

## **Advanced Programming Workshop #3**

Sergio Nicolás Mendivelso

V 3.0

I feel good with the backend process, the user stories and the UML diagrams, so, it won't change anymore, in this new version of the program, I will create a graphic interface for the program.

### **Business Model:**

In this type of platforms, the users can see a catalog of the vehicles. So, there exist designers who create the vehicles and upload it to the platform. This can be useful to sell their products.

### **Business rules:**

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

Validation: The program 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 who 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 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 consupcion 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

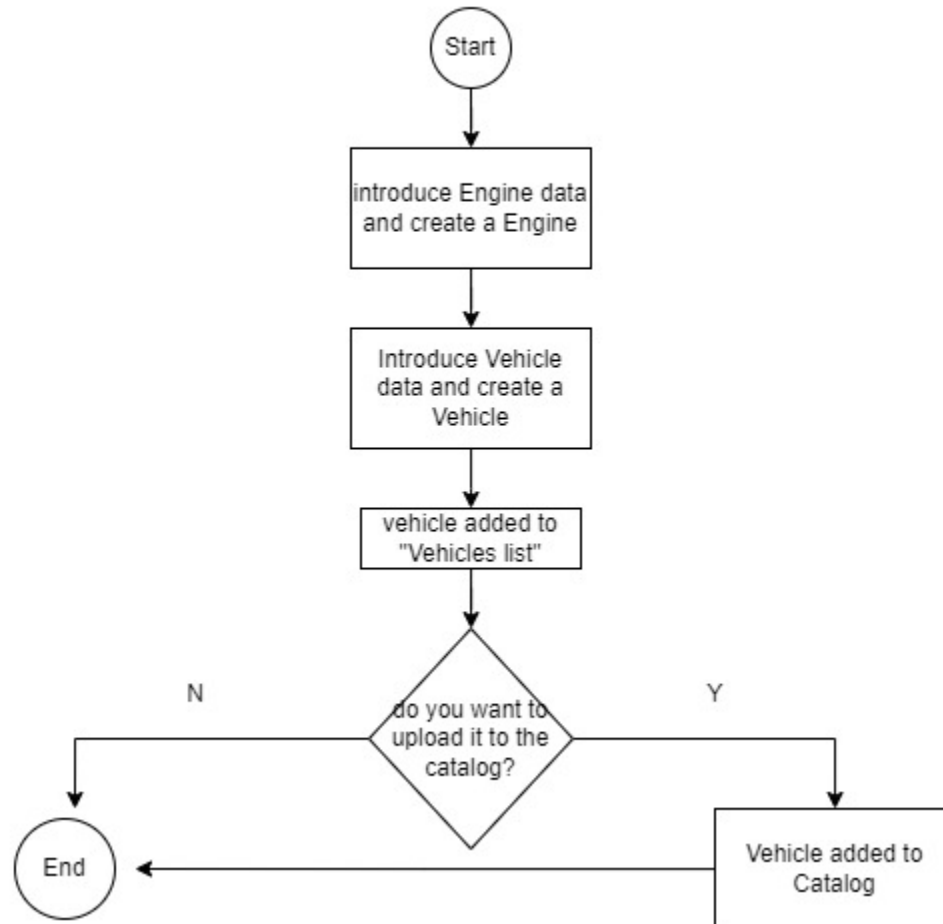
  

Catalog	
Responsability: Show created vehicles list Update created vehicles list adding new vehicles Show the menus	Collaborators: User Designer Vehicles Database

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

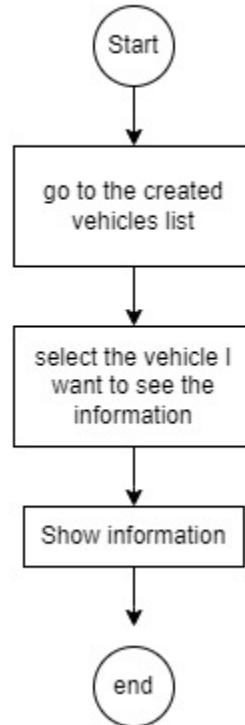
**Activity Diagrams:**

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



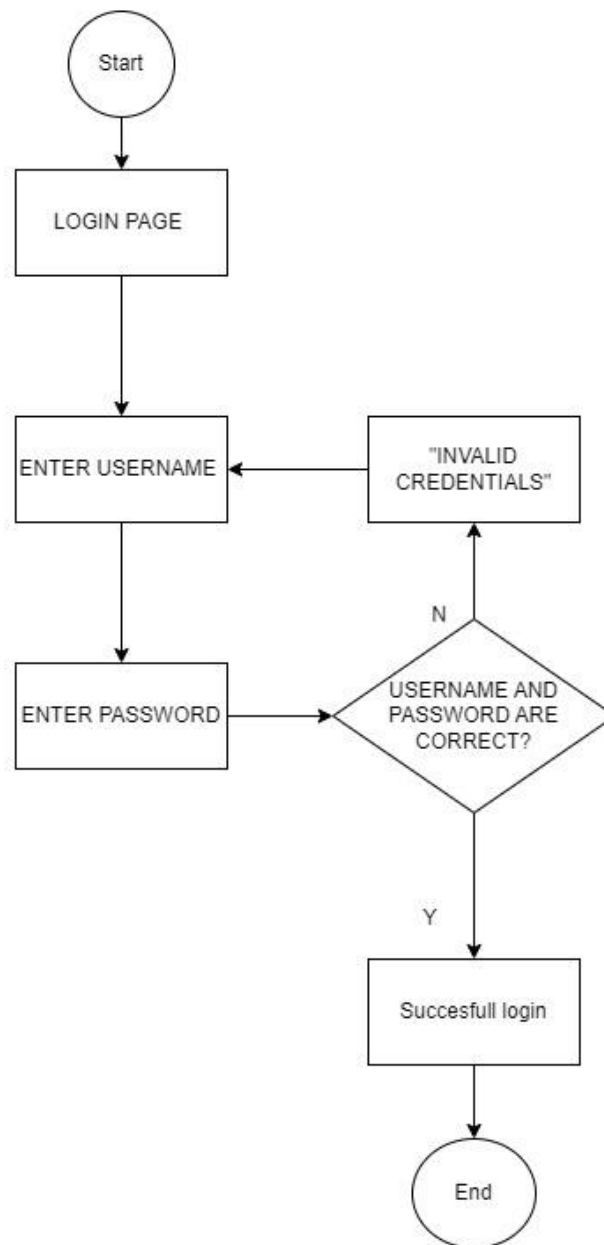
This diagram shows how to create a vehicle being a Designer, and how it can be upload to the catalog.

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.

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

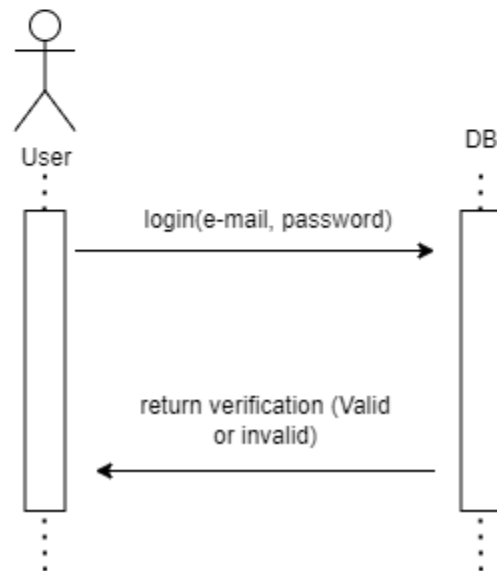
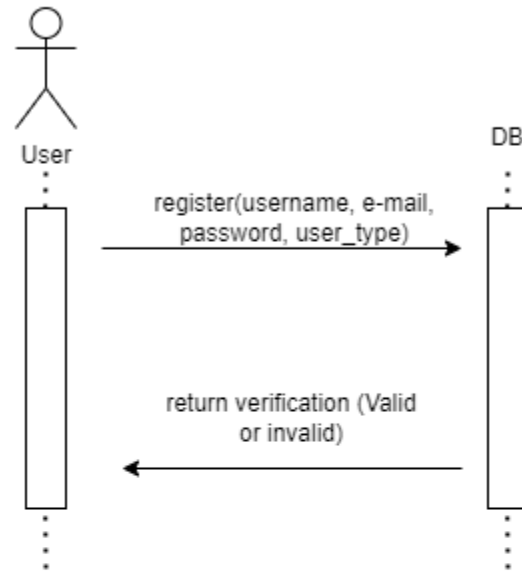


This diagram shows how log int to the system.

**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.

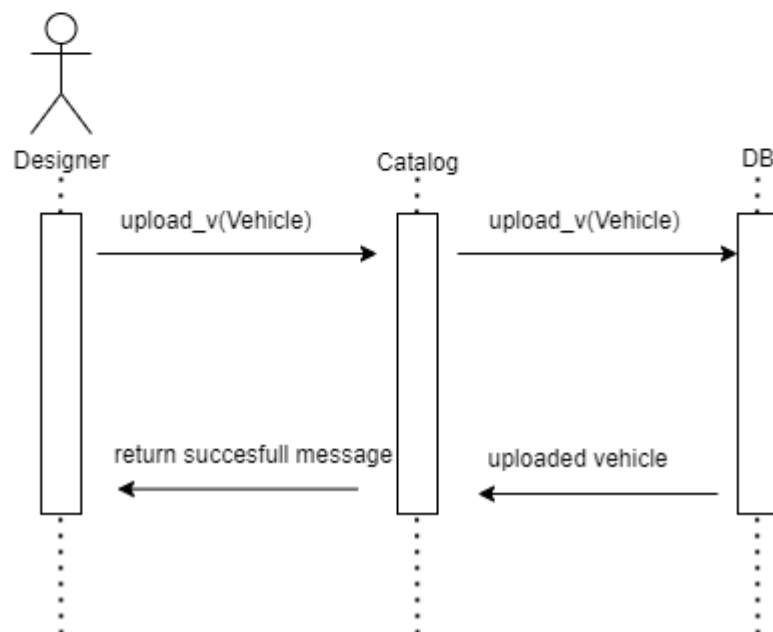
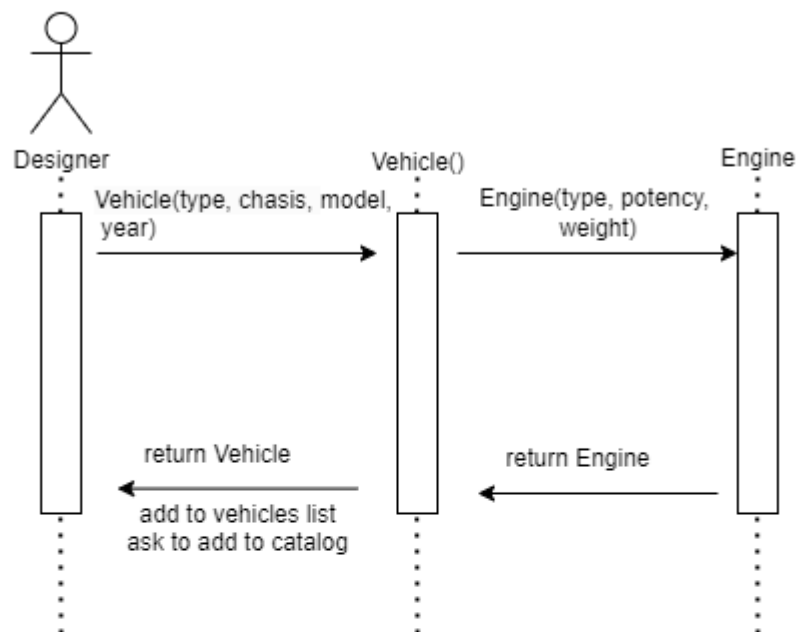
## Sequence Diagrams:

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



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

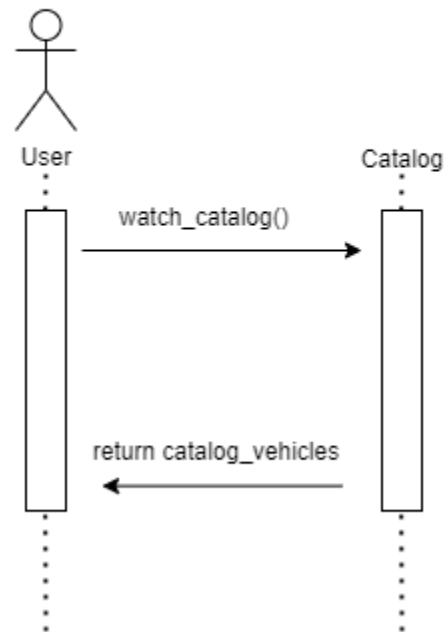
As Designer I want to create a various types of vehicles to have a list of them and upload it



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.



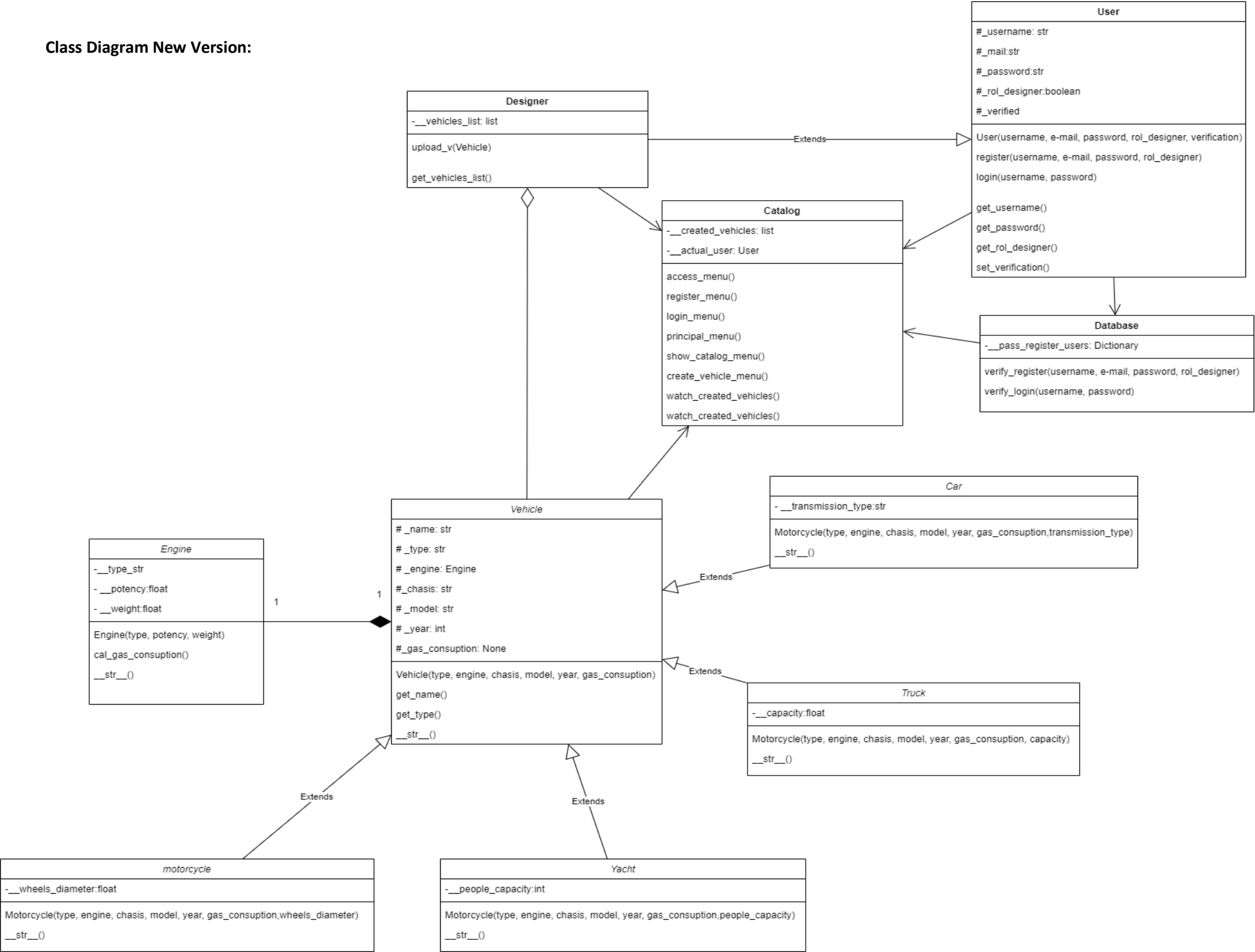
As User I want to watch another created vehicles in a catalog



This diagram shows how a User can watch the catalog and its 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 classes: Main\_controller, View, Launcher
2. New Classes: User, Designer, DB and Catalog
3. New module separation: vehicles and users.
4. New Relations, aggregation, association, composition.
5. Now vehicle has a name attribute
6. No more get-Setter methods, just one that method that return all the attributes of the vehicles, and the engine (`__str__()` method). The method to get the vehicle type keeps.
7. `cal_gas_consumption()` 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_consumption()` 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 and 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.

## UI prototypes:

The next prototypes were created with the Figma Tool:

1. The user needs a form to register and login

# Welcome to vehicles catalog!!

What do you want to do today?

Register

Log in



---

# Register

Username:

Email:

Password:

Are you a designer?

☒ Yes☐ No

---

# Log in

Username:

Password:

2. The user needs a form to add a new engine to the application.
3. The user needs a form to add a new vehicle to the application.

# Welcome username

What do you want to do today?

See catalog

Create vehicle

Watch created vehicles list

## What type of vehicle do you want to create?

Car



truck



motorcycle



yacht



# Create your vehicle

first, let's create the engine of your car

Type of engine

Potency of engine (Kilowats)

Weight of your engine (Kilograms)

Now, lets create your vehicle

vehicle's name

Chassis (A or B)

Vehicle's model

Year of the model

Special attribute

Wanna upload to catalog?

☐☐

Create vehicle

4. The user needs a page to see all engines in the application.
5. The user needs a page to see all vehicles in the application.

# Vehicles catalog

Name	Type	Creator	Engine type	Potency	Engine weight	
FIG-121	Car	User	Engine type	potency	Engine weight	
FIG-122	Yatch	User	Engine type	potency	Engine weight	
FIG-123	Motorcycle	User	Engine type**	potency	...	Engine weight
FIG-124	Truck...	User	Engine type	potency		Engine weight
FIG-125	Car	User	Engine type	potency		Engine weight
FIG-126	Yatch	User	Engine type	potency		Engine weight
FIG-130	Motorcycle	User	Engine type	potency		Engine weight
FIG-131	Truck	User	Engine type	potency		Engine weight

Data structures JSON: