Car-Rental System

Phase I is due on 20/10/2019
Phase II is due on 17/11/2019

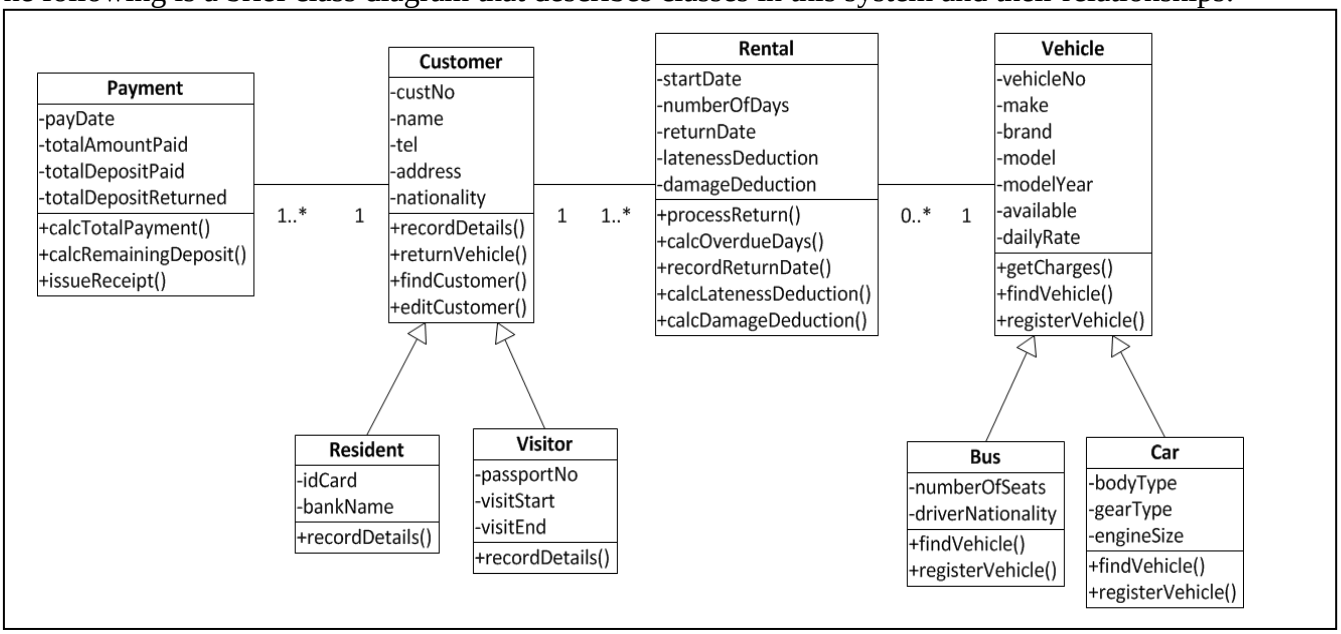
It is required to develop a Java application for a car-rental company. The company has two types of vehicles, which are available for rent: cars and buses. New customer data are entered when a renting process starts. There are two types of customers: residents and visitors. For the sake of simplicity, both citizens and resident foreigners are considered as residents. When a customer rents a vehicle, a rental object is created that records details of the rent, such as start date, number of days … etc. The Payment class represents payments' details such as pay date, total amount paid, etc.

Renting Process:
Customer details (see the class diagram) must be recorded. For resident customers the company needs to record additional details such as bank name and the customer's ID card number. For visitors, the passport number and the visit dates are recorded. Visitors must pay the car rent in advance for the number of days they request. In addition, visitors must pay a deposit amount equal to QR 15,000, which will be refunded when they return the rented car in a good condition. Visitors can only rent cars (not buses). When the car is returned by the visitor, any damages are evaluated and the cost of repairs is deducted from the deposit amount. The remaining deposit is returned to the customer.

For resident customers no deposit is needed as long as the customer gives details and a copy of his/her credit card. The payment is collected when the car is returned. Again, damages' value is deducted directly from the credit card account.

For all customers, if the vehicle is returned beyond the expected return date, the rent charge for each of the additional days is calculated as 1.5 times the regular daily rate.

The following is a brief class diagram that describes classes in this system and their relationships.



This is a team-project to develop the required system. The system must provide the following functionalities: (*all these functionalities are to be finished in Phase I*)

1- Add a new vehicle to the system (car or bus).
2- Add a new customer to the system (resident or visitor).
3- Rent a car with the above-mentioned restriction (only residents can rent buses). Receive payment and issue a receipt to the customer. A rented vehicle should be marked as unavailable.
4- Return a vehicle. When a vehicle is returned, its availability is updated. The return date is noted, and if the return is overdue, additional cost is calculated and deducted from the deposit or credit card account. Damages are also evaluated and the remaining deposit is returned.
5- Inquire about specific vehicles; it should print the vehicle details, its availability, and the expected return date, if already on rent.
6- Inquire about all rented vehicles and their due dates.
7- Inquire about customers.
8- Produce detailed tabular lists of all vehicles and all customers.
9- Produce detailed tabular lists of all rented vehicles names of customers who rented them and expected dates of returns.

Your system should have GUI interface to interact with system users and should use files to save data permanently. (*This requirement is to be finished in Phase II*)

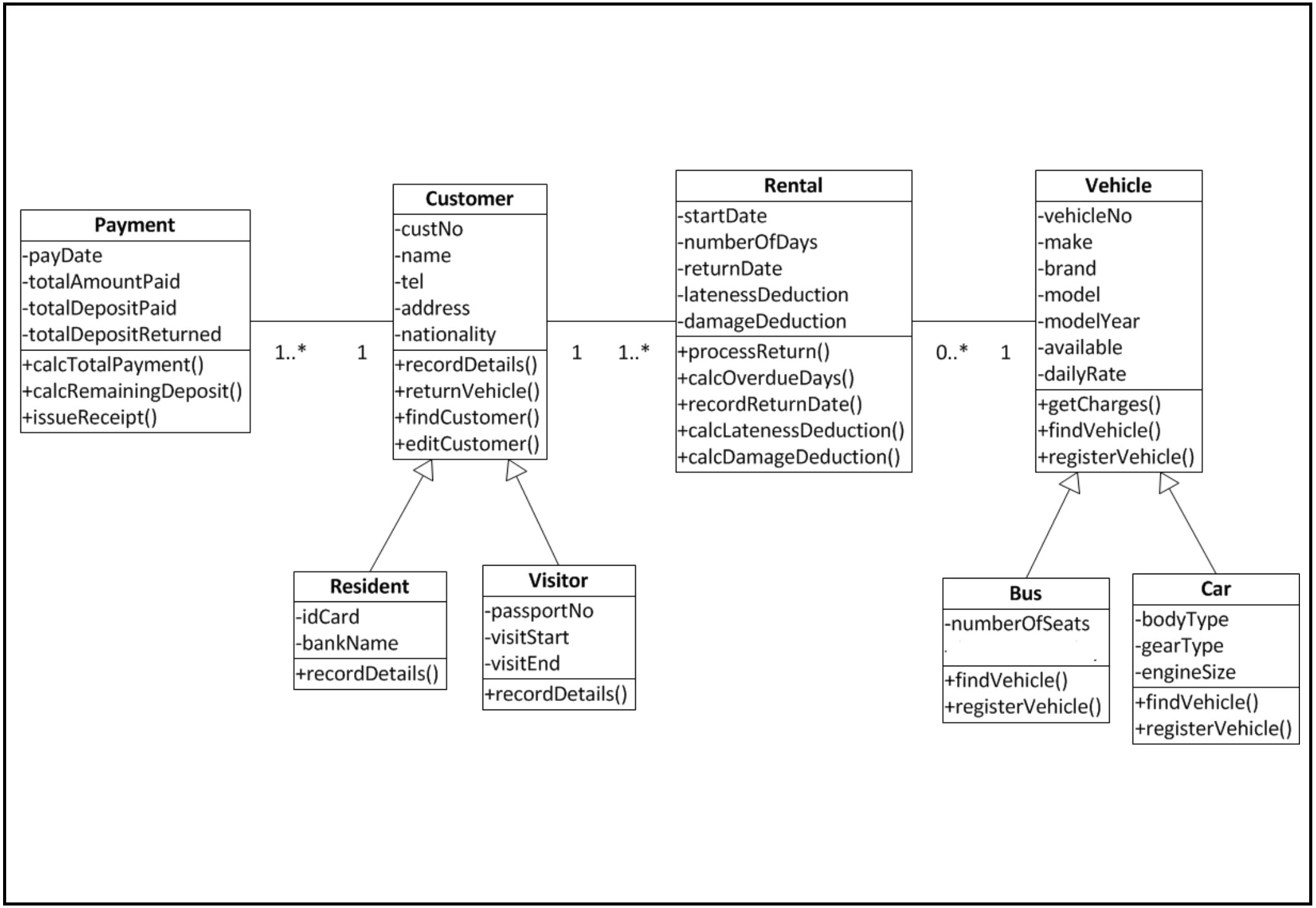
Instructions:

1. This is a group-work project (3 students per group, exceptionally a group of 4 may be allowed).
2. Please immediately form your group and inform your instructor by sending names of the team members by email, also register your name on one of the Blackboard groups on your course website.
3. Please start immediately and plan your time so that you finish within the project period. Submissions after the due date will incur a 25% penalty for every day or part of a day.
4. Name of the member who created a class must be written as the author using Javadoc comments. Also, add the date of creation of the class. The version number must also be included.
5. Each group should submit one softcopy of their work by uploading an archive of their project folder to the course website on Blackboard.
6. Also, upload an MS-Word document containing a cover page that lists your names and QU ID's. In addition, the document should contain the source code of all the developed Java classes (font size should be 12 points, please) and screen shots of all input/output. Include in your document, all the web documents generated using Javadoc tool (copy/paste from the browser window.)
7. Each member MUST submit a one-page confidential report (called the self-report) describing his/her exact contribution to the project and explaining the advantages and disadvantages of working in teams, based on your current experience with this project.

Evaluation criteria:

The breakdown of the project grade is as follows:

- 30% for the functionalities required above, and the report that shows the code and the results of all the test cases. This includes correct indentation and formatting of your code, use of meaningful variable name, abiding by Java naming conventions etc. (Phase I)
- 20% for authoring and producing the Javadoc documentation+Phase1 report (Phase I)
- 25% for adding a GUI to the application. (Phase II)
- 15% File management (saving, loading, updating) (Phase II)
- 10% for presentation of work (Phase II)


Payment
-payDate
-totalAmountPaid
-totalDepositPaid
-totalDepositReturned
+calcTotalPayment()
+calcRemainingDeposit()
+issueReceipt()
1..*
1
Customer
-custNo
-name
-tel
-address
-nationality
+recordDetails()
+returnVehicle()
+findCustomer()
+editCustomer()
1
1..*
Rental
-startDate
-numberOfDays
-returnDate
-latenessDeduction
-damageDeduction
+processReturn()
+calcOverdueDays()
+recordReturnDate()
+calcLatenessDeduction()
+calcDamageDeduction()
0..*
1
Vehicle
-vehicleNo
-make
-brand
-model
-modelYear
-available
-dailyRate
+getCharges()
+findVehicle()
+registerVehicle()
Resident
-idCard
-bankName
+recordDetails()
Visitor
-passportNo
-visitStart
-visitEnd
+recordDetails()
Bus
-numberOfSeats
+findVehicle()
+registerVehicle()
Car
-bodyType
-gearType
-engineSize
+findVehicle()
+registerVehicle()