Advanced Programming Workshop #2

Sergio Nicolás Mendivelso

V 2.0

This program create various types of vehicles and add them to a list. It have classes for Engine, Vehicle ,Car, Truck, Yacht, and Motorcycle. The program will not manage MVC model anymore, it will be like a monolith system, also it will allow register an login.

Bussines Model:

In this type of platforms, the users can see a catalog of the vehicles. So, exists the designers who creates the vehicles an upload it to the platform. This can be usefull to sell their products.

Business rules:

Login an register: The user has to register to access the information to keep the security

Validation: The programm has a way to verify the data when somebody is registered or a vehicle is created (The year of creation has to be a number.. Etc).

Vehicles: A designer is the one how can create a vehicle.

User Stories:

As User (Designer)I want to create an account to log in and see the created vehicles.

As Designer I want to create a various types of vehicles to have a list of them and upload it

As Designer I want to watch the information of my created vehicles list

As User I want to watch another created vehicles in a catalog

Entities:

User

Designer

Vehicles

Catalog

Account

CRC cards:

Vehicle	
Responsability: Provide information about the vehicle	Collaborators: Engine Designer Catalog

Engine	
Responsability: Provide information about the vehicle consuption and potency is added to a vehicle calculate gas consumption of the vehicle	Collaborators: Vehicle Designer Catalog

User	
Responsability: Register and login watch vehicles catalog	Collaborators: Catalog Database

Designer	
Responsability: Create Vehicles and engines Upload their new vehicles to the catalog	Collaborators: Vehicle Engine Catalog

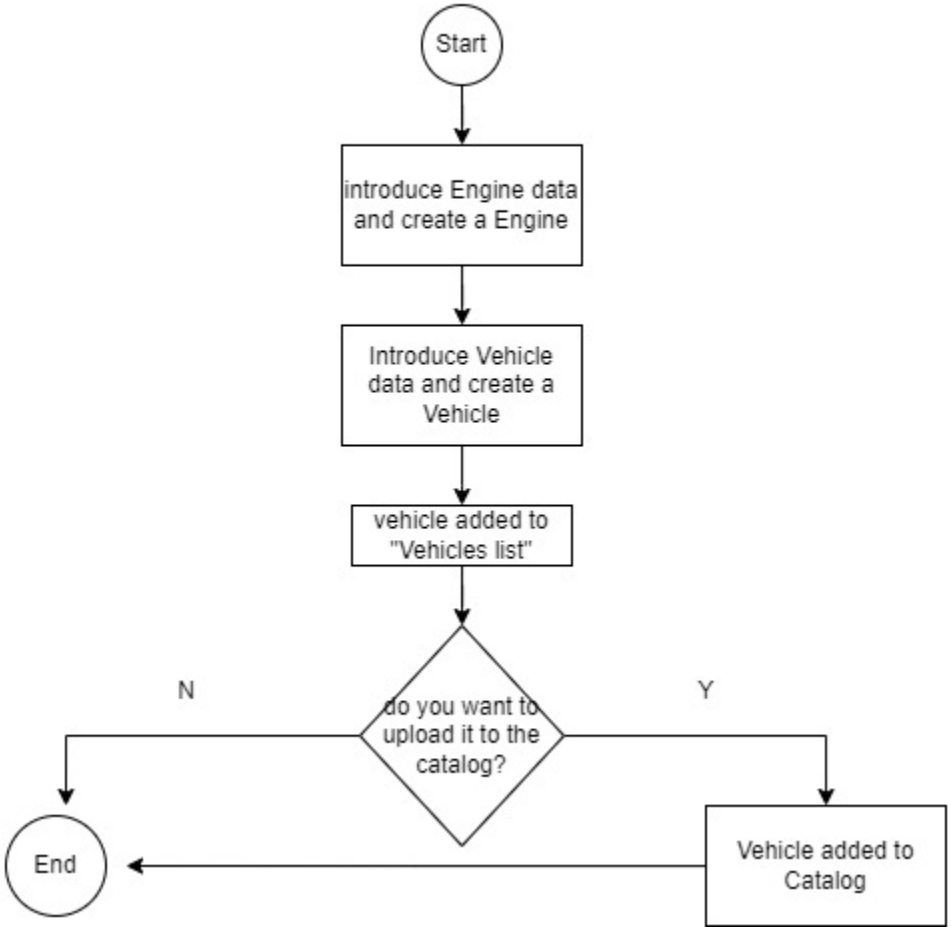
Database	
Responsability: manage the user registration and login in order to add security Save the users information	Collaborators: User

Catalog	
Responsability: Show created vehicles list Update craeted vehicles list adding new vehicles Show the menus	Collaborators: User Designer Vehicles Database

Note: These are the enough CRC Cards, because the yacht, car, motorcycle and truck classes are just concrete versions of vehicle.

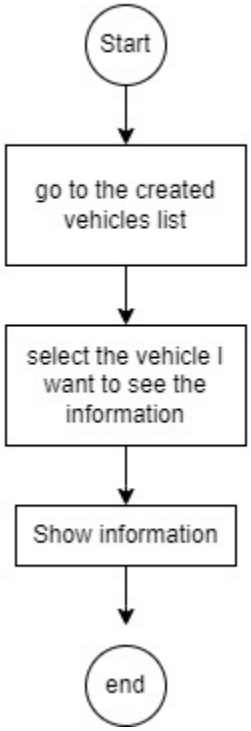
Activity Diagrams:

As Designer I want to create a various types of vehicles to have a list of them and upload it

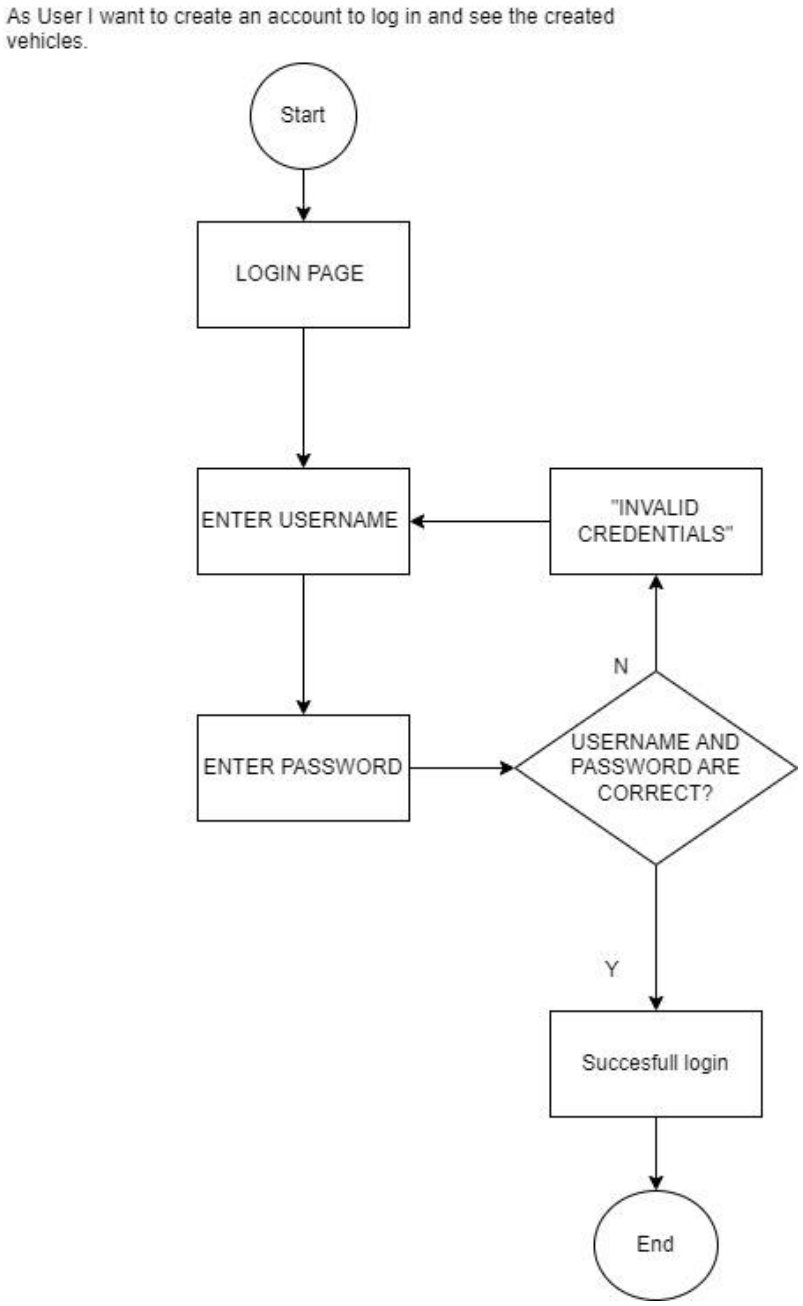


This diagram shows how to create a vehicle being a Designer, and how it can be upload to the catalog.

As Designer I want to watch the information of my created vehicles list



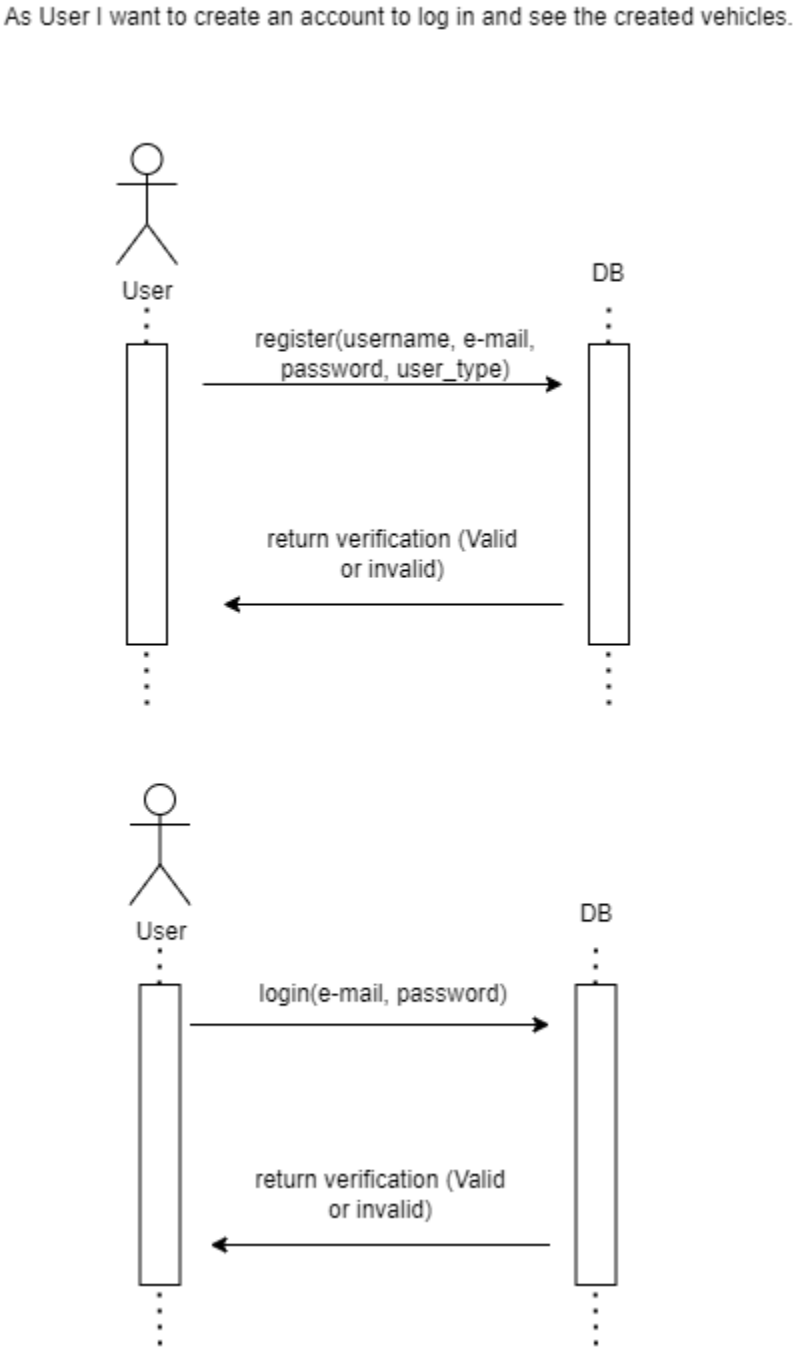
This diagram shows how to watch the created vehicles list for the Designers.



This diagram shows how log int to the system.

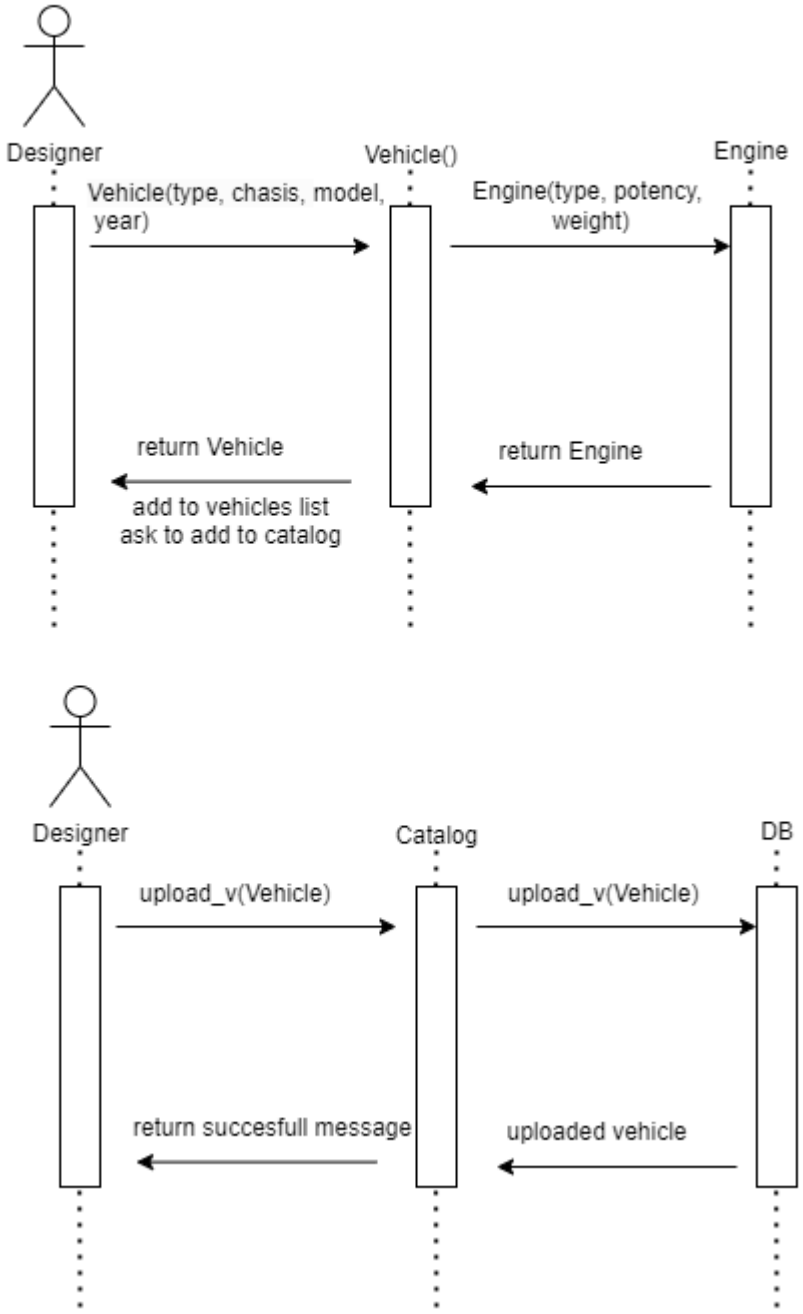
Note: The user Story “As User I want to watch another created vehicles in a catalog” Has not an activity diagramm, because it is a obvious and short process.

Sequence Diagrams:

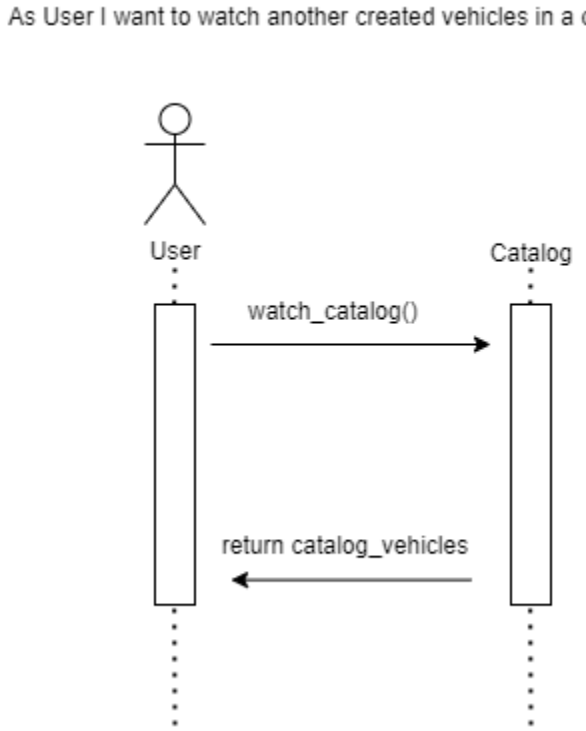


These diagrams shows how an user object and a DataBase object interact to obtain the verification of the register or the login.

As Designer I want to create a various types of vehicles to have a list of them and upload it



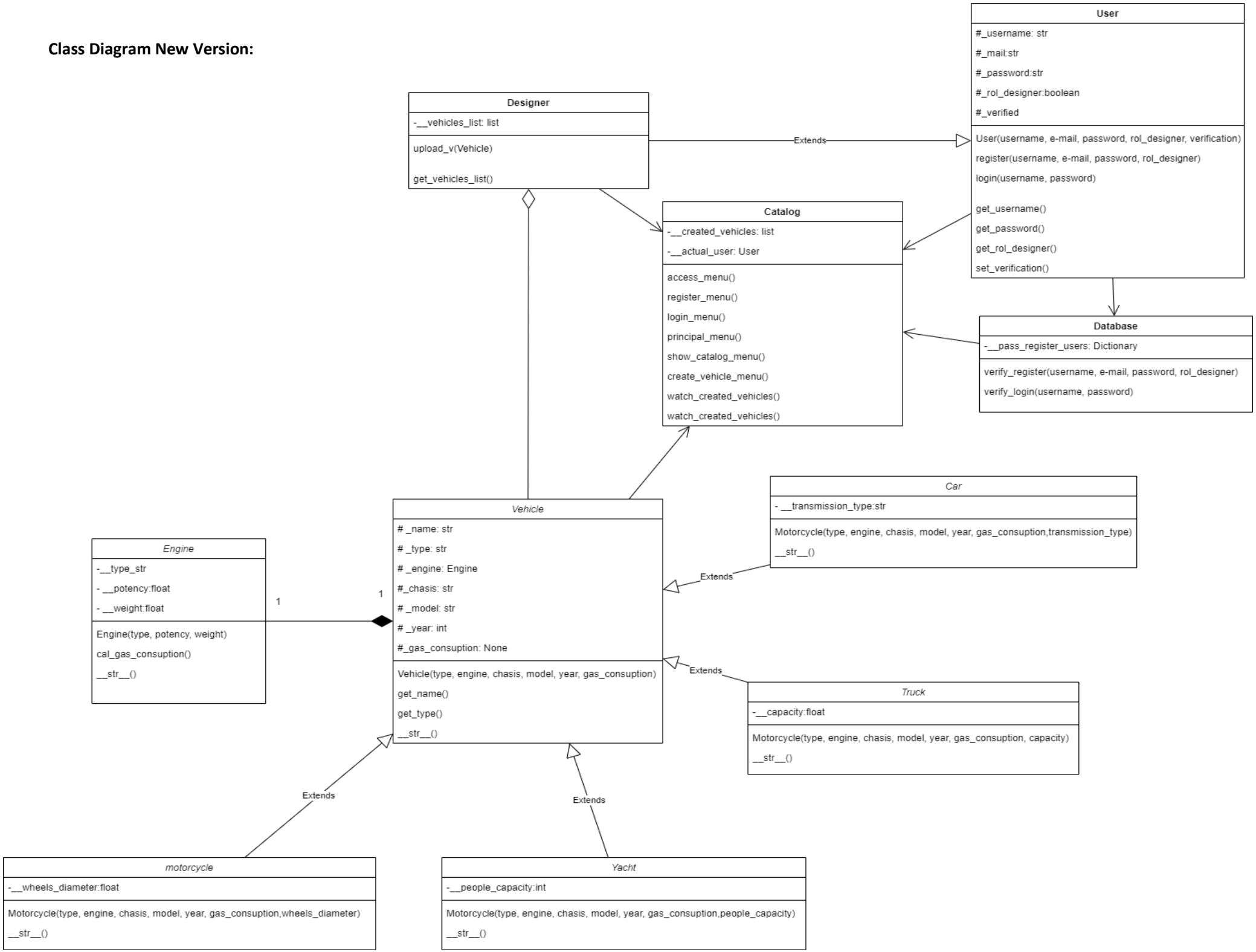
These diagrams shows how a designer object create a vehicle, so it has to create an engine too, also shows how it can upload a vehicle to the catalog and to the DataBase object.



This diagram shows how a User can watch the catalog and it information.

Note: The User story “As Designer I want to watch the information of my created vehicles list” has not an object interaction. So it has not a sequence diagram.

Class Diagram New Version:



New changes:

1. No more MVC model. Eliminated clasess: Main_controller, View, Launcher
2. New Classes: User, Designer, DB and Catalog
3. New module separation: vehicles and users.
4. New Relations, agreggation, association, composition.
5. Now vehicle has a name attribute
6. No more get-Setter methods, just one that method that return all the atrributes of the vehicles, and the engine (__str__() method). The method to get the vehicle type keeps.
7. cal_gas_consuption() method will pass from Main class to Engine class

General Description:

- The first change I did, were the vehicles classes. I create an __str__() method in all the vehicle classes and also add the name attribute in Vehicle class
- I use a composition relation in the Engine class to vehicle class.
- I change the cal_gas_consuption() method from the controller to the Engine class, I think to add it into the Vehicle class, but the method has more engine attributes than vehicle attributes, so the method is shorter an readable in the Engine class.
- After that, I create the Users module.
- I created the “users” file composed by User and Designer class. I created all its methods and attributes in both classes
- Login have the mail and password parameters.