

# Rental App

Emily Reynolds

## Table of Contents

Problem Statement – p3

Functional Requirements – p4

System Sequence Diagram – p8

Activity Diagram – p11

UI Specification – p14

Project Plan – p19

Problem statement: When people fly to new cities, they often rent vehicles. These vehicles are usually undesirable, and the end-user does not have a choice as to the specific vehicle they receive. Users are even subjected to vehicle-class changes without notice after booking due to usage. The only options that allow for users to pick out a specific vehicle are peer-to-peer.

Objectives of the system: This system should help travelers to rent a more desirable vehicle, with finite control over which exact vehicle is rented. The rental company will be able to charge higher rates for the same vehicle by giving users more control over the transaction. Specific Features:

Typical customers:

- Airport travelers
- Road trip travelers
- Rideshare operators
- Rental companies
- Travel agencies

Project planning and development approach:

1. Software:
  - a. Front-end: React
  - b. Runtime: Node.js
  - c. Back-end: Kotlin
  - d. Database: MongoDB
2. Hardware:
  - a. PC
  - b. Servers
3. Network:
  - a. No speed requirements

Specific Features:

1. End users can view and rent specific vehicles
2. Rental Companies can manage cars in the system including:
  - a. Location
  - b. Maintenance
3. Rental companies can track users including:
  - a. Insurance
  - b. Travel history
  - c. Vehicle preferences
  - d. Vehicle state at time of return (aggregate renter score)

**Emily Reynolds 2/3/23**

**Problem Statement:** When people fly to new cities, they often rent vehicles. These vehicles are usually undesirable, and the end-user does not have a choice as to the specific vehicle they receive. Users are even subjected to vehicle-class changes without notice after booking due to usage. The only options that allow for users to pick out a specific vehicle are peer-to-peer services without the trust conferred by large rental corporations.

**Glossary of Terms:**

Vehicle class – classification of vehicles including classifications like luxury car, mid-size sedan, SUV, and Pickup truck

Reservation – a claim for a specific vehicle for a specific amount of time starting at a specific location and ending at a specific location

**System Requirements:**

No.	Priority	Description
REQ-1	High	Customers can view and select vehicle inventory at their specific pickup location.
REQ-2	High	Customers can return to any location, but must be selected at time of rental
REQ-3	High	Service is available via webapp
REQ-4	High	Customer can select specific vehicle three to five days before rental. Same-day rentals are not guaranteed.
REQ-5	High	Rental company employees must be able to manage location of vehicles and details of customer rentals via an admin-panel
REQ-6	Medium	Cost is determined by pickup location, dropoff location, vehicle, and duration of rental
REQ-7	Medium	Payment must be made using credit or debit
REQ-8	Medium	Additional services are offered before checkout like refueling, toll pass, and insurance
NF-REQ-1	High	Should be able to be hosted on minimal server hardware
NF-REQ-2	Medium	Should have a 99.9% uptime
NF-REQ-3	Medium	Should have modern webapp appearance that is simple to navigate for all levels of users
NF-REQ-4	Medium	Should not visibly lag upon loading new vehicles with reasonable internet connection
NF-REQ-5	Medium	Should be easily configured and support different business models in different locations and for different clients
UI-REQ-1	High	Must have legible design with clearly defined features
UI-REQ-2	Medium	Should be attractive, at least not off-putting to look at
UI-REQ-3	Medium	UI should utilize a simple one-column design with a navbar at the top
UI-REQ-4	Medium	There should be pictures of the specific car that is being offered rather than stock images

UI-REQ-5	Medium	User should be able to rate vehicles and view previous rentals

## Rental Company App

Key Stakeholders:

- Rental company employees
- People who rent cars
- Rental company admins

Actors and goals:

- Primary actors
  - o Customer: This actor can reserve a specific vehicle from a specific location and pay for it upon receipt of the vehicle.
  - o Rental company employee: This actor can track vehicles, update history including maintenance, modify reservations, and dispense keys to customers along with receiving payment.
- Secondary actors:
  - o Rental company admin: This actor can add or decommission vehicles, set prices, add or remove store locations, and any task available to employees.
  - o System: This actor is responsible for maintaining accuracy of vehicle locations and allowing for customers to easily reserve their desired cars.

Use Cases:

Rental company admin (18) :

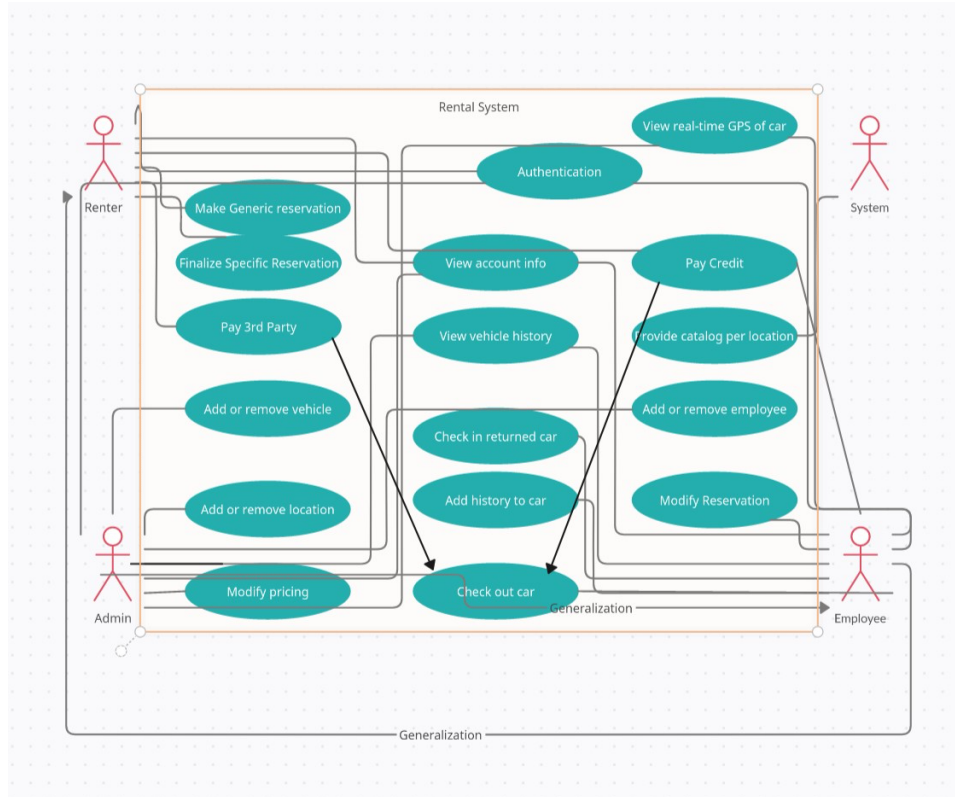
- Add or remove vehicle (2)
- Add or remove employee (2)
- Add or remove location (2)
- Add or modify pricing (2)
- View Account info (2)
- View Vehicle history (4)

Customer (12) :

- Make generic reservation at a specific location (2)
- Make specific reservation within specified timeframe (3)
- Pay for vehicle using credit/debit card (2)
- Pay for vehicle using proprietary payment systems (Apple, Samsung, Google) (5)

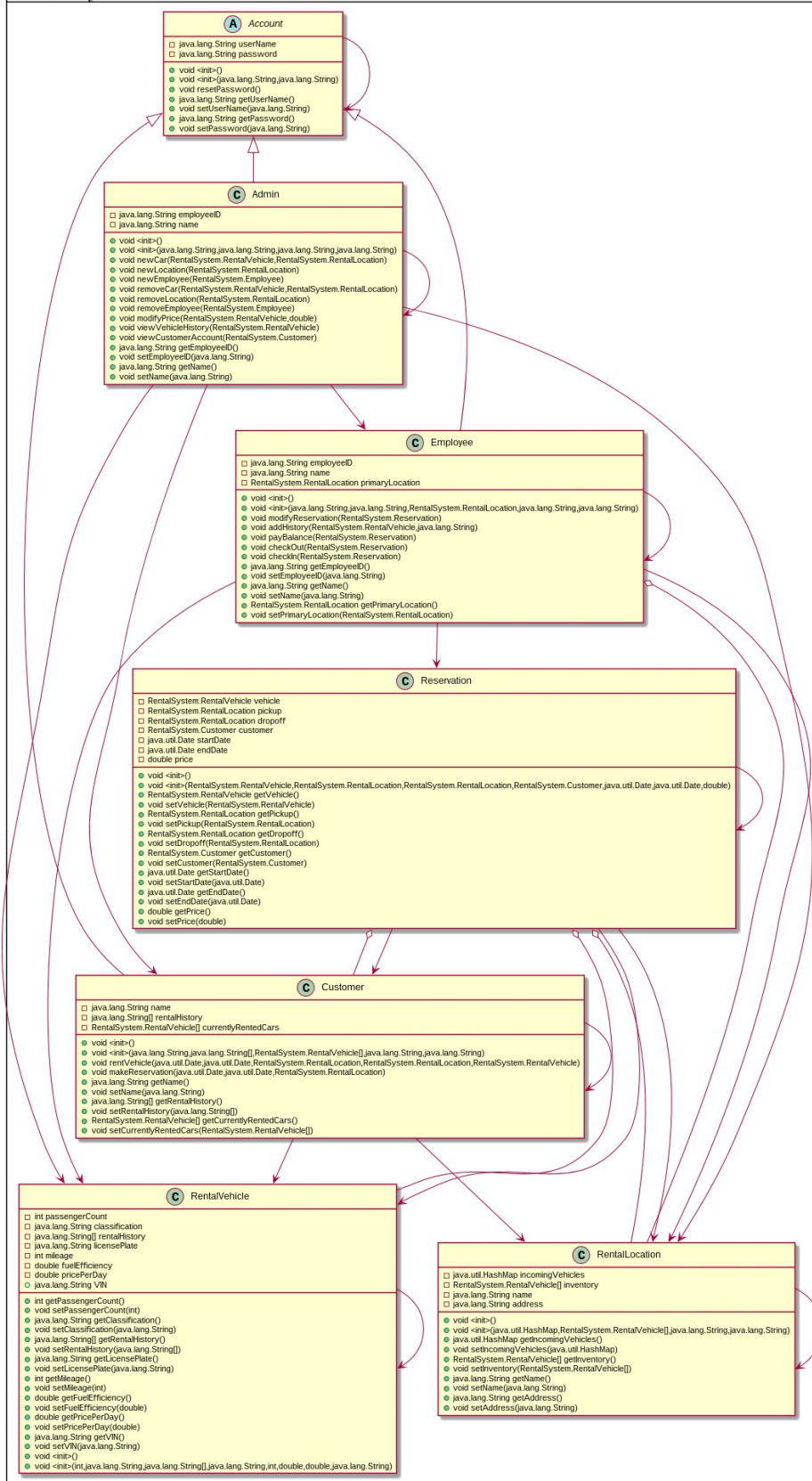
Rental company employee (10) :

- Accept receipt of returned vehicle (2)
- Modify history of vehicle (2)
- Distribute keys for rental (1)
- View real-time location of vehicle (2)
- Modify reservation (2)
- Accept customer payment via credit card (1) System (2) :
- Providing catalogs of vehicles by location (2)
- Authentication (4)



Class diagrams:

# RentalSystem



#### RentalVehicle:

The vehicle class is primarily an abstraction that contains vital information about the rental vehicle in question. It represents the vehicle's classification as a string. Individual classifications are only useful for sorting. Other vital stats include VIN, license plate, mileage, how many seats, and fuel economy.

#### RentalLocation:

The RentalLocation is any physical location that a rental can be checked in or out from. They have a list of in stock vehicles as well as a map containing vehicles that are scheduled to arrive soon and their projected arrival.

#### Reservation:

Reservations are accumulator classes that have pickup and dropoff date and location as well as a vehicle and customer account.

#### Account:

Account is an abstract class that contains only a username and password. It is extended as Customer, Admin, and Employee.

#### Customer:

Customers have a history, list of currently rented cars. They can rent RentalVehicles.

#### Employee:

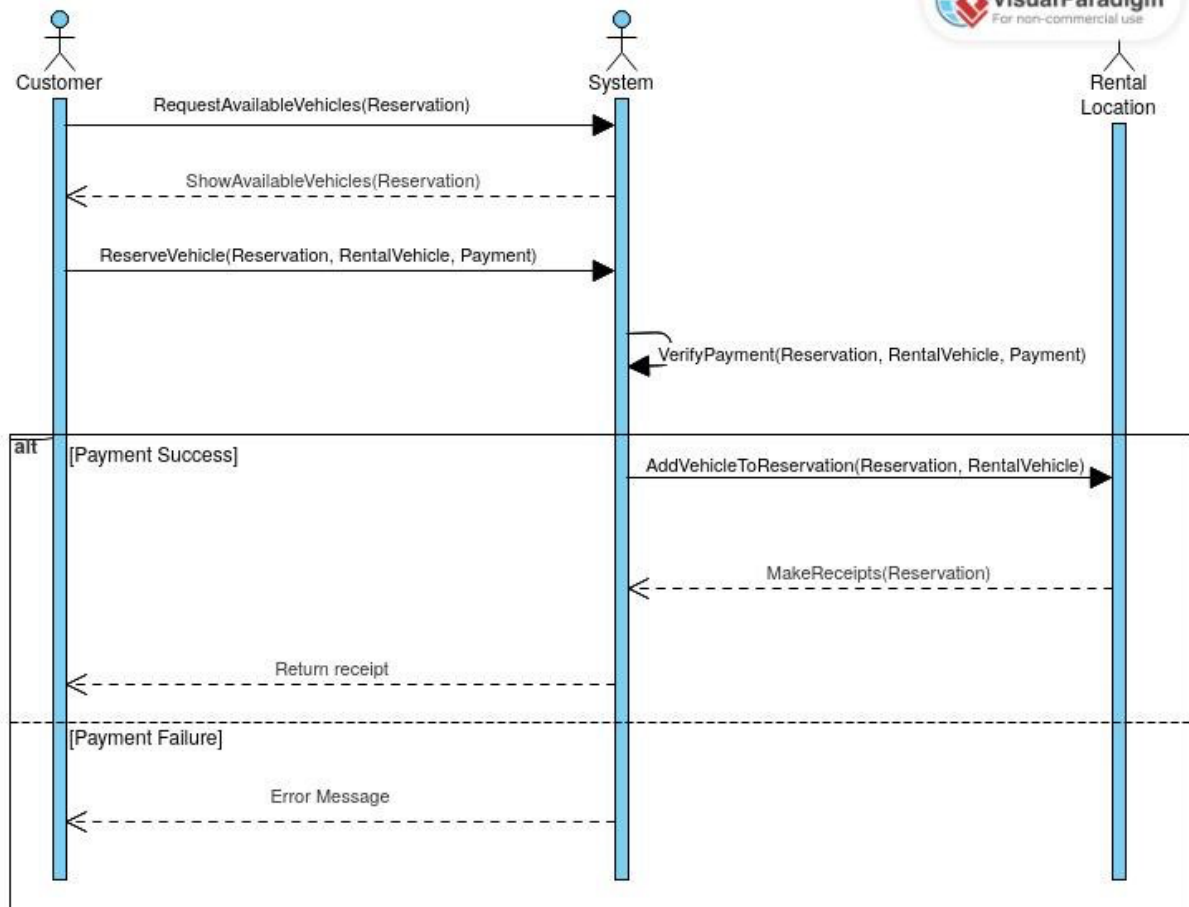
Employees have employee id numbers, names, and a primary working location. They have the ability to modify reservations, check out and check in vehicles, and add history to vehicles.

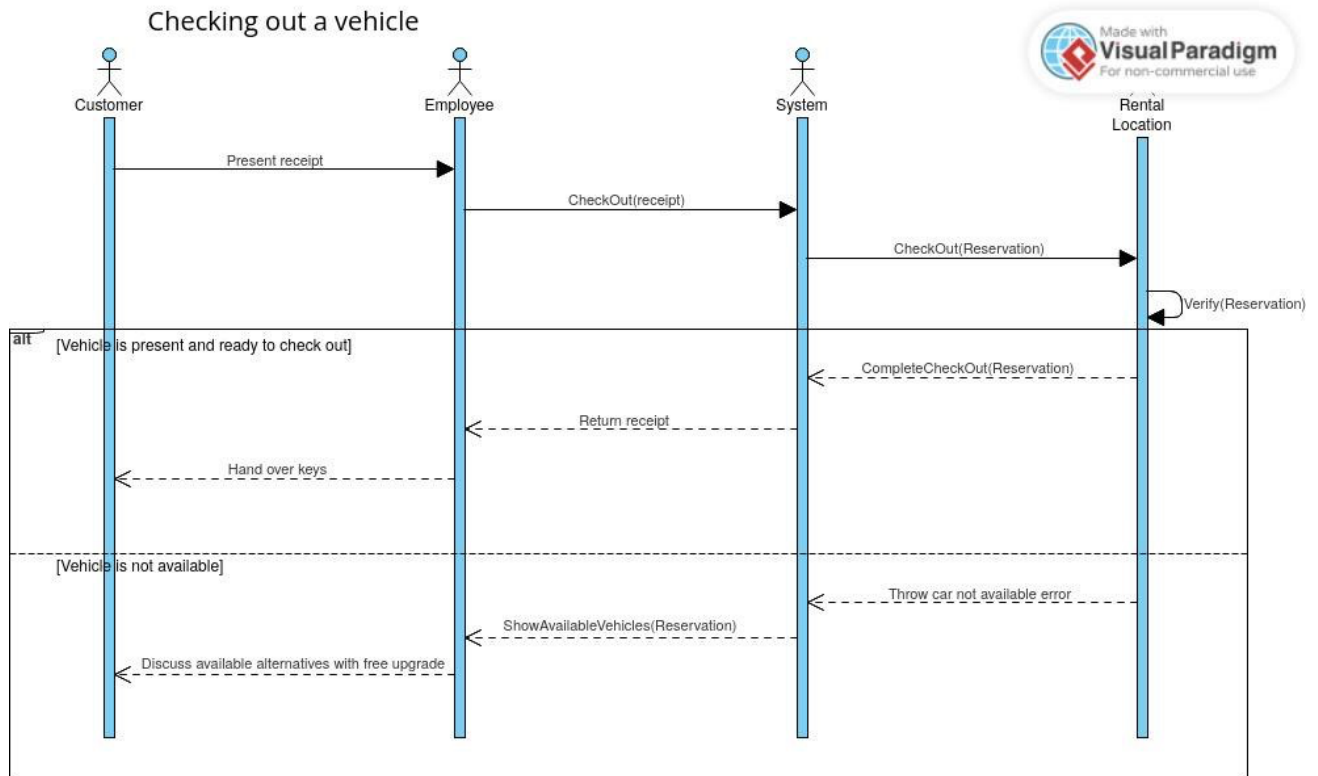
#### Admin:

Admins have employee id numbers and names. They are able to create and destroy vehicles, locations, and employees.



## Reserving a specific vehicle



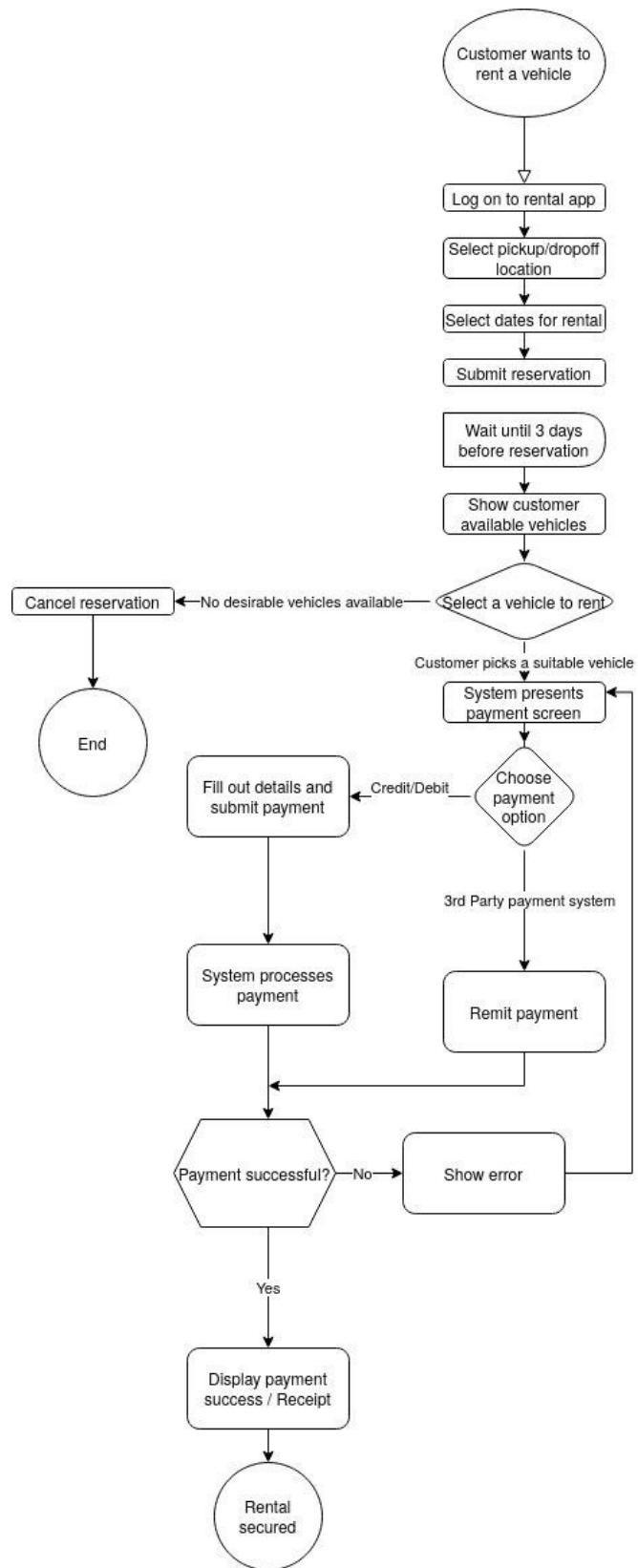


Use Case: Customer reserving a vehicle

Actions: Customer selects dates and locations for rental. Once selected, customer waits until the rental period (3 days prior). Customer is given a choice of vehicles. If none are desirable the customer can end process here. Otherwise, the customer selects a vehicle and pays for it.

States:

- Start: Customer wants a vehicle
- Final: No desirable car available
- Final: Car rented

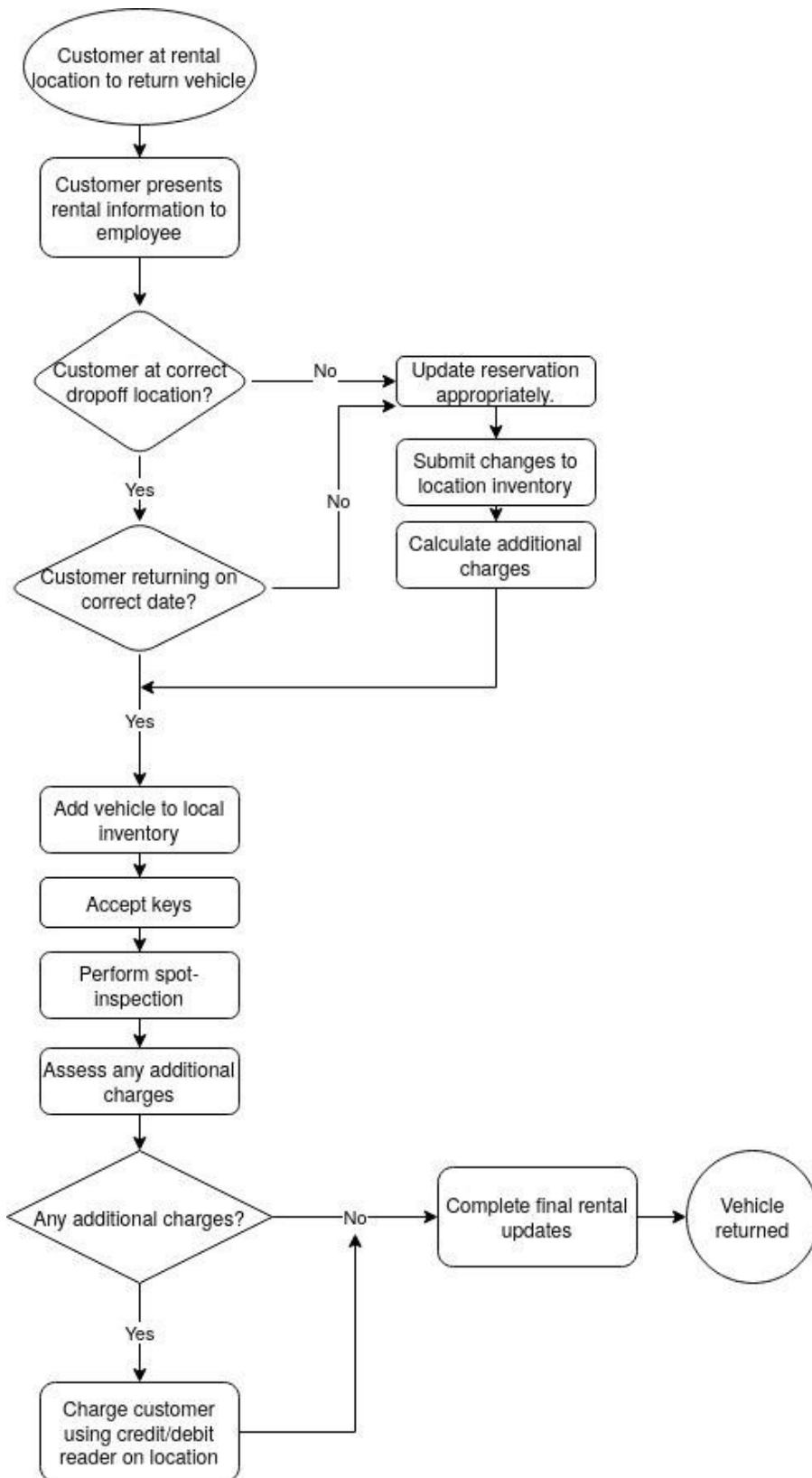


Use Case: Customer returning a vehicle

Actions: Customer arrives at a rental location to drop off car. Employee rectifies rental if location or enddate have changed from original reservation. If there is an additional charge for reservation changes, it is tabulated. The employee takes keys and does a spot inspection. If anything is noticeably damaged, additional charges are tabulated. All additional charges are presented to customer and paid. Return is completed

States:

- Start: Customer arrives at a dropoff location
- Final: Car returned



rentacar.com

# New Vehicle

Rental Location:

Make:

Model:

Year:

Starting Mileage:

VIN:

Vehicle Class:   
Convertible  
Sedan  
Pickup Truck  
Small SUV  
Medium SUV  
Full Size SUV

Passengers:   
3  
4  
5  
6  
7  
8  
9

MPG:

Base Price:

License #:

Creating a new vehicle is done by a corporate employee. The portal seen above is a one page app. It requires end-users to interact with 9 text fields, 2 drop-down menus, and then submit. The rental location must be verified to exist. 12 distinct interactions.

rentacar.com

# New Location

Address 1:

Address 2:

City:

State:

ZIP-Code:

Phone Number:

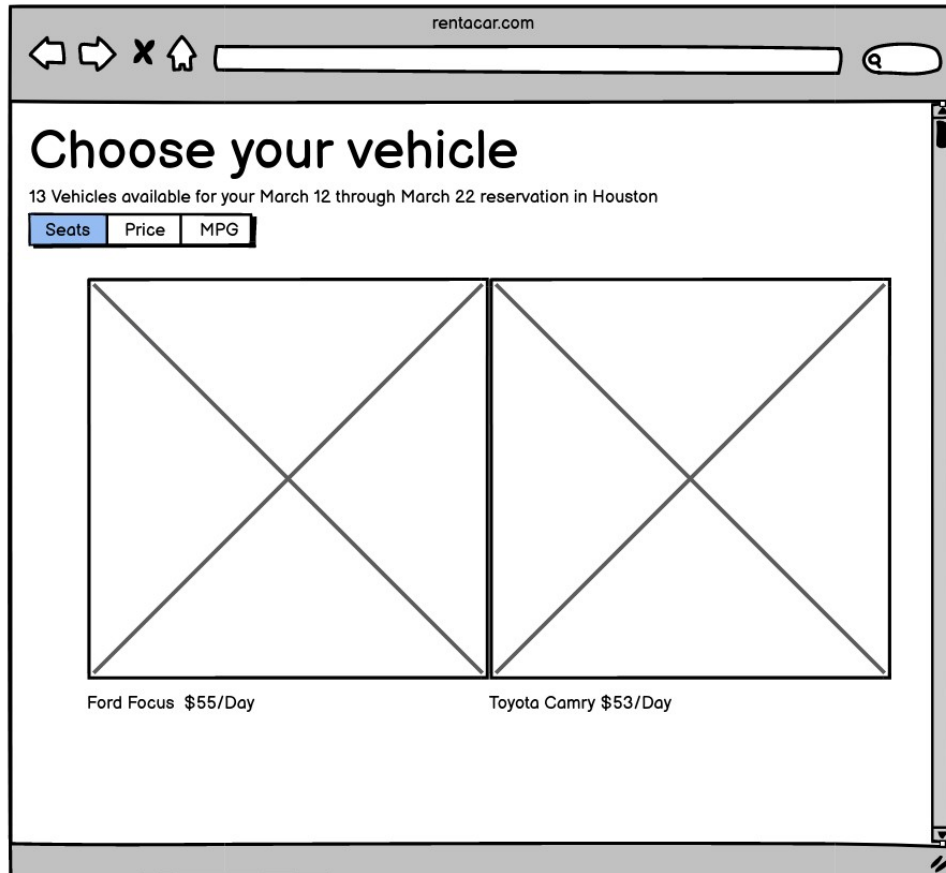
Vehicle Capacity:

☒ Airport location? ☒ Enter current vehicles?

Adding a new location is also done by a corporate employee. The Employee must add all address information, a phone number, and vehicle capacity. There are two on/off switches. One indicates whether or not the location is at an airport. The other indicates whether or not the employee will immediately be adding vehicles to the location. If this is the case, a the user will immediately be taken to the new vehicle screen after hitting submit. 10 distinct interactions.

Making a generic interaction is the first time a customer will interact with the system. The User will have to enter a start and end date for the rental and the pickup location. If the user then selects that they will drop off at the same location, they will be able to submit immediately. If, however they want to drop off the vehicle elsewhere, they will need to enter that information. 6 distinct interactions.





When a customer is allowed to select their rental vehicle, they will be presented with a gallery similar to Google images. They can scroll through images, sort them by seat number, price, or mpg. Upon clicking on a vehicle users will be taken to a portal similar to creating the vehicle (done by admins) but all of the text is static. The amount of times a user interacts with the system is dependent upon how quickly they find a desirable vehicle. After selecting a vehicle the user would then go through the payment process. >= 3 distinct interactions

rentacar.com

## License # ABC 123 History

History

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore

Add new History logs here if anything of note has happened...

Damage ▾

Service

Police

Submit

A rental company employee would need to add history to a vehicle if there is something extraordinary that happened during its most recent rental. At this portal the employee would be able to see all previous history entries and is able to add new history items. The User would be able to add text, then select the category of the update, and submit it.

3 distinct interactions

## Plan of work:

- w1 - 2: Design front end in React and connect to mongodb database using node ----
  - I'm still working on the front-end design for customers. It is coming along. The database connection has been tested.
- w3 - 5: build management system for rental companies
  - Create vehicles
  - Manage vehicles on location
  - Request maintenance
  - I'm also currently working on designing how the data will be modeled in this system
- w6 - 7: Build out the authentication system for both user types
- w8: test the main features accomplished, record the demo for the mid-term
- w9: implement employee functionality
- w10: implement customer functionality
- w11: beautify everything
- w12 - 14: writing test cases for the implemented features or continue building possible features you'd like to implement
- w15: record the demo for the final presentation