Fully dressed use cases

**Functional Requirement:** Users are required to have a driver's license to create an account

**Use Case UC1:** Require Driver’s license

**Primary Actor**: User

**Stakeholders and interests:**

User: wants to be able to rent a car

Rental Company: wants their customers to be able to be legally allowed to drive

**Preconditions**:

* The user would like to create an account

**Success guarantee:**

* The user could create an account if they have a valid driver’s license
* The user could not create an account if they do not have a valid driver’s license

**Main success scenario:**

1. Customer goes to register on the rental agency’s website
2. The user puts in all the information required to register, such as email and password/google sign in/apple sign in, and then is requested to upload a picture of their driver’s license for verification.
3. Once they register, the system processes the image, and validates it as a real driver’s license, and saves the user’s information in the database
4. The user is then returned to the website in a signed in state
5. The user then browses the website looking at cars available for rent
6. The user closes their browser tab

**Extensions:**

3a. Invalid driver’s license

1. Once they register, the system processes the image, and validates it as an invalid driver’s license and presents an error message to the user.
2. The user is prompted to reupload an image of their drivers license
3. Try step 2 in main success scenario again

**Special requirements:**

* Driver’s license must be issued by United States

**Technology and data variation list:**

* Validation is done by scanning the ID’s barcode in the photo

**Frequency of Occurrence:** Once for every customer

**Functional Requirement**: The system should require users to submit pictures of their rental vehicle upon pickup and returning. This will provide proof of any damage to the vehicle. This is additional to recording the condition of the car.

**Use Case UC2:** Submit Rental Vehicle Picture

**Primary Actor**: User

**Stakeholders and interests:**

User: wants to not be liable for any damages done to the vehicle before it was in their possession

Rental Company: wants to charge customers if they cause damages to their vehicles

**Preconditions**:

* The user has already went through the transaction process
* The user has already checked-in and received the keys for their vehicle

**Success guarantee:**

* The photos that the user has captured of the vehicle are saved
* The employees are able to view the pictures that the customer uploaded
* The user is able to view the pictures that they uploaded of the vehicle

**Main success scenario:**

1. The customer has received the keys for their rental vehicle
2. The customer walks to their rental vehicle in the lot
3. The customer takes photos of all the pre-existing damage to their vehicle with their smartphone
4. The customer uploads the photos that they took of the damage to the system to be saved

**Extensions:**

* None

**Special requirements:**

* Should be able to upload as many photos as they would like

**Technology and data variation list:**

* Pictures should be able to be uploaded as JPEG,PNG,and HEIF

**Frequency of Occurrence:** Once per rental vehicle a customer rents

**Functional Requirement**: Managers should be able to designate a car is in need of repair

**Use Case UC3:** Mark vehicle in need of repair

**Primary Actor**: Manager

**Stakeholders and interests:**

Manager: wants to be able to mark cars as in need of repair and see vehicles that need repair

Customer: does not want to rent a vehicle that is in need of repair

**Preconditions**:

* The manager has already been alerted/noticed that a vehicle is in need of repair

**Success guarantee:**

* The vehicle is marked as needing repair
* The vehicle does not show up for customers when they are searching for vehicles to rent

**Main success scenario:**

1. The manager wants to mark a car as needing repair
2. The manager navigates to the system’s website
3. The manager logs in
4. The manager finds the vehicle that needs repair on the website
5. The manager clicks “needs repair”
6. The manager enters some more information about the repairs that are needed

**Extensions:**

* None

**Special requirements:**

* None

**Technology and data variation list:**

* The system should be able to store variable length strings for the repair description

**Frequency of Occurrence:** Whenever a vehicle is in need of repairs

**Functional Requirement**: The user should be able to report an issue with their rental vehicle during their rental period

**Use Case UC4:** Report issue during rental period

**Primary Actor**: Customer

**Stakeholders and interests:**

Customer: wants to be able to get help with their vehicle during their rental

Rental Company: wants to know of any issues that their customers are having

**Preconditions**:

* The customer is already in possession of a rental vehicle

**Success guarantee:**

* The employees are notified of the issue that the customer is having
* The employees are able to provide roadside assistance or technical support

**Main success scenario:**

1. Customer is in their rental vehicle
2. Customer gets a flat tire while driving their rental vehicle
3. Customer goes onto the rental website on their smartphone
4. Customer logs in
5. Customer reports the flat tire on their current rental page
6. The manager is notified of the customer’s issue and sends an employee out to provide roadside assistance
7. The employee arrives to the customer and helps replace their flat tire

**Extensions:**

2a. Customer does not know how to turn on the headlights of their rental vehicle

1. Customer goes onto the rental website on their smartphone
2. Customer logs in
3. Customer asks how to turn on the headlights of their rental
4. The manager gets notified of the customer’s issue
5. The manager responds to the customer with instructions on how to turn on the headlights in that model of vehicle

**Special requirements:**

* There is no limit to the number of issues that a customer can file

**Technology and data variation list:**

* The system should be able to store variable length strings for the customer’s issue and manager’s response

**Frequency of Occurrence:** Whenever a customer runs into an issue during their rental period

**Function Requirement** Create a car rental transaction

**Use Case UC5:** Create transaction

**Primary Actor**: Cashier

**Stakeholders and interests:** The operation owner wants the transactions to take place smoothly so that they have a good reputation and can make more sales. The customer wants the transaction to go smoothly so they can enjoy the rest of their day and not be bogged down by renting a vehicle

**Preconditions:** The desired vehicle is in stock. The customer has enough money to pay for the rental vehicle

**Success Guarantee**: The customer is subject to receive access to their rental vehicle. Sale is secured. Receipt generated

**Main Success:** Customer walks up to the register and asks for a rental vehicle of their choice, the cashier finds the available vehicle the customer wanted, the cashier then creates the rental transaction with the customer’s payment method

**Extensions:**

1. Customer walks up to the register and does not know what vehicle they want. Cashier looks for available vehicles and finds one that is suitable for the customer, the customer approves, and hands over the card. The cashier enters the card into the system. Card is charged. the customer takes the rental vehicle.
2. Customer walks up to the register and does not know what vehicle they want. Cashier looks for available vehicles and finds one that is suitable for the customer, the customer approves, and hands over the card. the cashier enters the card into the system. Card is declined. Customer leaves, cashier cancels the transaction.

**special requirements:**

Credit card authorization responds within 10 seconds 99% of the time.

The transaction is not canceled when the card is declined in the case when the customer has another payment method

**technology and data variation list:**

Rental vehicles have a UUID coding scheme.

Credit card information can be entered in a variety of ways

**Frequency of occurrence:** Every time a customer wants to rent a vehicle

**Functional Requirement:** Users should be able to see any fees that they accumulated during their rental such as tolls, gas fees, etc

**Use Case UC6:** View bill/invoice

**Primary Actor:** Customer

**Stakeholders and interests:** The operators wants customers to be aware of the money they owe so that they do not have to reach out to debt collectors to receive their money, which loses money. Customers want to see the cost breakdown so they are aware of what they are paying for.

**Preconditions:** The customer has returned the rental vehicle

**Success Guarantee:** The customer has returned the rental vehicle.

**Main Success:** Customer receives invoice/bill after returning their rental vehicle and sees all of the accrued costs from their rental

**Extensions:**

**special requirements:**

Invoice is delivered to customer within 1 day of rental returns

Customers have a way to dispute charges with the operators.

**technology and data variation list:**

Rental invoices/bills have a UUID coding scheme.

Multiple ways to contact people to dispute card.

**Frequency of occurrence:** Every time a customer rents a vehicle.

**Functional Requirement:** Users should be able to filter vehicles by how many seats the vehicle has

**Use Case UC7:** Filter vehicles by seat

**Primary Actor:** Customer

**Stakeholders and interests:** The operators want customers to find the vehicle they need so they do not go to another rental company.The customer wants to find the vehicle they need quickly so they do not have to go find another rental company.

**Preconditions:** The system is operational.

**Success Guarantee:** The system shows vehicles with the number of seats the customer wants to see.

**Main Success:** The customer selects how many seats they wish to filter by. The system displays a list of the vehicles that match the query.

**Extensions:**

**Special requirements:**

None

**technology and data variation list:**

None

**Frequency of occurrence:** Whenever a customer wants to filter by number of seats

**Functional Requirement:** The user should be able to get rewards for renting cars multiple times

**Use Case UC8:** Receiverewards

**Primary Actor:** Customer

**Stakeholders and interests:** The operators want customers to be incentivized to continue renting with them. The customer wants a discount on their rental.

**Preconditions:** The customer has given a valid email address/phone number upon renting their vehicle(s)

**Success Guarantee:** The customer provides a valid email address/phone number and completes a transaction

**Main Success:** The customer rents a vehicle. The customer is then rewarded with points

**Extensions:**

**Special requirements:**

None

**technology and data variation list:**

None

**Frequency of occurrence:** Whenever a customer completes a rental transaction.

**Functional Requirement:** Users should be able to save cars they are interested in renting, view the list of cars they are interested in, and edit said list.

**Use Case UC9:** Keep Car Wishlist

**Primary Actor**: Customer

**Stakeholders and interests:**

User: wants to be able to manage a list of cars they are interested in

Rental Company: wants to be able to see which cars are in higher demand

**Preconditions**:

* The user has an account and is logged in
* Each car in the inventory has an option for the user to save it to their list

**Success guarantee:**

* The user can create and manage a list of cars they are interested in if they have an account
* The user can not create or manage said list if they do not have an account

**Main success scenario:**

1. If the customer does not have an account then they must create one
   1. The user chooses to register and puts in all the information required to register, such as email and password/google sign in/apple sign in, and then is requested to upload a picture of their driver’s license for verification.
2. Once they user has an account they should view our inventory of available cars
3. If the user finds a car they are interested in they should click an add to wishlist button
4. Once cars are added to the list the user can view their wishlist under their account tab and view or remove cars from the list
5. When the user is finished they log out and close their browser tab

**Extensions:**

**Special requirements:**

* None

**Technology and data variation list:**

* The list should be able to store as many cars as the customer wants
* Even if the list is empty the customer should be able to view it so they are aware it is a feature

**Frequency of Occurrence:** Every customer can maintain one wishlist

**Functional Requirement:** Users should be able to compare available cars.

**Use Case UC10:** Compare Cars

**Primary Actor**: Customer

**Stakeholders and interests:**

User: wants to be able to compare available cars side by side to see what best suits their needs

**Preconditions**:

* The user can create and manage a list of cars they are interested in if they have an account
* Each car in the inventory has a compare option available to the user

**Success guarantee:**

* The user can compare up to three cars at a time and all of the information of the chosen cars will be shown side by side
* The user can not use the compare functionality without an account
* The user can not use the compare functionality if only one car is selected

**Main success scenario:**

1. If the customer does not have an account then they must create one
   1. The user chooses to register and puts in all the information required to register, such as email and password/google sign in/apple sign in, and then is requested to upload a picture of their driver’s license for verification.
2. Once they user has an account they should view our inventory of available cars
3. If the user finds a car they want to compare they should click the compare button
4. The user will be able to select up to two more cars either from searching the inventory or from their wishlist
5. Once the desired cars are selected the user should click the compare selected cars button
6. Each selected car will be shown in a column with all of the important information for each car in a row - the user should be able to easily see which car is best suited for them in all of the important aspects (a row for image, price, detail, availability, size, etc.)

**Extensions:**

**Special requirements:**

* None

**Technology and data variation list:**

* The user can compare two or three cars depending on how many they select

**Frequency of Occurrence:** Whenever a customer chooses to compare cars

**Functional Requirement:** Employees should be able to approve or deny customer rental request.

**Use Case UC11:** Approve Transaction

**Primary Actor**: Employee

**Stakeholders and interests:**

Employee: checks to see if the customer is suitable

Rental Company: only wants to rent to customers that have a good/ reliable driving and credit history

**Preconditions**:

* A customer completed a transaction with the company
* An employee is logged into the system

**Success guarantee:**

* The employee must approve the transaction if the customer meets the credit and driving history criteria
* The employee must deny the transaction if the customer fails to meets the credit and driving history criteria

**Main success scenario:**

1. When a customer completes a transaction the employee should view the list of transaction request
2. Based on the provided insurance information the employee should verify that the user meets the credit and driving histoy criteria
3. If the customer meets the criteria then the employee must approve the transaction so it can go through
4. Once the transaction is approved the customer will be notified

**Extensions:**

**Special requirements:**

* None

**Technology and data variation list:**

* None

**Frequency of Occurrence:** Whenever a customer request a transaction

**Functional Requirement:** The system must allow users to login with a username and password.

**Use Case UC12:** Validate User

**Primary Actor**: Customer

**Stakeholders and interests:**

User: wants to be able to access their accounts

**Preconditions**:

* The user has already a registered account

**Success guarantee:**

* The will be granted access upon entering their username and correct password
* The will not be granted access upon entering an incorrect username or an incorrect password

**Main success scenario:**

1. When a user wants to login to their account they navigate to the login page
2. User correctly enters their username
3. User correctly enters their password
4. User tries to login and does so successfully which redirects them to their account page

**Extensions:**

**Special requirements:**

* None

**Technology and data variation list:**

* The system should be able to efficiently store account login information
* The system should grant access in less than 3 seconds
* The system should be able to process 10000 login request per minute

**Frequency of Occurrence:** Whenever a user wants to login to their account

**Functional Requirement:** Employees should be able to add a car to the fleet

**Use case UC13:** Add Car

**Primary Actor**: Employee

**Stakeholders and interests:**

Employee: wants to be able to add to the rental fleet

Rental Company: wants to make more cars available to rent

**Preconditions**:

* Employee is identified and authenticated
* Employee has identified a car that has not been added to the fleet yet

**Success guarantee:**

* Car inventory and availability is updated for the rental company

**Main success scenario:**

1. New car becomes eligible to be added to the fleet
2. Employee accesses inventory system
3. Employee enters car identifying information. Also enters information about the condition of the car.
4. Employee confirms the car information.
5. System adds a new entry in the inventory system and marks it as available to rent.
6. System sends confirmation that the car was added to the fleet.

**Extensions:**

3a. Invalid car information

1. System signals that the information is invalid.
2. Employee responds to error:

2a. The information is corrected by the Employee

1a. System validates information and continues.

2b. Information still invalid, repeat 3a. until correct or Employee exits

3b. Incomplete car information

1. Cashier responds:

1a. Completes car information and sends to system

1b. Unable to complete car information and cancels car addition.

4a. Employee does not confirm car information

1. Addition of car to fleet is canceled
2. Systems returns to prompt to add car to fleet

5a. System unable to add car to inventory system

1. System retries attempt to reach inventory storage system

1a. System is able to reach inventory storage system and continues

1b. System is unable to reach inventory system, logs error, and returns information to the user.

**Special requirements:**

**Technology and data variation list:**

3a. Car information must include a valid VIN number, License plate number and issuing state, color, make, and model

**Frequency of Occurrence:** Once for every new car to be added to the fleet

**Functional Requirement:** Employees should be able to remove a car from the fleet

**Use case UC14:** Remove Car

**Primary Actor**: Employee

**Stakeholders and interests:**

Employee: wants to be able to remove from the rental fleet

Rental Company: wants to make list of cars available to rent accurate or to sell cars from the fleet

User: Would like to purchase rental car for themself

**Preconditions**:

* Employee is identified and authenticated
* Employee has identified a car that has to be removed from the fleet

**Success guarantee:**

* Car inventory and availability is updated for the rental company

**Main success scenario:**

1. Car is designated for removal by Employee
2. Employee accesses inventory system
3. Employee makes car unavailable to be rented.
4. Employee enters car identifying information. Also enters information about the condition of the car.
5. Employee confirms the car information.
6. System removes entry in the inventory system.
7. System sends confirmation that the car was removed from the fleet.

**Extensions:**

3a. Car is currently being rented by a User

1. System informs Employee car is currently rented out and duration of rental.

2. System exits request.

4a. Invalid car information

1. System signals that the information is invalid.
2. Employee responds to error:

2a. The information is corrected by the Employee

1a. System validates information and continues.

2b. Information still invalid, repeat 3a. until correct or Employee exits.

1. Car made available to be rented

4b. Incomplete car information

1. Cashier responds:

1a. Completes car information and sends to system

1b. Unable to complete car information and cancels car addition.

1. Car made available to be rented

5a. Employee does not confirm car information

1. Removal of car from fleet is canceled
2. Car made available to be rented
3. Systems returns to prompt to remove car from fleet

6a. System unable to add car to inventory storage system

1. System retries attempt to reach inventory storage system

1a. System is able to reach inventory system and continues

1b. System is unable to reach inventory system, logs error, and returns information to the user.

1. Car made available to be rented

**Special requirements:**

**Technology and data variation list:**

4a. Car information must include a valid VIN number, License plate number and issuing state, color, make, and model

**Frequency of Occurrence:** Once for every new car to be removed from the fleet

**Functional Requirement:** The System and Employees should be able to reserve a car in the fleet

**Use case UC15:** Reserve Car

**Primary Actor**: Customer, Employee

**Stakeholders and interests:**

Employee: wants to be able to reserve cars for rental customers who reserve in person

Rental Company: wants to accommodate customers who did not/cannot use the online app

Customer: wants to be able to have a car reserved for them

**Preconditions**:

* Employee or Customer is identified and authenticated
* Employee or Customer has identified a car that a customer wants to rent

**Success guarantee:**

* Car inventory and availability is updated for the rental company
* Customer request has been completed

**Main success scenario:**

1. System presents current available inventory to the Employee or Customer
2. Employee or Customer chooses car from available inventory
3. Employee or Customer completes rental transaction
4. Selected car is marked as rented in the inventory system
5. System confirms car is no longer able to be rented with Customer and Manager

**Extensions:**

1a./4a. System unable to update inventory data

1. System retries attempt to reach inventory storage system

1a. System is able to reach inventory system and continues

1b. System is unable to reach inventory system, logs error, and returns error information to the user.

2a. Customer or Employee chooses unavailable car

1. System reports car as unavailable.

2. System displays available inventory

3a. Rental transaction fails:

1. System reports and logs failure of transaction, including reason.

2. System returns to screen presenting current inventory.

3. Repeat from 2.

**Special requirements:**

* Inventory system must support concurrent access

**Technology and data variation list:**

3a. Car information must include a valid VIN number, License plate number and issuing state, color, make, and model

**Frequency of Occurrence:** Once for every car rental transaction

**Functional Requirement:** Managers should be able to track a car's location

**Use case UC16:** Track Location

**Primary Actor**: Manager

**Stakeholders and interests:**

Manager: wants to be able to see where a rental car is located to ensure proper use

Rental Company: wants to prevent loss or damage of inventory

**Preconditions**:

* Manager is identified and authenticated
* Selected car has GPS tracking capabilities

**Success guarantee:**

* GPS location for car has been updated
* Manager informed of location update

**Main success scenario:**

1. Manager is presented with eligible cars from inventory system
2. Manager selects car from eligible inventory
3. System sends car information to external GPS system
4. System receives location information from external GPS system
5. System reports location information to the manager, updates car information in inventory system

**Extensions:**

2a. Desired car is not part of eligible inventory

1. Discrepancy is reported by system to Manager

2. Information about car, including rental status, displayed to manager

3. System returns to eligible inventory

3a./4a. External GPS system is unreachable

1. System retries attempt to reach external GPS system

1a. System is able to reach external GPS and continues

1b. System is unable to reach external GPS

1. System logs error

2. System returns error information to the user.

**Special requirements:**

* External GPS system must be failure tolerant

**Technology and data variation list:**

3a. External GPS system must support multiple car brands and transmitter technologies

**Frequency of Occurrence:** Once for every car location lookup

**Functional Requirement:** System should allow coupon codes to be entered for discounted rental rates

**Use Case UC17**: Offer discounts

**Primary Actor**: Customer

**Stakeholders and interests:**

Customer: wants to apply coupon codes for their rental

**Preconditions**:

* Customer has created an account on the rental website
* Customer has access to the rental website

**Success guarantee:**

* Customer is able to enter the coupon codes and obtain a discounted rental rates

**Main success scenario:**

1. Customer sign in to the rental website
2. Customer navigates to the Deals and Promotions section of the website
3. Customer enter the coupon code to the Enter Code input field
4. System indicates that the coupon was successfully redeemed and display the discounted rental rate

**Extensions:**

4a. Invalid coupon code

1. System signals that the coupon is invalid and rejects the coupon
2. Customer is able to re-enter a different coupon to be redeemed

**Special requirements:**

* System allows the customer to redeem maximum of 1 coupon code per rental

**Technology and data variation list:**

3a. Coupon code is entered to the input field by keyboard

**Frequency of Occurrence:** Once for every rental

**Functional Requirement:** System should allow for the creation of custom offers for enterprise customers who have negotiated a custom rate

**Use Case UC18**: Customize Offer

**Primary Actor**: Employee

**Stakeholders and interests:**

Customer: wants to be able to obtain a custom rental rate

Employee: wants to have business deals with the customer

**Preconditions**:

* Customer has created an account on the rental website
* Customer has negotiated a custom rental agreement with the agency

**Success guarantee:**

* Customer obtain custom discounts for their rental

**Main success scenario:**

1. Employee navigate to the registered customer’s profile
2. Employee enters the negotiated rate to the Rental Rates input field in the customer’s profile
3. Customer sign in to the rental website
4. System indicates that the customer had obtained a custom rate for their rental

**Extensions:**

2a. Invalid rental rates input

1. System signals that the rental rate is invalid and rejects the input
2. Rental agency is able to re-enter a different coupon to be redeemed

4a. No custom rate

1. No indication is displayed to the customer
2. Customer is charged with the default rental rate

**Special requirements:**

2a. Rental Rates input field takes input between 0 and 100

**Technology and data variation list:**

2a. Rental rates is entered to the input field by keyboard

**Frequency of Occurrence:** Once for every custom customer

**Functional Requirement:** Users should have the ability to see what cars are at a specific rental location

**Use Case UC19**: Check availability

**Primary Actor**: Customer

**Stakeholders and interests:**

Customer: wants to be able to view available cars

**Preconditions**:

* Customer has created an account on the rental website
* Customer has access to the rental website

**Success guarantee:**

* Customer view available cars at a rental location

**Main success scenario:**

1. Customer sign in to the rental website
2. Customer enter a desired location at the Location input field
3. System redirects to a map with nearby rental locations displayed
4. Customer navigate through the map and clicks on a desired rental location
5. Information about the rental location is displayed on the tooltip
6. Customer clicks on See Cars button
7. System redirects to a list of available cars for rental

**Extensions:**

2a. Invalid location

1. System signals that the location is invalid
2. Customer enters a different location to the input field

7a. No available cars

1. System displays “No Availability” to the customer

**Special requirements:**

* System must display up to date information about the rental availability

**Technology and data variation list:**

2a. Location is entered to the input field by keyboard

**Frequency of Occurrence:** Once for every customer

**Functional Requirement:** The system should flag vehicles and alert employees when a vehicle is approaching a deadline for preventative maintenance

**Use Case UC20**: Notify Maintenance

**Primary Actor**: Employee

**Stakeholders and interests:**

Employee: wants to be notified of future maintenance

**Preconditions**:

* Employee is identified and authenticated
* Maintenance schedule for each vehicle is tracked

**Success guarantee:**

* Employee is notified of maintenance deadline of a vehicle
* Rental vehicle is flagged for maintenance

**Main success scenario:**

1. System flags a vehicle that has a 2 weeks deadline for maintenance
2. System notifies employees that a vehicle is near the maintenance deadline
3. Employee navigate through the list of rented vehicle
4. Employee locate the flagged vehicle

**Extensions:**

**Special requirements:**

4a. Flagged vehicle shows the time remaining until the maintenance deadline

**Technology and data variation list:**

2a. Employees are notified about the flagged vehicle through email and alerts

**Frequency of Occurrence:** Once for every rental vehicle

**Functional Requirement:** The system must allow the user to login with a google or apple account.

**Use Case UC21:** AllowLogin with google/apple account

**Primary Actor**: User

**Stakeholders and interests:**

User: wants to be able to login

Rental Company: wants their customers to be able to create an account on their system

**Preconditions**:

* The user would like to create an account

**Success guarantee:**

* The user could create an account if they link their google/apple account
* The user could not create an account if the account is not valid

**Main success scenario:**

1. Customer goes to register on the rental agency’s website
2. The user chooses an option to register, such as google sign in/apple sign in.
3. Once they register, the system processes the account information, image linked to the account, and saves the user’s information in the database
4. The user is then returned to the website in a signed in state
5. The user then browses the website looking at cars available for rent
6. The user closes their browser tab

**Extensions:**

3a. Invalid account/email

1. Once they choose an option to their account, the system processes the account information, and validates the case that the account is accurate and presents an error message to the user.
2. The user is prompted to login through a different google/apple account
3. Try step 2 in main success scenario again

**Special requirements:**

* The google/apple account must link to a valid account

**Technology and data variation list:**

* Validation is done by using google’s/apple’s feature when an error message is displayed telling that the account information is inaccurate.

**Frequency of Occurrence:** Each time the person wants to login the account.

**Functional Requirement:** The system must allow the user to update their account information.

**Use Case UC22:** Modify account information **Primary Actor**: User

**Stakeholders and interests:**

User: wants to be able to change account information like password, date of birth etc.

Rental Company: wants their customers to be able to change their information to keep it most up to date.

**Preconditions**:

* The user should have an account.

**Success guarantee:**

* The user will be able to update their information.
* The user could not update their information due to invalid input or similar information is added

**Main success scenario:**

1. Customer goes to the account page
2. The user chooses an option to edit next to the field they would like to change
3. Once they update, the system processes the new information, and saves the user’s information in the database
4. The user is then returned to the website in a signed in state
5. The user then browses the website looking at cars available for rent
6. The user closes their browser tab

**Extensions:**

3a. Invalid account information

1. Once they choose an option to their account, the system processes the account information, and validates the case that the account is accurate and presents an error message to the user.
2. The user is prompted to edit the account information to previous or new one
3. Try step 2 in main success scenario again

**Special requirements:**

* The google/apple account information added must be valid

**Technology and data variation list:**

* None

**Frequency of Occurrence:** Based on user’s choice

**Functional Requirement:** The system must allow the user to cancel a rental up to 48 hours before the start of the rental period.

**Use Case UC23:** Cancel rental vehicle

**Primary Actor**: User

**Stakeholders and interests:**

User: wants to be able to cancel their booking.

Rental Company: wants their customers to be able to get cancellation information 48 hours before the start time.

**Preconditions**:

* The user should have a booking in their account.

**Success guarantee:**

* The user will be able to cancel their booking just 48 hours before the start time.
* The user could not cancel their booking since it is less than 48 hours of booking time.

**Main success scenario:**

1. Customer goes to the account page
2. The user looks up their booking information and chooses the cancellation option
3. The system processes the information and checks if there are at least 48 hours before the booking time
4. The user gets confirmation that the booking has been canceled
5. The user is then returned to the website in a signed in state
6. The user then browses the website looking at cars available for rent
7. The user closes their browser tab

**Extensions:**

3a. The booking is not canceled due to the buffer period getting over

1. The user tries to cancel the booking but it is less than 48 hours
2. The user is prompted that the booking cannot be canceled or there might be additional fees.
3. The user proceed with either canceling and paying the fees, or keeping the rental.

**Special requirements:**

* None

**Technology and data variation list:**

* Email confirmation of the cancellation

**Frequency of Occurrence:** Based on user’s choice

**Functional Requirement:** Customers should be able to rent more than one vehicle at a time

**Use Case UC24:** Book multiple cars

**Primary Actor**: User

**Stakeholders and interests:**

User: wants to be able to make multiple bookings.

Rental Company: wants their customers to be able to rent more than one car at a time.

**Preconditions**:

* The user has one booking in the account already.

**Success guarantee:**

* The user will be able to book more than one rental at the same time.

**Main success scenario:**

1. Customer goes to the account page
2. The user looks up their booking information which already has 1 car
3. The user looks up more cars available for rent on the main page or the saved liked cars
4. The user chooses the car they would like to rent and enters all the information
5. The system processes the information and checks if everything is valid
6. The payment is processed and the user gets a confirmation about the rental booking
7. The user can choose to return to the website in a signed in state
8. The user then browses the website looking at cars available for rent
9. The user closes their browser tab

**Extensions:**

3a. Car not available

1. The user chooses a car to rent but it is not available
2. The system prompts the user that this is not available while also displaying additional information of similar suggested cars that are available.
3. Try step 2 in main success scenario again

**Special requirements:**

* There is already a booking in the account
* The system displays suggested cars which are available

**Technology and data variation list:**

* Email confirmation of the booking

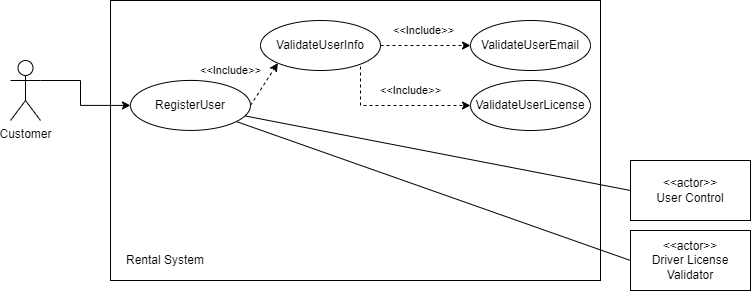
**Frequency of Occurrence:** Based on user’s choice

Use case diagrams

**Use Cases:**

* UC1: Require Driver's license

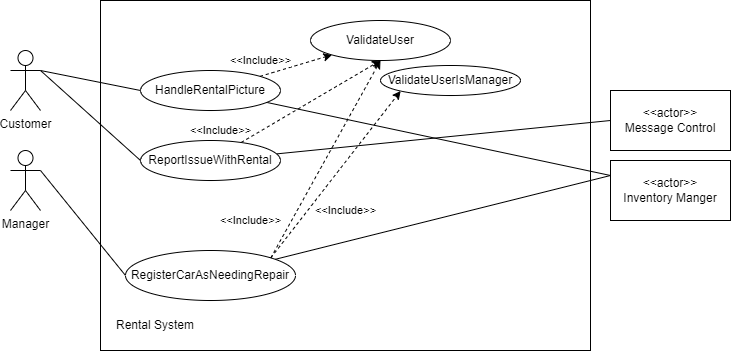
**Use case diagram:**



**Use Cases:**

* UC2: Submit Rental Vehicle Picture
* UC3: Mark vehicle in need of repair
* UC4: Report issue during rental period

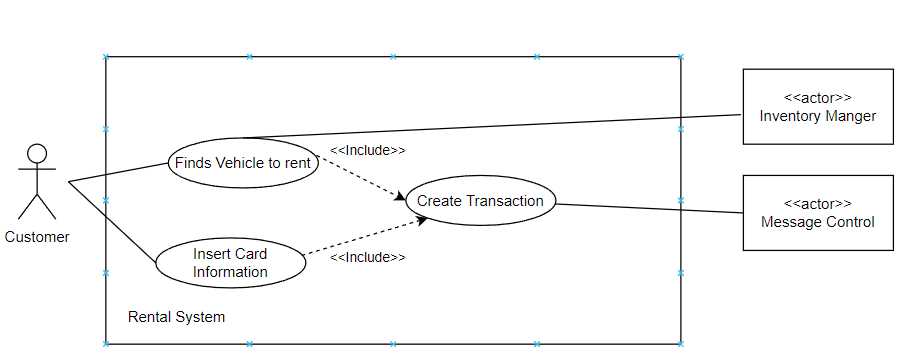
**Use case diagram:**



**Use Cases:**

* UC5: Create transaction

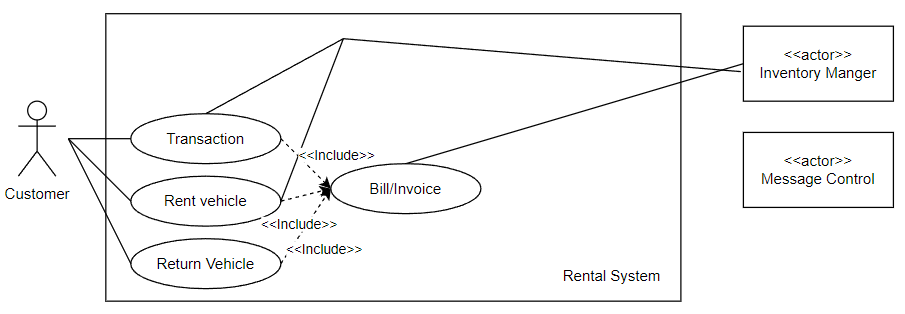
**Use case Diagram**



**Use Cases:**

* UC6: View bill/invoice

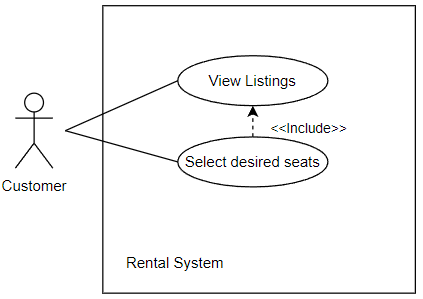
**Use case Diagram**

****

**Use Cases:**

* UC7: Filter vehicles by seat

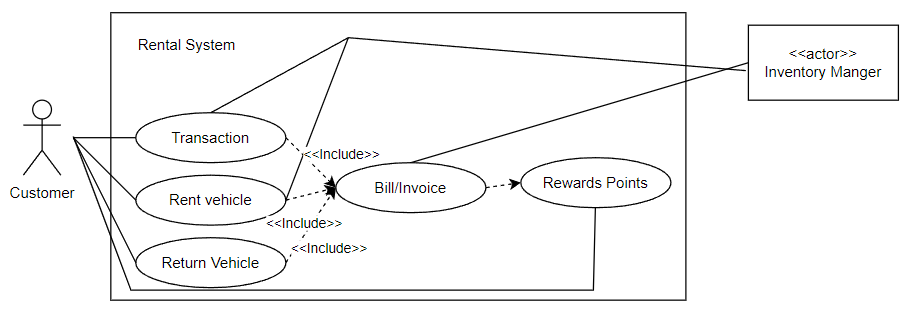
**Use case Diagram**

****

**Use Cases:**

* UC8: Receiverewards

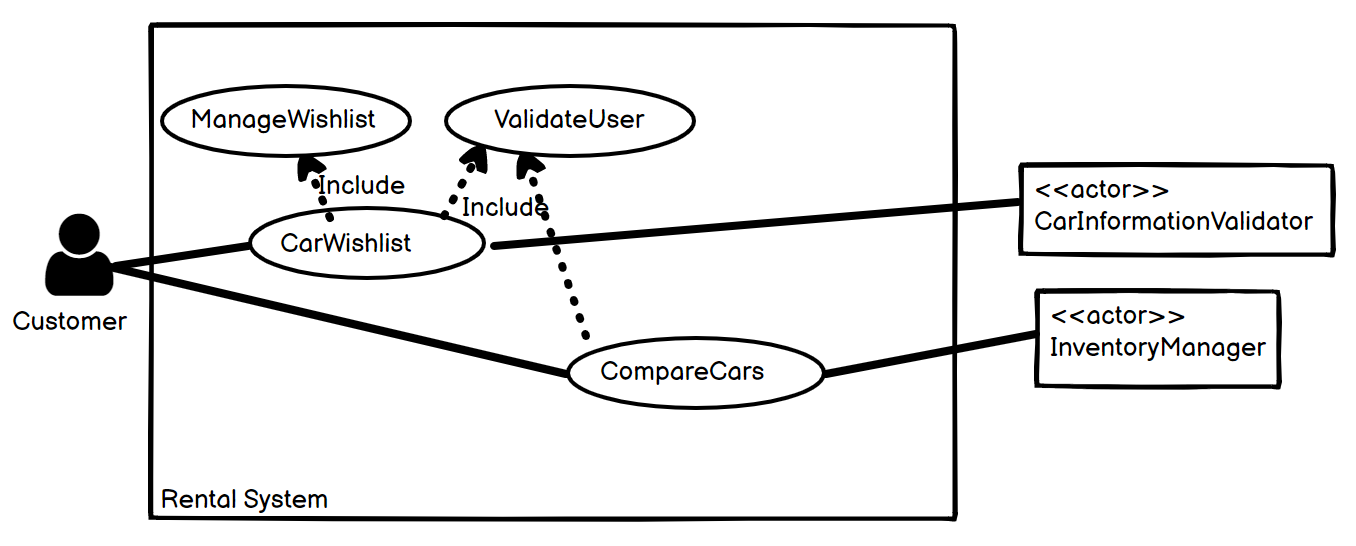
**Use case Diagram**

****

**Use Cases:**

* UC9: Keep car wishlist
* UC10: Compare Cars

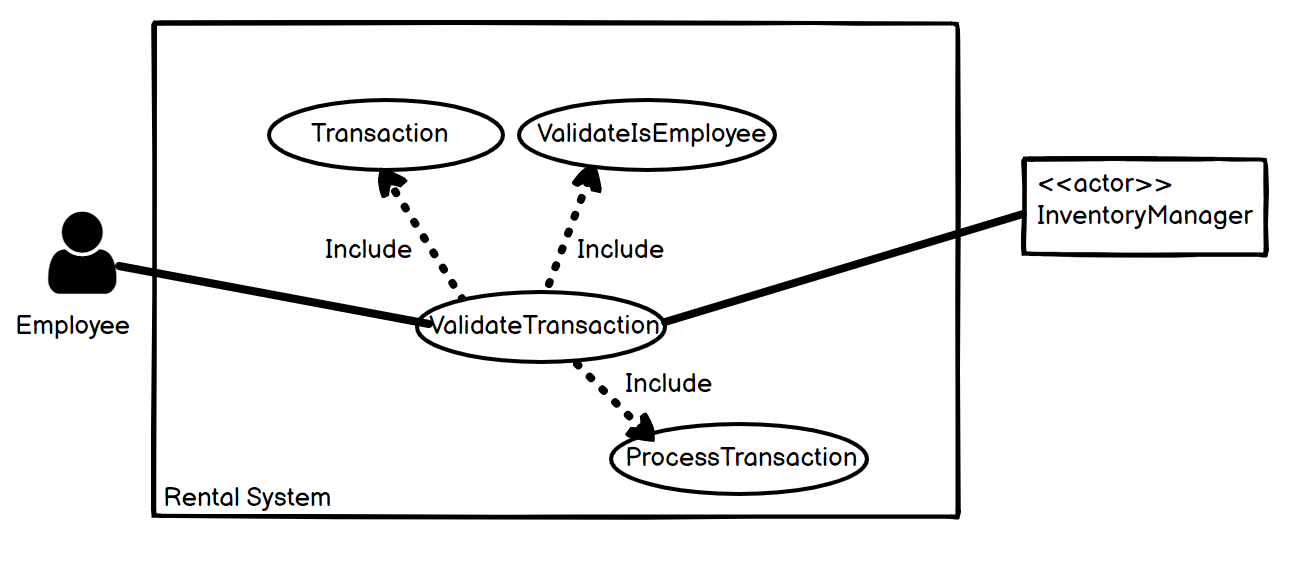
**Use Case Diagram:**

****

**Use Cases:**

* UC11: Approve Transaction

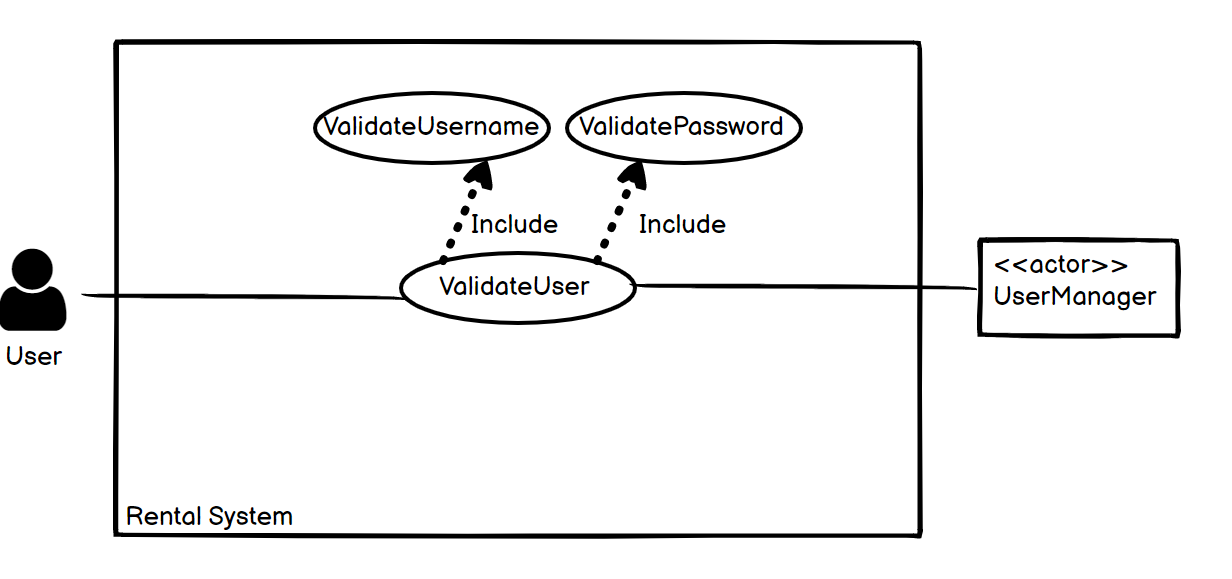
**Use Case Diagram:**

****

**Use Cases:**

* UC12: ValidateUser

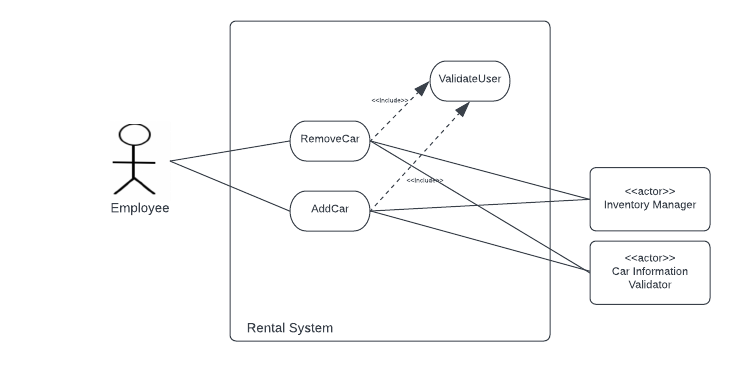
**Use Case Diagram:**

****

**Use Cases:**

* UC13: Add Car
* UC14: Remove Car

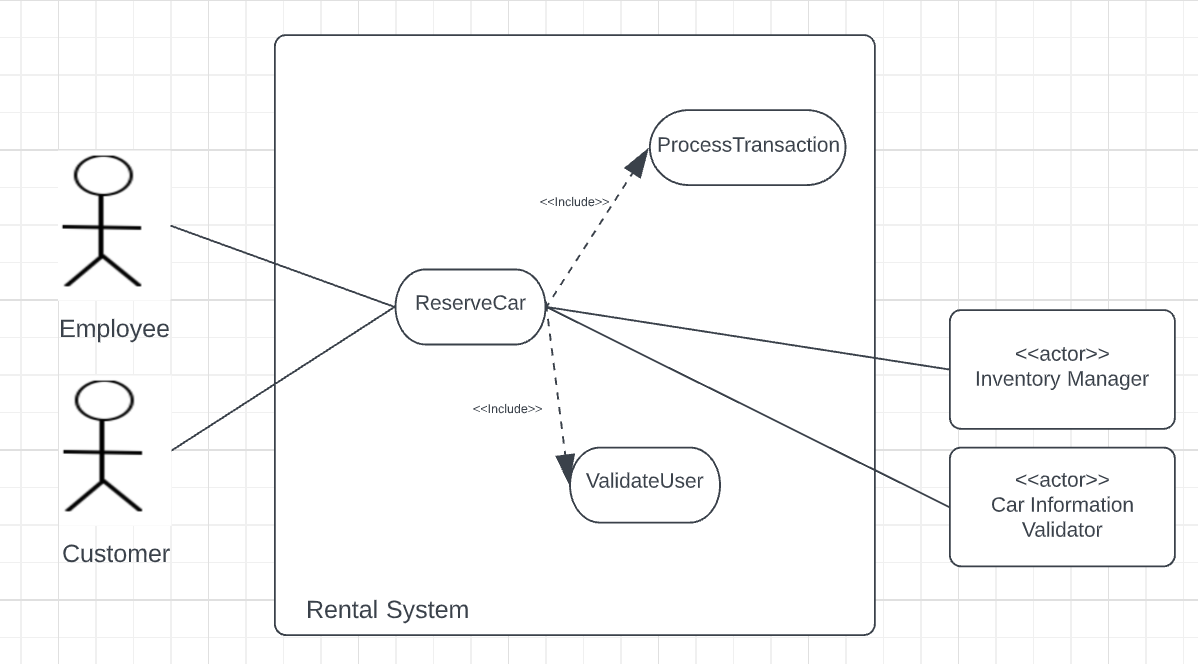
**Use Case Diagram**

****

**Use Cases:**

* UC15: Reserve Car

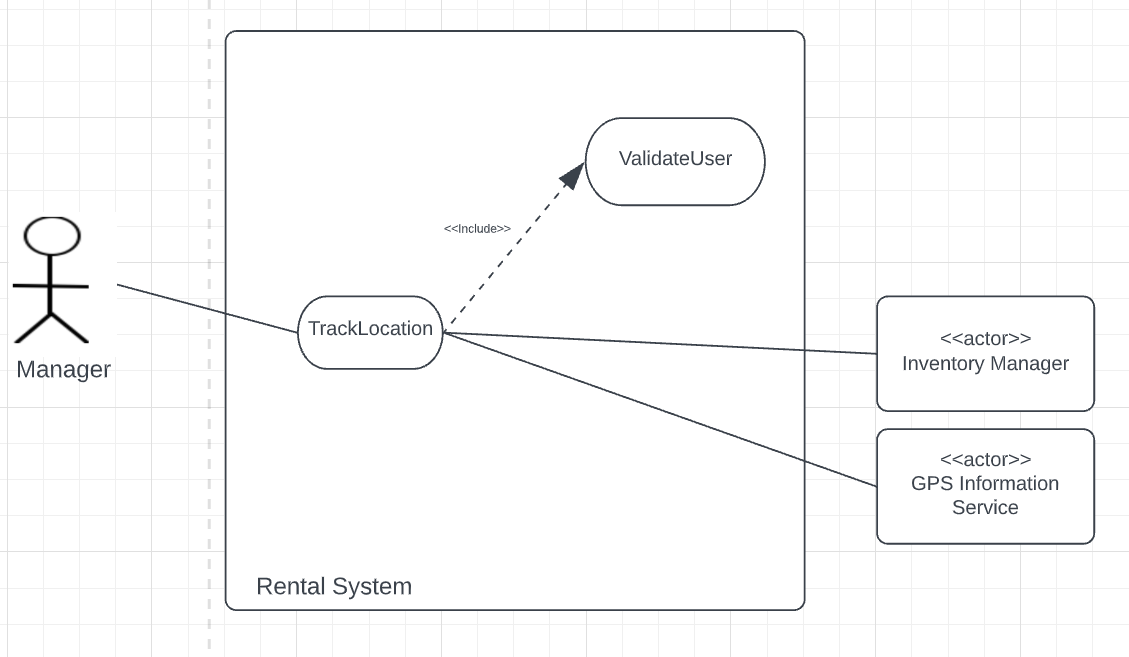
**Use Case Diagram**

****

**Use Cases:**

* UC16: Track Location

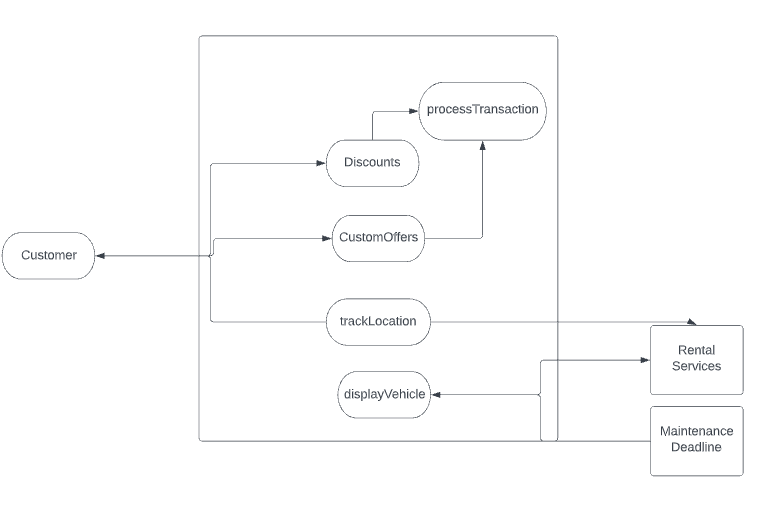
**Use Case Diagram**



**Use Case:**

* U17: Offer Discounts
* U18: Custom Offer
* U19: Check Availability
* U20: Notify Maintenance

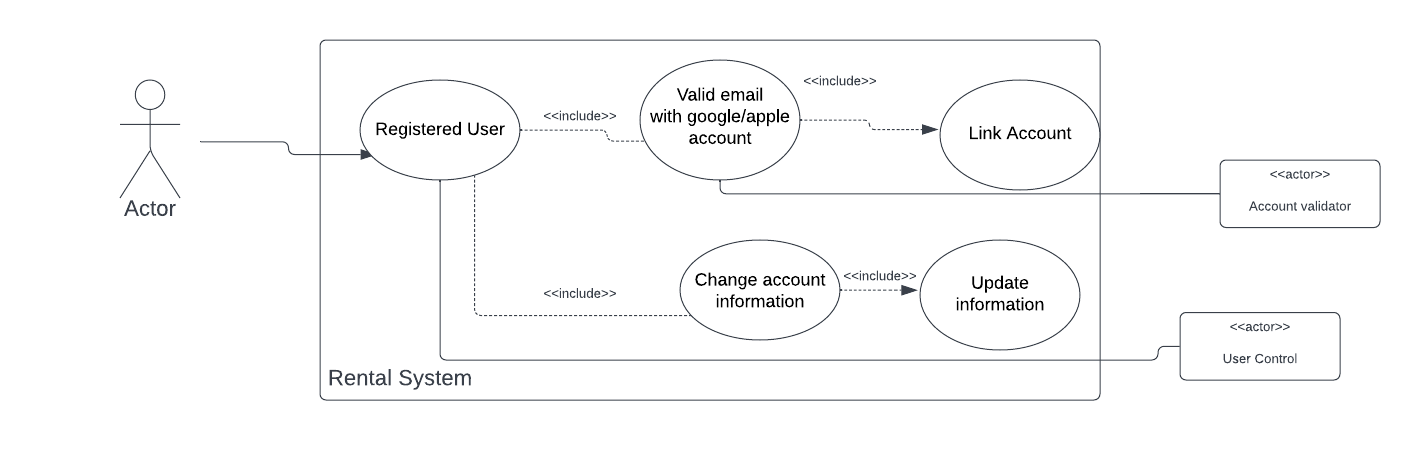
**Use Case Diagram:**

****

**Use Cases:**

* U21: AllowLogin with google/apple account
* U22: Modify account information

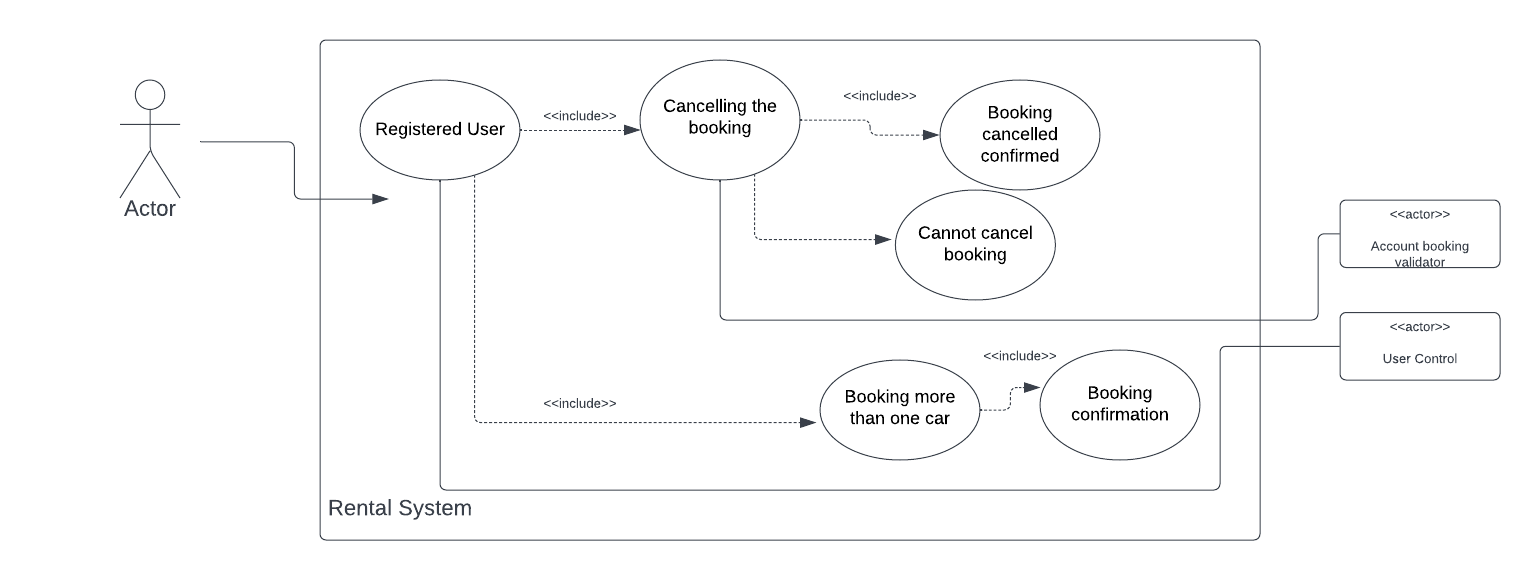
**Use Case Diagram:**



**Use Cases:**

* U23: Cancel rental vehicle
* U24: Book multiple cars

**Use Case Diagram:**

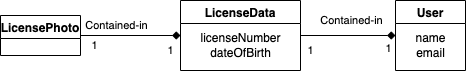


Conceptual class diagrams

**Use Cases:**

* UC1: Require Driver's license

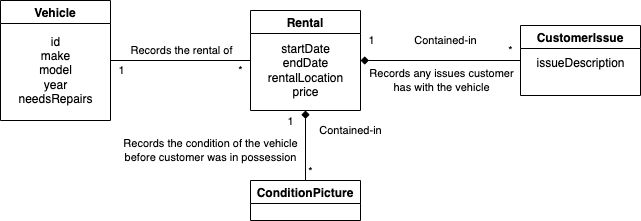
**Conceptual class diagram:**



**Use Cases:**

* UC2: Submit Rental Vehicle Picture
* UC3: Mark vehicle in need of repair
* UC4: Report issue during rental period

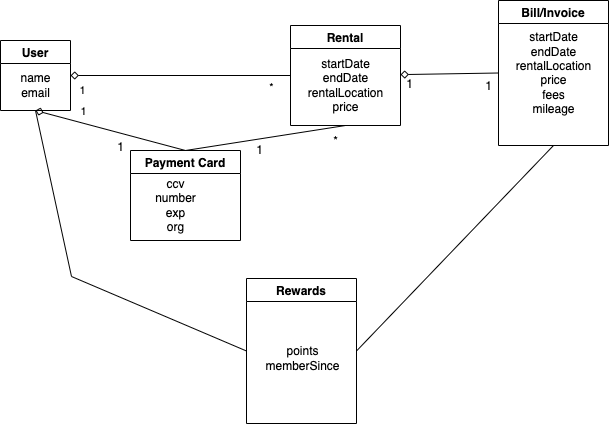
**Conceptual class diagram:**



**Use Cases:**

* UC5: Create a car rental transaction
* UC6: Users should be able to see any fees that they accumulated during their rental such as tolls, gas fees, etc
* UC7: The user should be able to get rewards for renting cars multiple times

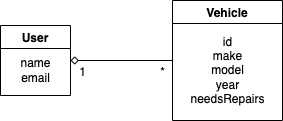
**Conceptual Class Diagram:**



**Use Cases:**

* UC8: Users should be able to filter vehicles by how many seats the vehicle has

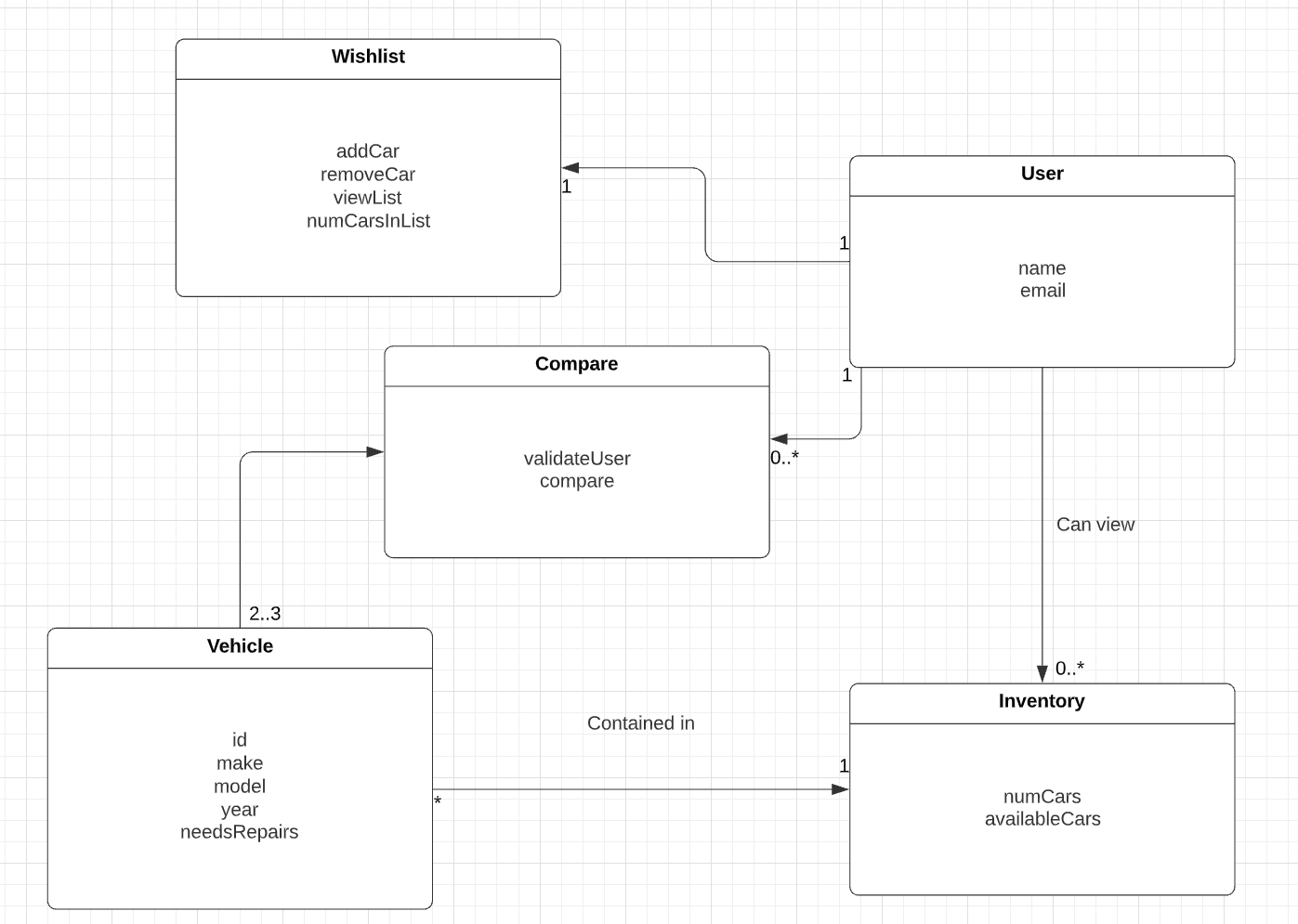
**Conceptual Class Diagram:**

****

**Use Cases:**

* UC9: Users can manage a car wishlist
* UC10: Users can compare cars

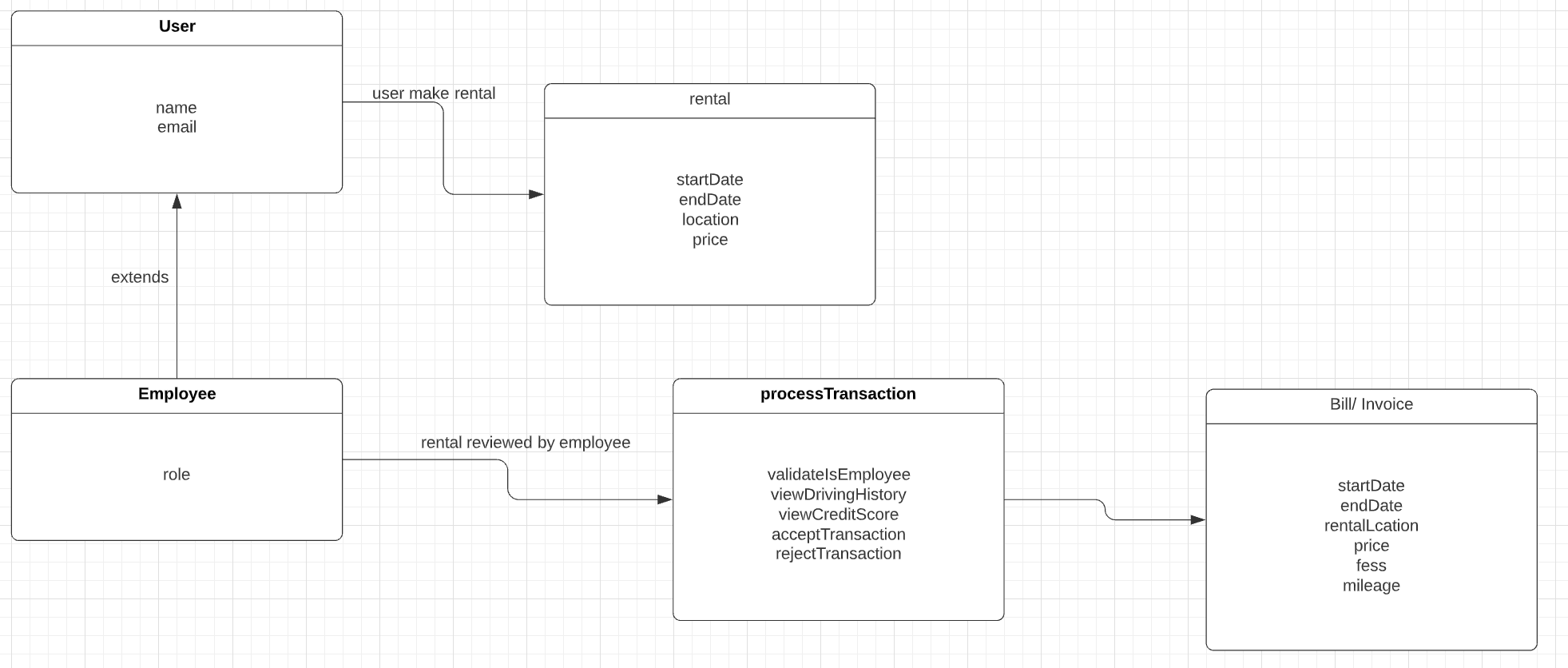
**Conceptual Class Diagram:**

****

**Use Cases:**

* UC11: Approve Transaction

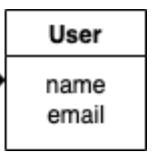
**Conceptual Class Diagram:**

****

**Use Cases:**

* UC12: ValidateUser

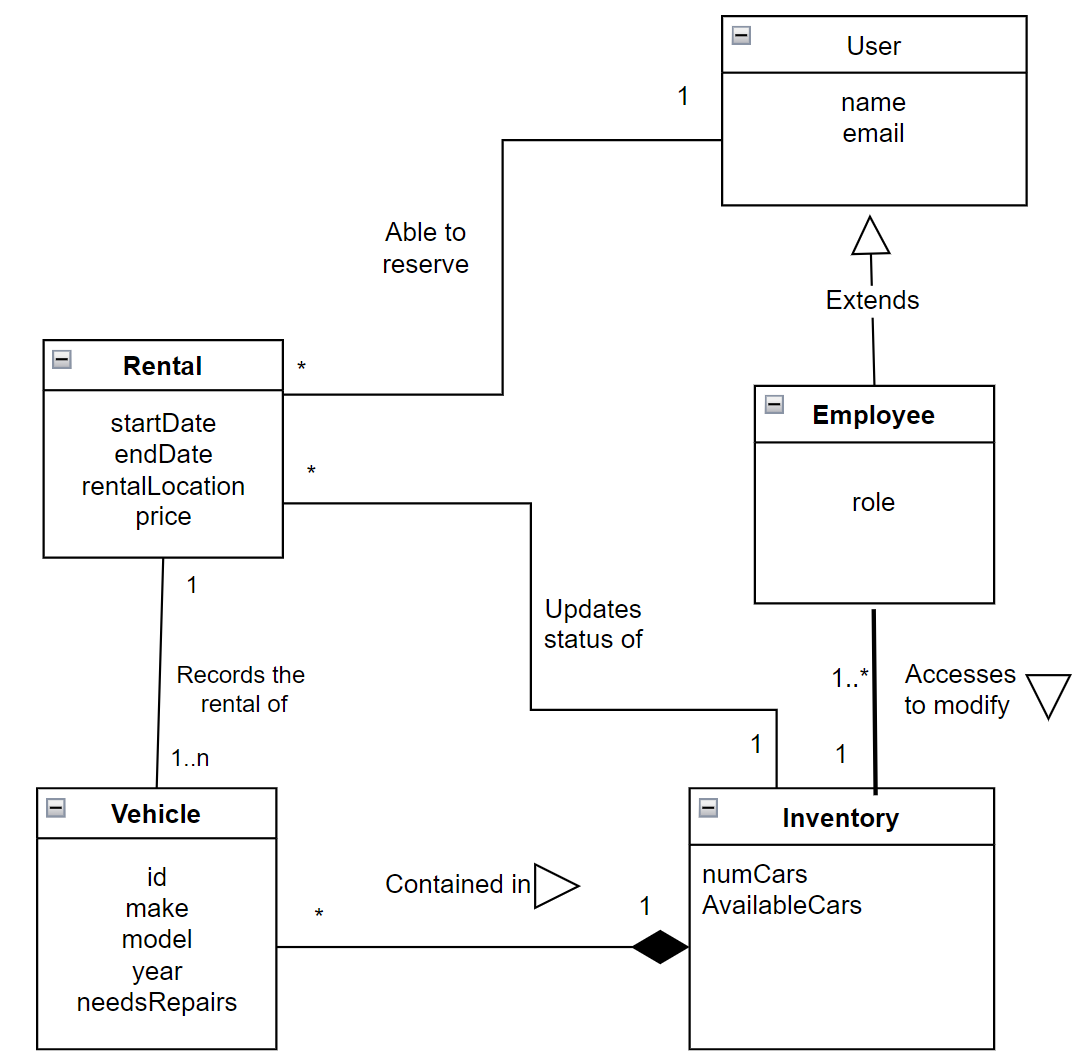
**Conceptual Class Diagram:**

****

**Use Cases:**

* UC13: Add Car
* UC14: Remove car
* UC15: Reserve Car

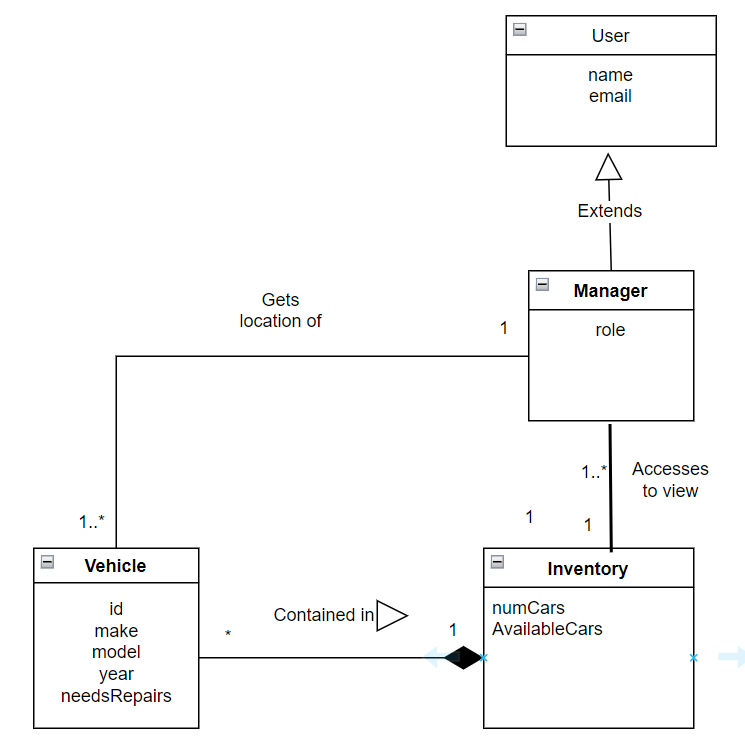
**Conceptual Class Diagram:**



**Use Cases:**

* UC16: Track Location

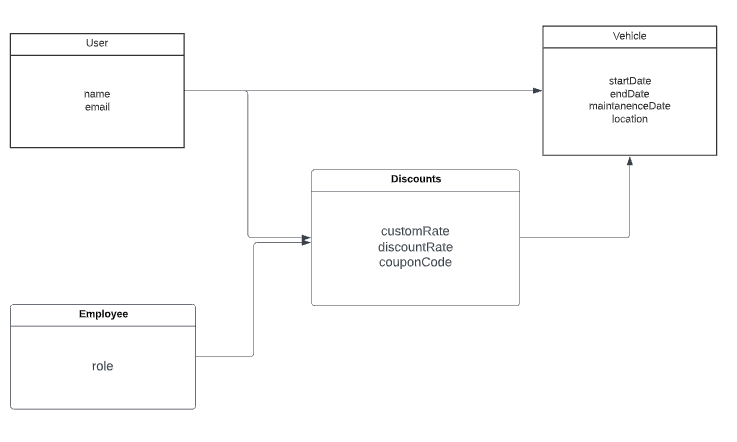
**Conceptual Class Diagram:**

****

**Use Cases:**

* U17: Offer Discounts
* U18: Custom Offer

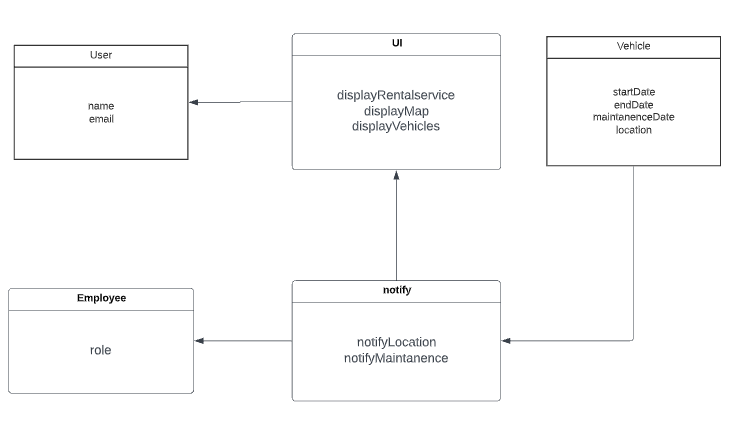
**Conceptual Class Diagram:**

****

**Use Cases:**

* U19: Check Availability
* U20: Notify Maintenance

**Conceptual Class Diagram:**

****

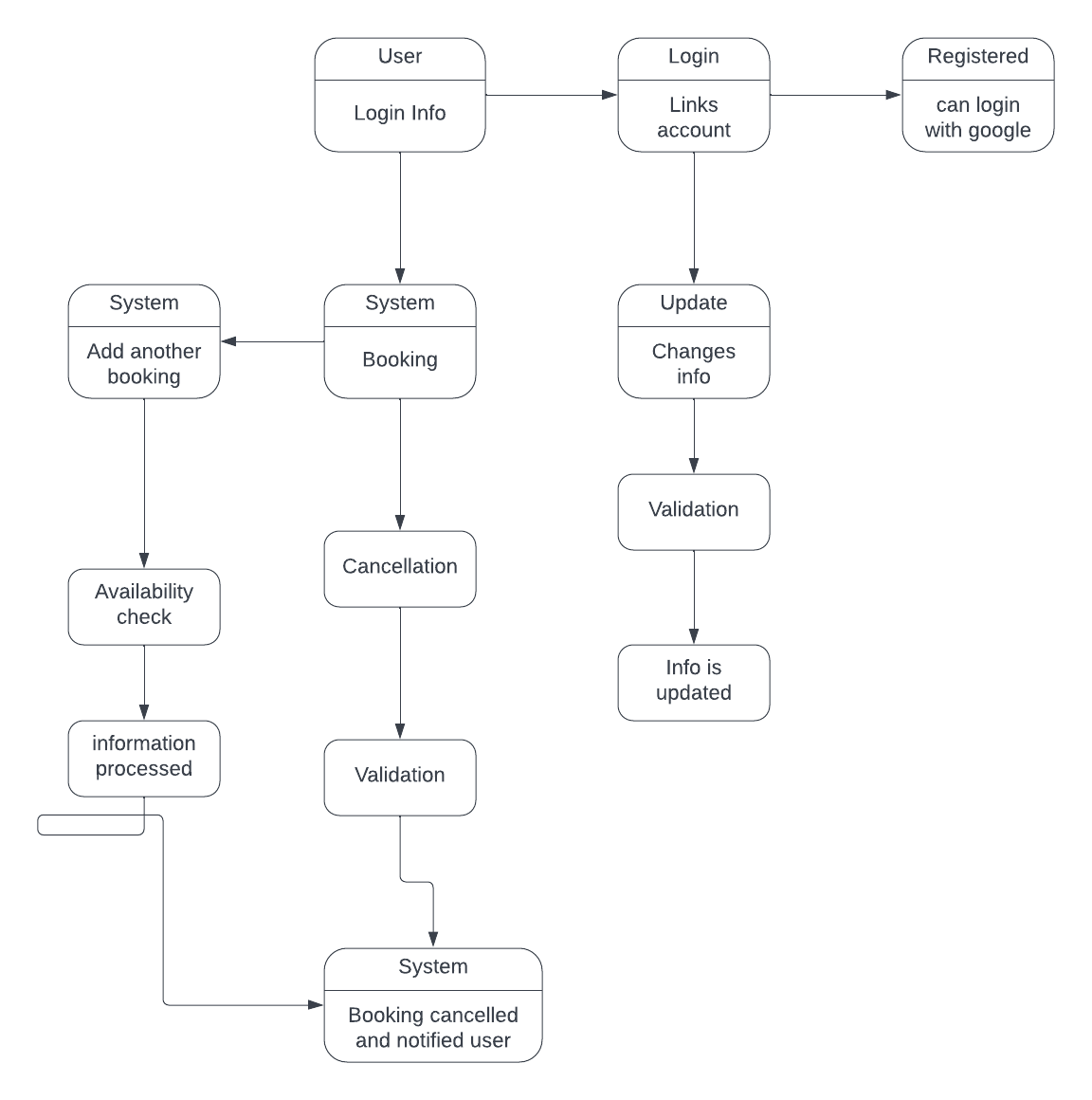
**Use cases:**

U21:Google/apple login

U22: Change information

U23: Booking cancellation  
U24: Multiple bookings

**Conceptual Class Diagram:**



**Supplementary specifications:**Since we want to provide customers with the most up to date data, we have a non-functional requirement that the system must support updating the inventory at least once every 1 minute. This would allow customers to see the most accurate data when looking for a rental vehicle. People that see a vehicle in the rental lot should always be able to see the same vehicle in the system otherwise they would think the system is bad or inaccurate..

Because we are targeting companies with our software, we have a non-functional requirement that the system must support 20 most popular languages in the world. This would allow the rental companies that use our system to reach customers regardless of their primary language, since the United States is home to a diverse population that speak many different languages.

We want our application to be accessible to everyone, so when designing our user interface we will not use any colors associated with colorblindness. Having a website accessible to everyone will not only make our application more user friendly, but it will also make it more desirable to companies who may want to use it.

Our application must be accessible and usable from anywhere, not just the car rental location. Therefore, it must be a primarily browser based application, to allow for use regardless of computer platform or locality for car rental services. This also allows Managers and Employees to check the status of rentals from the lot or while helping a customer, to allow for optimal efficiency and customer service.

To ensure maximal accessibility for both customers and employees, our system must be built to natively support both desktop and mobile browsers. By supporting both platforms, we allow our employees to most effectively manage their bookings while solving technical and accessibility issues. The customer also feels supported with access to applications on multiple devices.

We want the system to be always working to accept new transactions. In order to prevent limiting the number of transactions we need to process, we have a non-functional requirement to ensure that the system is capable of scaling when necessary to support any number of transactions that are thrown at it.