Design a Car Rental System

Requirements :

- User can login to the car rental website
- User can search, select cars based on different parameters(location, car brands, rental price, reviews)
- User can check availability(pick-up date, drop off date, pick-up time, drop off time)
- Book car
- Cancel booking

```
Class User
Data: username, password , phoneNumber, email, creditCard, validLicense
Behavior: loginToCarRentalWebsite(){
                CarRentalWebsite.login(email,password);
           }
Behavior: searchCars() {
                 CarRentalWebsite.viewListOfCars(pickUpTime, dropOffTime, requiredCarFeatures)
         }
Behaviour : bookCar(){
                CarRentalWebsite.bookCar(carID,pickUpDate, pickUpTime, dropOffDate, dropOffTime,numberOfDays);
             }
Behavior :  cancelBooking(bookingID){
               CarRentalWebsite.cancelBooking(bookingID);
            }
```

```
Class : CarRentalWebsite
Data : websiteName, List<Car> availableCars, List<CarRentalProvider> providersList,

Behavior : login(email,password){
               if(email and password match){
                System.out.println("Successfull Login");
               } else {
                  System.out.println("Incorrect email or password");
                 }
            }
Behavior : viewListOfCars(pickUpDate, pickUpTime, dropOffDate, dropOffTime, requiredCarFeatures){
             foreach(carRentalProvider in providersList) {
                 foreach(car in carRentalProvider.inventory()) {
                    if (car.featuresCompare(requiredCarFeatures) && car.isAvailable(pickUpDate, pickUpTime, dropOffDate, dropOffTime)) {
                                availableCars.add(car);
                         }
                 }
            }
            return availableCars;
         }
Behavior: bookCar(carID,pickUpDate, pickUpTime, dropOffDate, dropOffTime,numberOfDays){
            //get a car object from carId
              Car car;
            //calculate the total price
               float totalprice = (car.price + insurancePrice) * numberOfDays;
            //get creditCardDetailts
              if(CreditCardDetails.checkCardValidation()){
                    Payment.makePayment(totalprice);
                    //generate bookingID
                    return bookingDetails;
              } else {
                System.out.println("Card Expired");
               }
         }

Behavior : cancelBooking(bookingId){
              //fetch user object and totalprice that user payed
              Payment.refund(user,price)
          }
```

Class CarRentalProvider

Data : providerName , phoneNumber, email , List<Car> cars, pick-up date,pick-up time

```
Behavior :  confirmBooking(){
               if ( cars are available for the pick-up dates and time){
                 System.out.println("Booking is confirmed");
               } else {
                  System.out.println("Not Available");
                 }
             }
Behavior: inventory() {
                  return this.cars;
        }
```

```
Class Car
Data : carID, licenseNumber , carFeatures , carModel, price
Behavior: isAvailable(pick-up date , pick-up time,drop-off date, dro-off time){
               if(available){
                 return true;
               } return false ;
             }
Behavior: featuresCompare(requiredCarFeatures) {
                    if (this.carFeatures == requiredCarFeatures) {
                              return true;
                    } return false;
          }
```

```
Class CreditCardDetails
Data : username , cardNumber , expiryDate , bankName , cardType
Behavior : checkCardValidation(){
                Date today = new Date(today's date);
                if(today < expiryDate){
                  return true;
                }
                return false;
             }
```

```
Class Payment
Date : amountToBepayed , balanceCredit
Behavior : makePayment(float amount ToBePayed){
               //user can make payment if he/she has balance in his/her account
                 if(amountToBepayed <= balanceCredit ){
                   //user makes payment
                    System.out.println("Successfull payment");
                  } else {
                     System.out.println("Declined payment");
                   }
               }
Behavior : initiateRefund(){
                //initiates the refund once booking cancelled
             }
```

```
Class CarInsurance
Data : insuranceCompanyName , validDuration , price , ageLimit,
coverageDetails  // for eg : accident , stolen or damage coverage etc
Behavior : provideInsurancePolicy(){
                    if(user has valid driving license){
                      if(ageLimit >=18 && ageLimit<= 25 ) {
                        System.out.println("price is $50");
                      } else if (ageLimit >25 && ageLimit<= 50 ) {
                          System.out.println("price is $30");
                        } else if (ageLimit >50 && ageLimit<= 70 )  {
                            System.out.println("price is $50");
                          } else {
                              System.out.println("Sorry, won't provide insurance);
                            }
                   }else {
                       System.out.println("user don't have valid driving license");
                    }
              }
```