Submitted by :

Omer Gery – 316124437

Daniel Dolev – 316601996

Classes , Enums :

**UI Side:**

**UI Class** : holds the menu methods and it is responsible for the program flow. It has a reference to the logic side and it is responsible for making contact with the logic side.

**Enums:**

**eMainMenuOptions:**This enum holds the possible user’s selections in the main menu.

**eVehicleDisplayOptions**: This enum holds the possible user’s selections whether to see all the vehicles in the garage , or only the vehicles which belongs to a certain fix-state.

**Logic Side:**

**Garage**: Holds the data for the logic side, has a dictionary of garage-vehicle. Also has methods to make changes and add vehicles to the garage.

**Vehicle**: This is an abstract class that holds the basic methods and data that are common to all the vehicles.

**Electric Vehicle**: This is an abstract class that holds the basic methods and data for all the electric vehicles. Inherits from vehicle.

**Fuel Vehicle**: This is an abstract class that holds the basic methods and data for all the fuel

vehicles. Inherits from vehicle.

**Car**: This class holds the data for cars , this class is being used by both Electric Car and Fuel Car.

**Bike**: This class holds the data for bikes , this class is being used by both Electric Bike and Fuel Bike.

**Truck**: This class holds the data for Fuel Truck. Possible to hold data in the future for Electric Truck.

**Garage** **Vehicle**: This class represents a Vehicle in his garage-storing form , means it holds the Vehicle, his state in the garage , and the owner details.

**Electric Bike/Fuel Bike /Fuel Truck/Electric Car/Fuel Car** : These classes are the actual objects used in the logic side , they inherit from Electric Vehicle/Fuel Vehicle and composes Car/Truck/Bike.

**Tire**: This class represents a vehicle tire data and methods.

**Value out of range exception**: We implanted this class to be responsible for creating logic – out of range exceptions.

**Vehicle Owner**: This class holds the data for a vehicle owner in the grage : phone and name.

**Vehicle Param**: This class is used to represent a generic vehicle data member, this class was made in order to implement run-time making of vehicles.

**Vehicle Builder**: This class is responsible for creating new vehicles, by using a static method it gets the user input for the parameters and the users selection of vehicle and return a new vehicle.

**Enums:**

**GarageEnums:** this class aggregates the enums of the logic side – except for the vehicle types list,

It has the following enums:

eFixState: This enum holds the vehicle possible-fix states in the garage.

eFuelType:This enum holds the possible fuel vehicle-fuel types.

eNumberOfDoors:This enum holds the possible car number of doors.

eColor:This enum holds the possible colors for a car.

eBikeLicenceType:This enum holds the possible bike licence types.

The enum eVehicleType is in **Vehicle Builder:** it holds the possible vehicle-types in the garage.

Diagram:

