

System Specifications

Parking Garage Project

B. Goodacre, J. Jacob, R. Romanowski, R. Roman, M. Rodriguez

2/18/2011

This is the first main report of Software Engineering class. It details the specifications of the project. The team has met multiple times and worked together and independently to put together this report that will keep up on track to completing this large project.

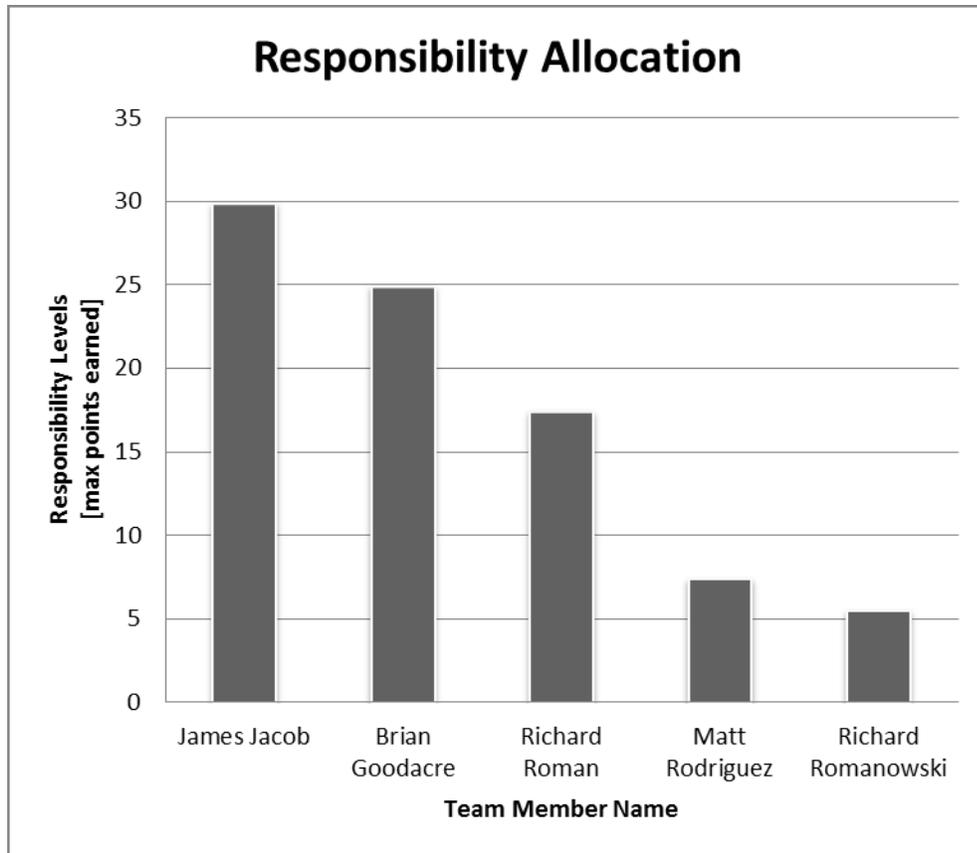
Table of Contents

Individual Contributions Breakdown.....	2
Customer Statement of Requirement	3
Essay Version	3
List Version	6
Glossary of Terms.....	7
Functional Requirements Specifications.....	9
Stakeholders	9
Actors and Goals	9
Use Cases	10
Casual Description.....	10
Fully-Dressed Description.....	11
Use Case Diagram	13
System Requirements – Use Case Traceability Matrix.....	14
System Sequence Diagrams	15
Nonfunctional Requirements	20
Domain Analysis.....	21
Domain Model.....	21
Concept Definitions.....	24
Association Definitions	25
Attribute Definitions.....	26
System Operation Contracts.....	27
Mathematical Model	28
User Interface Design	28
Preliminary Design.....	28
User Effort Estimation	31
Plan of Work.....	32
References.....	34

Individual Contributions Breakdown

Below are the estimated efforts of each team member for each of the major parts required for this report:

Responsibility	Effort	Team Member Names				
		Brian Goodacre	James Jacob	Richard Romanowski	Richard Roman	Matt Rodriguez
Project Management	1	30%	30%	14%	14%	14%
Team Effort Breakdown	1	80%	5%	5%	5%	5%
Compiling Report	3	100%				
Sec.3: Customer Statement of Requirements						
Report	6				50%	50%
Requirements	6				100%	
Sec.4: Glossary of Terms	5			70%	30%	
Sec.5: Functional Requirements Specification						
Write up	7		90%		10%	
Use Case Diagram	8	60%	40%			
Use Case Traceability Matrix	6				100%	
System Sequence Diagram	7	60%	40%			
Sec.6: Nonfunctional Requirements	4		100%			
Sec.7: Domain Analysis						
Write up	10		100%			
Domain Model	8	60%	40%			
Sec.8: User Interface Design	6			30%		70%
Sec.9: Plan of Work	7	100%				



Customer Statement of Requirement

Essay Version

This project is designed to fully automate a parking garage. Another goal is to increase profit by using methods to achieve optimum occupancy. At the moment, the parking garage is being run with no form of computerized system. Additionally, the system is being organized by employees entering the information manually into excel type programs. Additionally people either pay at machines, or at tollbooths at the exit. Currently, people are allowed to park as long as they want, and pay an hourly rate. Its quite apparent that a system like this would be likely to have many issues in terms of accuracy as well as overall efficiency and speed. Economically it is also impractical to have people in charge of checking parking spots as well as writing down all of the customer information manually when a computer and a camera could achieve the same thing faster, cleaner, and be cheaper. The new system will encourage customers to make reservations through the website, or mobile phone app. This reservation will include time and duration of stay, and will allow the customer to choose certain parking preferences such as distance from the elevator or floor level. Because one car can also be registered to two different accounts, at the time of reservation, the user will have the option of inputting the license plate number of the car that he/she will be using.

The garage is currently being remodeled so that the parking decks above the ground level will only be accessible via an elevator that will lift the cars to different decks. Additionally there will be a door in the elevator that will open to allow for ground floor access. There will be a

System Specifications: Parking Garage Project
14:332:452 SOFTWARE ENGINEERING

ramp that the cars can use to exit the garage so there is no two-way traffic at any time to prevent collisions and drastically decrease the chances of an accident. The ground level of the garage will be reserved specifically for walk-ins. When that fills up, walk-ins will be permitted to go to other floors provided that there are open and unreserved spots for the duration given by the customer at the time of entrance. The garage will rely on cameras rather than electronic tagging for vehicle recognition. Customers will be required to register at the company website in advance to making reservations at the parking garage. The customer will be asked to provide demographic as well as personal information as well as a valid email and credit card number. The customer will not be tied to his license plate number since many drivers have access to multiple cars but it will be possible to input a license plate number to associate with the customer profile. The system will also have support for guaranteed reservations, which will require the customer to sign a monthly contract with the garage for a guaranteed spot. This will be appealing to commuters as well as corporate customers who wish to keep spaces reserved for their employees and visitors. Such customers are very valuable to the garage as they provide a guaranteed amount of steady income. Additionally, these customers have the option of informing the garage that they will not be using the spot on any given day. This helps because it allows the garage to sell off one more spot.

The system will rely on cameras, sensors, and a database to manage the garage which will seemingly put a little bit of pressure in terms of data accuracy on the customer. For example if the license plate recognition system doesn't recognize the registration number, the elevator will not move causing the customer to have to input their membership information such as a unique membership ID number before they are allowed to progress to their spot. This also happens if the car has no front license plate. If the user is registered to an account, he/she will have the option of adding the car license plate number to his/her account.

Customers who are registered may be allowed to take a walk-in parking spot without a reservation if there are available spots at the time. If the vehicle registration number is recognized, and the system cannot find an existing reservation for that customer or vehicle, then the customer will be able to specify the expected duration and time of departure using the keypad in the elevator. They are referred to as registered walk-ins. For every reservation there will be a half hour grace period, in which the spot will be held for the patron who made the reservation. After the expiration of the grace period, the parking spot will be marked unreserved and added to the list of open parking spots. A customer who does not show up without notifying the garage that they will not be arriving will be billed for the entire duration of the interval they initially signed up for. A customer who does show up, but after the grace period will be allowed to extend their reservation, assuming there are spots available to accommodate their needs, otherwise they will be out of luck. The customer will be billed from the start time detailed in the reservation, until the end time, or if time was extended, the new end time.

A customer also has some flexibility if they want to stay longer than they were expecting to stay. A customer can extend their session up to a half-hour before the scheduled expiration granted there are not any conflicting reservations with spots. They can extend an unlimited amount of times as long as there are still unreserved spots available. In contrast, the customer will be billed at a higher rate for every minute he or she spends in their spot after their expected and set duration time. There will be an immediate email notification when this occurs.

System Specifications: Parking Garage Project
14:332:452 SOFTWARE ENGINEERING

As with any business that depends on predetermined duration, people will leave earlier than expected. In this case, the customer will still be billed for the full duration. The spot will then be marked vacant and will be added to the list of open spots.

The customer is allowed to have three standing reservations, as long as each reservation is at least one hour apart from each other. This is meant to encourage people to merge multiple reservations into a single reservation. One problem that may arise frequently is when someone has a reservation that relies on someone leaving at a very close time to his/her arrival. If said person in the spot doesn't leave at the expected time someone might have to wait for the spot. If a spot is open at the time than the waiting customer will be redirected to that spot, otherwise it will be a problem since there will be a waiting customer with no spot to park in. There be rain checks for when this occurs and for the times when the garage is full due to overbooking. Rain checks will allow a free parking reservation of equal or lesser time, on a different day. We assume that people will leave the premises when they are asked to and will not take an open spot that is most likely reserved for someone else.

The payment system for customers will be entirely electronic. Guaranteed reservations will pay a monthly bill that secures the spot for the specified days and times. Customers that make a reservation, will be billed with the given credit card number, at the conclusion of the reservation. Walk-ins will be billed at the entrance after providing a credit card number to the keypad.

While much of the success of the system relies on the customers being able to easily navigate the user interface of the system, it also requires that the electronic devices being used work effectively when integrated into the system. One of the most important devices is our license plate reading camera. There will be one in the elevator at the entrance and one at the exit. The first camera will attempt to match the license plate number with an account and any standing reservations that the account has. The second will read the license plate numbers of cars exiting to report when they exit and how they correspond to the reservation, or specified duration (walk-ins). It will be able to report if a car has overstayed, left early, or left on time. For this project we assume that the initial camera always can read the license plate and only does not recognize a number if it is not registered. We assume that the exit license plate always reads the license plates.

Another device that will be necessary is the sensor on each parking spot. This sensor will only be able to report when a spot is vacant or occupied. We assume that the sensor will always work, and will only report an occupied status when a car is in the spot, and not for instance when the garage is dark. We also assume that each car parks in the assigned spot. In order to have sensors that check the car parked, the expenses for the project would be greatly increased.

There will be a display in the entrance that will indicate if there are spots available for walk-in customers. There will also need to be a digital display in the elevator that displays messages to the driver. These messages will include prompts for a keypad, and also what spot the car has been assigned. The keypad will serve the function of data input. Drivers will be able to enter their account / reservation number, license plate number, and if they are walk-ins or have a reservation. It will work with the license plate reader to make sure the registration numbers match the license plates so people can't come in and take other people's spots. We assume that the elevator will always stop at the correct floor.

In order to maximize profit, the garage will use a system of overbooking. We will use two Poisson variable equations to model the arrival and departure of cars correspondingly.

System Specifications: Parking Garage Project
14:332:452 SOFTWARE ENGINEERING

Information such as past activity by customers will be taken into account. Every spot will be taken into account for this with the exception of the spots that are reserved in guaranteed reservations. The customers that have these spots are our top customers, and the benefit of using one of the spots given a no show, is greatly outweighed by the risk of not having a spot reserved for these customers. If the latter happens, we run the risk of losing some of the best customers.

As mentioned before, we must make some assumptions in this project in order to simplify it. If we did not this project would be much more complicated, and we would run the risk of not completing it in the time allotted. We assume that if a vehicle is registered to one or more accounts, the driver both knows and has permission to use one of the account numbers. We also assume that the customer has an email address, a credit/debit card, and a mobile phone with SMS text capabilities.

All storage of information will be in an on-site database. We will be using an on-site database, instead of a remote database because it will not require extra calls to a remote site. The database will store the information in the registered accounts, the occupancy status of each spot, the current parking reservations and statistics about garage usage. The garage operator will be able to use the database to view the user accounts, set prices for different services and rates, and view past customer activity. Some of these services can be a reservation price, walk-in price, overstay penalties, and discounted off-peak hours. Customers will be able to access database information via the website or a mobile app to check parking space availability, change reservation up to a half hour before the start of the reservation, view reservation number, and extend reservation.

The system will be tested using a simulator. A website interface will be designed for the garage. This will allow the basic functions mentioned previously. The website will be able to handle concurrent requests. The simulator will consist of a program/website that will simulate actual parking. It will include arrival/departure buttons. The program will ask for entry of license plate number. If the plate number is not recognized, the account number will be requested. A walk-in option will also be available on the keypad. The program will inform the user of his/her parking spot, and the user will confirm that the car is parked correctly. Another part of the testing program will inform the database which spots are occupied and which spots are taken. This part of the program determines if there are no shows, late arrivals, extended reservations, or failures to depart on time, and determines when the spot is released. The parking operator will be able to configure the simulator with parameters such as the total capacity of the parking garage (number of floors, spots per floor, etc.), different parking rates for different services, and average arrival/departure rate for the Poisson equations. Additionally, he/she should be able to view various information including the parking statistics for certain days, the number of overbooking reservations, and the number of no shows, extensions, people who leave late, and early departures.

List Version

1. The customer shall be able to register for an account online, and be able to make reservations for parking.
2. The customer shall be able to make reservations and view reservation information for parking on the mobile smart-phone app.
3. Upon entry into the elevator, the system should be able to identify if the customer has a reservation, through either the camera recognition system or from the keypad, and transport

System Specifications: Parking Garage Project
14:332:452 SOFTWARE ENGINEERING

the user to the appropriate floor and provide the customer with his or her assigned parking spot.

4. If the customer has no reservation he or she should be able to specify walk-in status and input duration of stay. The system will then assign them an unreserved spot and transport them to the appropriate floor.
5. The system shall be able to report when a car is overstaying its reservation by using sensors in the corresponding parking spot.
6. The system shall be able to report which parking spots are open and which spots are occupied using the parking spot sensors.
7. The system shall be able to report when a car is leaving and determine if it has left on time or has overstayed its specified time slot.
8. The system shall email the customer who has overstayed the time slot and inform him or her that she will be incurring further charges.
9. The system should allow the garage manager to access database information and adjust certain values such as overbooking variables and rates.
10. The system shall be able to compute overbooking algorithms to attempt to maximize profit for the garage.
11. The system shall be able to assign parking spots to reservations and walk-ins taking into account the schedule of all of the reservations and walk-in durations.
12. The system shall be able to adjust these assignments if parking spots are not vacated when they are expected to be, or if they are vacated sooner than expected.
13. The system shall be able to charge credit / debit cards for payment, and keep a payment schedule for monthly bills.
14. The system shall be able to change (extend) reservation times easily, and rearrange the parking spot assignments accordingly.
15. The system shall be able to recognize invalid reservations (inaccurate information, or contiguous reservations).
16. The system shall be able to identify when the garage is full, and credit the accounts of those customers whose reservations cannot be fulfilled (rain checks).
17. The system shall be able to report on a display at the entrance if there are spots for walk-in customers.

Glossary of Terms

Camera – device that reads in the license plates of cars and provides this information to the system

Customer – anyone who parks in the parking garage (garage)

Database – the entity that will be storing all of the information for the garage such as user accounts, rates, and past garage data.

Device – object that will be installed in the garage, including license plate readers and occupancy sensors

Extensions – the act of redefining the end time of the reservation. Extensions can be made up to a half hour before the end of the reservation, provided there are unreserved parking spots.

Elevator display – displays the output for the customer on the elevator

Elevator interface – includes the elevator keypad (keypad) and elevator display (display) that allows customer input and output on the elevator.

Elevator keypad – allows the customer to input information on the elevator

Grace period – an allotted amount of time of 30 minutes that starts at the beginning of a reserved interval. During this holding time, the spot will not be given away even if the customer hasn't arrived to allow for lateness. This grace period can be extended to longer amounts of time for an additional fee. The customer will still be billed for this time period.

Guaranteed reservation – a type of reservation; a monthly contract that allows the customers to make a contract with the parking garage for a parking spot for a predetermined period outlining specific hours.

License plate reader – a digital camera combined with a recognition system to send license plate number of vehicle to system interface

Mobile App – this is a user interface designed for smart phones. This will provide the more basic capabilities to customers on the go. Customers will specifically be able to make reservations and view information about their reservations.

No-Show – the act of missing a reservation. The customer will still be billed for the reservation

Occupancy sensor – sensors installed in parking spots to determine if the parking spot is occupied or vacant

Overbooking – the act of accepting more reservations than available parking spots. This is a strategy used to maximize profit, and takes into account any reservations that are not fulfilled. This will be determined with an equation for a Poisson variable.

Overstay – the event in which a customer does not leave his or her spot until after the reservation has terminated. The customer will be charged an increased rate, and will be notified via email.

Rain Check – a credit that goes to the account of any customer that is unable to use reservation because of lack of available parking spots. The customer is able to make another reservation of equal or lesser duration for free in the future.

Register – to visit the company website and complete a reservation and become a registered customer. Any customer who does not register is an unregistered customer, and may only obtain reservations via a walk-in.

Registered customer – any customer who has previously registered on the company website and provided all necessary information at registration time, and who has a registration identification number associated with them.

Registration number – a number that is associated with each registered customer that uniquely identifies them

Registration time – when the customer registers with the company. The customer will provide demographic information, a valid email, and a valid credit card number. The customer may also provide license plates of his or her vehicles.

Reservation – an agreement between the parking garage company (company) and the customer to hold a parking spot in advance

Reservation confirmation number – a uniquely given number given to a customer for each reservation he or she holds. This number is used as another option to identify the customer if the license plate recognition system fails, and/or the customer is driving in a vehicle that is not registered to them.

Vehicle – any car that can be parked in the parking garage. Note that large vehicles such as large trucks or busses will be unable to use this parking garage.

Walk-in – a reservation made at time of arrival. May be made by an unregistered or registered customer

Website – the interface online that will allow customers to register accounts, and vehicles. Customers will be able to make reservations, pay bills, and view account history.

Functional Requirements Specifications

Stakeholders

- Commuters
- Parking garage owners
- Local businesses
- Parking garage employees
- Tourists & shoppers

Actors and Goals

- Customer (Initiating)
 - To make a reservation for a parking space
 - To find his/her parking space within the garage
- Employee (Initiating)
 - To check the availability of open parking space within the garage
 - To update the parking fees for the garage
- CameraIn (Initiating and Participating)
 - To identify the license plate number of a vehicle entering the garage and send the data to the system in order to use the elevator
- CameraOut (Participating)
 - To identify the license plate number of a vehicle exiting the garage and send the data to the system in order to open the gate
- Sensor (Participating)
 - To identify whether or not a parking space is occupied and send the data to the system
 - To identify whether or not a parking space is vacant and send the data to the system
- Keypad (Participating)
 - To acquire the customer reservation number and send the data to the system in order to use the elevator
- DebitMachine (Participating)
 - To charge the customer for his/her stay within the parking garage if not registered
- VehicleIn (Initiating)
 - To navigate to the appropriate parking space within the parking garage
- VehicleOut (Initiating)

- To navigate to the parking garage exit and pay the parking fee

Use Cases

Casual Description

- updateReservations
 - Allows the customer to check his/her reservations with the parking garage and add to, change, or remove from them as necessary
- makeReservation
 - Allows the customer to make a new reservation with the parking garage
- deleteReservation
 - Allows the customer to cancel a reservation with the parking garage
- findParkingSpace
 - Allows the customer to use his/her smartphone to navigate to the assigned parking space within the garage
- checkLotStatus
 - Allows the parking garage employee to check what parking spaces are open within the parking garage
- updateParkingFees
 - Allows the parking garage employee to change the fees and time periods for parking spaces, reservations, and overtime parking
- useElevator
 - Allows the elevator camera to scan the license plate number of the incoming vehicle in order to use the elevator and open/close the elevator doors
- openDoors
 - Opens the elevator doors on the appropriate level of the parking garage for incoming vehicles to navigate to their parking space
- closeDoors
 - Closes the elevator doors when the elevator is ascending or descending
- spaceOccupied
 - Identifies that the parking space is occupied and sends this information to the system
- spaceVacated
 - Identifies that the parking space is vacant and sends this information to the system
- updateDatabase
 - Updates the database whenever any information changes, including account information, account history, garage statistics, and garage rates
- openGate
 - Opens the exit gate when either the camera has identified the vehicle license plate number and debited the corresponding account or the customer has manually paid the fee
- payFee
 - Allows an unregistered customer to manually pay their parking fee via cash or credit/debit card
- chargeCar
 - For a registered user, the duration that car has spent in the parking space has been calculated and the user charged, this includes guaranteed reservations

Fully-Dressed Description

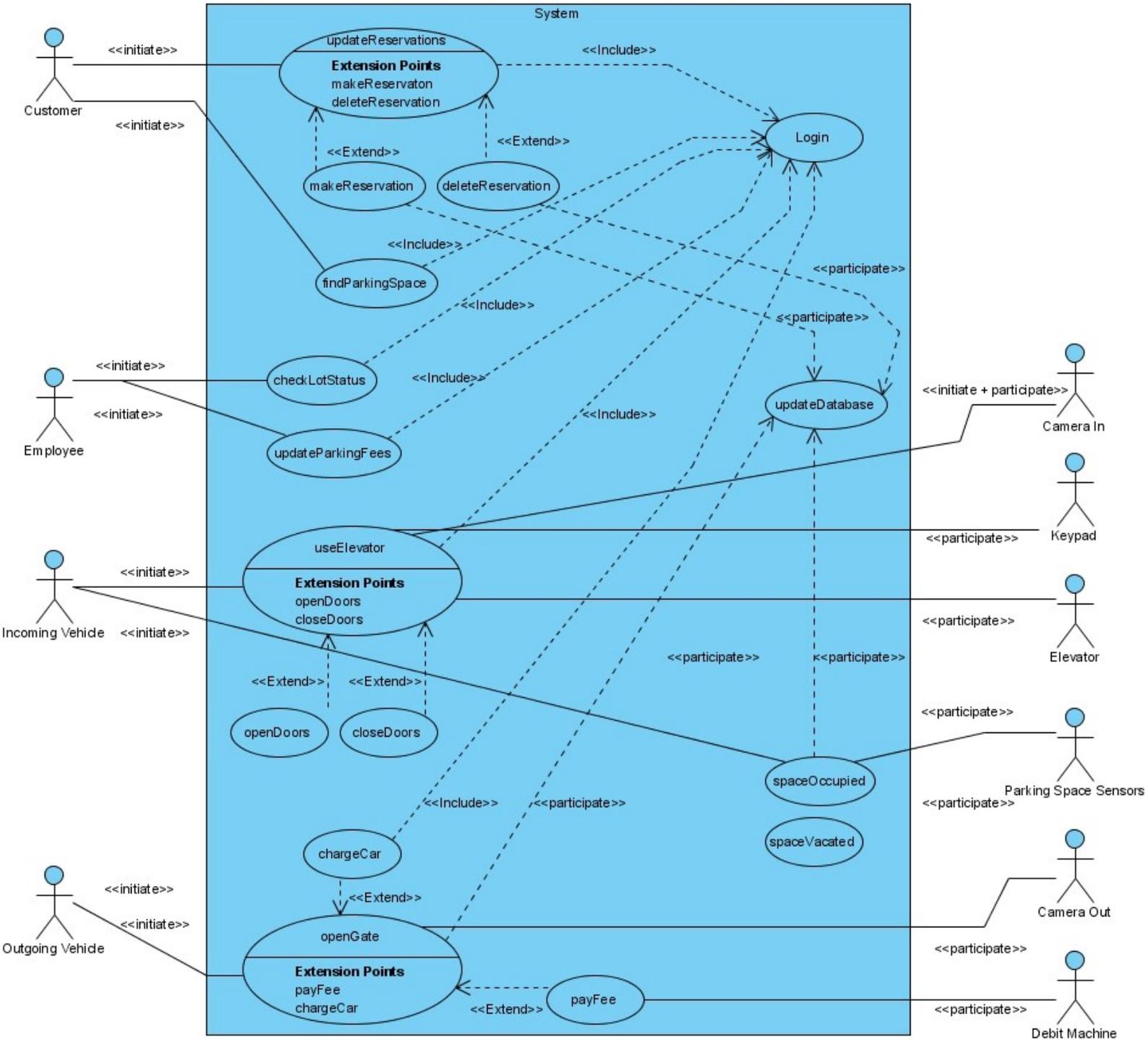
- Use Case UC-1: updateReservation
 - Related Requirements: req1, req2, req15
 - Initiating Actor: Customer
 - Actor's Goal: To make changes to his/her parking garage reservations
 - Preconditions:
 - Customer has registered with the parking garage reservation system
 - Postconditions:
 - Parking garage database is updated with new reservations
 - Flow of Events:
 - →: Customer loads parking garage service website and logs in
 - →: Customer enters updated reservation information
 - Time period, preferences, vehicle
 - Service website updates system database with new reservation
 - ←: Customer received reservation confirmation
 - →: Customer logs out of service website
- Use Case UC-2: useElevator
 - Related Requirements: req3, req4
 - Initiating Actor: VehicleIn
 - Actor's Goal: To enter the parking garage and find a parking space
 - Preconditions:
 - Elevator car is located on ground floor
 - Front door of elevator is open
 - Vehicles pulls into elevator car
 - Postconditions:
 - Elevator car returns to ground floor
 - Flow of Events:
 - →: Vehicle pulls into elevator car
 - ←: Camera scans vehicle license plate number and sends data to database
 - ←: Elevator doors close, elevator rises to appropriate floor, elevator doors open
 - →: Vehicles exits elevator car
- Use Case UC-3: openGate
 - Related Requirements: req7
 - Initiating Actor: VehicleOut
 - Actor's Goal: To pay parking fee and exit the parking garage
 - Preconditions:
 - Exit gate is lowered
 - Postconditions:
 - Exit gate is lowered again
 - Flow of Events:
 - →: Vehicle pull up to exit gate
 - ←: Camera scans vehicle license plate number and sends data to database
 - ←: Exit gate raises
 - →: Vehicle exits parking garage
- Use Case UC-4: payFee

System Specifications: Parking Garage Project
14:332:452 SOFTWARE ENGINEERING

- Related Requirements: req8, req10, req13
- Initiating Actor: VehicleOut
- Preconditions:
 - Exit gate is lowered
 - Vehicle license plate number is not registered with system
- Postconditions:
 - Exit gate is lowered again
- Flow of Events:
 - ←: DebitMachine prompts customer with payment data
 - →: Customer feeds money into debitMachine
 - ←: Exit gate raises
 - →: Vehicle exits parking garage
- Use Case UC-5: spaceOccupied
 - Related Requirements: req5, req6, req8
 - Initiating Actor: VehicleIn
 - Preconditions:
 - Parking space is currently empty
 - Postconditions:
 - Flow of Events:
 - →: Vehicle pull into parking space
 - ←: Sensor identifies that parking space occupied and updates the system

Use Case Diagram

1. Use case diagram



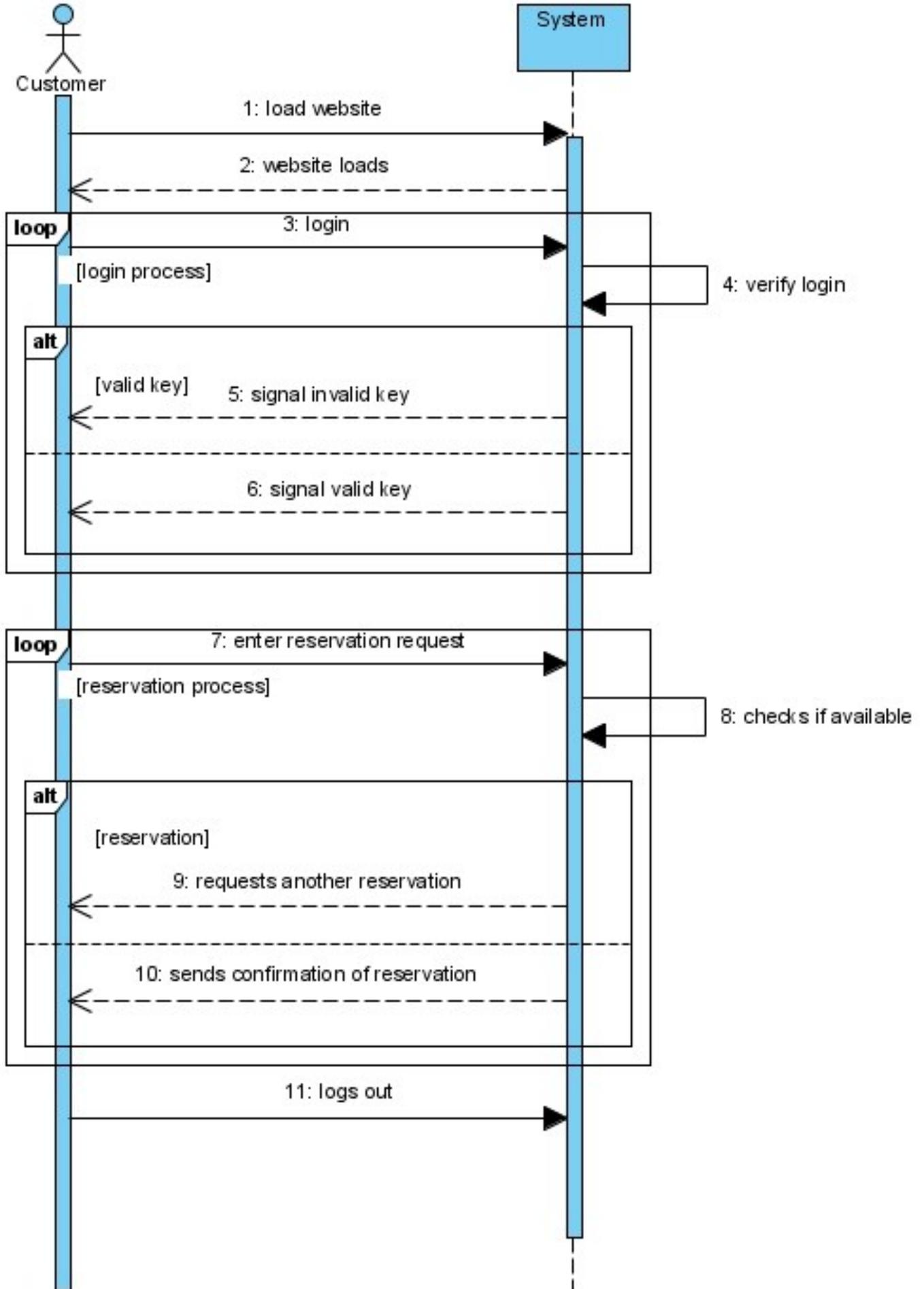
System Specifications: Parking Garage Project
14:332:452 SOFTWARE ENGINEERING

System Requirements – Use Case Traceability Matrix

	Use Cases	updateReservations	makeReservation	deleteReservation	findParkingSpace	checkLotStatus	updateParkingFees	useElevator	openDoors	closeDoors	spaceOccupied	spaceVacated	updateDatabase	openGate	payFee	chargeCar
Requirements																
Register Online		←											←			
Make Reservation Online		←											←			
Make Reservation on App		←											←			
Identify Reservation (at entrance)							←									←
Identify Walk-In (at entrance)							←							←		
Specify Duration of Walk-In													←			
Report Overstay														←	←	
Report Vacant Spot												←				
Identify exiting Vehicle												←	←	←	←	
Email Overstay												←				
Adjust Rates							←					←				
Adjust Overbooking Variables												←				
Compute Overbooking Equations												←				
Assign Parking Spot			←									←				
Adjust Parking Spot Assignment		←		←		←						←				
Charge Credit/Debit Card												←	←	←	←	
Adjust Reservation		←		←								←				
Recognize Invalid Reservation/Parking Duration							←									
Recognize Full Garage												←				
Provide Rain Check Credit												←				
View Garage Status (Employee)						←										
View Reservation Status					←											
Open Elevator Door							←	←								
Close Elevator Door							←		←							
Identify Occupied Parking Spot										←						
Pay Bill Online																←

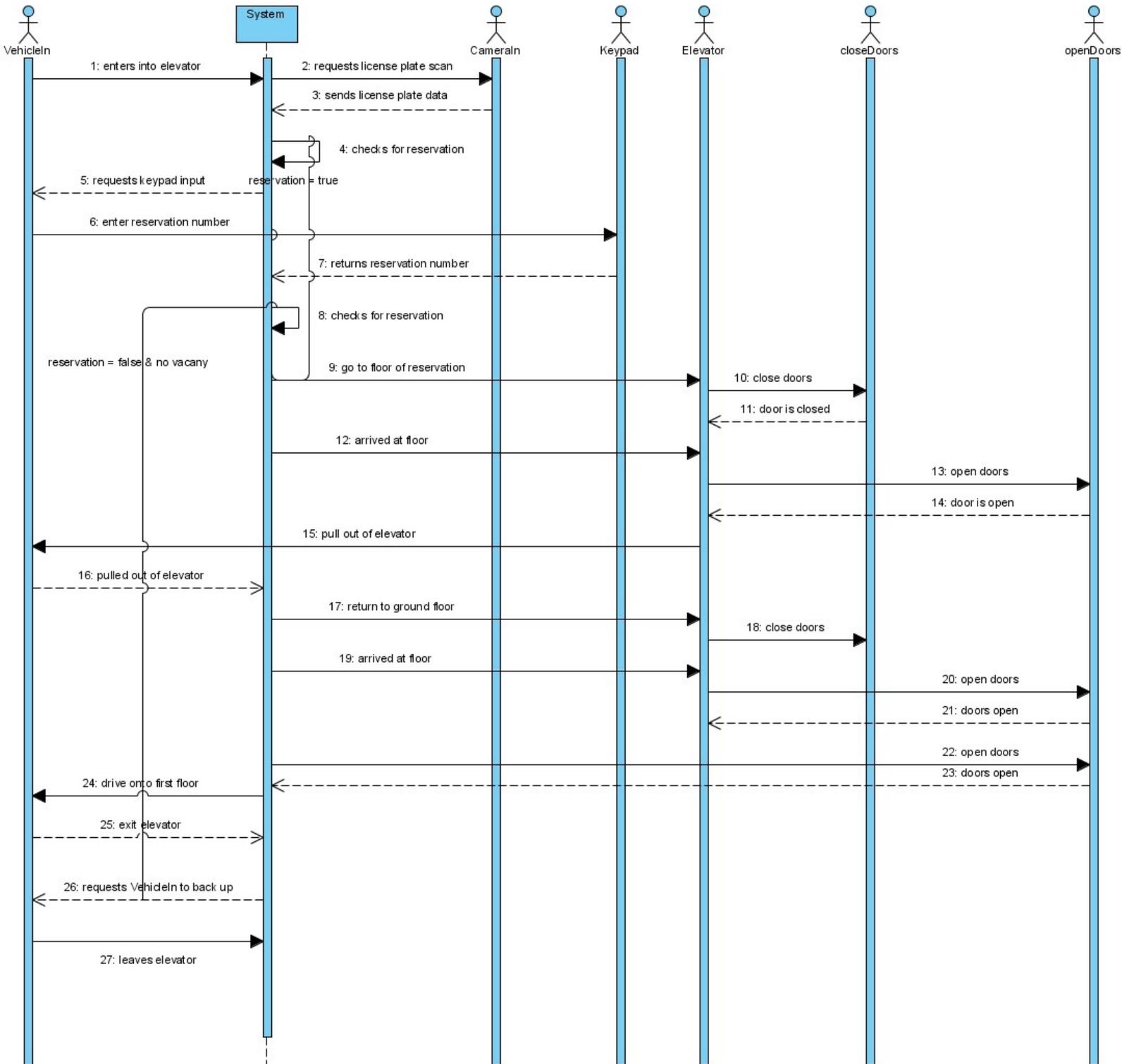
System Sequence Diagrams

1. Update Reservation
2. Use Elevator
3. Open Gate and Pay Fee
4. Space Occupied

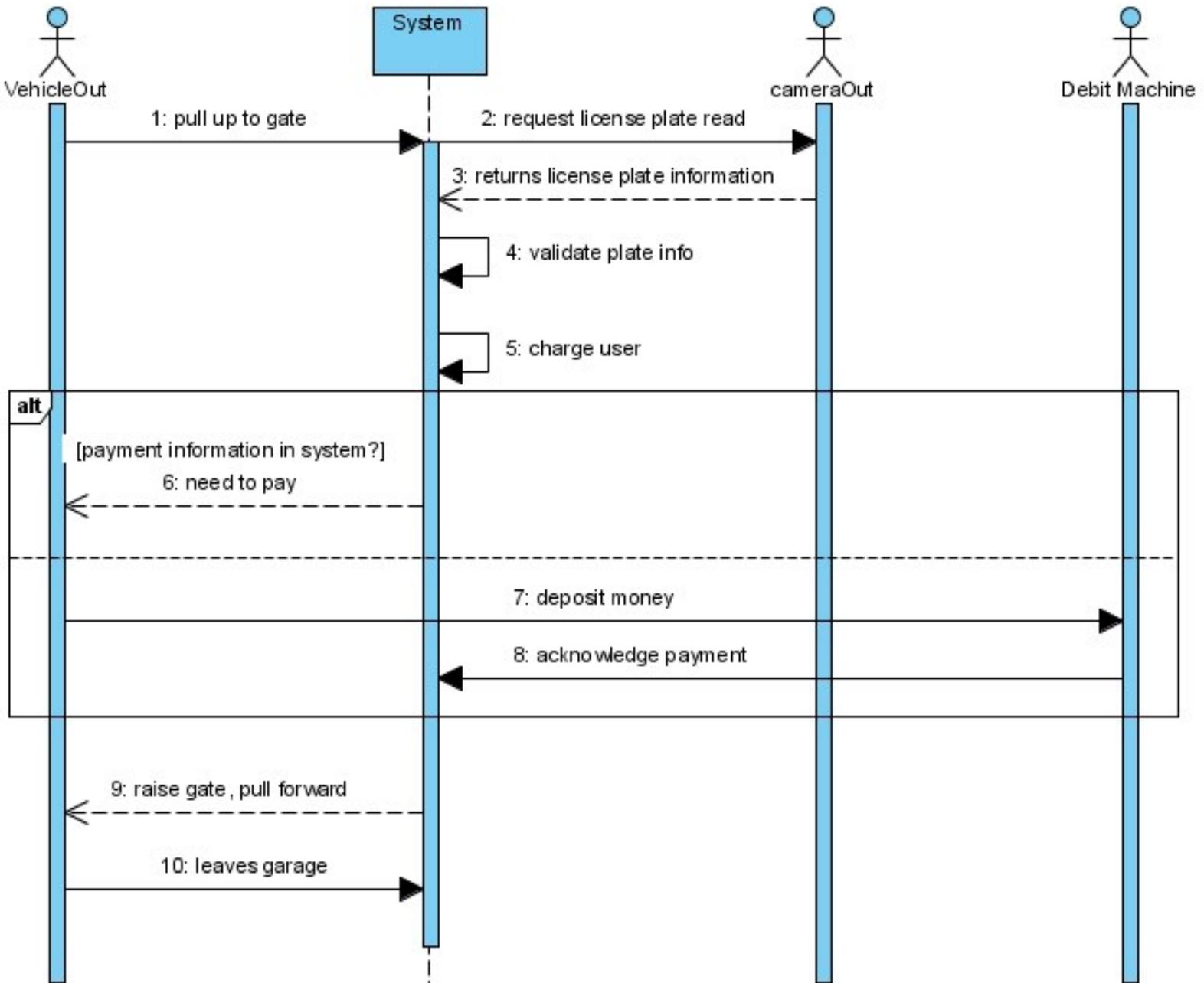


System Specifications: Parking Garage Project

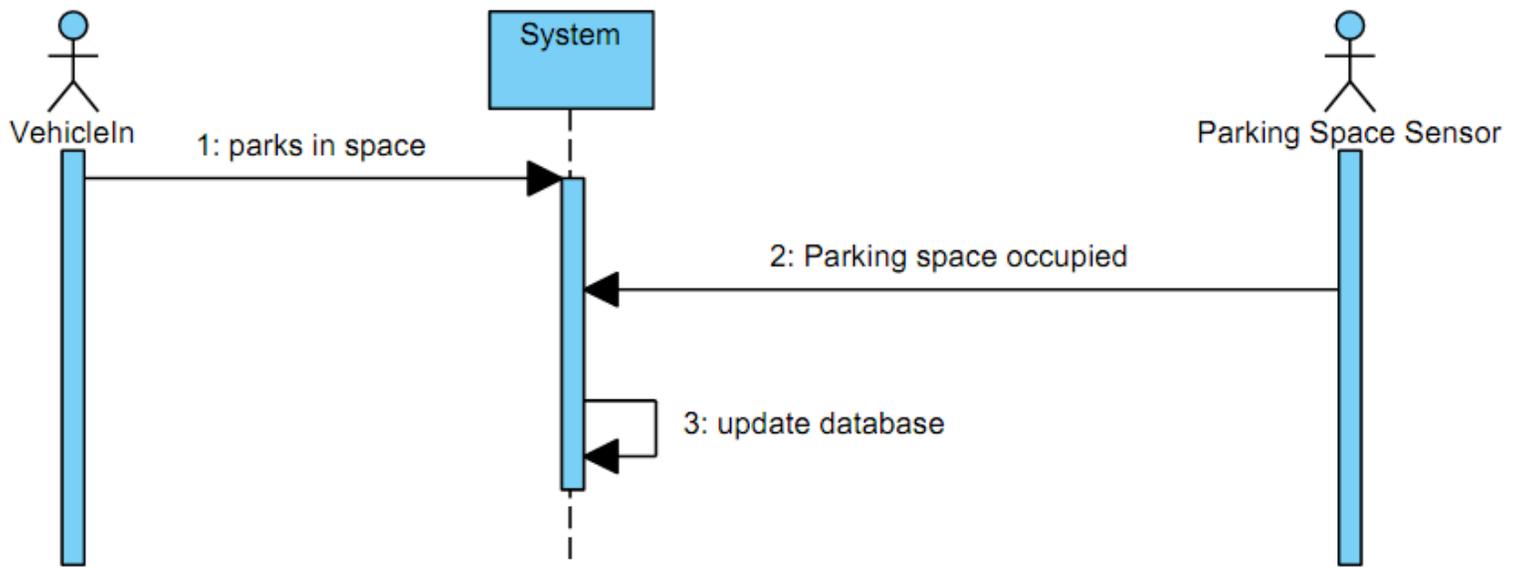
14:332:452 SOFTWARE ENGINEERING



System Specifications: Parking Garage Project
14:332:452 SOFTWARE ENGINEERING



System Specifications: Parking Garage Project
14:332:452 SOFTWARE ENGINEERING

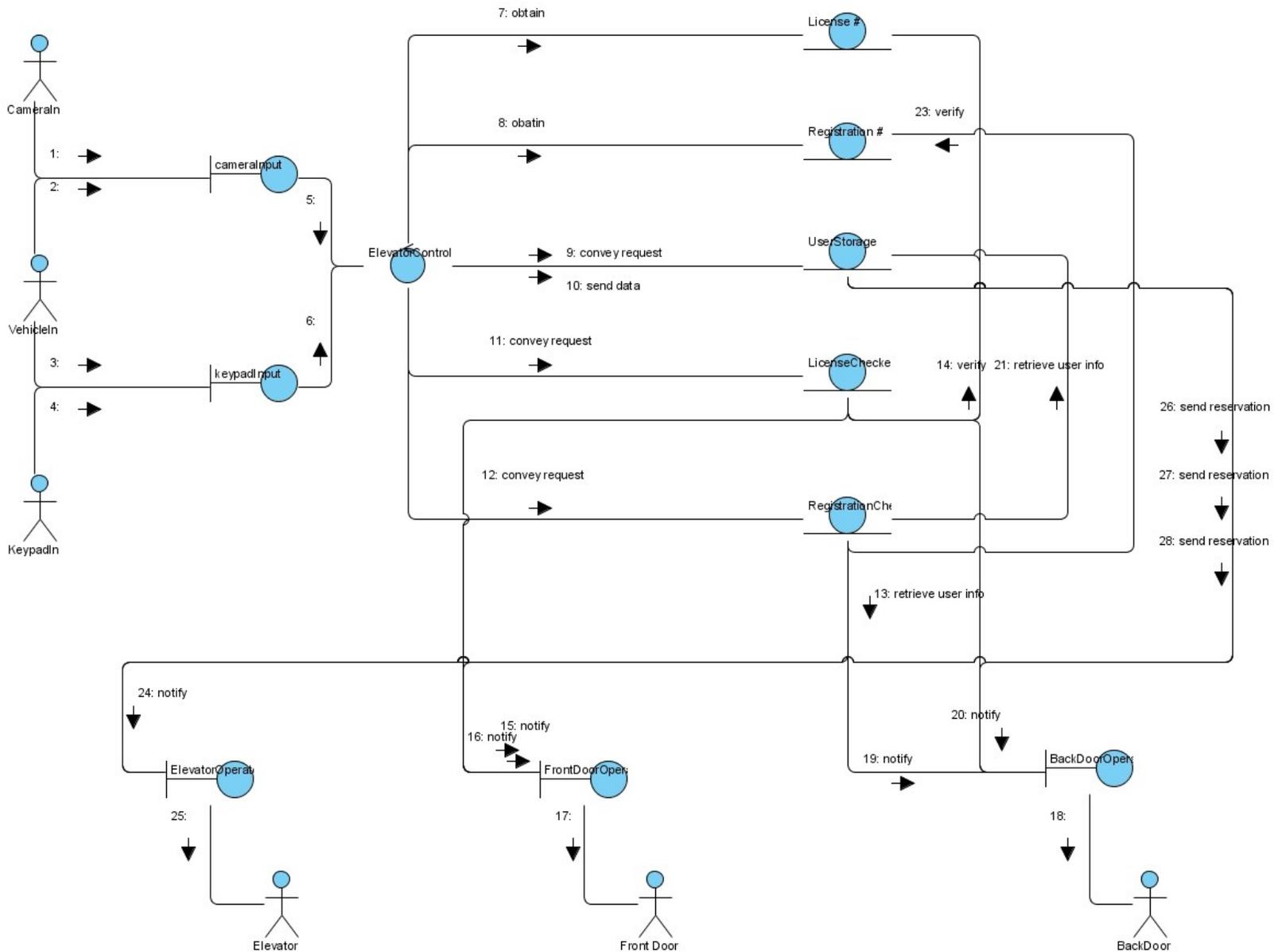


Nonfunctional Requirements

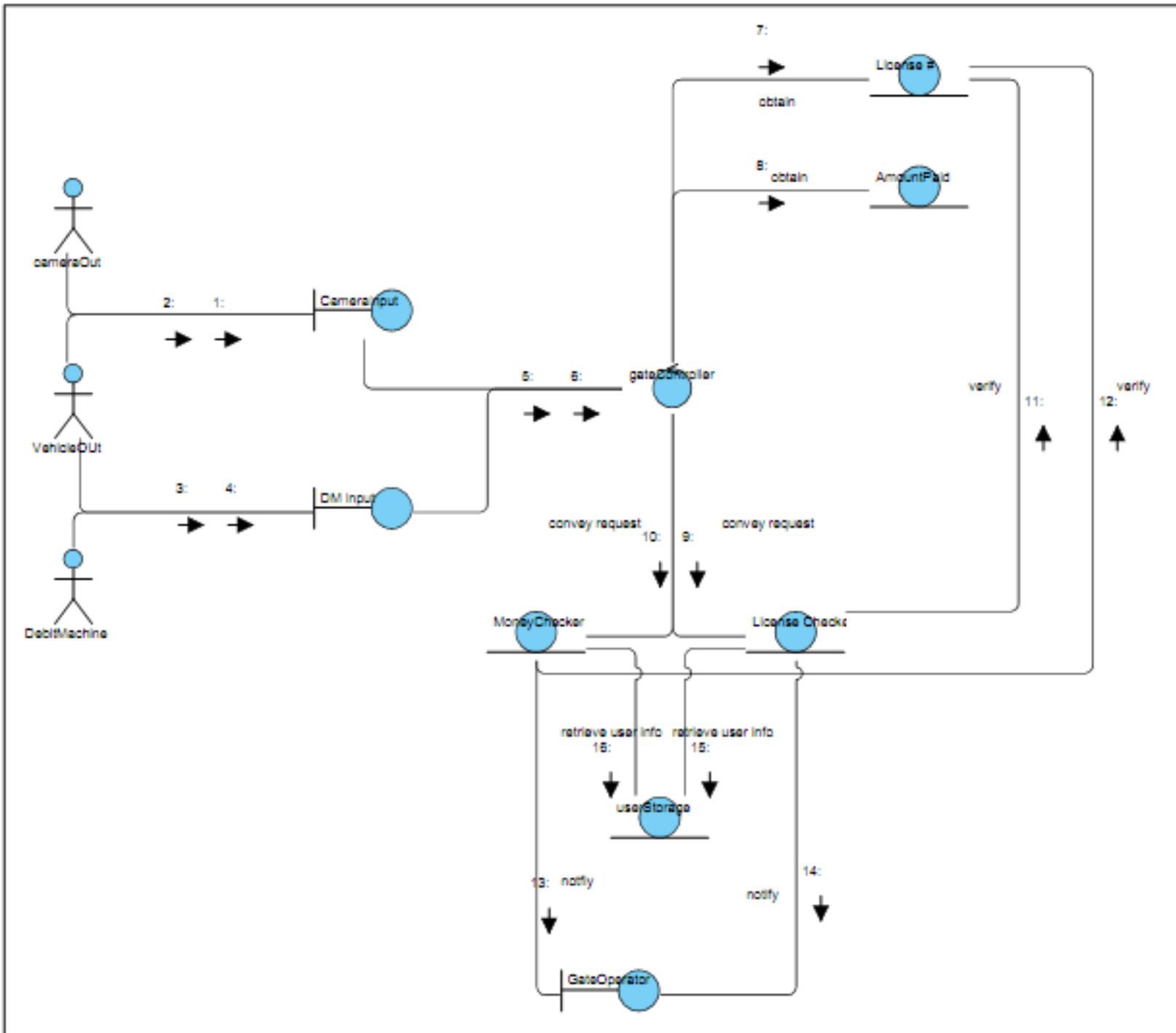
- Back-up the parking lot data from the system
 - In order to prevent data loss within the system in the event of a power failure, the parking garage data will be backed up frequently. Thus, the parking data could be loaded from the backup in the case of data loss.
- Customers reservations are always satisfied
 - Any customers that make reservations are guaranteed to have a parking space during the time period that they requested. This can be accomplished by maintaining a balance as to how many cars are inside the parking garage at all times
- Transactions within the elevator and at the exit gate are quick and easy
 - In order to keep the flow of cars in and out of the garage moving, the actions taken within the elevator and at the exit gate need to be quick and easy. The camera should be able to scan the plate quickly and the user must be able to enter their reservation information quickly on the keypad. At the exit gate, the customer must be able to see, understand, and pay their bill in a timely manner with little difficulty.

Domain Analysis

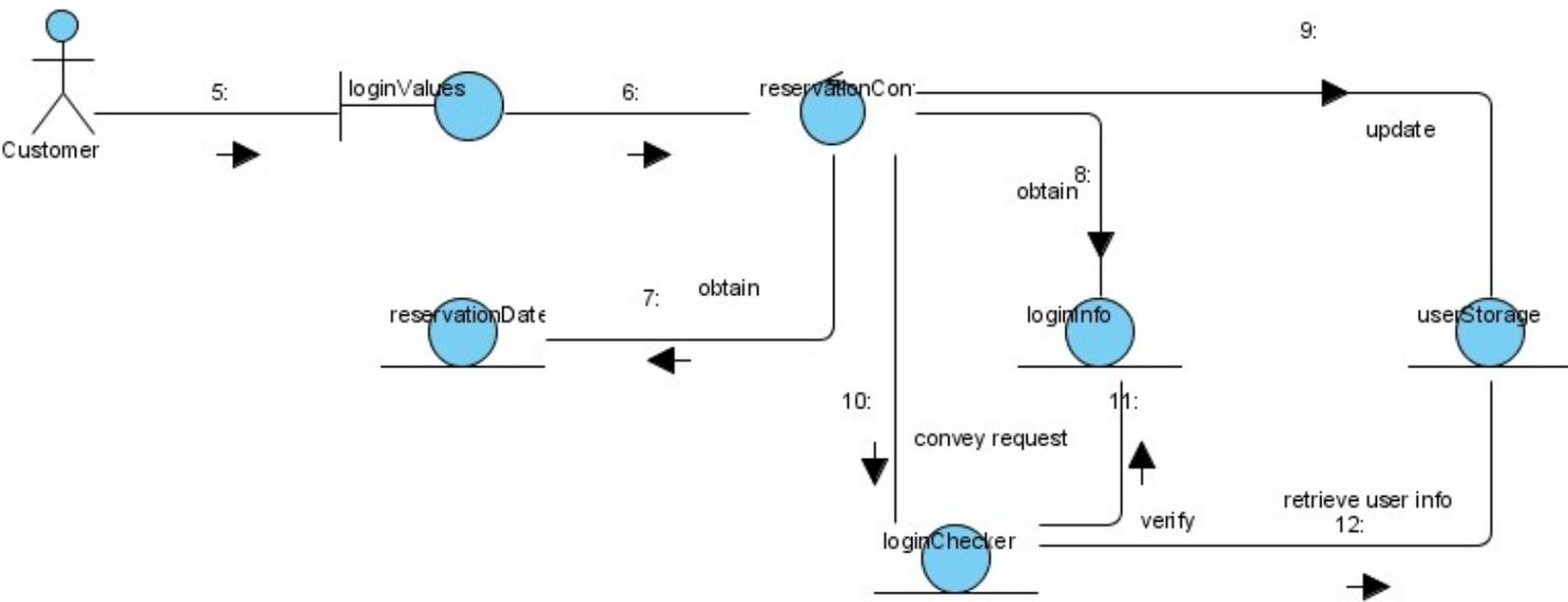
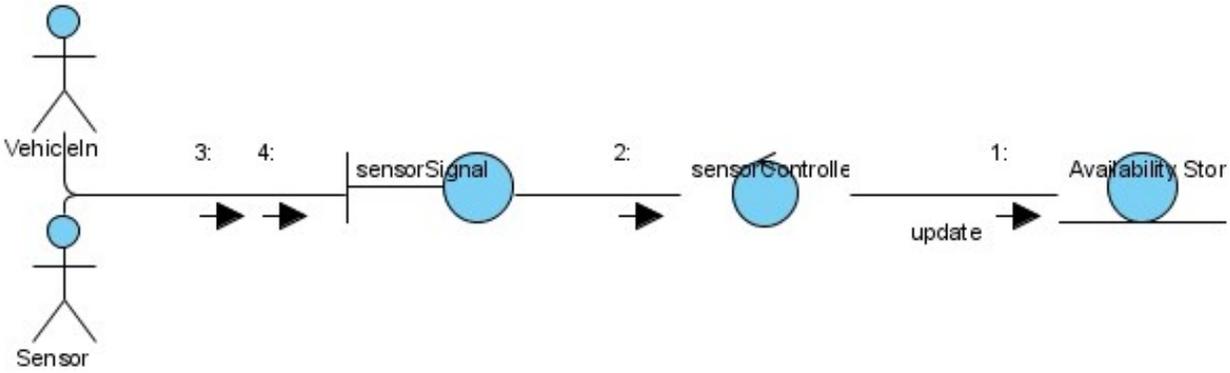
Domain Model



System Specifications: Parking Garage Project
 14:332:452 SOFTWARE ENGINEERING



System Specifications: Parking Garage Project
 14:332:452 SOFTWARE ENGINEERING



Concept Definitions

- ElevatorController (doing)
 - Coordinate actions of all concepts associated with the Use Elevator use case and delegate work to other concepts
- LicenseChecker (doing)
 - Verify if license plate number sent by Camera is registered to a user within the system
- RegistrationChecker (doing)
 - Verify if registration number sent by Keypad is valid
- LicenseNumber (knowing)
 - Container for vehicle license plate number
- RegistrationNumber (knowing)
 - Container for reservation registration number
- UserStorage (knowing)
 - Container for collection of user profiles, including registered license plate numbers, current registration numbers, and parking reservations
- FrontDoorOperator (doing)
 - Operate the front door of the elevator to open/close it
- BackDoorOperator (doing)
 - Operate the back door of the elevator to open/close it
- ElevatorOperator (doing)
 - Operate the elevator car to move it between floors
- GateController (doing)
 - Coordinate actions of all concepts associated with the Open Gate use case and delegate work to other concepts
- MoneyChecker (doing)
 - Verify if payment input by customer is appropriate amount
- AmountPaid (knowing)
 - Container for amount paid by customer at exit gate
- GateOperator (doing)
 - Operate the exit gate to open/close it
- SensorController (doing)
 - Coordinate actions of all concepts associated with the sensor use cases and delegate work to other concepts
- AvailabilityStorage (knowing)
 - Container for collection of parking space statuses
- ReservationController (doing)
 - Coordinate actions of all concepts associated with the Reservation use case and delegate work to other concepts
- LoginInfo (knowing)
 - Container for login credentials entered by user
- ReservationData (knowing)
 - Container for reservation information entered by user
- LoginChecker (doing)
 - Verify if login credentials entered by user are registered within system

Association Definitions

- ElevatorController – LicenseChecker: convey request
 - ElevatorController passes request to LicenseChecker to authenticate given license plate number
- ElevatorController – RegistrationChecker: convey request
 - ElevatorController passes request to RegistrationChecker to authenticate given registration number
- ElevatorController – LicenseNumber: obtain
 - ElevatorController obtains LicenseNumber from CameraInput
- ElevatorController – RegistrationNumber: obtain
 - ElevatorController obtains RegistrationNumber from KeypadInput
- ElevatorController – UserStorage: convey request
 - ElevatorController passes request to UserStorage to send reservation to boundary concept
- ElevatorController – UserStorage: send data
 - ElevatorController sends license plate number to UserStorage to be stored
- LicenseChecker – UserStorage: retrieve user info
 - LicenseChecker retrieve user information from UserStorage
- LicenseChecker – LicenseNumber: verify
 - LicenseChecker verifies LicenseNumber to make sure it is valid
- LicenseChecker – ElevatorOperator: notify
 - LicenseChecker notifies ElevatorOperator to raise/lower the elevator to appropriate floor
- LicenseChecker – FrontDoorOperator: notify
 - LicenseChecker notifies FrontDoorOperator to open/close the front door of elevator
- LicenseChecker – BackDoorOperator: notify
 - LicenseChecker notifies BackDoorOperator to open/close the back door of elevator
- RegistrationChecker – UserStorage: retrieve user info
 - RegistrationChecker retrieve user information from UserStorage
- RegistrationChecker – RegistrationNumber: verify
 - RegistrationChecker verifies RegistrationNumber to make sure it is valid
- RegistrationChecker – ElevatorOperator: notify
 - RegistrationChecker notifies ElevatorOperator to raise/lower the elevator to appropriate floor
- RegistrationChecker – FrontDoorOperator: notify
 - RegistrationChecker notifies FrontDoorOperator to open/close the front door of elevator
- RegistrationChecker – BackDoorOperator: notify
 - RegistrationChecker notifies BackDoorOperator to open/close the back door of elevator
- UserStorage – FrontDoorOperator: send reservation
 - UserStorage sends reservation info to FrontDoorOperator so door is opened/closed at appropriate floor
- UserStorage – BackDoorOperator: send reservation

System Specifications: Parking Garage Project
14:332:452 SOFTWARE ENGINEERING

- UserStorage sends reservation info to BackDoorOperator so door is opened/closed at appropriate floor
- UserStorage – ElevatorOperator: send reservation
 - UserStorage sends reservation info to ElevatorOperator so elevator car moves to appropriate floor
- GateController – LicenseNumber: obtain
 - GateController obtains LicenseNumber from CameraInput
- GateController – AmountPaid: obtain
 - GateController obtains AmountPaid from DMInput
- GateController – LicenseChecker: convey request
 - GateController requests that LicenseChecker verify the validity of LicenseNumber
- GateController – MoneyChecker: convey request
 - GateController requests that MoneyChecker verify that AmountPaid is sufficient
- LicenseChecker – LicenseNumber: verify
 - LicenseChecker verifies the validity of LicenseNumber
- LicenseChecker – UserStorage: retrieve user info
 - LicenseChecker retrieves user information from UserStorage
- LicenseChecker – GateOperator: notify
 - LicenseChecker notifies GateOperator to open/close gate
- MoneyChecker – AmountPaid: verify
 - MoneyChecker verifies AmountPaid is sufficient
- MoneyChecker – UserStorage: retrieve user info
 - MoneyChecker retrieves user information from UserStorage
- MoneyChecker – GateOperator: notify
 - MoneyChecker notifies GateOperator to open/close gate
- SensorController – AvailabilityStorage: update
 - SensorController updates the AvailabilityStorage with new data
- ReservationController – LoginInfo: obtain
 - ReservationController obtains LoginInfo from customer
- ReservationController – ReservationData: obtain
 - ReservationController obtains ReservationData from customer
- ReservationController – UserStorage: update
 - ReservationController update UserStorage with new ReservationData
- ReservationController – LoginChecker: convey request
 - ReservationController request that LoginChecker verify validity of LoginInfo
- LoginChecker – UserStorage: retrieve user info
 - LoginChecker retrieve user information from UserStorage
- LoginChecker – LoginInfo: verify
 - LoginChecker verify validity of LoginInfo

Attribute Definitions

- LicenseNumber
 - licenseString: String of license number characters from camera input
- RegistrationNumber
 - registrationString: String of registration number characters from keypad input

System Specifications: Parking Garage Project
14:332:452 SOFTWARE ENGINEERING

- attempt: Value of number of times registration string entered with no success
- FrontDoorOperator
 - frontDoorStatus: Current position of the front door (open or closed)
- BackDoorOperator
 - backDoorStatus: Current position of the back door (open or closed)
- ElevatorOperator
 - elevatorStatus: Current activity of elevator car (waiting or transitioning)
 - elevatorNow: Current floor the elevator car is on
 - elevatorNext: Floor elevator car is currently moving too
- AmountPaid
 - paidValue: Numeral value of amount paid from debit machine input
- GateOperator
 - gateStatus: Current position of gate (open or closed)
- LoginInfo
 - username: String of username characters from customer input
 - password: String of password characters from customer input
 - loginAttempt: Value of number of times attempted to log in within time period
- ReservationData
 - timeslot: time period during which reservation is requested

System Operation Contracts

- updateReservation
 - Preconditions:
 - UserStorage is not empty
 - username < 20 characters
 - 8 characters < password < 20 characters
 - loginAttempt < 5
 - loginAttempt = 0, for first login attempt by customer
 - timeslot is a valid time period for parking garage operation
 - Postconditions:
 - loginAttempt = 0, if username and password are valid
- useElevator
 - Preconditions:
 - UserStorage is not empty
 - registrationString = 10 characters
 - attempt < 5
 - attempt = 0, for first registrationString input attempt by customer
 - frontDoorStatus = closed
 - backDoorStatus = open
 - elevatorStatus = waiting
 - elevatorNow = 1
 - elevatorNext = 0
 - Postconditions:
 - attempt = 0, if registrationString was valid
- openGate
 - Preconditions:

- UserStorage is not empty
- gateStatus = closed
- Postconditions:
 - gateStatus = open
- payFee
 - Preconditions:
 - UserStorage is not empty
 - gateStatus = closed
 - paidValue = 0
 - Postconditions:
 - paidValue \neq 0
 - gateStatus = open
- spaceOccupied
 - Preconditions:
 - None
 - Postconditions
 - None

Mathematical Model

In order to facilitate the practice of overbooking, the system has an occupancy management formula which is used in order to achieve 100% occupancy within the parking garage. Thus, the parking garage can take more reservations than parking spaces ahead of schedule with the knowledge that a certain number of reservations will be unfulfilled and that a certain number of customers will approach the garage with walk-in parking. The formula accounts for reservations of all types, unfulfilled reservations, overtime parking, undertime parking, and walk-in customers. It is:

$$\begin{aligned} & \text{Total number of open parking spaces} \\ - & \quad \text{Current reservations} \times \text{Percentage unfulfilled} \\ - & \quad \text{Predicted overtime parking} \\ - & \quad \text{Predicted walk-ins} \\ + & \quad \text{Predicted undertime parking} \\ = & \quad \text{Number of parking spaces available} \end{aligned}$$

The predictions for unfulfilled reservations, overtime parking, walk-ins, and undertime parking will be made based on prior experience collected from the parking garage database.

User Interface Design

Preliminary Design

The user interface of any system needs to be clear, user friendly, and accessible. The UI for the parking garage will be no different. In order to register for use of the garage, the customer will be required to go online to the garage website and enter in their personal as well as vehicular

information. The navigation of the website will be no harder than loading up the webpage and hitting the “Register Now” button. Upon hitting said button, the user will be asked to enter their information in order to create an account for the garage. The registration page will look something along the lines of...

PARKING GARAGE NEW USER REGISTRATION

Personal Information

First Name	<input type="text"/>		
Last Name	<input type="text"/>		
Address	<input type="text"/>		
City	<input type="text"/>	State	<input type="text"/>
Zip Code	<input type="text"/>		
Email Address	<input type="text"/>		

Vehicle Information

License Plate	<input type="text"/>	Car Year	<input type="text"/>
Car Make	<input type="text"/>	Car Model	<input type="text"/>
Insurance	<input type="text"/>		

Finished, Create My Account

Obviously there will be another page for credit card information assuming the customer wants to put himself on a payment schedule or subscription based payment system, or even make a reservation in advance, but this is what the general registration site would need to look like. Upon registering their information and vehicle they will be given a registration number that will link their license plate to their account and give them a unique ID for use at the parking garage.

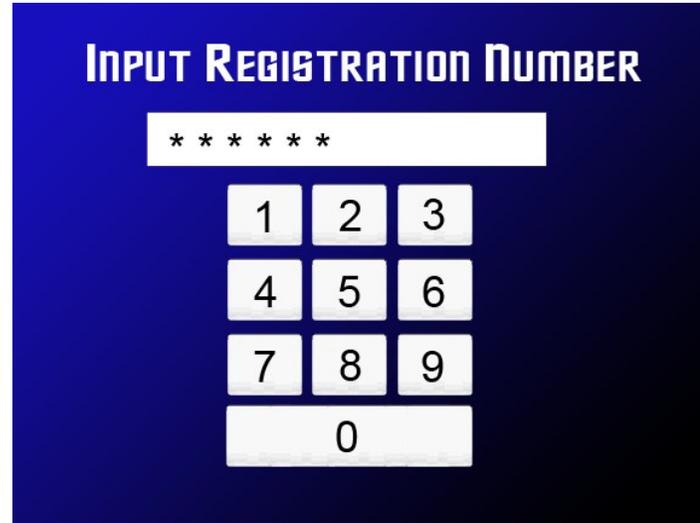
Upon arriving at the garage with the unique ID registered the customer will be prompted with a welcome screen which will pop up on the touch menu at the garage entrance which will look something along the lines of the menu on the left. The user will have the option to choose



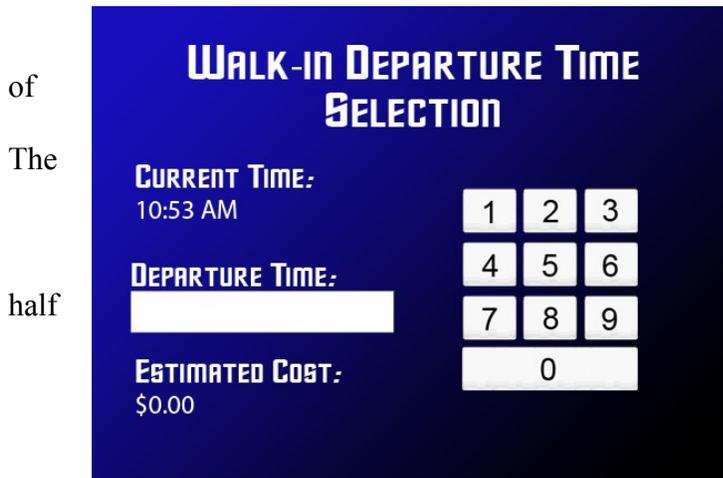
either a reservation or a walk –in depending on which one they are doing. A reservation would require either a call-ahead manually input by the current worker at the garage or an online input of their reservation. The customer upon choosing which of the two options fits their needs for the day will be brought to another menu in order to get their registration number to check for their reservation status or set up a walk-in reservation on the spot.

The menu for the reservation option will prompt the

customer for their number with a screen along the lines of the following menu. The registration number will be the number associated with the account of the user upon creation on the website. Upon a correct input of the registration number the customer will be given his/her parking spot and the gate will open to allow them to go park. The registration number will need to match the license plate number scanned by the camera at the entrance. If the registration number is input incorrectly multiple times the customer will have the option of either creating a walk-in or being asked to leave. There is not much we can do about customers not having correct information though maybe there will be a system to allow them to just check their reservation from the interface display.



If the customer selects the walk-in option at the welcome menu they will be asked what time they would like to leave. The menu would display the current time and do the math with the



departure time in order to figure out what the price is going to be at the end their stay. The reason why the price is estimated is because it is not definitive. customer may choose to leave early or maybe even extend their stay depending on the current situation and granted they give the management a hour notice. The menu will only have a keypad for numerical inputs as well as

of
The
half

maybe a forward and back button to allow the user to progress to the next screen to get their spot, or go backwards in case they chose the wrong menu.

Once the menus are successfully navigated, the customer will be shown which spot they are going to get at the lot and the elevator will allow them access to the appropriate floor. The UI will need to be very simple and quick to ensure the customers can come in and park quickly without creating a queue at the garage and causing traffic. The easier and cleaner the menu design is, the better it will function and the more appealing it will be to the customers. I do not think any sound recognition will be necessary for the system to work, since it is all visual and relies on accurately punched in information.

User Effort Estimation

Website Registration (Two Clicks and Multiple Keystrokes)

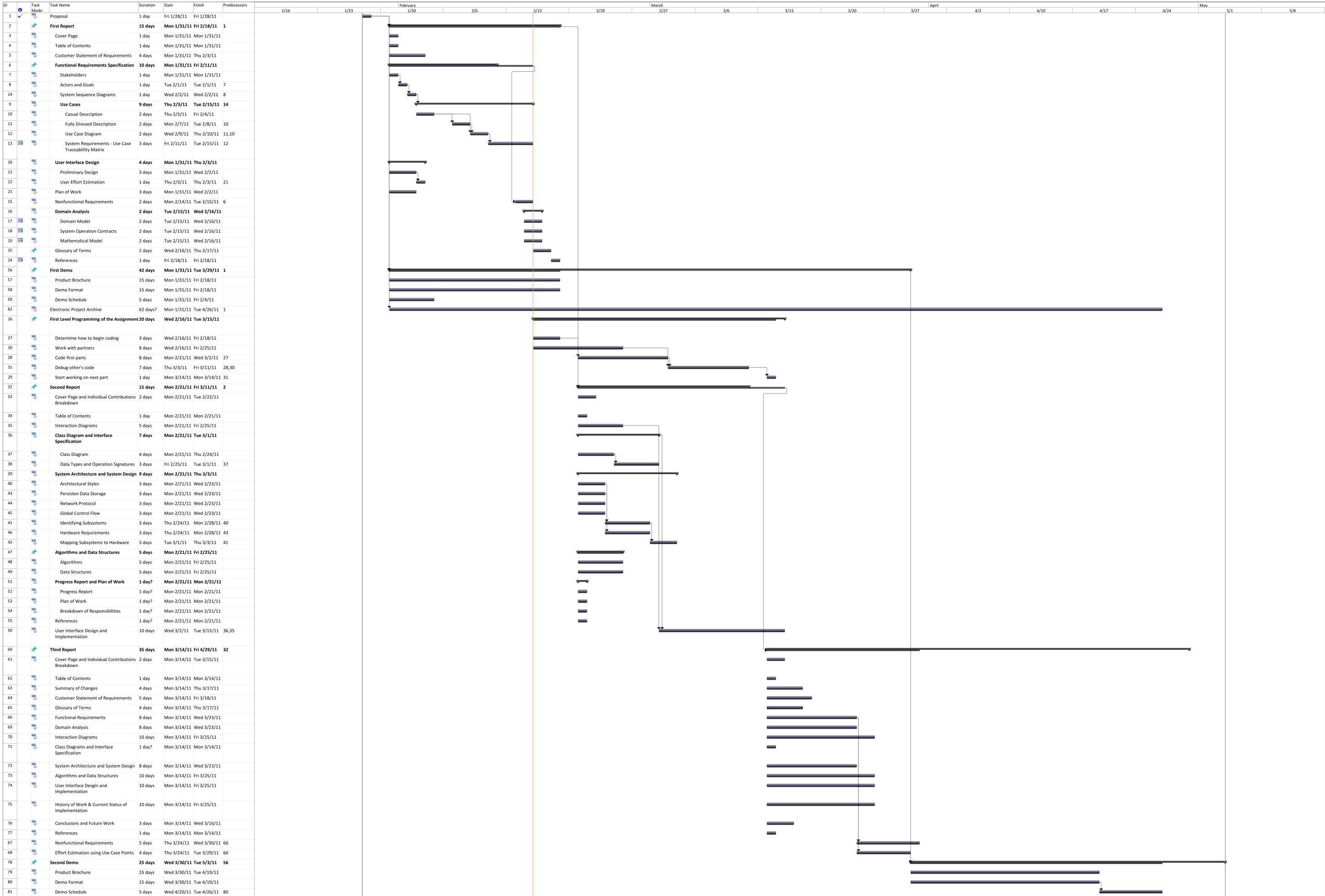
- a. Click the Register button on the homepage
- b. Use the keyboard to enter the information into the pages
- c. Click the proceed button to create an account in the garage database

UI Usage (Two to Four Clicks and Multiple Keystrokes)

- a. Select whether you are registered or a walk-in
- b. Enter your registration number or your projected departure time
- c. Click next to get your spot and your price
- d. Add in a few clicks for misclicks or messups

Plan of Work

(attached on next page)



References

There are no sources in the current document.