Fit-A-Lot

Group 5

Report 3

Group Number 5:

Matthew Brazza	mattbrazza@gmail.com
Justin Cruz	jaa.cruz16@gmail.com
Sheldon Wong	sheldonwong@gmail.com
Parth Patel	psp022@gmail.com
Sean Wang	seanvwang@gmail.com

Instructor: Professor Ivan Marsic

URLs:
https://sites.google.com/site/parkinggaragesp13/home
https://se1.engr.rutgers.edu/~13group5

Revision History:

v. 1.0	5th May 2013 via Sheldon Wong's Sakai Dropbox
v. 2.0	12th May 2013 via Sheldon Wong's Sakai Dropbox

Contribution Breakdown

All team members contributed equally.

Table of Contents

Customer Statement

Parking garages today currently operate without any computerized system. They usually require employees to walk around and manually check the occupancy of individual spots. The owners are concerned that they are not maximizing profit due the inefficient managements of parking spots. While some parking garages already monitor occupancy using a sensor system that keeps a tally of how many vehicles enter and leave the parking structures, this implementation can be taken a step further to further increase automation and efficiency. The whole system should be able to be implemented regardless if there is an elevator with in the structure or not. It will be a flexible program that can be applied to most parking garages that utilize valets. The new system will encourage customers to make reservations online and make the parking process a hassle-free experience. The system should properly allocate the parking spot reservation to maximize occupancy within the parking structure and be able to take into account contracted reservations, confirmed reservations, and walk-ins.

Several assumptions are made in order to simplify the system. We will assume that the customer has email and SMS texting capabilities and will always be able to read their SMS messages instantly. Additionally, it is assumed that the license plate reader will always correctly read the license plate regardless if the license plate is dirty or damaged.

To use an automated system to its maximum potential, it requires a higher influx of customers, both old and new. A method to achieve this is by streamlining the process of parking a customer’s car. By reducing the hassle on the customer’s end, it will increase customer satisfaction and therefore increase customer retention. The parking garage can also offer several novel features to attract additional customers, such as security features, contracted parking, and reservation notifications. Additionally, the use of valets could improve customer relations by involving a human element in the automation process. However, the valet should primarily be ferrying the vehicles to their parking locations and letting the system handle the logistics of where the vehicle should be parked. Ideally, there will be a mobile application that will direct the valet to the proper parking location. This application will also allow valet employees to update the system on currently occupied or empty spots.

Online reservations can be made and a license plate reader will read the car’s license plate and determine if a previous reservation exists or a new one needs to be made. This will require less interaction for the customer while they are driving into the actual parking structure. The website will allow customers to reserve a time slot in which to park days, weeks, or even months ahead of time and the system will store this information and properly arrange parking reservations to maximize the number of reservations that can be made. Additionally, the system will automatically send an SMS message to the customer when their parking reservation is about to finish.

The website should have a host of features that will benefit both the customer and the manager of the parking garage.The website should allow customers to create accounts so that it can store their personal information so that they can reserve parking spots even more quickly or to even be able to make contracted reservations.The manager will also be able to access an administrative section of the site in order to set certain variables such as policies, pay rates, etc.Employees will have their own accounts to handle day to day operations in the parking garage.A customer who wants to make a reservation will not necessarily have to make an account in order to reserve a spot, but a credit card will be needed to link the customer to a reservation.The website will primarily add more convenience for both the employees of the parking garage and the customers who will be parking at that garage.

The most important feature that would be needed in an automated system would be to maximize the space in the parking structure.The reservations would need to be condensed so that there is no wasted time. The system would also need to be able to notify potential customers that there are available parkings spots and for how long they can park.The system should be able to properly handle customers who have no prior reservations and be able to determine when to know that it can not handle additional walk-ins. A database will need to be created so that the website, employees, license plate reader and the actual system can interact with each other. This way the system can track the cars (by means of the license plate reader and valet employees) that enter and leave the garage to ensure that reservation times are being followed and be updated in real time. There will be no parking sensors in place given how expensive it would be to have every parking spot equipped with one.

To maximize security there needs to be a system that can ensure that the customer receives his or her car back in the same condition.Not only will the system be keep track the condition of the customer’s car, it will also ensure that the manager will not be held liable for possible damages not incurred within the garage.To accomplish this, cars will be brought to a checkpoint where six cameras will take pictures of the car from all angles before it enters the garage and right as its about to leave .With the recorded images stored in the database, the manager has a way of making sure the car that enters the garage is the same as when the car leaves the garage.He/she will be able to dispute or verify any claims from a customer who may have believed that their car was damaged within the parking garage.

There should also be a way to track usage statistics so that parking trends can be extrapolated. By keeping a track of trends, the manager is able to get a better feel for the behaviors of his/her customers. To implement the recording of trends, a license plate reader will also be installed at the checkpoint where the car are documented by the six security cameras. The license plate reader records the plate number and stores it in the garage’s database to keep as a record. The manager can then look at the data to see how many times a certain customer enters the garage and make assumptions on frequent visitors this way. Also, by seeing how many cars enter or use the parking garage at certain times based on the data collected, the manager is able to keep track of busy times during the day as well as during the year.

There will also need to be a way so that the system can be tested to check that it works properly and that there are no design breaking bugs. A simulator will need to be created to test the system under real-world situations and conditions. This simulator will need to test that reservation made online are properly received by the system and that valet personnel can access this data using a mobile application. The simulator will also stress test the virtual parking garage to make sure that the system handles certain exceptions correctly. For example, if someone overstays and the system needs to move reservation spots around to compensate. This simulation will also provide a proof of concept of the overall system and that it works. This simulator will have several variables that can be changed, so that certain scenarios can be tested. It can change things such as: size of the parking garage (number of floors, number of spots) and the numbers of cars that enter/leave.

There are three main goals that need to be accomplished for this project. The first is that there needs to be a website that can interact with the database and be accessed by both the customers and the manager. The second is that there needs to be a mobile application that can be utilized by the employees to update the current status of the garage. The third and probably most important goal is to create an algorithm that can maximize space within the parking garage to also maximize revenue.

Glossary of Terms

Contracted Reservations: Reserved multiple use parking spots that are paid for by customers who require the spot for multiple days. Done by visiting the website. e.g. If the customers' needs the spot everyday for work.

Confirmed Reservations: Reserved single use parking spots that are paid for ahead of time. Done by visiting the website.

Customer: Person(s) who enter the garage with either a parking reservation or looking to make a walk-in reservation.

Database: An interface to which the website and the android application connect to, which is also used to store customer data and parking information.

Driver's License Plate Reader: The Driver's License Reader is going to be a scanner at a checkpoint that will record the license plate number of the vehicle and store it into the systems database.

Drive-ins: Used to describe the moment a parking customer drives into the garage (at this point you don't know if they have a reservation or not.

***Extension:** Customers are allowed to increase the amount of time their car is in the garage, as long as there are available time slots once their time is over.

Manager: The owner of the parking garage. Heads valets and in charge with setting the standards within his or her own garage.

Mobile Application: Used by the valets to quickly verify the reservations of customers and spot availability. Verification needed when license plate reader is unable to confirm a reservation. It is also used by customers when entering the garage to confirm a reservation or to occupy a spot.

No-Show: The act of missing a reservation. The manager will still collect the payment due to paying upon reservation.

Overbooking: The act of accepting more reservations than available parking spots. Typically if a spot contains a no-show, overbooking would fill that spot in order to make more money.This strategy is used to maximize profit, but isn't used in our system.

Overstay: An unavailable spot that may impede future reservations due to the customer not returning within the allotted reserved time.

Reservation: An agreement between the parking garage(company) and the customer to hold a parking spot in advance.

Registration: The action a customer takes to sign-up and enter their information into the company database. When registering before reserving, the customer is asked to input their name, phone number/email, credit card number, address, and date of birth. By registering, it allows the customer to reserve parking spots a lot faster.

***Security Check Cameras:** There will be six cameras placed at a checkpoint that will record the condition of the car before it enters and leaves the parking garage.

Understay: A newly vacant spot due to a customer leaving before the conclusion of their reserved time.

Valet: Employee that will provide the personal interaction with each customer. Will be using the mobile application in order to verify reservations and move the cars.

Walk-ins: Parking customers who walk into the garage without a reservation.

System Requirements

Enumerated Functional Requirements

Identifier	P.W.	Description
REQ-01	4	The system shall scan license plates
REQ-02	5	The system shall recognize registered customers via plate number
REQ-03	2	The system shall record occupied and vacant spots
REQ-04	5	The system should display information on spot availability
REQ-05	4	The system shall allow the valet to view customer information via tablet/smartphone
REQ-06	4	The system shall allow for creation of profile on website through registration
REQ-07	4	The system shall allow valet to create a profile for walk-ins
REQ-08	5	The system shall process payments based on length of parking time
REQ-09	4	The system should allow for early payments for reservations
*REQ-10	5	The system should allow for manual input in case of system failure
REQ-11	5	The system shall manage the parking garage (eg.payroll, pricing, etc.)
REQ-12	3	The system shall require registration to make reservations
*REQ-13	4	The system shall allow for reservation and notify customer if spots are not available.
REQ-14	4	The system shall consolidate parking spots to maximize space availability
REQ-15	5	The system should track tendencies within the parking at the garage
REQ-16	5	The system shall have web and mobile app integration

REQ-17	5	The system shall extract information such as name, address and drivers' license number using the driver's license reader.

Enumerated Nonfunctional Requirements

Identifier	P.W.	Description
REQ-18	5	The system shall accept 'walk-ins', customers who have not placed a reservation. They will be given a spot depending on the vacancy
REQ-19	5	A customer may create an account online to park using the online website.They must provide the following information: 1)First Name 2)Last Name 3) A valid phone number 4)A valid email address 5)A valid credit card 6)Address 7)Date of Birth
REQ-20	4	The interaction between the valet employee and the system or the interaction between the customer and the system shall be minimal it should not take more than five minutes to complete a reservation
REQ-21	3	The data which is backed up, shall be encrypted using disk encryption software
REQ-22	5	Customers shall be able to edit their account information such as their address,add a cars information, edit a cars information.
REQ-23	4	Customers shall be able to view past transactions
REQ-24	3	When creating an account online , a customer must confirm their email address
REQ-25	5	System can be tested to check such that it works properly and that there are no bugs.
REQ-26	5	A simulator will need to be created to test the system under real-world situations and conditions. This simulator will need to test that reservation made online are properly received by the system and that valet personnel can access this data using a mobile application.The simulator will also test the virtual parking garage to make sure that the system handles certain exceptions correctly
REQ-27	5	The system shall be capable of managing customer information and the availability of parking spots

*REQ-28	4	The manager will be able to login into the system to view monthly reports,payroll information,change prices for the parking garage.The manager will also be able to view the number of reserved cars and the number of walk-ins.
REQ-29	5	The system shall be consistent with the customer as well as the manager
*REQ-30	3	The system should have a help page that will help customers fill in the required information to complete a reservation online
REQ-31	4	A customer may cancel a reservation an hour before reservation start time to avoid any penalty charges.
REQ-32	5	A registered customer may edit their reservation an hour before their reserved time.No change to the system can be made within an hour of their reservation.
REQ-33	5	Account information, Parking data and daily reports shall be backed up once a day, to prepare for any natural or human-induced disasters that may occur.
REQ-34	2	Recovery time shall be no greater than 5 minutes if needed
REQ-35	5	The system shall be a 100% accurate when providing customers with a parking spot if available
REQ-36	4	The system shall be incorporated with any parking garage and can be altered for parking garages with different architures
REQ-37	5	The system shall not lose a reservation placed online line because of the system backups that take place regularly
REQ-38	3	The system should be an easy installation and shall be minimized in such a way that configuring the system for a new customer will not take more than one week

On-Screen Appearance Requirements

Identifier	**P.W.**	**Description**

REQ-41	4	In order to make an account online, the user needs to have a device that has internet access to use the site or app.When making the account, the user must be able to input registration information, such as last name, first name, date of birth, phone number, etc.
REQ-42	4	To log into an account, the user must be able to provide his or her account ID and password
REQ-43	4	To edit or delete an account, the user must be able to log into his or her account.There the user will be given the option **to** delete or edit the account. If the user selects to delete the account, a message will appear asking if they are sure about deleting it with “yes” and “no” buttons.If the user select “yes”, the message will disappear and he or she will be directed back to the home page.If the user selects “no”, the message will disappear and he or she will remain in the account page. If the user selects to edit the account, the text will turn into fields that will allow them to edit any of their information.
REQ-44	5	If the user wishes to make a reservation online, he or she will be asked how long he or she wishes to reserve for. If the user does not have an account, then he or she will be required to input information, such as last name, first name, date of birth, phone number etc. If the user does have an account, then the personal information fields on the reservation page will be auto-filled. When reservation is completed, the user will be asked for credit/debit card information to pay for the reservation.After the transaction has completed, the user will be given a confirmation number.
REQ-45	5	To cancel or modify a reservation, the user will be asked for his or her confirmation number.The user will then be given the option to cancel their reservation or to modify any information on his or her reservation page. If the user selects to cancel the reservation, a message will appear

		asking if the user is sure about cancelling it with “yes” or “no” buttons.If the user chooses “yes” the message will disappear and he or she will be directed back to the home page.If the user chooses “no”, the message will disappear and he or she will remain on the reservation page. If the user selects to modify the reservation, the text on the page will turn into fields that will allow them to modify any of the information.
REQ-46	5	If the valet wishes to access the valet account, he or she must be able to provide the account ID and password.On the valet account page, the first thing the valet will see is the bit map.On the bit map the valet will be able to highlight the spots and times that are occupied or vacant.
REQ-47	5	If the valet wishes to see the customer account/reservation information, he or she will have to select that button or link on the home page.After selecting it, he or she will be directed to another page that will ask for the customer’s reservation confirmation number.If the customer does not have the confirmation number, then the valet will have the option to input the customer’s personal information, such as his or her name, date of birth, etc.

Functional Requirements Specification

Stakeholders:

This system is created for implementation in current parking garages and or lots to help parking garage owners increase profits.This system will also be of interest to people who can maintain the system.Below are examples of people and organizations who would be interested:

- Parking Garage Owners
- Valet Parking Services
- Business Enterprises
- Users - Reserved, Contracted, Walk-Ins
- Database Manager
- Business Analyst

Actors and Goals:

Actors	**Goals**
Manager	To manage the employees, parking garage prices, and analyze statistics
Valet	Verify customer information and park the cars
Valet Assistant Interface	Terminal for customers to input their information and time they want to park for
Customer	Make reservations and bring car to the garage.To bring in revenue.
License-Plate Reader	Read the license plate and pass the information to the system
Cameras	Take pictures of car to mark the enter/exit conditions of the car
Vacancy Display	Displays to customers if there are vacant spots within the garage
Website	Website for customers/employees to examine/change reservations and additional customer information
Database	Stores data for the website, and the parking occupancy within

	the garage
Drivers License Reader	Extract information from the drivers license such as name,address city state and zip code.

Use Cases:

Casual Description

UC1: Register- to create an account that will allow a driver to make a reservation. This will require the user to create an account on the internet and provide certain information.

UC2: Reserve Online- allows a driver to save a spot for a given period of time as long as there is a vacancy.

UC3: Walk-In- this allows the driver's who walk-in without a reservation to make an on- the-spot reservation by inputting the amount of time they want to stay and will allow people who walk-in with a reservation move to the park use case.

UC4: Park- to park a driver's car inside the garage.

UC5: Overstay- implements the policies on drivers who have reserved and overstayed.

UC6: Exit- for the driver to obtain his car and notify the valet that a car must be retrieved.

UC7: Manage Prices- allows the manager to manage prices on any fees he will be charging his customers.

UC8: Manage Employee Information- allows the manager to change and update employee information and salary.

UC9: Check statistics- allows the manager to obtain statistics of how many people parked in a day and for how long.

Full-Dressed Descriptions

Use Case UC-1:	**Register**
Related Requirements:	REQ-09, REQ-10, REQ-15, REQ-21, REQ-24, REQ-26, REQ-32

Initiating Actor: Any of: Customer, Valet Actor's Goal: To create an account that will be stored in the database allowing for reservations for the parking lot. Participating Actors: Website, Database Preconditions: The system will request all the required information needed from the customer. Postcondition: The customer's account will be stored within the database and backed up once a day. Flow of events for Main Success Scenario: -> 1. Customer/Valet accesses the website and chooses the "Register" option. <- 2. The system returns the display that states the required information. -> 3. The customer fills out he the required data fields. <- 4. The system takes the information to verify it. If not valid, move back to 3 If valid, continue <- 5. Information is stored into the database.

Use Case UC-2: **Online Reservation**
Related Requirements: REQ-07, REQ-10, REQ-11, REQ-12, REQ-15, REQ-16, REQ-31, REQ-32 Initiating Actor: Customer Actor's Goal: To successfully reserve a spot within the parking garage.

Participating Actors: Website, Database

Preconditions: The user should be logged into his account.The system will prompt the user about their requested parking spot. It will also display whether parking is available.

Postcondition: The system will put the requested parking time into the database.

Flow of events for Main Success Scenario:

-> 1.User enters time that he wants to reserve.
<- 2.System checks to see if there is an open slot. If valid spot, the system will confirm reservation. If no valid spot, the system will ask to input new time (go back to 1)
-> 3.The system will direct customer to a page which displays ‘reservation sucessful’.

Use Case UC-3: Enter Garage

Related Requirements: REQ-01, REQ-02, REQ-03, REQ-04, REQ-07, REQ-08, REQ-10, REQ-20

Initiating Actor: Customer

Actor’s Goal: To reserve a block of time in the garage to park their car.

Participating Actors: Customer, Database, License Plate Reader, License Card Reader, Valet Assistant Interface

Preconditions: There is an open

lot inside the parking garage that the customer can park in. Postconditions: The car will have a spot reserved in the garage. Flows of Events for Main Success Scenario: -> 1.The customer drives up to the entrance gate and presses the start screen. <- 2.The license plate reader reads the license and the system finds a reservation that is associated with the customer. If found the screen will display “Verify information then proceed to the valet ahead.” (Proceed to number 9.) If not found, the screen will ask for time the customer would like to stay, while showing the maximum time they will allowed to stay. (Continue to next step.) -> 3.The customer inputs the amount of hours and minutes he would like to stay. <- 4.The systems verifies a valid input. If valid, it will go to the next screen and ask for phone number If not valid, it will go back to number 3. -> 5.The customer enters valid phone number. <- 6.The systems verifies a valid input. If valid, it will go to the next screen and ask to input driver’s license. If not valid, it will go back to number 5. -> 7.The customer enters driver’s license into driver’s license reader. <- 8.The systems extracts information <= 9.The system signals the user to proceed to the next station to park.

Use Case UC-4: **Park**
Related Requirements: REQ-04, REQ-07, REQ-08

Initiating Actor: Customer

Actor's Goal: To park the customer's car in the garage.

Participating Actors: Valet, Database, Cameras, Valet Assistant Interface

Preconditions: The user has a reservation created (walk-in or online).

Postconditions: The car will be parked in the garage.

Flows of Events for Main Success Scenario:

-> 1.The customer drives up to the cameras.
<- 2.Valet notifies user to leave keys in the car and go.
-> 3.Valet triggers cameras to take photo of cars in six different angles.
<- 4.Pictures are stored in the database.
-> 5.Valet confirms security check is done.
<- 6.System notifies valet on valet assistant interface where to park
<- 7.Valet parks the car.
-> 8.Valet verifies he parked the car by pressing button on app.
<- 9.Database is updated.

Use Case UC-5:	**Overstay**
Related Requirements:	REQ-08,REQ-11,REQ-14,REQ-29
Initiating Actor:	System
Actor's Goal:	To follow the policies on overstays

Participating Actors: System, Database

Preconditions: The customer has overstayed.

Postconditions: The car will be removed or reservation is extended.

Flows of Events for Main Success Scenario:

-> 1. System updates database as soon as the customer overstays
System will know because the car has not exited
<- 2.The valet is notified about an overstay.
-> 3.The valet will follow policies on the overstay
If the customer has overstayed and the parking lot is full then the car will be towed
If the customer has overstayed and the parking lot is not full ,then the customer will simply be overcharged and the reservation will be extended

Use Case UC-6:	**Exit**
Related Requirements:	REQ-04, REQ-07, REQ-18
Initiating Actor:	Customer
Actor's Goal:	To obtain and return the car to the customer.
Participating Actors:	Valet, Database, Valet Assistant Interface
Preconditions:	The customer

has requested his car and has fully paid. Postconditions: The car will be removed from the garage. Flows of Events for Main Success Scenario: -> 1.The customer requests for the car and pays. <- 2.The valet will obtain the car and driven to the customer. -> 3.License Plate reader sees the car leaving. <- 4.Database is updated and car is returned.

Use Case UC-7: **Analyse Profits**
Related Requirements: REQ-30 Initiating Actor: Manager Actor's Goal: To view the profits associated with the parking garage. Participating Actors: Database, Website Preconditions: The manager is logged in with his account. Postconditions: The profits will be entered and updated. Flows of Events for Main Success Scenario: -> 1.Manager selects "Insert data" <- 2.The website will move to the page that allows the manager to enter income data. -> 3.The manager enters the profits in the form and hits enter.

<- 4.The manager is then forwarded to an interactive chart where he is able to analyze the data.
-> 5.The database is updated with the profits the manager has entered.

***Use Case UC-8: Manage Employee Information**

Related Requirements: REQ-30

Initiating Actor: Manager

Actor's Goal: To change employee information like salary or personal information.

Participating Actors: Database, Website

Preconditions: The manager is logged in with his account.

Postconditions: The employee information will be changed and updated.

Flows of Events for Main Success Scenario:

-> 1.Manager selects “Manage Employee Information”
<- 2.The website will move to the page that allows the manager to manage employee information.
-> 3.The manager changes the employee information online and confirms the change.
<- 4.The database is updated with a time stamp recording when the change occurred.

***Use Case UC-9: Check Statistics**

Related Requirements: REQ-30

Initiating Actor: Manager

Actor's Goal: To view statistics of the parking garage.

Participating Actors: Database, Website

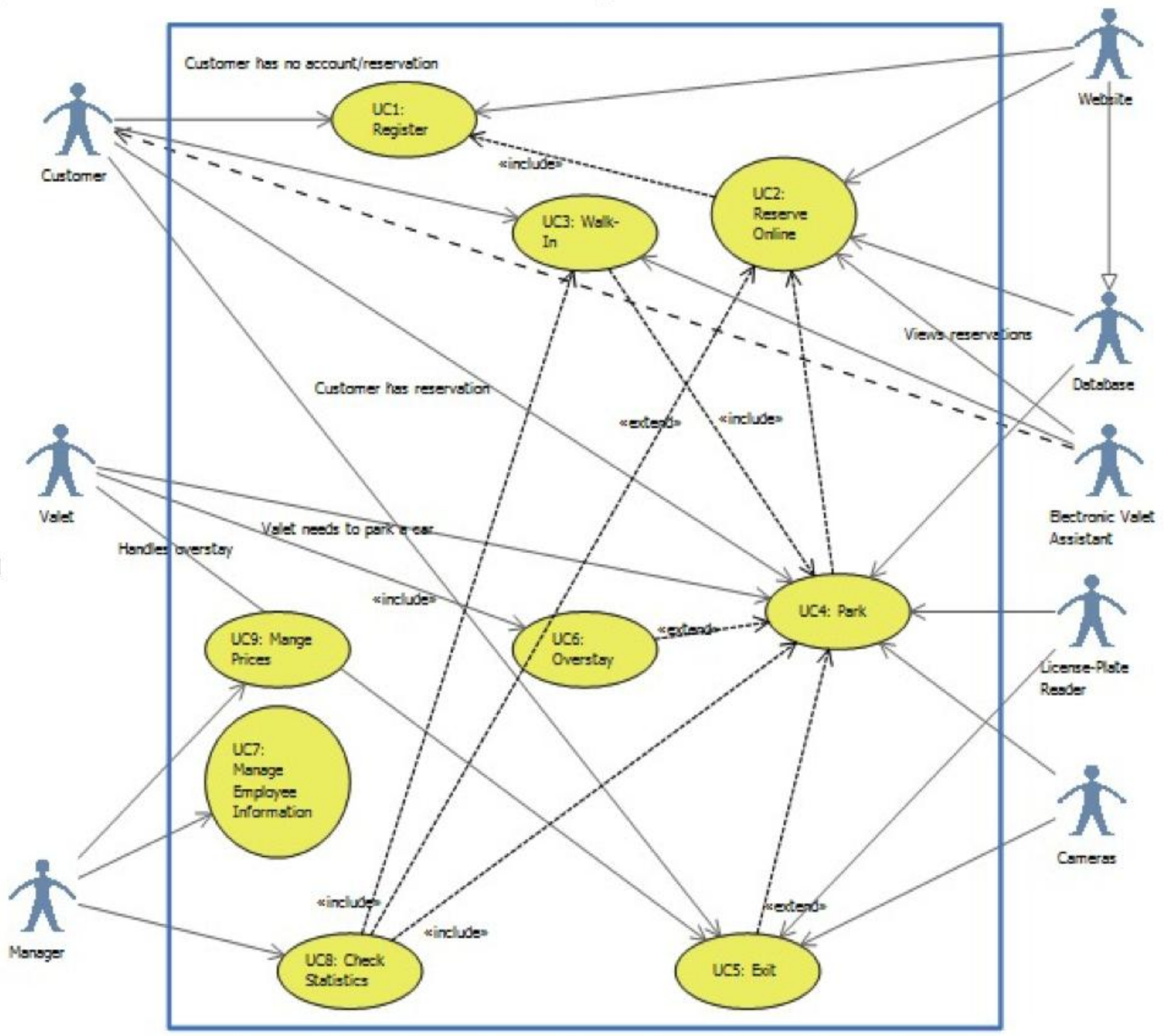
Preconditions: The manager is logged in with his account.

Postconditions: The parking garage statistics are provided to the manager.

Flows of Events for Main Success Scenario:

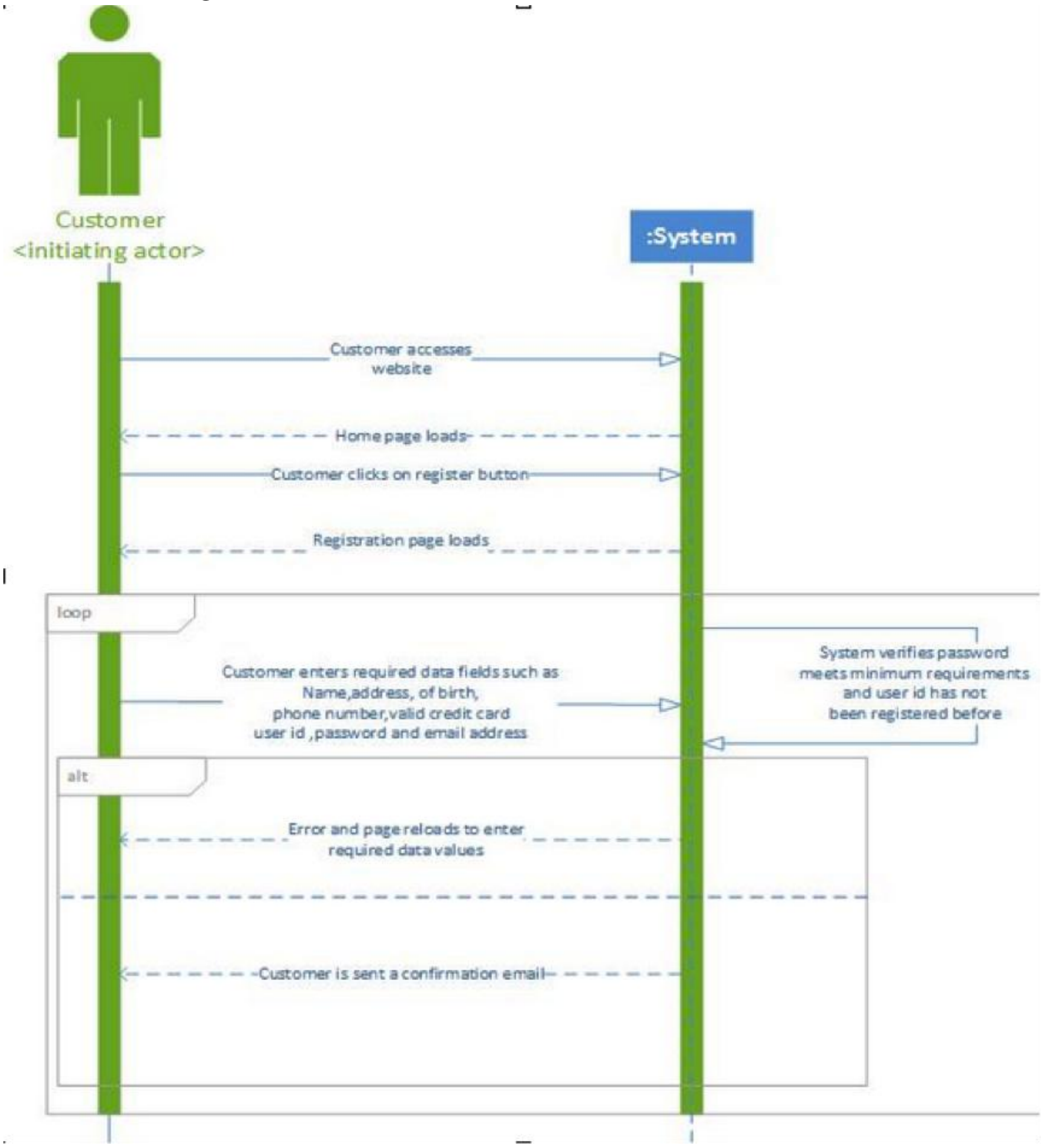
-> 1.Manager selects "Check Statistics"
<- 2.The website will move to the page that allows the manager to view parking garage statistics.

Use Case Diagram

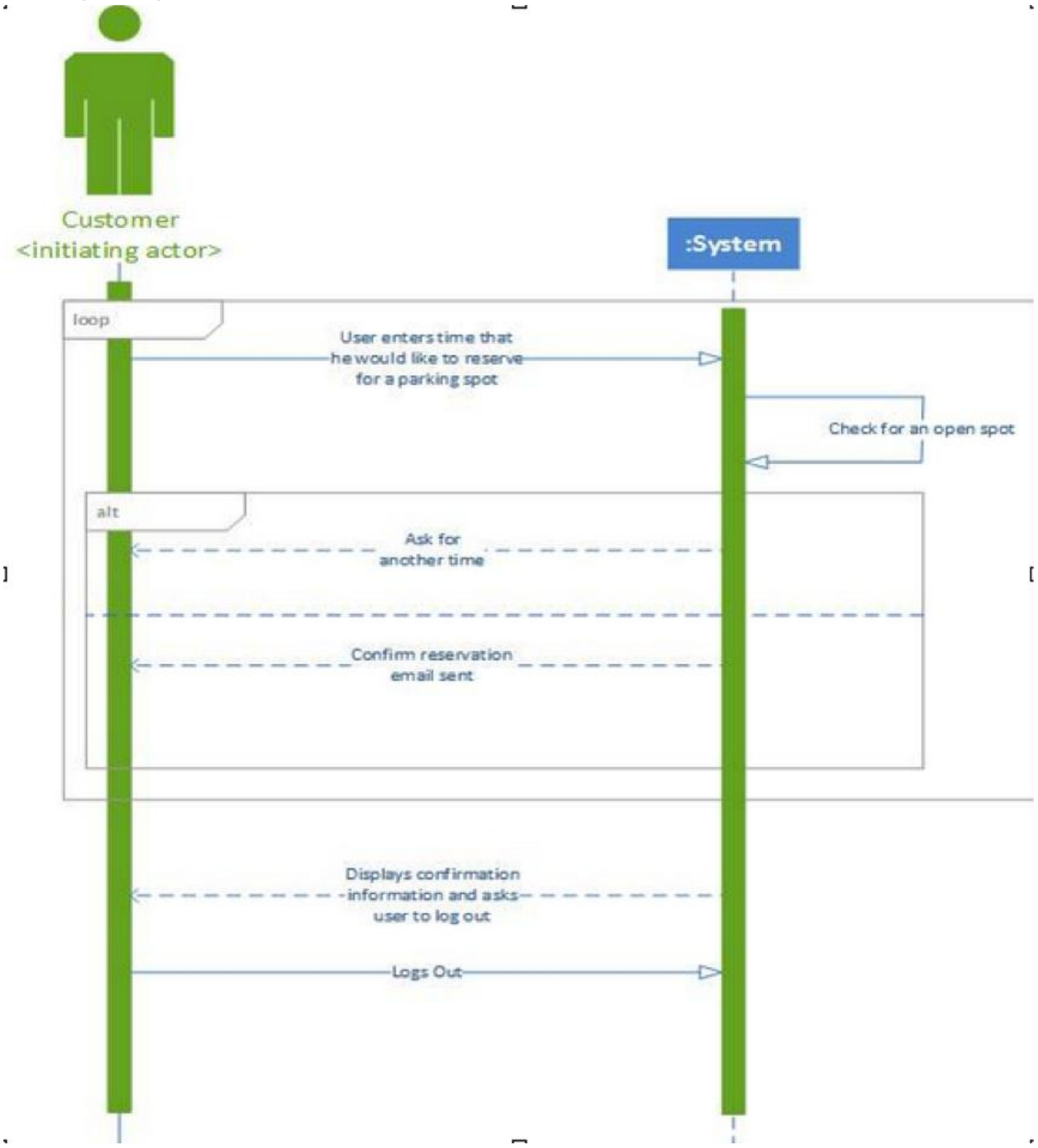


System Sequence Diagrams:

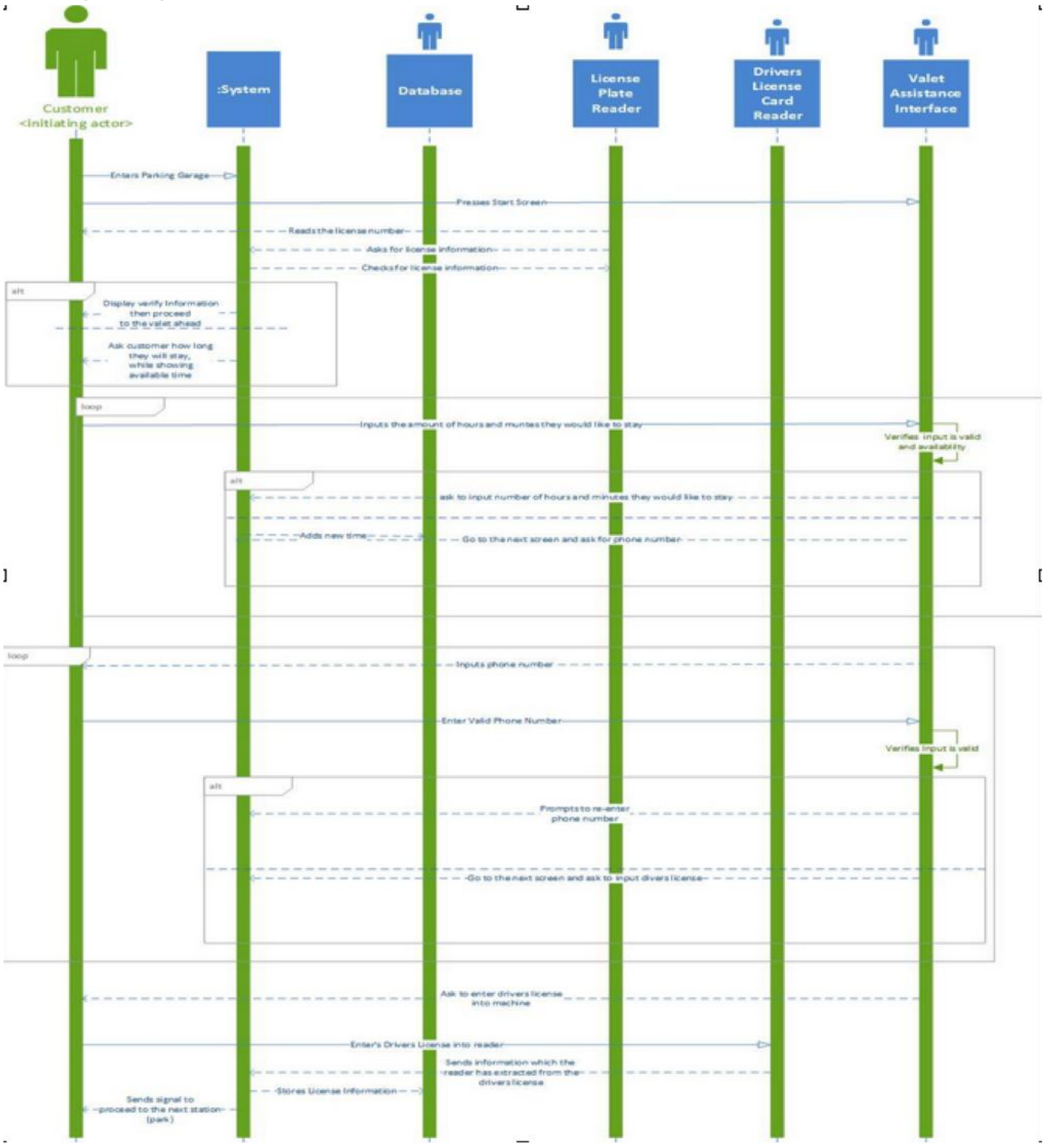
Use Case UC-1: Register


Customer
<initiating actor>
:System
Customer accesses website
Home page loads
Customer clicks on register button
Registration page loads
loop
Customer enters required data fields such as Name,address, of birth, phone number,valid credit card user id ,password and email address
System verifies password meets minimum requirements and user id has not been registered before
alt
Error and page reloads to enter required data values
Customer is sent a confirmation email

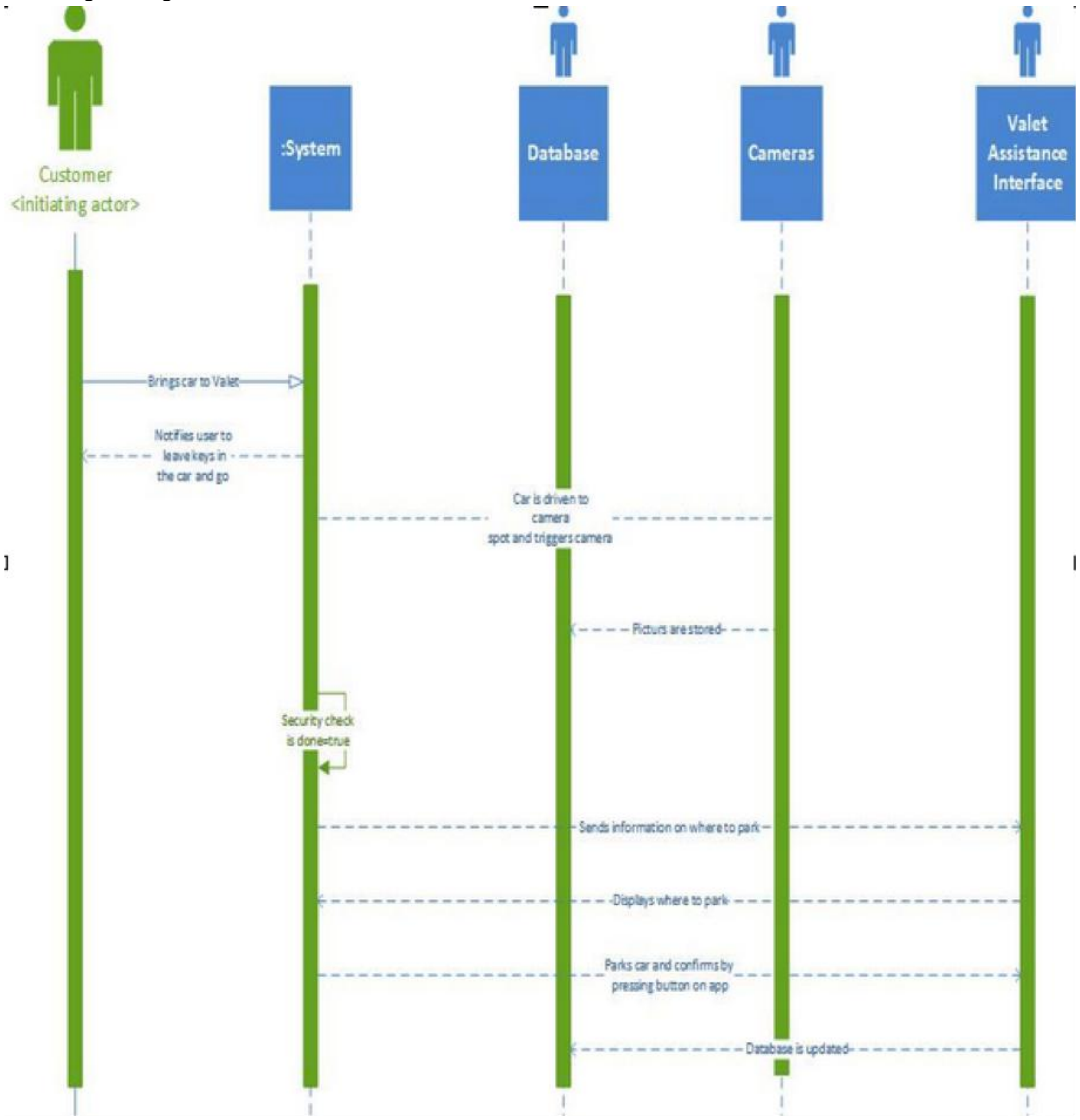
Use Case UC-2: Online Reservation


Customer
<initiating actor>
:System
loop
User enters time that he would like to reserve for a parking spot
Check for an open spot
alt
Ask for another time
Confirm reservation email sent
Displays confirmation information and asks user to log out
Logs Out

<u>Use Case UC-3: Enter Garage</u>


Customer
<initiating actor>
:System
Database
License Plate Reader
Drivers License Card Reader
Valet Assistance Interface
Enters Parking Garage
Presses Start Screen
Reads the license number
Asks for license information
Checks for license information
alt
Display verify Information then proceed to the valet ahead
Ask customer how long they will stay, while showing available time
loop
Inputs the amount of hours and muntes they would like to stay
Verifies input is valid and availability
alt
ask to input number of hours and minutes they would like to stay
Adds new time
Go to the next screen and ask for phone number
loop
Inputs phone number
Enter Valid Phone Number
Verifies input is valid
alt
Prompts to re-enter phone number
Go to the next screen and ask to input divers license
Ask to enter drivers license into machine
Enter's Drivers License into reader
Sends information which the reader has extracted from the drivers license
Stores License Information
Sends signal to proceed to the next station (park)

Use Case UC-4: Park

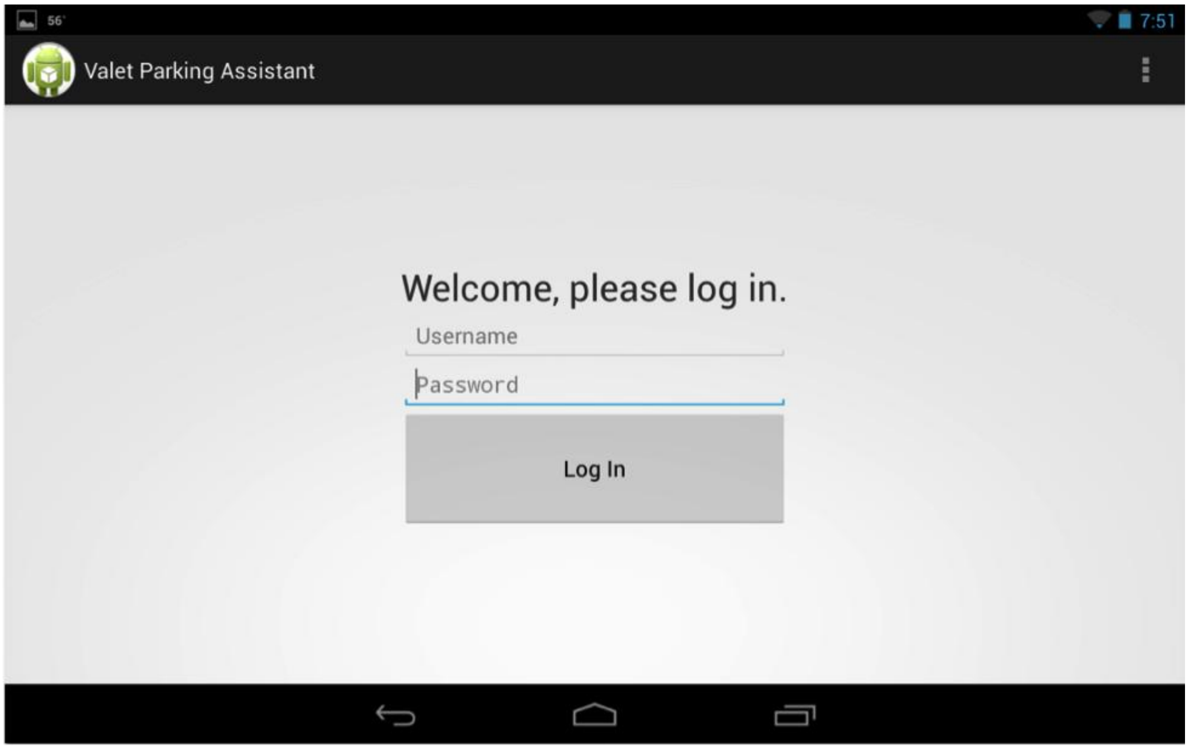

Customer
<initiating actor>
:System
Database
Cameras
Valet Assistance Interface
Brings car to Valet
Notifies user to leave keys in the car and go
Car is driven to camera spot and triggers camera
Picturs are stored
Security check is done=true
Sends information on where to park
Displays where to park
Parks car and confirms by pressing button on app
Database is updated

User Interface Specification:

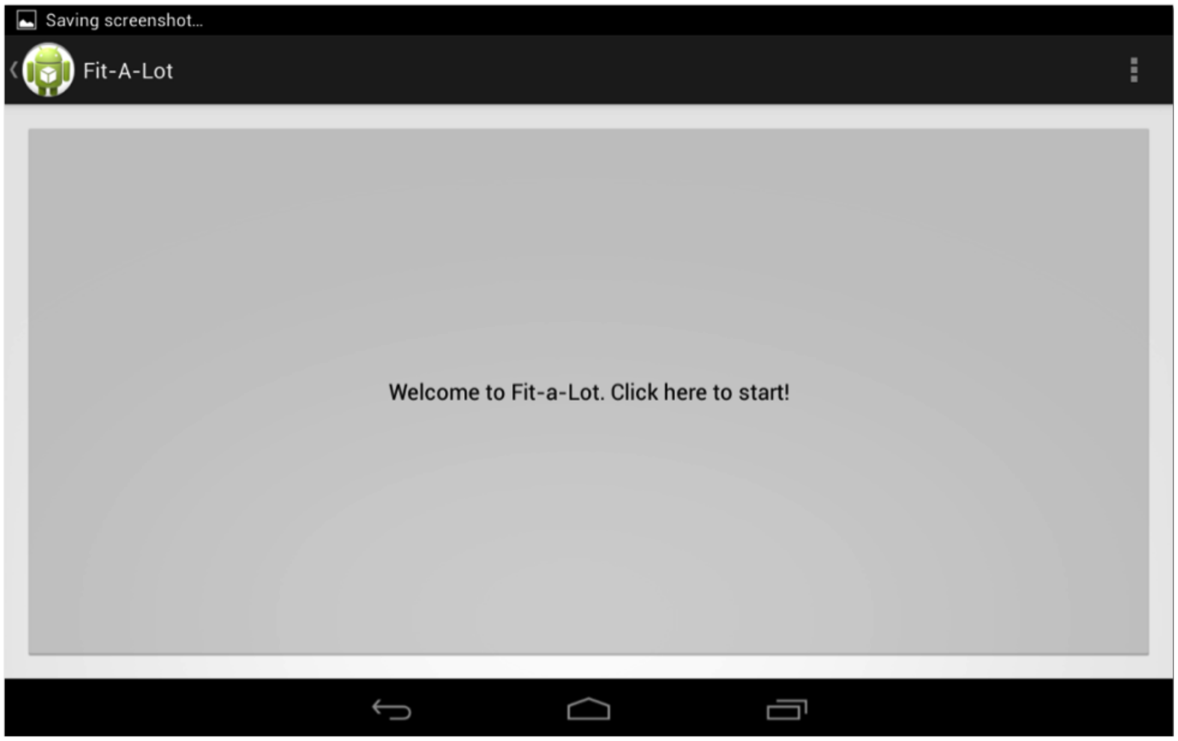
Screens were adjusted were adjusted since report 2. Here are the most recent updates to our interfaces. This doesn't include anything that will be added to them before the final demo.

Valet Assistant Interface:

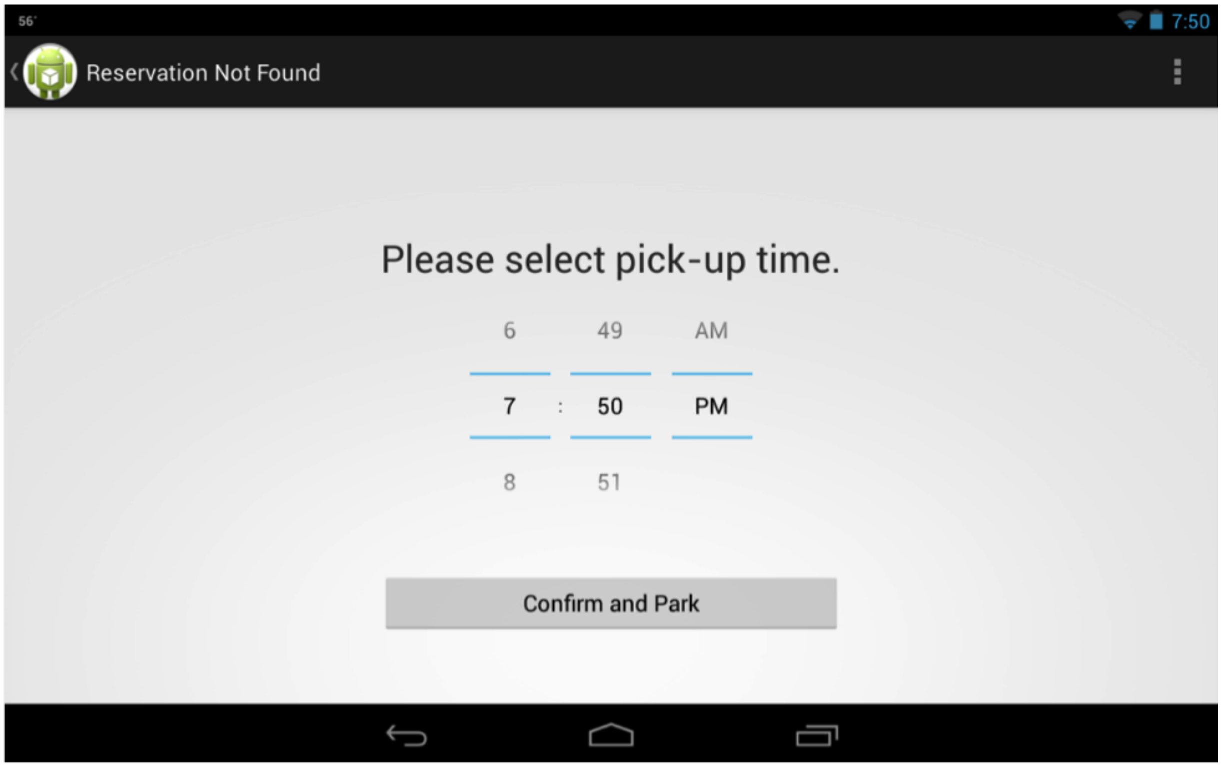
(Screen 1) This is the first screen that the customer will see. It will display "Welcome, please log in" and prompt for Username and Password. If the user is a walk-in and doesn't possess a registration, they will be taken to Screen 2. If the user made a reservation already, it will take them to Screen 6.



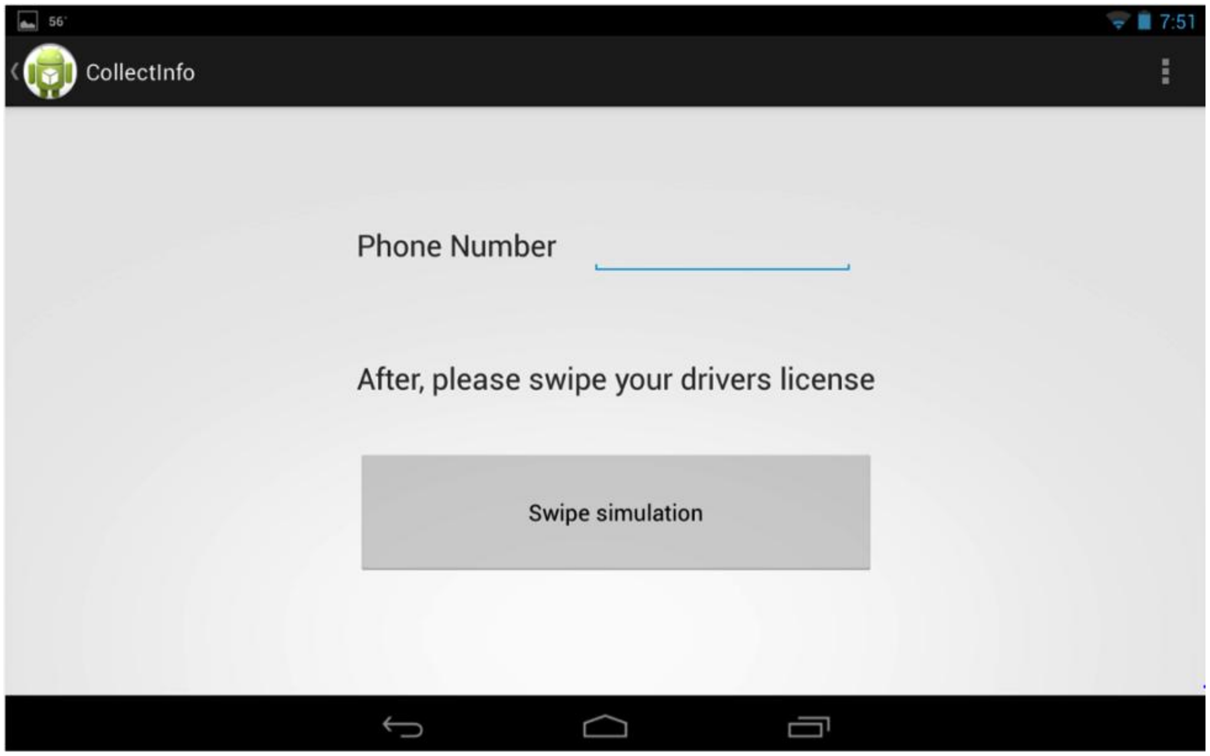
(Screen 2) This screen has been changed since report 2. It will now just display "Welcome to Fit-A-Lot, Touch to Start"



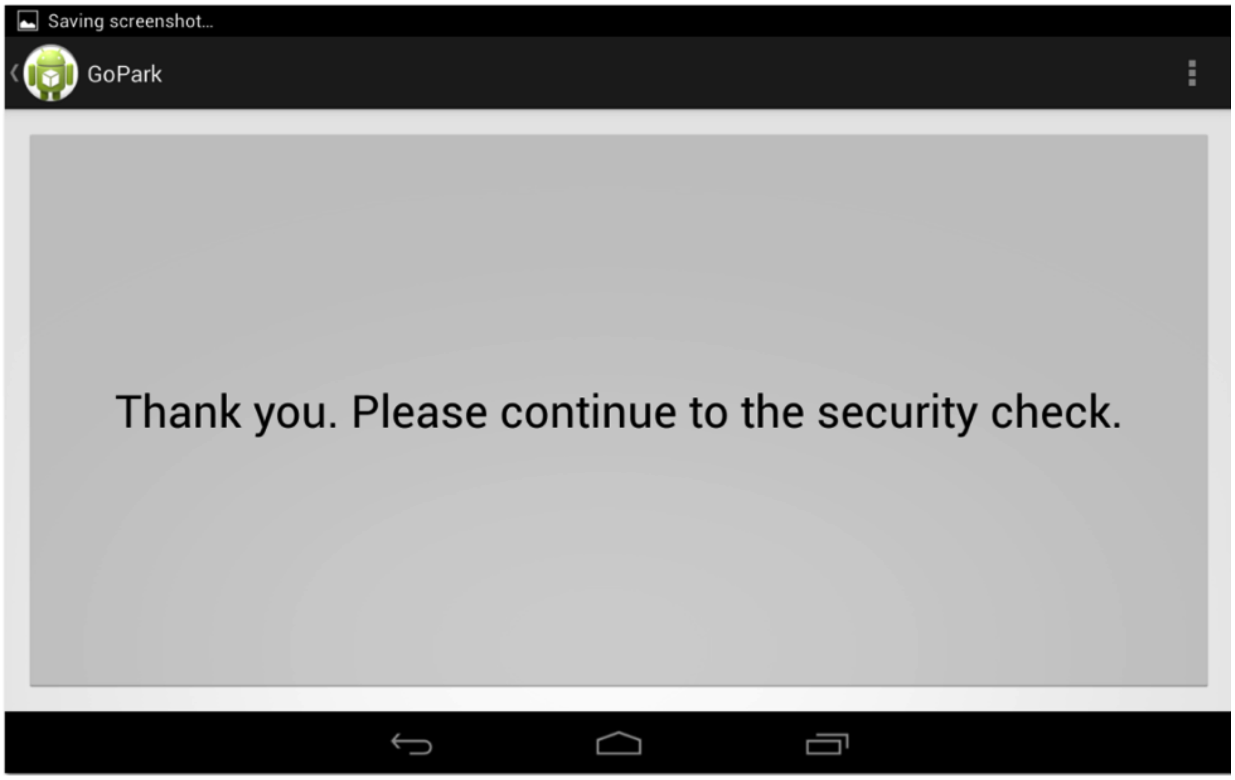
(Screen 3) This screen will display "Please Select Pick-Up Time" and will prompt the user to enter a pick up time for their vehicle. It will be a slide option when dealing with entering the time.



(Screen 4) This screen will display an input for the user's phone number. A number pad will appear that will allow for this input when you tap the blank area. Once the number is inputted, it will as the user to swipe their credit card.

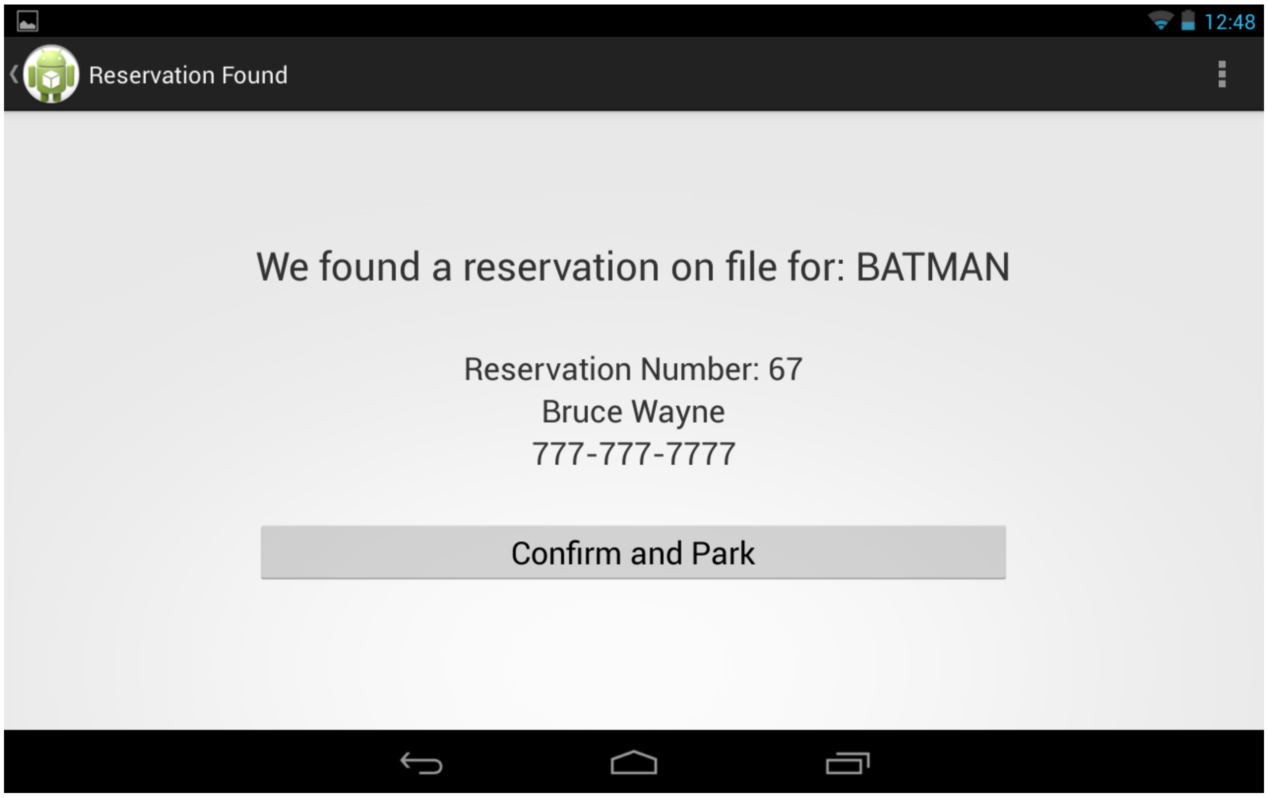


(Screen 5) This screen will display, "Thank You. Please continue to the security check."



Thank you. Please continue to the security check.

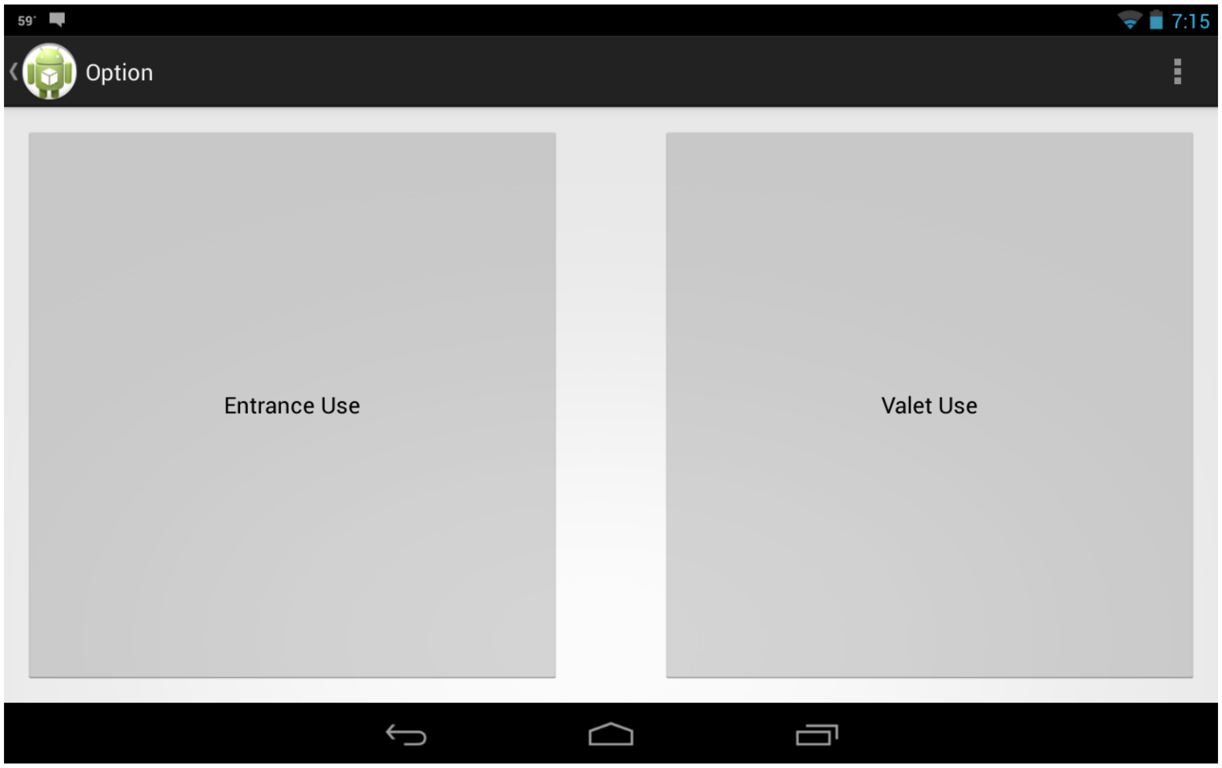
(Screen 6) This screen will display "We have found a reservation" and will also display the reservation number, First and Last Name, and Phone Number.



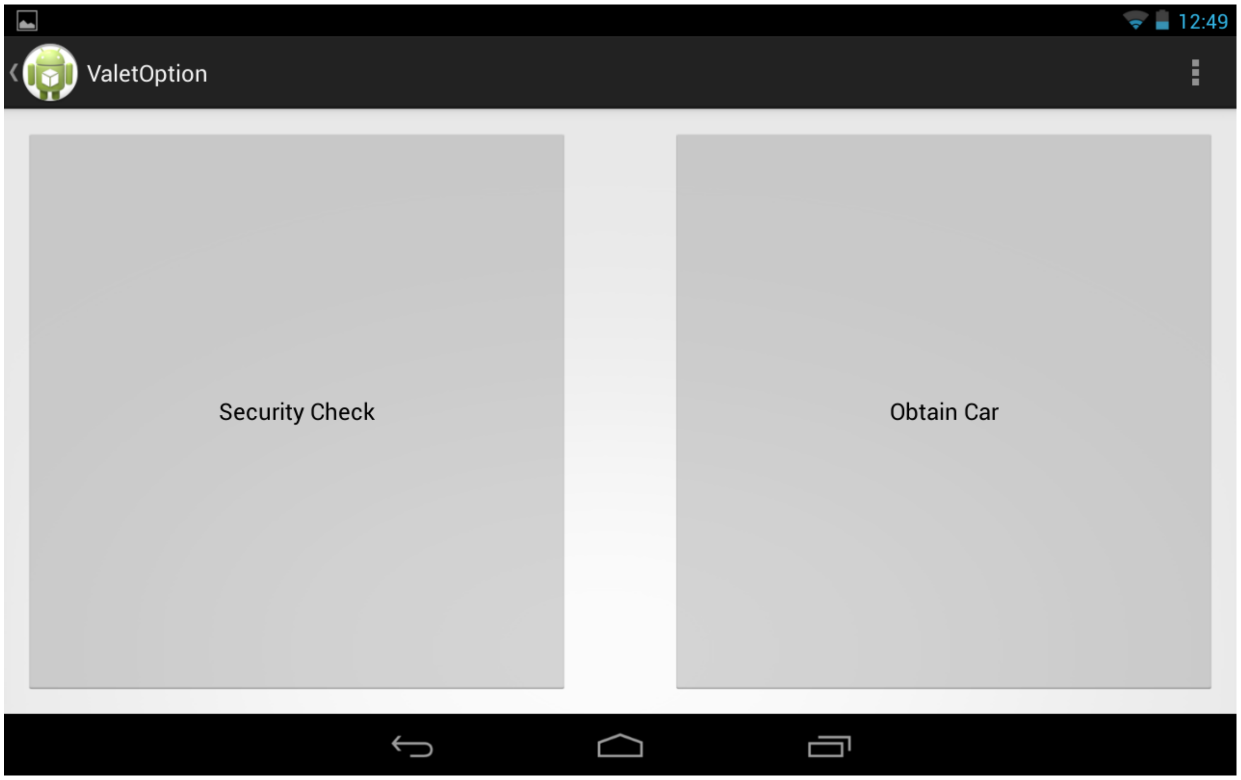
We found a reservation on file for: BATMAN

Reservation Number: 67
Bruce Wayne
777-777-7777

Newer implementations added after Demo 1 was:

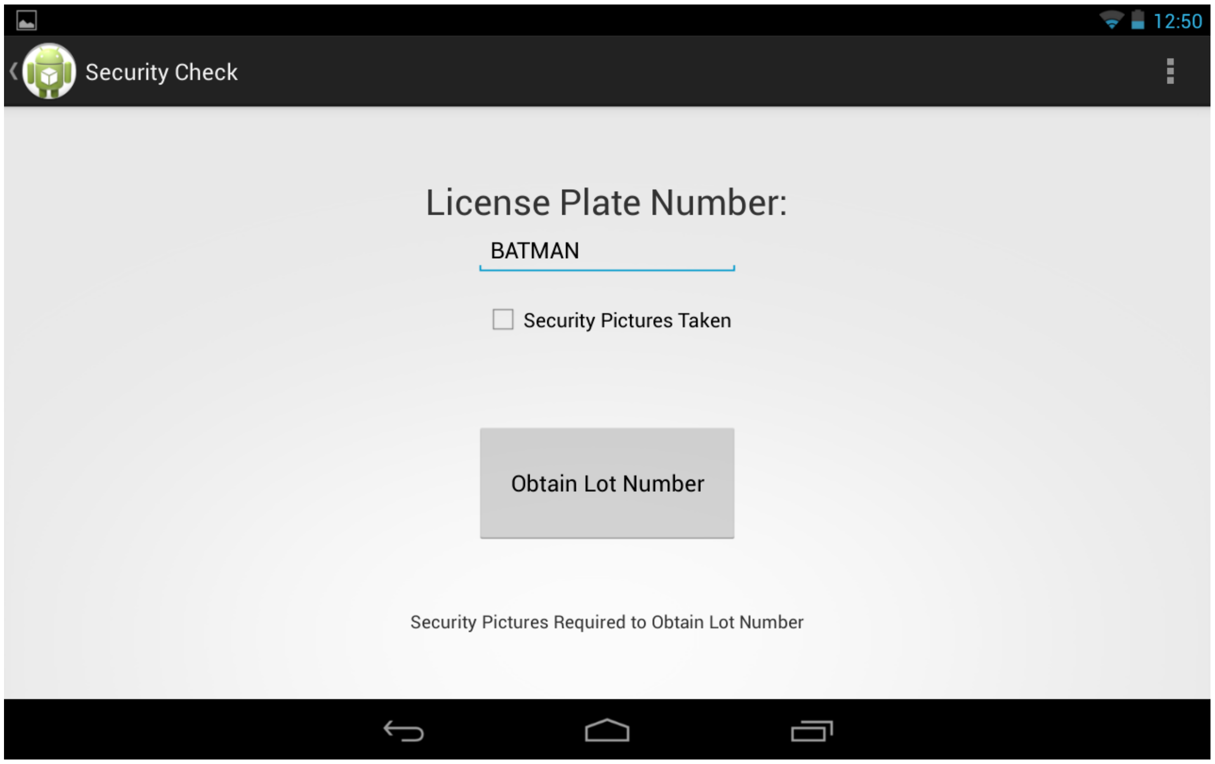


This shows after log-in and allows the user to select which feature they want to use the entrance use shown above or the valet aspect of the application.



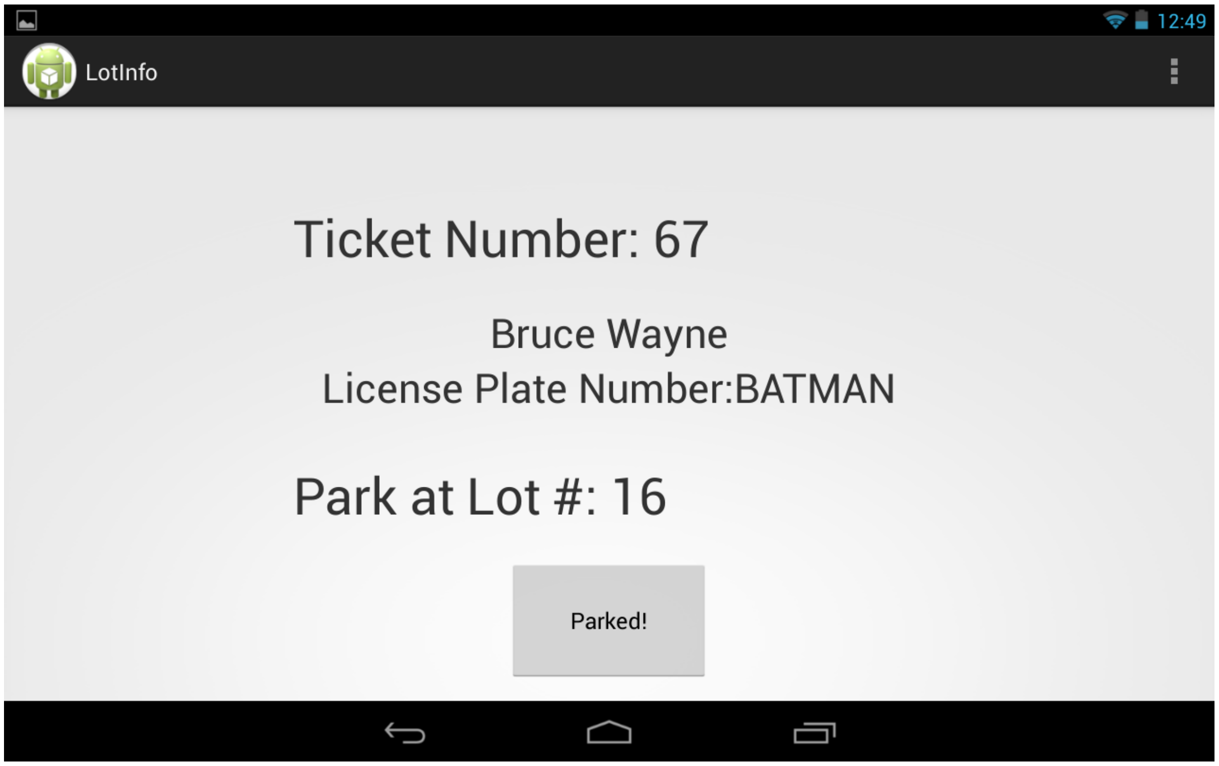
This allows the valet to select which feature he needs to use: Security checking or Obtaining a car.

The valet has a feature to handle the security check after the click of valet use:



Security Pictures Required to Obtain Lot Number

The valet selects a car to help by inputting the license plate number (this just for testing purposes, the real implementation would be using states to saying that the car has completed entering the garage and is at the security check state. At this point the valet can select which car to serve instead of typing the license plate number). There is an error check to make sure that the security pictures are taken before the button is pressed. The error is shown under the check box. When the box is checked the button will become functional and display this the message above.



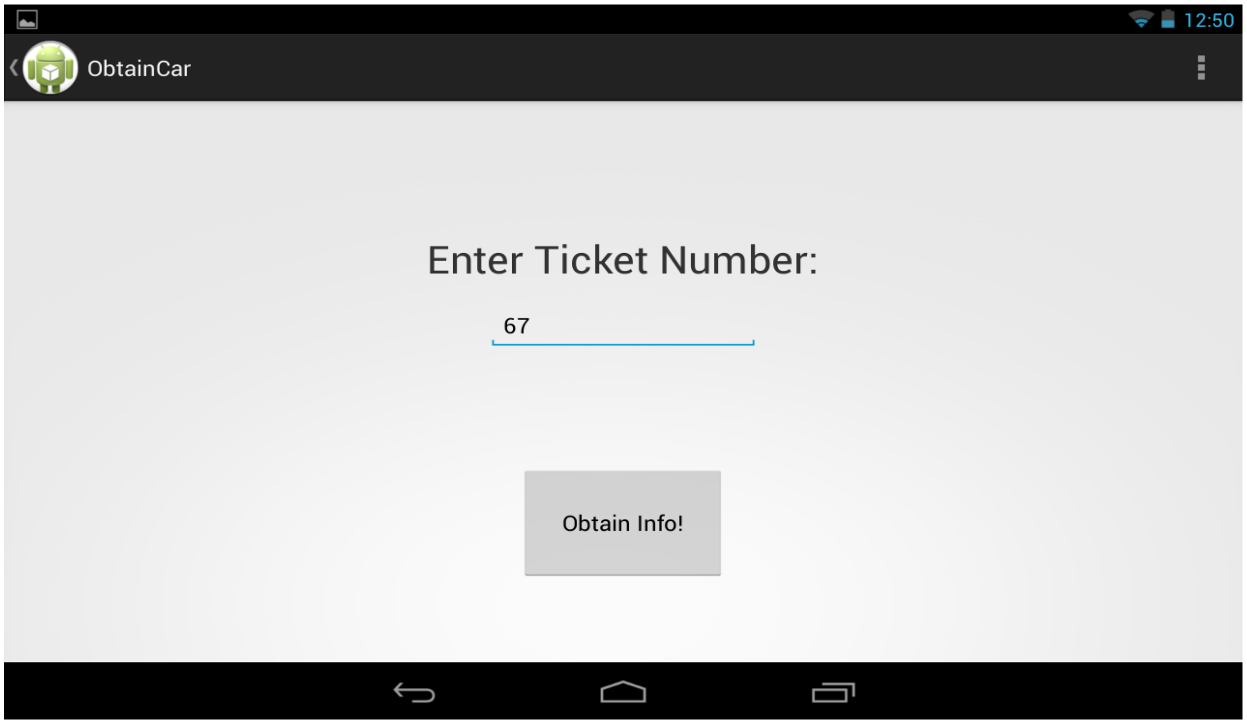
Ticket Number: 67

Bruce Wayne
License Plate Number:BATMAN

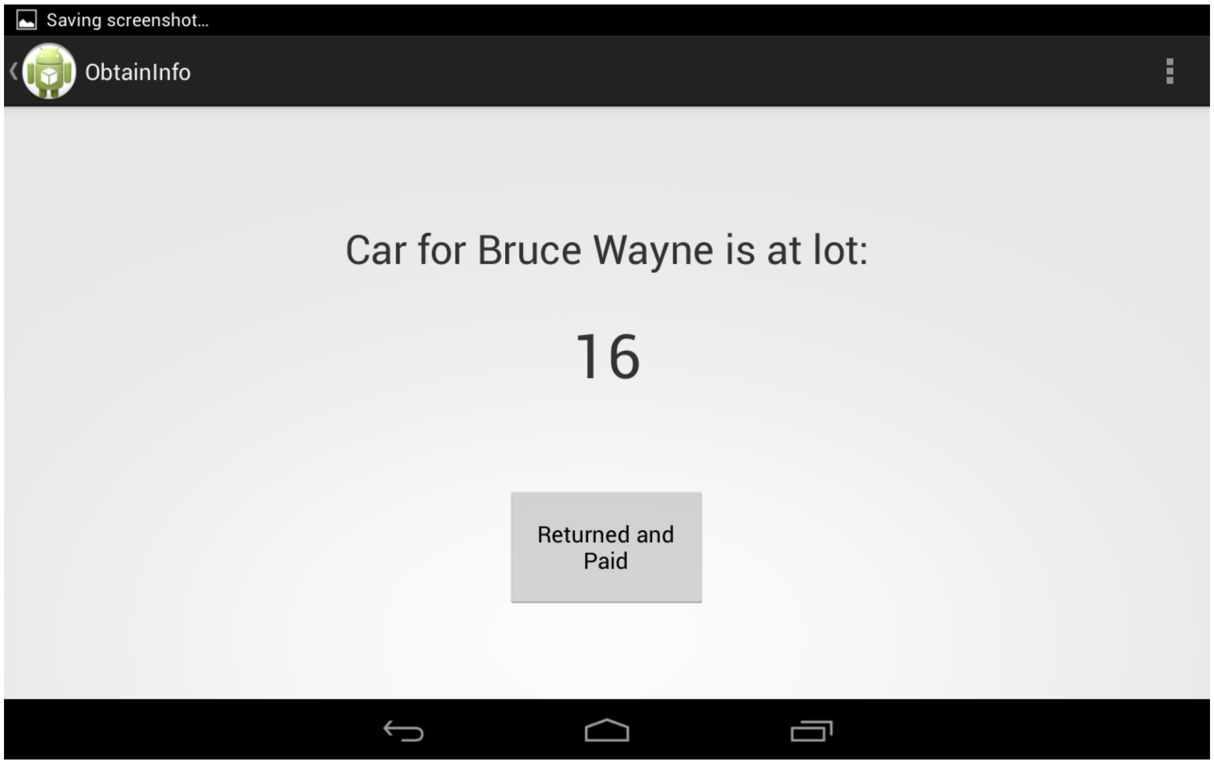
Park at Lot #: 16

This screen will display the user's ticket number. Also, the name and license plate for verification, and also what lot to park for the valet. The valet will then press the parked button and will change the state of the car to parked.

The obtain feature looks like this:



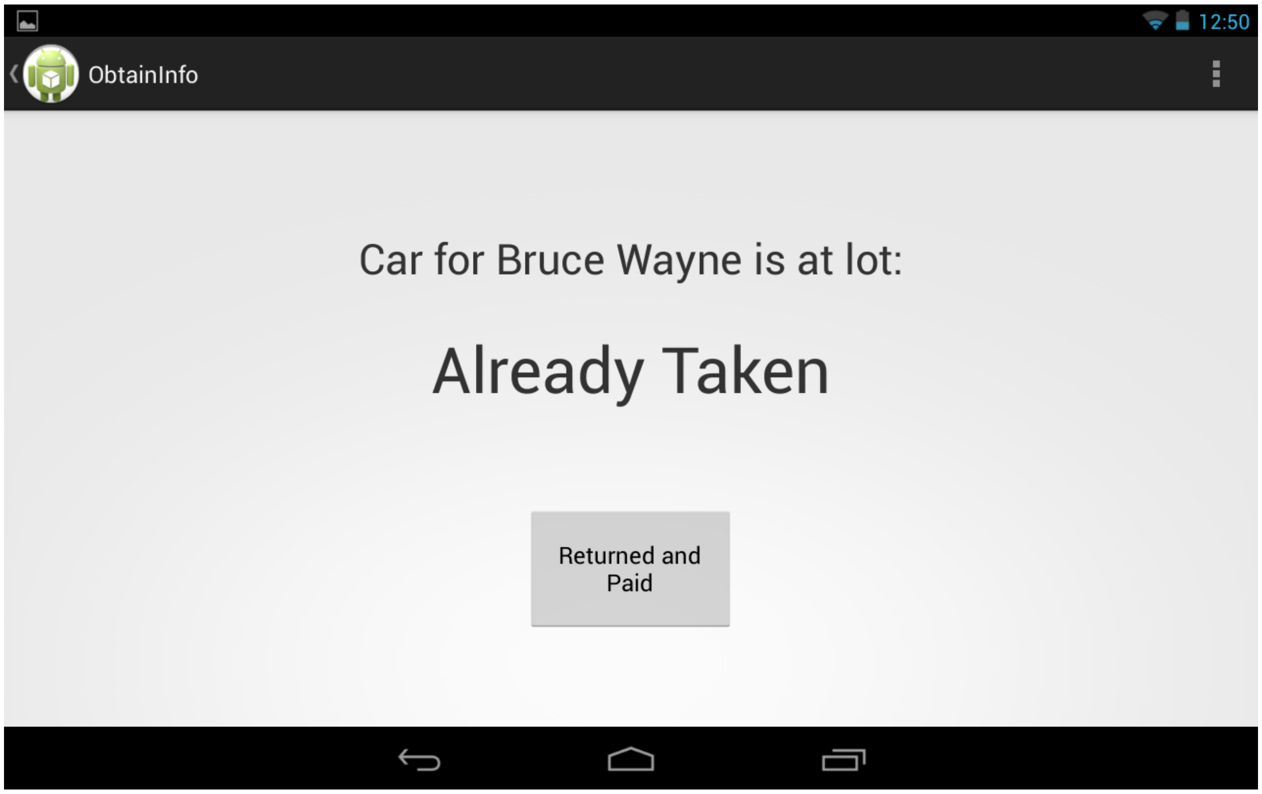
The valet would enter in the ticket number provided by the customer. Then it will display where to retrieve the car:



Car for Bruce Wayne is at lot:

16

If the car is already taken there would be a display message that says:

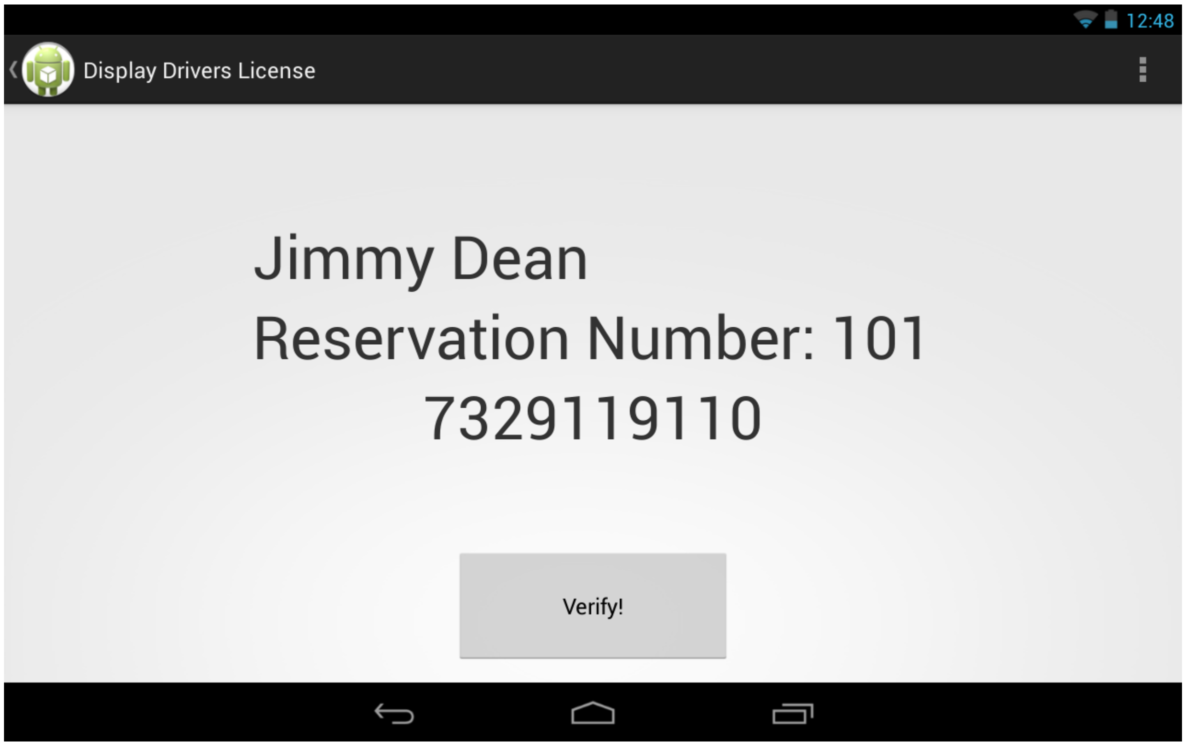


Car for Bruce Wayne is at lot:

Already Taken

This occurs when the button returned and paid has already been pressed and a state updates in the system that the car is gone.

For walk in we also created a simulation for one person:



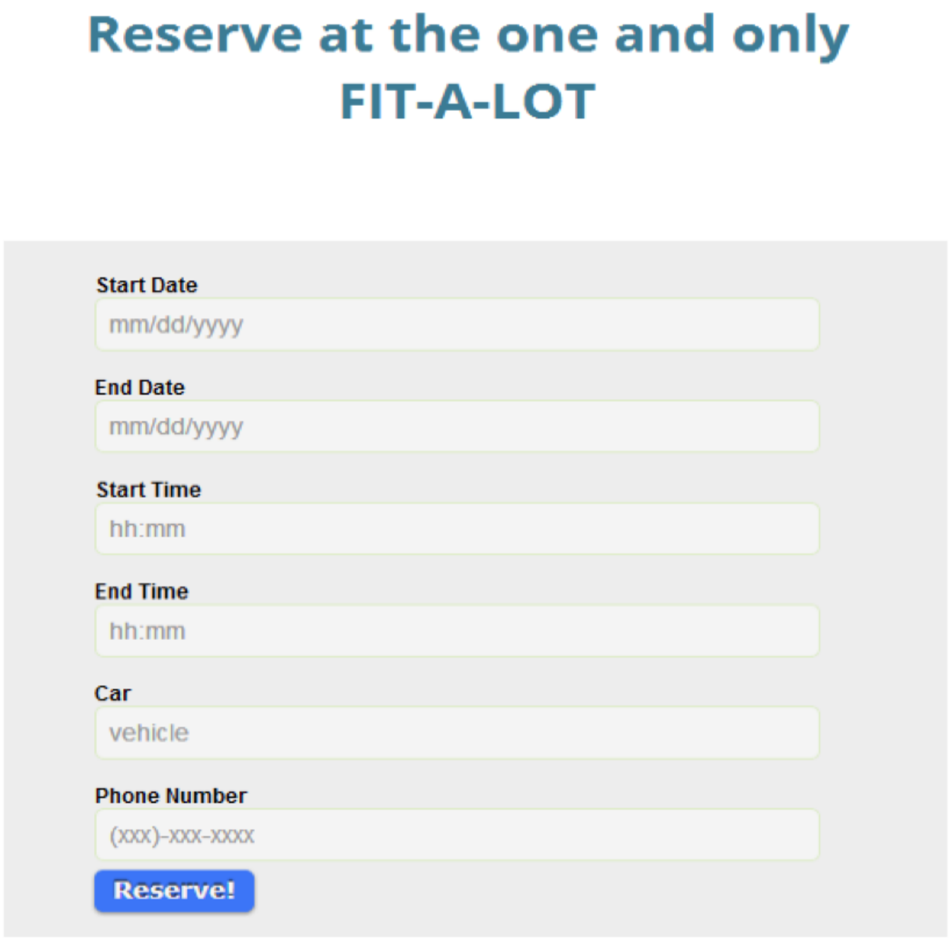
Jimmy Dean
Reservation Number: 101
7329119110

When the swipe simulation button is pressed Jimmy's information is displayed and he is provided a reservation number. The process is the same as with the fake character we created, Bruce Wayne.

Customer Website Interface:

1 User visits website homepage
   a If the user has an account, he/she may log-in.
      i He/she may log-in if he/she can provide the account ID and password
   b If the user does not have an account, he/she may create one
      i He/she may register if he/she provides last and first name, email address, date of birth, home address, phone number, etc.
   c If the user selects log-out at any given time, he/she will be redirected to the homepage with his/her account logged-out
2 User will have ability to edit or delete account once logged-in
   a If the user selects to delete, they will be prompted with a confirmation message first, then action will be enacted
   b If the user selects to edit, the account profile page will go from static, to editable and will save changes once user clicks the save button
3 Users will have ability to reserve parking
   a If the user has an account, they will be asked to log-in and the reservation page will be auto-filled in the background.

   b. If the user does not have an account, they will be asked to submit information similar to that of registration (see preliminary figure 1.1)
   c. Once all information is submitted, either manually or automatically, the user will be asked to submit credit-card information to finalize reservation
   d. Once reservation is finalized, the user will be given a confirmation number
4. Users will have ability to cancel or modify reservations
   a. The user will be asked for his/her confirmation number
   b. Once number is inputted correctly, user is asked if he/she wishes to cancel reservation or modify reservation
   c. If the user selects to cancel, they will be prompted with a confirmation message first, then action will be enacted
   d. If the user selects to modify, the reservation information will be displayed and editable and will save changes once user clicks the save button



Reserve at the one and only
FIT-A-LOT

Start Date
mm/dd/yyyy

End Date
mm/dd/yyyy

Start Time
hh:mm

End Time
hh:mm

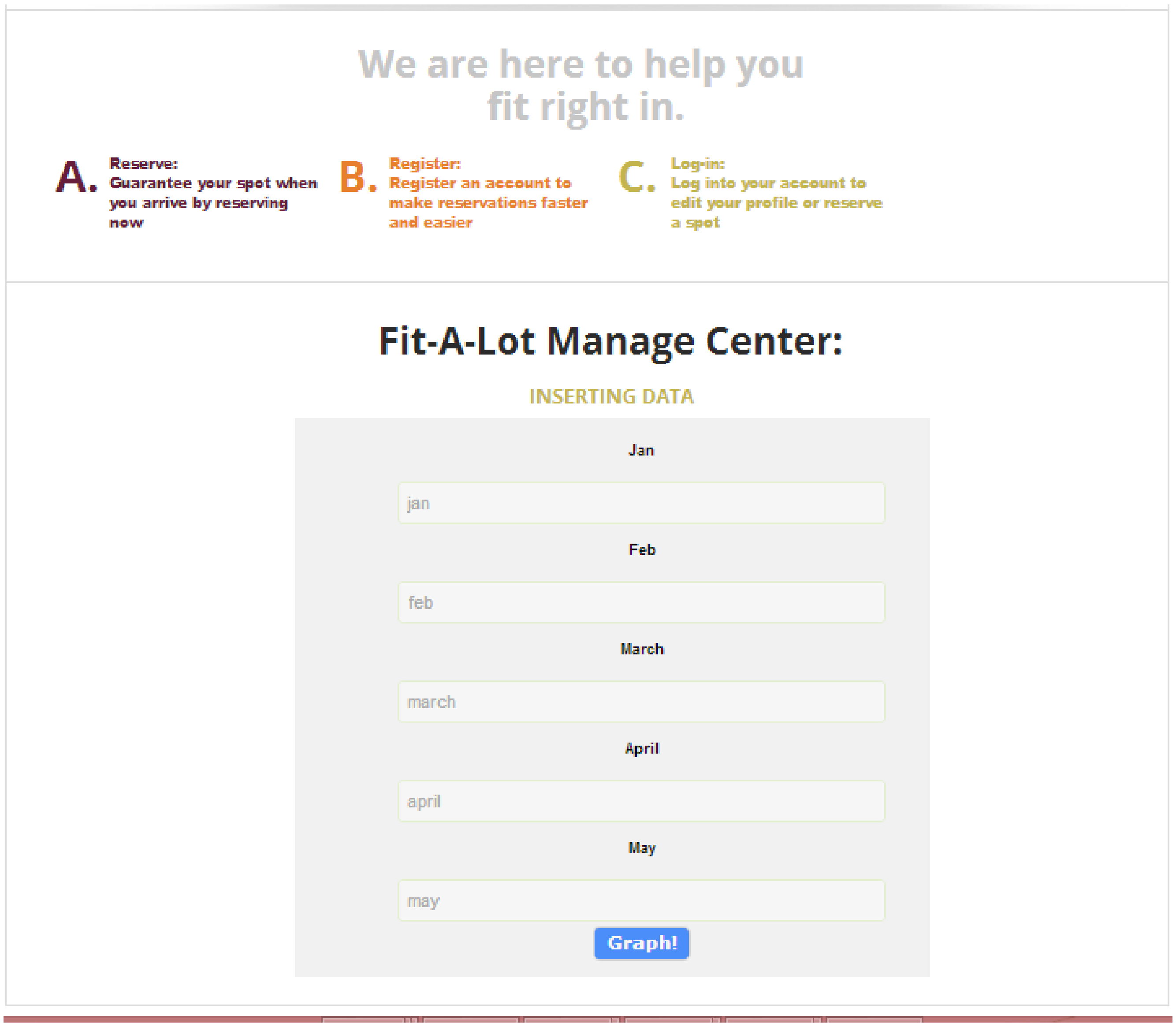
Car
vehicle

Phone Number
(xxx)-xxx-xxxx

Reserve!

Figure 1.1 shows a user reserving a spot without an account.It also captures some of the other features such as viewing account, logging-out, and registration.

Data Analysis:


We are here to help you fit right in.

A. Reserve: Guarantee your spot when you arrive by reserving now

B. Register: Register an account to make reservations faster and easier

C. Log-in: Log into your account to edit your profile or reserve a spot

Fit-A-Lot Manage Center:

INSERTING DATA

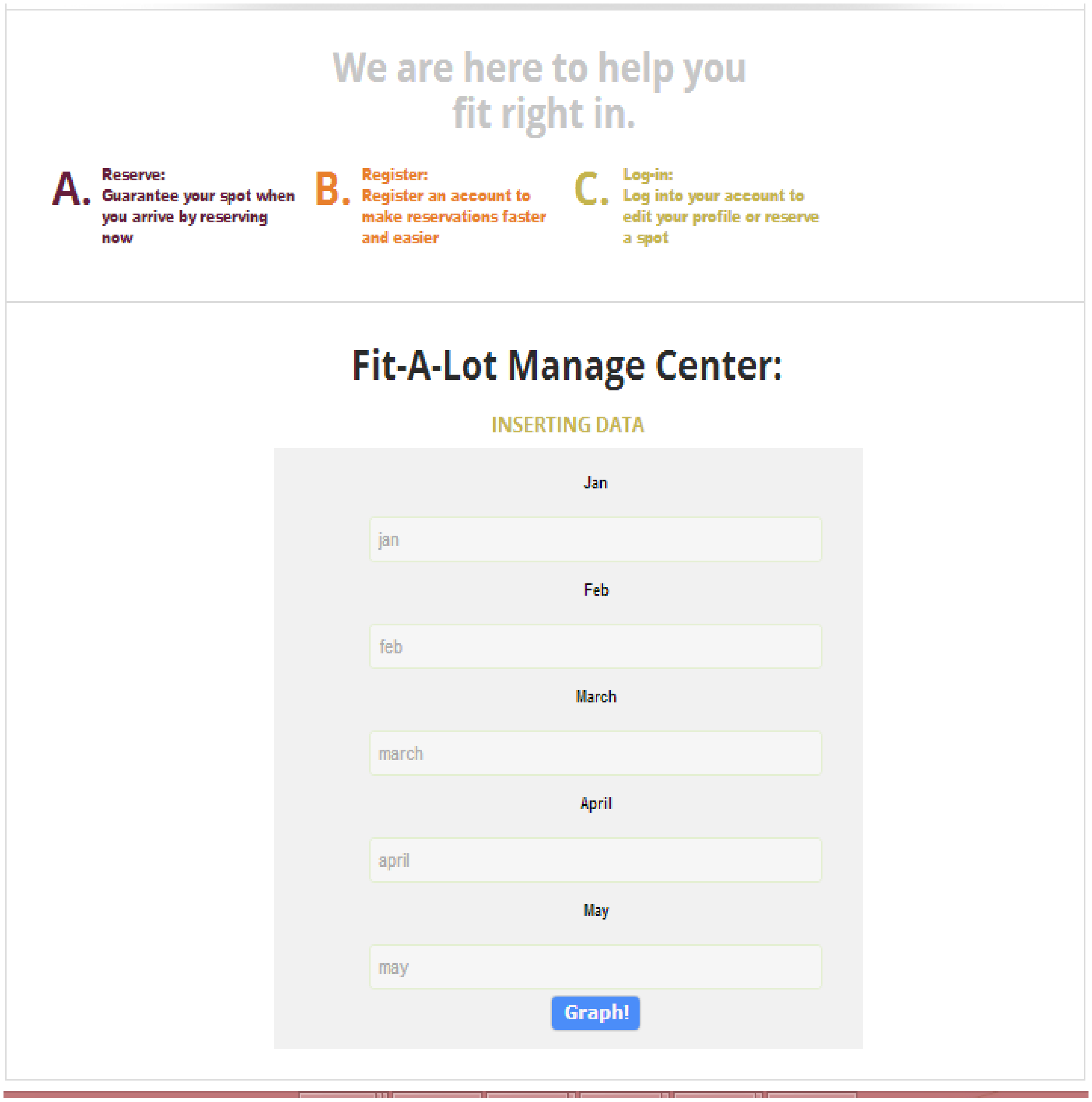
Jan
jan

Feb
feb

March
march

April
april

May
may


We are here to help you fit right in.

A. Reserve: Guarantee your spot when you arrive by reserving now

B. Register: Register an account to make reservations faster and easier

C. Log-in: Log into your account to edit your profile or reserve a spot

Fit-A-Lot Manage Center:

INSERTING DATA

Jan	jan
Feb	feb
March	march
April	april
May	may

Figure 1.2 shows a form when the manager is logged in. The form is utilized to enter data which the manager has recorded to improve profits.

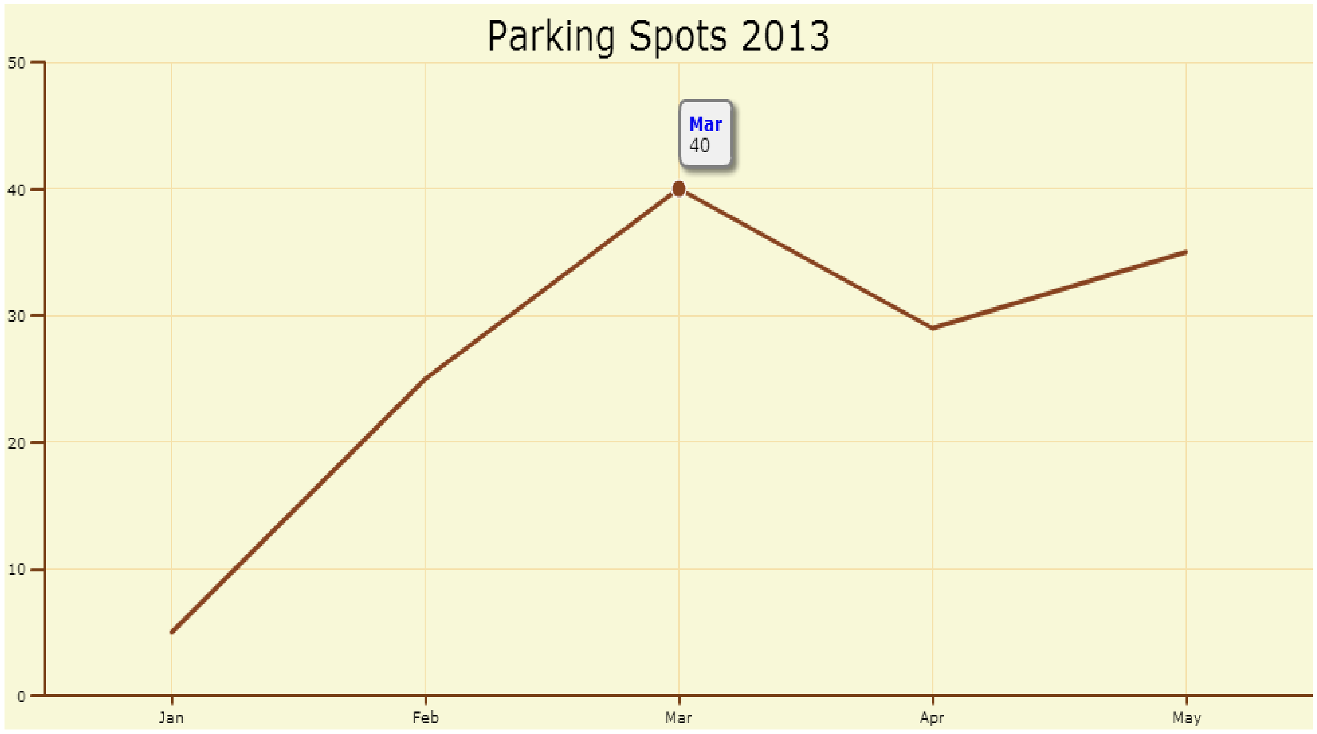
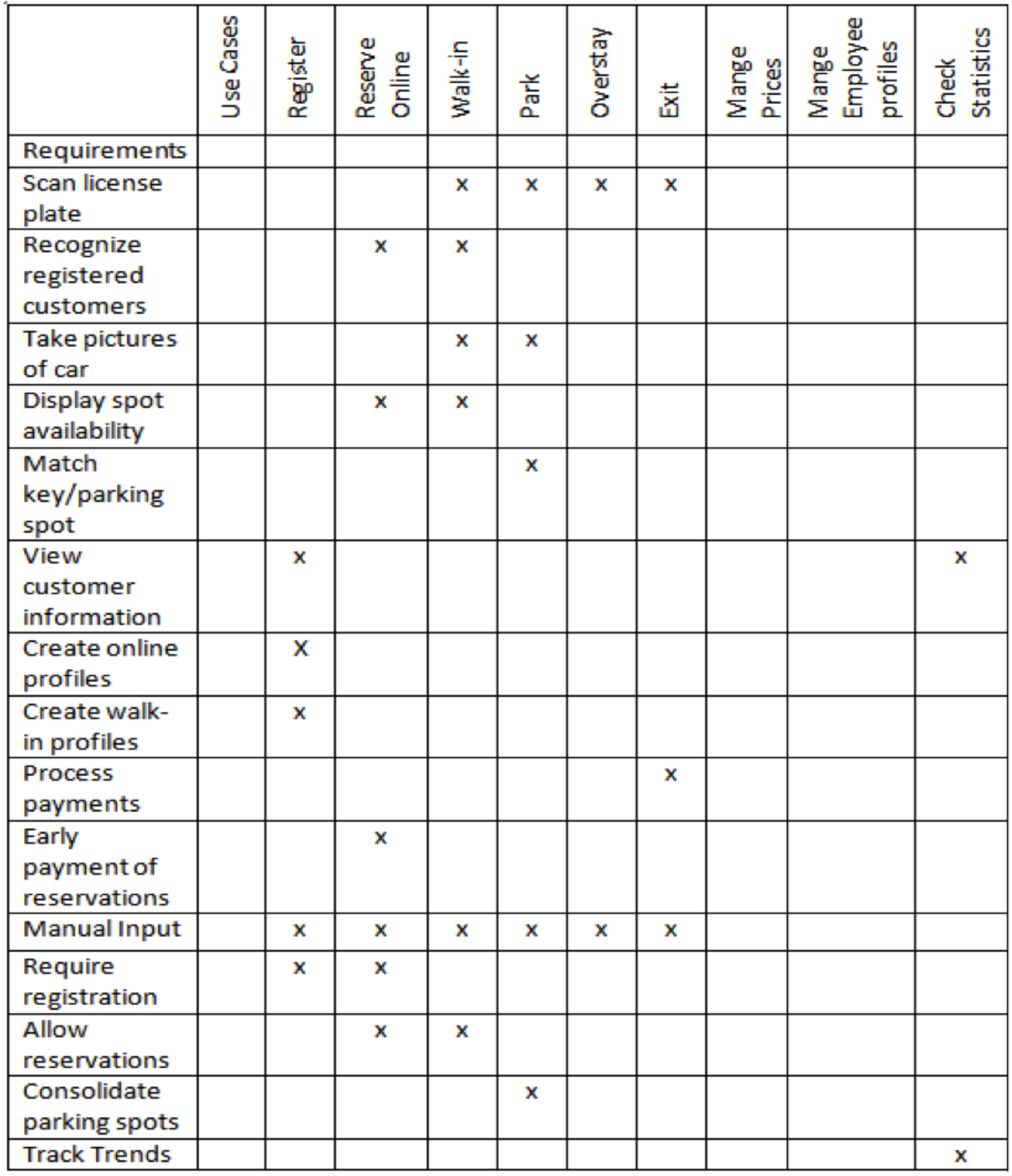


Figure 1.3 shows a graph which the manager may utilize to increase profits at the parking garage

Traceability Matrix:



	Use Cases	Register	Reserve Online	Walk-in	Park	Overstay	Exit	Mange Prices	Mange Employee profiles	Check Statistics
Requirements										
Scan license plate				x	x	x	x			
Recognize registered customers			x	x						
Take pictures of car				x	x					
Display spot availability			x	x						
Match key/parking spot					x					
View customer information		x								x
Create online profiles		X								
Create walk-in profiles		x								
Process payments							x			
Early payment of reservations			x							
Manual Input		x	x	x	x	x	x			
Require registration		x	x							
Allow reservations			x	x						
Consolidate parking spots					x					
Track Trends										x

Effort Estimation for Use Case Points

The Valet Assistant Interface minimizes the amount of strokes a customer has to use by using one stroke to start the parking process; using a keypad to type on average 3 strokes for number of hours and minutes the user would like to stay; 9 strokes for the phone number; and 1 stroke using the driver's license reader to obtain information.

Registering an account on the website keeps the amount of clicking to a minimum.It an average of 15-20 keypad strokes for the First and Last Name; 10 strokes for the phone number; average of 15-20 strokes for email address; 14-16 strokes for the credit card number; 10-20 strokes for the address; and 8 strokes for entering the birthdate; and 1 stroke to confirm.

Reserving a spot online minimizes the amount of strokes by using 1 stroke to start the reservation; 1 stroke to select "Register" or "Already have an account"; average of 3 strokes for number of hours and minutes the user would like to stay; 9 strokes for the phone number, 14-16 strokes for the credit card number; 1 stroke for completing; and 1 stroke for confirmation.

The manager can access the database to find out logistics and trends with minimum amount of strokes by first using 1 stroke to select the administrator option; an average of 10-30 strokes depending on the login information; 1 stroke to select data type, then depending on what the manager selects, the amount of strokes are different:

- Manage Prices (1 stroke to select the option; 1 stroke to select either "Adjust/ Hour rates", "Adjust Possible Deals", or 1-2 strokes to select other options; 4-8 strokes to enter/adjust prices.)
- Manage Employee Information (1 stroke to select the option; 1 stroke to select type of Employee; 1 stroke to select the Employee, 1 stroke to select "Employee Data" or "Manage Pay". If Employee Data, 1 or 2 clicks to browse data. If Manage Pay, 5-6 strokes to enter pay amount and 1 stroke to confirm.)
- Check Statistics (1 stroke to select the option, 1 stroke to choose what type of trends to observe; 1-2 strokes to analyze the statistics in the database; 1 stroke to exit the option)
- After the manager is complete with their action, it takes 1 stroke to log out.

Technical Complexity:

Technical Factor	Description	Weight	Perceived Complexity	Calculated Factor
T1	Distributed system	2	1	2*1 = 2
T2	Performance objectives	1	2	1*2 = 2
T3	End-user efficiency	1	3	1*3 = 3

T4	Complex internal processing	1	2	1*2 = 2
T5	Reusable design or code	1	1	1*1 = 1
T6	Easy to install	0.5	3	0.5*3 = 1.5
T7	Easy to use	0.5	5	0.5*5 = 2.5
T8	Portable	2	4	2*4 = 8
T9	Easy to change	1	2	1*2 = 2
T10	Concurrent use	1	3	1*3 = 3
T11	Special security features	1	4	1*4 = 4
T12	Provides direct access for third parties	1	4	1*4 = 4
T13	Special user training facilities are required	1	2	1*2 = 2
	Technical Factor Total:			**37**

TCF = C_1*C_2*(Technical Factor Total) = (0.6)(0.01)(37) = **0.222**

UUCW:

Use Case	Description	Category	Weight
Register (UC-1)	Average User Interface. 5 steps for main success scenario. Two participating Actors (Website, Database)	Average	10
Online Reservation (UC–2)	Simple User Interface. 3 steps for main	Simple	5

	success scenario. 2 participating actors (Website, Database)		
Enter Garage (UC-3)	Complex User Interface. 9 steps for main success scenario. 5 participating actors (Customer, Database, License Plate Reader, License Card Reader, Valet Assistant Interface)	Complex	15
Park (UC-4)	Complex User Interface. 9 steps for main success scenario. 4 participating actors (Valet, Database, Cameras, Valet Assistant Interface)	Complex	15
Overstay (UC-5)	Simple User Interface. 3 steps for main	Simple	5

	success scenario. 2 participating actors (System, Database)		
Exit (UC-6)	Average User Interface. 4 steps for main success scenario. 3 participating actors (Valet, Database, Valet Assistant Interface)	Average	10
Manage Prices (UC-7)	Average User Interface. 4 steps for main success scenario. 2 participating actors (Website, Database)	Average	10
Manage Employee Information (UC-8)	Average User Interface. 4 steps for main success scenario. 2 participating actors (Website, Database)	Average	10
Check Statistics (UC-9)	Simple User Interface. 2 steps for	Simple	5

	main success scenario. 2 participating actors (Website, Database)		

UUCW (Parking Garage) = 3 x Simple + 4 x Average + 2 x Complex = 3 x 5 + 4 x 10 + 2 x 15 = **85**

Environmental Factors:

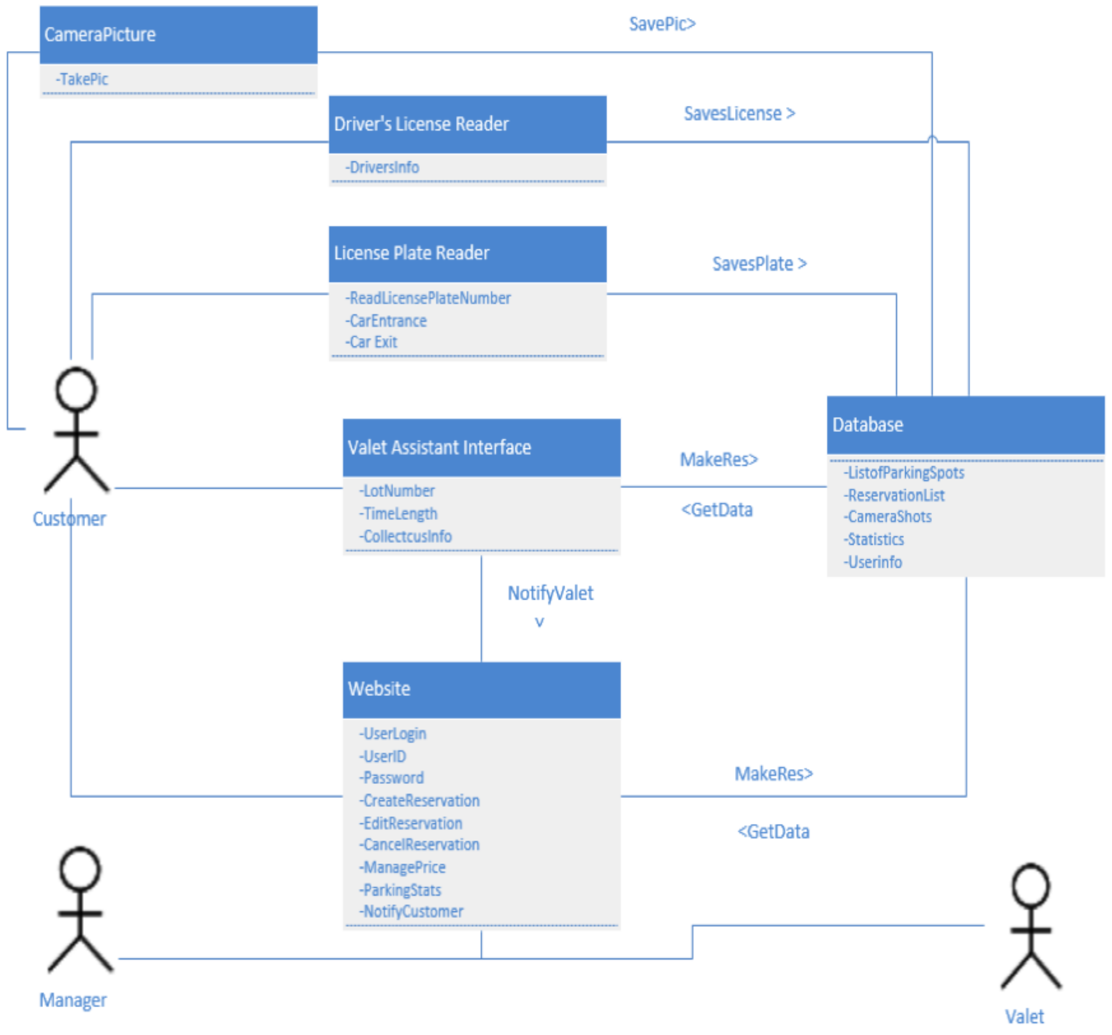
Environmental Factor	Description	Weight	Perceived Impact	Calculated Factor
E1	Beginner familiarity with the UML-base development	1.5	1	1.5 x 1 = 1.5
E2	Some familiarity with application problem	0.5	2	0.5 x 2 = 1
E3	Some knowledge of object-oriented approach	1	2	1 x 2 = 2
E4	Beginner lead analyst	0.5	1	0.5 x 1 = 0.5
E5	Motivated Team Membe	1	4	1 x 4 = 4

	rs			
E6	Stable Requirements Expected	2	4	2 x 4 = 8
E7	Part-Time staff will be involved	-1	5	-1 x 5 = -5
E8	Programming language with average to complex difficulty will be used.	-1	3	-1 x 3 = -3
	Environmental Factor Total:	9		

ECF = Constant-1 + Constant-2 x Environmental Factor Total = 1.4 + -0.03 x 9 = **<u>12.33</u>**
UCP = UUCW x TCF x ECF = 85 x .222 x 12.33 = **<u>230.571</u>**

Domain Analysis

Domain Model:



Concept Definitions (D-doing; K-knowing; N-neither):

Responsibility Description	Type	Concept Name
To check if the incoming customer has a reservation	K	License Plate Reader

To collect information incase the customer has no reservation	D	Valet Assistant Interface
To show the outside how many parking spaces are left	D	Outside parking sign
To obtain customer information and make customer online reservations	N	Website
*To manage the prices of fees related to the parking garage	D	Website
To manage employee information and statues	D	Website
To obtain parking lot number and key storage number	N	Valet Assistant Interface
To notify the manager that a customer has overstayed	D	Database
To notify the customers about their overstay	D	Website
To signal to the database that the car has left the parking garage	D	License Plate Reader
To analyze and observe parking garage statistics	K	Website
To park the car into the lot	D	Valet
To obtain the car out of the lot	D	Valet
To collect information from the drivers license	D	Drivers License Reader

Association Definitions:

Concept Pair	Association Description	Association Name
License Plate Reader ←→ Database	License Plate Reader sends license number of the car to the database.The database then	getlicenseplate

	stores it within it’s memory for future use.	
Valet Assistant ←→ Database	Valet Interface notifies the database that something either needs to be verified or edited. (e.g Whether or not a customer reserved a spot, or changing customer information. The database then retrieves the information and sends it back to the interface for use.)	displayinfo
*Camera ←→ Database	The camera takes pictures of the car and sends them to the Database. The database then stores the images within memory for future use.	Storepic
Website ←→ Database	The website accepts reservations from customers and sends the details to the database. The database stores the parking	Sendinfo, recieveinfo

	reservation and details within its memory for future use. The website may also pull information (such as customer or reservation information) from the database.	
Valet Assistant Interface ←→ Outside parking sign	The database will possess knowledge of the spots remaining within the facility and send that information to the outside parking sign. The parking sign will receive the information and display it outside.	Display_spots
*Manager ←→ Database	Manager requests data from the database via the website (e.g Employee Information, Trends). The database returns the requested data to the manager for managerial use.	Display_manager
Driver's License Reader←→ Database	Driver's License reader	Getinfo

	sends driver's license of customer to database. The database then stores it within its memory for future use.	
Valet Assistant Interface ←→ Website	Valet Assistant Interface notifies the website that a walk-in reservation has been made, or a customer with an existing reservation has arrived.	Update_info

Attribute Definitions:

Concept	Attributes	Attribute Description
Database	ListofParkingspots	Record of all the parking spots in the garage and who is parked where and for how long, accessible by the valets
	ReservationList	List of all the reservations made within the parking garage.
	Statistics	Contains parking garage information e.g monthly reports,employee information
	Userinfo	Contains customer information such as name,address,drivers license number,phone number

Valet Assistant Interface	LotNumber	Parking lot number where the customers vehicle has to be parked.
	TimeLength	Longest length of time a walk-in customer can stay at the present moment
	Collectcustinfo	Collects customer information when they have entered the garage
Website	UserLogin	Request for username and password
	Userid	Customers userid
	password	password used to login
	Createreservation	Creates reservation for the customer
	Editreservation	Change reservation time
	CancelReservation	Customer may cancel reservation
	*ManagePrice	Manager must be logged in and can edit prices for parking spaces
	*ParkingStats	Webpage to view parking garage statics such as reports , no. of spots occupied,no. of spots empty
	NotifyCustomer	Notify customers if they have

		overstayed and how time they have left
License Plate Reader	ReadLicensePlateNumber CarEntrance CarExit	Reads the customers license plate to verify if they have a reservation or if they are a walk-in Notifies the database that the car has entered Notifies the database that the car has exitted
Drivers License Reader	DriversInfo	Extracts information from the drivers license
Camera Picture	TakePic	Picture is taken of car to ensure safety of the car

Traceability Matrix:

Priority Weight	Use Case	Customer	Valet	Electronic Valet Assistance Interface	Cameras	License Plate Reader	License Reader	System	Website	Database
29	US-1								X	X
35	US-2								X	X
31	US-3	X		X		X	X			
10	US-4		X	X	X					
18	US-5							X		X
10	US-6		X	X						X
4	US-7								X	X
4	US-8								X	X

4	US-9								X	X

System Operation Contracts (Responsibility Description):

Operation	Register
Preconditions	● The customer does not have an account ● Username is between 5=<x<=20 characters ● Username has not been registered ● Password is x>8 ● Name,Address,Phone number,Drivers License Number will be inputted to complete registration
Postconditions	● Account is successfully created and stored into database

Operation	Online Reservation
Preconditions	● The user should be logged onto his account ● The required time for the customer should be available ● If the required time is not available then the customer will have the option of choosing another time. ● Payment must be completed
Postconditions	● The reservation will be completed and added to the database

Operation	Enter Garage(a) (Contracted Reservation/Confirmed Reservation)
Preconditions	● License plate reader has scanned the license plate ● Database has found a reservation ● Customer will have verified information
Postconditions	● Customer will be forwarded to the park area

Operation	Enter Garage(b) (Walk-Ins)
Preconditions	● License plate reader has scanned the license plate

	● Database has not found a reservation ● Customer has created a reservation on the spot ● Customer has inputted a valid phone number ● Customer has inserted drivers license into the drivers license reader to extract information
Postconditions	● Customer will be forwarded to the park area

Operation	Park
Preconditions	● Customer has successfully completed a reservation ● Customer has driven up the cameras ● Customer has left keys with valet ● Pictures are stored on the database ● Valet has confirmed security check
Postconditions	● Car has been parked in the garage

Operation	Overstay(a)(The parking lot is full)
Preconditions	● Customer has stayed beyond their reserved time ● System has updated database on overstay
Postconditions	● Customers car will be towed

Operation	Overstay(b)(The parking lot is not full)
Preconditions	● Customer has stayed beyond their reserved time ● System has updated database on overstay
Postconditions	● Customers reservation has been extended and the customer has been overcharged

Operation	Exit
Preconditions	● Customers has requested his car ● Customers has successfully paid ● License plate reader has scanned the car leaving

	● Database has been updated
Postconditions	● Customer exits with car

Operation	*Manage Prices
Preconditions	● Manager has logged on with his account ● Manager has selected the manage prices page on the website ● Manger has updated prices and confirmed the price change ● Database has been updated with a timestamp
Postconditions	● Prices for the parking spots have been changed

Operation	*Manage Employee Information
Preconditions	● Manger has logged onto his account ● Manager has selected the manage employee page on the website ● Manager has added or changed employee information ● Database has been updated
Postconditions	● Employee information has been changed

Operation	*Check Statistics
Preconditions	● Manager has logged onto his account ● Manager has selected check statistics page on the website
Postconditions	● Manager has viewed parking garage statistics

Simulation of Arrivals and Departures:

Having a single customer at a time to park in the garage would not exhibit interesting behaviors. On the other hand, it would be difficult to allow many users to simultaneously simulate the parking activity. We would need to develop a server that can handle many simultaneous interactions and recruit many people. We will simulate many artificial customers by using two Poisson processes. One process will simulate artificial customer arrivals: customers will arrive one at a time and their arrivals will be modeled as a Poisson process. The other process will simulate how artificial customers depart the garage, also one at a time.

For a Poisson process with average arrival rate λ, the probability of seeing n arrivals in the time interval Δt equals:

$$\Pr(n) = \frac{(-\lambda\Delta t)^{n} e^{-\lambda\Delta t}}{n!} ;\text{where } E\{n\} = \lambda \cdot \Delta t$$

Inter-arrival time t (time between successive arrivals) in a Poisson process follows exponential distribution with parameter λ:

$$\Pr(t) = \lambda e^{-\lambda t}, t \geq 0 \text{and} \Pr(t) = 0, t < 0; \text{where} E\{t\} = \frac{1}{\lambda}$$

To generate exponentially distributed random numbers, generate a uniformly distributed random number u on the unit interval [0, 1]. Then apply the following function to obtain an exponentially distributed random number rx:

$$rx(u) = \frac{-\ln(u)}{\lambda}$$

where ln() is the natural logarithm (using basis e). Let us assume that the unit interval is one hour, so the parameter specifies the average number of arrivals per hour.

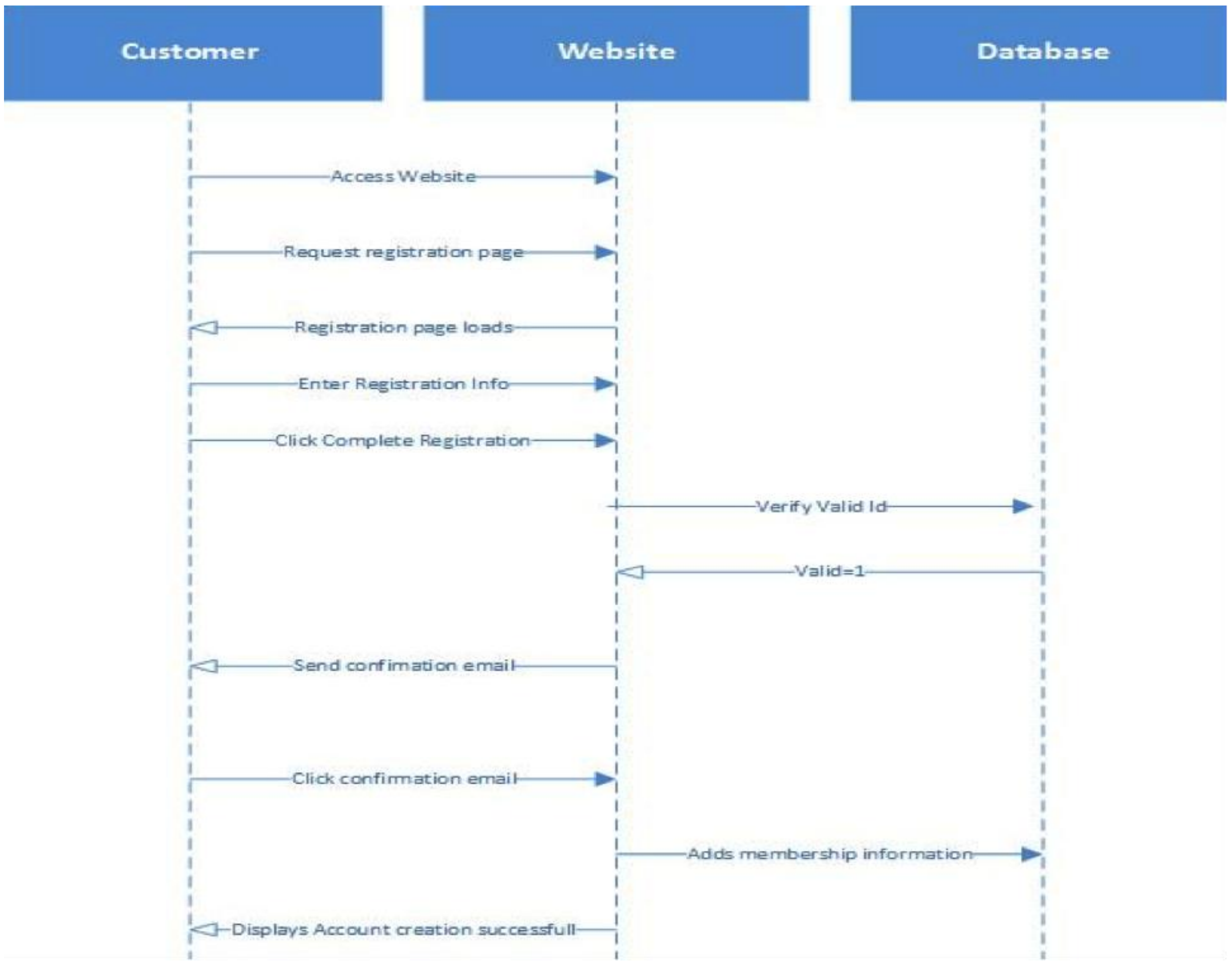
This module runs two threads in infinite loops as follows. The first thread simulates arrivals:

1 Query the database if there are currently any “Available” spots. If yes, select one randomly and change its state to “Occupied.” If there are no available parking spaces, record this attempt as an “Overbooked” event in the statistics table, maintained by Module-5 in Figure 4.
2 Generate an exponentially-distributed random number rx using equation (4). Convert the number to the time scale, e.g., if rx = 0.3, then t(rx) = 0.3 x 60 minutes = 18 minutes. This number represents the time of the next arrival.
3 Suspend thread to sleep for t(rx) time. When the thread wakes up, go to Step 1.

A similar thread runs the departures process. The departures thread selects a random occupant/customer from the database for departure. We must be careful to allow dislodging of only artificially generated customers. A more realistic simulation would also simulate reservations and another Poisson process.

Interaction Diagrams

Online Registration:



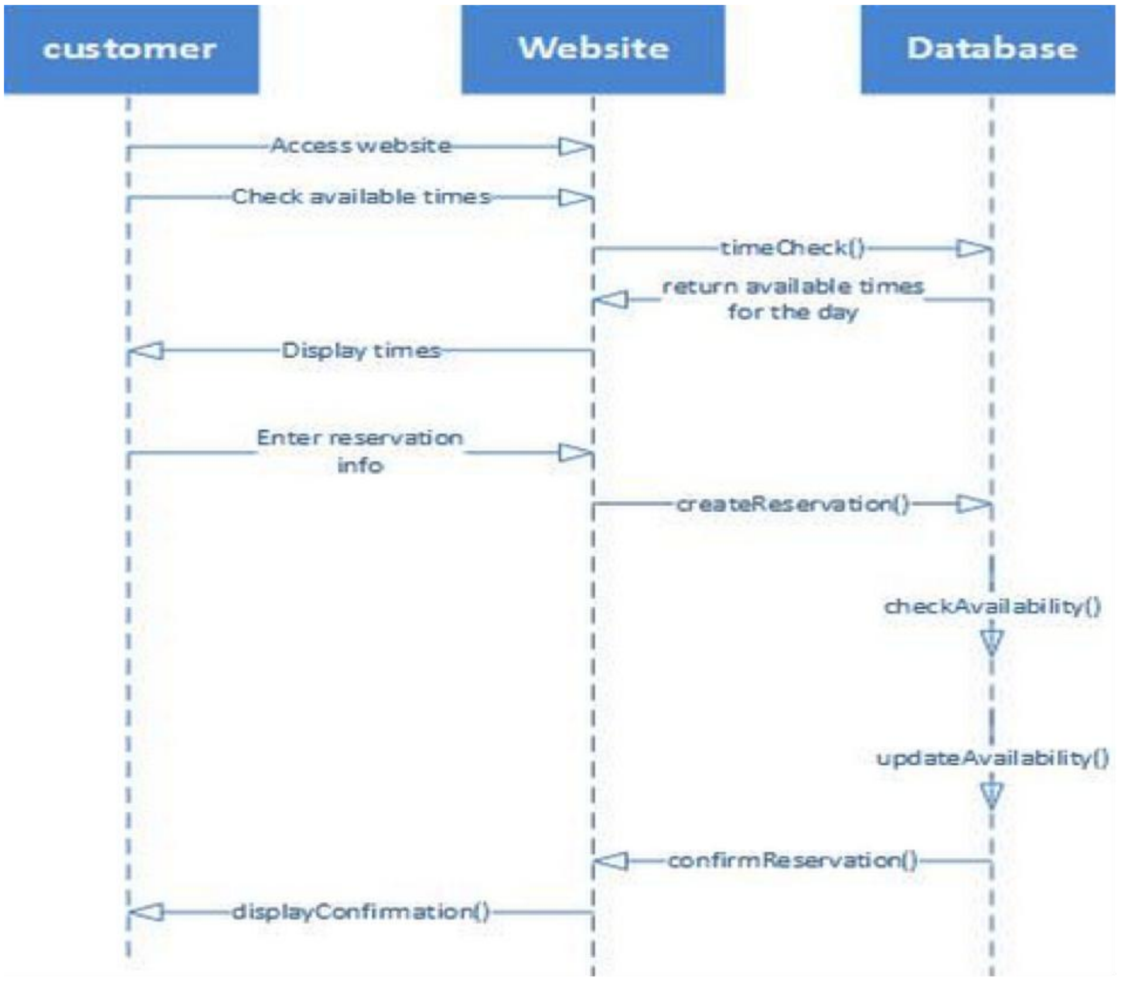
Customer ⇔Website

- C → W: The customer access the website to enter the registration information and then interacts with the website one more time by entering in confirmation email.
- W → C: The website sends the confirmation email to the customer when all registration information is entered to verify the validity of the person and once verified, displays a “Creation Successful” to customer.

Website ⇔Database

- W → D: The website will verify the validity of the entered user identification by checking it with database records and once confirmed, adds the new account to the database.
- D → W: Database sends the result of the identification check back to the website.

Online Reservation:



Customer ⇔Website

- C → W: The customer uses the website to check available times and input reservation information.
- W → C: The website displays the available times and confirms the reservation to the customer

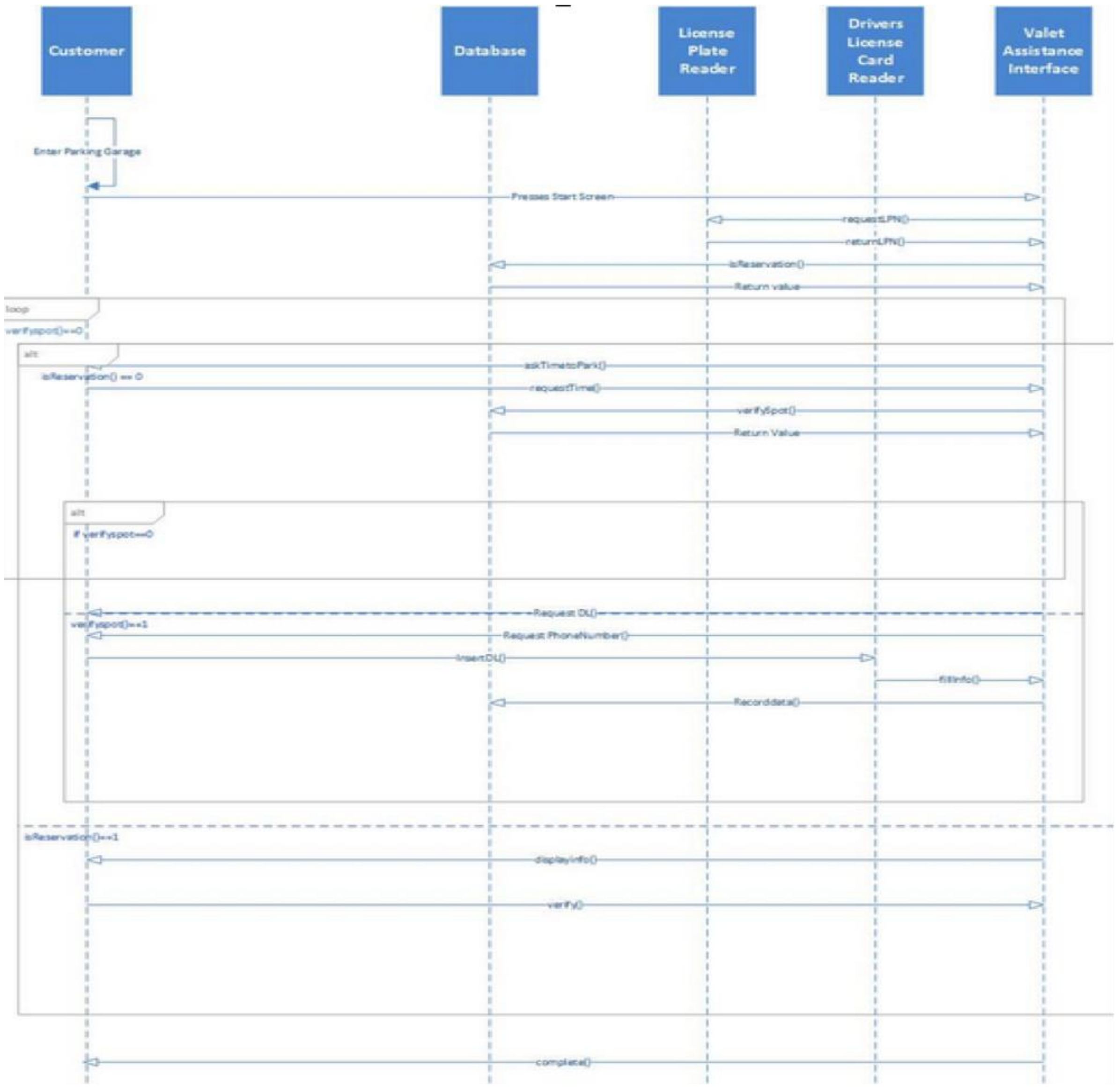
Website ⇔Database

- W → D: The website requests from the database available parking times and stores newly created reservations.
- D → W: The database returns available times upon request from the website and confirms the reservation after it's been verified.

Database ⇔Database

- D → D: The database upon being creating a new reservation, checks itself to see whether or not the spot is available and update the status of the parking spot.

Entering the Garage:



Customer ⇔Valet Assistant Interface

- C → VAI: The customer starts the registration verification process by pressing start and when they verify the information displayed by the interface when found..
- VAI → C: The valet assistance interface interacts with the customer in requesting the estimated time to park and phone number when they don't have a reservation and displaying the customers reservation information after they successfully make one. The interface will also complete the process once the customer verifies the information.

License Plate Reader ⇔Valet Assistant Interface

- VAI → LPR: The valet assistance interface requests the license plate number from the reader when the car is at the checkpoint.
- LPR → VAI: The LPR sends the requested license plate number to the valet assistance interface.

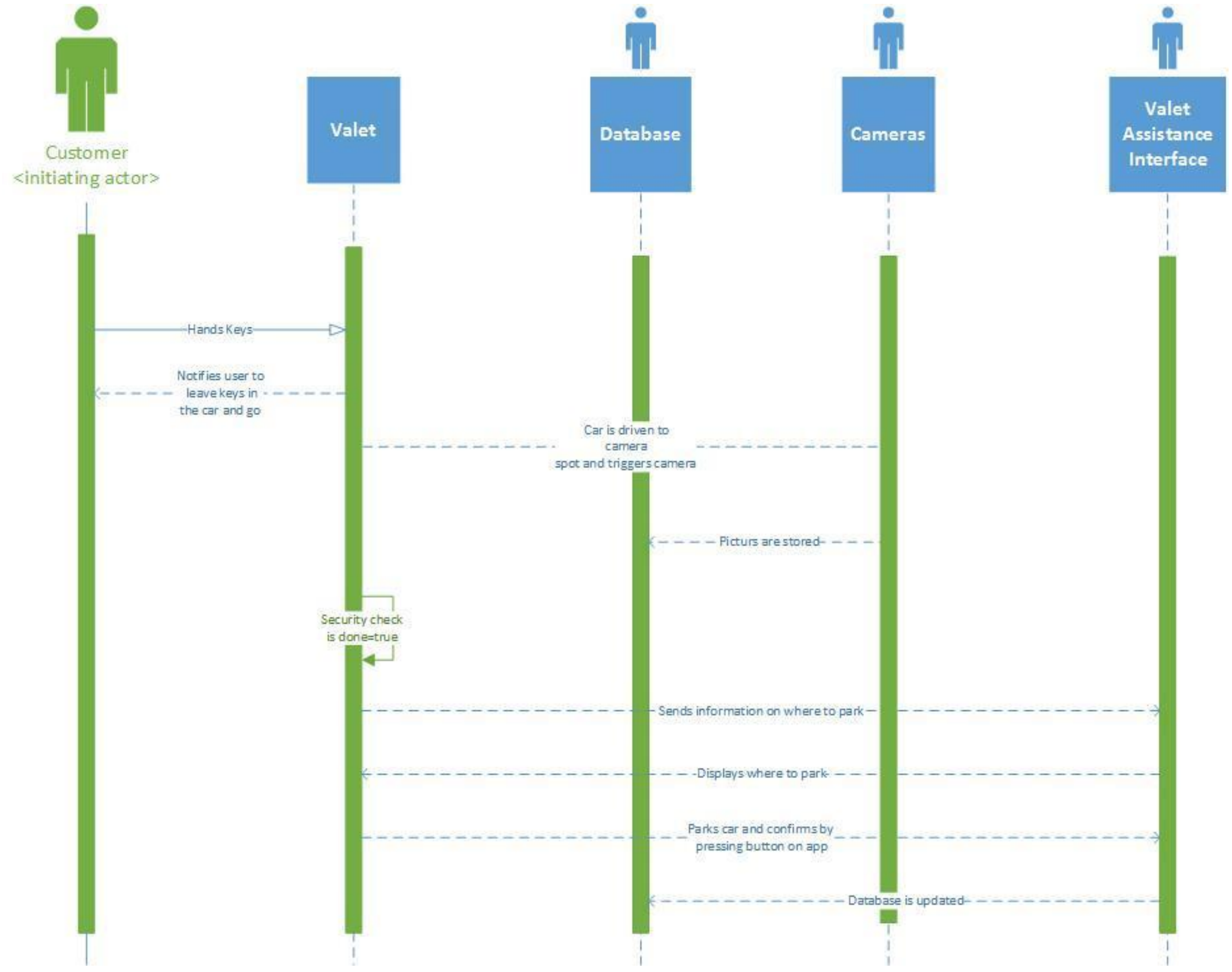
Database ⇔Valet Assistant Interface

- D → VAI: When requested, the database returns to the VAI whether or not the customer has a reservation as well as whether or not the spot.
- VAI → D: The Valet Assistant Interface requests from the database, whether or not a customer has a reservation, a spot is available, and stores new updated information to the database on new reservations.

Drivers License Card Reader ⇔Valet Assistant Interface

- DLCR → VAI: The Drivers License Card Reader takes the scanned information and fills in the customer information.

Security Check/ Park:


Customer
<initiating actor>
Valet
Database
Cameras
Valet Assistance Interface
Hands Keys
Notifies user to leave keys in the car and go
Car is driven to camera spot and triggers camera
Picturs are stored
Security check is done=true
Sends information on where to park
Displays where to park
Parks car and confirms by pressing button on app
Database is updated

Customer ⇔Valet

- C → V: The customer hands his keys to the valet.
- V → C: The valet obtains the keys and provides the customer his ticket. Also parks the car as soon as the VAI has provided the lot number.

Valet ⇔Valet Assistant Interface

- V → VAI: The Valet makes of that the security photos have been taken.

- VAI → V: The Valet Assistance Interface provides the valet with a parking lot number after the security check has been done.

Database ⇔Valet Assistant Interface

- D → VAI: The Database provides the Valet Assistant with the proper lot to park the vehicle.
- VAI → D: Valet Assistant Interface requests for a final parking lot number.

Database ⇔Database

- D → D: The database updates its information after completing requests.

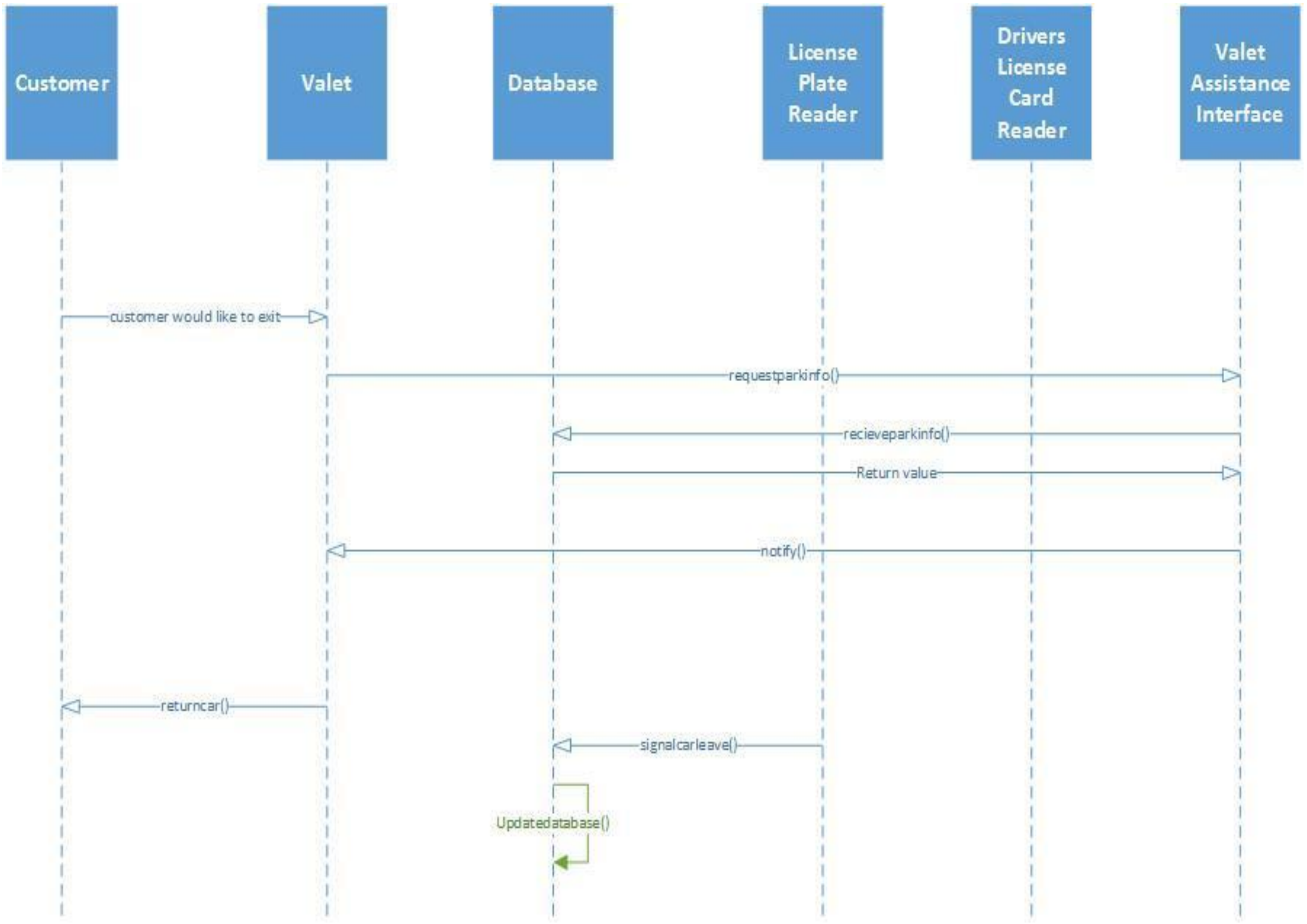
Camera ⇔Valet

- V → CAM: The valet presses a button which triggers the cameras to take the secruity photos

Camera ⇔Database

- CAM → V: The camera photos are sent to the database to be stored for security purposes.

Exiting the Garage:



Customer ⇔Valet

- C → V: The customer interacts with the valet by asking for their car back.
- V → C: The valet interacts with the customer by returning their car.

Valet ⇔Valet Assistant Interface

- V → VAI: The Valet requests parking information from the Valet Assistance Interface.
- VAI → V: The Valet Assistance Interface returns the information on the requested parking space.

Database ⇔Valet Assistant Interface

- D → VAI: The Database sends to the Valet Assistance Interface the customer's parking information

Valet ⇔Camera

- V → C: The valet uses the cameras to take the pictures of the car.

Database ⇔Camera

- C → D: The Camera sends the pictures taken to be stored in the Database.

License Plate Reader ⇔Valet Assistance Interface

- LPR → VAI: The License Plate Reader sends the license plate number of the car to the Valet Assistance Interface so the valet can verify that it is the correct car.

Class Diagrams and Interface Specifications

Design Patterns:

Certain design patterns that we learned in class have been implemented in our project. Using these patterns, we simplified our design and made it more flexible so that our design can be implemented in various situations.

A pattern we implemented was proxies. Due to the lack of funding, we were not able to obtain certain hardware components such as a driver's license reader and a license plate reader. We implemented a proxy design pattern to simulate these hardware components. The added benefit was that this design pattern allowed simulating the device as if we had the actual component and allows us to demonstrate system functionality.

We want to add a feature to the valet assistant that will allow the valet to know the states of cars as they come in. This will aid the valet in knowing when to obtain a car and will also help the system know when a car has left. This state design is intended to help the system work more closely together so that the process of parking and leaving is more efficient. There will be five states: Entering the garage, security check, parked, obtain, and completed. The entering state will assist the system with tracking how many cars are entering the parking structure. The security check state will notify the valet which car he needs to assist. The parked state will be the state when the valet has finished parking the car. When the customer wants his car back, the customer will input his ticket number and then this will notify the valet that a car needs to be obtained. The completed state will notify when the customer has paid and left the garage. We have also used the state design to show whether or not the parking spots are reserved and being used. A state design pattern seems to be the most effective in aiding the system and the employees with knowing what to do in different states.

Another pattern that we implemented is a decorator design pattern. This allows us to add more features to our automated parking garage system and website. For our website, this allows for different pages and functions to be added. A feature that we will implement in the future is an Employment page. This page will allow employees to manage their salaries to promote a paperless system. Another feature that we wanted to add is a management page that will allow the owner to change prices and analyze information collected. The decorator pattern also would allow us to implement our texting notifications. This texting feature would be convenient for customer because they would be able to get a notification of their parking status. Currently, the decorator pattern is implemented by our addition of a line graph to our web page that helps the manager track his sales each month.

Object Constraint Languages:

1)manager	**2)Employee**	**3)Price**

Invariant: Context manager inv : self.salaray Precondition: Context manager :: (s:integer) Pre:self.salary=0 Postcondition: Context manager::(S:integer) Post:self.salaray= h*w	Invariant: Context employee inv : self.salaraye Precondition: Context employee :: (e:integer) Pre:self.salarye=0 Postcondition: Context employee::(e:integer) Post:self.salaraye= h*w	Invariant: Context price inv : self.hour Context price inv : self.month Precondition: Context price :: (h:integer) Pre.selfhour=0 Context price:: (m:integer) Pre.selfmonth=0 Postcondition: Post:total = self.hour + self.month
4) Account	**5)Reservation**	**6)Garage**
Invariant: Context account inv: self.useraccess Precondition: pre:self.useraccess=true Postcondition: Post:self.useraccess=self.webaccess	Invariant: Context reservation inv:self.reserve Context reservation inv:self.spotnum Precondition: Context reservation::(r:intger) Pre:self.spotnum=true(spot is available) Postcondition: If self.spotnum=true then self.spotnum=self.spotnum-1(spot is reserved and there is one less spot) else return false	Invariant: Context garage inv:spots Precondition: Context garage::(g:integer) Pre: self.spots=vacantspots Postcondtion Postcondition: self.spots =self.spots-self.spotnum

System Architecture and System Design

Architectural Styles:

The overall style of our system follows a component-based methodology. Our system will be comprised of multiple hardware and software system that will need to be able to function on their own and relay their information to the appropriate component. The aspects of component -based styles are that they are: reusable, replaceable, no context specific, extensible, encapsulated, and independent. The benefits of of component-based styles is that they have ease of deployment, ease of development, reduced cost, and are reusable.

Each of our components such as the license plate reader, license card reader, Valet Assistant Interface, and cameras are all components that are taken advantage of and will need to communicate with one another to create a working system. A lot of these components already exist in the real world which makes this system take advantage of component-based methodologies.

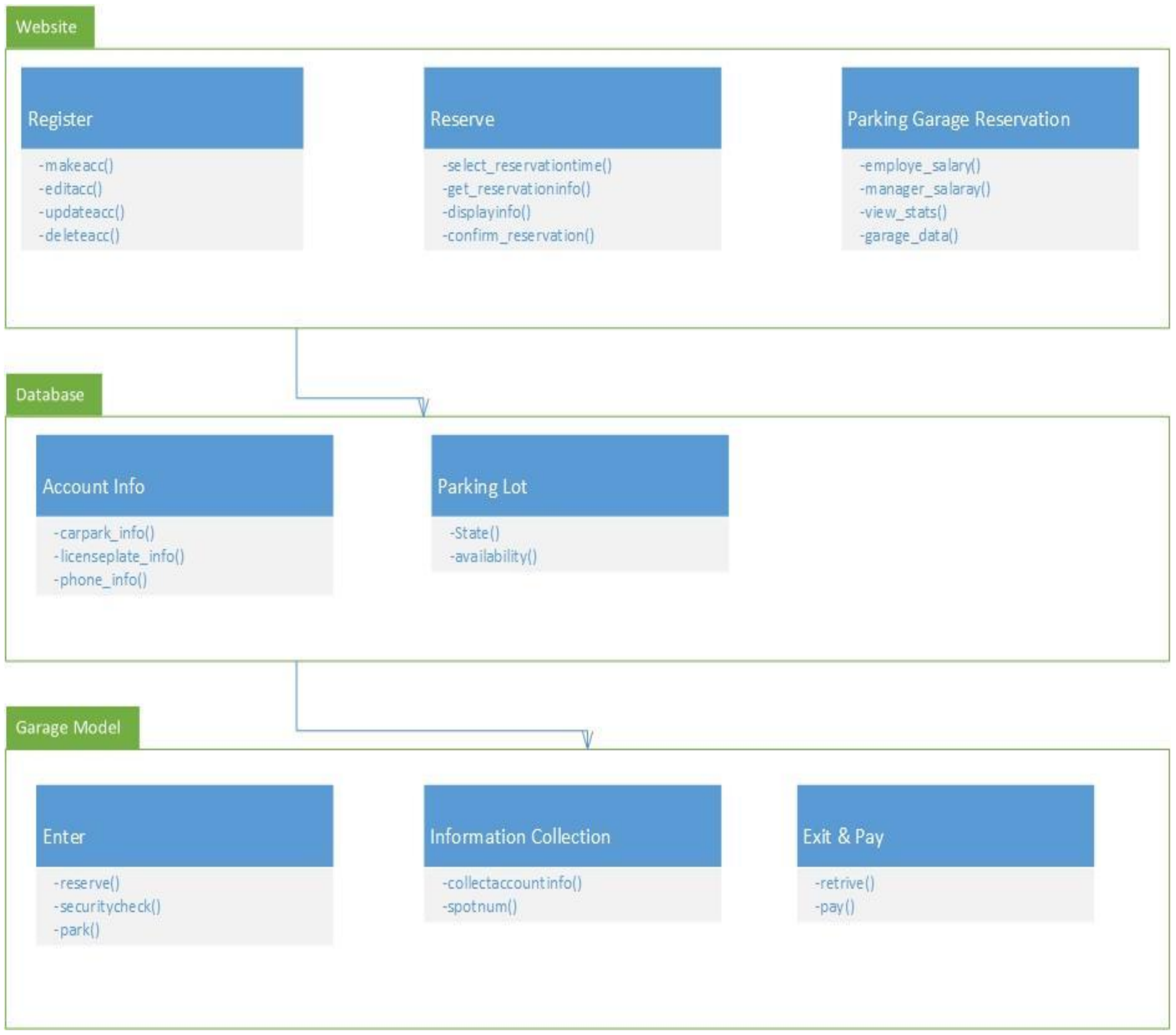
The database and website will follow a client/server architectural style. The reason for the client/server architecture is because we need something to maintain the data and retrieve information. The website will need to access the server which will then access information that needs to be displayed on the the user's web browser. This server will also be needed when the Valet Assistant Interface communicates with the database.

The database and website together will be of an event-driven architecture. The database and website are both highly dependent on either user input or the current status of the parking spots within the garage structure. It therefore requires that these two systems be event-driven. These two aspects will not be the only thing that is event driven. Because our garage components are all event driven we decided to use a pipe and filter model.

We are using a pipe and filter model to maintain this parking garage. The speed of our information collection will "pump" the car through the garage and "sink" the information into our database. The faster the processes the faster the user can park. We are trying to optimize the time the user waits by using the License Plate Reader to query the system if the driver has a reservation. If the user does have a reservation, the driver would be able to continue to the valet where the car is now in the companies's hands.

This pipe and filter model also describes how our event-driven model works. The car entering the garage and triggering the Assistant Interface, allows for the system to start collecting information to send the driver to the security check. The completion of the information collection, allows the flow to the speedy security check which is handled by the cameras and the valet's use of the interface. Then, this completion of the security check leads to the Parking of the Valet and Ticket for Driver to retrieve his car later in the day. The customer's return triggers the retrieval of the car which sends a Valet to obtain the car and trigger the Exit subsystem. This Exit subsystem causes the License Plate Reader to send the License Plate of the car leaving to the database. The database then updates the status of the lot for future use. The car is returned to the customer as he pays the cashier. The filtering deals with the exceptions such as a car not wanting to park and signaling the car to exit or managing invalid inputs for information collection.

Identifying Subsystems:



The subsystems of our System Architectures comprises of the:

- For the Garage Model
  - Enter
  - Information Collection
  - Exit and Pay
- For the server
  - Database

- For the Website
    - Register
    - Reserve
    - Parking Garage Management

Our information collection subsystem is comprised of our License Plate Reader, Driver's License Reader, and the Cameras, database. This subsystem is responsible for the collection of important information needed to run securely run the parking garage.

The Enter subsystem is comprised of the Assistant Interface, Driver's License Reader, and the License Plate Reader, database. This subsystem will maintain that the entrance stage is complete to park the car, else leave the garage.

The Exit and Pay subsystem is comprised of updating the parking lot through the use of the License Plate Reader and the database and money management to validate the return of the car to the Driver.

Mapping Subsystems to Hardware:

The system will have to be able to run on multiple devices. In particular, the system will have to be able to communicate with the Valet Assistant Interface. The Information Collection subsystem should be able to maintain the information that is provided by the external devices such as the License Plate Reader, Cameras, and the Drivers License Reader. It is responsible for making sure the information collected is valid and is properly stored into the database.

Server Model:

Maintaining Data

- Database Server
- Web Server

Website Model:

Register:

- Database Server
- Web Server

Garage Model:

Enter Subsystem:

- Driver's License Reader
- License Plate Reader
- Tablet (running on Android for project implementation - VAI App)
- Database Server
- Web Server

Security Check Subsystem:

- Cameras
- Tablet (running on Android for project implementation - VAI App)
- Database Server

- Web Server

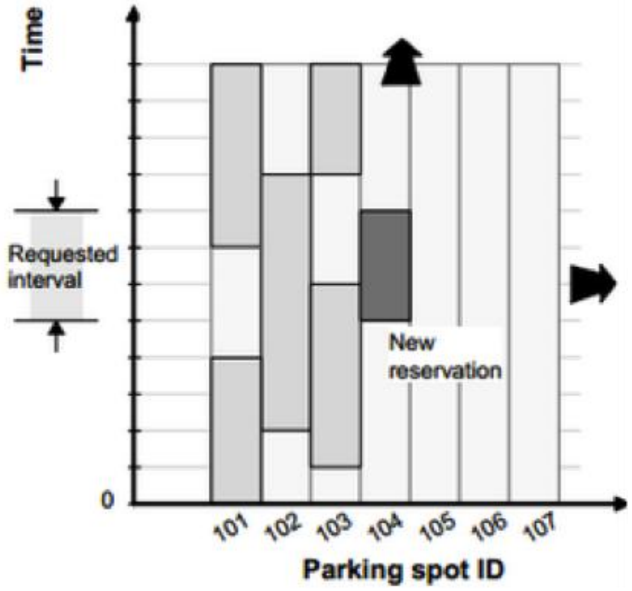
Exit and Pay Subsystem:

- Driver's License Reader
- License Plate Reader
- Tablet (running on Android for project implementation - VAI App)
- Database Server
- Web Server

Persistent Data Storage:

The system will need to be able to preserve the states of the parkings spots within the parking structures for an indefinite amount of time. The preservation of these states will be stored in a flat file in a form that is a variation of a bitmap.

The file format will be of the following:



The file format will consist of a 2-dimensional matrix (contained in an array) that contains the parking spot ID and the time interval that are reserved. The matrix will consist of integer digits from 0-3 for the four possible states of reservation the parking spots can be in: unoccupied, occupied, reserved unoccupied and reserved occupied.

Data storage will differ for other data that will be preserved. Employee information that is stored on the database will follow the Entry-attribute-value model, which can also be stored as a flat file. Information that could stored are as follows: Name, salary, job title, and shifts.

Global Control Flow:

Our system is event-driven and we will be able to have multiple requests happening in short periods of time.

Time dependency:

There is some time dependency after the cameras are taken. The camera will need a few seconds to automatically adjust to the lighting. This will be solved by the valet pressing the capture picture button as soon as he sees that the light is green signaling the camera is ready for a picture.

Also when integrating the system design for multiple valets, concurrency will be an issue. Many of the concurrent aspects will be depended on time and need to be managed properly. This work will be worked on in the future. This time dependency also adds on to memory coherence, which means that we need to make sure that the database is updated when it needs to be to insure proper functionality of our system.

Hardware Requirements:

- Tablet 8GB(min): as the Valet Assistant Interface
- LED Display: for outside to show vacancy of the garage
- Cameras: 10MP for security check
- Driver's License Reader: to quickly obtain information with one swipe
- License Plate Reader: 10MB min to read the plate
- Database: 100GB at least to store all the information.
- Hard-drive: At least 50 GB of space. This disk will store only cached information.
- Server: to process the information valet interface and the license plate reader
- Internet connection: to connect the garage system with the garage. (At least 2Mbps)

Algorithms and Data Structures

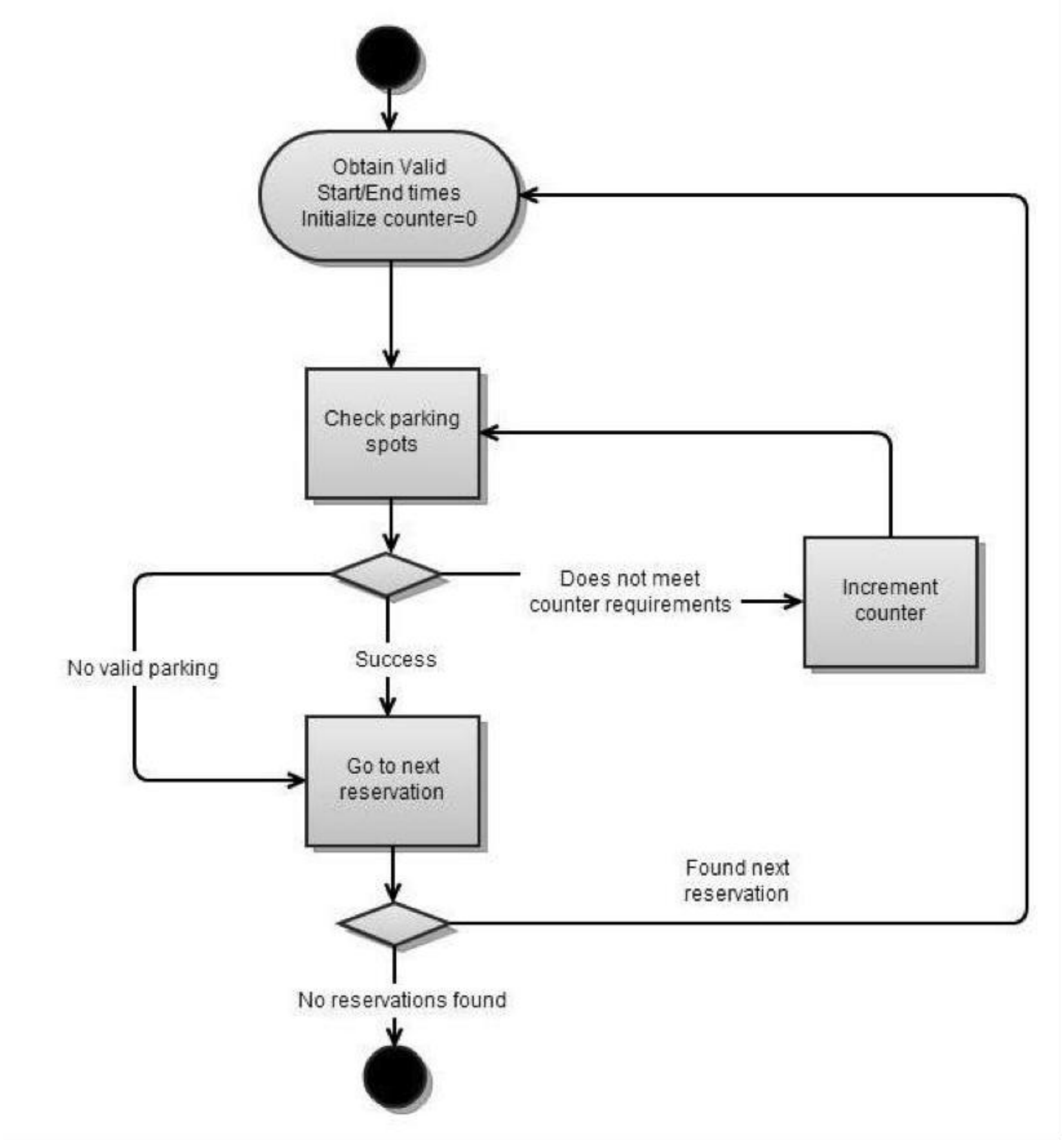
Algorithms:

The mathematical model for the simulation of arrivals and departures will be fairly simple to implement. The algorithm to describe the arrivals/departures of vehicles will closely follow the original equations defined in Report 1. Within a conditional while loop, this algorithm will continue to execute with a random number generator until the garage is full.

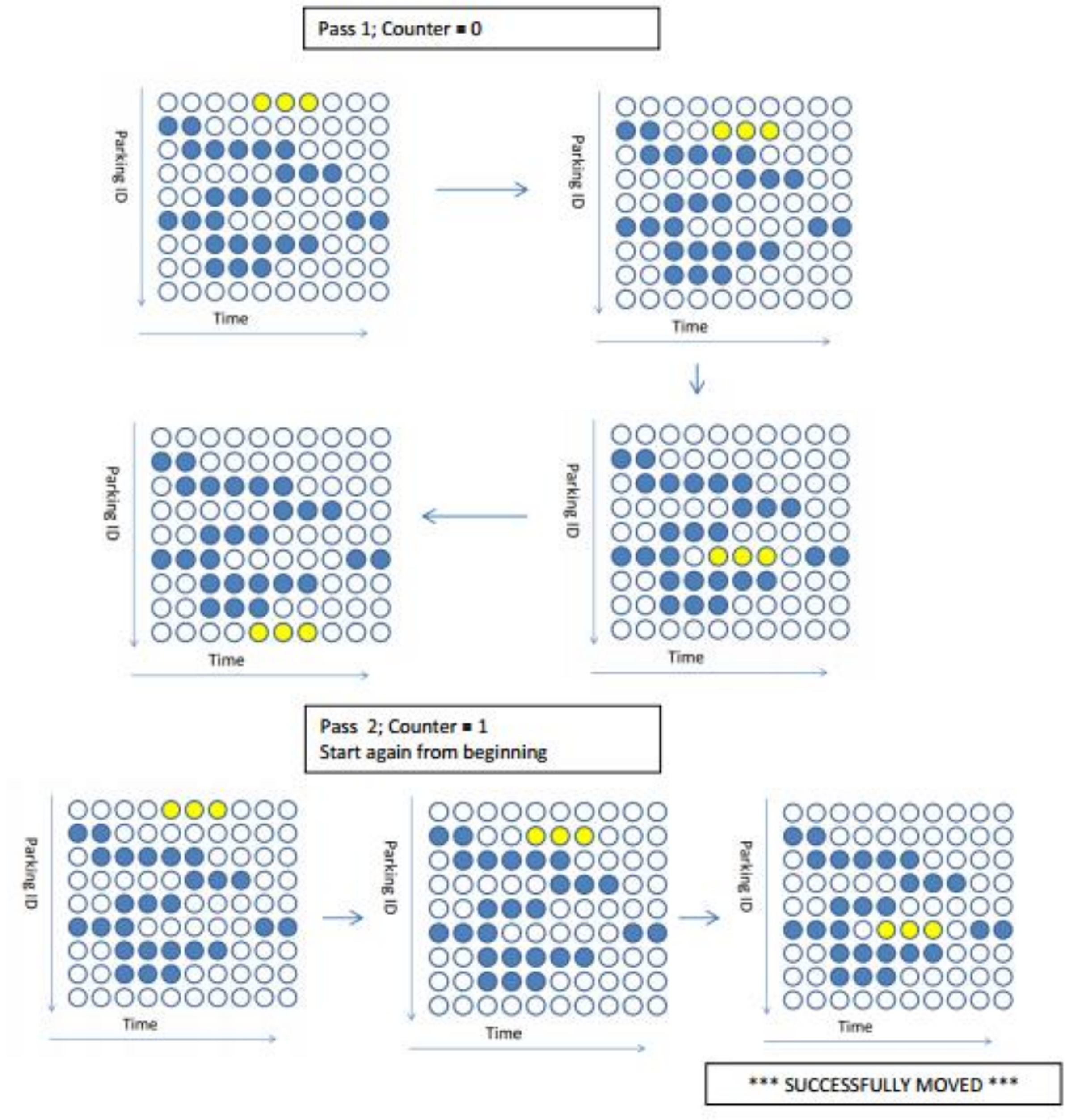
There will also be an algorithm that will sort the reservations within the garage. The data will be stored as integers within an array. The algorithm will note in the array matrix where the parking reservation begins and ends. These two integers are critical for determining where the spot can be shifted to. The algorithm will need to check each parking spot and check if there is no conflict between the beginning and end times with any previously placed reservations. Since the consolidation of these parking reservations is the number one goal, it is important to minimize the unused times between parking reservations. To take this into account, a counter which is initialized to zero will be used.

The counter is used to describe the maximum time units (here our time unit is described as 15 minute intervals) there can be between the end of the swapped reservation and a reservation in place or the beginning of the swapped reservation and end of a reservation in place. If the algorithm cannot find a spot to place the reservation, then the algorithm goes onto the next reservation to begin swapping. However, if there are available spots but does not meet the maximum time unit requirement, the counter will be incremented by one and the process repeats until the conditions are satisfied. It it is important to note that the counter cannot increment more than the number of time units there were in the original reservation spot.

The flow chart below describes the steps that occur during this process.



The following diagrams are a visual representation of what would occur every step.



Data Structures:

Our system does not use "complicated" structures such as trees, hash tables, or linked lists, but we do utilize arrays, MySQL tables, and bitmaps. The logic in this decision is that since our system has such a large portion in a database on a web server, we should avoid implementing things that are not inherently or implicitly supported, such as the aforementioned trees and linked list.

Since the web language is HTML and CSS, PHP, and MySQL, we decided to optimize the built in features with the MySQL table system, which is based in Structured Query Language (SQL), where queueing something is as simple as a stating something like:

```
$> SELECT * FROM my_table WHERE field1 = 'attr1' AND field2 > attr2
```

This would quickly search, within table <my_table>, for entries whose <field1> stores the string <attr1> and in which <field2> is greater than <attr2>. This is powerful for getting specific users within a certain time frame, or logging in a user that has matched their username and password.

The other prominent structure is a bitmap used to help keep track of reservation times within the garage. The bitmap is essence a two-dimensional array that can store information in its cells. It may not be the most efficient data structure in computer science, but it is one of the bests that is able to easily move from web applications to mobile applications and can operate in many languages.

User Interfaces Design and Implementation

Valet Assistance Interface:

(Screen 1) In report 1, the customer would be initially brought to the start screen that would display whether or not the lot/garage has vacancies with the longest interval at which they can park. To cut back on excess information, this screen will just ask whether or not the customer has a reservation with the option buttons of "Yes" and "No".

(Screen 2) "Yes" If the customer selected yes, it will display the same information as stated in Report 1.

"No" If the customer selected "No", it will display at the top of the screen whether or not the lot/garage has vacancies and also display the longest interval the customer can park. The rest of the screen will contain the same information as Report 1 in which the customer will enter the amount of hours and their phone number, but instead of them having the "Confirm" option, it will be "Next".

(Subscreen 2) This screen will be added, increasing the user effort by 1 click, or more if they decide to press the "Back" button. This screen will display the information that the customer had just entered and will ask if it is correct. If it is correct, then the customer will hit the "Confirm" option. If it is incorrect, the customer will hit the "Back" option and correct any information that was inputted incorrectly in the previous window. In the end, this will save both sides the time it takes to correct any mishaps that may occur in the future due to incorrectly inputted information.

(Screen 3) This screen will be the same as in Report 1. It was ask the customer to scan their driver's license for confirmation.

As in Report 1, there will be another assistant upon exiting and the customer will be asked to touch the screen to start.

(Screen 1) This screen will be the same as in Report 1 and have a "Start" option.

(Screen 2) This screen will ask the customer to scan their driver's license as in Report 1.

(Screen 3) This screen will be added and will display what car the system has for that customer and ask for confirmation with the "Yes" button or "No" button.

(Subscreen 3) "Yes" The screen will display "Thank You for Parking" as well as the cost of parking.
"No" The screen will display "Please contact an employee, sorry for the inconvenience"

For the Valet Assistance Interface (Entering), the initial screen is easier to follow now that it will just display "Do you have a reservation?" with the options Yes or No. A confirmation screen was added that will increase user effort by a minimum of 1 click, but in the long run it will save effort on both sides.

For the Valet Assistance Interface (Exiting), the program was elaborated on more. Two extra screens were added, increasing user effort by a minimum of two clicks. One is a confirmation page that will display whether or not the correct car is linked to their driver's license and is ready for retrieval. Then based the response, it will take them to the final screen, displaying the cost of parking, or whether or not a problem needs to be resolved.

Valet Application/Website Interface (For Employees):

1. The first page/screen will remain the same as explained in Report 1. It will ask the valet to input their ID and if valid, it will bring them to the next page. If it is incorrect, they will be prompted again until a correct ID is inputted.
2. The second page/screen will generally remain the same, displaying the reservation list and a "Customer Information" Link. There will now also be options "Back to Main Page" and "Log Out"
3. The third page/screen will be changed to make it easier for the valet. On this page, they will be given a text box in which they can enter a customer's name. If the customer exists within the database, the valet will be brought to the next page. If not, they will be prompted to enter the name again.
4. The fourth page/screen will generally remain the same. It will display the customer's personal information (ex. Birthday, Phone #, Driver's License Plate, etc.). This page/screen will now also include "Back", "Back to Main Page", and "Log Out" function.
5. The final page/screen will display "You have successfully logged out".

For this application, an extra window was added for successful log out at the end, not really increasing user effort as it will just display it. User effort was decreased in page 3. In report 1, The customer information page required entering personal information, when now it will just ask for the name before displaying personal information.

Customer Website Interface:

To create an account the user will have to take the following steps.

1. **Navigation:**
   a. Open homepage
   b. Click on the 'register' button located in circle
   c. Complete data entry(As shown below)
   d. Press enter
2. **Data entry**
   a. Click on the First Name data field
   b. Fill in customers first name
   c. Press the tab key to move to the next field 'Last Name'
   d. Fill in customers 'last name'
   e. Press the tab key to move to the next field 'Email'
   f. Fill in customers 'Email'
   g. Press the tab key to move to the next field 'Email again'
   h. Fill in customers 'Email again'
   i. Press the tab key to move to the next field 'Password'
   j. Fill in customers 'Password'

- k Press the tab key to move to the next field ‘Password again’
- l Fill in customers ‘Password again’
- m Press the tab key to fill in the security question
- n Fill in customers ‘security question'
- o Press the tab key to move to the next field ‘Question answer’
- p Fill in customers ‘question answer’

To create a reservation the user will have to take the following steps:

3 **Navigation:**
- a Open homepage
- b Click on the reserve option
- c Complete data entry(As shown below)
- d Press enter

4 **Data Entry**
- a Enter the customers users name
- b Press tab and move to the next field ‘password’
- c Enter the correct password
- d Press enter to go the reservation page.
- e Enter Vehicle License Number
- f Press tab to go to the next field (drop-down menu) ‘Start Reservation Time’
- g Select the time customer will be there
- h Press tab to go to the next field (drop-down menu) ‘End Reservation Time’
- i Select the time customer will leave
- j Confirm required time

To edit account information the user will have to take the following steps:

5 **Navigation:**
- a Open homepage
- b Point mouse to ‘account’ option
- c Select Account Info and choose edit Account
- d Make Changes as needed.
- e Click on save changes

To edit a reservation the user will have to take the following steps.

6 **Navigation:**
- a Open homepage
- b Point mouse to ‘reserve’ option
- c Select Edit Reservation
- d Enter Reservation number
- e Press enter
- f Make Changes as needed.
- g Click on save changes

To log out of the respective account:

7 **Navigation:**
- a Point mouse to ‘account’ option
- b Click on the log-out option
- c Click yes to confirm account log-out

Design Tests

Each unit will be tested before integration with one another. The units that need to be tested are (These also cover the requirements mentioned in Report #1):

- Valet Assistant Interface
- Website
- Database
- Classes and Methods

Testing the Valet Assistant Interface:

Goal: The Valet Assistant Interface is responsible for interacting with the driver upon entrance of the garage and interacting with the Valet as he/she works, parking and retrieving cars.

Test Cases:

- Check if driver has reservation
  - This makes sure that the interface can communicate with the license plate reader and then checks the database
  - To check this we need to see if we are able to search for a reservation given a license plate number provided by the reader
- Check for valid input
  - Filters input to keep the system safe
  - To check this we will try to put invalid input such as letters for phone number and numbers for first and last name.
- Minimal strokes for driver (ease of use)
  - Use the valet assistant interface to see if modifications to the layout is needed.
  - To check this we will play with the interface and get input to see how to change the layout to make it more user friendly
- Valet Login
  - Allow the valet to use the Valet side of the system to do his/her job
  - To check this we will see if the valet can log in using a valid username and password
- Security Check
  - Obtain pictures from camera and send to the database
  - To check this we need to see if the pictures are stored into the database and able to obtain the pictures from the database to show the driver incase he wants to see them

Testing the Website:

Goal: The website is responsible for allowing the drivers to make online reservations and create an account to manage their reservations.

Test Cases:

- Create account
    - To check this we will see if we can create an account and verify it with an email address
- Create reservation
    - To check this we will see if a reservation is stored in the database and make sure that a driver cannot make a reservation in a time slot that is taken already
- Manager login
    - To check this we will see if the manager can log in using a unique username and password
- Manage employee information
    - To check this we will see if the website displays proper employee information and make sure that changes can be made. We will also see if we can delete and add more employees
- Manage garage fees and prices
    - To check this we will see if a change in the in managers log in will update the fees

Testing the Database:

Goal: The database is responsible for recording driver's information, vacancy of the parking garage, employee information, garage fees, security check pictures and reservations.

Test Cases:

- Record data
    - To check this we must check the database and see if information is being stored properly and if we are able to obtain the correct information

Testing Classes and Methods:

Goal: The classes and methods in our system are responsible for making the units work, these are the tools that the units use to accomplish their task.

Test Cases:

- Refer to: Data Types and Operational Signatures
    - These are the classes and methods we will be testing
    - To test these we will see this we will use a combination of print and if statements to see if the output is valid

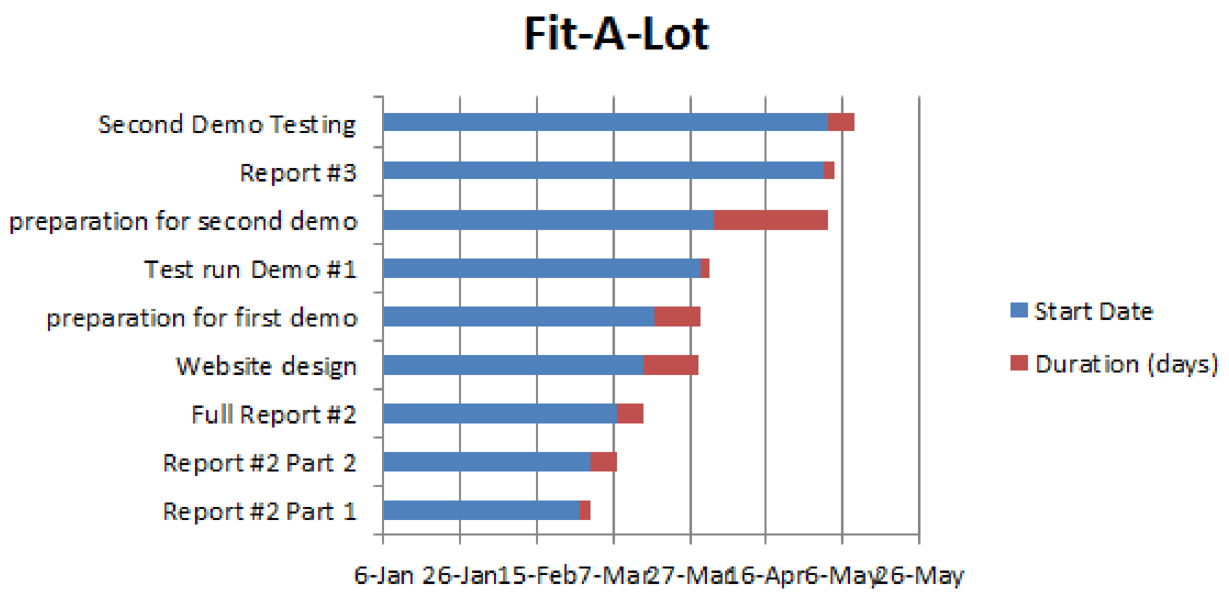
Integration Testing Strategy:

When the units are complete we will slowly integrate a few units together at a time to ensure proper integration of the whole system. We plan to complete the database first and test if we can properly store information into it. While this is happening we will also make sure that our website it working properly such that we can create accounts and reserve and the proper information is being stored into the database. The valet assistant interface will be tricky but once the website and the database seem like they are working properly we will try to tackle the challenge of integrating the valet assistant interface.

History of Work, Current Status, and Future Work

During our time on this project we set various goals for ourselves in Report #1 and #2, this included having a fully functional website that could optimize the resources for a customer or manager, having an app that could be used on tablet located at the garage, and having an algorithms to optimize the parking of the garage. As time progressed, these goals were analyzed and updated for time constraints and prioritizing features.

Previous goals:



Now that we reached the end, we were able to complete:

- Mobile application for customer at garage
- Mobile application for valet at garage
- Website that can allow a customer to make an account
- Website that can allow a customer to reserve a spot
- Website that can allow a manager to see statistics about the garage
- Algorithm that can maximize parking
- Display to show availability of garage at certain times

With our automated parking system as a foundation, others may be able to come along and continue our work by implementing various expansions. This can include concurrency systems to handle the arrival of more than one car at a time, such as the case in major cities. Others may also be able to fully realize our original, maximized, vision of better website security and functionality with respect to having dedicated designs for a customer compared to that of a manager.

Summary of Changes

- Updated Mapping of Subsystems to Hardware
- Updated Identifying Subsystems
- Added Interaction Diagrams (Security Check/Park)
- Design Patterns: Tried implementing state pattern
- Updated User Interface Specification (Android Application Update)
- Updated Website Design and Implementation
- Updated User Effort Estimation
- Marked future work with (*)

References

1 http://www.idscanner.com/IDWedgeBT/IDWedgeBT.htm
2 Personally went and parked at garage. This reference will be used to help us figure out how to make parking better than current implementations. (Contact Justin for more information or refer to report 1)
    a Wills Eye Hospital (Philadelphia, PA)
3 http://en.wikipedia.org/wiki/Non-functional_requirement
4 http://en.wikipedia.org/wiki/Exponential_distribution
5 http://en.wikipedia.org/wiki/Gantt_charts
6 http://en.wikipedia.org/wiki/Software_componentry
7 http://en.wikipedia.org/wiki/Event-driven_architecture
8 http://en.wikipedia.org/wiki/Entity-attribute-value_model