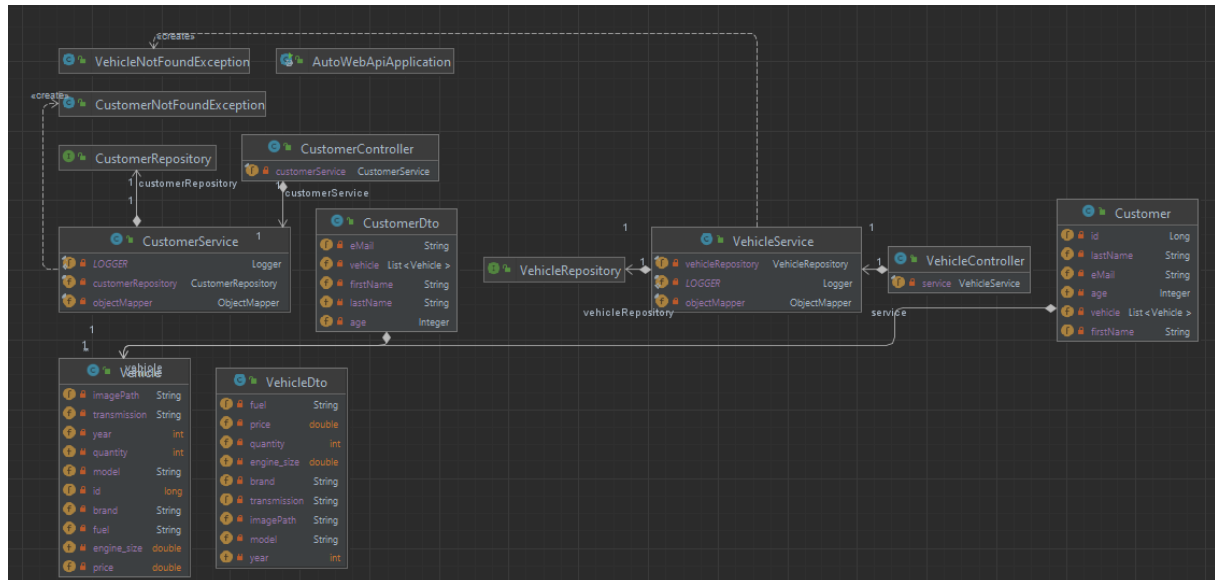


Auto_Dealer project

The diagram class picture:



Classes:

- **Customer**
- **Vehicle**
- **CustomerRepository**
- **VehicleRepository**
- **CustomerService**
- **VehicleService**
- **CustomerDto**
- **VehicleDto**
- **CustomerNotFoundException**
- **VehicleNotFoundException**

CustomerRepository \ VehicleRepository:

- Acts as a data access layer for customer information.
- Provides methods for CRUD (Create, Read, Update, Delete) operations on customer data.

CustomerService \ VechicleService:

- Business logic layer for managing customer data.
- Interact with the CustomerRepository \ VechicleRepository to perform operations like adding, retrieving, updating, or deleting customer information.

CustomerDto \ VehicleDto:

- A lightweight object used for data transfer between layers.

Customer \ Vehicle class:

- Represents a real-world customer\vehicle with attributes (Specific attributes depend on the application).
- Has methods for getting and setting customer \ vehicle information.

CustomerService has a one-to-one relationship with CustomerRepository. This means a CustomerService instance has a reference to a single CustomerRepository instance.

CustomerService has a one-to-one relationship with VehicleService. Similar to the previous relationship, a CustomerService instance has a reference to a single VehicleService instance.

CustomerController has a one-to-many relationship with CustomerService. A CustomerController interact with multiple CustomerService objects.

CustomerController has a one-to-many relationship with VehicleController. Similar to the previous relationship, a CustomerController interact with multiple VehicleController objects.

CustomerRepository has a one-to-many relationship with Customer. This suggests a CustomerRepository instance manage or store information for multiple Customer objects.

VehicleRepository has a one-to-many relationship with Vehicle. Similar to the previous relationship, a VehicleRepository instance manage or store information for multiple Vehicle objects.

CustomerDto might be used for data transfer between layers of the application. It has attributes corresponding to the data members of the Customer class.