Fort Hays State University

CSCI 441 - Software Engineering

Group 10

Network of Automated Parking Garages

Report One

September 25, 2022

Team Member:	*Email:*
Dustin Threet	dlthreet@mail.fhsu.edu
Hannah Carr	hmcarr@mail.fhsu.edu
Justin Henley	jahenley@mail.fhsu.edu
Victor Munoz	vemunoz@mail.fhsu.edu

Project Website: https://fhsu-csci-675-group-10.onrender.com

Individual Contributions Breakdown

	Dustin Threet	Hannah Carr	Justin Henley	Victor Munoz
Project Management	25%	25%	25%	25%
Section 1	25%	25%	25%	25%
Section 2	25%	25%	25%	25%
Requirements Specification	25%	25%	25%	25%
Section 3	25%	25%	25%	25%
Section 4	25%	25%	25%	25%
Section 5	25%	25%	25%	25%
Section 6	25%	25%	25%	25%
Section 7	25%	25%	25%	25%
Section 8	25%	25%	25%	25%
Graphics	25%	25%	25%	25%

Table of Contents

Work Assignment

Given that our group has 4 members, we have refrained from dividing our team “into 3 or 4 sub-teams.” We prefer instead to divide flexibly on a per-task basis with consideration for workload balance and individual competences.

Name	Competences	Work Assignment
Dustin Threet	Software Development, JavaScript, NodeJS, AngularJS, Bootstrap, HTML, CSS	Local Garage Locator Page, Walk-In Display, Reservations & Walk-Ins
Hannah Carr	Software Development, Database design, SQL, C#, Java, HTML, CSS, Python	Local Garage Locator Page, Database Diagram, Operators Adding a Garage to a Network of Garages
Justin Henley	REST APIs, JavaScript, NodeJS, ReactJS, HTML, CSS	Register Page, Sign In Page, Walk-In Display, Customer Account Registration
Victor Munoz	Business Management, C++, Java, JavaScript, HTML, CSS	Register Page, Sign-In Page, Location-based Garage Finder

1 - Customer Problem Statement

1.1 - Problem Statement

1.1.1 - Problem Diagnosis

In the USA, cars are a central part of life. They represent the freedom of the open road and the independence to travel anywhere at any time. However, wherever one drives, there must be adequate parking. American cities devote significant amounts of land to both on-and off-street parking. Hoenhe, et al. estimated in 2017 that the Phoenix, Arizona Metropolitan area offered 4.3 parking spaces per registered non-commercial vehicle. These 12.2 million parking spaces were the equivalent of 10% of the Phoenix metro's total land area at that time (Hoehne et al., 2019).

Despite all this land and all these dedicated parking spaces, finding parking in a city can still be a hassle. Garages operate on various access control systems, from older paper tickets to more modern automated systems. Pricing can be convoluted and opaque for customers, who may only discover the pricing scheme upon arrival. Finding a garage at all is only half the battle, as a driver may arrive only to find the lot already full. This inefficiency in finding parking leaves drivers wandering the city, wasting time and fuel and contributing to congestion. It also leaves garage owners with no way to advertise their available spaces and pricing to drivers in the area.

A few services have already arisen to tackle this inefficiency, however imperfectly. Parkopedia (Parkopedia) allows users to search for parking lots by location, viewing results on a map with projected pricing. However, lots do not report their availability to the site. A user is only aware of the potential to park, with no guarantee of an available parking space. Parking.com (SP Plus Corporation) likewise allows users to find garages and pricing on a map but adds the ability to reserve spaces for set periods. However, users are emailed a parking pass for their reservation, requiring them to manage their own credentials to park. Finally, SpotHero (SpotHero) builds on Parking.com's model to support garages with both manual and automatic parking control. With support for QR code-scanning automatic gates, gate attendants, and ticket-and-cashier type parking lots, SpotHero makes fewer demands on the lot owner for integration into the service. However, this increases the demands on users, who must read and understand the instructions for each different garage they use.

So far, none of the available solutions provide a truly unified parking solution for users and garage owners. Users cannot simply drive into an automated garage using their single parking website account. Garage owners cannot take advantage of the online platform's payment system to handle all transactions at the garage. The online infrastructure for managing pricing and occupancy is separated from the onsite infrastructure, requiring operators to manage data on multiple platforms. Our project seeks to exploit this gap in the market.

1.1.2 - Proposed Solution

To offer value to customers and users, our solution proposes a network of garages operating our system. Users will benefit from a single service membership that allows them to find, reserve, and park in any member lot with automatic payments.

Existing and prospective garage owners will benefit from a full software solution that reduces the need for on-site management of the lot. They will benefit from an expanded pool of registered customers directed to their lots, with a single system of payments. Users will have a streamlined parking experience with familiar hardware, automatic payments, and no parking passes to manage or passwords to remember.

Customers

For customers, a full software solution to garage control satisfies the needs of new projects and presents a straightforward upgrade solution for existing businesses. The software will be sold as a configurable solution based on the client's specific needs. In this case, the client has license plate readers at the entry to the garages and an automated car elevator to take drivers to their assigned floors. Each driver will be assigned a parking space upon entry, and the system will use space occupancy sensors to detect mistakes. Entry and exit events paired with occupancy sensors provide a simple, fault-tolerant way to track garage occupancy automatically. The pricing and number of available parking spaces are displayed on an electronic display outside the garage and mirrored online to inform and attract customers.

By allowing users to simply reserve *a* space in a chosen garage, instead of a specific space, we eliminate complex tracking of individual spot availability. Space assignments can be given at time of entry to the garage, allowing assignments to reflect the real-world state of the garage regardless of drivers who overstay or park in the wrong space. This also reduces demand on users, who are not required to be penalized for failing to find their assigned space, providing users with a lower-stress parking experience.

Out of the box, a garage management portal will be offered for viewing current occupancy, reservations, and historical data. It will allow operators to configure the specifications of their garages as well as add new garages to their network. Prices can also be set through this portal, giving operators a single system for managing both onsite and online garage displays. This simplification of management, combined with gate and reservation automation, requires less staff and reduced work hours for managing gates, counting spaces, generating reports, or handling reservations. Reduced labor needs and an optimized reservation system mean cost savings for garage owners while maximizing profit capabilities. Historical data and statistics provide a useful metric for our clients, allowing them to configure their reservation settings to optimize garage usage based on no-show and cancellation rates. Such data and statistics also allow clients to compare revenues and occupancy rates before and after the conversion to our system, making it easy to measure the return on investment.

The solution is scaled to a single system per client, each of which may own or operate many parking garages in many locations. The solution is flexible, allowing it to be deployed on-premises or in any cloud system. The customer may choose to buy only the software, requiring them to make all configuration changes to the software and deploy and manage upgrades themselves or through a third-party vendor. Alternatively, customers may be given an option to pay for our managed services. Our team would then continue to manage deployments, upgrades, and necessary configuration changes to match the design and structure of their garage network. This allows the client to pick a solution that matches their individual needs based on their IT capabilities and workload.

Users

For users, a single website provides streamlined access to every garage in the network. A user can reserve spaces and pay for parking, with an option to bind a license plate for automatic recognition. Users can use their device location or enter a destination address to see a map of available parking garages nearby. Each location will display the number of vacant spots and pricing scheme, with an option for the user to make a reservation for an available lot. After making a reservation with their license plate number, users simply drive up to the garage gate, where the license plate reader recognizes their car, and park in any open space. As membership is required to park in the garage, and members are billed monthly, users exit the garage by simply driving to the exit gate and waiting for the license plate reader to recognize their vehicle.

The combination of easy online discovery of garages with automated garage systems saves users time and money. They do not have to drive around the city looking for open garages. They also do not have to worry about arriving at an open garage only to discover the lot charges exorbitant rates. On arrival, there is no token to hold, no QR code to scan, and no map or space number to memorize. Locate, reserve, drive in, drive out.

1.1.3 - Novelty

The garage system acknowledges the efforts of past teams to solve this problem, incorporating their most appealing innovations while reaching further for new features. The Fall 2018 project distinguished itself with a ‘smart’ pricing system, passcode-secured pedestrian entry, and an interface for ad-hoc users to select their own parking space on arrival. Our project incorporates their choice to reduce congestion within the garage by allowing only a single active vehicle per floor.

The Spring 2019 project introduced allowing users and administrators to search and filter data on vehicles parked in the lot. It allowed first-time users to park without an account and sign up online afterward. We are most interested in their decision to restrict drivers to only registering for an area of the garage, not a specific space. Our project expands on this idea by assigning parking spaces on arrival as available, with users only reserving the right to park somewhere in the chosen lot. This reduces the amount of input necessary to reserve a parking space and allows for tolerance of misparked users.

The Fall 2019 project took a decidedly hi-tech approach to garage management. A blockchain-based payment system brought data encryption and decentralization. Docker-based containerization allowed for portability and real-time data exchange between garages, while Electronic Toll Collection (ETC) tags handled payment. A license plate blacklist alerted managers to users who should be turned away or turned in to law enforcement. However, we see limited utility to the additional layers of blockchain and Docker data containerization compared to traditional solutions. ETC tags also add hardware and software complexity to the system for limited utility, as the system already identifies users by license plate and charges them automatically.

This project innovates by zooming out and setting the focus on the wider world of city parking. Rather than focus on providing service for managers of a single garage, we help owners of multiple garages to create a network of locations, managed through a single portal. Customers benefit from simplified management of all their locations, and automatic data collection and statistics for each. Users benefit from a single account, simple reservations, and automatic payments for the entire network of garages managed by the customer. The user experience is simplified by allowing users to make reservations by garage, with spaces assigned only on arrival. The choice to remove space selection at reservation time reduces user input and demands on attention, further reduced by introducing a fault-tolerant system that tracks the actual space the user parked in, rather than assuming that users will never make mistakes.

1.2 - Decomposition Into Sub-Problems

1. **Time spent looking for parking**: Our program aims to drastically reduce the time a driver spends looking for parking. This is especially important in large metropolitan cities where finding street parking near your destination is a toss-up, at best. Users will be able to pull up a list of available garages and select one at their convenience.

2. **Knowing that there are vacancies prior to arrival**: Another hurdle is the surprise of arriving at a garage only to realize that there are no vacancies. When a user

searches for garages near their location, they are only presented with a list of garages with vacancies so as to eliminate the guessing game.

3. **Adjusting to various access control methods**: Different garages by different owners of franchises have individualized access control measures. This means that the user is constantly having to adjust to different payment methods. For example, most garages print a ticket used to measure the time your vehicle spent in the garage. Some of these garages will allow users to pay at the exit booth from their vehicles, others require that the users pay at a kiosk prior to reaching their vehicle and exiting.

4. **Ease of use on the go (number of clicks when making reservations)**: When searching for parking, most people default to searching for garages using applications like google maps or searching the web. The difficulty here is that once they choose one, they have to navigate to their website to get additional information on access and payment methods. Alternatively, they can look for garages through different services, mentioned in the customer problem statement, but they still run into the issue of having to manage their own reservation and click through various pages and links.

5. **Customer's ability to quickly and seamlessly adjust pricing, track data, and add garages to their network**: Owners or operators of multiple garages may be faced with managing garage data and payment on multiple, different, systems. Data tracking across multiple systems to estimate and understand parking trends make it difficult to manage overbooking in an effective and predictable way

6. **Vertically integrated payment system for owner/operators**: As stated in the subproblem above, managing multiple systems of information, including payment software can make it cumbersome to track and manage payment information and history. This is especially true when different garages accept different forms of payment as compared to others.

7. **Automation of services**: Overhead costs are inherent in every business and reducing these costs are critical in increasing profits. Investment in technology that streamline processes to increase efficiency are major contributors to a business' ability to increase equity.

1.3 - Glossary of Terms

Confirmed Reservation: Represent registered customers who make reservations as the need arises.

Guaranteed Reservation: Represent registered customers who made a *contract* with the parking garage for a parking spot, such as those going to work who need to park on a daily basis during a predetermined period.

Overstays: Currently parked customers who wish to extend their stay beyond the time of their original reservation.

Understays: Currently parked customers who arrive on time but decide to leave before their predicted time of departure.

Walk-ins: Customers who arrive at the parking garage without a contract of reservation.

No-show: Customers who have a reservation but do not arrive for that reservation within the grace period of the reservation start time.

Occupancy State: Refers to the state of parking spots which can be either “available”, “reserved”, or “occupied”.

License Plate Reader: A camera device that recognizes the characters on vehicle license plates and uses it to confirm an existing reservation.

Grace Period: A predetermined amount of time for which a reservation will be held after the beginning of a reservation period. If the customer does not arrive within this given window, the occupancy state is converted from “reserved” to “available”. The grace period is applied to the end of a walk-in user to avoid “overstay” fees.

Overbooking: Accepting more reservations for spots than are available by forecasting the number of no-show reservations, overstays, understays, and walk-ins.

Spot Occupancy Sensor: A sensor that detects whether a parking spot is available or currently occupied.

Buffer time: The time interval between reservations. Allows the current user time to leave the space before filling it with another user, and helps to account for overstays.

User Interface/Experience (UI/UX): What the user sees and interacts with when accessing and navigating the application/website.

User: A registered customer of the parking software.

Garage Operator: Registered garage owners/managers or operators who manage pricing and garage details.

Rain Check: a discounted or free reservation at a later date

Customer/Client: Those who buy the software, and make decisions about managed services, hosting, configuration changes, etc.

2 - Goals, Requirements, and Analysis

2.1 - Business Goals

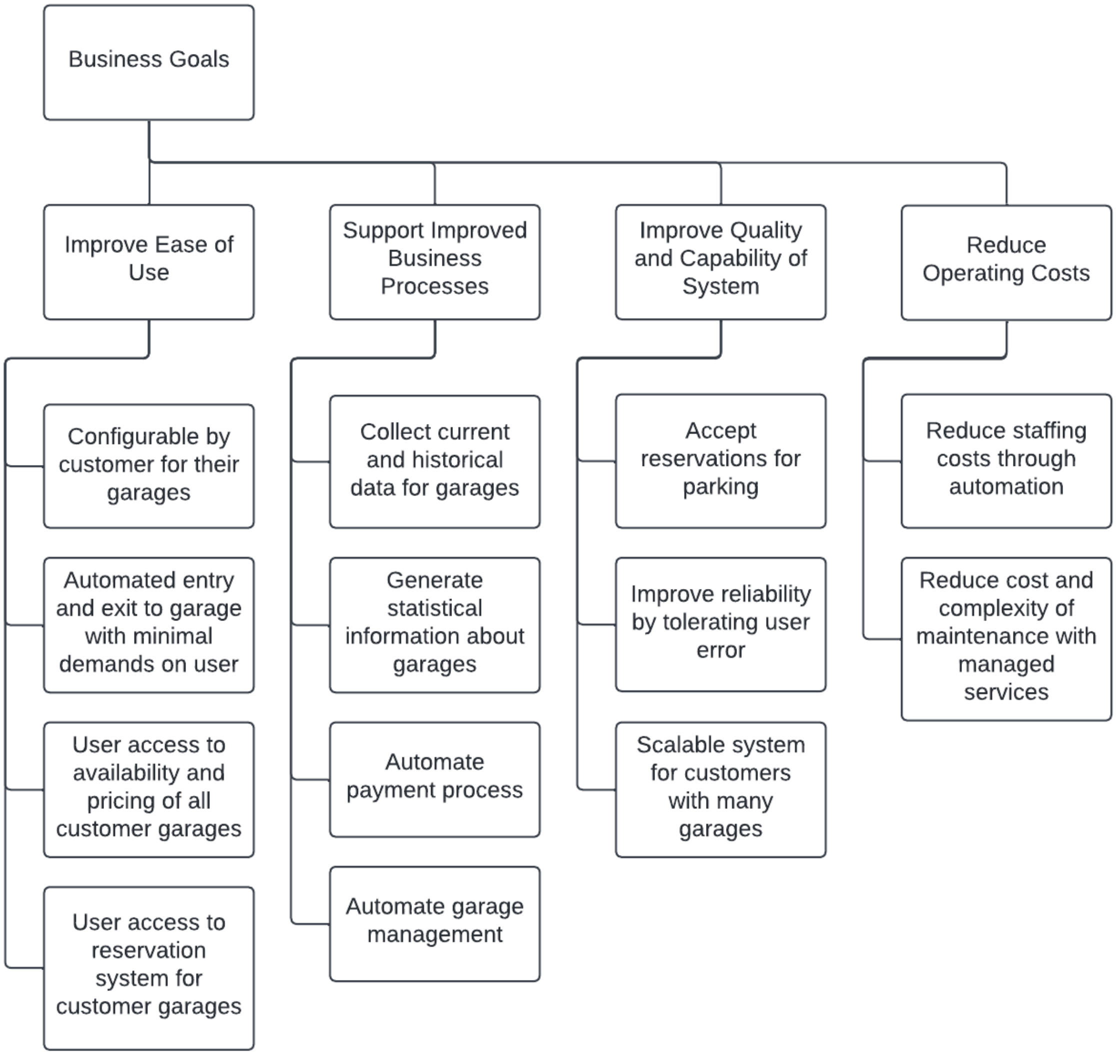

Business Goals
Improve Ease of Use
Support Improved Business Processes
Improve Quality and Capability of System
Reduce Operating Costs
Configurable by customer for their garages
Automated entry and exit to garage with minimal demands on user
User access to availability and pricing of all customer garages
User access to reservation system for customer garages
Collect current and historical data for garages
Generate statistical information about garages
Automate payment process
Automate garage management
Accept reservations for parking
Improve reliability by tolerating user error
Scalable system for customers with many garages
Reduce staffing costs through automation
Reduce cost and complexity of maintenance with managed services

Figure 1 - Hierarchical Diagram of Business Goals

We have divided the business goals into goals and subgoals. The primary goals are familiar to any business, while the subgoals are particular to this project. The primary goals and hierarchical structure of figure 1 above borrow from the technical report *Categorizing business goals for software architectures* (Kazman and Bass, 2005), which also guided the extraction of the subgoals from the customer problem statement.

While three of the top-level goals follow the organization suggested, what was a subgoal in the above paper has been made a primary goal of our project: improve ease of use. For both customers and users, usability is the difference between enthusiastic uptake and simple necessity. For customers, we set our sights on an automated garage and improved management system. For users, we target easy access to both information and the physical garage.

Automated data collection, statistics, payments, and garage access improve upon traditional manual garages. A reservation system, support for managing multiple locations, and fault-tolerance improve upon the quality and capabilities of older garage technology. The reduced staffing required by the automated garage, reduced further with the use of a managed service, reduces operating costs for customers of our system.

2.2 - Enumerated Functional Requirements

REQ	PW	Description
User Interaction		
REQ-01	10	User Registration - A user can register for an account
REQ-02	9	User Registration - a user can reset their password
REQ-03	5	User Registration - A Garage Operator account can upgrade a user account to a garage operator
REQ-04	10	User Registration - A user must register for an account before parking in the garage
REQ-05	10	Parking Space Requests - A user can search for a parking space by location and/or date
REQ-06	10	Parking Space Requests - A user can reserve a spot in a specific garage in the customer’s network

REQ-07	7	Parking Space Requests - A user can make multiple guaranteed reservations for a permanent spot
REQ-08	8	Parking Space Requests - Users can update their guaranteed reservations
REQ-09	6	Parking Space Requests - A user can choose to pay a fee for a larger “grace window”
REQ-10	7	Parking Space Requests - A user may miss their reservation and pay full price
REQ-11	7	Parking Space Requests - A user may show up late to their reservation and pay full price
REQ-12	7	Parking Space Requests - A user may leave their reservation early and pay full price
REQ-13	5	Parking Space Requests - A user can extend their reservation up to a half hour early, if there is availability
REQ-14	5	Parking Space Requests - A walk-in user can extend the duration of their stay
REQ-15	7	Parking Space Requests - A user can overstay their reservation and will pay a fee
REQ-16	8	Parking Space Requests - A user can have up to three reservations at a time
REQ-17	5	Parking Space Requests - A user can merge reservations that are close together
REQ-18	8	Parking Space Requests - A user may book a reservation in a full garage
REQ-19	9	Parking Space Requests - A user can modify a reservation prior to start
REQ-20	9	Parking Space Requests - A user can cancel a reservation prior to start
REQ-21	8	Parking Space Requests - A user can view all reservations on one screen
REQ-22	5	Parking Space Requests - Users will be billed monthly
REQ-23	10	Account Management - Users can add or remove Vehicles

		from their account
REQ-24	10	Account Management - Users can change their automatic payment method from their account
REQ-25	10	Account Management - Users can update their contact information
Garage Access Control		
REQ-26	10	Arrivals - A user must be able to trigger the elevator to access a reserved spot
REQ-27	9	Arrivals - A user’s license plate can be scanned and used to find reservation or account
REQ-29	10	Arrivals - A user will receive a spot number to park in via display board after confirming their reservation
REQ-30	10	Arrivals - A user can park on the first floor without a reservation
REQ-31	8	Arrivals - A user must provide their member number and estimated duration on arrival as a walk-in
REQ-32	9	Arrivals - A walk-in user will receive a spot number once the estimated duration is provided
REQ-33	4	Arrivals - A walk-in user will pay a fee if they do not park in the given spot
REQ-34	9	Arrivals - A user can search for a reservation by reservation number, or member number if the reservation is not automatically found by the elevator
REQ-35	7	Arrivals - A user will receive a notification if they enter an invalid reservation or member number to the console
REQ-36	8	Arrivals - The user may not park without a license plate
REQ-37	8	Arrivals - A user may show up within the grace period of their reservation with no penalties
REQ-38	7	Arrivals - A user can show up after their grace period and receive an available “reservation” spot
REQ-39	4	Arrivals - A user will receive a fee if they park in the incorrect spot for their reservation

REQ-40	6	Arrivals - A user will be asked to leave if they do not have a valid reservation, the garage is full, or they do not have a license plate
REQ-41	3	Arrivals - If a user does not leave when asked, security will be notified
REQ-42	7	Arrivals - If a user arrives to a full garage, with a reservation, due to overbooking, the user will be asked to leave via a display board and be given a rain check
REQ-43	3	Arrivals - If there is current availability to accommodate the reservation at another garage, the user will be provided with options between the rain check or alternate garages
REQ-44	10	Departures - A user must be able to trigger the exit
REQ-45	10	Departures - A user’s reservation will end on departure
REQ-46	5	Departures - A walk-in user can leave early and will NOT be charged for their entire expected duration
Monitoring Occupancy and Space Assignment		
REQ-47	10	Monitoring Occupancy - Current Occupancy will be counted via sensors on parking spots
REQ-48	7	Monitoring Occupancy - Occupancy will be displayed on entrance to the garage and by garage for users on the website
REQ-49	9	Monitoring Occupancy - Expected Reservation occupancy will be considered for future dates based off the current count of reservations for that date range
REQ-50	10	Space Assignment - A user will be assigned a space upon arrival
REQ-51	10	Space Assignment - Assigned Spaces will be displayed on the display board in the elevator
REQ-52	10	Space Assignment - A user with a reservation will be assigned a space on the upper floors
REQ-53	6	Space Assignment - A walk-in user will be assigned a space on the bottom floor
Simulation of Arrivals and Departures		

REQ-54	9	An Admin must be able to enter the license plate information that would be read by the elevator camera
REQ-55	9	An Admin must be able to enter the Member Number or Reservation Number that the user would enter on the console
REQ-56	9	An Admin Must be able to update the occupancy of the parking spots as read by the sensors
REQ-57	9	An Admin must be able to enter the license plate information that would be read by the exit ramp camera
REQ-58	3	An Admin will be able to view the results of the Automatic Simulation
Statistical Data Collection		
REQ-59	3	A garage operator will be able to view a dashboard for garage statistics
REQ-60	5	A garage operator will be able to view understays, overstays, overbook rating, and occupancy rating
REQ-61	2	A garage operator will be able to view data by week, by month and by year
System Administration		
REQ-62	10	A garage operator must be able to add garages to their network
REQ-63	9	A garage operator must be able to configure the number of floors and spots per floor for each garage
REQ-64	6	A garage operator must be able to deactivate a garage
REQ-65	7	A garage operator must be able to set prices for reservations, walk-ins and guaranteed reservations for each garage
REQ-66	4	A garage operator must be able to set and update the overbooking percentage for each garage
REQ-67	4	A garage operator must be able to configure fees

2.3 - Enumerated Nonfunctional Requirements

REQ	PW	Description
Functional		
REQ-68	9	The elevator entrance and exit will have operational cameras to detect license plates
REQ-69	7	The elevator will rotate floors when possible to ensure the user with the reservation has enough time to get to their parking spot
REQ-70	7	An external image processing system local to the garage will process the images captured into a payload to be sent to the system via API
REQ-71	9	Each Spot will have a light reactive sensor to detect occupancy of that spot
REQ-72	9	Spot sensors and cameras will be assumed to be operational at all times
REQ-73	6	Displays will be connected to the software via internet connection
REQ-74	8	Consoles will be touch screen so users can select multiple options and type in their reservation/member numbers
Usability		
REQ-75	9	The interface must be clean and easy to use without instruction
REQ-76	7	Required fields will be marked and error messages will be clearly displayed so mistakes can be corrected by the user
REQ-77	9	In order to use the system, an automatic payment method must be set up by the user, and contact methods must be provided
REQ-78	8	The spots will be assigned when the user arrives, therefore in order to avoid congestion the algorithm must prioritize efficiency

Reliability		
REQ-79	9	The system must be implemented on a stable server
REQ-80	5	Managed Services (if chosen by client) will provide credits for any downtimes
REQ-81	7	5G connectivity is suggested for mobile users
Performance		
REQ-82	6	Algorithms must be efficient in order to limit wait time for users
REQ-83	7	Database updates will be efficient and prioritize updates that need to be reflected on the UI immediately
Supportability		
REQ-84	10	The system must be usable on mobile device, computer and tablet via web browser
REQ-85	10	Major web browsers such as Chrome, Firefox, and Safari must be supported
REQ-86	9	Disability Accommodations must be made in the web design

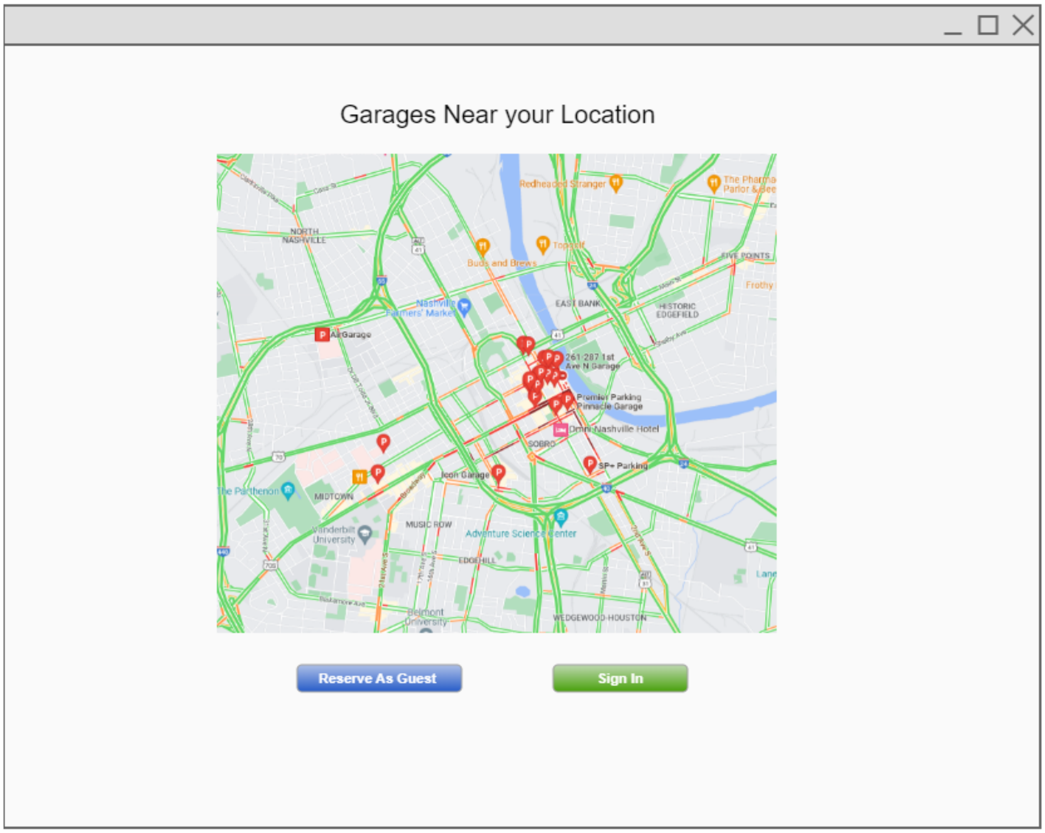
2.4 - User Interface Requirements

REQ	PW	Description
User Interface Requirements		
REQ-87	9	Users will be able to find a garage to book near them.
REQ-88	6	New user account creation.
REQ-89	6	Users will access their account to register / alter registrations/ view payment history.
REQ-90	4	Owners will be able to add new garages to their network(s).
REQ-91	9	Users will be able to reserve their spot online.

REQ-92	9	Walk-in user will be able to reserve a ground floor spot
REQ-93	9	Users enter their payment option securely.

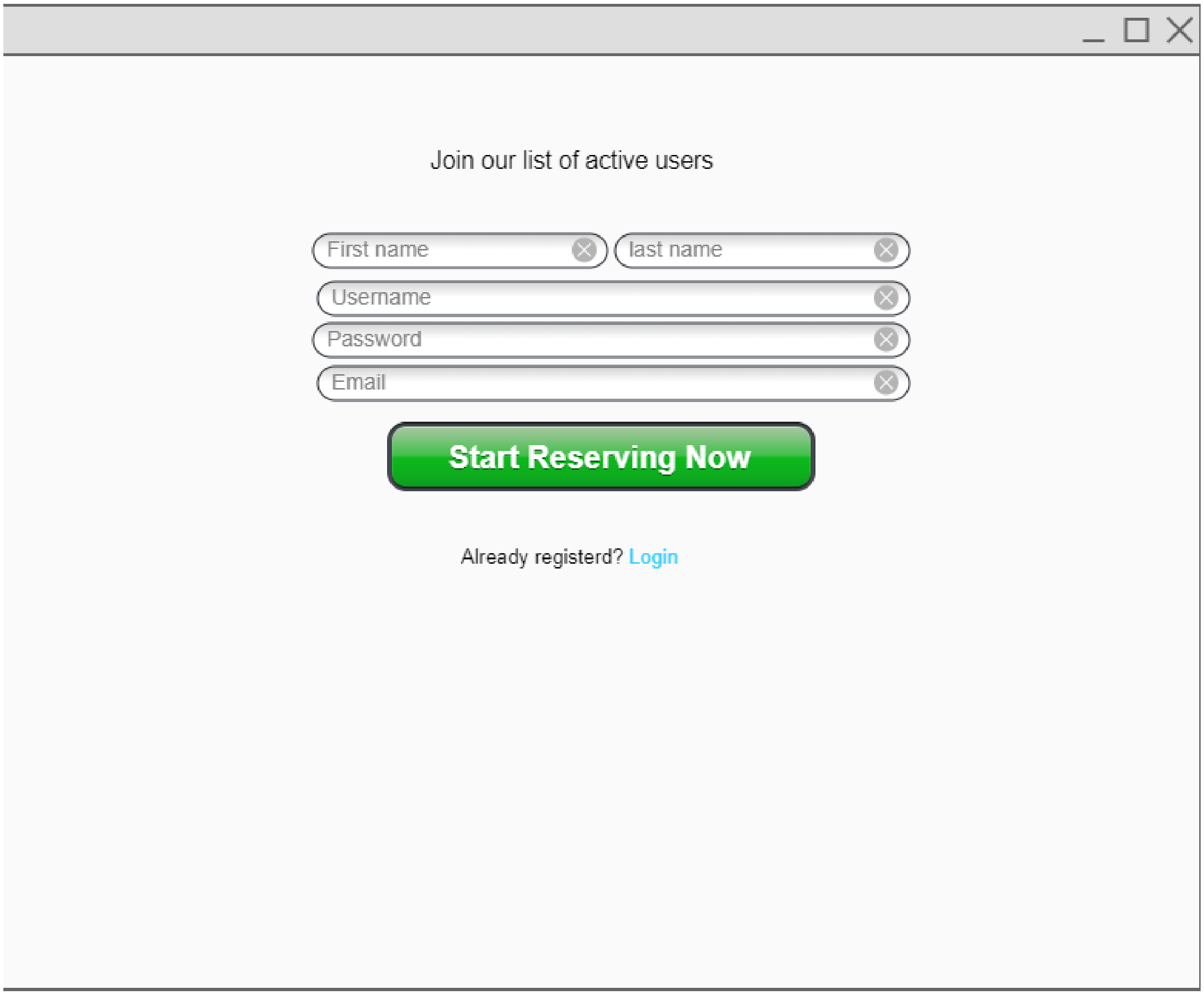
Near Garage Locator Page (Main):

User will be able to locate a garage near them, and select to reserve is a guest, or sign in to reserve.



Registration Page:

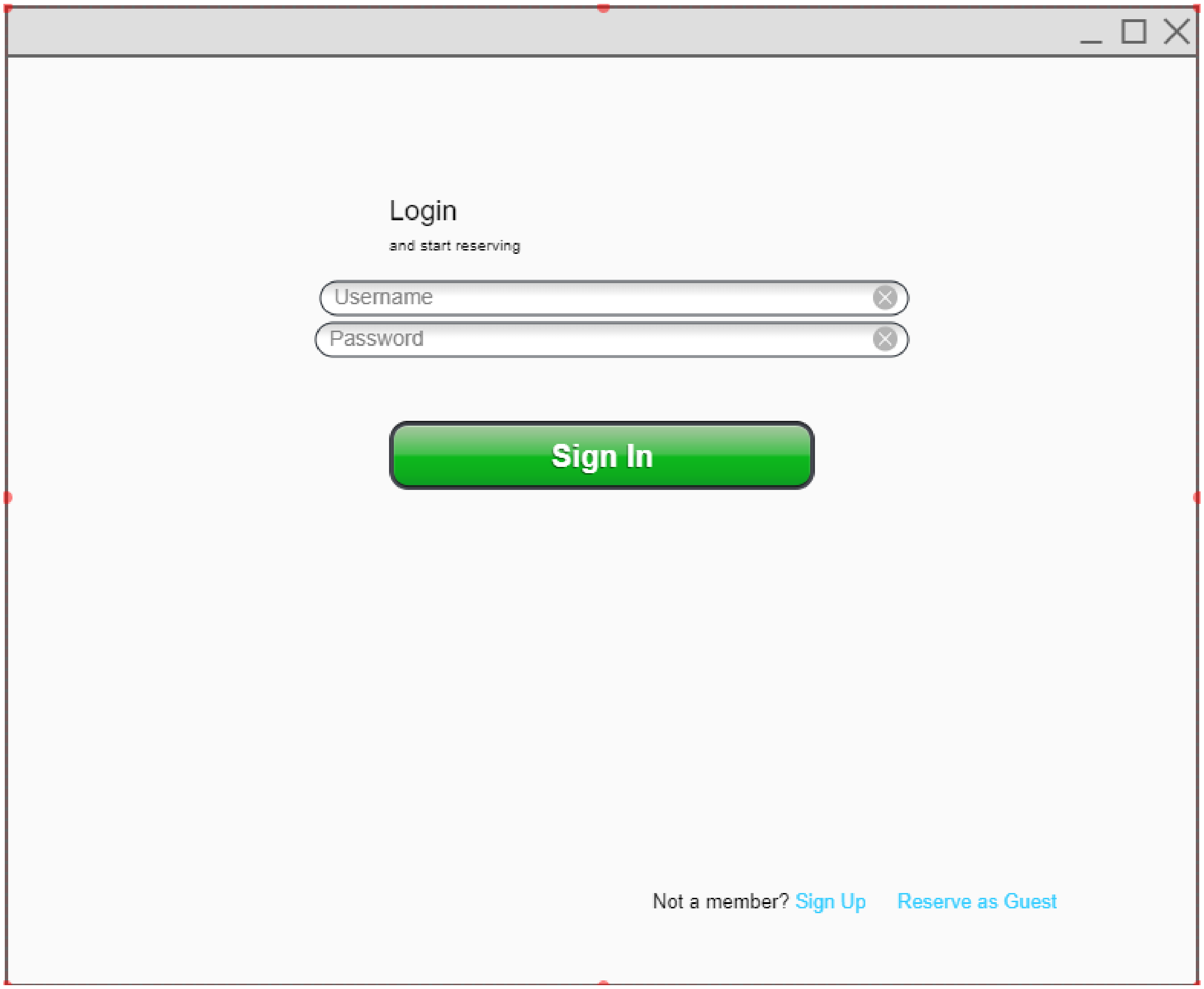
Registration page gives a new user the ability to create a name with the required information.



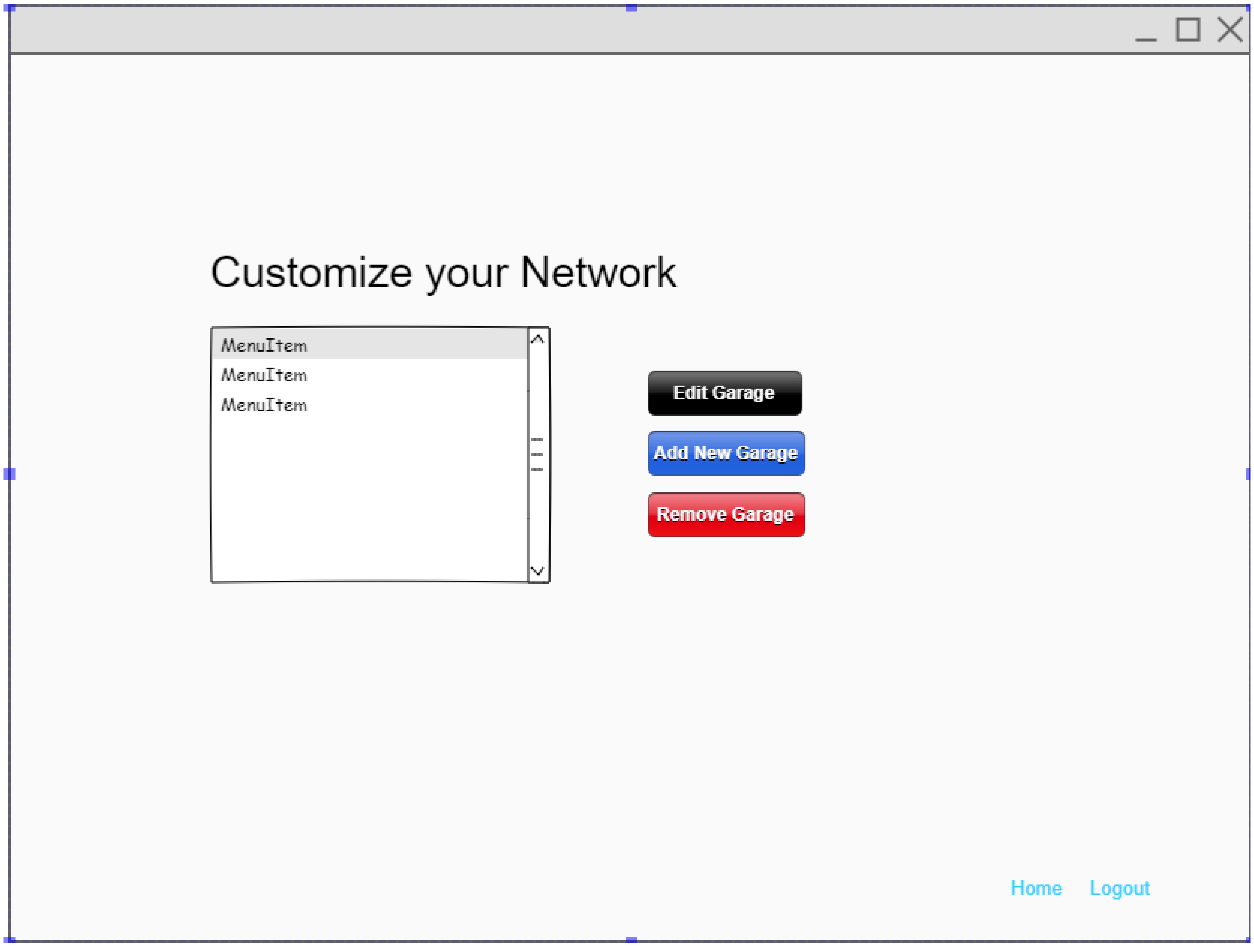
Join our list of active users

Already registerd? Login

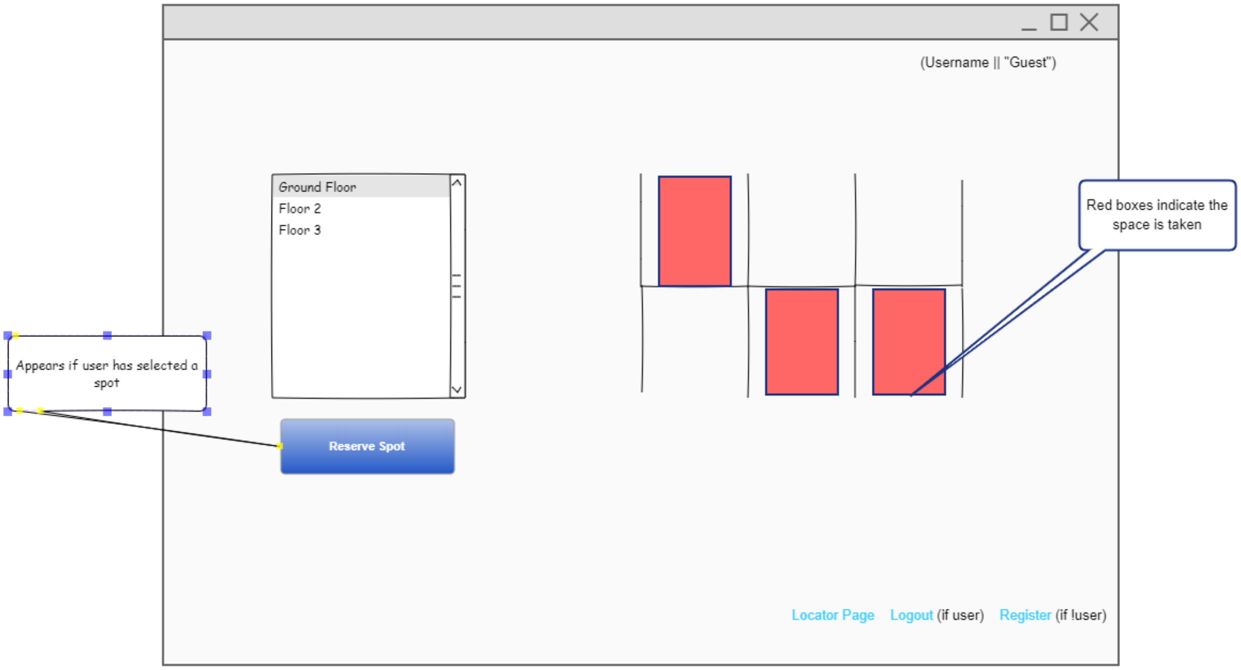
Login Page:


Login
and start reserving
Username
Password
Sign In
Not a member? Sign Up
Reserve as Guest

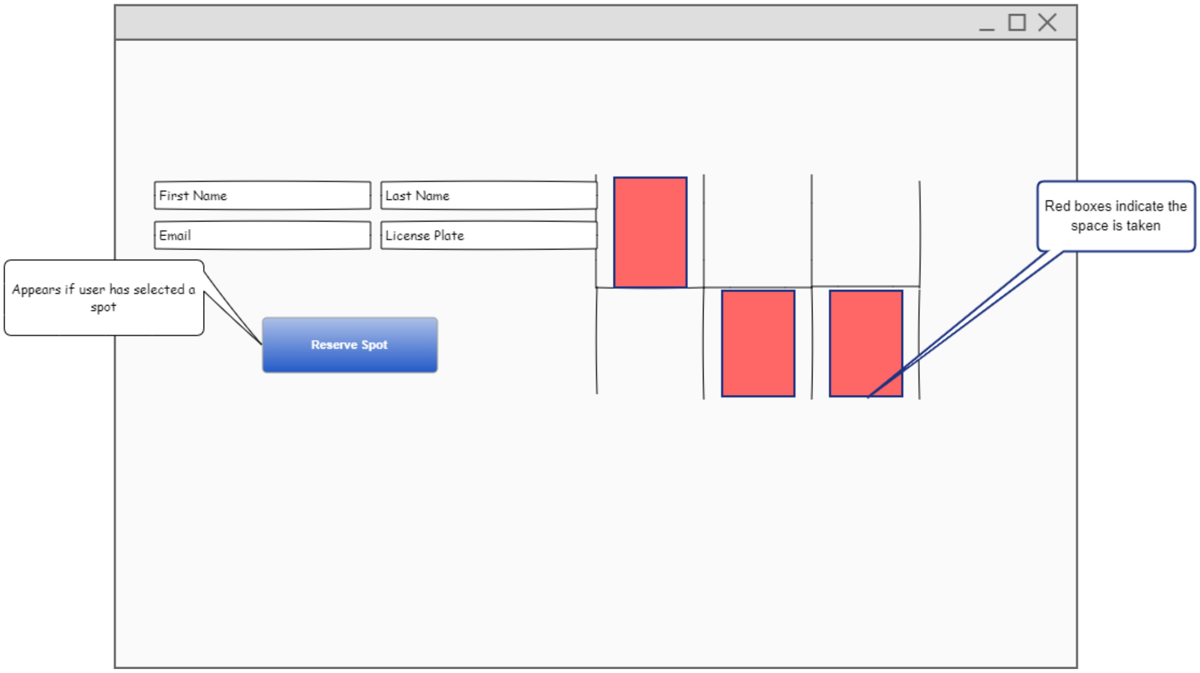
Garage Network Page



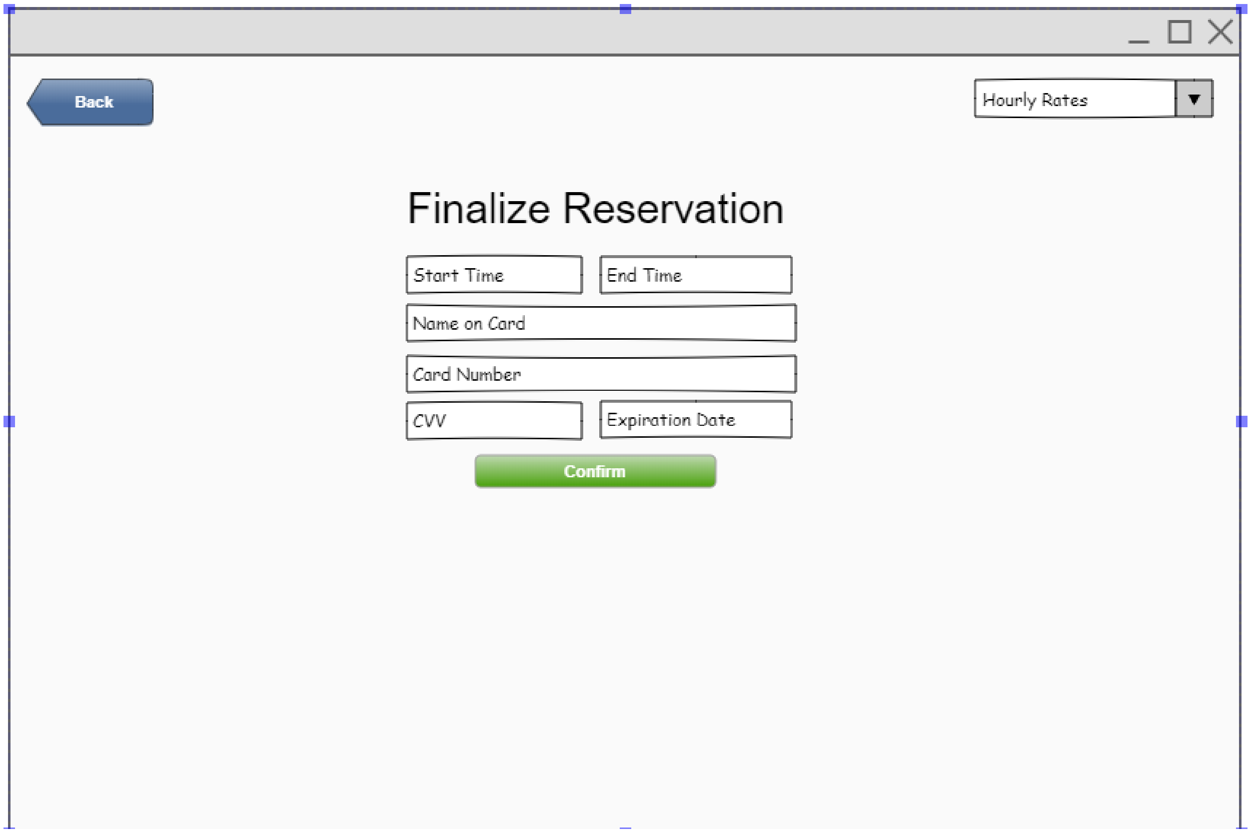
Online Registration Page



Walk-In Page


First Name
Last Name
Email
License Plate
Red boxes indicate the space is taken
Appears if user has selected a spot
Reserve Spot

Payment Page



3 - Use Cases

3.1 - Stakeholders

Stakeholders include all persons who have stand to gain and profit from implementing or converting their garage to support this system of management, as well as persons who enroll/sign-up to use this garage service.

1. Garage Owners: Adopt this system in pursuit of diversification in order to pursue financial and grow their financial interests.
2. Managers: Increase in operational efficiency and data collecting to more effectively manage operations and consume statistics.
3. Investors: Those who choose to invest in adopting this garaging system to earn on future financial gains.
4. Users: Consumers who register to use this garaging system to more efficiently locate and schedule guaranteed parking quickly with an easy to use interface.

3.2 - Actors and Goals

Actor	**Actor's Goals**	**Use Case(s)**
Owner/Manager	Update pricing	UC-16
Owner/Manager	View current parking status (occupancy/vacancies)	UC-14
Owner/Manager	View garage stats	UC-14
Owner/Manager	Add a garage to network of garages	UC-15
User	View garages within a given distance	UC-02
User	Reserve a space	UC-02
User	Modify reservation	UC-03, UC-07
User	Park as a walk-in customer	UC-10b
User	Exit the garage	UC-05, UC-06
User	Retrieve reservation with reservation number	UC-02, UC-03
User	Add payment information	UC-02, UC-03, UC-08

User	Receive reservation confirmation	UC-02, UC-03
User	Enter the garage	UC-04, UC-03
Database	Maintain user registry	UC-01
Database	Maintain space availability	UC-13
Database	Maintain statistical information	UC-14
Camera	Read license plate information to confirm reservation status	UC-10
Parking Sensors	Mark occupied parking spaces	UC-12, UC-13
Digital Display	Allow walk-in customers to make an on the spot reservation	UC-10,UC-13

3.3 - Use Cases

3.3.1 - Casual Descriptions

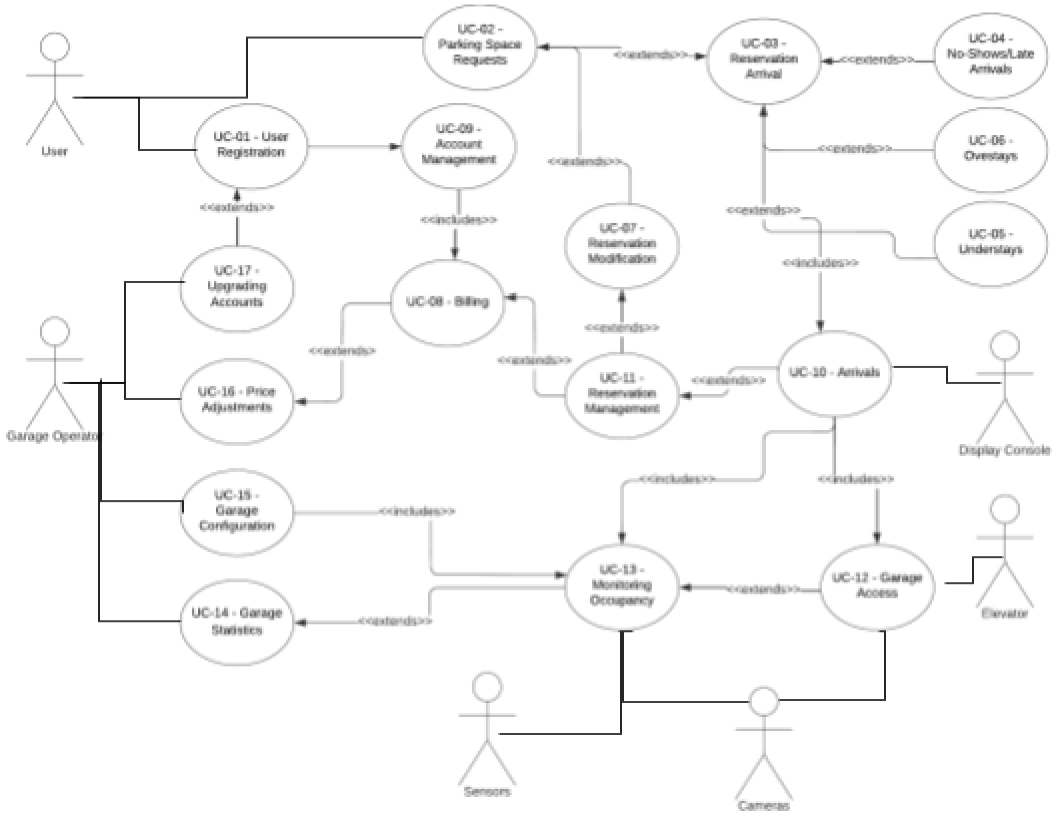
UC	REQs	Description
User Interaction		
UC-01	REQ-01, REQ-04, REQ-02, REQ-88	User Registration - A user can register for an account with their email, phone number and their password. On login, a user will use this information to access their account or reset password,
UC-02a	REQ-05, REQ-06, REQ-07, REQ-16, REQ-87, REQ-91, REQ-49	Parking Space Requests (Single) - A user can search for a parking space by location and/or date and receive a list of available spots based on their criteria. A user can request a spot and make up to 3 non-overlapping reservations if they have a card on file.
UC-02b	REQ-07, REQ-08, REQ-91	Parking Space Request (Permanent/Guaranteed) - A user can make multiple guaranteed reservations for a permanent spot. A user can select the frequency of the reservations and the timeframe these reservations are needed.

UC-03	REQ-09, REQ-37, REQ-38	Reservation Arrival - A user with a reservation can arrive within the Grace Period with no charge, choose to pay a fee for a larger Grace Period, or show up outside of the Grace Period and only receive a spot if the garage is not full and still pay full price
UC-04	REQ-10, REQ-11	No-Shows/Late Arrivals - A user may miss their reservation or show up late and still pay full price. If the user shows up late, they only may receive a spot if availability allows.
UC-05	REQ-12, REQ-46	Understays - A user may leave their reservation early and pay full price. Walk-in's do not pay full price if they leave early
UC-06	REQ-13, REQ-14, REQ-15	Overstays - A both walk-in and reservation users can extend their reservation up to a half hour early, if there is availability. If they do not extend their reservation, a fee will apply. Walk-ins have the Grace period applied to the end of their stay.
UC-07	REQ-17, REQ-19, REQ-20, REQ-89	Reservation Modification - A user can modify or cancel a reservation prior to start. If the reservation is within an hour of the previous reservation they can be merged.
UC-08	REQ-22, REQ-21, REQ-89	Billing - Users will be billed monthly for all reservations in the month. They will receive a monthly statement via email and can view all historical reservations and their price.
UC-09	REQ-23, REQ-25, REQ-24	Account Management - Users can add or remove Vehicles from their account, update contact information and add/update payment methods
Garage Access Control		
UC-10a	REQ-26, REQ-27, REQ-29, REQ-34, REQ-50, REQ-51, REQ-52 REQ-35	Reservation Arrivals - A user's reservation can be found via associated license plate, member number or reservation number. If the entered reservation or member number is invalid, the user will be notified. The user will see a spot number on the display board and be taken to the proper floor.
UC-10b	REQ-30,	Walk-in Arrivals - A user can park on the first floor

	REQ-31, REQ-32, REQ-50, REQ-51, REQ-92, REQ-35	without a reservation by providing a member number and an estimated duration, then receive a spot number on the first floor. If the entered member number is invalid, the user will be notified.
UC-11	REQ-44	Reservation Management - A reservation will be marked complete when the user triggers the exit and leaves the garage.
UC-12a	REQ-40, REQ-43, REQ-33	Garage Access - A user will not be allowed to park if the walk-in garage is full and they do not have a reservation. If the garage is full and the user does have a reservation, they will be asked to go to a different garage or receive a rain check discount.
UC-12b	REQ-41, REQ-36, REQ-42	Request to Leave - If a spot is not available or a user does not have a license plate, they will be asked to leave. If the user does not leave, security will be notified.
Monitoring Occupancy and Space Assignment		
UC-13	REQ-47, REQ-48, REQ-39	Monitoring Occupancy - Current Occupancy will be counted via sensors on parking spots and displayed on the entrance to the garage and on the website. A user will pay a fee if they do not park in the given spot
Statistical Data Collection		
UC-14	REQ-59, REQ-60, REQ-61	Garage Statistics - A garage operator will be able to view a dashboard for garage statistics with understay, overstay, overbooking rate, occupancy rating, and other statistics
System Administration		
UC-15	REQ-62, REQ-63, REQ-64, REQ-66, REQ-18, REQ-90	Garage Configuration - A garage operator must be able to add garages to their network and define the number of spots on each floor, activate/deactivate the garage, and set the overbooking rate.
UC-16	REQ-65, REQ-67	Price Adjustments - A garage operator must be able to set prices for reservations, walk-ins, guaranteed

		reservations for each garage and fees
UC-17	REQ-03	Upgrading Accounts - A garage operator can upgrade a user account to a garage operator

3.3.2 - Use Case Diagram



3.3.3 - Traceability Matrix

	UC-01	UC-02a	UC-02b	UC-03	UC-04	UC-05	UC-06	UC-07	UC-08	UC-09	UC-10a	UC-10b	UC-11	UC-12a	UC-12b	UC-13	UC-14	UC-15	UC-16	UC-17
REQ-01	10																			
REQ-02	9																			
REQ-03																				5

REQ-04	10																			
REQ-05		10																		
REQ-06		10																		
REQ-07		7	7																	
REQ-08			8																	
REQ-09				6																
REQ-10					7															
REQ-11					7															
REQ-12						7														
REQ-13							5													
REQ-14							5													
REQ-15							7													
REQ-16		8																		
REQ-17								5												
REQ-18																		8		
REQ-19								9												
REQ-20								8												
REQ-21									8											
REQ-22									5											
REQ-23										10										
REQ-24										10										
REQ-25										10										
REQ-26											10									
REQ-27											9									
REQ-29											10									
REQ-30												10								
REQ-31												8								
REQ-32												9								

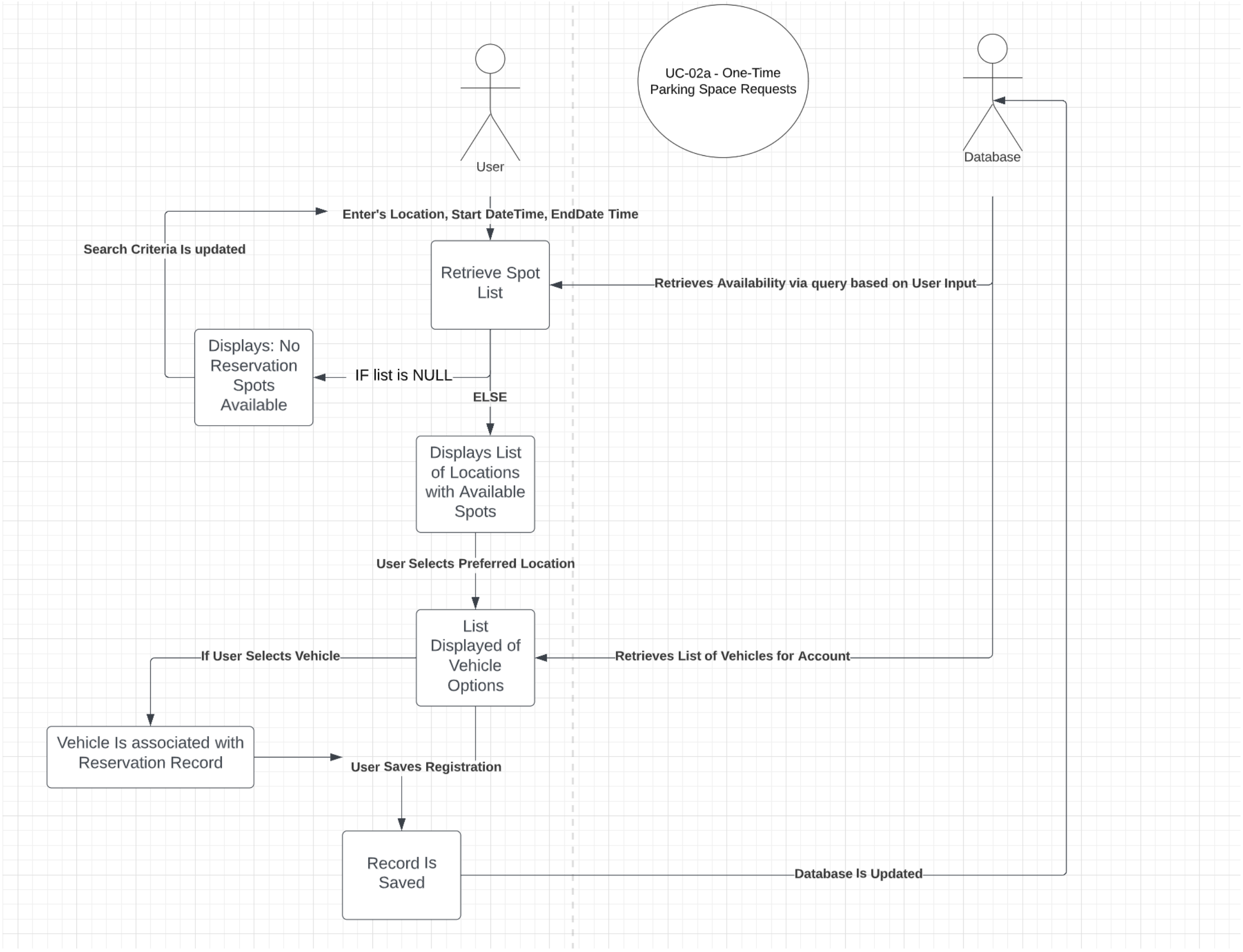
REQ-33														4						
REQ-34											9									
REQ-35											7	7								
REQ-36															8					
REQ-37				8																
REQ-38				7																
REQ-39																4				
REQ-40														6						
REQ-41															3					
REQ-42															7					
REQ-43														3						
REQ-44													10							
REQ-45													10							
REQ-46						5														
REQ-47																10				
REQ-48																7				
REQ-49		9																		
REQ-50											10	10								
REQ-51											10	10								
REQ-52											10									
REQ-53												6								
REQ-59																	3			
REQ-60																	5			
REQ-61																	2			
REQ-62																		10		
REQ-63																		9		
REQ-64																		6		
REQ-65																			7	

REQ-66																		4		
REQ-67																			4	
REQ-87		9																		
REQ-88	6																			
REQ-89								6	6											
REQ-90								4	4									4		
REQ-91		9	9																	
REQ-92												9								
REQ-93										9										
TOTAL	35	62	24	21	14	12	17	32	23	39	75	69	20	13	18	21	10	41	11	5

3.3.4 - Fully-Dressed Description

UC-2a - Requesting a Single Parking Spot

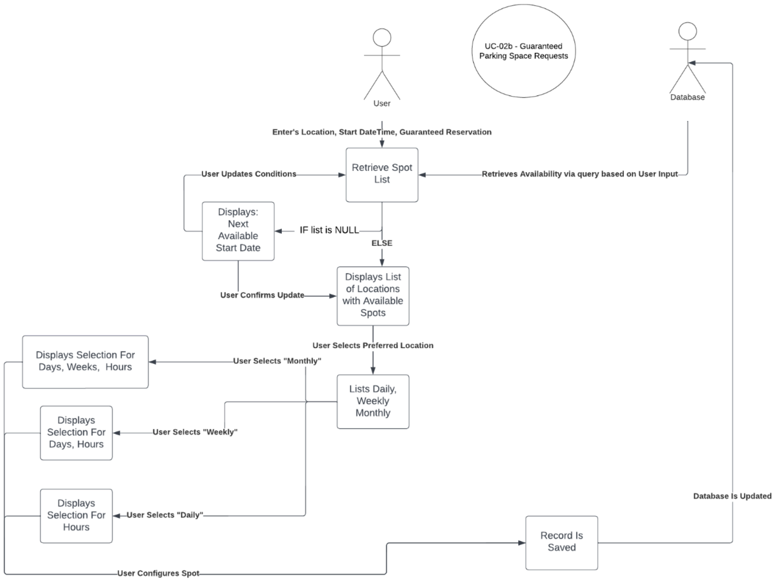
- **Related Requirements -** REQ-05, REQ-06, REQ-07, REQ-16, REQ-87, REQ-91, REQ-49
- **Initiating Actor -** User
- **Actor's Goal -** Find and Request a one-time parking spot
- **Participating Actors -** Database
- **Pre-Condition -** The user is signed in, has a card associated to the account, and there are available spots
- **Post-Condition -** A reservation is made
- **Flow of Events -**
    1. User enters search conditions (location, distance, datetime range)
    2. Database populates a list of garages with availability matching those conditions
    3. User selects a reservation option
    4. (Optional) User selects vehicle to be associated to the reservation
    5. User saves reservation information and receives reservation number
    6. Database is updated with new reservation
- **Diagram -**


UC-02a - One-Time Parking Space Requests
User
Database
Enter's Location, Start DateTime, EndDate Time
Search Criteria Is updated
Retrieve Spot List
Retrieves Availability via query based on User Input
Displays: No Reservation Spots Available
IF list is NULL
ELSE
Displays List of Locations with Available Spots
User Selects Preferred Location
List Displayed of Vehicle Options
If User Selects Vehicle
Retrieves List of Vehicles for Account
Vehicle Is associated with Reservation Record
User Saves Registration
Record Is Saved
Database Is Updated

UC-2b - Requesting a Permanent/Guaranteed Spot

- **Related Requirements -** REQ-07, REQ-08,
- REQ-91
- **Initiating Actor -** User
- **Actor's Goal -** Request a recurring guaranteed spot
- **Participating Actors -** Database
- **Pre-Condition -** The user is signed in, has a card associated to the account, and there are available spots for their start date
- **Post-Condition -** The user creates a recurring reservation
- **Flow of Events -**
    1. User enters search conditions (location, start date)
    2. User selects "Guaranteed" spot
    3. User selects requirements
        - Frequency: Weekly, Monthly Daily
        - If Daily: Only time range appears
        - If Weekly: Day selector AND Time Range Appears

        - If Monthly: Day Selector, Time Range, AND Week Selector appears
    4. User Saves reservation
    5. Database notifies user IF the start date cannot be accommodated and updates the start date to the earliest available
        - User confirms changes
    6. Database is updated with new reservation
- **Diagram -**



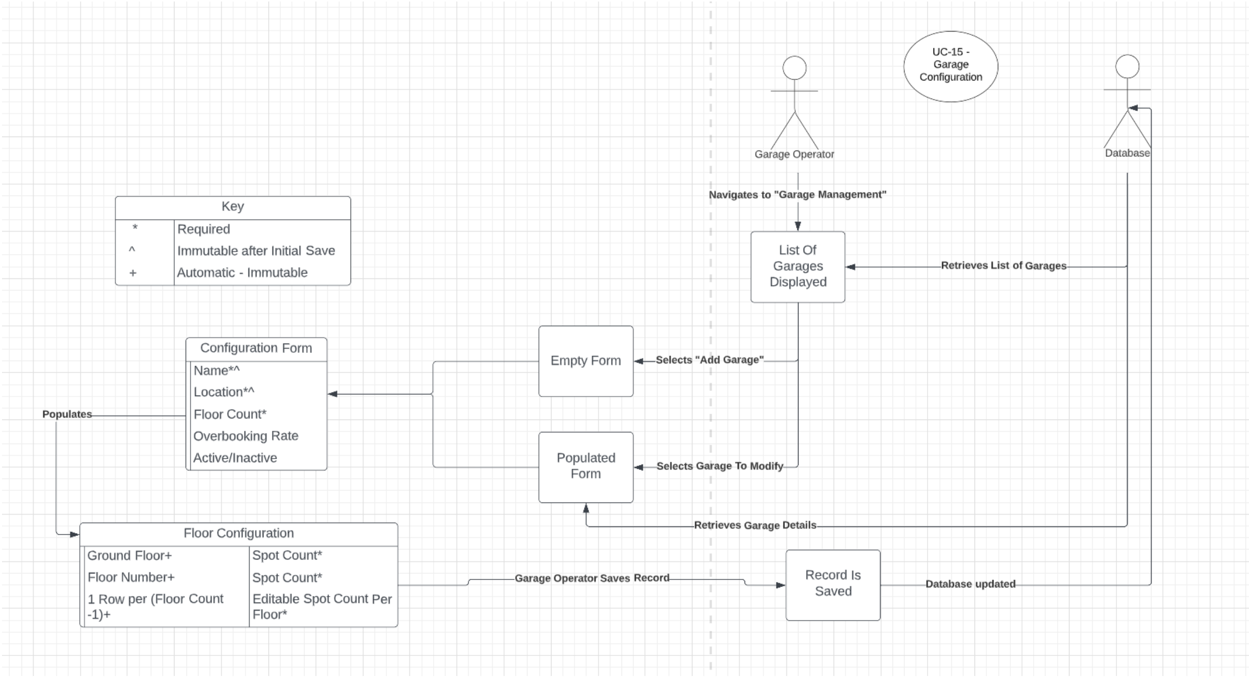
User
UC-02b - Guaranteed Parking Space Requests
Database
Enter's Location, Start DateTime, Guaranteed Reservation
Retrieve Spot List
User Updates Conditions
Retrieves Availability via query based on User Input
Displays: Next Available Start Date
IF list is NULL
ELSE
Displays List of Locations with Available Spots
User Confirms Update
User Selects Preferred Location
Displays Selection For Days, Weeks, Hours
User Selects "Monthly"
Lists Daily, Weekly Monthly
Displays Selection For Days, Hours
User Selects "Weekly"
Displays Selection For Hours
User Selects "Daily"
Database Is Updated
Record Is Saved
User Configures Spot

UC-15 - Adding a Garage to a Customer's Network

- **Related Requirements -** REQ-62, REQ-63, REQ-64, REQ-66, REQ-18, REQ-90
- **Initiating Actor -** Garage Operator
- **Actor's Goal -** Add/Update a garage's configuration
- **Participating Actors -** Database
- **Pre-Condition -** Garage Operator is logged in
- **Post-Condition -** Database is updated with new garage information
- **Flow of Events -**
    1. Garage Operator navigates to garage management page
    2. Garage Operator Selects Add or Modify Garage
        - Add:
            1. Name is REQUIRED
            2. Location is REQUIRED
            3. Number of Floors is REQUIRED
            4. Garage operator enters number of spots for each floor
            5. Overbooking rate is added

        6. Active/Inactive toggle is swapped to activate
    - Modify:
        1. Number of Floors can be modified
        2. Number of spots can be modified
        3. Overbooking rate may be modified
        4. Active/Inactive toggle may be modified
3. Record is saved, database is updated

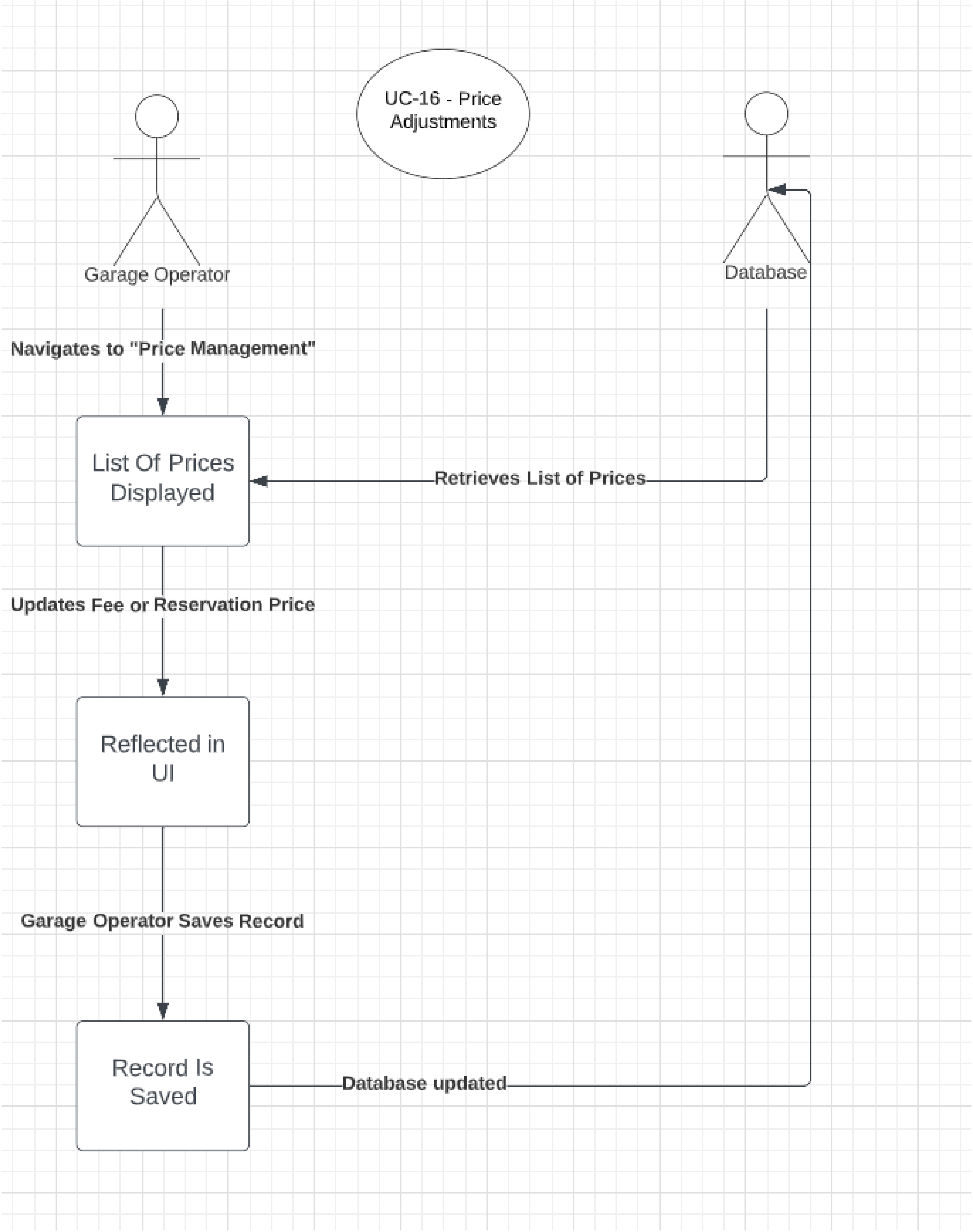
- **Diagram -**



UC-16 - Adjusting Prices for a Garage

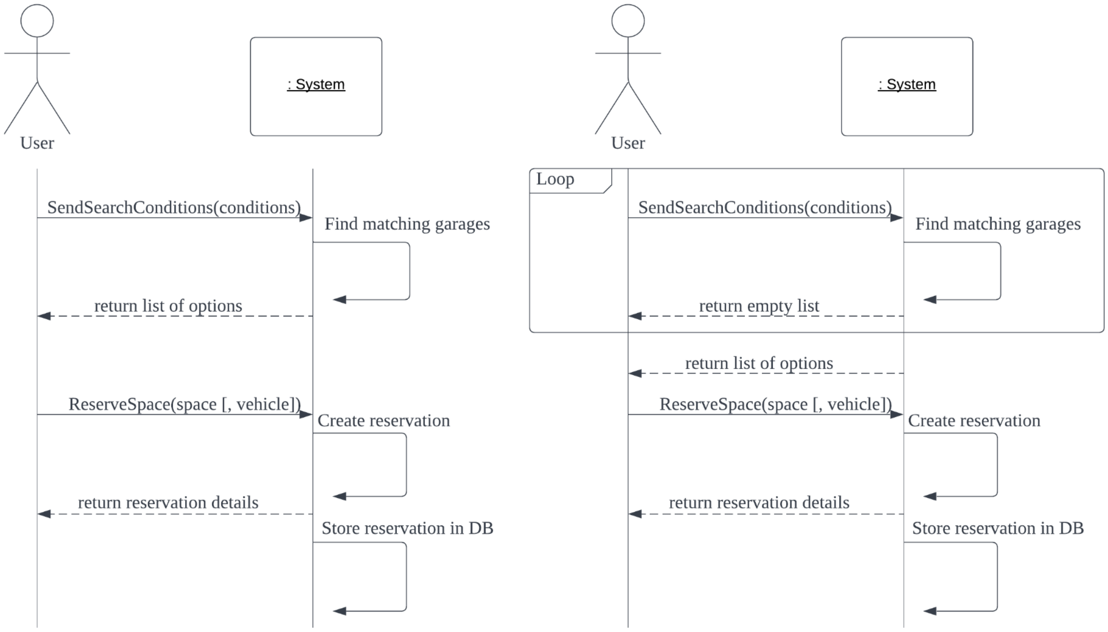
- **Related Requirements -** REQ-65, REQ-67
- **Initiating Actor -** Garage Operator
- **Actor's Goal -** Configure Prices and Fees
- **Participating Actors -** Database
- **Pre-Condition -** Garage operator is signed in
- **Post-Condition -** Database is updated with new pricing
- **Flow of Events -**
    1. Garage Operator navigates to pricing page
    2. Garage Operator Selects pricing type to modify
    3. Updates Pricing
    4. Saves Record and Database updates

- **Diagram -**


UC-16 - Price Adjustments
Garage Operator
Database
Navigates to "Price Management"
List Of Prices Displayed
Retrieves List of Prices
Updates Fee or Reservation Price
Reflected in UI
Garage Operator Saves Record
Record Is Saved
Database updated

3.4 - System Sequence Diagrams

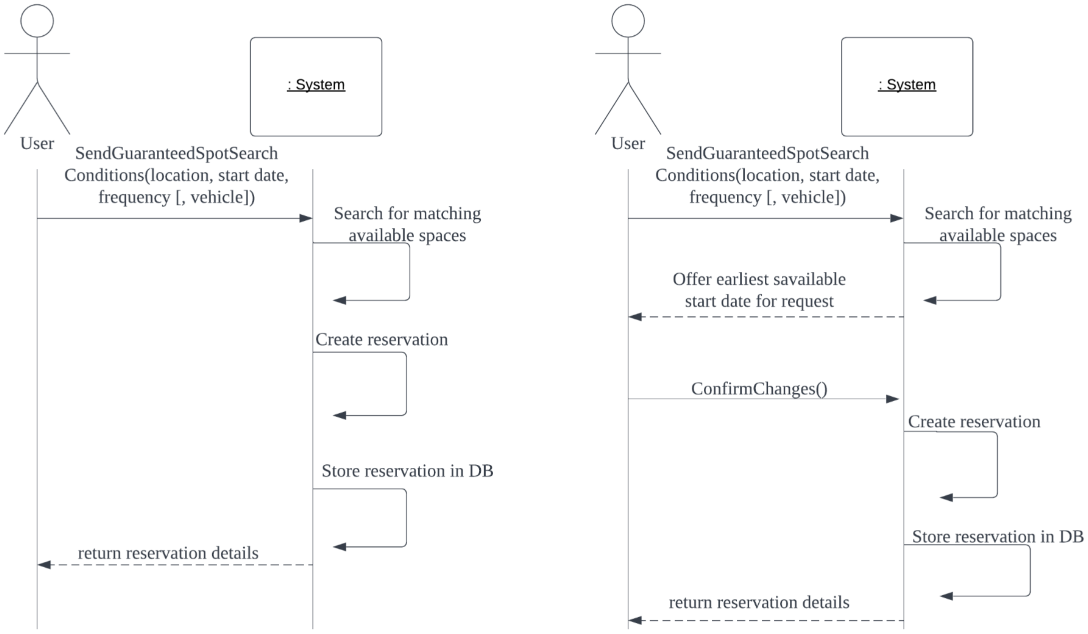
UC-2a - Requesting a Single Parking Spot



(a)
Main success case

(b)
No available spaces for
the given conditions

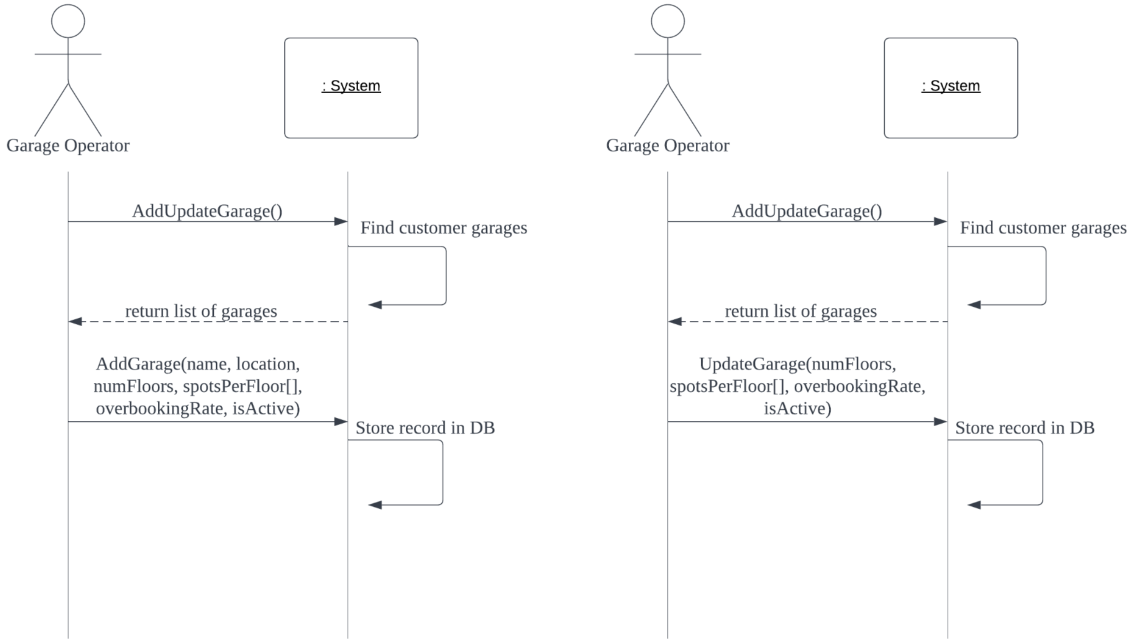
UC-2b - Requesting a Permanent/Guaranteed Spot


User
: System
SendGuaranteedSpotSearch Conditions(location, start date, frequency [, vehicle])
Search for matching available spaces
Create reservation
Store reservation in DB
return reservation details
User
: System
SendGuaranteedSpotSearch Conditions(location, start date, frequency [, vehicle])
Search for matching available spaces
Offer earliest savailable start date for request
ConfirmChanges()
Create reservation
Store reservation in DB
return reservation details

(a)
Main success case

(b)
System offers earliest available start date for search if exact match is not found

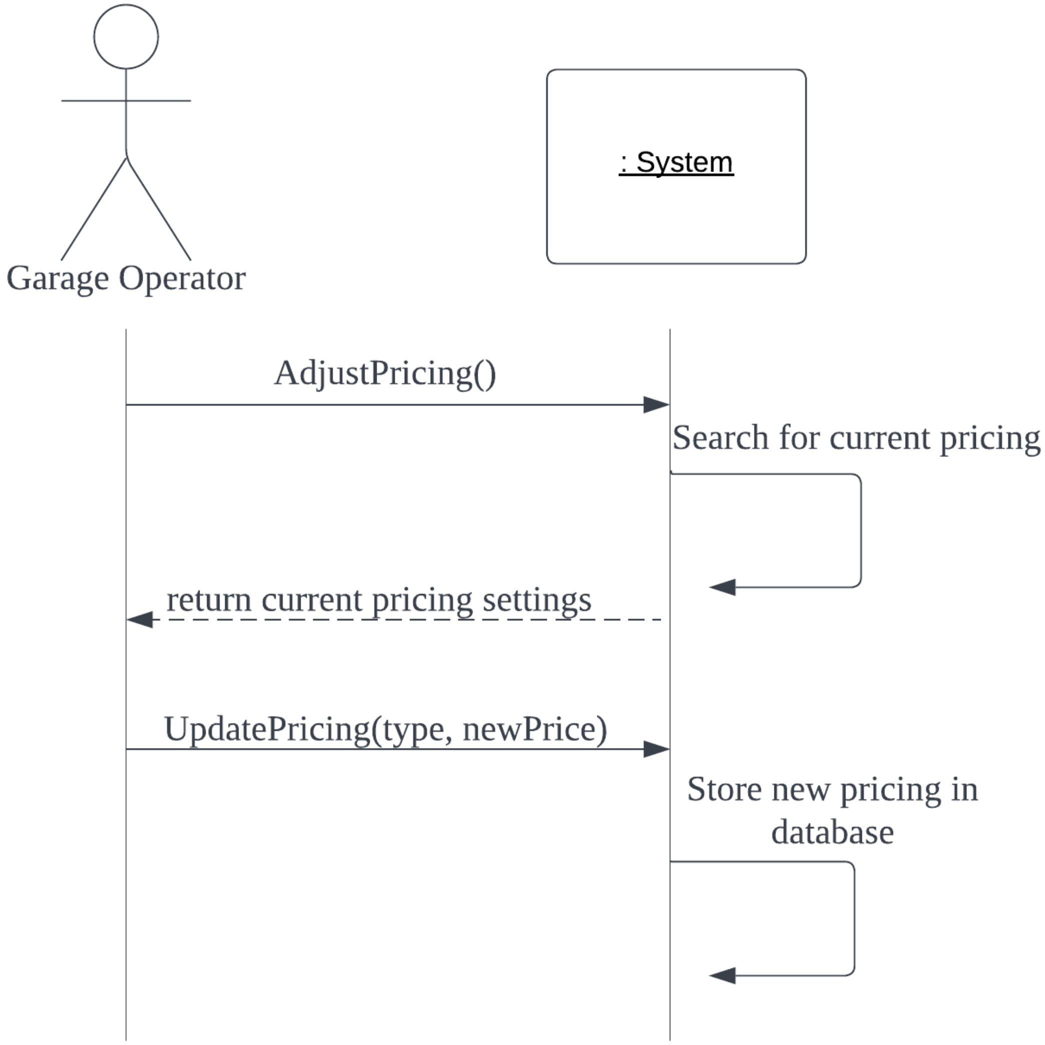
UC-15 - Adding a Garage to a Customer’s Network



(a)
Adding a new garage

(b)
Updating an existing garge

UC-16 - Adjusting Prices for a Garage

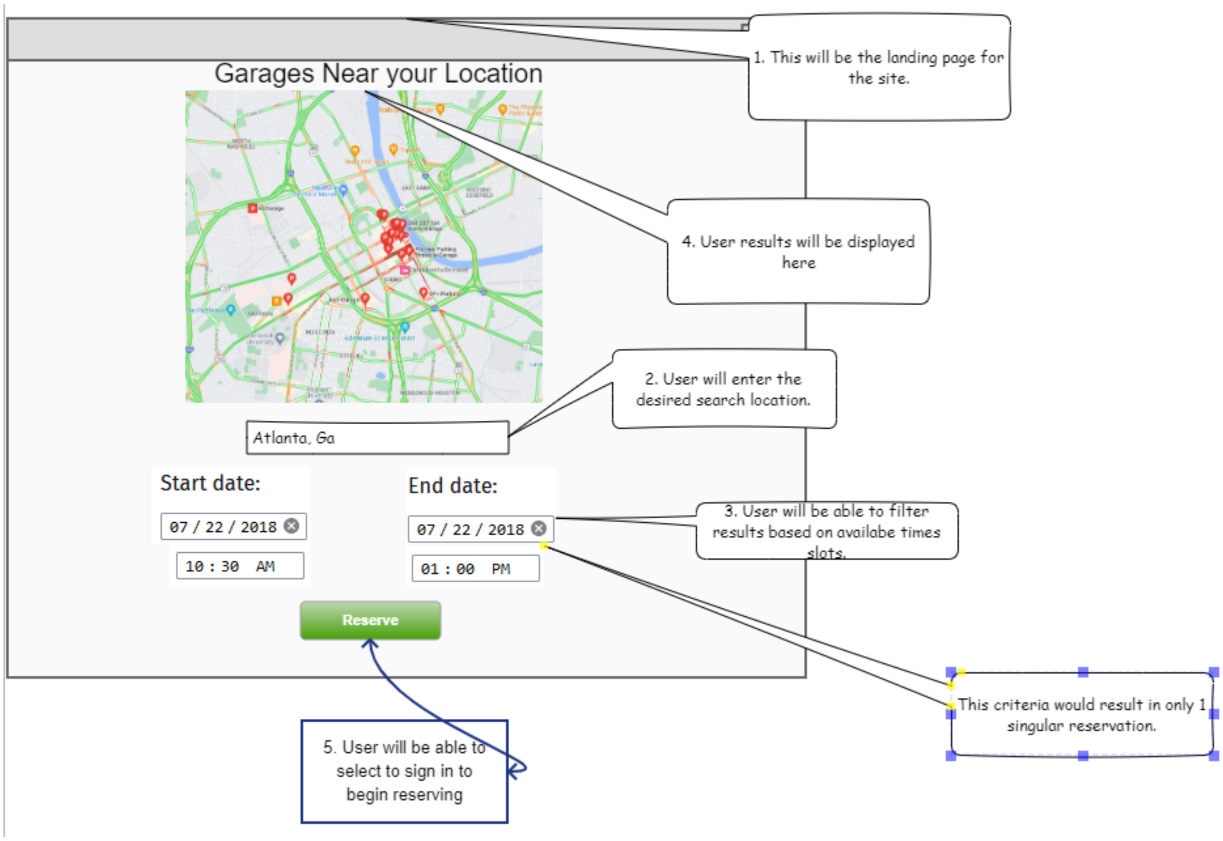

Garage Operator
: System
AdjustPricing()
Search for current pricing
return current pricing settings
UpdatePricing(type, newPrice)
Store new pricing in database

(a)
Main success case

4 - User Interface Specification

4.1 - Preliminary Design

UC-02a — Requesting a Single Parking Spot



Atlanta, Ga

Start date: 07 / 22 / 2018 10 : 30 AM

End date: 07 / 22 / 2018 01 : 00 PM

1. This will be the landing page for the site.

2. User will enter the desired search location.

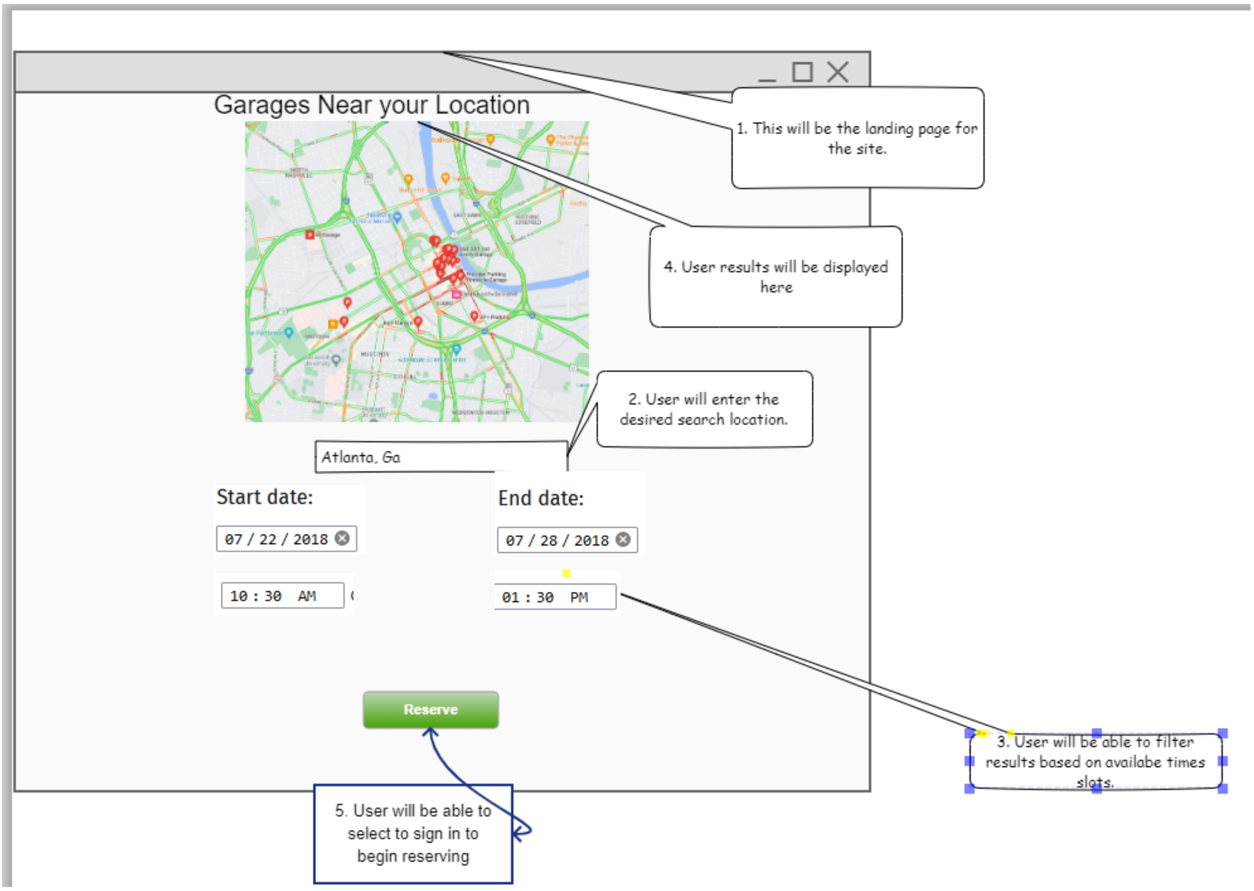
3. User will be able to filter results based on availabe times slots.

4. User results will be displayed here

5. User will be able to select to sign in to begin reserving

This criteria would result in only 1 singular reservation.

UC-02b — Requesting a Permanent/Guaranteed Spot



Garages Near your Location

1. This will be the landing page for the site.

2. User will enter the desired search location.

Atlanta, Ga

3. User will be able to filter results based on availabe times slots.

4. User results will be displayed here

Start date: 07 / 22 / 2018

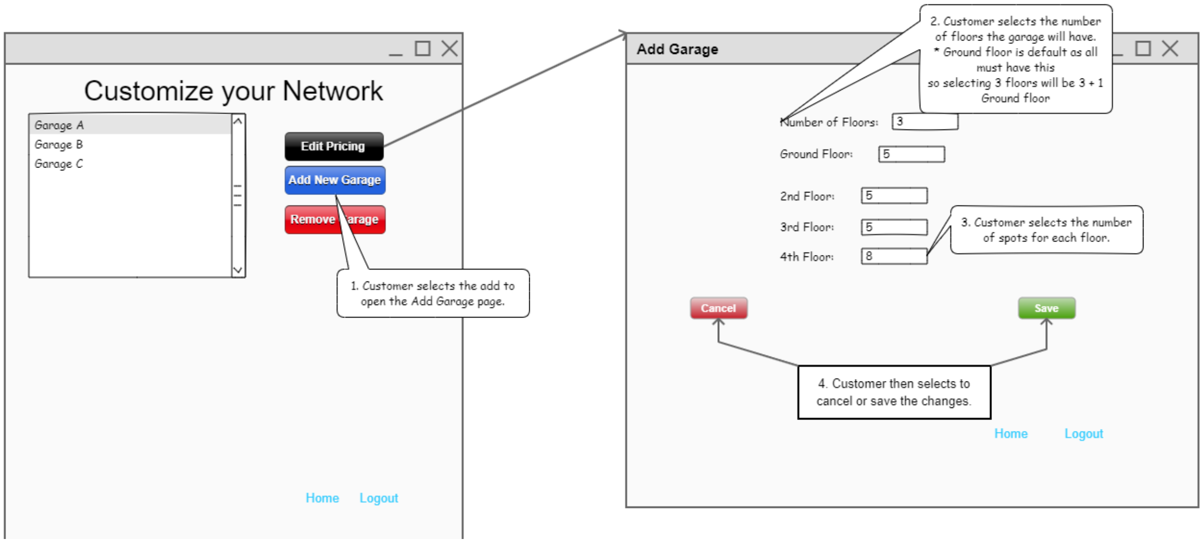
10 : 30 AM

End date: 07 / 28 / 2018

01 : 30 PM

5. User will be able to select to sign in to begin reserving

UC-15 — Adding a Garage to a Customer’s Network



Customize your Network

Garage A
Garage B
Garage C

1. Customer selects the add to open the Add Garage page.

Add Garage

2. Customer selects the number of floors the garage will have.
* Ground floor is default as all must have this
so selecting 3 floors will be 3 + 1 Ground floor

Number of Floors: 3

Ground Floor: 5

2nd Floor: 5

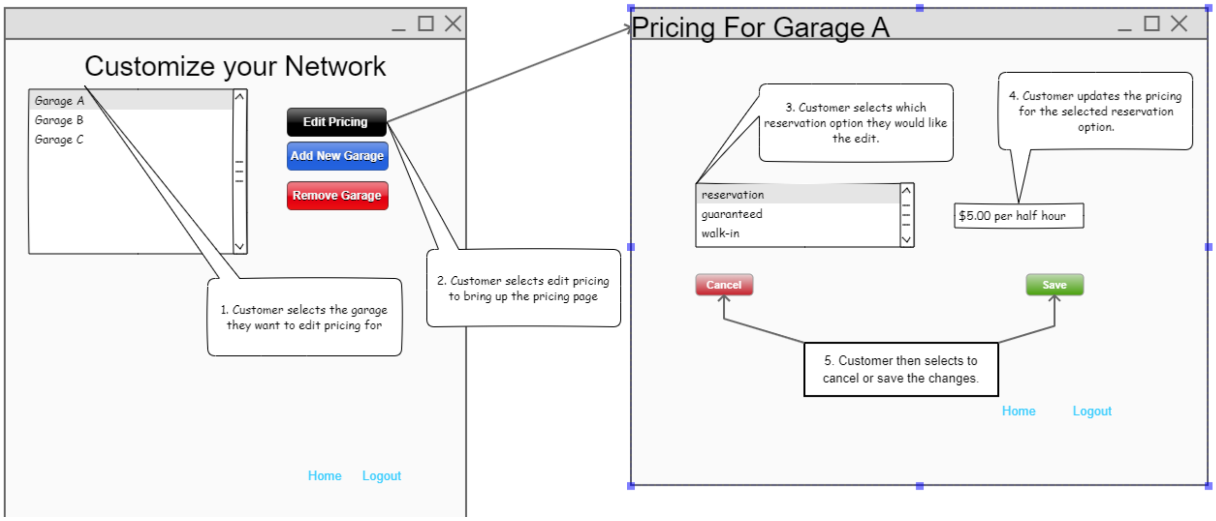
3rd Floor: 5

4th Floor: 8

3. Customer selects the number of spots for each floor.

4. Customer then selects to cancel or save the changes.

UC-16 — Adjusting Prices for a Garage



1. Customer selects the garage they want to edit pricing for

2. Customer selects edit pricing to bring up the pricing page

3. Customer selects which reservation option they would like the edit.

4. Customer updates the pricing for the selected reservation option.

5. Customer then selects to cancel or save the changes.

4.2 - User Effort Estimation

UC-02a — Requesting a Single Parking Spot

1. **NAVIGATION:** Total 1 mouse click as follows

   — after completing data entry as shown below —

   1. Click ‘Reserve’
2. **DATA ENTRY:** Total 3 mouse clicks and 33 keystrokes, as follows
   1. Click on the location box
   2. Type the name of the location, “Atlanta, GA”
   3. Press the “Tab” key to move to the start date field
   4. Type “7/22/2018”
   5. Press the “Tab” key to move to the start time field
   6. Click on “10:30 AM”
   7. Press the “Tab” key to move to the end date field
   8. Type “7/22/2018”
   9. Press the “Tab” key to move to the end time field
   10. Click on “1:00 PM”

UC-02b — Requesting a Permanent/Guaranteed Spot

1. **NAVIGATION:** Total 1 mouse click as follows

   — after completing data entry as shown below —

   1. Click ‘Reserve’
2. **DATA ENTRY:** Total 3 mouse clicks and 33 keystrokes, as follows
   1. Click on the location box
   2. Type the name of the location, “Atlanta, GA”
   3. Press the “Tab” key to move to the start date field
   4. Type “7/22/2018”
   5. Press the “Tab” key to move to the start time field
   6. Click on “10:30 AM”
   7. Press the “Tab” key to move to the end date field
   8. Type “7/28/2018”
   9. Press the “Tab” key to move to the end time field
   10. Click on “1:30 PM”

UC-15 — Adding a Garage to a Customer’s Network

1. **NAVIGATION:** Total 2 mouse clicks as follows
   1. Click “Add New Garage”

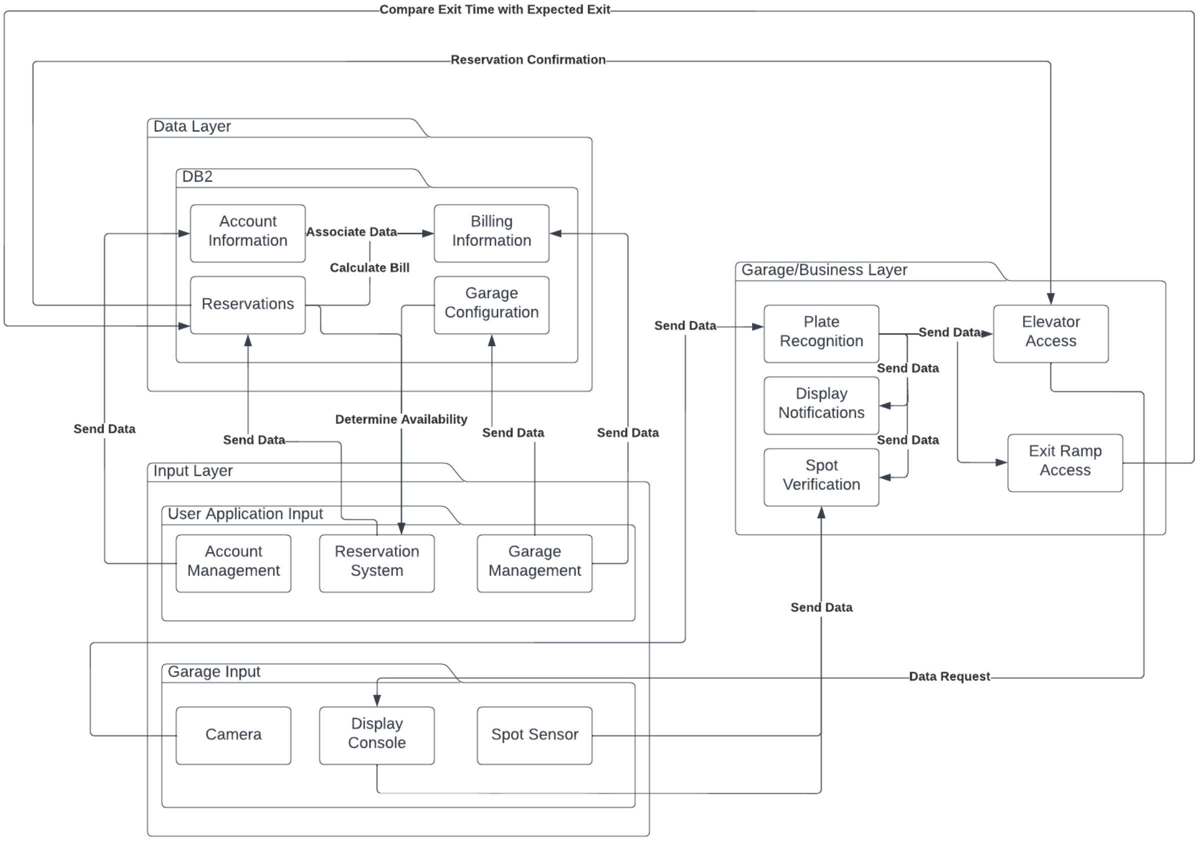
      — after completing data entry as shown below —
   2. Click “Save” to save the changes, or “Cancel” to exit without saving the changes
2. **DATA ENTRY:** Total 1 mouse click and 9 keystrokes, as follows
   1. Click on the “Number of Floors” field
   2. Enter “3”
   3. Press the “Tab” key to move to the “Ground Floor” field
   4. Enter “5” as the number of spaces on that floor
   5. Press the “Tab” key to move to the “2nd Floor” field
   6. Enter “5” as the number of spaces on that floor
   7. Press the “Tab” key to move to the “3rd Floor” field
   8. Enter “5” as the number of spaces on that floor
   9. Press the “Tab” key to move to the “4th Floor” field
   10. Enter “8” as the number of spaces on that floor

UC-16 — Adjusting Prices for a Garage

1. **NAVIGATION:** Total 4 mouse clicks as follows
    1. Click on a garage name from the list of existing garages
    2. Click the “Edit Pricing” button
    3. Click on a reservation option in the list of pricing types
       — *after completing data entry as shown below* —
    4. Click “Save” to save the changes, or “Cancel” to exit without saving the changes
2. **DATA ENTRY:** Total 2 mouse clicks and 5 keystrokes, as follows
    1. Click on the dollar amount field
    2. Click “half-hour” as the time period for pricing
    3. Enter “5.00”

5 - System Architecture

5.1 - Identifying Subsystems



As shown in the diagram, our system consists of 3 subsystems: the input layer, the data layer, and the Business (Garage) layer. Within the input layer, there are inputs done on the application such as Account Management information, Reservation Information and Garage configuration information. Additional inputs are done in the garage itself upon arrival or departure. These include inputs from automated systems such as the sensors, the hardware and inputs done on the display console.

In the data layer lies the database. This will be a DB2 Database hosted on IBM Cloud. The database will maintain all records for account information, reservations, garage configuration and more. The input layer and the business layer will send and receive data from the database as needed throughout the usage of the application.

The Garage/Business Layer consists of the actions taken within the garage when a user arrives. This includes considering inputs by the ‘Garage Input’ piece of the Input Layer, and data from the database that was initially sent via the ‘User Application Input’ piece of the input later. This is where many business rules will exist for processing the parking of the users.

5.2 - Architecture Styles

5.2.1 - Software Architecture

Our Software will follow Model-View-Controller Architecture. The Model will consist of our business logic and all the data. Thus, including the DB2 database and constraints on the database. This will hold user data, garage configuration, reservations, and more. This data from the Model is reflected in the View, also known as the UI. The view handles all visual components and the layout of the application. The user will input data into the view to be handled by the Controller and checked for requirements and other business logic before updating the database. In some cases, the controller will update the view immediately to reflect errors in the entered data based on the defined business logic and requirements (MDN).

5.2.2 - Network Architecture

The network architecture is going to be a Client-Server architecture. In this scenario, we have two separate clients: Both the user’s device (computer, phone, tablet) and the console on-site at each garage. Both of these clients will communicate with the server the software is hosted on. Since the customer may choose their implementation method, the server may be hosted on-premises by the client’s internal IT team, or on the cloud by us or a third-party vendor.

5.3 - Mapping Subsystems to Hardware

Our system runs on multiple machines accessing a database concurrently. Users and managers must be able to access the service via individual computers (clients). The system (server) is hosted either by the customer or through a managed service. The database is cloud-based and hosted offsite.

5.3.1 Garage Input Hardware

Garage input hardware includes the camera and spot sensor. The camera facilitates license plate recognition as a form of reservation verification against reservation records in the database. The spot sensor identifies occupied spaces. This is also communicated to the database through the garage/business layer. Occupancy is then updated and stored in the database based on spot sensor data.

5.3.2 Garage/Business Hardware

The garage hardware includes the elevator and exit control arm. The elevator provides garage access when the reservation information is confirmed, either by plate recognition or user reservation number. The plate reader at the exit reads the vehicle's license plate and updatest the exit time, the exit arm then lifts allowing the vehicle to exit the lot. This information is processed in the database to establish billing details.

5.4 - Connectors and Network Protocols

For this project, we have chosen to use the HTTP protocol combo of NodeJs with Express routing. NodeJs was chosen specifically for its familiarity with a number of individuals in the group, as well as its low barrier to entry, zero cost, and extensive documentation. Express routing will be combined with our Node environment. Express (ExpressJs) was chosen to pair with NodeJs due to its vast choices of HTTP utility methods, as well as the very easy-to-use and well-documented API (https://expressjs.com/en/4x/api.html). Another reason ExpressJs was chosen as a partner with NodeJs is that Express routing is much faster and more robust than the majority of NodeJs' built-in server protocols. In a great writeup by Hiren Dhaduk, he goes into great detail as to why Express is so widely used with NodeJs. During Dhaduk's explanation he points out ease of use, and response time making Express superior to NodeJs, especially if your application has more than one page, which this project does. (Dhaduk, 2022).

Other web frameworks do exist that rival ExpressJs, such as Fastify. Fastify touts itself as even faster than ExpressJs. However, Fastify is much newer than Express, and it is far less adopted. Checking the npm website, Fastify has 710,000 downloads a week (Fastify.io, 2022), while Express pushes up over 25 million downloads per week (ExpressJS, 2022). This larger adoption of Express means it is much more well documented than Fastify. While Fastify claims to be faster than Express, tests are usually done by websites or blogs with a bias toward Fastify. This makes the results somewhat unclear or unreliable.

All the factors above made the combo of NodeJs and Express the ideal choice for our group and project.

5.5 - Global Control Flow

5.5.1 Execution Orderness

Execution is both procedural and event-driven. Reservations are procedural in that the user always creates or modifies a reservation following a sequence of steps. Reservation confirmation and occupancy data are event-driven. Reservation confirmation is initiated when the license plate reader recognizes and validates the plate information. This triggers the elevator. When the spot sensor is triggered, garage occupancy is updated in the database. When a vehicle leaves, the spot sensor is deactivated releasing the spot and making it available again. Billing is event-driven because the billable time is locked in once the vehicle exits the garage. If a vehicle leaves after their original reservation time, the exit time is adjusted to correctly bill the user.

5.5.2 Time Dependency

We have a real-time system. Reservations are based on a start and end time. Billing is calculated based on start and time. Real-time is also necessary to calculate overstays to correctly bill users. There are no limits to reservation windows.

5.6 - Hardware Requirements

Hardware requirements may be broken into several categories:

- **User hardware**
    - The device used by each user to access the service
- **Garage management terminal**
    - The device used by the customer to administer their garage network
- **Garage external informational display**
    - The display outside the garage for displaying available spaces
- **Garage elevator informational display**
    - The display inside the car elevator for communication with users
- **Entrance and exit terminal displays**
    - The terminals used to display and enter reservation details to enter or exit the garage if the system fails to match the license plate to a reservation
- **System server**
    - The server running the garage software system

Other key hardware, including the car elevator, spot occupancy sensors, exit gate arm, and license plate cameras will be sourced from Commercial Off-The-Shelf (COTS) parking-garage hardware vendors.

Minimum Requirements	
User Device	
Display	Color; 360px * 640px viewport size
Network	4G mobile data or 10 Mbps
Software	An up-to-date install of one of Google Chrome, Mozilla Firefox, Microsoft Edge, or Apple Safari
Garage Management Terminal	
Display	Color; 768px * 1024px viewport size
Network	4G mobile data or 10 Mbps
Software	An up-to-date install of one of Google Chrome, Mozilla Firefox, Microsoft Edge, or Apple Safari
Garage External Informational Display	
Brightness	5,000 nits
Pixel Pitch	To be determined per installation based on expected viewing distance
Size	
Garage Elevator Informational Display	
Brightness	2,500 nits
Pixel Pitch	To be determined per installation based on expected viewing distance
Size	
Entrance and Exit Terminal Displays	
Input	Touchscreen
Size	20 inches

Resolution	Full HD
System Server	
Storage:	500 GB SSD
RAM:	4GB
Network:	100 Mbps

6 - Project Size Estimation

6.1 - Unadjusted Actor Weight (UAW)

Actor	Description	Complexity	Weight
Garage Operator	Interacts with the system by managing pricing and garage configuration and has the capability to view garage statistics dashboard	Average	2
User	Interacts with the system by entering user information, searching for spots, making reservations, receiving and managing autopayments, triggers garage system, enters information on display console and can view garage availability	Complex	3

Database	Holds and manages Data entered into the application	Simple	1
Camera	Reads license plate and searches for memberships/reservations to then find appropriate parking spot	Average	2
Parking Sensors	Used to detect current occupancy and determine if a user parks in the correct spot	Average	2
Display Console	Used to display and receive client information. Such as reading member numbers, reservation numbers, displaying spot for user to park, notifications on current availability, suggesting other garages if no availability	Complex	3
Total	1 x Simple (1) + 3 x Average (2) + 2 x Complex (3) = 13		

6.2 - Unadjusted Use Case Weight (UUCW)

UC	Description	Category	Weight
UC-01	User Registration	Average	10
UC-02a	Parking Space Requests (Single)	Complex	15
UC-02b	Parking Space Request (Permanent/Guaranteed)	Complex	15

UC-03	Reservation Arrival (Grace Period)	Simple	5
UC-04	No-Shows/Late Arrivals	Average	10
UC-05	Understays	Average	10
UC-06	Overstays	Average	10
UC-07	Reservation Modification	Average	10
UC-08	Billing	Simple	5
UC-09	Account Management	Simple	5
Garage Access Control			
UC-10a	Reservation Arrivals (Garage)	Complex	15
UC-10b	Walk-in Arrivals	Complex	15
UC-11	Reservation Management	Average	10
UC-12a	Garage Access	Complex	15
UC-12b	Request to Leave	Average	10
Monitoring Occupancy and Space Assignment			
UC-13	Monitoring Occupancy	Average	10
Statistical Data Collection			
UC-14	Garage Statistics	Simple	5
System Administration			
UC-15	Garage Configuration	Simple	5
UC-16	Price Adjustments	Average	10
UC-17	Upgrading Accounts	Simple	5
Total	6 x Simple (5) + 9 x Average (10) + 5 x Complex (15) = 195		

6.3 - Unadjusted Use Case Points (UUCP)

UUCP = *UAW* + *UUCW*

UUCP = *UAW* (13) + *UUCW* (195) = 208

6.4 - Technical Complexity Factors (TCF)

TF	Description	Weight	Perceived Complexity	Calculated Factor
T1	Distributed System	2	3	2x3=6
T2	Performance Objectives	1	3	1x3=3
T3	End-User Efficiency	1	3	1x3=3
T4	Complex Internal Processing	1	4	1x4=4
T5	Reusable Design or Code	1	1	1x1=1
T6	Easy to Install	0.5	2	0.5x2=1
T7	Easy to Use	0.5	4	0.5x4=2
T8	Portable	2	3	2x3=6
T9	Easy to Change	1	3	1x3=3
T10	Concurrent to Use	1	1	1x1=1
T11	Special Security Features	1	3	1x3=3
T12	Provides Direct Access for Third Parties	1	1	1x1=1
T13	Special User Training Facilities are Required	1	0	1x0=0
			Total:	34
TCF= 0.6+ (0.01 x 34) = 0.94				

6.5 - Use Case Points (UCP)

UCP = UUCP x TCF x ECF

UCP= 208 x 0.94 x 1 = 195.52

7 - Plan of Work

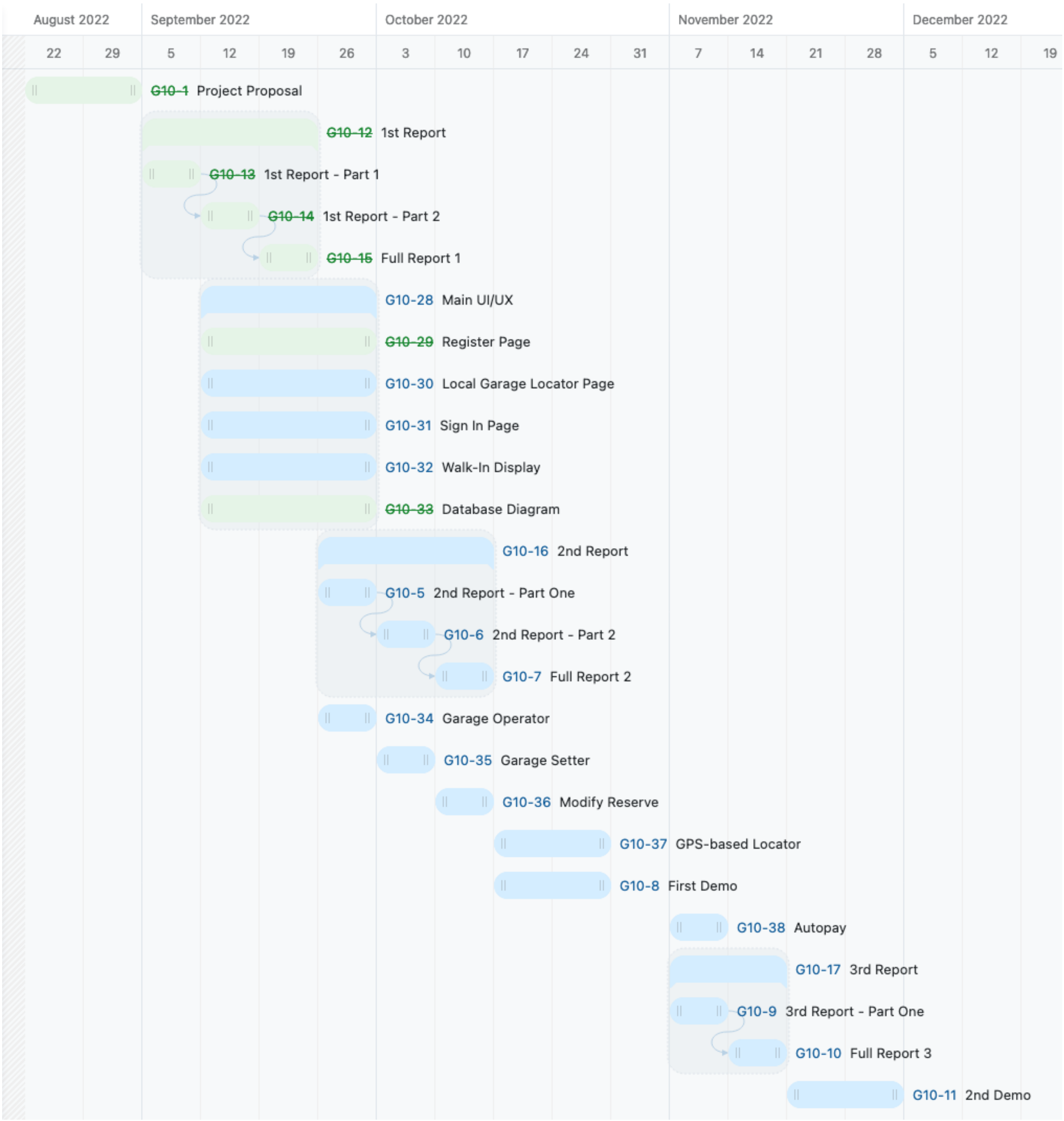
Plan of Work and Breakdown of Responsibilities

Project Target	Tasks	Description	Contributors	End Date	Technologies
Main UI/UX	Register Page	User's will be able to register with designated information.	Justin Henley, Victor Munoz	9/12/22 - 10/02/22	***UI/UX*** HTML, CSS, Bootstrap4, Javascript, AngularJs ***Backend*** C#,Express, NodeJs, DB(?)
	Local Garage Locator Page	Starting page, user able to locate nearby garages and select one for booking	Dustin Threet, Hannah Carr	9/12/22 - 10/02/22	
	Sign In Page	Login Page	Justin Henley, Victor Munoz	9/12/22 - 10/02/22	
	Walk-In Display	Display for walk-ins with available spots for them.	Dustin Threet, Justin Henley	9/12/22 - 10/02/22	
	Database Diagram	Basic structure for entities that will be stored in the DB	Hannah Carr	9/12/22 - 10/02/22	DB (?)
1st Report : Part 2	Functional Requirements		Justin Henley, Victor	9/18/22	

			Munoz, Hannah Carr, Dustin Threet		N/A
Full Report 1			Justin Henley, Victor Munoz, Hannah Carr, Dustin Threet	9/25/22	N/A
Garage Operator	Garage Operator After-Sign In	Owner user will be directed here after sign-in	Justin Henley, Victor Munoz, Hannah Carr, Dustin Threet	10/2/22	***UI/UX*** HTML, CSS, Bootstrap4, Javascript, AngularJs
2nd Report: Part 1	Use cases/ User interactions		Justin Henley, Victor Munoz, Hannah Carr, Dustin Threet	10/2/22	N/A
Garage Setter	Add new garage	Owner will be able to add a garage to network	Justin Henley, Victor Munoz, Hannah Carr, Dustin Threet	10/9/22	***Backend*** C#,Express, NodeJs, DB(?
2nd Report: Part 2	System architecture		Justin Henley, Victor Munoz, Hannah Carr, Dustin Threet	10/9/22	N/A
Modify Reserve	Alter reservations	User will be able to change, cancel, and extend reservations	Justin Henley, Victor Munoz, Hannah Carr, Dustin Threet	10/16/22	***UI/UX*** HTML, CSS, Bootstrap4, Javascript, AngularJs ***Backend*** C#,Express, NodeJs, DB(?
Full Report 2			Justin Henley, Victor Munoz, Hannah Carr, Dustin Threet	10/16/22	N/A
GPS-based Locator	Find garage	Locate garage based on user input	Justin Henley, Victor Munoz, Hannah Carr, Dustin Threet	10/28/22	***UI/UX*** HTML, CSS, Bootstrap4, Javascript, AngularJs

					Backend C#,Express, NodeJs, DB(?)
First Demo		Combine project to present		10/30/22	Project
Autopay	Alter autopay	See user's payment history, change payment options/ occurrence	Justin Henley, Victor Munoz, Hannah Carr, Dustin Threet	11/12/22	***Backend*** C#,Express, NodeJs, DB(?)
3rd Report: Part 1			Justin Henley, Victor Munoz, Hannah Carr, Dustin Threet	11/13/22	N/A
Full Report 3			Justin Henley, Victor Munoz, Hannah Carr, Dustin Threet	11/20/22	N/A
Second Demo		Combine project to present		12/24/22	Project

Gantt Chart


August 2022
September 2022
October 2022
November 2022
December 2022
22
29
5
12
19
26
3
10
17
24
31
7
14
21
28
5
12
19
~~G10-1~~ Project Proposal
~~G10-12~~ 1st Report
~~G10-13~~ 1st Report - Part 1
~~G10-14~~ 1st Report - Part 2
~~G10-15~~ Full Report 1
G10-28 Main UI/UX
~~G10-29~~ Register Page
G10-30 Local Garage Locator Page
G10-31 Sign In Page
G10-32 Walk-In Display
~~G10-33~~ Database Diagram
G10-16 2nd Report
G10-5 2nd Report - Part One
G10-6 2nd Report - Part 2
G10-7 Full Report 2
G10-34 Garage Operator
G10-35 Garage Setter
G10-36 Modify Reserve
G10-37 GPS-based Locator
G10-8 First Demo
G10-38 Autopay
G10-17 3rd Report
G10-9 3rd Report - Part One
G10-10 Full Report 3
G10-11 2nd Demo

Product Ownership

The project can be broken down into six specific modules, each with its own specifications. With the modules specified, ownership is assigned cleanly per module, with the responsibility for larger modules being shared by a pair of teammates.

Module-1: User Interaction

Product Owners: Dustin Threet, Justin Henley

Functional Requirements: REQ-01 through REQ-25

Nonfunctional Requirements: REQ-73 through REQ-78, REQ-81, REQ-84 through REQ-86

User Interface Requirements: REQ-87 through REQ-89, REQ-91 through REQ-93

Description: Allowing the user to reserve online, or as a walk in. Define users' ability to make reservations, adjust reservations, and adjust payment options.

Module-2: Garage Access Control

Product Owners: Hannah Carr, Victor Munoz

Functional Requirements: REQ-26 through REQ-46

Nonfunctional Requirements: REQ-68 through REQ-70, REQ-79, REQ-80, REQ-82, REQ-83

Description: Rules and parameters setup based on if the reservation is made as a Walk-In, or online reservation. Some privileges pertaining to the online registered users will not be carried over to the Walk-in users.

Module-3: Monitoring of Occupancy and Space Reassignment

Product Owner: Justin Henley

Functional Requirements: REQ-47 through REQ-53

Nonfunctional Requirements: REQ-71, REQ-72

Description: Tracking the current number of available parking spots, then displaying this for the user's reservation options, and stat tracking for owners.

Module-4: Simulation of Arrivals and Departures

Product Owner: Dustin Threet

Functional Requirements: REQ-54 through REQ-58

Description: Admin having full ability to replicate the full parking process: account creation, reserving, parking, leaving.

Module-5: Statistical Data Collection

Product Owner: Hannah Carr

Functional Requirements: REQ-59 through REQ61

Description: Create data that will be plugged into a dashboard, that will better allow an owner to view the flow of traffic in the selected garage.

Module-6: System Administration

Product Owner: Victor Munoz

Functional Requirements: REQ-62 through REQ-67

User Interface Requirements: REQ-90

Description: Allowing Owners to add/ edit garages in their network, including adding new floors to an existing garage, add a new garage, remove a garage from their network.

8 - References

Dhaduk, H., 2022. Node.js vs. express: Determining the best backend technology. Simform. Available at: https://www.simform.com/blog/node-js-vs-express/ [Accessed September 22, 2022].

ExpressJS, 2022. Express. npm. Available at: https://www.npmjs.com/package/express [Accessed September 23, 2022].

Fastify.io, 2022. Fastify. npm. Available at: https://www.npmjs.com/package/fastify [Accessed September 23, 2022].

Hoehne, C.G. et al., 2019. Valley of the sun-drenched parking space: The growth, extent, and implications of parking infrastructure in Phoenix. *Cities*, 89, pp.186–198. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0264275118311636 [Accessed September 6, 2022].

Kazman, R. & Bass, L., 2005. Categorizing business goals for software architectures. *Software Engineering Institute, Carnegie Mellon University*. Available at: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=7751 [Accessed September 6, 2022].

MVC, MDN - Web Docs. Available at: https://developer.mozilla.org/en-US/docs/Glossary/MVC [Accessed September 24, 2022]

Parkopedia, *Parkopedia.com* Available at: https://en.parkopedia.com/ [Accessed September 3, 2022].

SP Plus Corporation, *Parking.com*. Available at: https://parking.com/ [Accessed September 3, 2022].

SpotHero, *SpotHero.com*. Available at: https://spothero.com/ [Accessed September 3, 2022].