CS3354 Software Engineering
Final Project Deliverable 1

Car Rental App

Sam Ali

Austin Harris

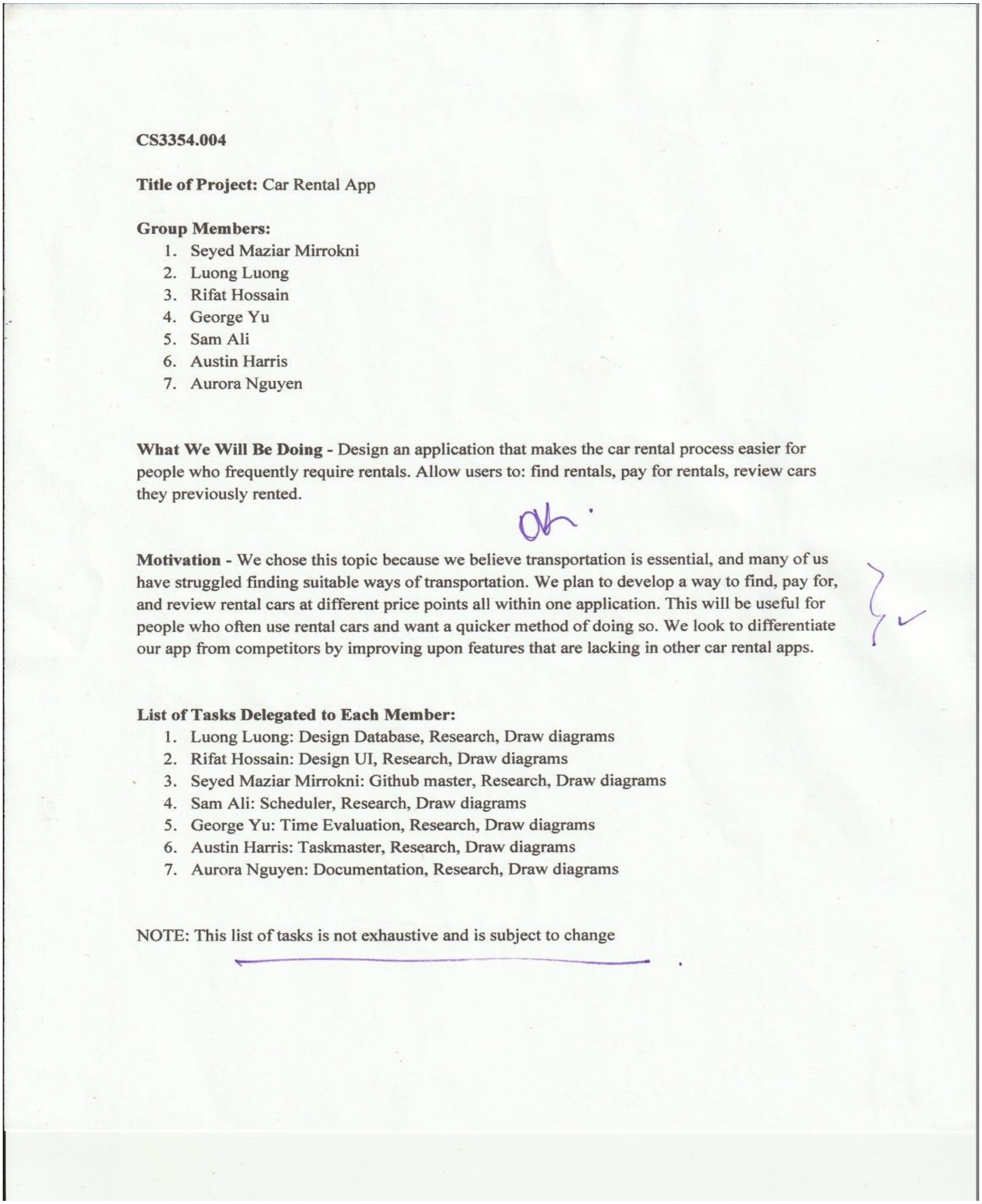
Rifat Hossain

Luong Luong

Seyed Maziar Mirrokni

Aurora Nguyen

George Yu



CS3354.004

Title of Project: Car Rental App

Group Members:

1. Seyed Maziar Mirrokni
2. Luong Luong
3. Rifat Hossain
4. George Yu
5. Sam Ali
6. Austin Harris
7. Aurora Nguyen

What We Will Be Doing - Design an application that makes the car rental process easier for people who frequently require rentals. Allow users to: find rentals, pay for rentals, review cars they previously rented.

ok.

Motivation - We chose this topic because we believe transportation is essential, and many of us have struggled finding suitable ways of transportation. We plan to develop a way to find, pay for, and review rental cars at different price points all within one application. This will be useful for people who often use rental cars and want a quicker method of doing so. We look to differentiate our app from competitors by improving upon features that are lacking in other car rental apps. ✓

List of Tasks Delegated to Each Member:

1. Luong Luong: Design Database, Research, Draw diagrams
2. Rifat Hossain: Design UI, Research, Draw diagrams
3. Seyed Maziar Mirrokni: Github master, Research, Draw diagrams
4. Sam Ali: Scheduler, Research, Draw diagrams
5. George Yu: Time Evaluation, Research, Draw diagrams
6. Austin Harris: Taskmaster, Research, Draw diagrams
7. Aurora Nguyen: Documentation, Research, Draw diagrams

NOTE: This list of tasks is not exhaustive and is subject to change

The feedback we received was “ok”. We took this as a sign that we are on the right track. We did, however, try to improve upon the proposal by making the role of each member more clear and divide tasks evenly between all of us.

Github

https://github.com/maziar-mirrokni/3354-DDExpress

Delegation of Tasks

Aurora: Sequence for Create Account/Delete Account; Q3 Software Process Model

Austin: Sequence for Search by Category/Refund Order; Q4.B; Draw Class Diagram

George: Sequence for Place/Modify Orders/View Order History;Draw Use Case Diagram; Draw Class Diagram

Luong: Sequence for Ban Account/View Bills

Maziar: Sequence for Payment Plan/Edit profile and password; Architectural Design; Q8; Draw Class Diagram

Rifat: Sequence for Add & Remove Listing/Select Car for Rental; Q4.A

Sam: Sequence for Login/Save Payment Method/View available cars; Draw Class Diagram

Software Process Model

We will be using incremental development. The incremental method will drastically reduce the cost of implementing requirements changes, as well as reducing the amount of analysis and documentation required so that we can focus more on the development of the app itself. It also allows us easier access to customer feedback so that we can continue to improve and refine the design of our app to make it easy to use for everyone.

Software Requirements

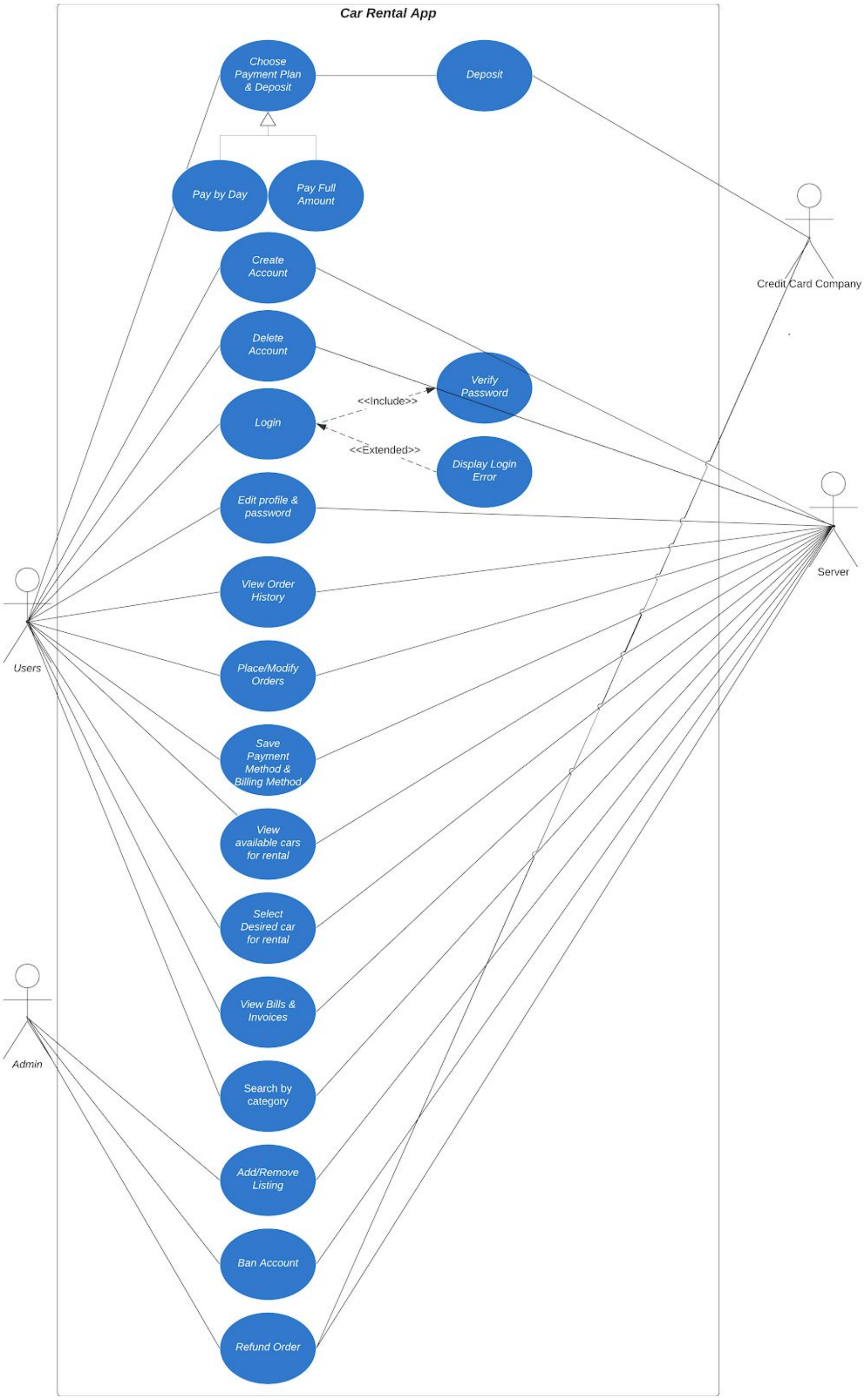
Functional Requirements:

- Create account for app: A user will be able to create a unique account with the system.
- Login to app: A user will be able to login to the app with a username and password authentication system.
- Search Car: A user shall be able to search for a car to rent based on make, model, year, cost, etc.
- Place/Modify order on rental Car: A user shall be able to place/modify the order of a car rental through the system.
- View Order History: A user shall be able to view their view history and view all of their previous orders/returns.
- Refund/return order: The system processes a return. The system records the data/time of return, and based on these parameters, the system determines the refund amount.

Non-functional Requirements:

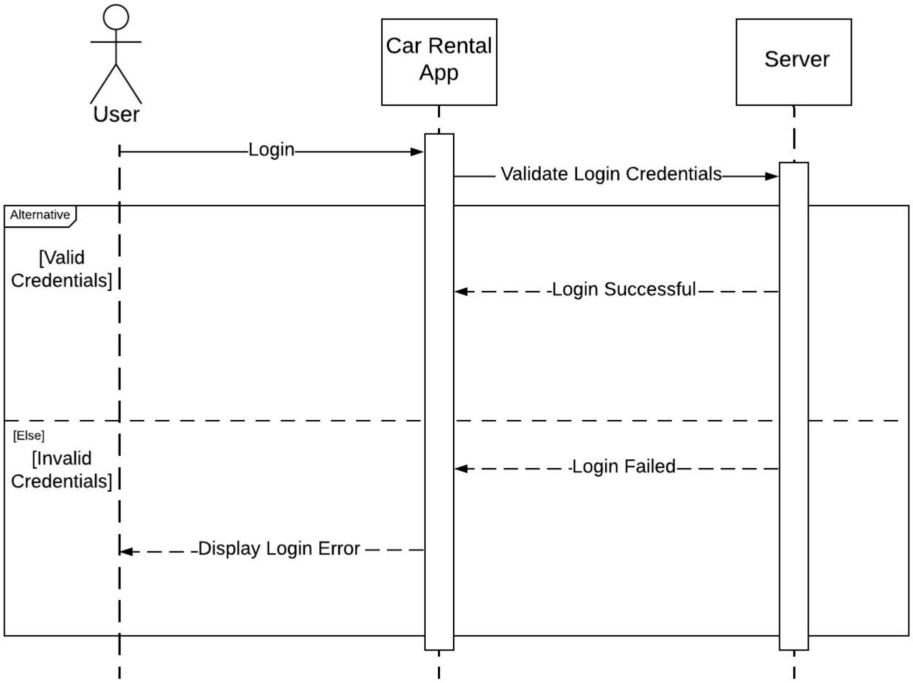
- Usability Requirements: Must be easy to understand for all users
- Performance Requirements: Server should be able to process 50,000 users
- Space Requirements: App must be less than 100 MB
- Dependability Requirements: Server downtime must be less than 10 seconds per day
- Security Requirements: All user data will be encrypted with 256 bit encryption
- Environmental Requirements: Must run on Android and iPhone
- Operational Requirements: Users will be required to make an account to use the app
- Development Requirements: Designed to be easy to port to other operating systems
- Regulatory Requirements: All users must have a valid drivers license
- Ethical Requirements: No user data will be sold
- Accounting Requirements: Should cost less than $100,000 to develop
- Safety/Security Requirements: All user data will be encrypted with 256 bit encryption

Use Case Diagram

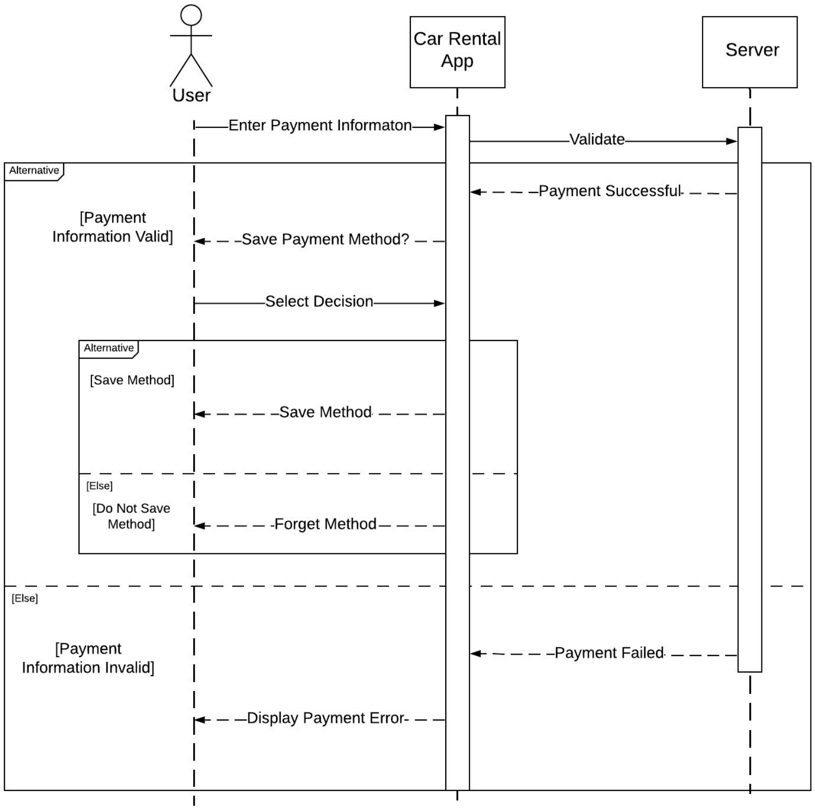


Sequence Diagrams

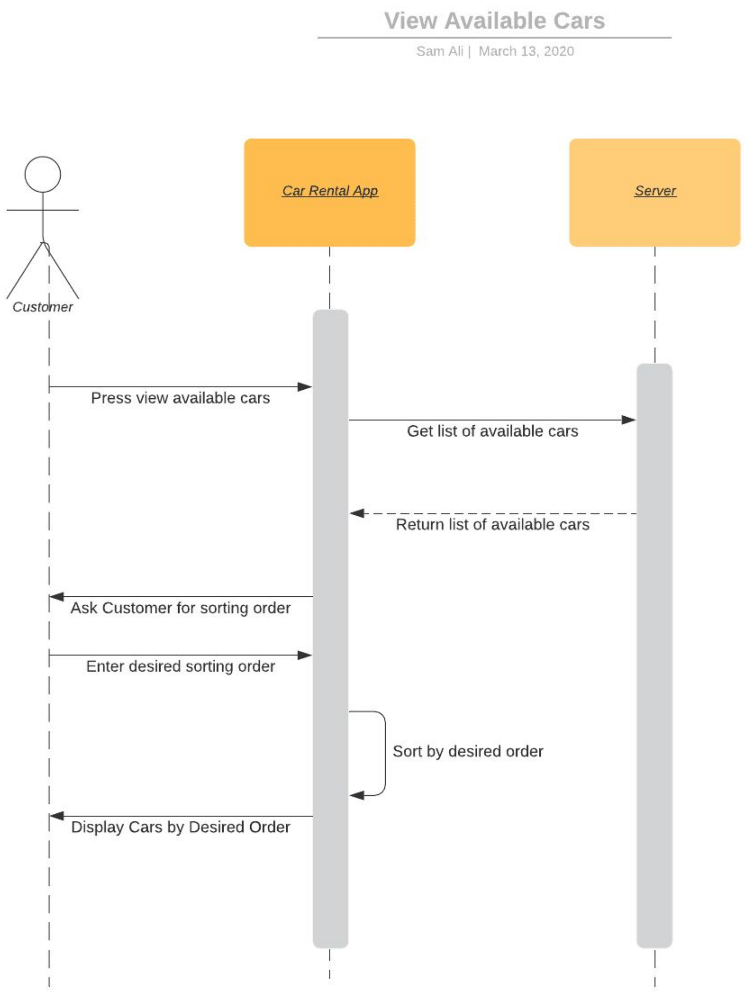
Login (Sam Ali)



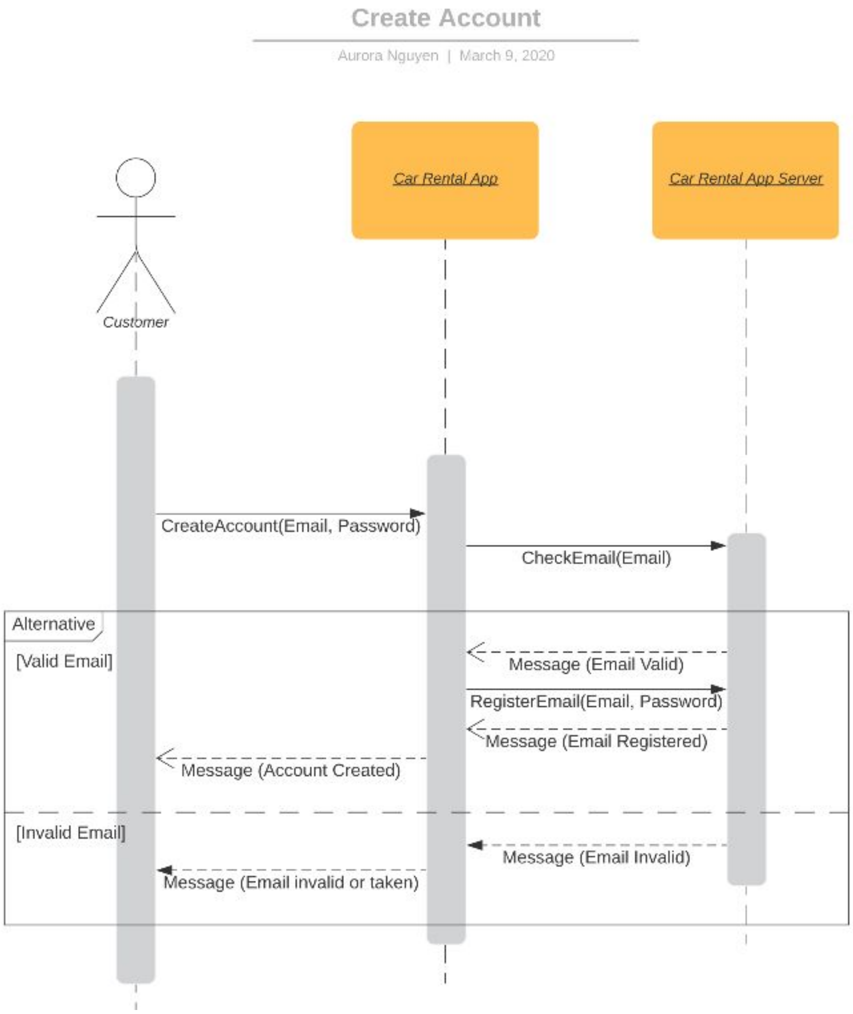
Save Payment Method (Sam Ali)



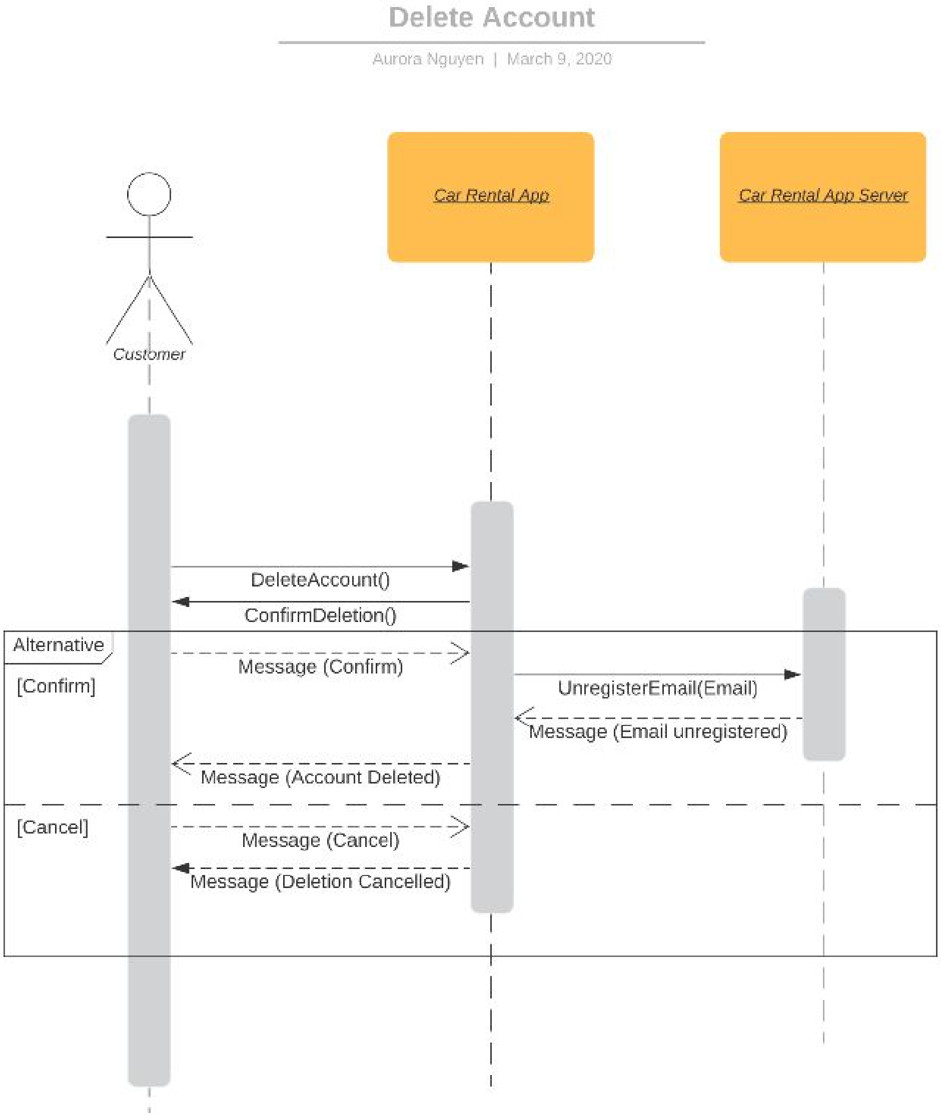
View Available Cars (Sam Ali)


View Available Cars
Sam Ali | March 13, 2020
Customer
Car Rental App
Server
Press view available cars
Get list of available cars
Return list of available cars
Ask Customer for sorting order
Enter desired sorting order
Sort by desired order
Display Cars by Desired Order

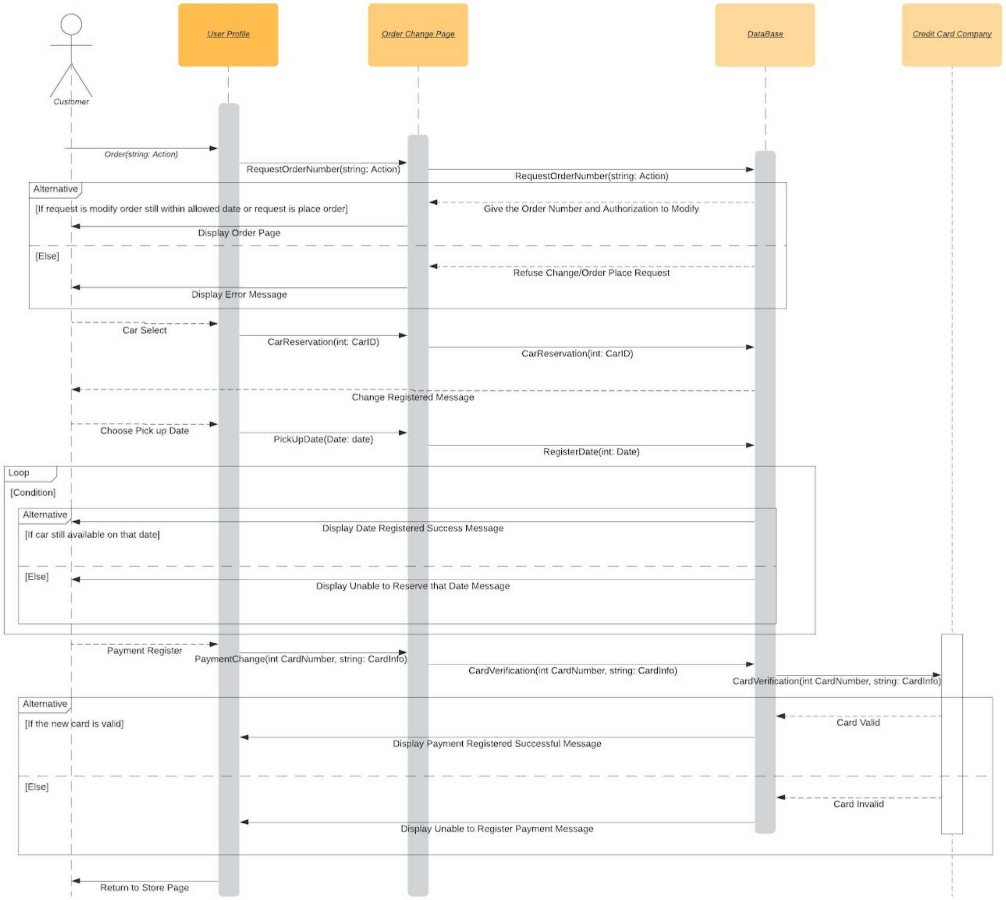
Create Account (Aurora Nguyen)


Create Account
Aurora Nguyen | March 9, 2020
Customer
Car Rental App
Car Rental App Server
CreateAccount(Email, Password)
CheckEmail(Email)
Alternative
[Valid Email]
Message (Email Valid)
RegisterEmail(Email, Password)
Message (Email Registered)
Message (Account Created)
[Invalid Email]
Message (Email Invalid)
Message (Email invalid or taken)

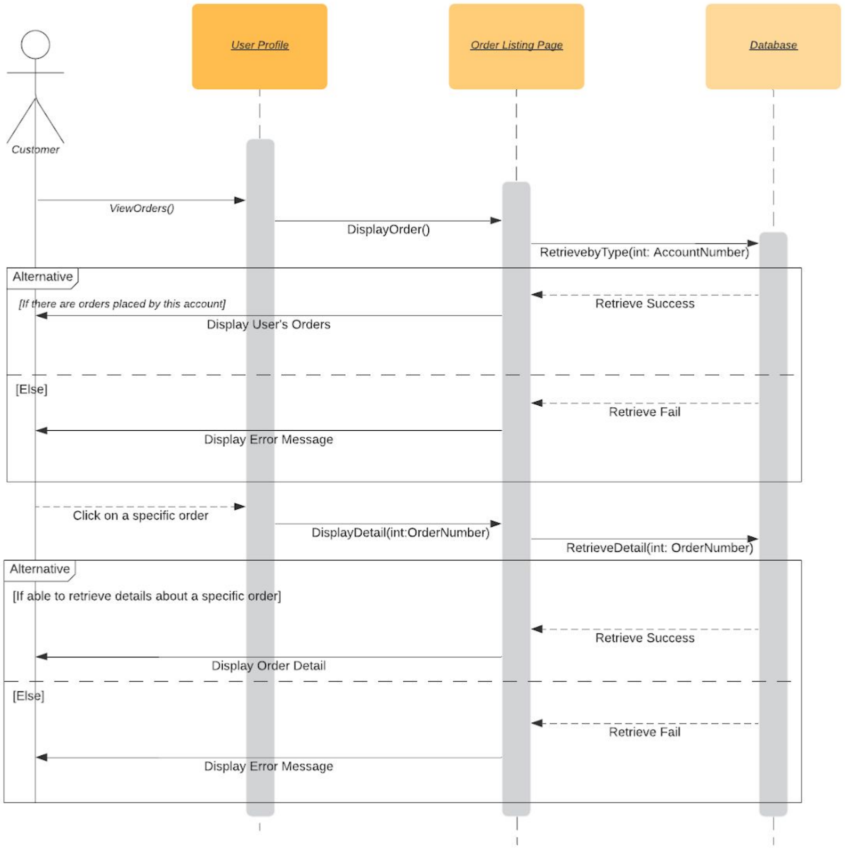
Delete Account (Aurora Nguyen)



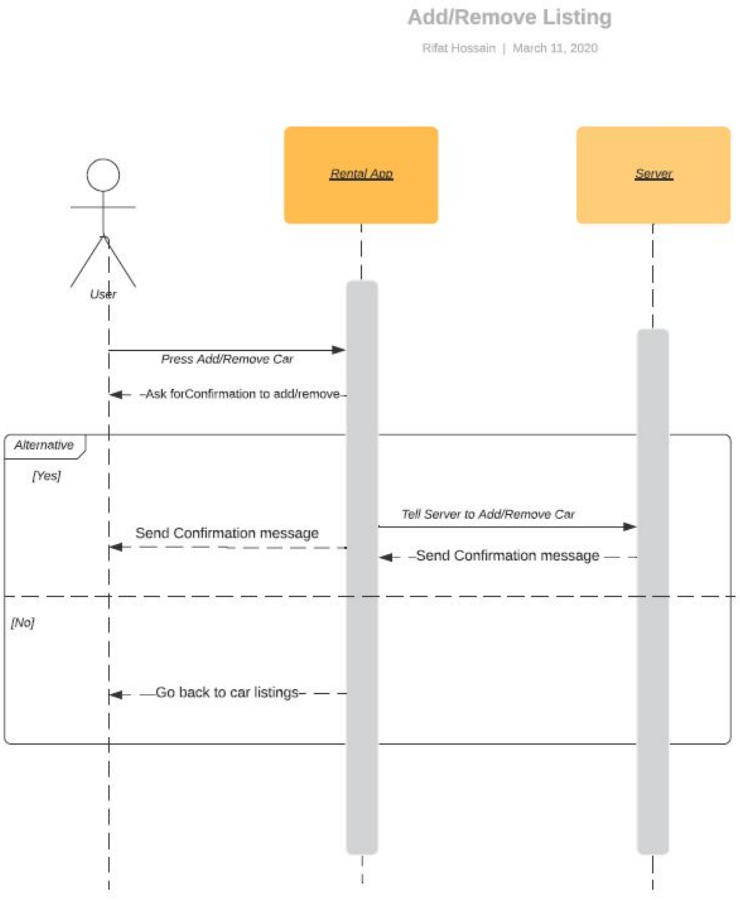
Edit/Place Order (George Yu)


User Profile
Order Change Page
DataBase
Credit Card Company
Customer
Order(string: Action)
RequestOrderNumber(string: Action)
RequestOrderNumber(string: Action)
Alternative
[If request is modify order still within allowed date or request is place order]
Give the Order Number and Authorization to Modify
Display Order Page
[Else]
Refuse Change/Order Place Request
Display Error Message
Car Select
CarReservation(int: CarID)
CarReservation(int: CarID)
Change Registered Message
Choose Pick up Date
PickUpDate(Date: date)
RegisterDate(int: Date)
Loop
[Condition]
Alternative
Display Date Registered Success Message
[If car still available on that date]
[Else]
Display Unable to Reserve that Date Message
Payment Register
PaymentChange(int CardNumber, string: CardInfo)
CardVerification(int CardNumber, string: CardInfo)
CardVerification(int CardNumber, string: CardInfo)
Alternative
[If the new card is valid]
Card Valid
Display Payment Registered Successful Message
[Else]
Card Invalid
Display Unable to Register Payment Message
Return to Store Page

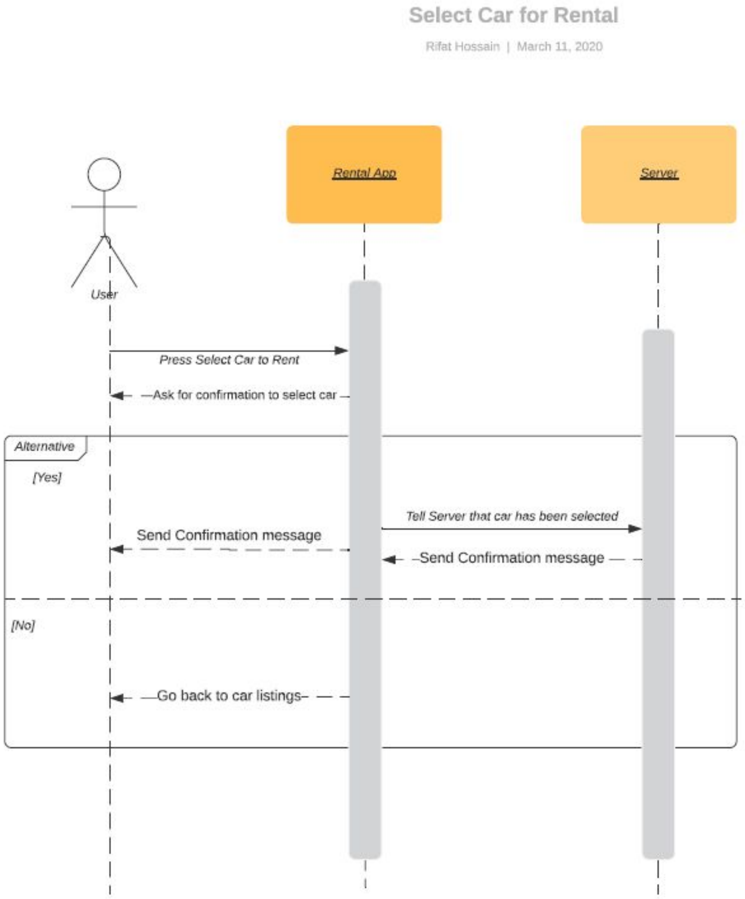
View Order History (George Yu)



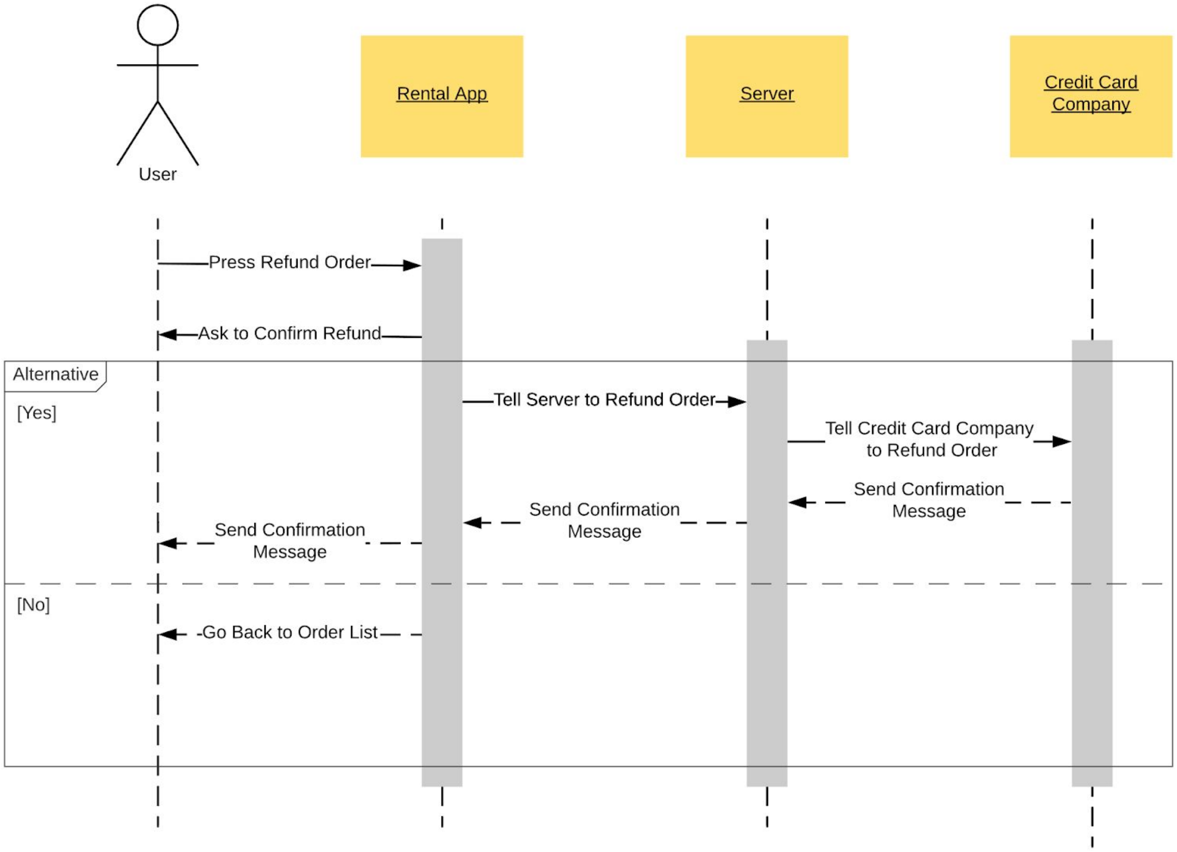
Add/Remove Listing (Rifat Hossain)


Add/Remove Listing
Rifat Hossain | March 11, 2020
Rental App
Server
User
Press Add/Remove Car
Ask forConfirmation to add/remove
Alternative
[Yes]
Tell Server to Add/Remove Car
Send Confirmation message
Send Confirmation message
[No]
Go back to car listings

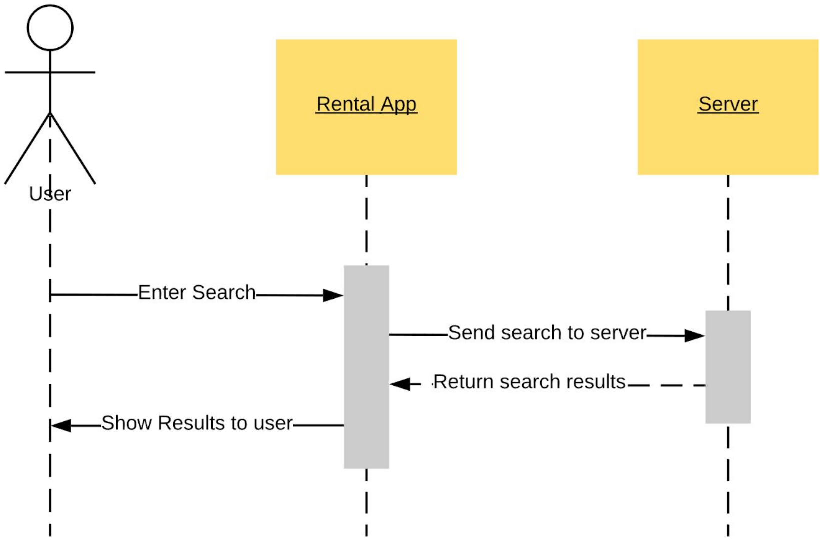
Select Car for Rental (Rifat Hossain)


Select Car for Rental
Rifat Hossain | March 11, 2020
Rental App
Server
User
Press Select Car to Rent
Ask for confirmation to select car
Alternative
[Yes]
Tell Server that car has been selected
Send Confirmation message
Send Confirmation message
[No]
Go back to car listings

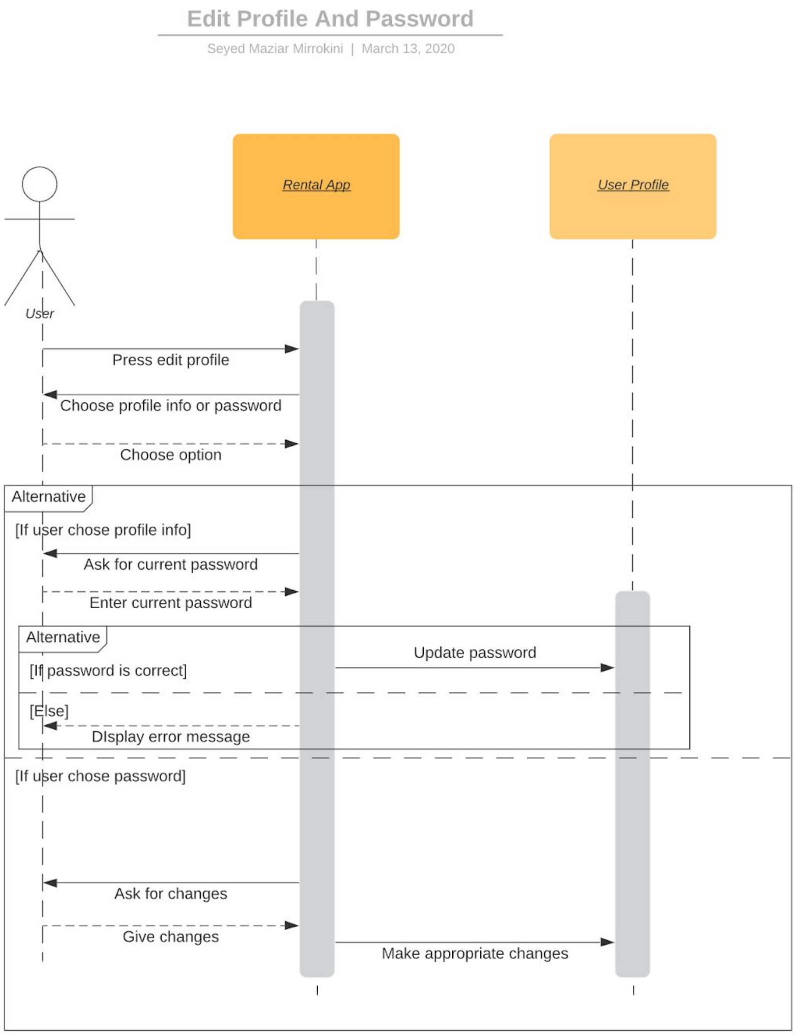
Refund (Austin Harris)


User
Rental App
Server
Credit Card Company
Press Refund Order
Ask to Confirm Refund
Alternative
[Yes]
Tell Server to Refund Order
Tell Credit Card Company to Refund Order
Send Confirmation Message
Send Confirmation Message
Send Confirmation Message
[No]
Go Back to Order List

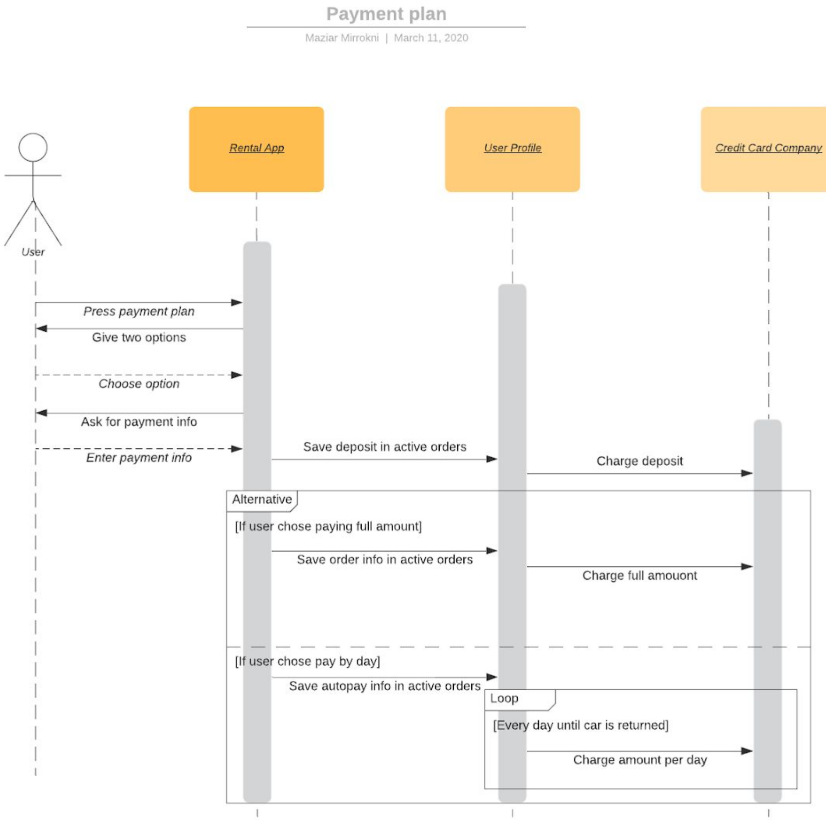
Search (Austin Harris)



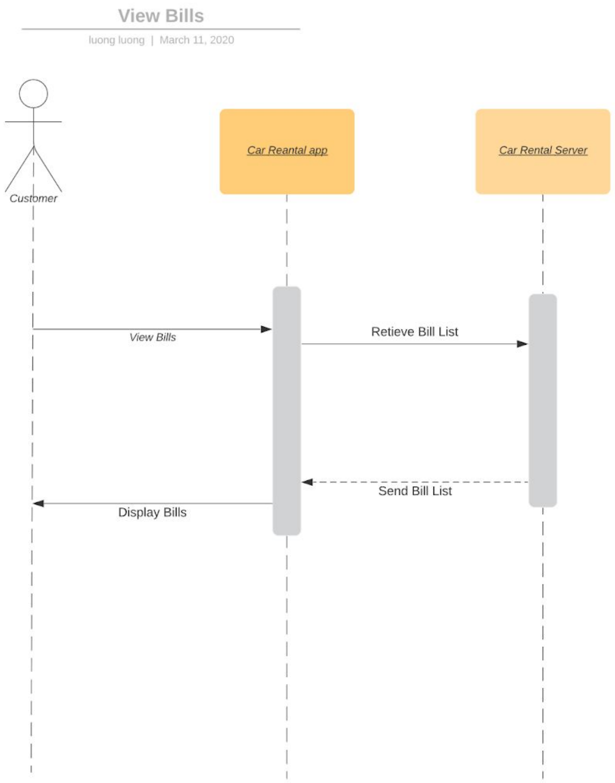
Edit Profile and Password (Seyed Maziar Mirrokni)



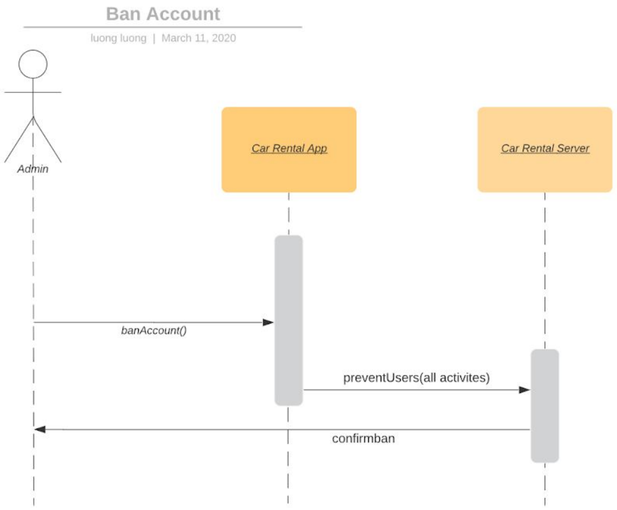
Payment Plan (Seyed Maziar Mirrokni)


Payment plan
Maziar Mirrokni | March 11, 2020
User
Rental App
User Profile
Credit Card Company
Press payment plan
Give two options
Choose option
Ask for payment info
Enter payment info
Save deposit in active orders
Charge deposit
Alternative
[If user chose paying full amount]
Save order info in active orders
Charge full amoount
[If user chose pay by day]
Save autopay info in active orders
Loop
[Every day until car is returned]
Charge amount per day

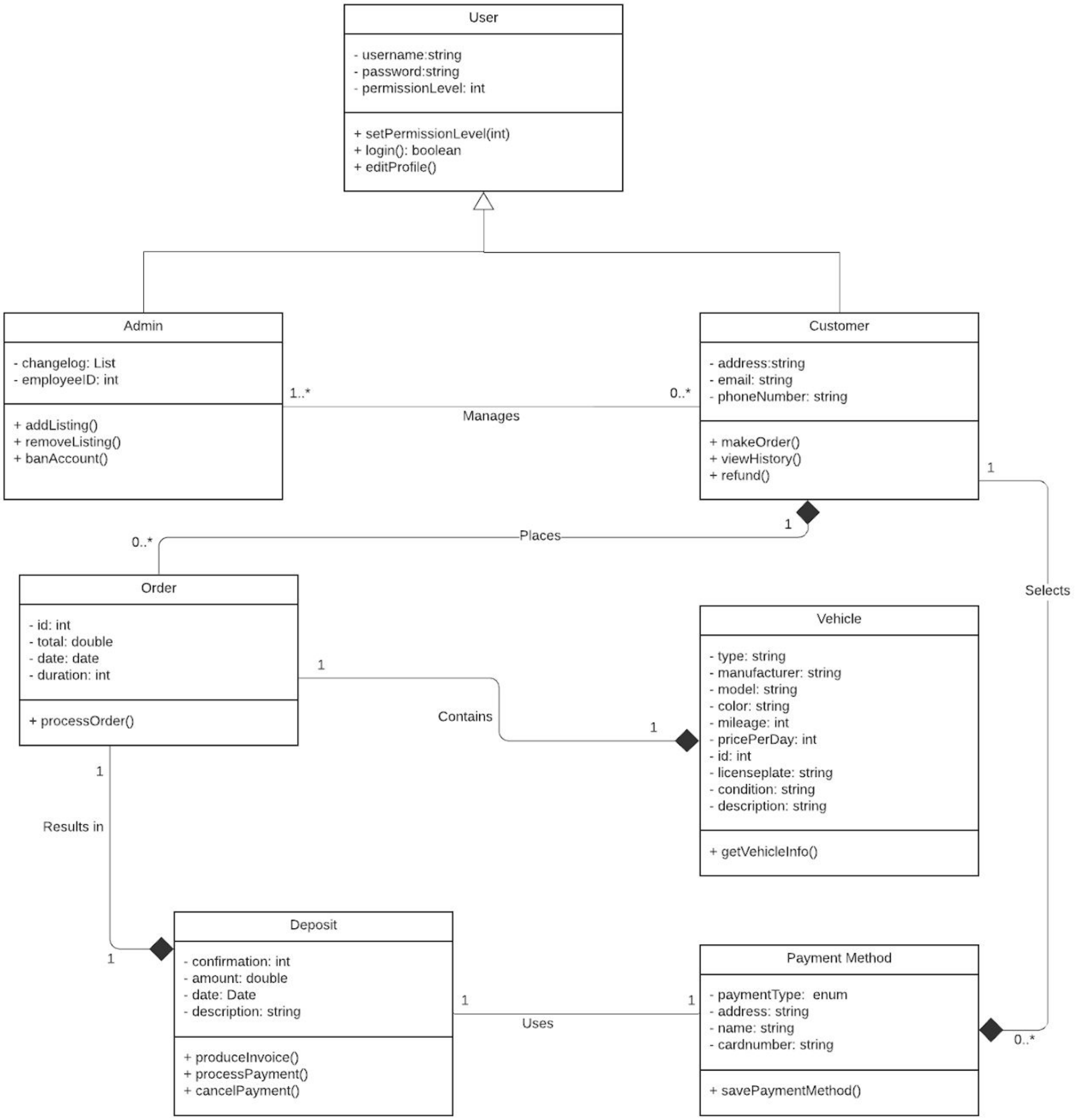
View Bills (Luong Luong)



Ban Account (Luong Luong)



Class Diagram



Architectural Design

Client-Server Architecture

Because of the nature of our application, we must be able to concurrently handle multiple requests from multiple users. Therefore, the client-server architecture is the most appropriate architectural pattern for our project.