System Specification: Parking Garage Automation

Matt Edwards, Eric Wasserman,
Abdul Hassan, Juan Antialon, Luke Steepy

Group Number 2

Project URL: http://code.google.com/p/parking-garage-automation/
http://www.park-a-lot.vacau.com/

Effort Estimation

The table below is a breakdown of the individual member's contributions to each part of this report.

	Abdul	**Matt**	**Eric**	**Luke**	**Juan**
Project Management *(10 points)*		90%	10%		
Sec.3: Customer Statement of Requirements (*6 points*)					100%
Sec.4: Glossary of Terms (*4 points*)		80%			20%
Sec.5: Functional Requirements Specification (*37 points*)	20%	35%	35%		10%
Sec.6: Nonfunctional Requirements (*6 points*)		20%	80%		
Sec.7: Domain Analysis (*25 points*)		50%	50%		
Sec.8: User Interface Design (*8 points*)	100%				
Sec.9: Plan of Work (*3 points*)	10%	10%	10%	10%	60%
Sec.10: References (*1 point*)		100%			
Total Points Worked	15.7	40.15	31.55	0.3	12.3

Table of Contents

Customer Statement of Requirements

The main goal of this software project is to maximize the occupancy of the parking garage and develop a user friendly mechanism that helps customers find and reserve the parking spots available.

The problem that we face is the following:

Our customer owns a parking garage that lacks a computerized system, thus he/she is currently losing profit due to the fact that there is lost opportunity (not maximizing the parking spots available). There is also a congestion problem inside the garage since customers usually are wondering around trying to find a spot available. In order to increase profits, and reduce personnel, the owner wants to create a system called “Park-a-lot” that will address the issues listed.

First, to make sure this new system will work, the parking garage will undergo some changes (remodeling) so that only the ground level will be available to use without the need of the elevator. The elevator will lift the cars to the different decks above the ground level, no other way will be provided to reach the decks above the ground. There will be one descending pathway to allow cars to exit the garage through the ground level. One thing to consider is that we only allow passenger vehicles to park in the garage, which means that other vehicles such as trucks, buses are prohibited. In order to be able to accommodate as many people as possible, walk-in customers can only park in the ground level. The remaining levels will be for the registered customers use, these customers would need to make a reservation in advance. The garage will have a camera-based licence plate recognition system, thus eliminating the cost of using special tags or anything similar for identification.

In order to use the parking garage, the customers will have to register and reserve through the company’s website. During the registration process, the person will need to enter all the necessary information(i.e. name, email, phone and credit card). Providing the license plate number(s) for the vehicle is not mandatory (during the registration), though it is highly recommended, this allows the registration of customer who use rented or borrowed cars. The system will allow the same car to appear under different customers. If the customer is using a rented car or has borrowed one then he/she will need to specify the registration plate number when making a parking reservation. If this registration plate number is not the same as in the customer’s profile then the system will still allow the reservation to go through; however, the temporary association created between the customer’s profile and the registration number will be erased after the customer has finished using the reservation. Another special feature of the system will be the ability to do guaranteed reservation, which means that customers can create a simple weekly or monthly contract with the garage. For further details on guaranteed reservations, check the section Types of Reservations.

Devices

To accomplish our goals, “Park-a-lot” is implementing the following devices:

S1 & S2. The cameras will be installed to act as license-plate readers. S1 is the camera at the elevator and S2 is the exit camera. The cameras will be using the license-plate recognition system that are often used in tolls. The basic idea is that when a car arrives at the elevator platform, S1 reads the registration number and later on S2 will read the registration number of the vehicle leaving.

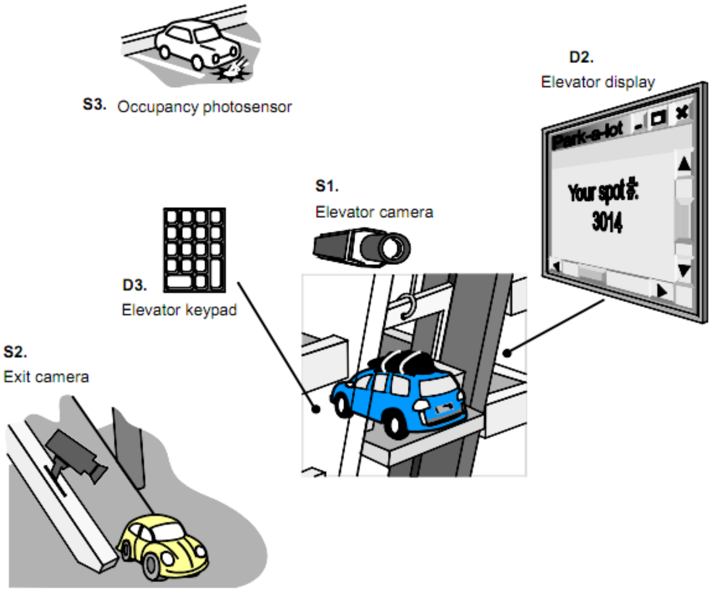
S3. There will be a sensor installed in every spot in the garage; this will help us determine whether the spot is available or not.

D1. This digital display allows walk-ins to check on the availability of the ground level parking, and has a built in credit card reader along it’s side. If the ground level is full, the display will indicate so.

D2. The digital display at the elevator will display different messages according to the specific situation. The messages that might appear in this display are: denied access to upper levels for non-registered customers or change/edit in the reservation of a registered customer.

D3. The keypad will be used in the elevator in cases when the registration number is not recognized by the camera. In this scenario, the customer will have to key in the confirmation number given by the system at the time of the reservation.

Although, this is not specifically a device, there will be a one-way barrier at the end of the exit driveway so that no vehicles will try to enter the higher levels of the garage.



Types of Customers

1. **Registered Customers** - have an online account with the parking garage, can make reservations online and receive monthly bills.
2. **Unregistered Customers** - do not have an online account and cannot make reservations in advance. These customers must pay at the time they park, and may only park on the ground level.

Types of Reservations

1. **Recurring Reservation** - Parking spaces may be reserved ahead of time and scheduled in a recurring manner (e.g. by week, month, every Monday of the month, etc.). The customer is charged a fixed rate per hour, and has the following advantages:
    a. The customer has the option to specify their grace period (in 30 minute increments). During this time their spot will not be given away, but the customer is required to pay for this grace period (hourly rate x length of grace period) even in the event of a no show. This is the definition of the **no-show penalty**.
    b. The grace period can be increased (in 30 minute increments) as long as its done so at least 30 minutes prior to the current expiration of the grace period.
2. **Confirmed Reservation** – Parking spaces may be reserved at least 15 minutes prior to arrival, in a non-recurring manner. The customer is charged an hourly rate, and has the following advantages:
    i. The customer is given a 30 minute, non-extendable grace period. If the customer misses his or her reservation, the customer will lose his or her claim to the spot.
    ii. The customer is only required to pay for the amount of time spent parking. There is no no-show penalty.
3. **Walk-In Reservation** – This type of reservation can be made by either a registered or unregistered customer, and is made upon arrival to the parking garage. Parking for unregistered customers is dependent on the available spaces in the lowest level of the garage. Registered customers may park in any open spot on any level of the garage.

Assumptions

Although our software solution will attempt to cover as many situations and scenarios as possible, the following general assumptions will be made.

A1. The camera’s license-plate recognition system does not fail, meaning it is correct all the time, regardless if the plate is dirty or has damages. Also, if the registration number is not recognized by the system then is is assumed that the car does not correspond to the registered customer.

A2. If the elevator camera’s license-plate recognition system does not identify the registration number and the customer fails to provide a correct one then the system will display a message on elevator display telling them to back up from the elevator. If this occurs then it is assumed that the customer obediently leaves the elevator.

A3. The sensors that detect the occupancy of the spots work correctly all the time, disregarding any malfunctioning of the devices. Also, every time a sensor detects occupancy it is because a vehicle is there and not another object.

A4. The elevator will lift the car to the corresponding deck and will not make any mistakes.

A5. The customer will not fail to park at his or her assigned parking spot.

A6. If the system recognizes the vehicle’s registration number then it is assumed that the customer driving the car is a registered customer. Specific scenarios in which a non-registered customer borrows a car from a registered customer and when a car is stolen from a registered customer are not considered.

A7. Organizations and companies may register accounts, meaning there may be multiple vehicles and people contained within this account.

A8. Lastly, the customer has access to his or her email and a cellphone. However it is not assumed that the customer will check his or her email frequently or that the customer has a smart phone capable of accessing web pages on the Internet or downloading apps.

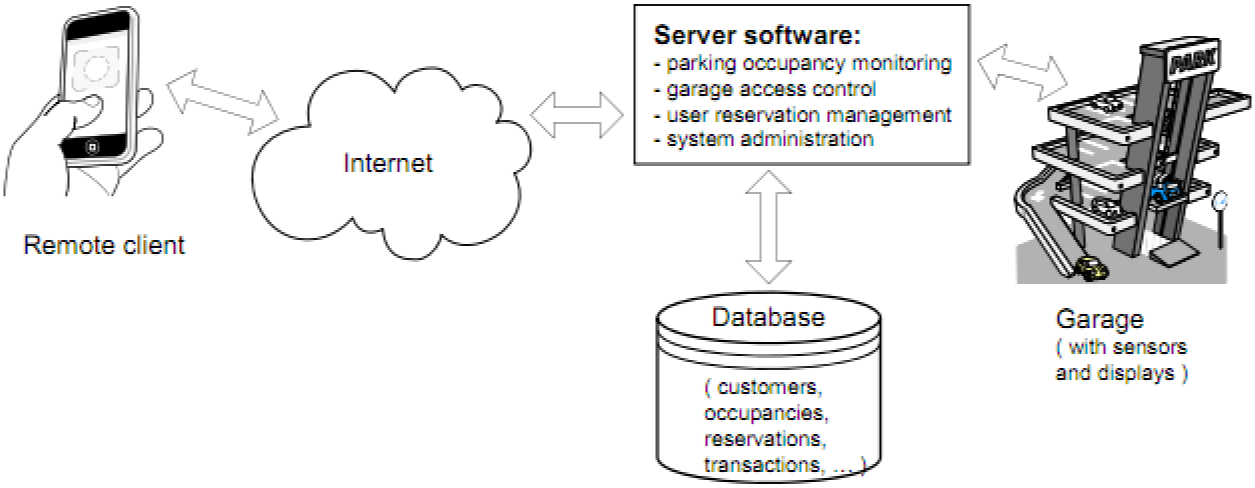
Business Policies

1. **Pricing** - A recurring reservation will cost less per hour than a confirmed reservation or walk-in, since the promise of using the spot (or facing a penalty) will increase profits. Confirmed reservations will be priced equal to walk-in reservations since there is no penalty for no shows.
2. **Cancellations** - It should be possible to cancel a reservation once it has been made but prior to the start date-time of the reservation. Any reservation can be canceled before the start date-time of the reservation, and that spot will then enter the pool of parking spaces eligible to be reserved by registered customers. Reservations cannot be canceled after the end of the grace period since the spot will have already been put back into the pool of rent-able spaces.

   a. Canceling a recurring reservation after the start date-time of the reservation but before the end of the grace period results in a charge to the customer's account for the time the spot is held.
   b. Canceling a confirmed reservation after the start date-time of the reservation but before the end of the grace period results in no charge to your account.
   c. Any cancellation immediately places the spot back into the pool of eligible to rent spots.
3. **Minimum Parking Time** - The minimum reserved parking time (for any type of reservation) will be one hour. All parking times must be in 30 minute increments.
4. **Discounts** - Discounts should be given to customers who arrive and depart consistently on time. A solid performance record would be 90-95% accuracy. Statistical data will be taken on each customer and discounts will be given accordingly.
5. **Rain Check Credit** - If a customer arrives at the parking garage with a reservation but all spots are full, the customer will be issued a credit to his or her account in the amount of 110% of the price they expected to pay for the current reservation. This credit can be applied to defray the costs of future bills, but will never be given as a cash refund - more like store credit, which can only be used to pay for future parking in the garage.

User Interaction Requirements

Below is a diagram of the "Park-a-Lot" system concept.


Remote client
Internet
Server software:
- parking occupancy monitoring
- garage access control
- user reservation management
- system administration
Database
(customers, occupancies, reservations, transactions, ...)
PARK
Garage
(with sensors and displays)

1. In order to implement this software solution successfully, there must be a database that will contain:
   a. All registered customers' information;
   b. The current state of the each spot in the garage;
   c. All the parking reservations (past, current and future);

    d. A record of all customer transactions (i.e. garage usage history, past reservations, punctuality or missed reservations).

2. The system administrator will be allowed to view the registered customers' profiles and customer statistics.
3. The system administrator is allowed to set the various prices and rates for the different services provided (i.e parking fee for a reserved interval, overstays, and the no-show fee).
4. The customer must be able to see the availability of the garage space by determining a date and time interval using the website for "Park-a-Lot". If their desired time slot is available then the system should allow the customer to make the parking reservation. After making the reservation successfully the customer will be issued a confirmation number.
5. If necessary or desired, the customer is able to modify the existing reservation(s) before the start time.
6. If necessary or desired, the customer is able to extend his or her departure time in case they need extra usage time on his or her current spot.
7. The customer needs to be notified of his or her parking spot number. Since the "Park-a-Lot" system does not have a driver-guidance system, the customer will need this number to find the correct spot.
8. If, for any circumstance, the camera's recognition system fails to recognize the vehicle's registration number then the customer will be asked to log in using the keypad and his or her customer ID number to retrieve a reservation from his or her account.

If the customer uses a phone or another device then the software has to have an easy and simple interface that allows the customer to manage his account and reservations quickly. In order to accomplish this task, the number of inputs (data entries) necessary must be kept at a minimum.

Business Requirements Table

Identifier	Requirement
REQ1	In the case that the license-plate recognition system fails to recognize the license plate number of a car then the elevator display will show a message notifying the customer that the system does not recognize the car's registration number and ask for a customer ID and password. If the customer does not have one then the system will search for available parking spots in the ground level and assign one to the customer. He or she will have to swipe a credit or debit card before going to park.
REQ2	If there are available spots in the garage, a registered customer can walk-in without a reservation. In this case, the user will still have to enter the elevator just as they would for a reservation. If the car's license plate number is recognized then they will have to enter in the desired end time on the elevator keypad. If the license plate number is not recognized, the user will have to enter his or her customer ID and password and then enter the desired park time.
REQ3	In the scenario that the customer does not show up on time for a reservation then the spot will be held for a "grace period". If the customer is able to make it before the grace period expires then he or she will be able to park for the duration of the reservation but will be billed for the the entire period reserved. Recurring reservations may have a variable grace period, the length of which is chosen at the time of making the recurring reservation. All other types of reservations have a fixed grace period of one half-hour.
REQ4	Missing a reservation occurs when a customer fails to arrive before the end of the grace period. A missed reservation cannot be fulfilled, and no spots are held in the garage to accommodate the reservation. If the customer arrives for a reservation late he or she will be prompted to park as a walk in.
REQ5	If the customer does not show up for his/her reservation or misses his/her reservation then: (a) for a recurring reservation, the customer will be charged the length of the grace period at the price per hour; (b) for a confirmed reservation, the customer will not be charged a penalty for the fixed grace period of one half-hour.

Identifier	Requirement
REQ6	In the scenario that the customer leaves the garage before his/her reservation expires then it is considered an understay. When this occurs the customer will still have to pay for the full reserved time. The sensor will recognize that the spot is vacant or available and will inform the system so that the spot can be assigned for another reservation.
REQ7	The system will allow the customer to extend a reservation (only existing reservations) at least a half hour before the expiration time if and only if there is availability for the extra period desired. This procedure can be done unlimited times as long as there are available spots in the garage.
REQ8	In the scenario that the customer does not leave on time and stays over his/her reserved time, this is called an overstay. The customer will be billed for any overage time at 1.5 times the hourly rate, known as the penalty rate.
REQ9	Each registered customer can have multiple unfulfilled reservations under his/her customer account. However, there must be at least a one hour gap in between consecutive reservations for the same license plate.
REQ10	In the scenario that a registered customer arrives to the garage but it is full because other customers did not leave at the scheduled time then the system will kindly tell the customer to leave without parking and the customer will be given a rain check credit to his/her account. Unregistered customers will simply be told there is no vacancy.
REQ11	The billing process for a registered customer will be as follows: He/she will be billed at the end of a month for charges incurred in that month, and the system will create a statement which will be emailed to the customer containing all the parking and penalty fees. The bill is automatically charged to the credit-card on file with the customer’s account.
REQ12	If a car’s registration number is associated with a single customer then the garage will bill this customer for all the usage done under that registration number. However, if the car is associated with 2 or more registered customers then the car’s usage will be billed to the customer who made the current reservation being fulfilled.
REQ13	Each garage will have a system administrator who can set and change prices and penalty fees for that garage.
REQ14	A customer may cancel a reservation up to 30 minutes before the reservation start time without being billed for any of the reservation.

Glossary of Terms

Cancelled Reservation - a reservation that has been cancelled, either by the customer at least thirty minutes prior to the start date time of the reservation or by the expiration of the grace period.

Confirmed Reservation - a reservation made in advance with a fixed grace period of thirty minutes.

Elevator Camera - an image capturing camera capable of utilizing image recognition to determine the license plate number on a vehicle inside the parking garage elevator.

Elevator Display - an LCD screen located inside the parking garage elevator for displaying information.

Elevator Keypad - a 102/105-key keyboard located inside the parking garage elevator for inputting information.

Exit Camera - same as elevator camera, and capable of using image recognition on license plates of vehicles exiting the garage.

Extended Reservation - A reservation that has been extended past the original end time.

Grace Period - the amount of time a customer is allowed to be late to check in for a reservation before the reservation is cancelled.

Missed Reservation - when a customer fails to arrive before the grace period is over.

No-Show Penalty - A fee a customer is assessed if he or she fails to fulfill a reservation before the expiration of the grace period.

Overstay - when a customer fails to depart at the scheduled time.

Penalty Rate - the rate a customer who overstays his or her reservation is charged for that overage time, 1.5 times the hourly rate.

Rain Check Credit - it is a credit given to the customer when he or she arrives at the parking garage and no spots are available to park in.

Reservation - an arrangement to park in a parking garage for a fixed amount of time at a certain fee per hour.

Recurring Reservation - a reservation, made in advance with a variable grace period, that occurs on some regularly repeating schedule.

Spot Sensors - sonar sensors capable of determining whether a parking spot inside the garage is occupied with a car or not.

System Administrator - person who will have deep access to the system, and be able to alter business requirements such a parking prices.

Understay- when a customer leaves the garage before his scheduled time of departure.

Walk In Reservation - a reservation made on-the-spot at a parking garage with a fixed grace period of thirty minutes.

Functional Requirements Specification

Stakeholders

A stakeholder is anyone who has interest in this system (users, managers, sponsors, etc.).

1. Registered Customers
2. Unregistered Customers
3. Garage Owner / Company
4. System Administrator
5. Security Personnel
6. Nearby Retail / Entertainment

Actors and Goals

An actor is anyone who will directly interact with the system. The two types of actors are initiating and participating.

1. Initiating
    a. Registered Customer
    b. Unregistered Customer
    c. System Administrator
2. Participating
    a. Elevator Keypad
    b. Elevator Display
    c. Elevator Camera
    d. Spot Sensors
    e. Exit Camera
    f. Database
    g. Event Log

Use Cases

Casual Description of Use Cases

UC-1 Reserve

A registered customer wants to make a reservation (either one-time or recurring). After being authenticated by the system, the system will show a reservation form. The user will input all required information (date and time, length of stay, etc.) and submit. The system will then validate the information submitted and create the users reservation, storing the data in the database.

UC-2 Park

A customer (registered or unregistered) arrives at the garage and wants to park. The customer drives up to the elevator, where the elevator camera will read the customers license plate number.

If the customer is a registered customer, the elevator display will show the customers reservation information as well as assign a spot to park. The elevator will then bring the car to the correct floor where the spot resides so that the user can park.

If the customer is an unregistered customer, the system will check for open spots on the ground floor, and inform the customer of open spots. The customer will then swipe his/her credit card before driving though to park.

UC-3 Manage Account

A registered customer wants to change their account details (email, license plates, credit card info, etc.). After first being authenticated by the system, the customer will be presented with a pre-filled form with all of their existing information. The customer will make whatever changes he/she wishes to make and submit. The system will validate the information and save it in the database.

UC-4 Manage Reservations

A registered customers wants to make changes to existing reservations on their account. After first being authenticated, the user will be presented with the options to either extend reservation or cancel reservation. Each of these two options are sub use cases.

UC-5 Register

An unregistered customer wants to register a new account. The system will show a registration form, which the user will fill out and submit. The system will then validate the submitted information (name, address, phone number, email, credit card info, license plate number, etc.) and store it in the database, assigning the user a unique customer id.

UC-6 Manage Garage

A system administrator wants to manage the garage remotely. After being authenticated by the system, the administrator will be presented with options to set parking prices, inspect usage history, as well as view current usage. All options are sub-use cases described later.

UC-7 Cancel Reservation

A registered customer wants to cancel an existing reservation, at least 30 minutes prior to the start of it. After first being authenticated by the system, the customer will choose which reservation they wish to cancel and submit. The system will then mark the reservation as cancelled and inform the customer.

UC-8 Extend Reservation

A registered customer wants to extend an existing reservation, at least 30 minutes prior to the expiration. After first being authenticated by the system, the customer will choose which reservation they wish to extend and submit The system will then mark the reservation as extended, change the end time, and inform the customer.

UC-9 Authenticate User

A registered customer wants to log in to the system. The system will present a log in form (email, password) which the customer will fill out and submit. The system will search for and find the customer in the database. The system will then start a session for the user, which will last until the user logs out or closes his/her browser.

UC-10 Set Prices

A system administrator wants to set the prices for his/her garage. After first being authenticated by the system, the system will present a form displaying all acceptable***** prices. The administrator will then choose a price and submit. The system will then validate and store the price in the database.

***** Acceptable prices would need to be shown so that the administrator cannot set arbitrarily high (or low) prices.

UC - 11 Inspect Usage History

A system administrator wants to view the usage history of his/her garage. After first being authenticated by the system, the system will gather all statistical data about their garage (overstay/understay percentages, etc.)

Actor’s Goal Tables

Initiating Actors	Actor’s Goal	Use Case Name
Registered Customer	To obtain a reservation for a parking spot for a given duration in advance.	Reserve (UC-1)
Registered Customer	To park in a parking spot to fulfill a reservation.	Park (UC-2)
Registered Customer	To manage the details of customer account.	ManageAccount (UC-3)
Registered Customer	To manage existing reservations in the customer’s account.	ManageReservations (UC-4)
Unregistered Customer	To park in a parking spot for a given duration.	Park (UC-2)
Unregistered Customer	To register for an account and become a Registered Customer.	Register (UC-5)
System Admin	To manage the parking garage prices and view parking usage history and statistics.	ManageGarage (UC-6)

Participating Actor	Actor’s Goal	Use Case Name
Elevator Keypad	To consume information.	UC-2
Elevator Display	To display information.	UC-2
Elevator Camera	To obtain license plate info from car.	UC-2
Spot Sensors	To determine if spot is occupied.	UC-2
Exit Camera	To determine when a car leaves.	UC-2

Fully-Dressed Description of Use Cases

Use Case UC-1: **Reserve**

Related Requirements: REQ9

Initiating Actor: Registered Customer

Actor's Goal: To reserve a parking spot for a future date and time.

Participating Actors: Database

Preconditions: Registered Customer is currently logged into the System.
include::AuthenticateUser (UC-9)

Postconditions: The System reserves the requested date and time for the customer in the Database.

Flow of Events for Main Success Scenario:

→ 1. **Registered Customer** selects menu option "Make Reservation".

→ 2. **Registered Customer** selects the desired date, start time, and end time for the reservation, along with any additional options (i.e. for a recurring reservation the customer may wish to extend the grace period).

← 3. **System** (a) checks the reservation **Database** for available reservations, (b) notifies **Registered Customer** that the reservation is made, and (c) updates the **Database** to include the new reservation.

Flow of Events for Extensions (Alternate Scenarios):

3a. **System** cannot find an available reservation for the specified date and time.

← 1. **System** (a) logs the attempted reservation and (b) signals to the **Registered Customer**.

→ 2. **Registered Customer** selects a different date and time to make a reservation.

3. Same as in step 4.

3b. **System** finds a conflicting reservation from **Registered Customer**'s account.

← 1. **System** notifies customer that he or she already has a reservation or part of a reservation during that date and time.

→ 2(a). **Registered Customer** can choose to cancel the existing reservation in favor of the new reservation. *include::CancelReservation* (UC-7). Then same as in step 4.

→ 2(b). **Registered Customer** can choose to book overlapping reservations. Then same as in step 4.

Use Case UC-2:	**Park**
Related Requirements:	REQ1, REQ2, REQ3, REQ4, REQ6, REQ8, REQ10
Initiating Actor:	Registered Customer, Unregistered Customer (collectively Customer)
Actor's Goal:	To park in the garage.
Participating Actors:	Elevator Display, Elevator Camera, Elevator Keypad, Spot Sensor Exit Camera, Database
Preconditions:	The elevator is currently empty.
Postconditions:	System marks reservation as completed in database.

Flow of Events for Main Success Scenario:

→ 1. **Customer** enters the elevator.

2. **Elevator Camera** recognizes license plate of vehicle.

← 3. **Elevator Display** displays any reservations returned from the **Database** the customer may have for the current date.

→ 4. **Customer** selects a reservation.

← 5. **System** (a) assigns an optimal spot to the **Customer** and (b) displays the parking spot on the **Elevator Display**.

→ 6. **Customer** exits the elevator.

← 7. **Spot Sensor** notifies the **System** when a vehicle is parked in the assigned parking space.

← 8. **Spot Sensor** notifies the **System** when a vehicle is no longer parked in the assigned space.

9. **Customer** exits parking garage and is recorded by **Exit Camera**.

Flow of Events for Extensions (Alternate Scenarios):

2a. **Elevator Camera** fails to recognize license plate of vehicle.

← 1. **Elevator Display** prompts **Customer** for either (1) a registered customer ID or (2) a method of payment for a walk in.

→ 1(a). **Registered Customer** enters ID number, password, and license plate number into **Elevator Keypad**.

1(b). *include::AuthenticateUser* (UC-9)

1(c). Same as in step 3.

→ 2(a). **Customer** slashes credit card to pay.

← 2(b). **Elevator Display** prompts user for reservation length.

→ 2(c). **Customer** enters reservation duration into **Elevator Keypad**.

2(d). Same as in step 5.

3a. **Customer** does not have any existing reservations.

← 1. **System** prompts user for reservation length.

→ 2. **Customer** enters reservation duration into **Elevator Keypad**.

3. Same as in step 5.

3b. **Customer** arrives after grace period has expired.

← 1. **Customer** is informed that grace period has expired and is offered the chance to park as a walk-in if space is available.

2. Same as in 5.

Use Case UC-3: **Manage Account**

Related Requirements:	None
Initiating Actor:	Registered Customer
Actor's Goal:	To edit the details of a customer's account.
Participating Actors:	Database
Preconditions:	Registered Customer is currently logged in to the system. *include::AuthenticateUser* (UC-9)
Postconditions:	Changes to Database are committed.

Flow of Events for Main Success Scenario:

→ 1. **Registered Customer** selects menu option "Manage Account".

← 2. **System** (a) displays current user account details, and (b) prompts **Registered Customer** to make changes to desired fields.

→ 3. **Registered Customer** makes the necessary changes to the form.

← 4. **System** verifies that the changes made are valid.

Use Case UC-4: **Manage Reservations**

Related Requirements:	None
Initiating Actor:	Registered Customer
Actor's Goal:	To extend or cancel existing rerservations.
Participating Actors:	Database
Preconditions:	Registered Customer is currently logged in to the system. *include::AuthenticateUser* (UC-9)
Postconditions:	Changes to Database are committed.

Flow of Events for Main Success Scenario:

→ 1. **Registered Customer** selects menu option "Manage Reservations".

← 2. **System** displays options for: (a) extending reservation, (b)cancel reservation.

→ 3. **Registered Customer** selects one of the options from step 2 and follows the corresponding instructions.

← 4. **System** (a) stores the changes made in the Database, and (b) signals to the **Registered Customer** the successful change.

Flow of Events for Extensions (Alternate Scenarios):

3a. Selected activity entails canceling reservation.

1. *include::CancelReservation*(UC-7).

3b. Selected activity entails extending reservation.

1. *include::ExtendReservation* (UC-8).

Use Case UC-5: Register

Related Requirements: None
Initiating Actor: Unregistered Customer
Actor's Goal: To create an account and become a Registered Customer.
Participating Actors: Database
Preconditions: Unregistered Customer has a valid email address with which to register.
Postconditions: The System stores all of the newly Registered Customer's information in the Database.

Flow of Events for Main Success Scenario:

→ 1. **Unregistered Customer** accesses **System** and selects menu option "Create Account".

→ 2. **Unregistered Customer** fills in personal info: name, address, state, zip, phone number, email address, password, billing credit card number, and license plate(s) and submits the info.

← 3. **System** (a) checks that all fields have been filled in, (b) verifies the email address is valid and unique within the **Database**, (c) verifies the credit card information is valid, and (d) updates the database to include the new **Registered Customer**.

→ 4. New **Registered Customer** can now make reservations in advance.

Flow of Events for Extensions (Alternate Scenarios):

3a. **System** identifies that not all of the fields have been filled in on the registration form.

← 1. **System** (a) detects error and (b) signals to the **Unregistered Customer** that they must complete the form and resubmit.

→ 2. **Unregistered Customer** fills in the missing data fields and resubmits the form.

3. Same as in step 3.

3b. **System** identifies that email address is invalid or has already been registered with the website

← 1. **System** (1) notifies the **Unregistered Customer** that the email address is invalid prompts the **Unregistered Customer** to change that information or (2) detects that the email address has already been registered in the system and alerts the **Unregistered Customer** that the email address has already been registered and to attempt to log into the account.

→ 1(a). **Unregistered Customer** changes the email address field and resubmits the form.

1(b). Same as in step 3.

2(a). **Registered Customer** leaves registration area.

3c. **System** could not verify the credit card information.

← 1. **System** prompts the **Unregistered Customer** to re-enter their credit card information.

→ 2. **Unregistered Customer** changes the information in the credit card field and resubmits.

3. Same as in step 3.

Use Case UC-6:	**Manage Garage**
Related Requirements:	REQ13
Initiating Actor:	System Admin
Actor's Goal:	To set parking prices or inspect usage history.
Participating Actors:	Database, Event Log
Preconditions:	System Admin is currently logged in to the system. *include::AuthenticateUser* (UC-9)
Postconditions:	Changes to System are committed and System Admin is logged out.

Flow of Events for Main Success Scenario:

→ 1. **System Admin** selects menu option "Manage System".

← 2. **System** displays options for: (a) setting prices, (b) inspecting parking usage history.

→ 3. **System Admin** selects one of the options from Step 2 and performs management activities.

→ 4. **System Admin** commits changes to **System**.

5. **System** verifies and commits changes to **Database**.

6. **System Admin** logs out of his or her account.

Flow of Events for Extensions (Alternate Scenarios):

3a. Selected activity entails setting parking garage prices.

1. *include::SetPrices* (UC-10).

3b. Selected activity entails viewing access history.

1. *include::InspectUsageHistory* (UC-11).

Use Case UC-7: **CancelReservation** (sub-use case)

Related Requirements: REQ14

Initiating Actor: Registered Customer

Actor's Goal: To cancel an existing reservation.

Participating Actors: Database

Preconditions: Some reservations exist for the Registered Customer in the Database.

Postconditions: Canceled Reservations are marked as canceled in the Database.

Flow of Events for Main Success Scenario:

← 1. **System** (a) displays the list of selected reservations to be canceled and (b) prompts the **Registered Customer** to confirm the action.

→ 2. **Registered Customer** selects "cancel selected reservations".

← 3. **System** updates the database to account for the canceled reservations.

Flow of Events for Extensions (Alternate Scenario):

2a. **Registered Customer** selects the "do not cancel reservations" option.

← 1. **System** returns to Manage Reservations screen.

2. *include::ManageReservations* (UC-4).

Use Case UC-8: **ExtendReservation** (sub-use case)

Related Requirements: REQ7

Initiating Actor: Registered Customer

Actor's Goal: To extend an existing reservation.

Participating Actors: Database

Preconditions: Some reservations exist for the Registered Customer in the Database.

Postconditions: Extended reservation is marked as extended in the Database and the start and/or end time is updated accordingly.

Flow of Events for Main Success Scenario:

← 1. **System** (a) displays the list of selected reservations to be extended, and (b) prompts the **user** to change the reservation end time for each reservation.

→ 2. **Registered Customer** enters the new end times for the selected reservations.

← 3. **System** (a) checks the database to see if the extensions can be made, and (b) notifies the **User** that the reservations have been extended.

Flow of Events for Extensions (Alternate Scenario):

3a. **System** identifies that the extension cannot be made due to reservation conflict..

← 1. **System** notifies the customer that the parking deck is completely booked and the reservation cannot be extended.

Use Case UC-9:	**AuthenticateUser** (sub-use case)
Related Requirements:	None
Initiating Actor:	Registered Customer, System Admin (collectively User)
Actor's Goal:	To be positiviely identified by the system.
Participating Actors:	Database
Preconditions:	The database contains user account information.
Postconditions:	

Flow of Events for Main Success Scenario:

← 1. **System** prompts the user for their customer ID and password.

→ 2. **User** supplies a valid customer ID and password.

← 3. **System** (a) verifies that the customer ID and password are valid, and (b) signals to the user the identification validity.

Flow of Events for Extensions (Alternate Scenarios):

2a. **User** enters an invalid customer ID and password combination.

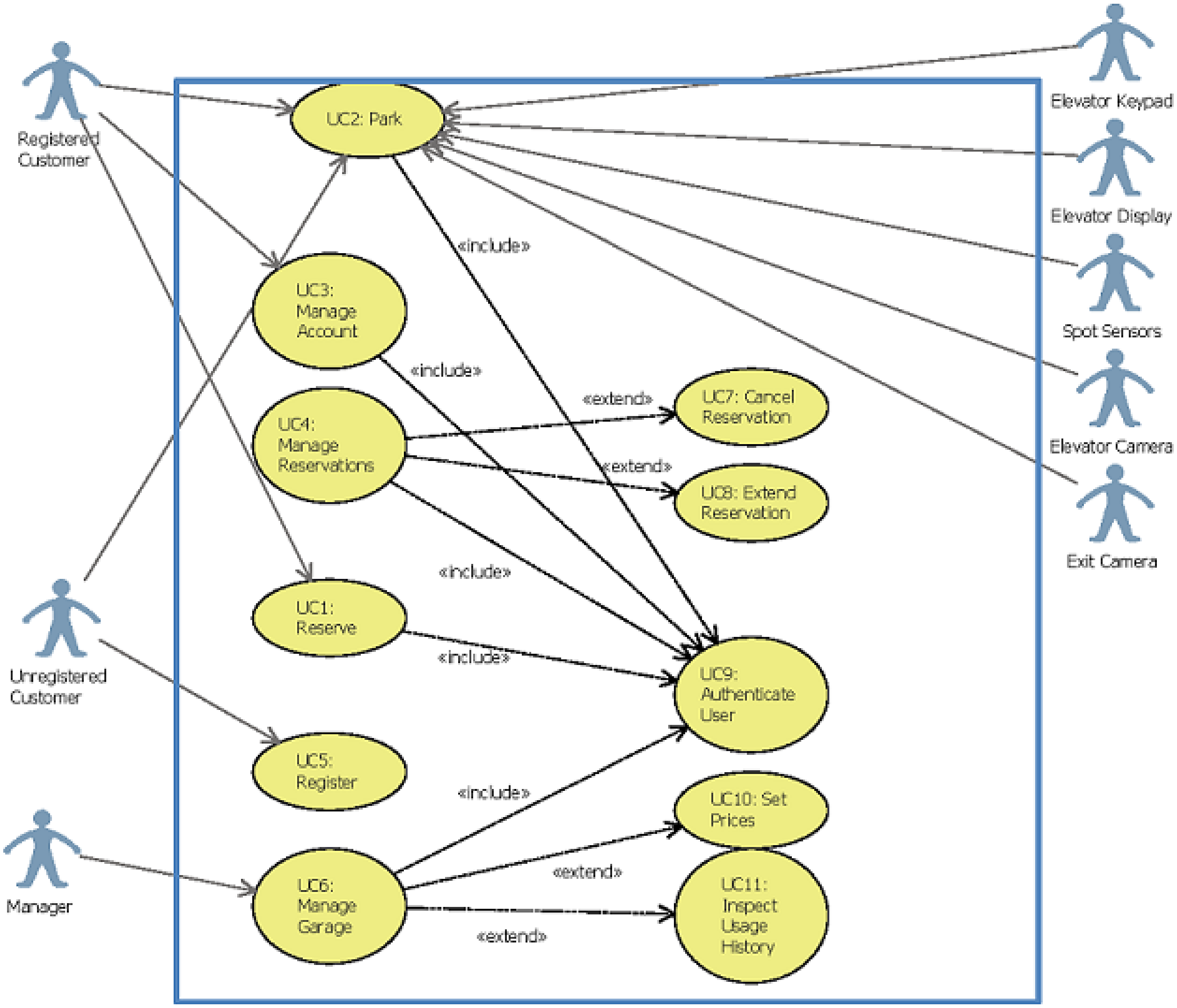
← 1. **System** (a) detects error, (b) marks a failed attempt, and (c) and signals to the **User** the credentials are invalid.

← 1(a). **System** (a) detects that the count of failed attempts exceed the maximum number, (b) informs the user to try again in 15 minutes, and (c) exits the screen.

→ 2. **User** supplies a valid customer ID and password.

3. Same as in step 3.

Use Case Diagram

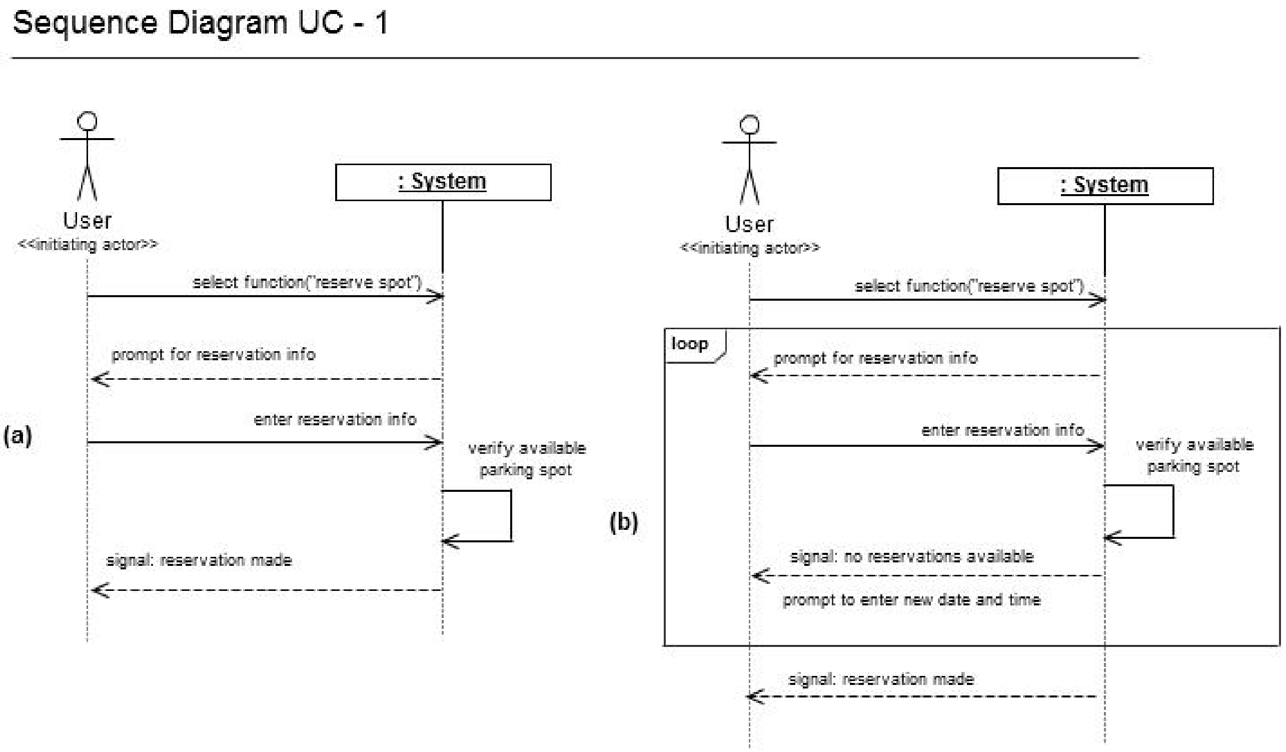


System Requirements - Use Case Traceability Matrix

Req ID	**Requirement Statement**	**Use Case**
REQ1	Elevator camera does not recognize license plate. User can be a registered customer without a reservation or a Walk-In Customer.	UC-2
REQ2	A Walk-In Customer can be registered or not registered. A registered customer needs their customer ID and password and a non registered customer needs a valid credit card.	UC-2
REQ3	Customers have until the end of their grace period before their reservation will be given away. Regardless of whether they show up or not they will be billed for the grace period. Recurring reservations may have a variable grace period, the length of which is chosen at the time of making the recurring reservation. All other types of reservations have a fixed grace period of one half-hour.	UC-2
REQ4	Missing a reservation occurs when a customer fails to arrive before the end of the grace period. A missed reservation cannot be fulfilled, and no spots are held in the garage to accommodate the reservation. If the customer arrives for a reservation late he or she will be prompted to park as a walk in.	UC-2
REQ5	If the customer does not show up for his/her reservation or misses his/her reservation then: (a) for a recurring reservation, the customer will be charged the length of the grace period at the price per hour; (b) for a confirmed reservation, the customer will not be charged a penalty for the fixed grace period of one half-hour.	
REQ6	In the scenario that the customer leaves the garage before his/her reservation expires then it is considered an understay. When this occurs the customer will still have to pay for the full reserved time. The sensor will recognize that the spot is vacant or available and will inform the system so that the spot can be assigned for another reservation.	UC-2
REQ7	The system will allow the customer to extend a reservation (only existing reservations) at least a half hour before the expiration time if and only if there is availability for the extra period desired. This procedure can be done unlimited times as long as there are available spots in the garage.	UC-4, UC-8

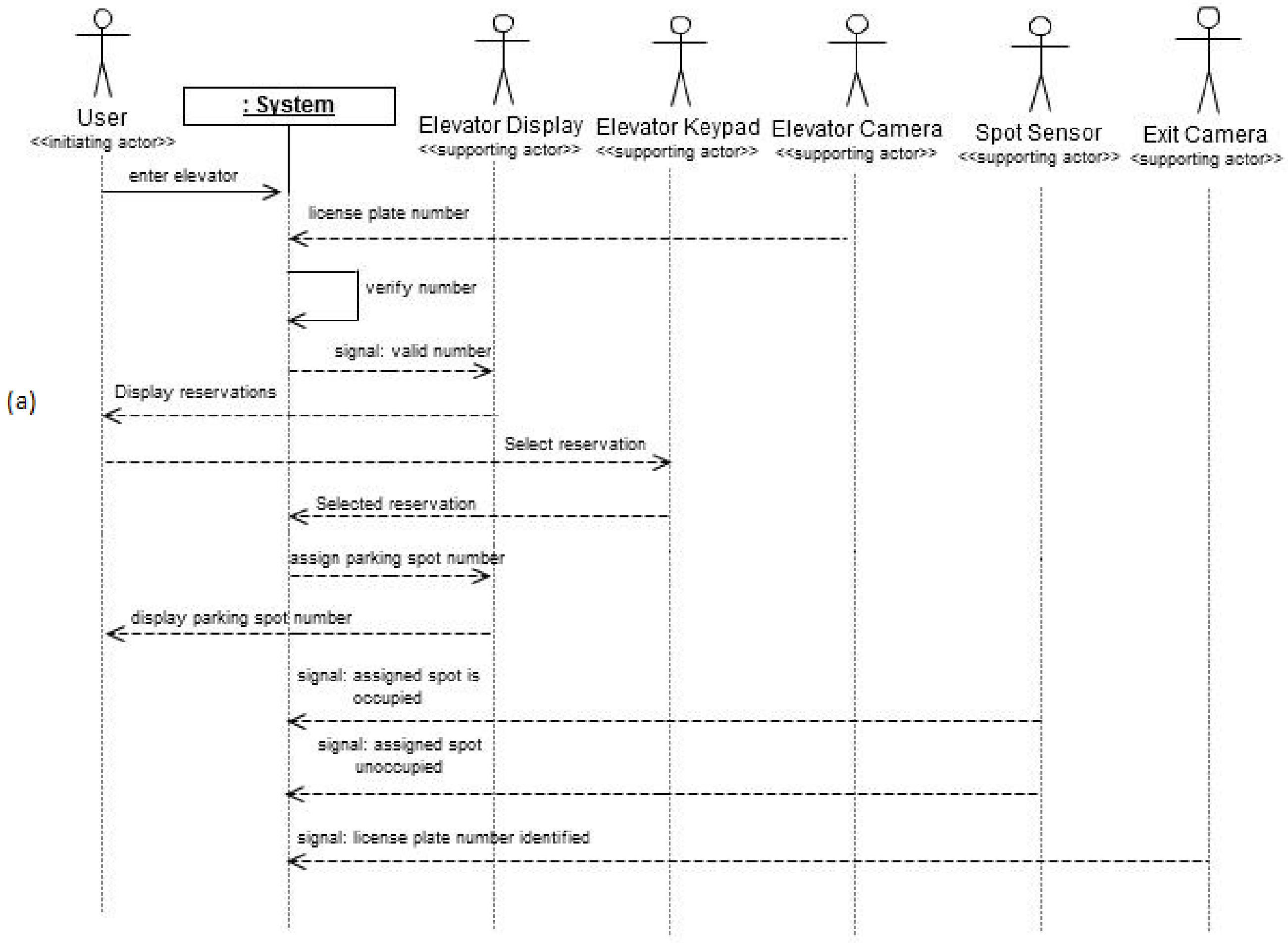
Req ID	Requirement Statement	Use Case
REQ8	In the scenario that the customer does not leave on time and stays over his/her reserved time, this is called an overstay. The customer will be billed for any overage time at 1.5 times the hourly rate, known as the penalty rate.	UC-2
REQ9	Each registered customer can have multiple unfulfilled reservations under his/her customer account. However, there must be at least a one hour gap in between consecutive reservations for the same license plate.	UC-1
REQ10	In the scenario that a registered customer arrives to the garage but it is full because other customers did not leave at the scheduled time then the system will kindly tell the customer to leave without parking and the customer will be given a rain check credit to his/her account. Unregistered customers will simply be told there is no vacancy.	UC-2
REQ11	The billing process for a registered customer will be as follows: He/she will be billed at the end of a month for charges incurred in that month, and the system will create a statement which will be emailed to the customer containing all the parking and penalty fees. The bill is automatically charged to the credit-card on file with the customer’s account.	
REQ12	If a car’s registration number is associated with a single customer then the garage will bill this customer for all the usage done under that registration number. However, if the car is associated with 2 or more registered customers then the car’s usage will be billed to the customer who made the current reservation being fulfilled.	
REQ13	Each garage will have a system administrator who can set and change prices and penalty fees for that garage.	UC-6, UC10
REQ14	A customer may cancel a reservation up to 30 minutes before the reservation start time without being billed for any of the reservation	UC-7

System Sequence Diagrams

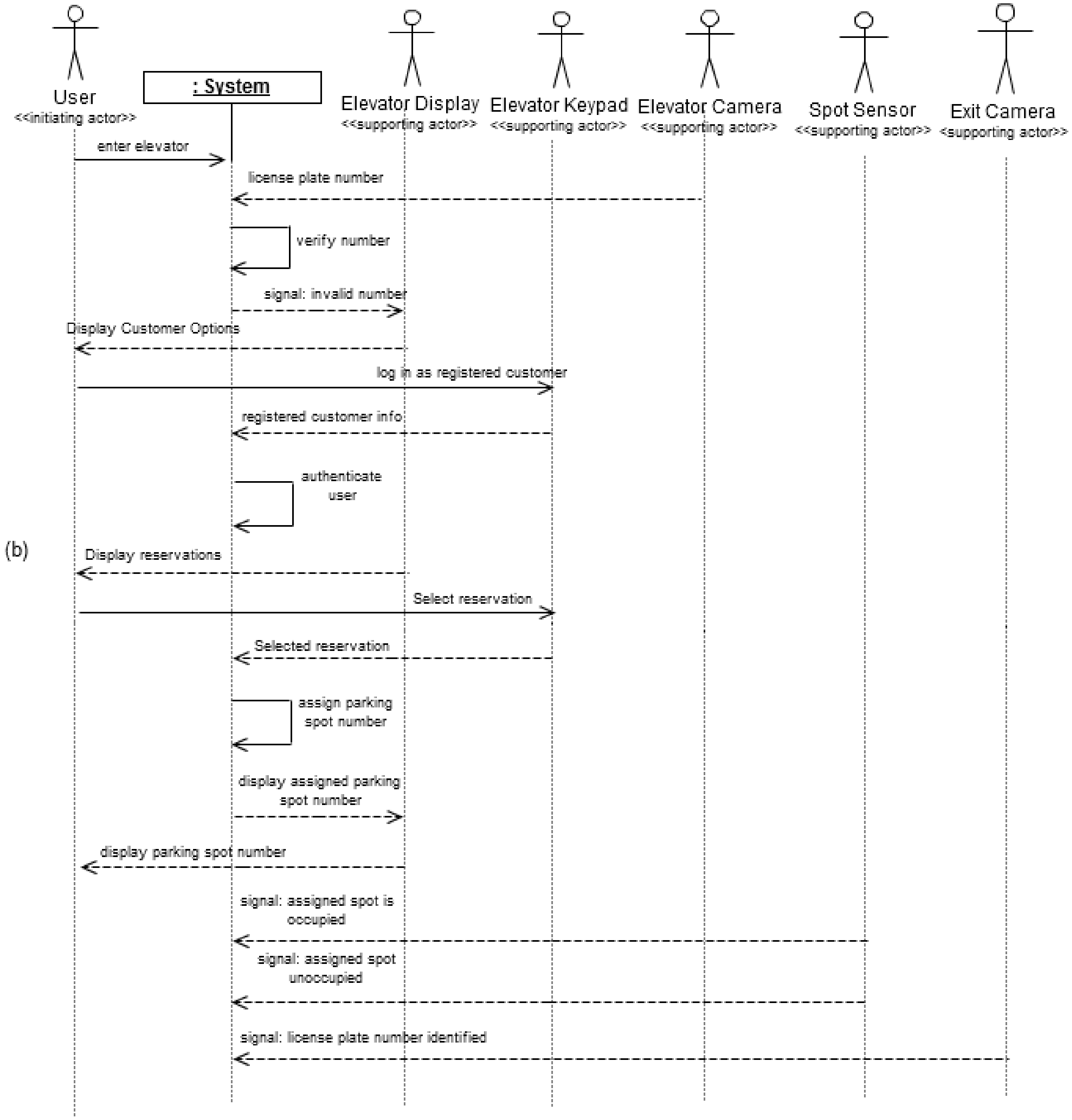

Sequence Diagram UC - 1
User
<<initiating actor>>
: System
select function("reserve spot")
prompt for reservation info
enter reservation info
verify available parking spot
signal: reservation made
(a)
loop
signal: no reservations available
prompt to enter new date and time
(b)

The System sequence diagram for UC-1 can be split up into two parts. Part (a) describes the sequence of events for the success scenario. The user requests a reservation and there are available reservations to be given out. Part b describes the sequence of events for the alternate scenario. The user requests a reservation at a specific date and time but the there are no available reservations to be given out. The user will continue to enter in a new date and time until an available reservation can be given out.

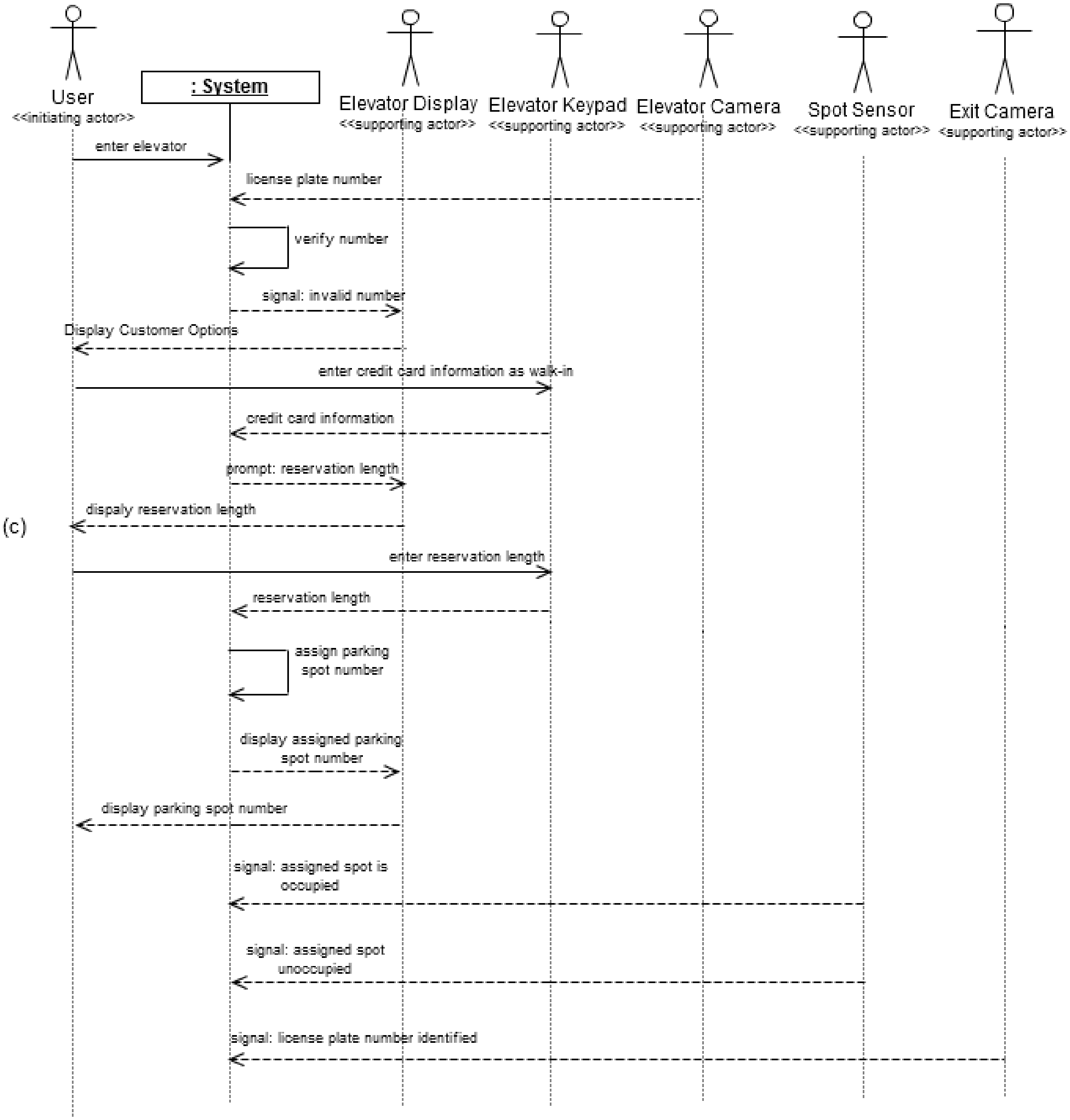
Sequence Diagram UC - 2



The system sequence diagram for UC-2 can be split into three parts. Part a describes the sequence of events for the success scenario. The customer enters the elevator, the elevator camera recognizes the the customer’s license plate number, and the system assigns the user a parking spot number.



Part b of the system sequence diagram for UC-2 describes the sequence of events for an alternate scenario. The customer enters the elevator but the elevator camera does not recognize the customer’s license plate number. The customer then signs in as a registered customer and selects a reservation. The system then assigns the user a parking spot number.



(c)

Part c of the system sequence diagram for UC-2 describes the sequence of events for another alternate scenario. The customer enters the elevator but the elevator camera does not recognize the customer’s license plate number. The customer then swipes their credit card and selects a reservation as a walk-in customer. The system then assigns the user a parking spot number.

Non-Functional Requirements

Fault-tolerance

- Park-A-Lot should remember the details of a user's interaction if the user interface should disconnect from the system.
- Park-A-Lot should quickly recover from a malfunction when a customer is inside the elevator.

Usability

- The interface presented to the customer should be as clean and simple as possible, while maintaining all important functionality.

Reliability

- Park-A-Lot should function correctly even if a customer inputs invalid entries into a reservation request form.
- Park-A-Lot should not lose a reservation through the use of persistent storage and regular backup.

Performance

- The Park-A-Lot elevator display should always display the correct output to the customer.
- The Park-A-Lot system should minimize connection times to the database and provide a quick and painless experience to the customer.
- Initially, Park-A-Lot can support at least 100 customers and 1,000 reservations. Over time should seek to increase these numbers ten-fold or more.

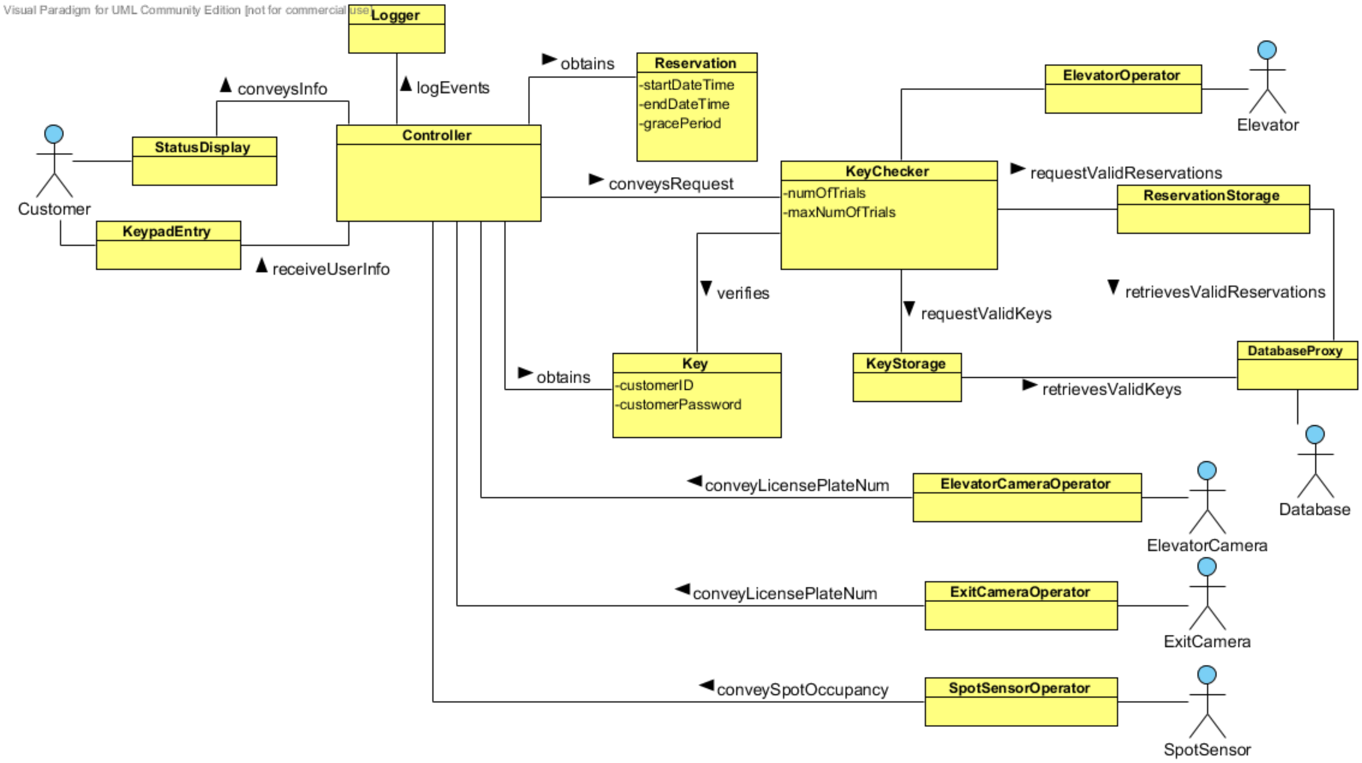
Security

- Other customers or unauthorized users should not have access to or be able to edit a customer's account details or reservations.

Domain Analysis

Domain Model

Domain model for UC-2



Reasons for model selection:

- **Cohesion** - The domain elements contain enough information to be completely independent objects (no overlap in knowledge between objects) and represent concrete ideas within the system. Therefore, our model has high cohesion since each object has several responsibilities, but does not attempt to do too much work.
- **Coupling** - The coupling of objects in our diagram is low, mainly because we have separated the key checking ability out from the Controller object. Key checking is a main concern in this model, and therefore deserves it's own object. The smaller objects surrounding KeyChecker help it accomplish its job. There is a high degree of coupling between Controller and many of the physical objects in the parking garage, but that is impossible to avoid since the controller needs to be in communication with all of the cameras and sensors in order to instruct other objects when to complete their tasks.
- **Expert Doer Principle** - The model satisfies this principle because it divides the task of checking keys and processing information about those checked keys into two distinct objects. The KeyChecker is the expert on checking customer authentication information, and informs the Controller of customer authenticity. The Controller is then able to quickly display information about Reservations to the customer.

(D - doing; K - knowing; N - neither)

Responsibility Description	Type	Concept Name
Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller
Shows the actor the current context, what actions can be done, and the outcomes of the previous actions.	N	StatusDisplay
Container for the customer ID and password that the user entered in.	N	KeypadEntry
Container for user’s authentication data, including Customer ID and password.	K	Key
Verify whether or not the key-code entered by the user is valid.	D	KeyChecker
Container for the collection of valid keys associated with the users.	K	KeyStorage
Container for user’s existing reservations.	K	Reservation
Container for the collection of reservations associated with each user.	D	ReservationStorage
Operate the elevator camera to identify a car’s license plate number in the elevator platform.	D	ElevatorCameraOperator
Operate the elevator to move to the correct floor and open the entrance door.	D	ElevatorOperator
Operate the spot sensor to determine if a car is parked in the parking spot.	D	SpotSensorOperator
Operate the exit camera to identify the car’s license plate number that is exiting the garage.	D	ExitCameraOperator
Log all interactions with the system in persistent storage.	D	Logger

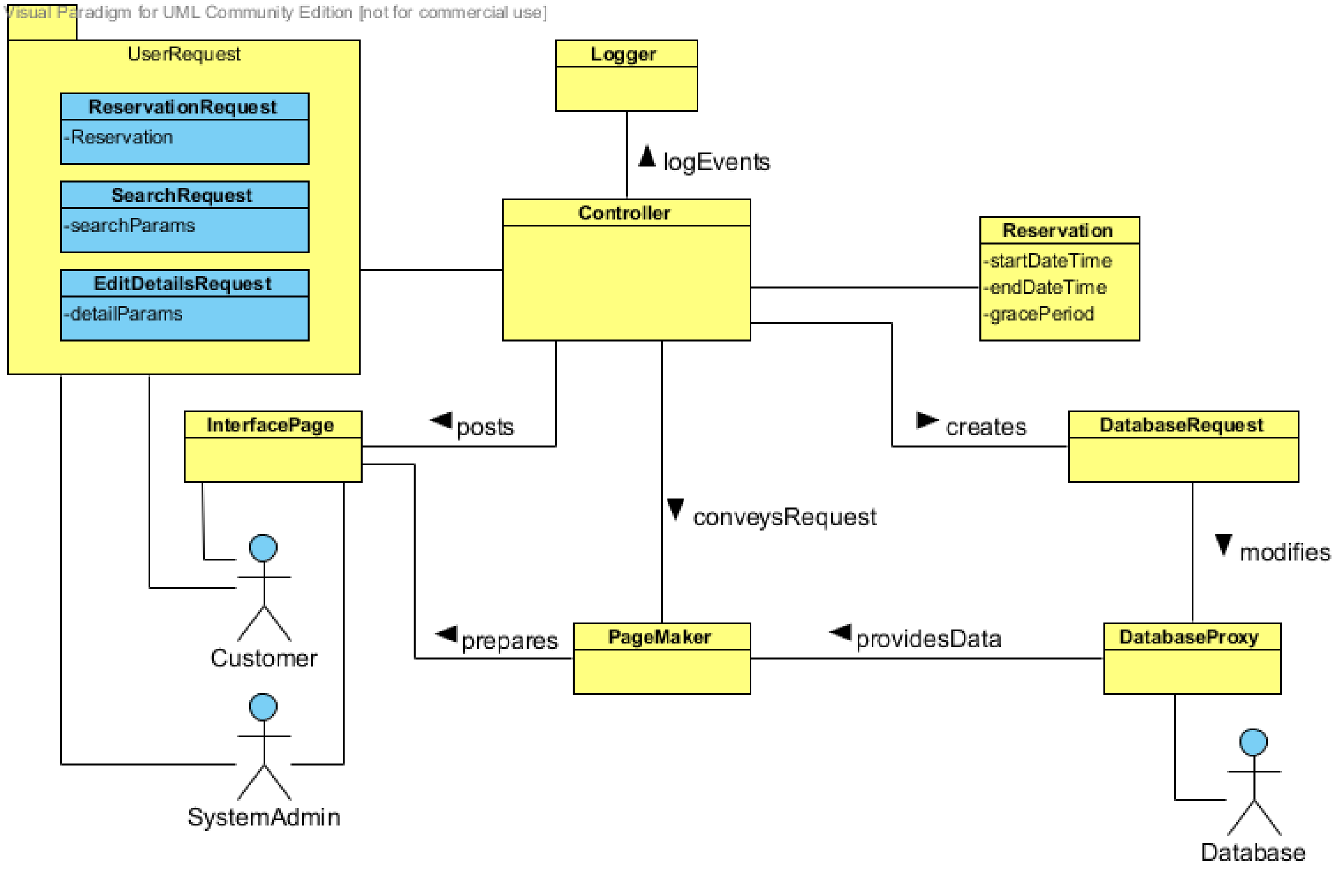
Concept Pair	Association Description	Association Name
Controller ↔ StatusDisplay	Controller passes information concerning the current context, what actions can be done, and the outcomes of the previous actions	conveysInfo

Concept Pair	Association Description	Association Name
Controller ↔ KeypadEntry	Controller receives user input information from KeypadEntry.	receiveUserInfo
Controller ↔ Logger	Controller logs information to persistent storage about system interactions.	logEvents
Controller ↔ Reservation	Controller obtains reservation information from Reservation container.	obtains
Controller ↔ KeyChecker	Controller requests KeyChecker validate customer ID/password input and receives any reservations that ID is linked to.	conveysRequest
Controller ↔ Key	Controller obtains verified user information from Key container.	obtains
ElevatorCameraOperator ↔ Controller	ElevatorCameraOperator conveys available license plate number to Controller when ElevatorCamera detects a car in the Elevator.	conveyLicensePlateNum
ExitCamera Operator ↔ Controller	ExitCameraOperator conveys available license plate number to Controller when ExitCamera detects a car leaving the parking garage.	conveyLicensePlateNum
SpotSensor Operator ↔ Controller	SpotSensorOperator conveys true/false if a spot is filled to Controller.	conveySpotOccupancy
KeyChecker ↔ Key	KeyChecker verifies if Key matches given user account information.	verifies
KeyChecker ↔ KeyStorage	KeyChecker requests a list of valid keys from KeyStorage container.	requestValidKeys
KeyChecker ↔ Reservation Storage	KeyChecker retrieves a list if reservations from ReservationStorage container.	requstValid Reservations
KeyChecker ↔ Elevator Operator	KeyChecker tells the ElevatorOperating what floor to go to.	signalOperateElevator
Reservation Storage ↔ DatabaseProxy	ReservationStorage queries DatabaseProxy for valid reservations.	retrievesValid Reservations
KeyStorage ↔ DatabaseProxy	KeyStorage queries DatabaseProxy for valid keys.	retrievesValidKeys

Concept	Attributes	Attribute Description
Reservation	Start Date Time	Start time for the reservation of a particular actor.
	End Date Time	End time for the reservation of a particular actor.
	Grace Period	Grace period for the reservation of a particular actor.
Key	Customer’s ID	ID number of customer.
	Customer’s Password	Password for the customer.
Key Checker	Number of Trials	Counter to track how many times the user has unsuccessfully entered in their key.
	Max Number of Trials	Maximum allowable times a user can enter in their key unsuccessfully before they are asked to leave.

Domain Model for Remaining Use Cases

Description of Domain Model



Reasons for model selection:

- **Cohesion** - The responsibilities assigned to each object in this model are not as great as they first appear, since the UserRequest object is actually a collection of three distinct objects which are used depending upon customer or system administrator input. The ReservationRequest is used in the case that the customer seeks to create a reservation, the SearchRequest in the case the customer or system administrator is searching for a set of reservations in the database, and the EditDetailsRequest in the case where a customer needs to edit some info in his or her account, or a system administrator needs to edit the details of an account. This gives each object a clearly defined set of responsibilities. The remaining objects are simply there to create and display the data fetched from the database, and each have a unique function, thus promoting high cohesion.
- **Coupling** - While there are many responsibilities assigned to each object, no object has an excessive amount of work to do. Each object has a distinct task to perform, and the Controller oversees these tasks and coordinates their efforts. The coupling is lower in

this domain model than in the previous one, since we have reduced the interactions between Controller and all other objects, while still maintaining a distinct set of tasks for each object.

- **Expert Doer Principle** - As in the previous domain model, this model also conforms to the expert doer principle since all of its objects that know information are the objects performing the tasks. Controller passes any information obtained from the user input to a DatabaseRequest that performs actions through the DatabaseProxy on the database. Also, the PageMaker creates the page using information from the DatabaseProxy which gets any information needed from the database using the DatabaseRequest passed to it. The Controller orchestrates this entire exchange. Therefore, each object that knows the information passes it to the correct object to do something with it.

(D - doing; K - knowing; N - neither)

Responsibility Description	**Type**	**Concept Name**
Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller
Shows the actor the current context, what actions can be done, and outcomes of the previous actions.	K	InterfacePage
Container for three distinct request types: a reservation request containing reservation data, a search request containing search parameters, and an edit details request containing changes to customer account details.	K	UserRequest
Render retrieved records into an HTML document.	K	PageMaker
Container for user’s requested reservations.	K	Reservation
Prepare a database query that best matches the actor’s search criteria and retrieve the records from the database.	D	DatabaseProxy
Form specifying details that need to be changed in the database.	K	DatabaseRequest
Log all interactions with the system in persistent storage.	D	Logger

Concept Pair	**Association Description**	**Association Name**
Controller ↔ InterfacePage	Controller posts the InterfacePage with the help of PageMaker.	posts

Concept Pair	Association Description	Association Name
Controller ↔ UserRequest	Controller obtains user information from UserRequest and forms a DatabaseRequest to act on that information.	receiveUserInfo
Controller ↔ Logger	Controller logs information to persistent storage about system interactions.	logEvents
Controller ↔ Reservation	Controller obtains reservation information from Reservation container.	obtains
Controller ↔ PageMaker	Controller conveys a request to PageMaker to prepare an HTML page that it will display to the customer through InterfacePage.	conveysRequest
Controller ↔ Database Request	Controller creates a DatabaseRequest from the UserRequest to specify which actions should be performed on the Database.	creates
PageMaker ↔ InterfacePage	Page Maker prepares the Interface Page.	prepares
DatabaseRequest ↔ DatabaseProxy	DatabaseRequest modifies the Database through DatabaseProxy to insert, delete, modify, or retrieve records in the Database.	modifies

Concept	Attributes	Attribute Description
Reservation	Start Date Time	Start time for the reservation of a particular actor.
	End Date Time	End time for the reservation of a particular actor.
	Grace Period	Grace period for the reservation of a particular actor.
Search Request	Search Parameters	Start Date, end date, customer ID.
Reservation Request	Reservation (s)	Date, start time, end time, type of reservation.
Edit Details Request	Detail Parameters	Customer ID, password, email address, name, address, zip code, state, phone number, credit card number, license plate number(s).

System Operation Contracts

Operation	Park
Preconditions	● parkingSpot = “unoccupied” , The parking spot will be occupied as soon as the customer exits the elevator.
Postconditions	● parkingSpot = “unoccupied”

Operation	Authenticate User
Preconditions	● Set of valid customer IDs is known to the system and is not empty ● numOfTrials < maxNumOfTrials ● numOfTrials = 0, for the first trial of the current user
Postconditions	● numOfTrials = 0, if the entered Customer ID contained in the set of valid keys.

Mathematical Models

There are no mathematical models to discuss.

User Interface Design

The system is very user interactive, so the user is presented with different reports and forms to both view and input data, as well as make different requests. The forms that a user may encounter will depend on the role that the user plays in the system (guest vs member vs admin). The possible forms that a user may encounter while interacting with the system are:

- Search Nearby Garages
- Registration
- Sign In
- Reserve Parking
- Edit Reservation
- Cancel Reservation
- Edit Account
- Set Prices (admin)
- View Garage Usage History (admin)

View Descriptions

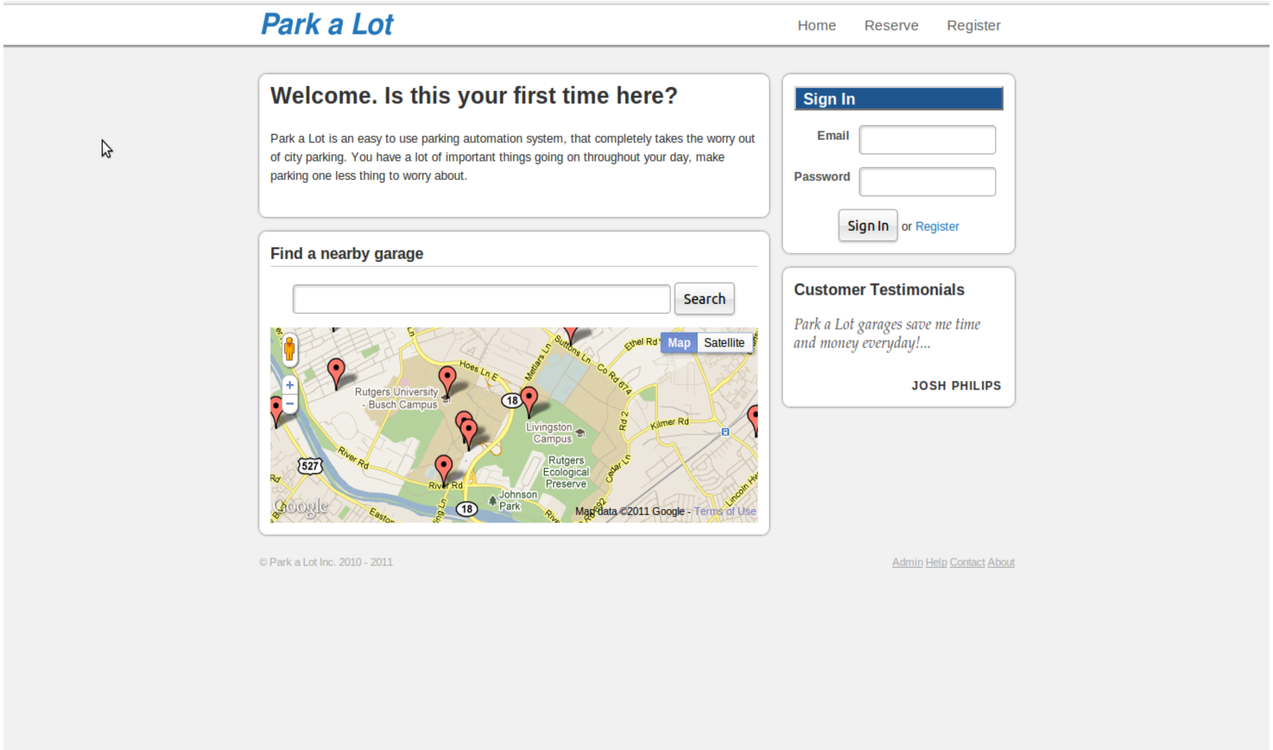
Below are different views within the system, along with a short description of the view. Some views have a sample screen shot as well.

Template

The template for the website is made of a consistent header, body, and consistent footer. The header contains the logo and the main navigation area, which has options to reserve parking, register for membership, and return to the home page. The footer contains copyright information as well as links to the help page, contact us page, and the about us page.

Home Page

The home page presents new users with a welcome message. There is a section for all users to search nearby garages which have implemented our system by entering a zip code. There is a section for registered customers to sign in directly by entering their email and password. There is a section where customer testimonials will be shown.



Welcome. Is this your first time here?

Park a Lot is an easy to use parking automation system, that completely takes the worry out of city parking. You have a lot of important things going on throughout your day, make parking one less thing to worry about.

Find a nearby garage

Customer Testimonials

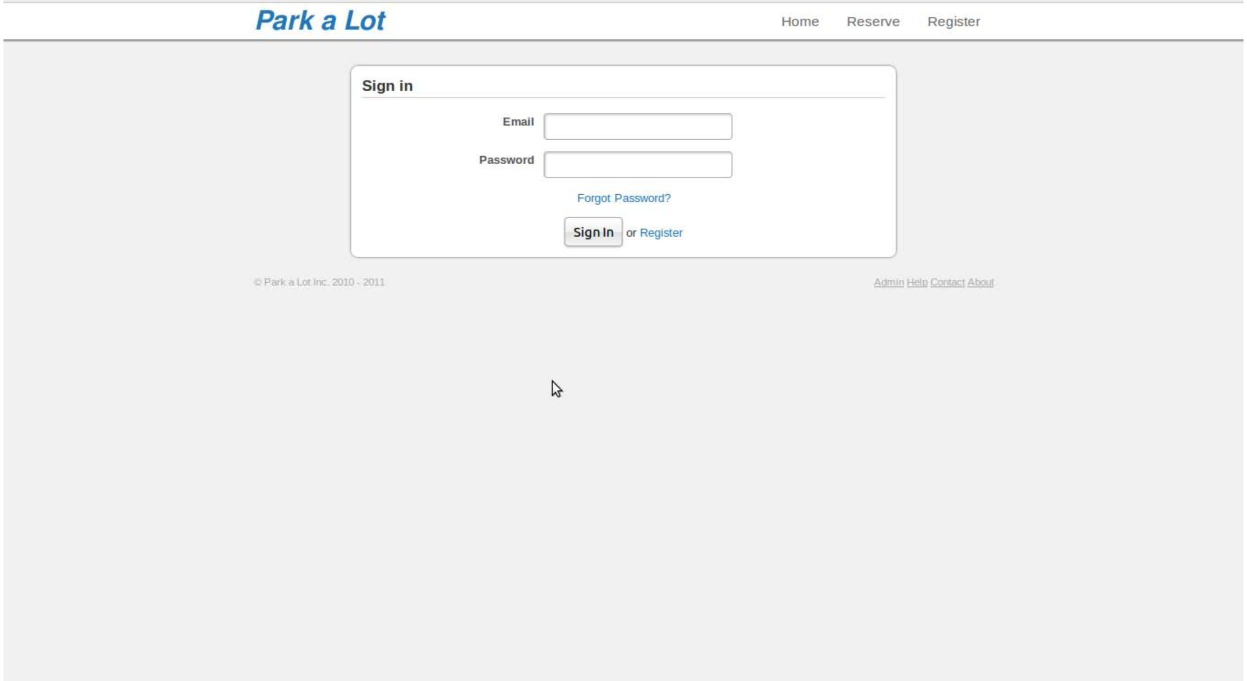
Park a Lot garages save me time and money everyday!...

JOSH PHILIPS

© Park a Lot Inc. 2010 - 2011

Sign In

The sign in page presents a form in which a user enters their email and password. The user can view this page after a failed attempt at signing in from the home page.

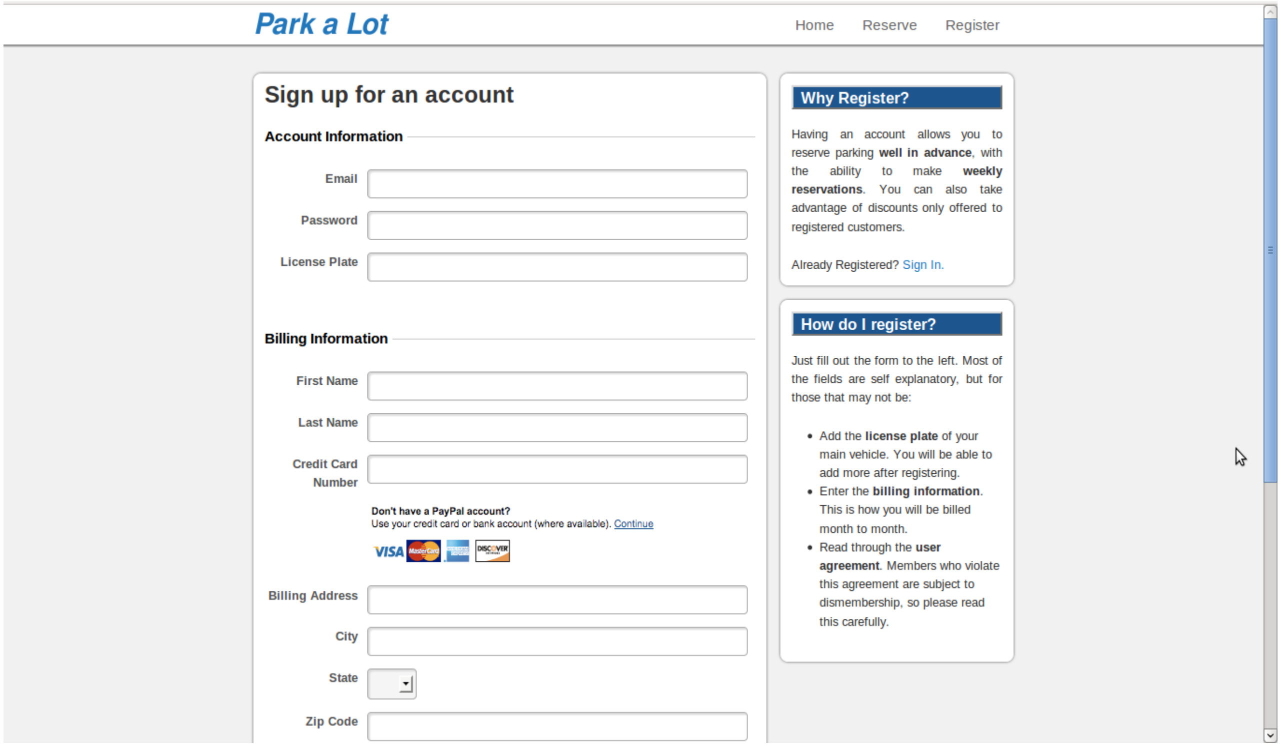


Sign in

© Park a Lot Inc. 2010 - 2011

Registration

The registration page presents a form in which a user enters their email, desired password, license plate number of their main vehicle, first and last name, credit card number, billing address, city, state, zip code, and home phone number. The user can view this page after clicking “Reserve” in the main navigation section.



Sign up for an account

Account Information

Email

Password

License Plate

Billing Information

First Name

Last Name

Credit Card Number

Don't have a PayPal account?
Use your credit card or bank account (where available). Continue

Billing Address

City

State

Zip Code

Why Register?

Having an account allows you to reserve parking **well in advance**, with the ability to make **weekly reservations**. You can also take advantage of discounts only offered to registered customers.

Already Registered? Sign In.

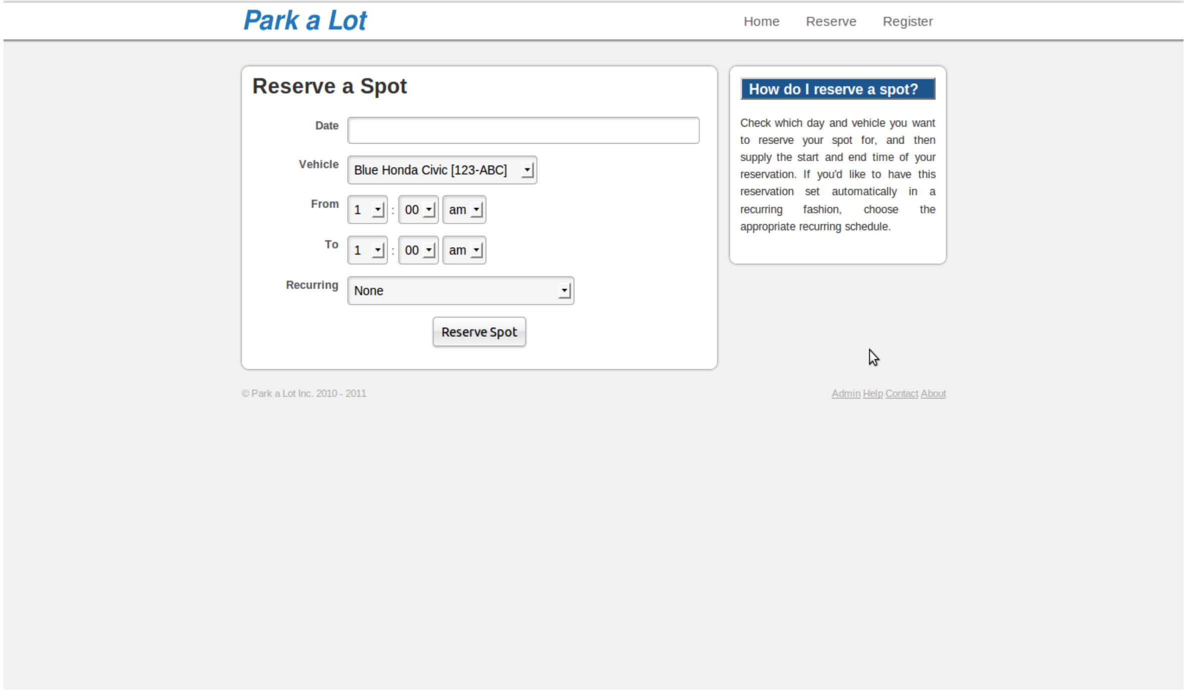
How do I register?

Just fill out the form to the left. Most of the fields are self explanatory, but for those that may not be:

- Add the **license plate** of your main vehicle. You will be able to add more after registering.
- Enter the **billing information**. This is how you will be billed month to month.
- Read through the **user agreement**. Members who violate this agreement are subject to dismembership, so please read this carefully.

Reserve Parking

The reserve parking page presents a form that the user fills out to reserve parking. The user enters their desired date, start time, end time, the vehicle they wish to reserve time for, and whether or not the reservation should occur more than once. The user can view this page after clicking “Reserve” in the main navigation section.



Reserve a Spot

Date

Vehicle: Blue Honda Civic [123-ABC]

From: 1 : 00 am

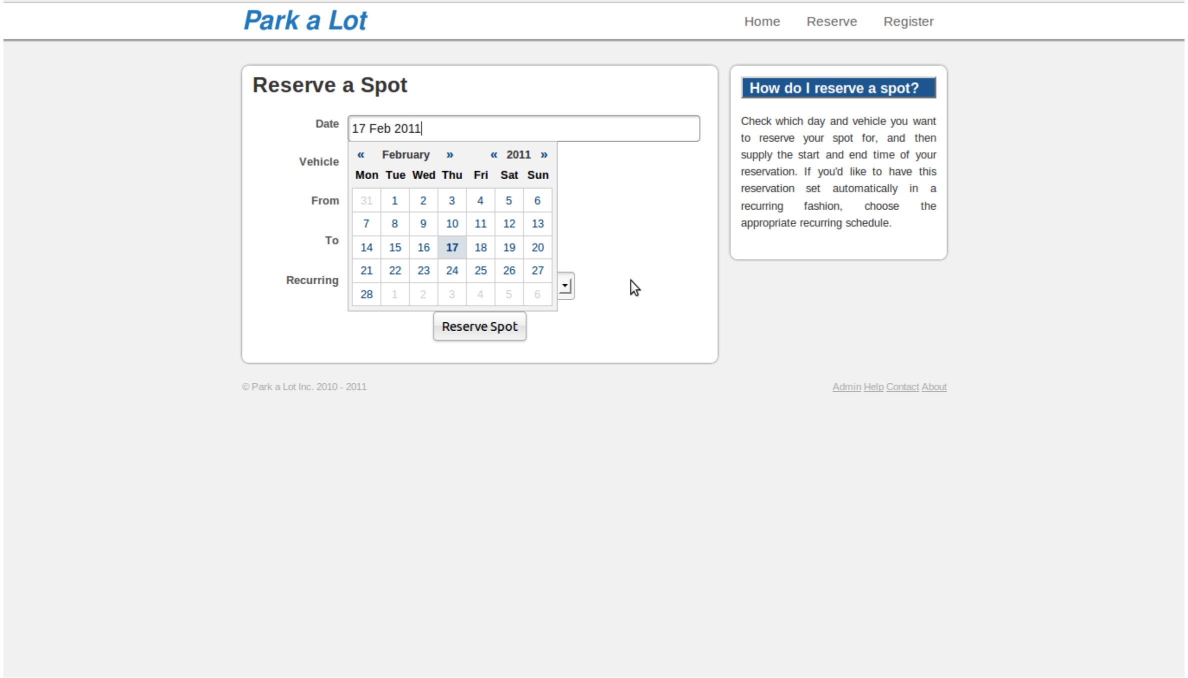
To: 1 : 00 am

Recurring: None

How do I reserve a spot?

Check which day and vehicle you want to reserve your spot for, and then supply the start and end time of your reservation. If you'd like to have this reservation set automatically in a recurring fashion, choose the appropriate recurring schedule.

© Park a Lot Inc. 2010 - 2011



Reserve a Spot

Date: 17 Feb 2011

« February » « 2011 »

Mon	Tue	Wed	Thu	Fri	Sat	Sun
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	1	2	3	4	5	6

Vehicle

From

To

Recurring

How do I reserve a spot?

Check which day and vehicle you want to reserve your spot for, and then supply the start and end time of your reservation. If you'd like to have this reservation set automatically in a recurring fashion, choose the appropriate recurring schedule.

© Park a Lot Inc. 2010 - 2011

User Profile

The user profile page displays information pertaining to each user. The user can view their upcoming and past reservations, as well as their usage history and current balance. From this page, the user can choose whether to edit/cancel future reservations. The user can view this page right after logging in.

Edit Reservation

The edit reservation page displays a pre-filled form containing the existing reservation data, and allows the user to edit any aspect of the form. The user can view this page after first selecting an existing reservation on their home page.

Cancel Reservation

The cancel reservation page displays information about the chosen reservation, as well as a confirmation button asking the user whether they are sure they really want to cancel the reservation. The user can view this page after first selecting an existing reservation on their home page.

Edit Account

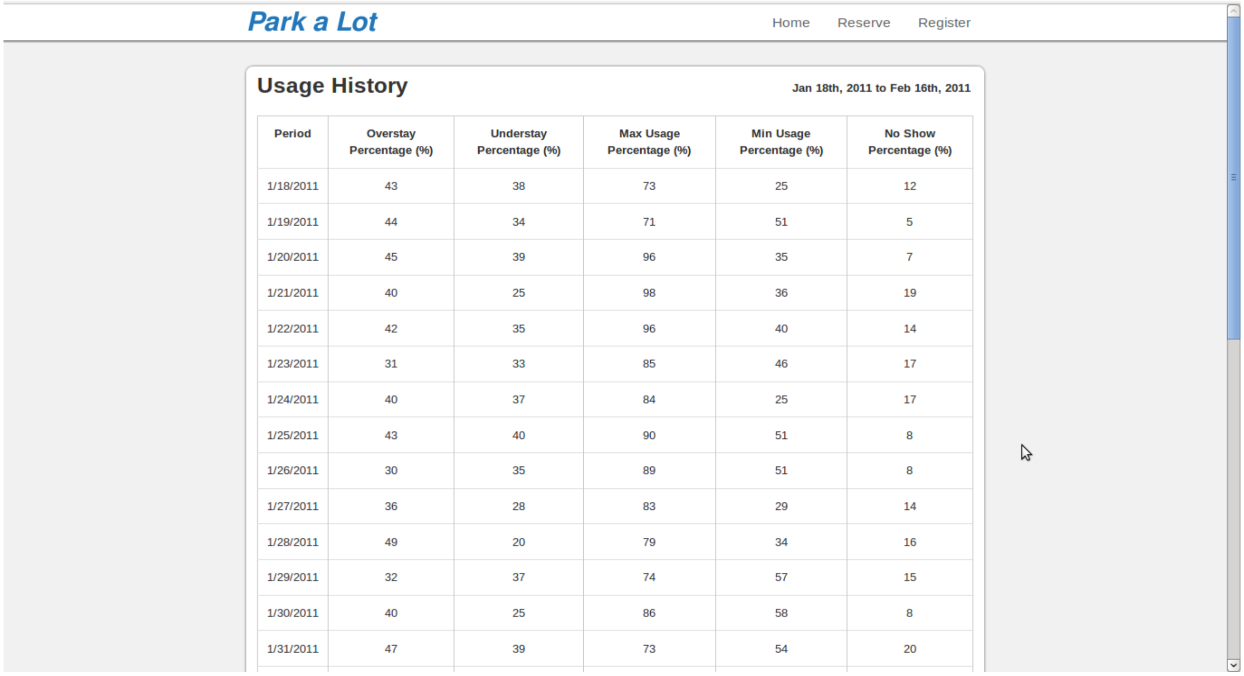
The edit account page allows a user to edit their account information (the information requested during registration). The page displays a pre-filled form containing the existing user information, and allows the user to edit any aspect of the form. The user can view this page after clicking edit account on their profile page.

Set Prices

The set prices page allows administrators to set prices for their garage. The page displays a form where a user enters price points for different types of members.

View Garage Usage History

The view usage page allows administrators to view aggregate data pertaining to their garage. The administrator can view the percentage of overstays, understays, maximum and minimum usage percentages, as well as percentage of no shows day by day.



Usage History

Jan 18th, 2011 to Feb 16th, 2011

Period	Overstay Percentage (%)	Understay Percentage (%)	Max Usage Percentage (%)	Min Usage Percentage (%)	No Show Percentage (%)
1/18/2011	43	38	73	25	12
1/19/2011	44	34	71	51	5
1/20/2011	45	39	96	35	7
1/21/2011	40	25	98	36	19
1/22/2011	42	35	96	40	14
1/23/2011	31	33	85	46	17
1/24/2011	40	37	84	25	17
1/25/2011	43	40	90	51	8
1/26/2011	30	35	89	51	8
1/27/2011	36	28	83	29	14
1/28/2011	49	20	79	34	16
1/29/2011	32	37	74	57	15
1/30/2011	40	25	86	58	8
1/31/2011	47	39	73	54	20

User Effort Estimation

Below are a few scenarios highlighting the effort needed to accomplish certain tasks.

Registration

Registering a new user for a membership. Assume the new user is on the home page.

1. Navigation: 1 click as follows
    a. Click “Register” link in the main navigation section.
2. Data Entry: 2 mouse clicks, 11 input fields, 10 additional keystrokes
    a. Focus cursor on email text field.
    b. Enter email address of user
    c. Press Tab key to move to the next field (password)
    d. Enter desired password
    e. Press Tab key to move to the next field (license plate)
    f. Enter license plate number of main vehicle
    g. Press Tab key to move to the next field (first name)
    h. Enter the first name of user
    i. Press Tab key to move to the next field (last name)
    j. Enter last name of user
    k. Press Tab key to move to the next field (credit card number)
    l. Enter credit card number
    m. Press Tab key to move to the next field (billing address)
    n. Enter billing address of user
    o. Press Tab key to move to the next field (city)
    p. Enter city of user
    q. Press Tab key to move to the next field (state)
    r. Select state of user
    s. Press Tab key to move to the next field (zip code)
    t. Enter zip code of user
    u. Press Tab key to move to the next field (home phone number)
    v. Enter home phone number of user
    w. Click “Sign Up” button at the bottom.

Sign In

Logging in an existing member and establishing a session. Assume the new user is on the home page.

1. Navigation: 0 clicks
2. Data Entry: 2 mouse clicks, 2 input fields, 1 additional keystroke
    a. Focus cursor on email text field.
    b. Enter email address of user
    a. Press Tab key to move to the next field (password)
    b. Enter password of user
    c. Click “Sign In” button at the bottom.

Reserve Parking

Logging in an existing member and establishing a session. Assume the new user is on the home page.

1. Navigation: 1 click as follows
   a. Click “Reserve” link in the main navigation section.
2. Data Entry: 2 mouse clicks, 9 input fields, 8 additional keystrokes
   a. Focus cursor on date text field.
   b. Enter date of desired reservation
   c. Press Tab key to move to the next field (vehicle)
   a. Select vehicle to set reservation for
   b. Press Tab key to move to the next field (start time hour)
   c. Select start time hour of the reservation
   d. Press Tab key to move to the next field (start time minute)
   e. Select start time minute of the reservation
   f. Press Tab key to move to the next field (start time meridian)
   g. Select start time meridian (am/pm) of the reservation
   h. Press Tab key to move to the next field (end time hour)
   i. Select end time hour of the reservation
   j. Press Tab key to move to the next field (end time minute)
   k. Select end time minute of the reservation
   l. Press Tab key to move to the next field (end time meridian)
   m. Select end time meridian of the reservation
   n. Press Tab key to move to the next field (recurrence)
   o. Select recurrence of the reservation
   p. Click “Reserve” button at the bottom.

Note: The user can choose to select their start date from the popup calendar, and can also choose to leave the default recurrence value to none if not a recurring reservation.

Plan of Work

The following plan of work lists milestones which we hope to accomplish as deliverables for this project.

Step 1: Class Diagrams and System Design

The first step we must take is to create all the class diagrams in which we will show all the classes and their respective associations. The class diagrams are critical because they help you see the interaction between the devices and the actors and give us an outline to follow while we are coding each class. The system design is one of the most important parts of actually developing the software since our system design will ultimately determine the way our software will run and how we will build it. As a group, we will all collaborate to create all the class diagrams necessary to meet all the business requirements and user cases described in this report. The main outline of the system design is to be able to take all the information (from the users as well as the system administrator) and to operate as efficiently as possible. Our system design will be done within the next two weeks, which it means that by March 4th, we should have all the class diagrams done and described in very detail as well as our system design that will allow us to start coding.

Step 2: Database Design

One of the main components of the system design is the database design. Most of the coding and creating the software depend on how the database will work and its design. For "Park-a-Lot" we will have two separate databases, one customer database and a garage database. Since we want to be able to use this software in many stations (different "Park-a-Lot"s) we need to implement a garage database that will store the information of the different garages and another one that will take care of all the different customers thought the different garages. By doing it so, the customers will be allow to reserve in different "Park-a-Lot"s without having to register multiple times. The database will store all the customer information as well as the billing information (all the storage and penalty fees). As a group we are planning to have this design done by March 4th.

Step 3: Software Design and Implementation, Coding

After having the class diagrams and our system as well as database design, we will proceed to start implementing the software. Based on our system design we will proceed to start writing our code in c. Our software design will be simple, yet effective and user-friendly. We want to make the software as easy to use so that people using the web (either through their phone or computer) will be able to manage their account, reserve, etc without having to spend so much time trying to figure things out. As we get the main functions working, we will be implementing more features so that the software will be even more productive and user-friendly. This step should be completed by the fourth week of march. All 5 of us will contribute in the coding section.

Step 4: Iterative Testing and Revision

The last major step in our software project is the iterative testing and the revision. Once the software is done, we need to make sure it runs smoothly from different computer/devices. Therefore this process will take some time since every detail needs to fixed and taken care of before the final submission of the project in May 5th. The system will be undergoing testing for about a week. The actual project should be finished by late April and from the day is finished, the testing will start until every possible error is taken care of. Depending on how many different types of testing and revisions are necessary, at most two the group will be working on it.

Project Milestone	**Est. Completion Date**	**Variance**
Class Diagrams and System Design	March 4	3 days
Database Design	March 4	3 days
Software Design and Implementation	March 22	1 week
Iterative Testing and Revision	April 26	2 weeks

References

[1] “Software Engineering”, by Ivan Marsic, Rutgers University, 2010