Vehicle Rental Management System:

This project aims to provide customers the ability to rent vehicles. Administrators will also be able to add vehicles, remove vehicles and check the history of rented vehicles.

Functionality:
Customer:

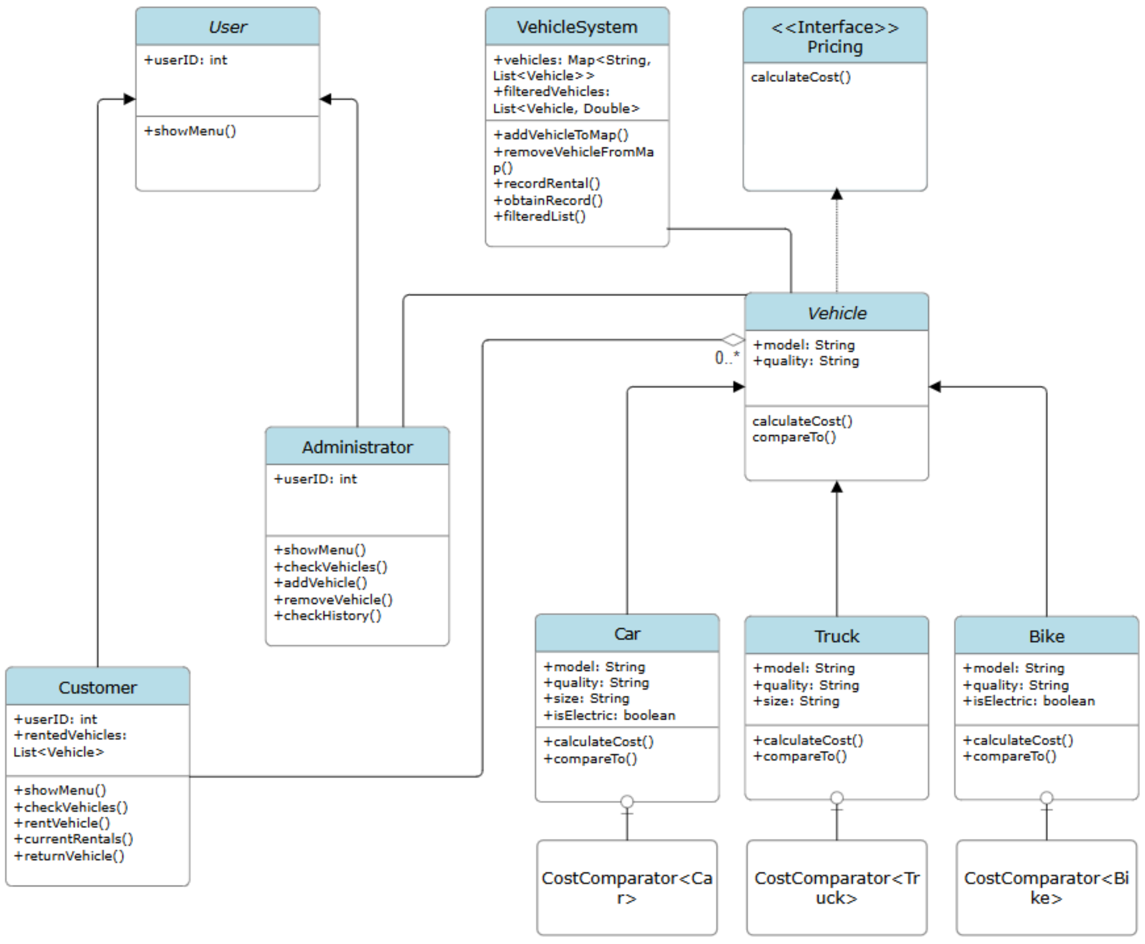
- Check vehicles, being able to filter through multiple fields (depending on the type of vehicle they are looking for)
- Rent vehicles, being able to filter through multiple fields
- Check their current rented vehicles
- Return vehicles

Administrator:

- Check vehicles, like a customer would
- Add new vehicles to get rented
- Remove vehicles from the rental
- Check the history of vehicles that have been rented

Class Hierarchy: There is an abstract User superclass with two subclasses Customer and Administrator. There is also an abstract Administrator superclass with three subclasses Car, Truck and Bike.
Interface: A pricing interface is implemented into the Vehicle class (therefore also its subclasses) to define unique ways to price different types of vehicles based on their fields.
Methods with runtime polymorphism: showMenu(), calculateCost() and compareTo().
Use of textIO: TextIO will be used to store the history of rentals.
Comparable & Comparator: The Vehicle (and its subclasses) need to use the comparator in order to show vehicles sorted in alphabetical order when shown to users. The Car, Truck and Bike classes also need a Comparator to sort vehicles by cost if a customer filters vehicles by price (only one type of vehicle can be searched for at a time).

Class Diagram:



For deliverable 2, the Car, Truck and Bike classes will be fully implemented including their Comparable classes and the Pricing interface. The methods addVehicleToMap() and filteredList() from the class VehicleSystem will also be implemented.