

TECHNICAL REPORT

User Stories

As a user, I want to register a account to that can access to the app.

As a user, I want to login to that use the specific functionalities according to my role.

As a user, I want to see all vehicles to that explore the available options.

As a user, I want to search vehicles for specific criteria to that find vehicles that fit my needs.

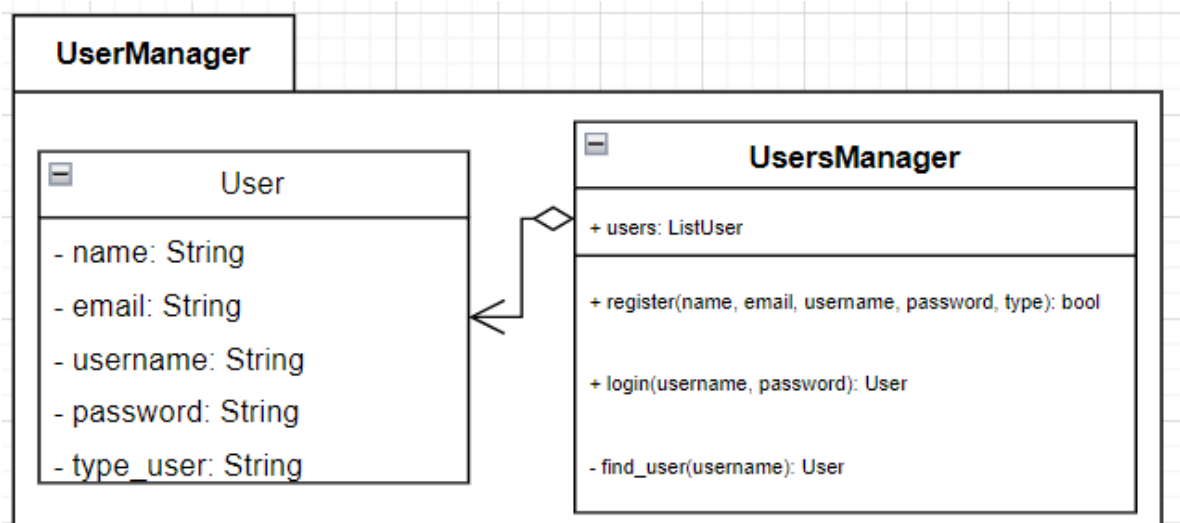
As a admin, I want to add new vehicles to that keep the catalog updated.

As a admin, i want to update the information of a vehicle to that fix errors or update details.

As a admin, i want to delete a vehicle to the catalog to that keep it updated.

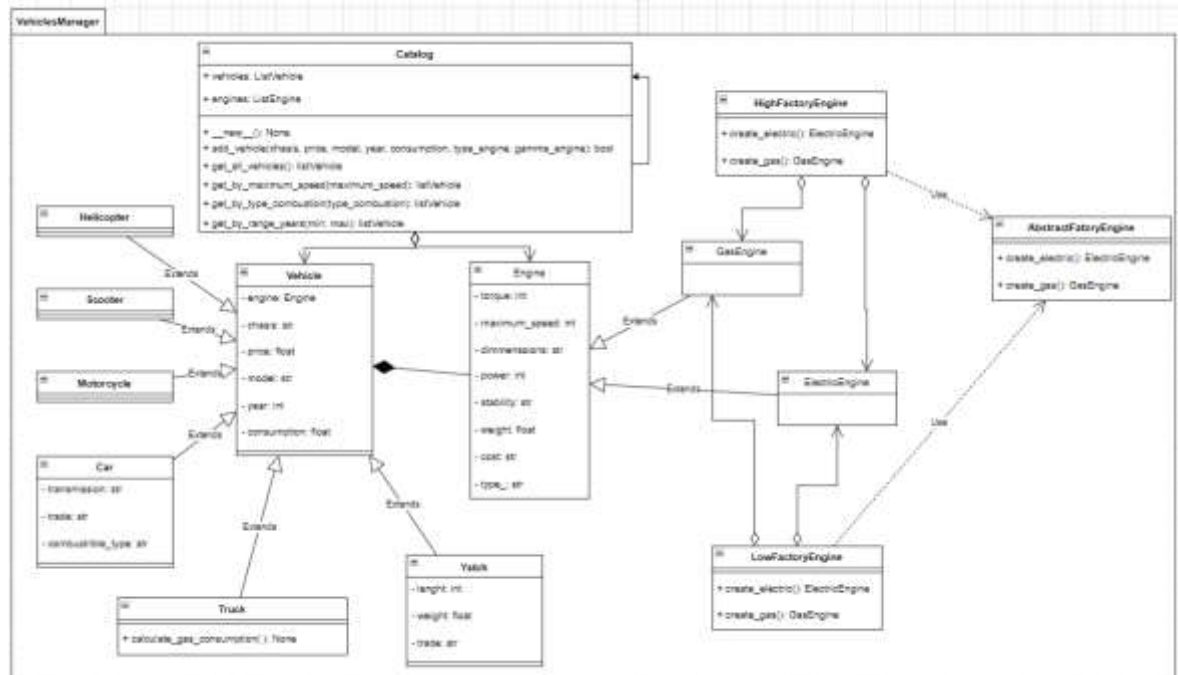
I divided the system into 5 components

1. UserManager



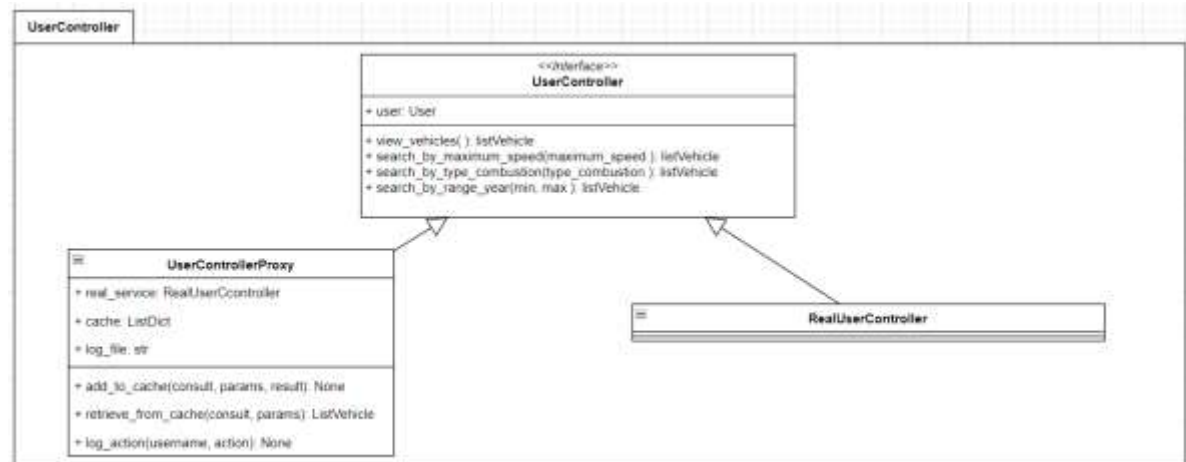
I create a user class that stores all the attributes that a user has, there is also the UsersManager class that is responsible for registration and login validations

2. VehiclesManager



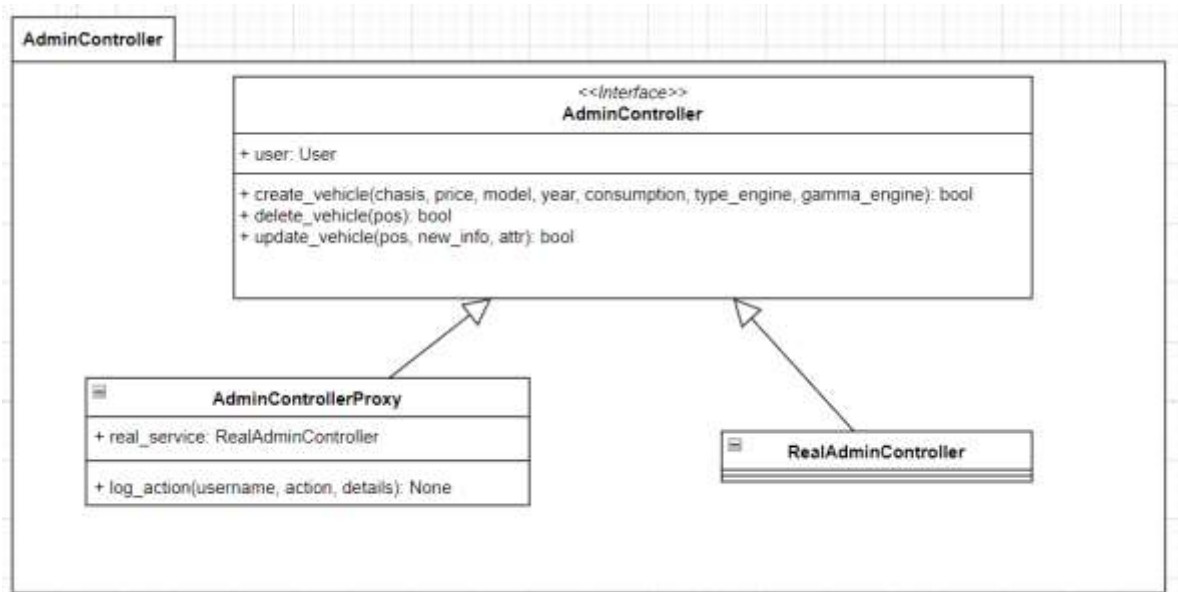
Take as a basis the model already made previously, in the catalog add new methods for searches, in addition to a list of engines with which you can create vehicles so as not to be creating repeated engines, in addition the relationship of the factories will be with the catalog and not with the vehicle.

3. UserController



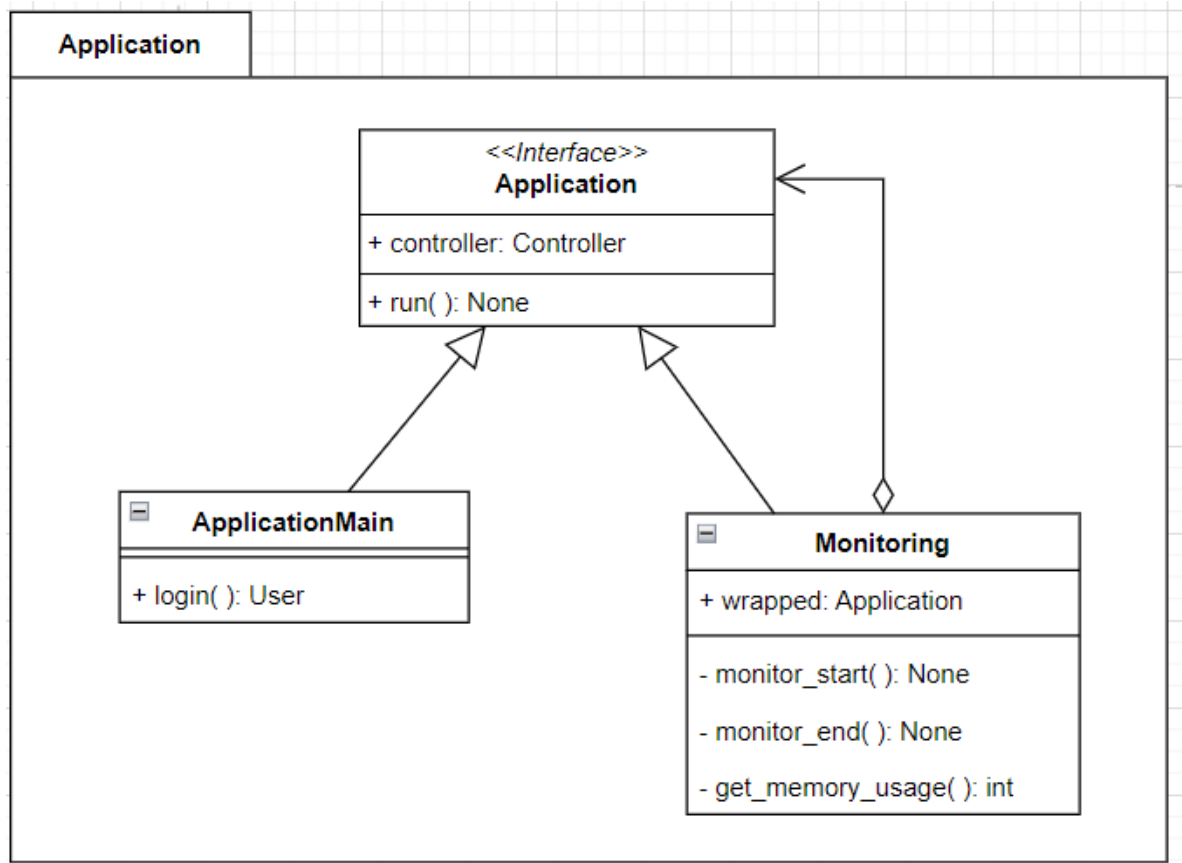
I decided to use the proxy pattern, the UserController interface contains the methods that represent the actions that a User can do in the application, the UserControllerProxy will be in charge of caching the last five queries to speed up the process when querying, it will also save all of them in a file the actions that the user does with the logging method

4. AdminController



We also use a proxy pattern, in this case as the admin does not make queries we do not keep a cache so the proxy will only save the actions performed by the admin in the log file

5. Application



The application interface has a run method where the application is executed and a

controller which is responsible for executing the methods depending on the type of user. The specific ApplicationMain class has the login method that will validate the user when starting the application. Using the decorator pattern, create a Monitoring class that extends Application, this will be in charge of measuring the memory consumption and execution time of the application