1. (5 pts) Provide the problem requirements and the conceptual model in UML for your project. You can reuse the ones made in Project 1.

Problem statement:

We aim to develop a web-based management system for a car rental company. It has two main functionalities:

1. Facilitate the rental company to manage their car fleet. For example: the rental company can add a new car to their fleet, update a car's rental price and other information, or delete an existing car from the fleet.
2. Provide a powerful tool for the rental company to monitor their business activities in all aspects. For example: the system can display all the cars owned by the rental company, fleets at different locations, amount of transactions per month, number of customers etc. By closely monitoring their business data, the rental company can adjust business strategies based on performance as well as gain a better understanding of their customers in order to promote the quality of service.

Specially, the application will support:

1. CRUD car entity (add new car to fleet, update car rental price, delete car from fleet)
2. CRUD customer (add new customer, update customer info, delete existing customer)
3. Query/filter cars that company owns by car type, year of service, mileage etc.
4. Query/filter car rental transactions and filter by date range, car type, location
5. Query/filter customer information by car type, rental history
6. Yearly statistics (total car rent, total number of transaction, monthly trend)

Business rules:

1. The rental company owns many different rental locations.
2. Each rental location can manage multiple different cars, but one car can only be located at one location at a time.
3. Customers rent a car, which they pickup from one rental branch and return it at another rental branch (not necessarily the same).
4. Every car can be rented by many different customers at different times.
5. Each customer can rent many different cars at different times.
6. Each customer can conduct multiple transactions with the rental company, but each transaction can only belong to one customer.
7. Each car has its own car make and car model info.

Nouns:

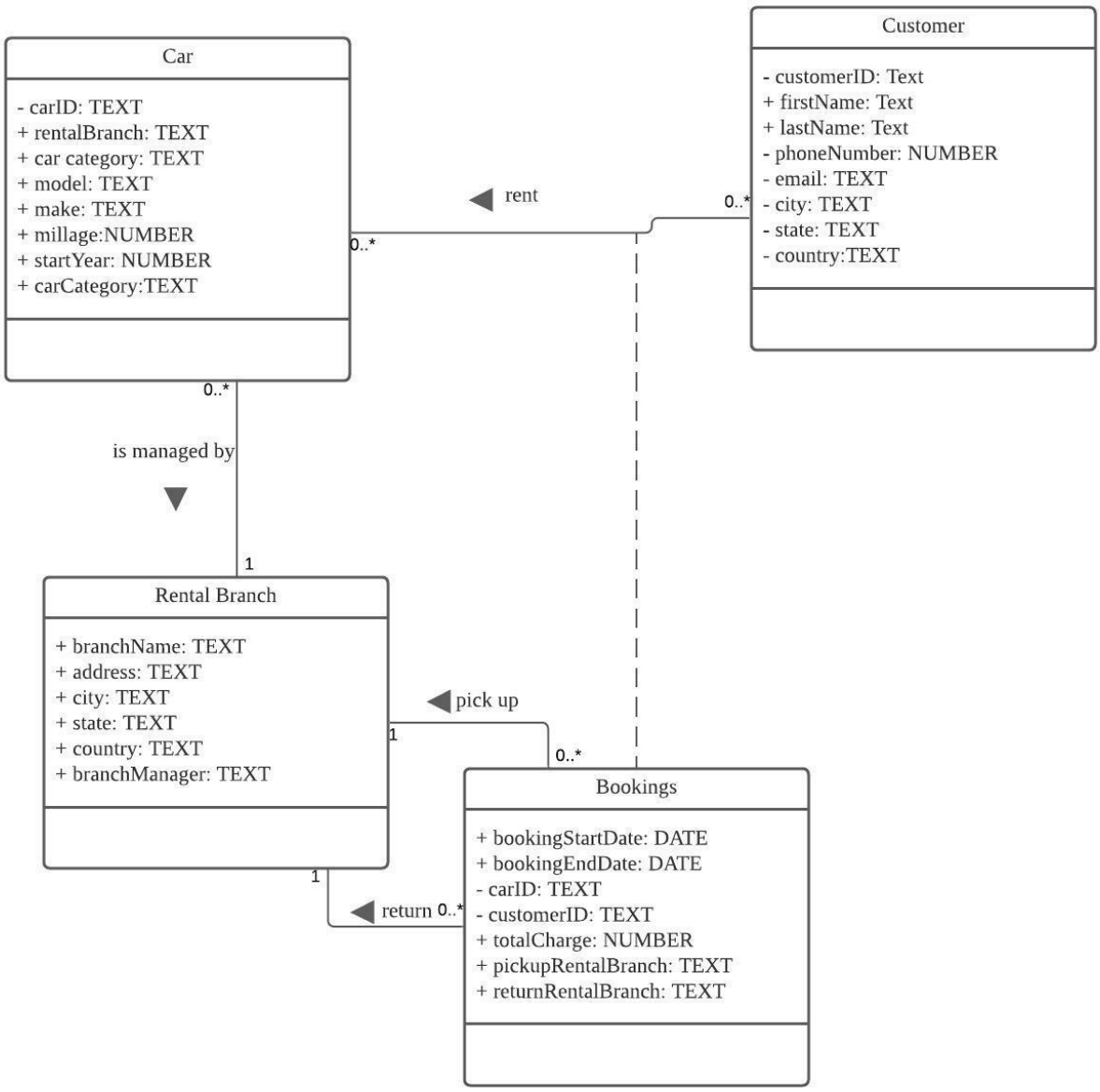
Car, Rental Branch, Customers, Bookings, Car Make, Car Model

Verbs:

Owns (rental branch), manage (cars), rented (by customers), rent(car), conduct (Bookings), has (car make/car model)

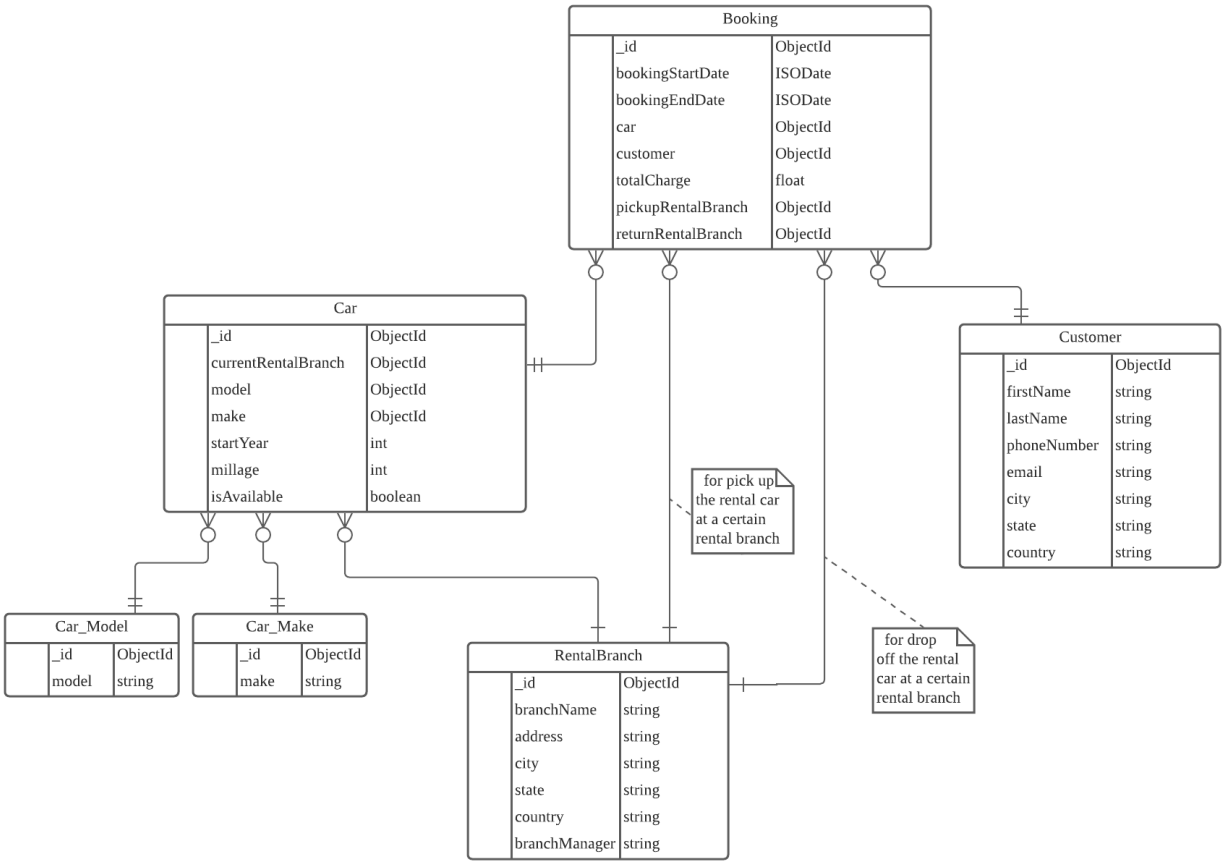
UML Diagram:

https://lucid.app/lucidchart/722ed355-65d1-47df-9cc8-3ba5faa57deb/edit?beaconFlowId=341BAFE5E164E51C&invitationId=inv_6ba983b6-83d2-4e74-866d-5fbff519bbd8&page=0_0#



2. (*15 pts*) Adapt the Logical Data model from your Project 2 to have hierarchical tables. This is, main (root) tables from which all the other tables relate to. This main tables will become later

your Mongo Collections. From your main tables you can have aggregation/composition, one to many and many to many relationships.



3. (*10 pts*) From this logical model define the main Collections (Documents/Tables) you will be using in your Mongo Database. Provide a couple of JSON examples of these objects with comments when necessary. Think about a document that you will give to another database engineer that would take over your database.

We'll have 6 collections -- Car, Customer, Booking, RentalBranch, CarModel, CarMake. We choose to use references to store the relationship between data, like the example JSONs below.

JSON examples

<u>Car collection:</u>

```
{
        "_id": ObjectId("fdasfdsr34rqfdtewf"),
        "currentRentalBranch": ObjectId("123123"),
        "model":  ObjectId("234234"),
        "make": ObjectId("456456"),
```

“startYear”: 2020,
“mileage”: 19898,
“isAvailable”: True
}

Customer collection:
{
“_id”: ObjectId(“fdasfdsafdsaf”),
“firstName”: “April”,
“lastName”: “Zhang”,
“phoneNumber”: “8883507088”,
“email”: “email@my.com”,
“city”: “Sunnyvale”,
“state”: “California”,
“country”: “USA”
}

Booking collection:
{
“_id”: ObjectId(“fdsrewrgfds”),
“bookingStartDate”: ,
“bookingEndDate”: ,
“car”: ObjectId(“432432rdwar”),
“customer”: ObjectId(“54f345”),
“totalCharge”: 56.78,
“pickupRentalBranch”: ObjectId(“fda324fdsafdsa”),
“returnRentalBranch”: ObjectId(“432432fdsafdsfds”)
}

RentalBranch collection:
{
_id : ObjectId(“123123”),
branchName: “5th Avenue”
Address: “5th Avenue”
City: “San Jose”
State: “California”
country : “United States”
branchManager: “Tim Cook”
}

CarMake collection:
{
“_id” : ObjectId(“456456”),

“make”: ”Honda”,
}

CarModel collection:
{
“_id”: ObjectId(“543543”),
“model”: “Elantra”
}