Car Rental System Design Specification

Prepared by: Andressa Wu, Sadia Abdirahim, Trevor Thayer (826194012)

1. **SYSTEM DESCRIPTION:** The purpose of this document is to specify the requirements for a simple software application for a Rental Car System. This system is to be used by both employees and customers to keep inventory, pricing, and payments up to date.

    1.1. Customer/Employee Management: The system should allow all users (both employees and customers) to create and manage personal accounts. For employees, it should consider different information authorization levels based on employee level. Customers should be able to view their rental history and update personal/payment information.

    1.2. Car Management: The system should allow employees with proper authorization to add and remove car models. Each model should have specifications of year, make, model, color, and gas mileage. Customers should be able to search for cars based on filters relevant to car specifications.

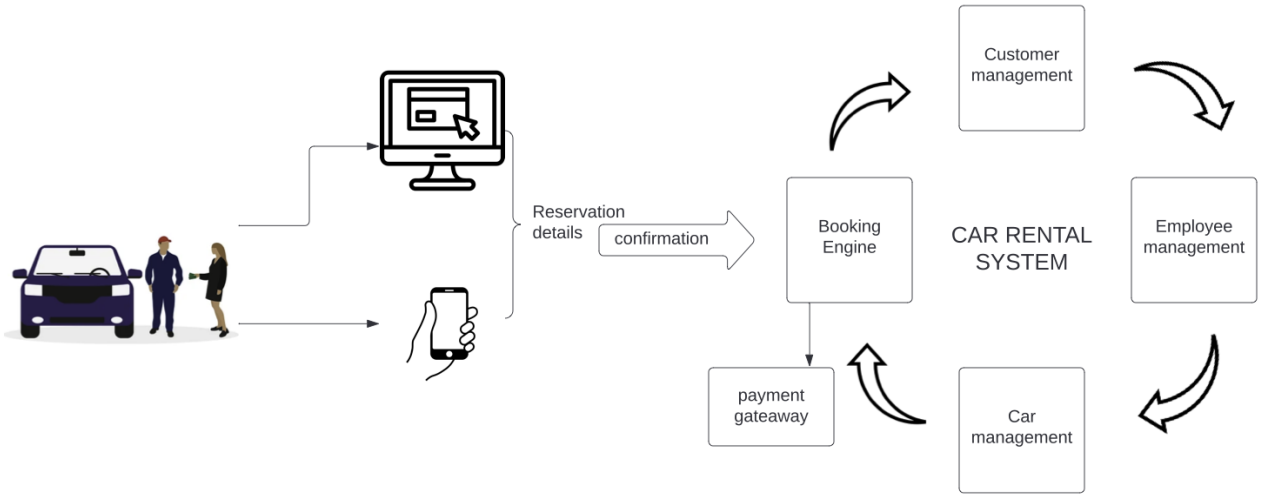
    1.3. Booking Management: The system should allow for reservations to be made in advance with specific dates, as well as in person. It should compare these dates with availability for the specific vehicle requested.

    1.4. Rental Management: Each employee should be able to check out vehicles to customers. They should also be able to take vehicles out of circulation based on damages. During each check in and check out, the system should require an update of date and time, gas level and any damages. The system should then calculate any additional fees based on the time the car was utilized and any damages.

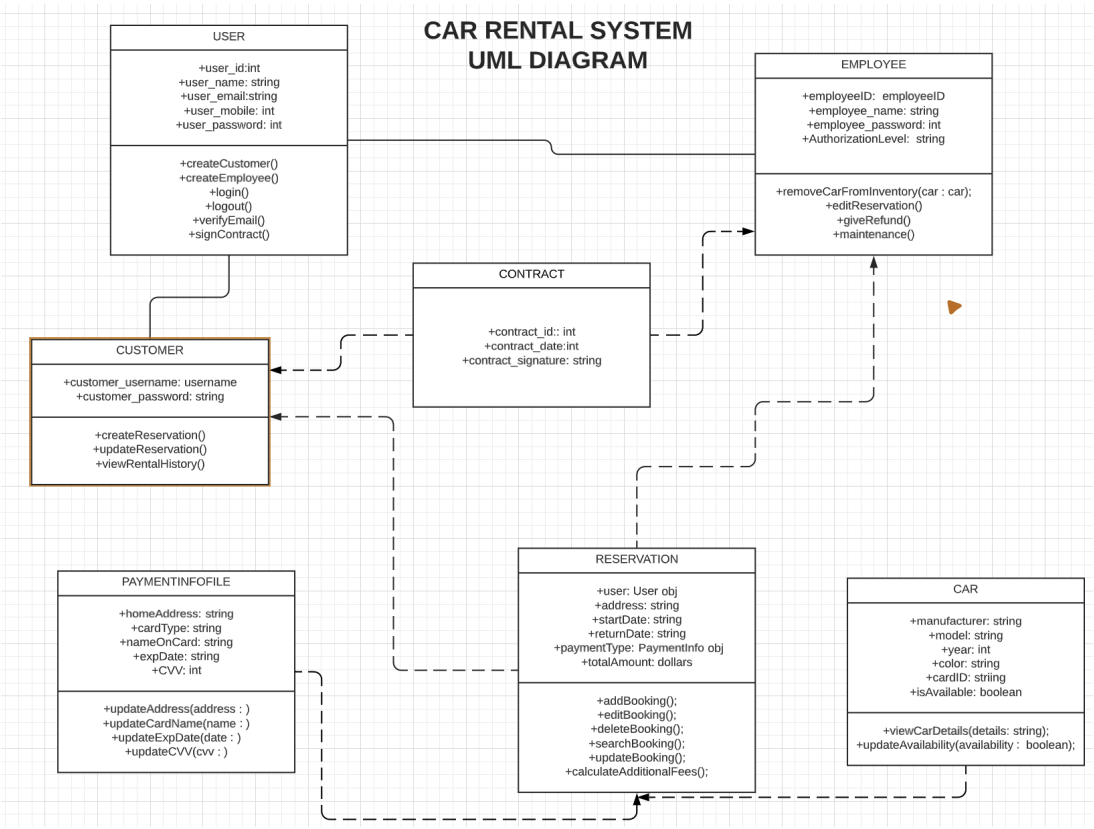
    1.5. Payment Management: The system should allow for payments in advance, as well as in person payments. Cash and card should be options for in person payments. Processing refunds should be handled as well. Each payment should be added to the account of the customer, as well as a record of all payments received.

2. SOFTWARE ARCHITECTURE OVERVIEW

2.1. Architectural Diagram of All Major Components:



2.2. UML Class Diagram:



2.3. Description of Classes/Operations/Attributes:

2.3.1. USER

The User class consists of both the Employee and Customer class. Its attributes contain a user ID, a username, an email, their mobile number, and their password. These attributes specify information about the customer and/or the employee. Operations representing the User class include createCustomer, createEmployee, login, logout, verifyEmail, and signContract.

2.3.2. CUSTOMER

The Customer class is associated with the User class because the User can either be a customer or an employee. The attributes of this class are customer_username that returns a username type and customer_password that return a string type. These two attributes let the Customer log in to the website or the app. The operations of this class are createReservation, updateReservation and viewRentalHistory. These operations allow the user to modify the reservation and also look for rental history.

2.3.3. EMPLOYEE

The Employee class is included in this system as another form of the User class - to be implemented separately from the Customer class. This is important because they need employeeID's and authorizationLevels for the system to determine what actions they can do. Employees should be able to request maintenance or remove cars from the system in case of damages or product recalls. They should also be able to edit reservations and give refunds upon request by customers.

2.3.4. CONTRACT

The Contract class is simply utilized as an agreement between the rental company and the customer to ensure liability is properly handled. This requires an identification number for the contract, as well as a date, and a signature from both parties. The User class ensures that customers and employees both have the ability to sign the contract.

2.3.5. RESERVATION

The Reservation class enables the process of booking an appointment for the vehicle. This class is utilized by both the employee and customer, and depends on the PaymentInfoFile class. This ensures that the payment information is provided prior to booking a reservation. The attributes of the reservation class include an object of the user class, the address of either the customer or the employee, the startDate of when the vehicle will be purchased and it's counterpart, the returnDate of the car rental. Other attributes entail a paymentType (object of the PaymentInfoFile class) and the totalAmount paid in dollars. The operations of this class include addBooking, editBooking, deleteBooking, searchBooking, updateBooking, and calculateAdditionalFees.

2.3.6. PAYMENTINFOFILE

The PaymentInfoFile class signifies the information provided by the card holder. Its attributes include the home address, type of card, the name provided on the card, its expiration date, and the CVV (3 digit code on the back). The operations of this class allows the user to update four entities which include their address, card name, expiration date, and their CVV.

2.3.7. CAR

The Car class depends on the Reservation class which also depends on the Employee class and the Customer class. In order to access the Car Class the customer must first make a reservation so they can choose the car and the employee can look at the reservation and the car details. The attributes of this class are manufacture, model, year, color, carID, isAvailable. The return types are string, int and boolean. These attributes give the customer/employee some information about the car. The operations are viewCarDetails and updateAvailability. These operations are behavioral features that allow the User to view or update some information about the car.

3. DEVELOPMENT PLAN AND TIMELINE

3.1. Partitioning of Tasks/Team Member Responsibilities:

Trevor - System Description, Description of Classes, UML diagram.

Andressa - Architectural Diagram, Description of Car and Customer class and UML Diagram.

Sadia - Description of User Class, Reservation class, PaymentInfoFile class, and UML diagram