and for walk-in's. As noted, the QR codes are scanned upon entrance and exit. Thus, the system checks that the car linked to the QR code on entrance is the same car on exit. For this, the system should link the customer's QR code to their car's license plate. The suggested implementation will add immense flexibility to customers, for they, after making a reservation, will be allowed to use any acceptable vehicle to park in the garage.

This system should allow for online reservations, which can be booked in advance depending on the owner's configuration. For example, customers can make reservations in advance up to a week, 2 weeks, or even a month. Once the reservation is confirmed the customer should be emailed a QR code that will be utilized when entering the garage. Additionally, the customer should have the ability to delete his reservation up to a week in advance of placement. Moreover, prior to the day of the reservation, a notification should be sent to the customer to remind him of the reservation. However, if the customer does not populate the reserved parking an hour past the reserved time, he or she should be fined, and the slot reservation should be terminated.

To mitigate the hassle of looking for a vacant parking slot, the design should allow for assigning each customer a specific spot upon entrance and/or reservation. Thus, this would improve the customer experience and improve customer retention rate. In addition to effectively utilizing the occupancy of the garage, this would allow the owner to have a elegant simulation of the state of the garage at any given point in time. Additionally, by assigning

customers specific slots, the system should be able to dynamically allocate the slots available for reservations and for walk-ins depending on the occupancy at any given point in time.

The website should allow a variety of user-friendly features. Moreover, the website should allow the customers to create accounts, which will consist of essential contact information. Additionally, this information should be reused by the system when the customer is proceeding with online reservations to fasten the process. Meanwhile, the customer account should also provide a backlog of when reservations were previously made. Various payment options should also be utilized when the customer is making reservations. Furthermore, there will be customers who do not wish to create an account, yet are still interested in online reservations; These customers should be able to do so by providing the minimum information per reservation as a one-time guest reservation. In addition, there should be a customer support section, which will allow the customer to report or troubleshoot any issues he or she is having. This will assure that customers are able to correctly make reservations and/or accounts if they are having difficulty, in addition to resolving any complaints that may arise while using the automated parking garage. Lastly, the website should allow the owner to access multiple features, some of which are available for the customer. The owner should be able to change the configuration settings related to the garage policies, including the pricing policy. Additionally, features exclusively available for the owner should include the ability to view a current, detailed simulation of the garage at any point in time, and to view all customers currently present in his or her garage.

1.3 Glossary of Terms

Admin A super-user who has more access to the system than the owner(s) of the parking deck/garage and who is allowed to make changes to the system etc., in order to fix it or make improvements.

Bulk Reservation Allowing one account to reserve multiple spots.

Camera A device used to scan the customers license plate entering and exiting as well as for security surveillance.

Customer Anyone who is reserving or walking into the garage to park.

Database A place to store persistent data such as users owners reservation information etc.

Drive-in A customer that drives into the garage without first making a reservation online.

External Actor Human interaction with the reservation system.

Internal Actor Secure terminals that interact with the reservation system located within the parking deck.

No-show When a reserved parking spot is not used in the time-frame it is reserved.

OAuth2 The way we use Facebook, Twitter, Google Plus, GitHub, etc., to log-in to the website.

Over-charging Charging the customer for using a parking spot for longer than they reserved.

Overbooking Booking more reservation spots than are currently available.

Overstay When a customer keeps their spot for longer than they reserved.

Owner The user of the system, who is able to configure the settings, business policies, and pricing for the garage.

Periodic Bulk Reservation A combination of bulk and periodic reservation.

Periodic Reservation Allow a customer to automatically reserve a spot either weekly monthly or yearly.

QR Code A code that will be used to authenticate a customer and tell them their parking slot number.

Rain-Check Allow the customer to make a reservation on another time because the garage is full.

Reservation Reserving a parking spot inside the garage in advance.

Reservation System The software that handles SCRUD for the reservations.

Resource Word used to describe anything that was created using SCRUD.

SCRUD Acronym used to describe common operations in web applications. Search, Create, Read, Update, Delete.

Sensor A device used to measure occupancy and see which spots are vacant or populated. There will be a sensor on each parking slot.

SMS Acronym for Short Message Service. Used to send text messages.

Terminal Access points around the parking garage, used for controlled customer interaction.

2 System Requirements

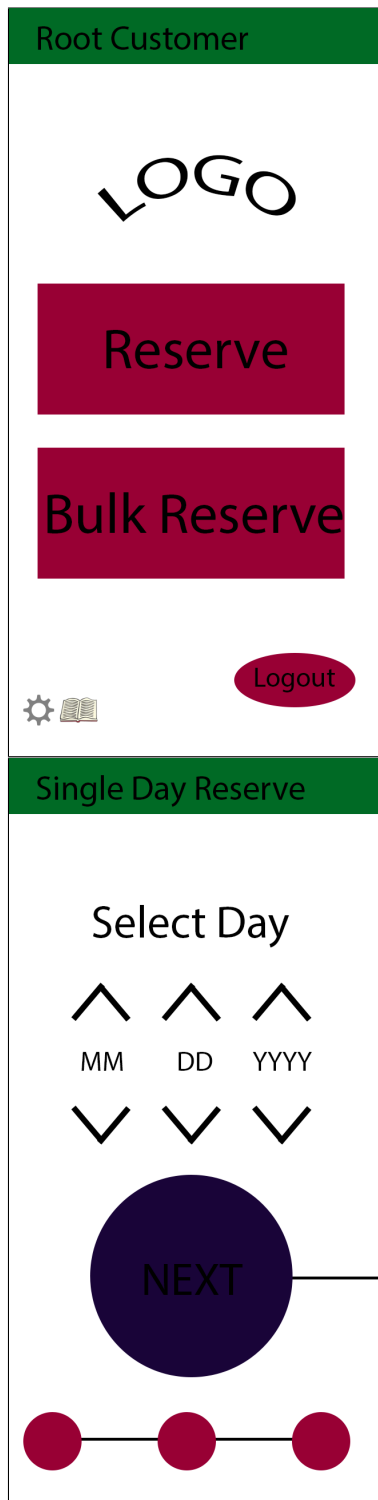
2.1 Functional Requirements

Identifier	Priority	Requirement
REQ_1	5	System shall handle user authentication of the Admin, Customer, and Owner
REQ_2	4	System shall be able to handle modifying all types accounts
REQ_3	5	System shall be able to understand the different privileges between the different accounts
REQ_4	4	System shall be able to delete accounts
REQ_5	4	System shall keep track of the modifications done to the accounts
REQ_6	2	System shall limit the amount of failed log-in attempts
REQ_7	3	System shall encrypt the user passwords
REQ_8	2	System shall email users when important events happen
REQ_9	3	System shall send email reminders when the users reservation date approaches
REQ_10	4	Customers who overstay shall pay an additional fee
REQ_11	5	Customers who do not have a reservation shall get a QR code from a terminal
REQ_12	4	Customers shall modify their reservation if desired to do so
REQ_13	3	Management shall decide on a time frame when a reschedule can be done with notice
REQ_14	4	The system shall assign the user to the closest available spot for efficiency
REQ_15	5	The system shall turn away walk-ins if the garage is full
REQ_16	4	The system shall give a rain check to users with reservation if they arrive and the garage is full
REQ_17	5	The system shall keep track of all reservations, walk-ins, and vacant spaces
REQ_18	4	The system shall generate a QR code for each reservation
REQ_19	5	The system shall implement algorithms to better maximize occupancy
REQ_20	4	The system shall employ an overbooking technique to filling spaces
REQ_21	5	The system shall keep track of all vacant parking spaces
REQ_22	4	The system shall indicate the operating status of garage (open or closed)
REQ_23	3	The system shall allow owner to modify the pricing/reservation policies
REQ_24	3	The system shall allow changes to be applied at future dates
REQ_25	5	The system shall be able to collect and hold all payments
REQ_26	3	The system shall allow the owner to modify garage operation times
REQ_27	3	The system shall allow the owner to close the garage for specific day(s) (holidays, maintenance, etc.)
REQ_28	5	The system shall display how many vacant parking spaces are available
REQ_29	5	The system shall display the number of reservations and walk-ins
REQ_30	4	The system shall display to the owner how much total profit has been made for the day, week, month, etc.
REQ_31	2	The system shall have special pricing for a set interval (holidays, and/or attraction events)
REQ_32	3	The system shall keep daily/weekly logs to show the revenue generated, along with the occupancy statuses
REQ_33	3	The system shall have a customer support section where customers can acquire assistance

2.2 Non-Functional Requirements

Identifier	Priority	Requirement
REQ_34	1	The system shall monitor security surveillance of the parking garage
REQ_35	1	The system should allow the owner to reserve multiple spots at once
REQ_36	5	The system shall have a sensor in every parking spot to sense if it is occupied or not
REQ_37	2	The system should have all data backed up, so not to lose any reservations, payments, etc
REQ_38	3	Customers should have access to all their parking history
REQ_39	5	The QR code and terminal shall work at all times

2.3 On-Screen Appearance Requirements



Root View

Priority: 5

The customer or guest, when loading the website, should see the logo of the garage, and options to reserve spaces in the garage.

Single Day Reserve View


Priority: 5

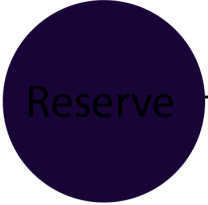
When the customer wants to reserve a time slot for a single day; they should be greeted with a form to pick a day, and a "next button" to progress the reservation creation process.


Bulk Reserve

Select Time

Hours







Reminder

Hey buddy,
I knew it's been
awhile, but don't
forget you made a
reservation at this
garage. So come by
later.

Time Select View

Priority: 5

After selecting a date, the customer should be presented with a dual slider adjustment bar. This bar will be the way the customer selects his time slot.


Reminder Email View

Priority: 4

When the date for the reservation approaches, the user should receive an email reminder.

QR Code

Here is your QR Code




Use this Code to enter the garage.

Customer Settings

Name

E-mail

[Privacy](#)

 Back

QR Code Email View

Priority: 4

After a customer creates a reservation, they should receive an email containing a QR Code for their reservation.

Customer Settings View

Priority: 5

Customer should see the list of pending and previous reservations in a sequential order.

3 Functional Requirements Specification

3.1 Stakeholders

This automation system is designed to vastly improve efficiency of a parking garage. Moreover, various sensors will be installed to maximize the occupancy of the parking garage. With the aid of these sensors and the system automation, the garage profit will definitely see noticeable increases.

3.2 Actors and Goals

Actors	Goals
Customer	Create reservations and Park in Garage. Main line of revenue.
Owner	Sets business policies, collects profits.
Admin	Has access to system diagnostics and other system information.
Reservation Website	How customers/owner/admin view and modify the reservation system.
Lot Sensor	How the system determines if parking spots are taken.
Entrance Terminal	How the customer scans qr and view spot number while entering.
Cameras	Used for security surveillance, and scanning license plate numbers
Garage Display	A display that will be placed outside of the garage to indicate occupancy status of the garage, such as how many vacant spots are available and how many spots are populated, in addition to garage operating times and prices.

4 Use Cases

4.1 Casual Description

UC-1: Registration - To allow the owners and/or customers to create an account in order to use the services provided by the system.

- Addresses REQ-1, REQ-2, REQ-3, REQ-4, REQ-5, REQ-6, REQ-7 and REQ-33, which are all stated in Section 2, System Requirements.

UC-2: Reservation - To allow customers and guests to create future reservations for parking slots in the garage.

- Addresses REQ-8, REQ-9, REQ-10, REQ-12, REQ-17, REQ-21, REQ-23, REQ-25, REQ-31, and REQ-33, which are all stated in Section 2, System Requirements.

UC-3: Walk-In - To allow customers to park in the garage without creating a reservation online if vacancy.

- Addresses REQ-15, REQ-17, and REQ-29, which are all stated in Section 2, System Requirements.

UC-4: SetPrice - Owner has the ability to manage prices for reservations, walk-ins, special events, etc.

- Addresses REQ-10, REQ-23, REQ-24, and REQ-31, which are all stated in Section 2, System Requirements.

UC-5: SetOperationTime - Owner has the ability to control when the parking garage will open and close.

- Addresses REQ-13, REQ-22, REQ-26, and REQ-27. which are all stated in Section 2, System Requirements.

UC-6: Statistics Display - To display current occupation statistics & history of occupation or profit.

- Addresses REQ-17, REQ-21, REQ-22, REQ-25, REQ-27, REQ-28, REQ-30, and REQ-32 which are all stated in Section 2, System Requirements.

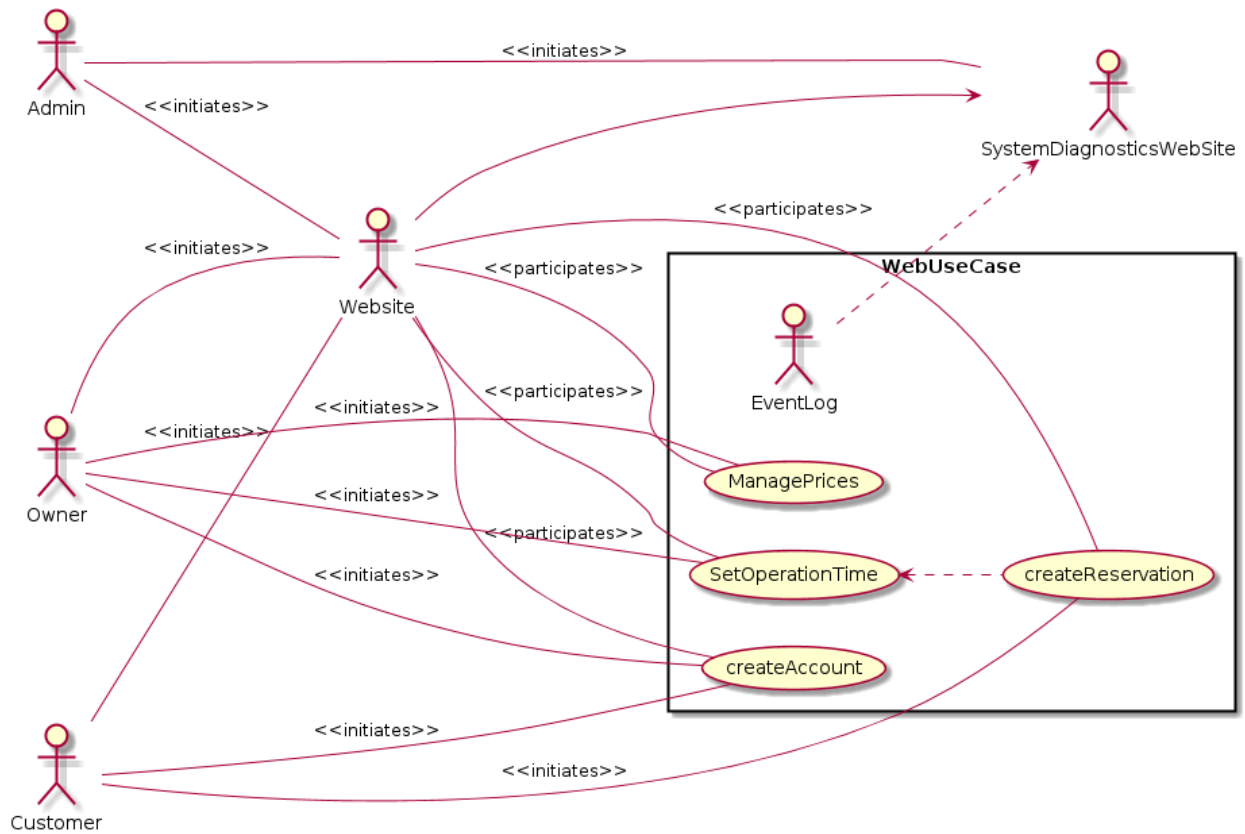
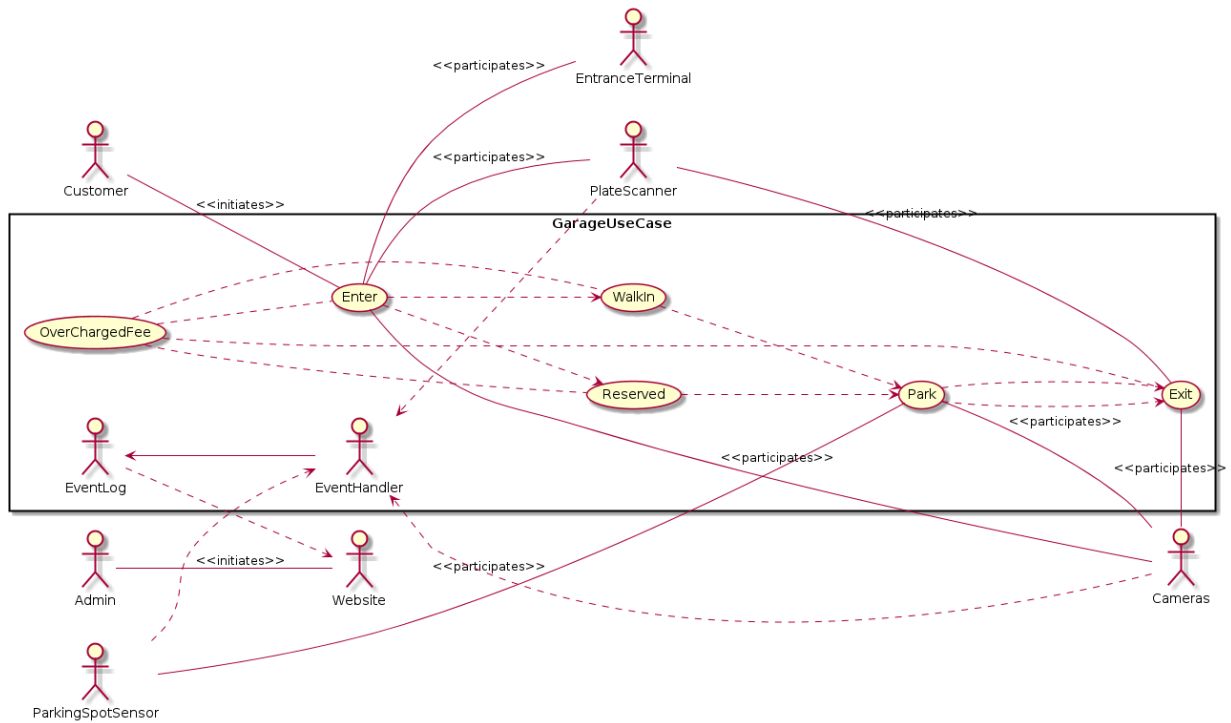
UC-7: Park - Customer scan QR code through entrance terminal and populate a nearby parking spot.

- Addresses REQ-11, REQ-14, REQ-15, REQ-16 REQ-17, REQ-18, REQ-19, REQ-20, REQ-21, REQ-28, and REQ-29 which are all stated in Section 2, System Requirements.

UC-8: Exit - Customer paid for the duration of stay and left the parking garage; Thus, leaving behind a vacant parking slot.

- Addresses REQ-17, REQ-21, and REQ-28, which are all stated in Section 2, System Requirements.

4.2 Use Case Diagram



4.3 Traceability Matrix

	Use Cases	Priority	UC-1: Registration	UC-2: Reservation	UC-3: Walk-In	UC-4: SetPrice	UC-5: SetOperationTime	UC-6: Statistics Display	UC-7: Park	UC-8: Exit
Requirements										
REQ-1	5	X								
REQ-2	4	X								
REQ-3	5	X								
REQ-4	4	X								
REQ-5	4	X								
REQ-6	2	X								
REQ-7	3	X								
REQ-8	2		X							
REQ-9	3		X							
REQ-10	4		X		X					
REQ-11	5							X		
REQ-12	4		X							
REQ-13	3					X				
REQ-14	4							X		
REQ-15	5			X				X		
REQ-16	4							X		
REQ-17	5		X	X			X	X	X	
REQ-18	4							X		
REQ-19	5							X		
REQ-20	4							X		
REQ-21	5		X				X	X	X	
REQ-22	4					X	X			
REQ-23	3		X		X					
REQ-24	3				X					
REQ-25	5		X				X			
REQ-26	3					X				
REQ-27	3					X	X			
REQ-28	5						X	X	X	
REQ-29	5			X				X		
REQ-30	4						X			
REQ-31	2		X		X					
REQ-32	3						X			
REQ-33	3	X	X							

4.4 Fully - Dressed Descriptions

Use Case UC-1: Registration

Related Requirements: REQ-1, REQ-2, REQ-3, REQ-4, REQ-5, REQ-6, REQ-7 and REQ-33, which are all stated in Section 2, System Requirements.

Initiating Actor: Customer or Owner

Actors Goal: To create an account for any of the initiating actors to allow them access to the services.

Participating Actors: Website

Preconditions: Website is up and running.

Post-conditions: Customer/Owner info acquired from the form are stored in the database.

Flow of Events for Main Success Scenario:

1. Owner/Customer: arrives at the website URL and selects "Registration"
2. Website redirects the Owner/Customer to fill out a form with required information.
3. Website creates an account with proper permissions.
4. Website redirects the Owner/Customer to his/her account.

Flow of Events for Extensions (Alternative Scenarios):

1. User: arrives at the website URL and selects "forgot Username/Password."
2. Website requests information required for the password/username reset.
3. Customer receives an email containing further instructions.

Use Case UC-2: Reservation

Related Requirements:	REQ-8, REQ-9, REQ-10, REQ-12, REQ-17, REQ-21, REQ-23, REQ-25, REQ-31, and REQ-33 , which are all stated in Section 2, System Requirements.
Initiating Actor:	Customer or Owner
Actors Goal:	To create a new reservation or to modify an existing one.
Participating Actors:	Website
Preconditions:	Website is up and running, the Customer/Owner already has an existing account.
Post-conditions:	Reservation is stored in the database.

Flow of Events for Main Success Scenario:

1. Owner/Customer arrives at the website and logs-in.
2. Owner/Customer presses on "Reservation" to create a new reservation.
3. Owner/Customer fills out a form to provide the desired time.
4. Website will check if vacant slot available and redirect the Owner/Customer to a "Confirm Reservation with Payment!" , with an option of paying right away.
5. Website will give Owner/Customer a QR code that they will use when entering and leaving the garage.

Flow of Events for Extensions (Alternative Scenarios):

1. Owner/Customer arrives at the website and log in.
2. Owner/Customer will press on "Reservation" to create a new reservation.
3. Owner/Customer fill out a form to provide the desired time.
4. Website will check if vacancy at inputted time.
5. Website will redirect the Owner/Customer to a "Unfortunately, the garage is fully occupied at this time!" page.
6. Website will ask the Owner/Customer if they want to try a different time.

Use Case UC-7: Park

Related Requirements: Addressed REQ-11, REQ-14, REQ-15, REQ-16 REQ-17, REQ-18, REQ-19, REQ-20, and REQ-21 which are all stated in Section 2, System Requirements.

Initiating Actor: Customer

Actors Goal: To claim reservation and park in garage.

Participating Actors: Reservation Website, Lot Sensor, Entrance Terminal

Preconditions: The garage has enough parking spots to accommodate the customer. The customer has an acceptable vehicle.

Post-conditions: Customer license plate is stored in the database. The parking spot is set to occupied.

Flow of Events for Main Success Scenario:

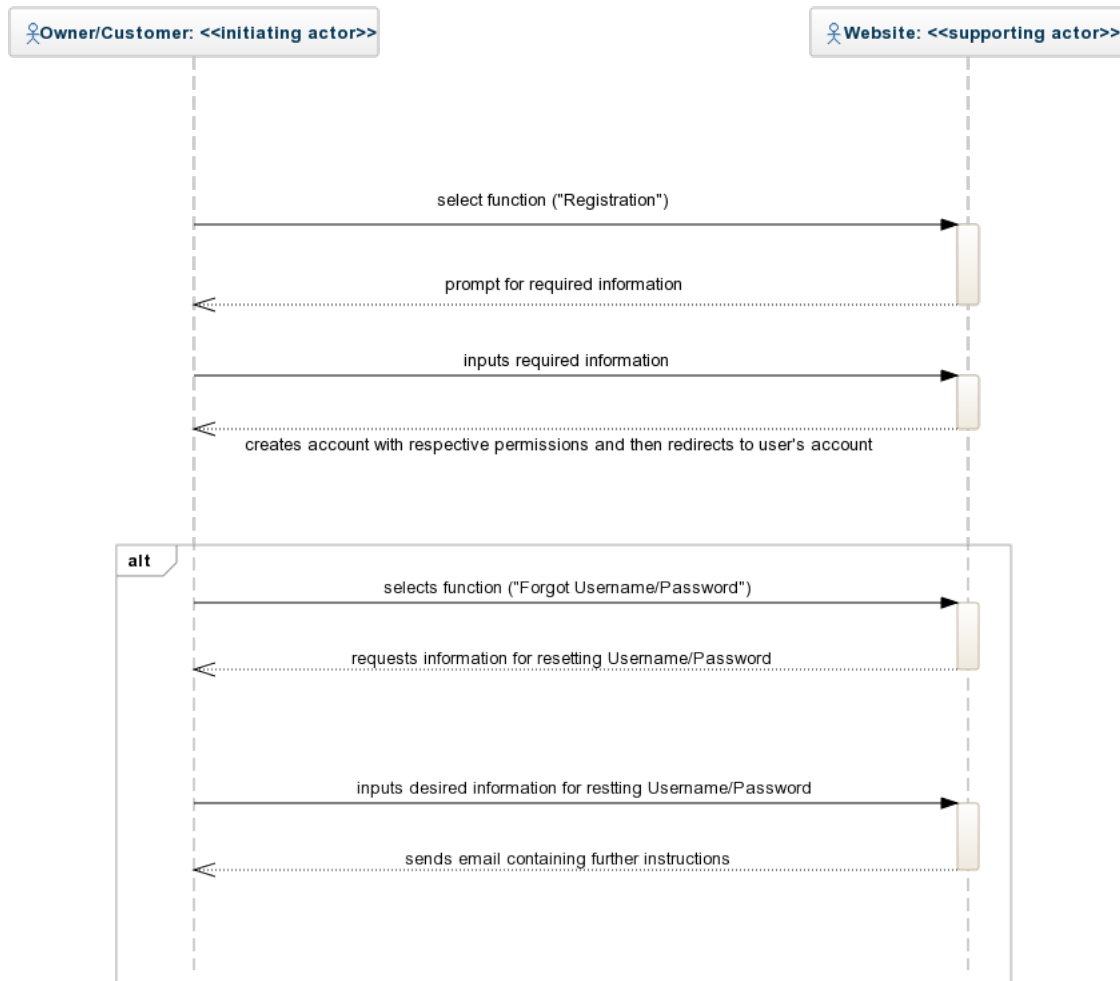
1. Customer arrives at parking garage entrance.
2. (a) Customer receives a QR code if they have not reserved a parking spot.
(b) Customer has reservation, and scans QR code to claim reserved parking spot.
3. The customer is allowed entrance into garage.
4. The system determines the closest, possible parking spot based on currently available spots and displays it to the customer.
5. Garage terminal directs the customer to the open parking spot.
6. The system updates the garage occupancy and adds details about the customer to the database (license plate, QR code).
7. The customer enters into the garage and correctly parks in their assigned spot.

Flow of Events for Extensions (Alternative Scenarios):

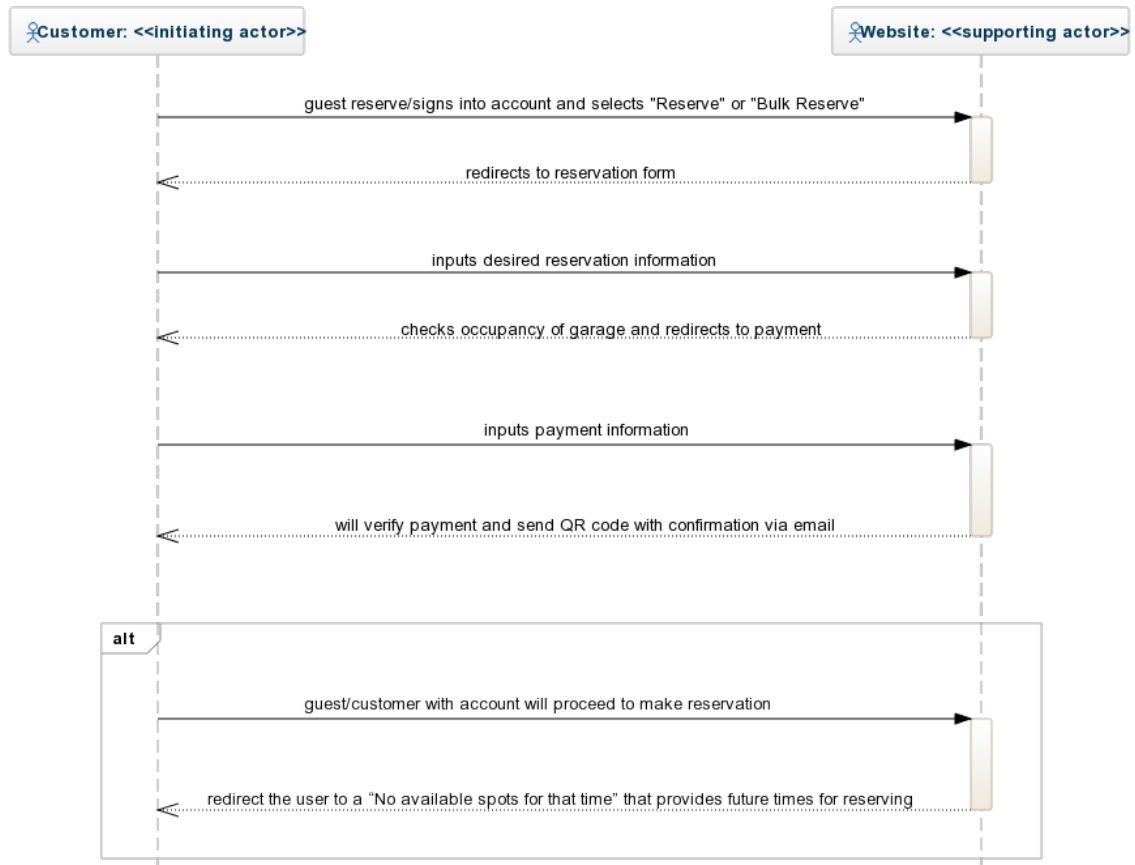
1. Customer arrives at parking garage entrance.
2. Customer is turned away if the garage has reached maximum occupancy.
3. (a) Customers who had previously made a reservation are given a rain check to park in the garage on another occasion.
(b) Walk-in customers are turned away

4.5 System Sequence Diagrams

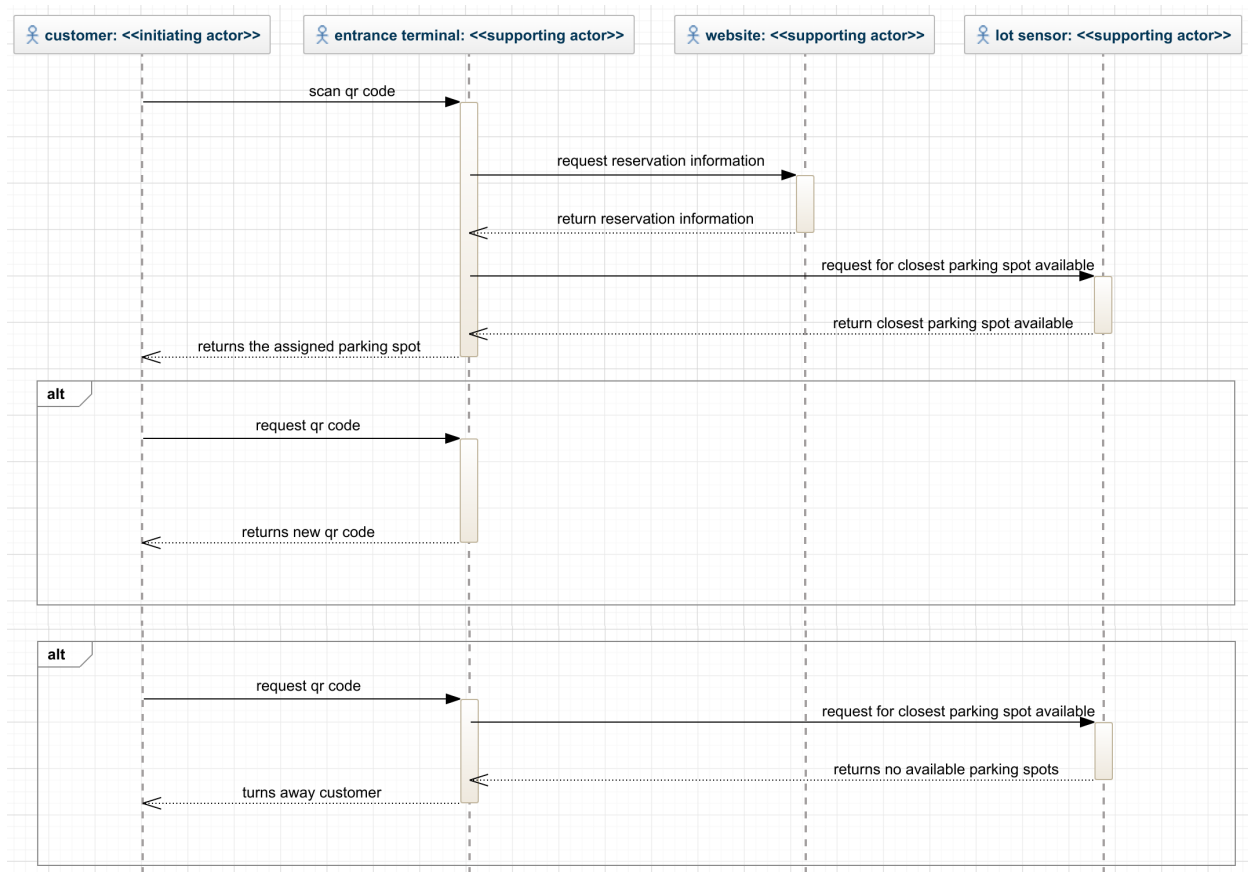
4.5.1 System Sequence Diagram for UC-1: Registration



4.5.2 System Sequence Diagram for UC-2: Reservation

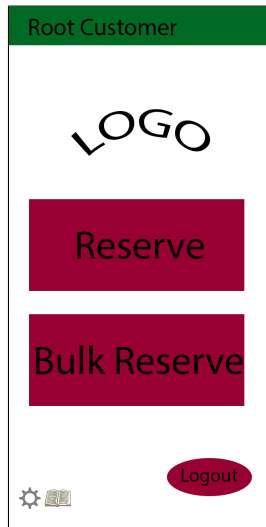


4.5.3 System Sequence Diagram for UC-7: Park

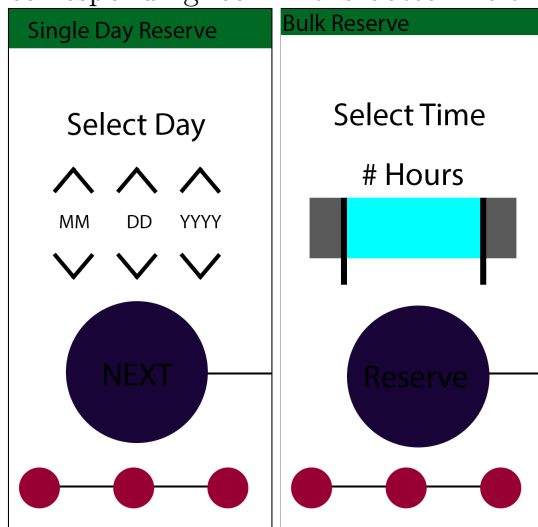


5 User Interface Specification

5.1 Preliminary Design



Once logging-in, this is the page that the customers will be presented with. Here he/she can choose whether they want to reserve a single spot, or if he wants to reserve multiple spots. In addition, the customer can change his/her credentials or read instructions by clicking the corresponding icon in the bottom left.



If the user wishes to reserve a single day, he/she clicks the "Reserve" button from the home screen, which will bring him/her to this page. Here he/she can choose the date for which he/she wants to make a reservation. After the user does that, they can confirm the input with the "Next" button, he will pick his parking hours with a slider. Finally, after clicking

the "Reserve" button, the reservation will be confirmed.

Bulk Reserve

Bulk Select

-/+ Spots

MM DD YYYY

NEXT

E-mail Bulk Reserve

Email Address

Text Edit

Text Edit

Text Edit

Next

If the customer selected "Bulk Reserve", an additional option will be given to select the amount of spots he/she would like to reserve. Next he/she will enter the email address of the person who will be parking in the reserved spot. Then, the respective QR codes and reminders will be sent to them. After these two steps, the rest of the process for bulk reservations is identical to making a single reservation.

Reservation Fail

Oops
Something
went wrong.

Bulk Reserve

Your spot
is reserved!

An email has
been sent.

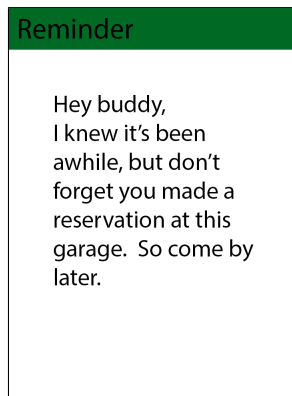
QR Code

Here is your
QR Code

Use this Code
to enter the
garage.

If the reservation failed, the customer will be presented with a failed screen, where he/she would have to go back and attempt to make the reservation again. If the reservation is successful, the customer will be presented a success screen, and a QR code will be sent to

the corresponding email.



When its almost time for the reservation, the person who will park in the reserved spot will receive an email reminding them. There is no need to reply to this email or do anything fancy; It simply serves as a reminder.

5.2 User Effort Estimation

UC-1: Registration

- Navigation:
 - Click "sign up" button/link - 1 Click
 - Click "sign me up" after data entry - 1 Click
- Data Entry:
 - Enter email address:
 - Enter password:
 - Confirm password:

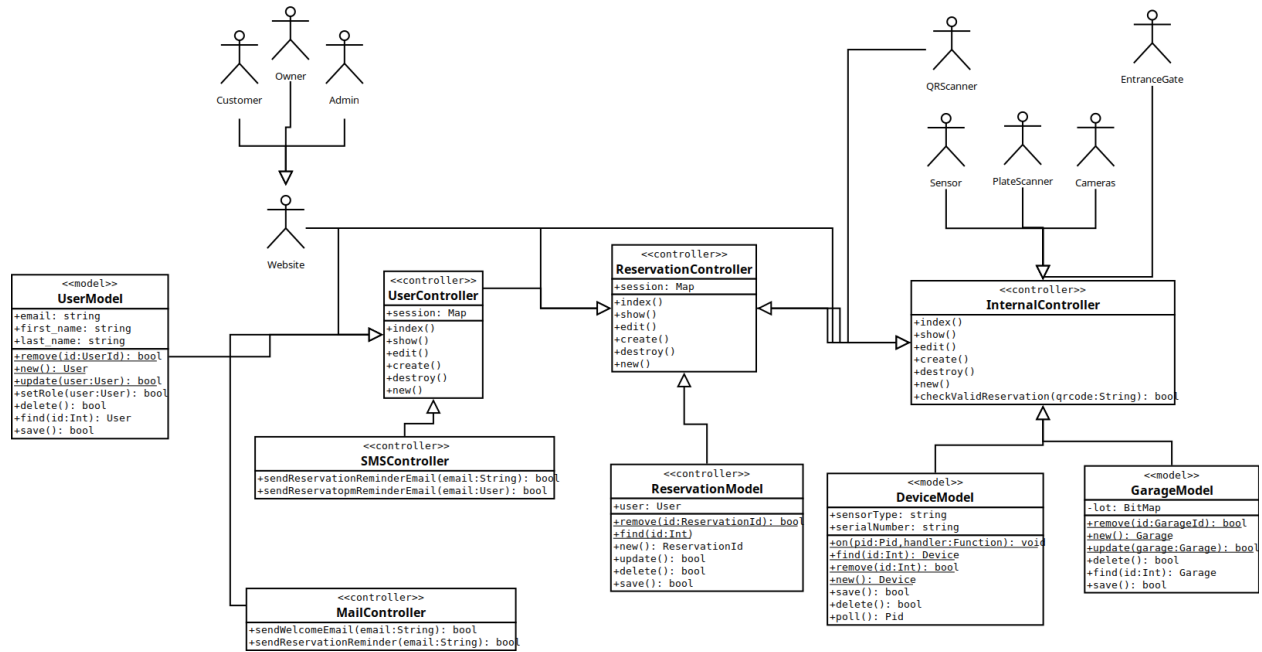
UC-2: Reservation:

- Navigation:
 - 1 click total
 - Click "Reserve" button
- Data Entry:
 - 2 clicks and 8 keystrokes in total
 - "Enter number of spots" - total of 1 keystroke
 - "Enter data into correct field" - total of 6 keystroke

- Click "Next" - 1 click
- "Enter length of stay" - total of 1 keystroke
- Click "Reserve" - 1 click

6 Domain Analysis

6.1 Domain Model



6.1.1 Concept Definitions

Responsibility	Type	Concept
Contains the business rules for creating different types of users	k	UserModel
Acts as the "router" from the website to the UserModel	d	UserController
Provides the utility to send email to certain users	d	MailerController
Provides the interface to send text messages	d	SMSController
Contains the business functionality for creating, modifying, and destroying reservations.	k	ReservationModel
The layer of abstraction between outside actors and the ReservationModel	d	ReservationController
Acts as the template for the way IO devices in the garage should be stored in the database	k	DeviceModel
How the garage is represented in the system. It will keep track of which spots are available for new reservations	k	GarageModel
Layer of abstraction between both the GarageModel and DeviceModel to outside actors	d	InternalController

6.1.2 Association Definitions

Concept Pair	Association Description	Association Name
Customer <-> Website	Customer interacts with the website to make reservations, create account, or view information about the garage's operating status and occupancy	CustomerWebInterface
Owner <-> Website	Owner interacts with the website to manage business-related functions	OwnverWebInterface
Admin <-> Website	Admin interacts with the website to fix bugs, update, or increase functionality	AdminWebInterface
Website <-> UserController	Accesses functions to manage and create new accounts.	ManagerUserAccounts
UserModel <-> UserController	UserController creates profiles for each of the different types of user accounts.	CreateUserAccounts
SMSController <-> UserController	UserController will use SMSController to send users various sms messages regarding reservations	SendSMSMessage
MailController <-> UserController	UserController will use MailController to send users various email messages regarding reservations	SendEmailMessage
Website <-> ResevationController	Links ReservationController with reservation statuses	ReservationWebInterface
UserController <-> ReservationController	Manages reservation for users	UserWebReservationInterface
ReservationModel <-> ReservationController	Controls and organizes reservations to prevent overbooking.	ReservationManagement
ReservationController <-> InternalController	InternalController will notify ReservationController of all entrances, exits, and currently available parking spaces.	ParkingSpaceUpdate
Sensor <-> InternalController	Sensors on each parking spot will allow the internal controller to know which spots are occupied.	IsParkingSpaceOccupied
PlateScanner <-> InternalController	Security feature that keeps track of plates entering and leaving garage.	PlateScan
Cameras <-> InternalController	Security feature that monitors the garage.	CameraSee
EntranceGate <-> InternalController	InternalController will control the EntranceGate to allow cars to enter and leave.	ControlFrontDoor
DeviceModel <-> InternalController	InternalController will interact with the Device Model to control sensors and devices in the parking garage.	SensorDeviceControl
GarageModel <-> InternalController	InternalController will interact with the Garage Model to control garage properties.	GaragePropertiesControl

6.1.3 Attribute Definitions

Table 1: User Model

Attribute	Description
Email	Get customer's Email Address
First Name	Get customers First Name
Last Name	Get customers Last Name
Remove	Deletes the customers user id from system
Update	Adds any changes to customers personal information
Set Role	Assigns a specific role the user whether it be Customer, Owner or Admin
Delete	Deletes the customers account
Find	Finds the customers information
Save	Saves any changes made to the customers information

Table 2: User Controller

Attribute	Description
Session	Initiate current user
Index	Search through list of user accounts
Show	Display current user account
Edit	Enter user editor to adjust user data
Create	Creates new user account
Destroy	Deletes a user account
New	Adds a new user account instance

Table 3: SMS Controller

Attribute	Description
Send Reservation Reminder Email	Sends out reminders regarding upcoming reservations to the customer

Table 4: Mail Controller

Attribute	Description
Send Welcome Email	Sends out an email introducing the garage
Send Reservation Reminder	Sends out an email reminding the customer about a reservation

Table 5: Reservation Controller

Attribute	Description
Session	Initiate current reservation
Index	Search through list of reservations
Show	Display current reservation
Edit	Enter reservation editor to adjust reservation data
Create	Create reservation
Destroy	Delete reservation
New	New reservation instance

Table 6: Reservation Model

Attribute	Description
User	Get attached User id
Remove	Remove reservation
Find	Search for reservation
New	Add new reservation data
Update	Update reservation data
Delete	Delete reservation data
Save	Save reservation data

Table 7: Internal Controller

Attribute	Description
Index	Search though list of internal devices
Show	Display current internal data
Edit	Enter internal data editor to adjust internal data
Create	Create a new QR code for walk-in
Destroy	Destroy QR codes upon exit so they can not be reused
New	Add a new device to the system
Check Valid Reservation	Will verify QR code is valid and paired with proper reservation

Table 8: Device Model

Attribute	Description
Sensor Type	Add type of sensor (license or spot)
Serial Number	Get serial number of device
On	Turn device on (When device detects events run function attached) This is an event listener
Find	Locate the device inside the garage
Remove	Remove the device / power off
New	Add a new device or device type
Save	Save current device information
Delete	Delete current device from model
Poll	Test device status (Turn device polling on)

Table 9: Garage Model

Attribute	Description
Lot	Will be implemented as a BitMap
Remove	Will remove the reserved spot and/or cancel reservation
New	Will reserve a spot for a new reservation that has been made
Update	Update changes made to garage - BitMap
Delete	Will change parking spot to empty on BitMap
Find	Find vacant or populated parking slot in the garage
Save	Will populate parking spot on BitMap

6.1.4 Traceability Matrix

Use Cases	PW	Domain Concepts								
		User Model	User Controller	SMS Controller	Mail Controller	Reservation Controller	Reservation Model	Internal Controller	Device Model	Garage Model
UC - 1	30	X								
UC - 2	36	X	X	X	X	X	X			
UC - 3	15					X	X	X	X	X
UC - 4	12		X							
UC - 5	13		X					X		
UC - 6	34		X					X		
UC - 7	51					X	X	X	X	X
UC - 8	15							X	X	X
Max PW		36	36	36	36	51	51	51	51	51
Total PW		66	95	36	36	102	102	128	81	81

6.2 System Operation Contracts

Operation: Register	
Precondition	<ul style="list-style-type: none"> • Website is up & running. • Username satisfies requirements (length, special characters). • Username not previously taken. • Password satisfies requirements (length, special characters, capitalization) • Required information, such as full name, address, email address, phone number, payment card are filled appropriately. • If information is not appropriately filled, display the error providing the incorrect segment.

Postcondition

- Account has been created with appropriate permissions & is stored in database.

Operation: Reservation

Precondition

- Website is up & running.
- Customer has logged into his account.
- The selected date-time segment is available.
- If the date-time segment is not available, closest segment is provided as a suggestion, and a re-selection is required.
- If payment card is not attached to account, customer is required to attach a payment card.
- Payment is requested.
- If payment is not successful, save a draft of the reservation.

Postcondition

- User is redirected to his account and a message saying "Registration Successful" is displayed.
- QR code is assigned to the user & reservation.
- Reservation confirmation is emailed to the user, with QR code attached.
- Reservation is saved in database, and parking spots available for that date-time segment is updated.

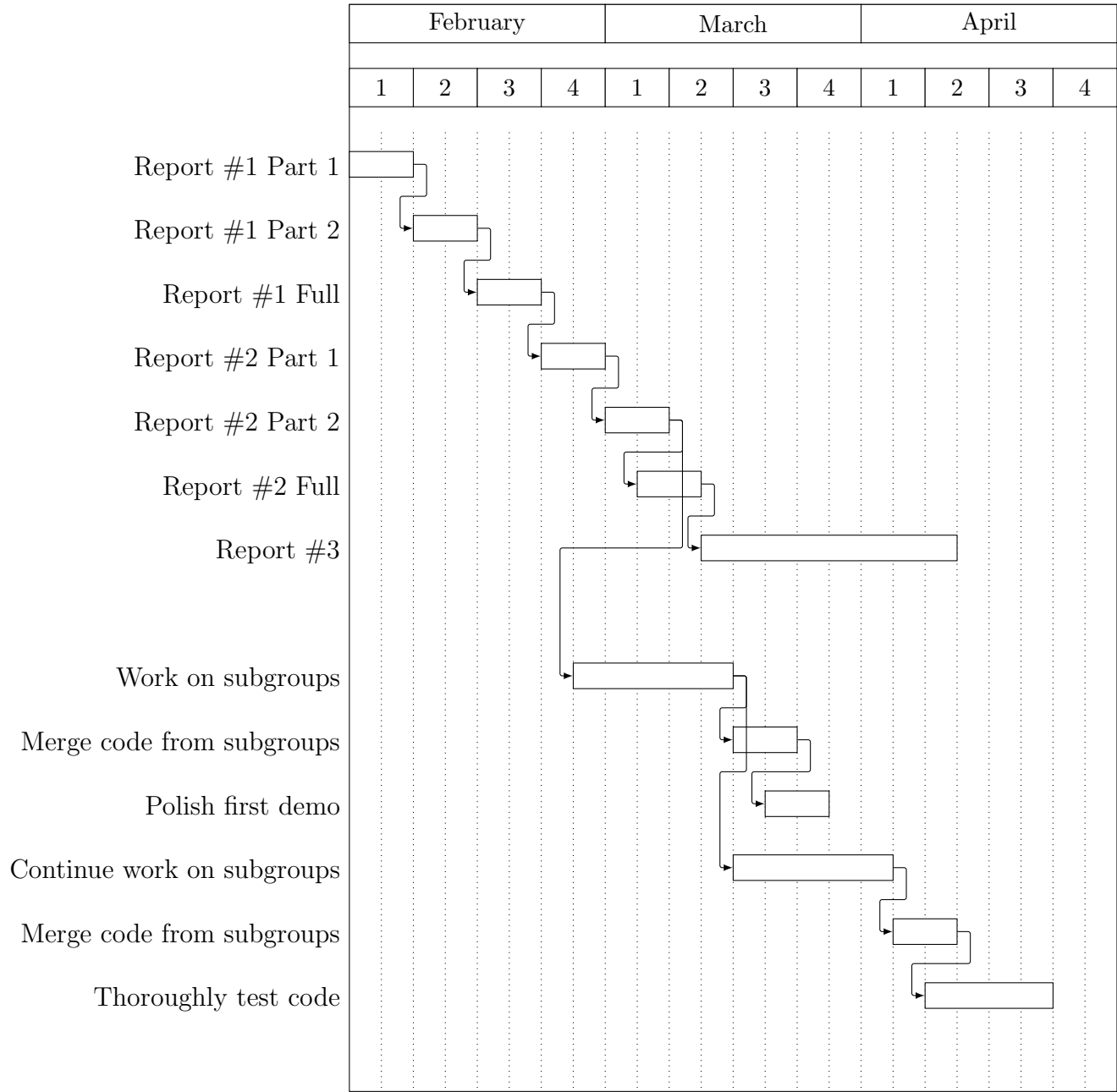
Operation: Park

Precondition	<ul style="list-style-type: none"> • Website is up & running. • The garage terminal has access to the database. • The customer has an acceptable vehicle. • The customer has a phone and will be able to communicate with the garage/website. • The customer has an acceptable vehicle. • The customer will receive a QR code upon arrival (walk-in). • Customer with a reservation will have his QR code when entering.
Postcondition	<ul style="list-style-type: none"> • The customer will have QR code linked to their account for every visit. • The customer's license plate is recorded and linked to customer's account in the database. • The customer parks in their assigned parking spot. • The parking spot is set to occupied. • Customer gets billed for time spent parked at garage.

Operation: Park (Alternative Scenario)	
Precondition	<ul style="list-style-type: none"> • Website is up & running. • The garage terminal has access to the database. • The customer has previously made a reservation, and has arrived on time. • The garage does NOT have enough parking spots to accommodate the customer. • The customer has a phone to be able to communicate with the garage/website.

Postcondition	<ul style="list-style-type: none"> • The customer's account is recorded and is credited a rain check, which he/she can use at a later time. (If reservation was not fulfilled) • The customer leaves without parking.
---------------	---

7 Plan of Work



8 References

https://en.wikipedia.org/wiki/Gantt_chart

<https://www.sharelatex.com/learn/>

<http://www.ece.rutgers.edu/~marsic/Teaching/SE/>