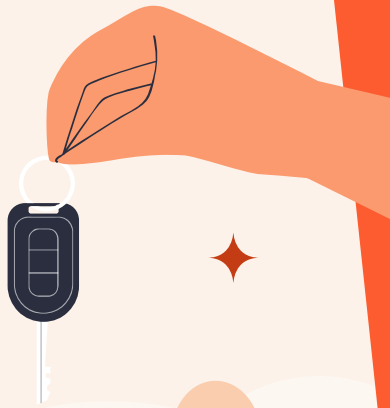
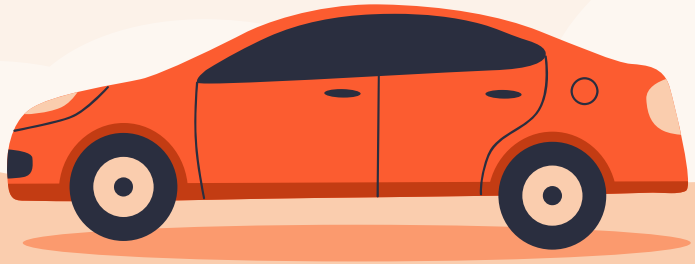
Garage Man

By: Krimika K., Aaron V., Giovanni Q., Andrew D., Aster L.





Developers

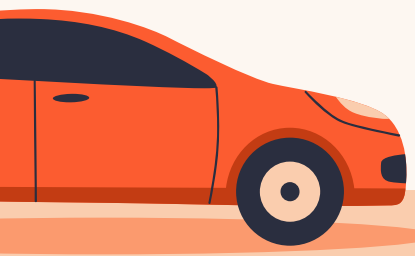
Giovanni
- Team manager, Documentation Writer, Programmer

Krimika
- Documentation Writer, Programmer, Quality Assurance, Tester

Aaron
- Programmer, Quality Assurance, Tester

Andrew
- Programmer, Quality Assurance, Tester

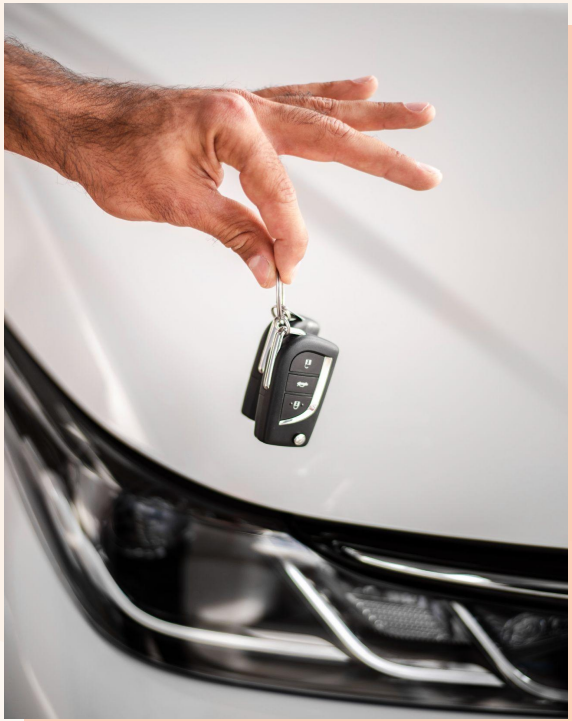
Aster
- Documentation Writer, Architect



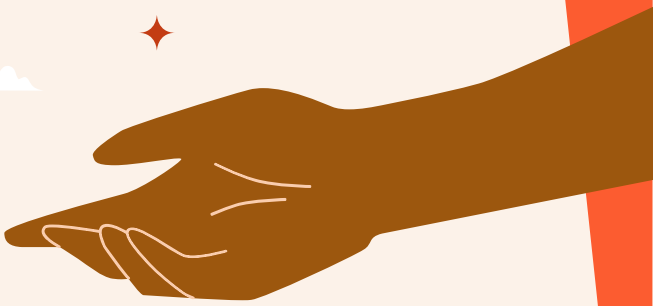
Garage Man

Automated user-centered and ticket-based parking management system

Allows reservation and on-arrival parking



01



User and System Requirements

Manage Online Reservations for Parking

- **How is reservations made?**
  - Allows users to reserve a spot through the website or check-in upon arrival.
    - The automated system issues a virtual ticket if there is capacity.
- **How are reservations managed?**
  - The reservation fee charged only after the reservation is complete, not upfront.
  - When a user selects "Reserve," the system sends a list of current reservations
  - Upon successful reservation, the system notes the reservation and remembers it.
  - If reservation is cancelled by user, user will receive a refund for the reservation cost, minus the reservation fee.
  - Users can see the start time and end time of their reservations on the Ticket page, whether the reservation is active or not.
- **How do we limit reservations?**
  - Garage Man will hold a car for a **maximum of 3 days** (72 hours) before attempting to warn the customer via automated SMS, or phone call. If the car remains within the next 12 hours, the car will be towed and they will incur a penalty on top of their subtotal.

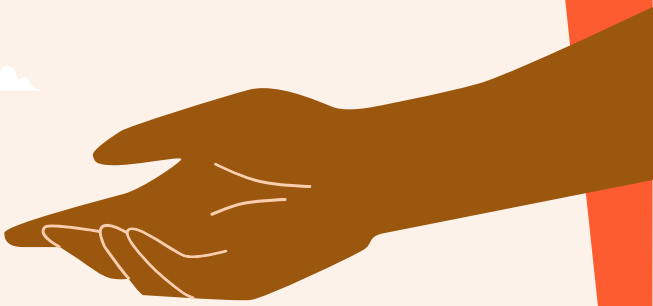
Functional Requirements

- Users can sign up using Google authentication and input their vehicle information.
- Signed-in users can generate a ticket if there is a vacancy and no active ticket.
- Users can see their active ticket information, reservation start time, and active reservation end time on the Home page.
- Users with active tickets can close their ticket to end their parking session and will be charged according to rates and policies.
- Users can make reservations for a later time, which are paid in advance with an additional fee.
- Users can view past tickets and reservations in the Tickets tab
- Users will be notified for payment via email with a receipt and parking policies.
- Check-in logic updates vacancy and returns an error if there are no vacancies.
- Successful check-outs charge the user and email them a receipt.
- User Main Pages show up-to-date vacancy, refresh at a predetermined interval, and update whenever vacancy is updated.

Non-functional Requirements

- **Product Requirement**
  - Will be compatible with any mobile device or PC with an internet connection and access to any internet browser.
  - Each function should take no longer than 5 seconds to be performed.
  - Private customer data will be encrypted within our firebase database, such as license plate numbers, emails, and passwords.
- **Organizational Requirement**
  - The entire project will revolve around React JS and Firebase Database
    - In order to store the user's information Firebase would be used to retrieve, delete and store data.
  - Accounts will be limited to Google due to their uniqueness and ease of use.
- **External Requirement**
  - Disclosing any personal information to other users is prohibited unless it's their email address, name, and ticket information.
  - The website will be available in US-English, currency will be in US-dollars and timezones will be modified according to user's location.

02

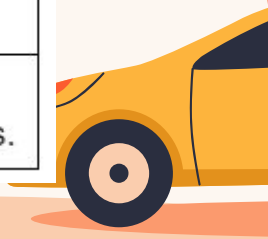


Use Cases

Use Case #1

Create User Account

Identifier	UC-001: Create User Account
Purpose	Allow customers to have an account with all of their information and history with the web app.
Requirements	-Data is stored efficiently and accurately - verify the customer's credentials and grant access to the customer's account information.
Development Risks	- data integrity - data security
Pre-conditions	-customer has a google account -customer has a vehicle
Post-Conditions	The customer is logged in to their account and can access their account information, view their parking history, and make reservations.



Use Cases #1.1

Create User Account

Typical Course of Action:

Step	Action	System's Response
1	The customer clicks "Sign in with google"	
2		The system opens a window for the user to select their google account
3	The customer selects their google account	
4		The system checks if the user exists in the data base
5		The system takes the user to the Create account page
6	The customer puts their information in the form	
7		The system pushes their information to the database and redirects them to the home page

Use Case #1.2

Create User Account

Alternate Course of Action:

Step	Action	System's Response
1	The customer clicks "Sign in with google"	
2		The system opens a window for the user to select their google account
3	The customer selects their google account	
4		The system checks if the user exists in the data base
5		The system takes the user to the Create account page
6	The customer puts their information in the form	
7	The customer forgets to add their vehicle's color	
8		The system displays an error and prevents user from submitting
9	The customer enters their vehicle's color	
10		The system pushes their information to the database and redirects them to the home page

Use Case #1.3

Create User Account

Exceptional Course of Action:

Step	Action	System's Response
1	The customer clicks "Sign in with google"	
2		The system opens a window for the user to select their google account
3	The customer selects their google account	
4		The system checks if the user exists in the data base
5		The system takes the user to the Create account page
6	The customer puts their information in the form	
7	The customer enters an invalid number of characters for their license plate	
8		The system displays an error and prevents user from submitting
9	The customer corrects their license plate	
10		The system pushes their information to the database and redirects them to the home page

Use Case #2

Make a Ticket

Identifier	UC-002: Make a Ticket
Purpose	Customer can make a ticket ahead of time or at the parking garage
Requirements	-If the reservation is ahead of time, the app must display available reservation times, allow customers to choose a time and confirm the reservation.
Development Risks	None
Pre-conditions	-The customer must be logged in to the app. -The customer must arrive in the vehicle registered in their account - If at the garage, there must be available spots in the lot
Post-Conditions	- The reservation is made - A ticket with the customer's reservation information is created.on the ticket page

Use Case #2.1

Make a Ticket

Typical Course of Action:

Step	Action	System's Response
1	The customer selects "Make a Reservation".	
2		The system displays the list of available parking times.
3	The customer selects the date and time of reservation.	
4	The customer confirms the reservation.	
5		The system creates a ticket with the current date, reservation date, start time, end time, and price and displays it in the ticket page

Use Case #2.2

Make a Ticket

Alternate Course of Action:

Step	Action	System's Response
1	The customer selects "Make a Ticket" while at the garage.	
2		The system displays the parking rates and policies
3	The customer tries to confirm a taken reservation.	
4		The system displays that the reservation slot is taken because the time slots havent been updated.
5	The customer chooses an open slot and confirms the reservation.	
6		The system creates a ticket with the reservation date, time, and price and displays it in the ticket page

Use Case #2.3

Make a Ticket

Exceptional Course of Action:

Step	Action	System's Response
1	The customer selects "Make a Reservation".	
2		The system displays the list of available parking spots.
3	The customer selects a parking spot.	
4	The customer selects the date and time of reservation.	
5	The customer tries to confirm the reservation.	
6		The system encounters an error while processing the reservation.
7		The system displays an error message.
8	The customer tries to make a reservation again	
9		The system creates a ticket with the current date, reservation date, start time, end time, and price and displays it in the ticket page

Use Case #3

Close Ticket

Identifier	UC-003: Close Ticket
Purpose	Allows the customer to cancel a reservation they have made
Requirements	-The customer must have an existing ticket or reservation. -The system should confirm the cancellation with the customer. -The system should automatically refund the customer's payment, if a reservation.
Development Risks	-Technical difficulties in processing refunds -Database errors when retrieving or updating reservation information.
Pre-conditions	-The customer must be logged in to their account. -The customer must have a valid ticket or reservation. -The reservation must be cancellable (i.e., within the cancellation window).
Post-Conditions	-The reservation is cancelled and removed from the customer's account. -The system sends a confirmation email to the customer. -The customer's payment, if applicable, is refunded.

Use Case #3.1

Close Ticket

Typical Course of Action:

Step	Action	System's Response
1	Customer selects the "Cancel Reservation" option	
2		System displays the customer's current reservations.
3	The customer selects their scheduled reservation	
4	Customer confirms the cancellation.	
5		System removes the reservation from the customer's account and refunds the cost.
6		System updates the parking space availability.

Use Case #3.2

Close Ticket

Alternate Course of Action:

Step	Action	System's Response
1	The customer selects "Close Ticket".	
2		System displays the customer's current ticket time, and price summary.
3	The customer confirms ticket closure	
6		System charges user for the parking duration and sends receipt with cost summary.
7		System notes the ticket as closed.
8		System updates the parking space availability.

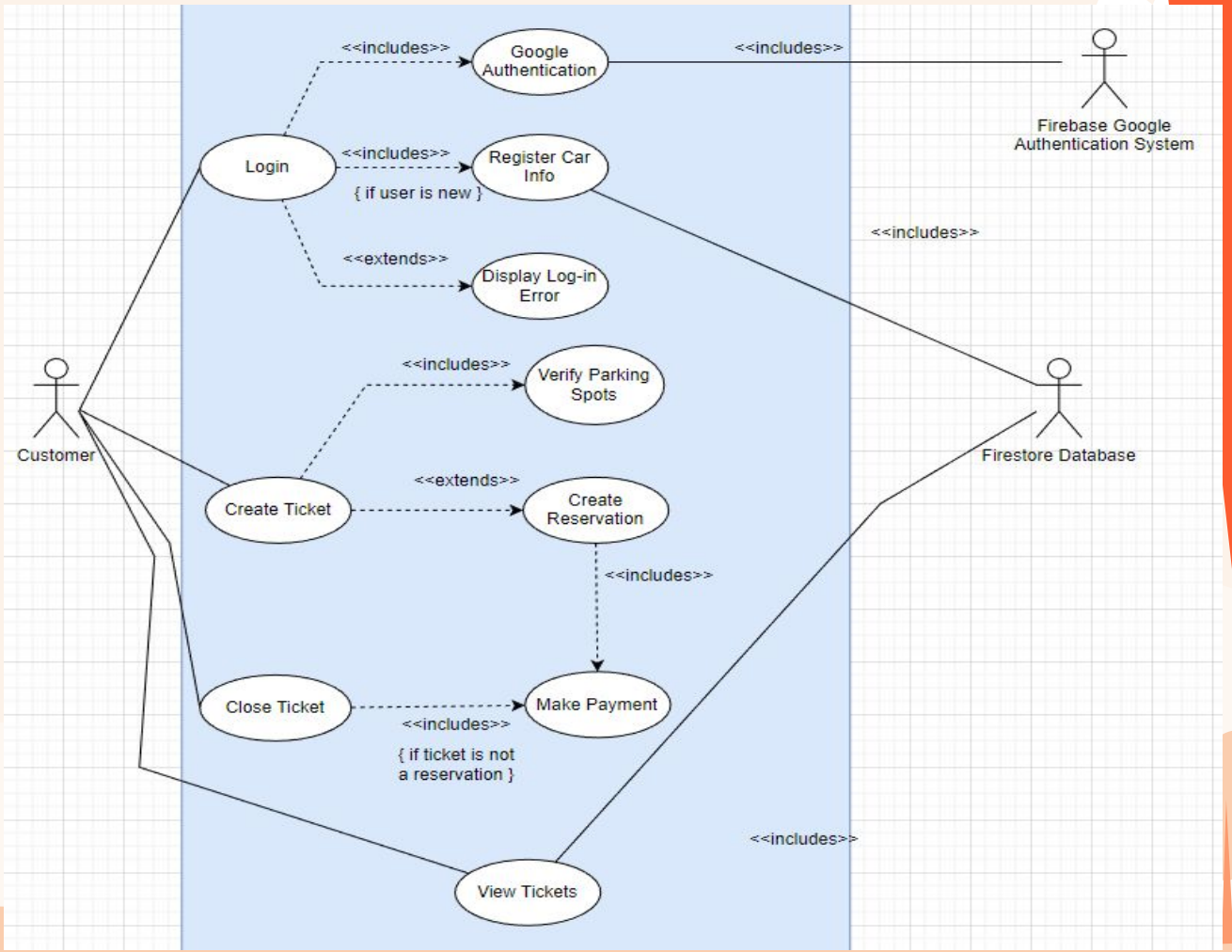
Use Case #3.3

Close Ticket

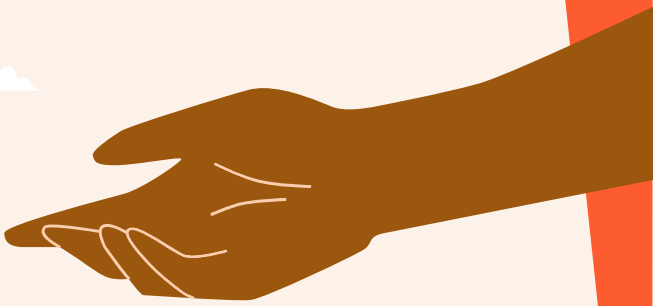
Exceptional Course of Action:

Step	Action	System's Response
1	Customer selects the "Cancel Reservation" option	
2		System displays the customer's current reservations.
3	The customer selects their scheduled reservation	
4	Customer confirms the cancellation.	
5		System is unable to process the cancellation due to technical difficulties or database errors.
6		System displays an error message and asks the customer to try again later or contact customer support.
7	The customer selects their scheduled reservation and confirms again	
8		System removes the reservation from the customer's account and deletes their ticket
9		System updates the parking space availability.

Use Case


Google Authentication
<<includes>>
<<includes>>
Firebase Google Authentication System
Login
<<includes>>
Register Car Info
{ if user is new }
<<includes>>
<<extends>>
Display Log-in Error
Customer
<<includes>>
Verify Parking Spots
Firestore Database
Create Ticket
<<extends>>
Create Reservation
<<includes>>
Close Ticket
Make Payment
<<includes>>
{ if ticket is not a reservation }
<<includes>>
View Tickets

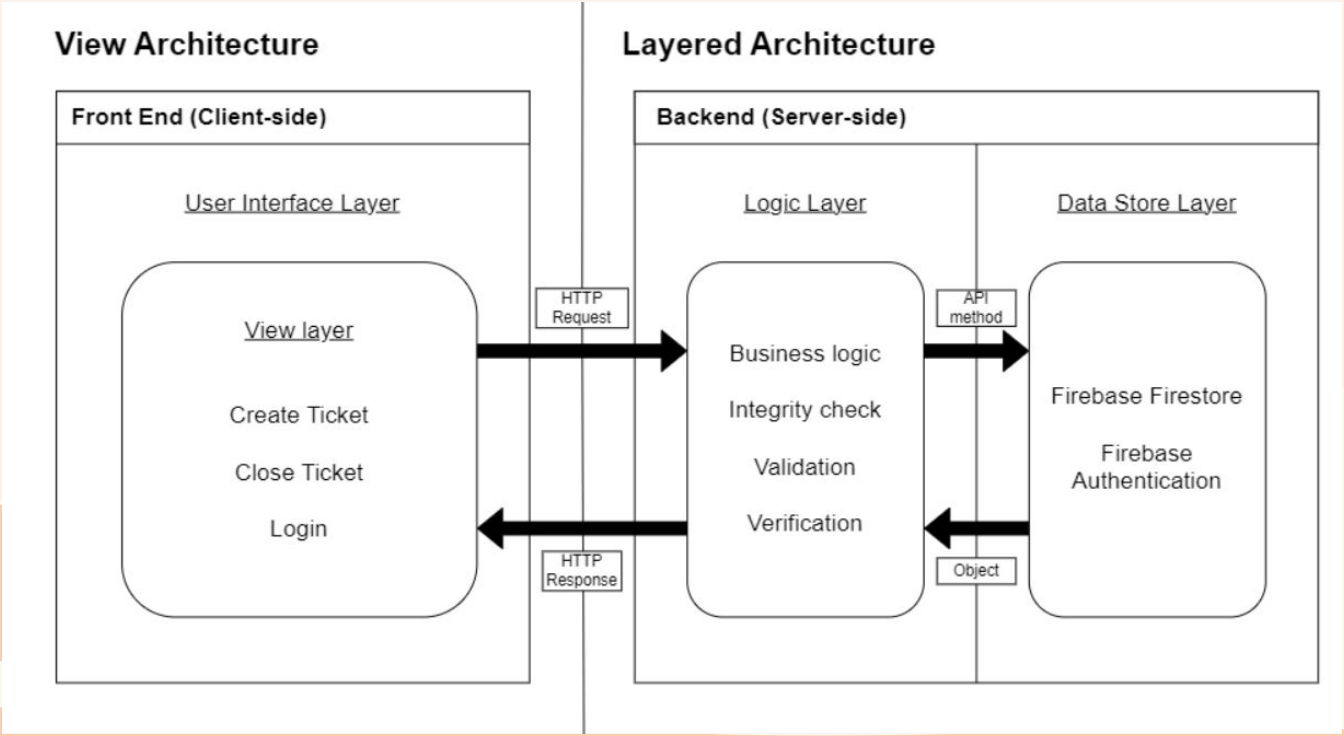
03



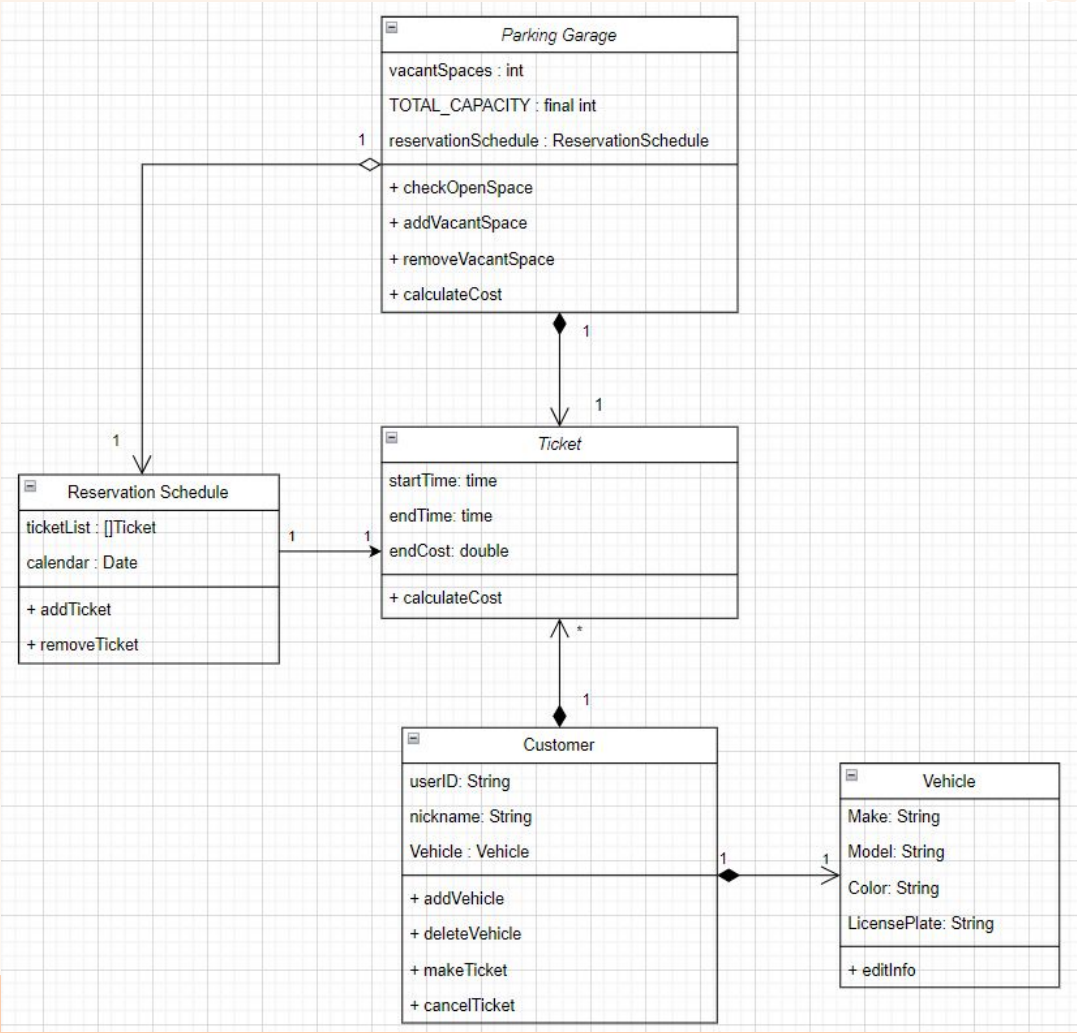
Architecture and Implementation

Architecture

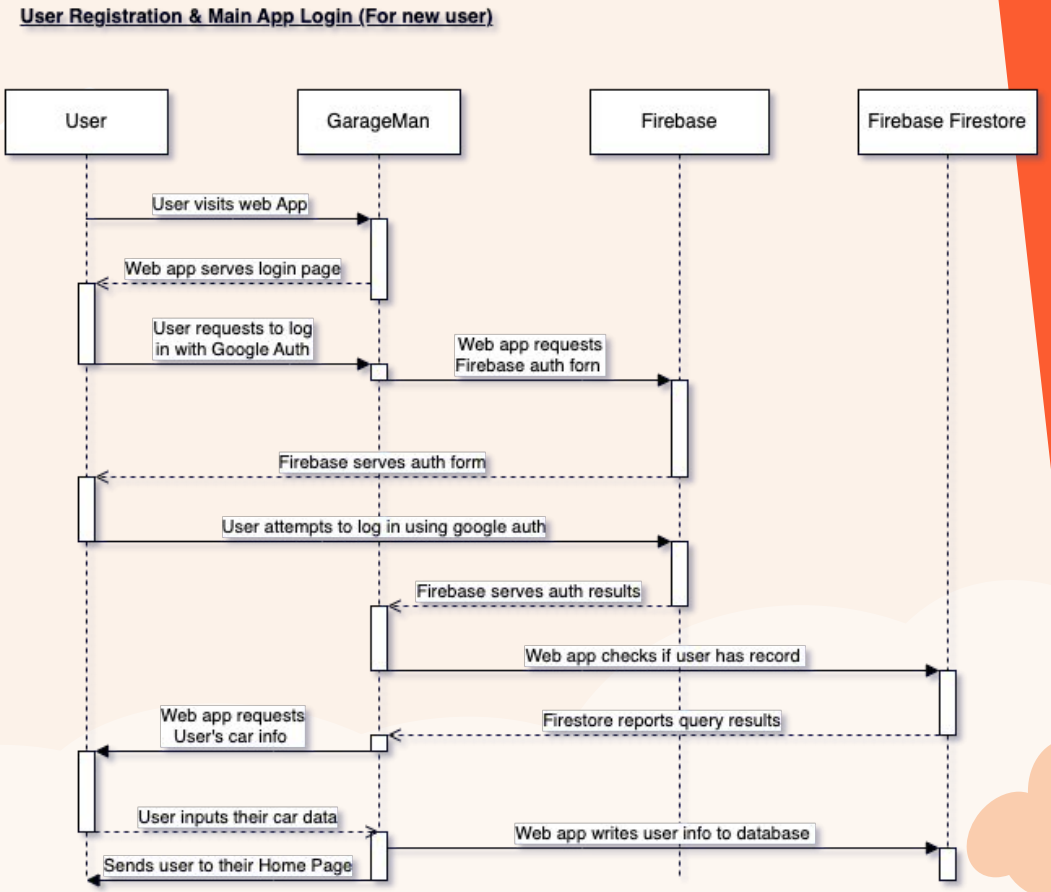
Multi-layered architecture



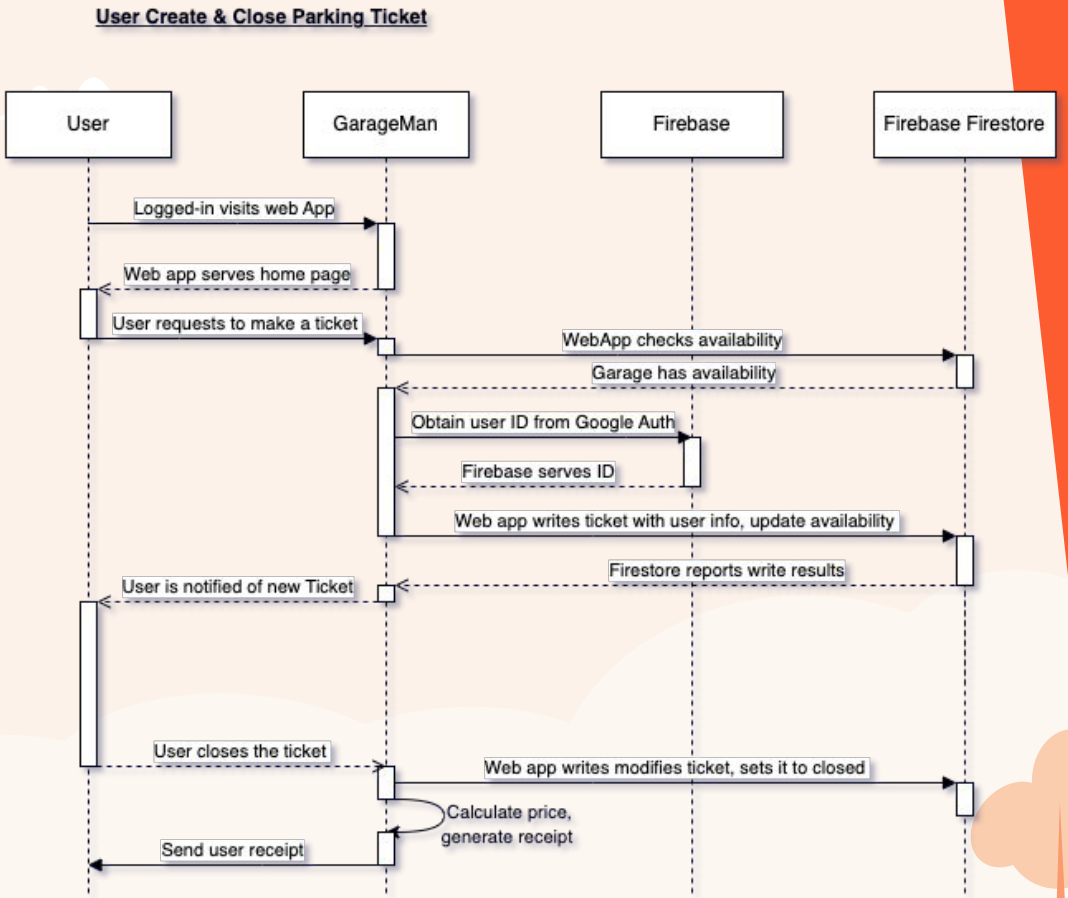
UML Diagram


Parking Garage
vacantSpaces : int
TOTAL_CAPACITY : final int
reservationSchedule : ReservationSchedule
+ checkOpenSpace
+ addVacantSpace
+ removeVacantSpace
+ calculateCost
Reservation Schedule
ticketList : []Ticket
calendar : Date
+ addTicket
+ removeTicket
Ticket
startTime: time
endTime: time
endCost: double
+ calculateCost
Customer
userID: String
nickname: String
Vehicle : Vehicle
+ addVehicle
+ deleteVehicle
+ makeTicket
+ cancelTicket
Vehicle
Make: String
Model: String
Color: String
LicensePlate: String
+ editInfo
1
1
1
1
1
1
1
1
*
1
1
1

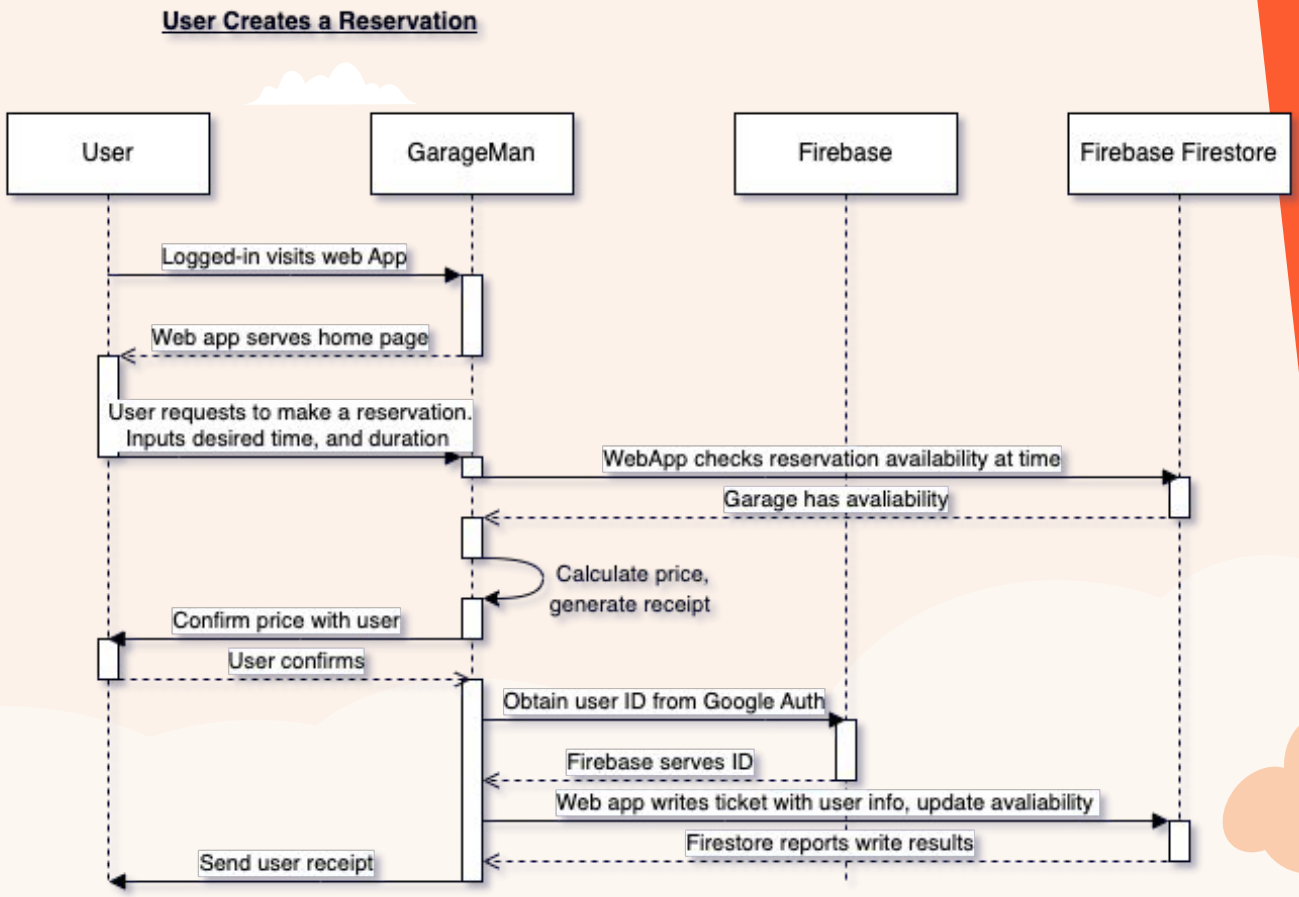
User Registration



Make Ticket
Close Ticket



Reservation Creation



Bill of Materials

- Javascript
- CSS
- React
- Firebase
- VS code/community

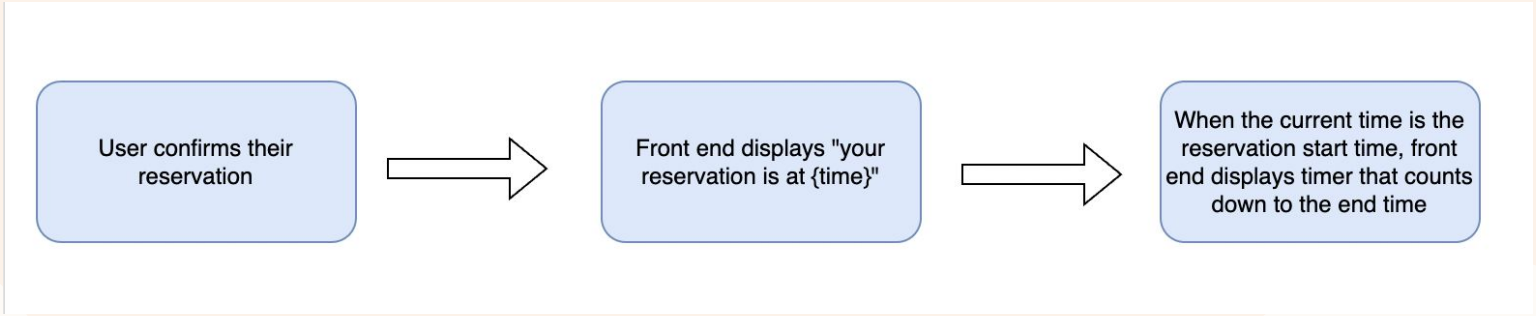
04

Demo

05

Features Not Implemented

Live Timer


User confirms their reservation
Front end displays "your reservation is at {time}"
When the current time is the reservation start time, front end displays timer that counts down to the end time

Payment Method

User enters their payment information when creating an account

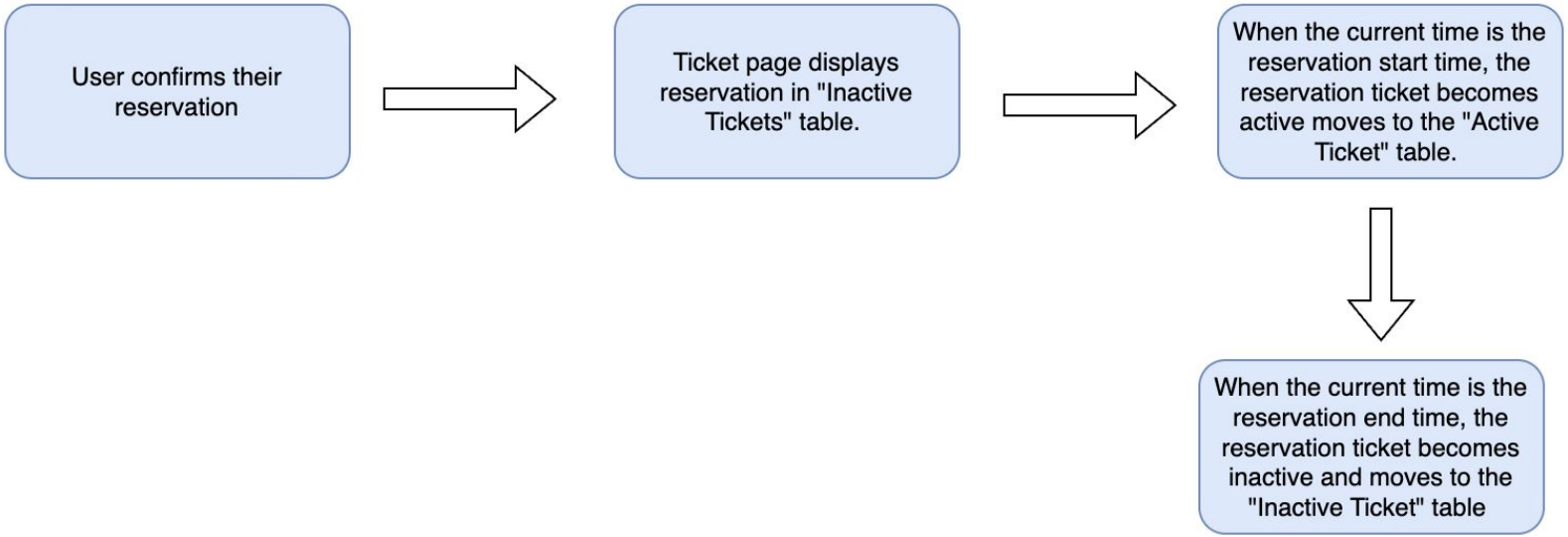
User makes a reservation and a $5 flat fee is charged

A receipt with the date, start and end time, and price of the reservation will be emailed to the user

The user's card will be charged once reservation time has ended

If the user cancels before their reservation, the flat fee will not be refunded

Reservation State Based On Time



Thank You!

This has been Garage Man!