COIT12200 Assignment 3 (Project) hints, tips and clarifications

Assignment 3 involves multiple table database access and software design pattern. When a database contains multiple tables, the application program may need a more complicated logic to deal with the relevant database access request – retrieval data and write data into the database persistently. These tables normally have foreign keys linked together. Manipulating data reading and writing are performed by a java program with the embedded SQL statements. Here I list some hints, tips and clarifications for your information.

1. The database should contain three tables – *Customer, Vehicle and Service* to store related data, where the primary key -CustomerID in the Customer table and the primary key -ServiceID in the Service table should be *integer, auto_increament*, whereas the primary key - Registration number of Vehicle table should be a short text. There should be foreign keys in some tables.(alternatively Vehicle table could have a separate ID).

2. The GUI is used to represent data input and display of manipulation results. You are allowed to have a flexible layout of GUI components that make sense for the data input and display in this application. (No requirement for login functionality). If suitable, you also can refer to the general design principle from the unit of HCI.

3. The requirements of functionality specifications.

    a) To add a customer, enter personal data and hit the 'save' button; the customer data should be stored in the customer table in the database.
    b) Search customer by name or phone
    Search customer by phone, simply enter a phone number and click the button, if the customer exists, it should show customer detail and associated vehicle details (belonged to that customer). Search by name is performed by entering first name and last name then hit a separate search button, it may find zero or one customer(For simplicity, at most it is one customer). Then relevant customer and vehicle details are displayed properly. It may be common that a customer owns multiple cars.
    c) 'Update customer's address or phone' only updates the Customer table for the required change of address or phone.
    d) 'Add a vehicle to customer' that adds the data of a vehicle to a new customer or existing customer, referred to the *PurchaseApp* example, where a customer places orders.
    e) Display all service. This expects to show each service and associated vehicle and customer. It would be too long with a line to display every detail. You can show partial data of each table of service, vehicle and customer.
    f) Search a service booking by the vehicle registration number. This could produce a list with several vehicles, because in the system database, the same vehicle (registration number) may undergo multiple services.
    g) Cancel a particular booking. You need to enter the registration number and service date to delete that record. You can either use text field input or JOptionPane pop up message box to input them.

h) Statistical data. Using SQL statements to get and show the required data as specified. You need to populate sufficient data for your own tests and demonstrations.

4. Use the MVP design pattern, where the business logic processing is performed through the model, whereas the view is responsible for the representation of data. The presenter bridges the gap between them through the binding method. This realizes the separation of business logic and data representation. In MVC model, controller class is responsible for event-handling. You can either use MVC or MVP design pattern.

5. Tips
   a) The input of service date from the text field is a string (such as in the format yyyy-mm-dd, 2020-04-05). The java code dealing with SQL Date needs using java.sql.Date. you need to convert it properly.
   b) After inserted a new record into the Customer table, the latest key of the customerID can be obtained by using:ResultSet's `getGeneratedKeys()` method; This key value will used as the foreign key in the Vehicle table if added a vehicle. Alternatively, you can use the cursor to move in the ResultSet from the top to the bottom with a while loop.
   c) if necessary, you may need spending a little time to review SQL statements which you studied in the database course. Particularly be familiar with the table joint in SQL statements.

6. The project is a team-work project. It is expected every member contributes your knowledge, design, and programming skills. The software implementation doesn't have a unique solution but may exist an optimal or better solution. The provided example of PurchaseApp (on the moodle website) is a good example to deal 1:m relationship in programming. Good luck.