Design:

The design made is based on the design patterns creationals and structurals which help us to provide an effective solution reducing memory and make the system scalable for the future plans.

The design patterns creationals put on the code are singleton and abstract factory, the singleton is used for create the catalog for the vehicles that all the users, it doesn´t matter if is an client or an admin

In the other hand, the design patterns structural help us make relations between interfaces and classes more simple, for the project I develop the patterns flyweight, which help us reduce memory due to we instance the object just one time for all the vehicles, proxy help us to authenticate and make the cache system, decorator serve to put the monitoring and finally the facade divide the system into subsystems

Entities

- Users
- Catalog
- Vehicle
- Engines

CRC cards

Vehicles	
- provide vehicle information	Engine Catalog

Truck	
- provide truck information - calculate gas consumption	Vehicle

Catalog	
- show a list of vehicles - add new vehicle (just manager)	Vehicles Engines

- let user apply filters to find vehicles	User Proxy

Engines	
- provide engine information	Vehicles Catalog

FactoryLowEngines	
- create low price engines	Vehicles

FactoryHighEngines	
- create high price engines	Vehicles

User	
- show menu - handle user interaction options	Catalog Decorator

FactoryHighEngines	
- create high price engines	Vehicles

Proxy	
- Get data and authentic the login	User Catalog

Decorator	
- Monitor the process of the user	User

Conclusions

The diagram class is better for the goal of the requirements, trying to reduce memory and add new functionalities. But the problem is the development, I took a lot of time programming and that bring consequences such as not complete all the menu with requirements.

The structural patterns help us to think in a better way for know how to solve the problem and the goods things will have in the code and what we can to improve the code, making a code scability, low coupling

Code before coding is a terrible idea, I did it the last workshop and now I make the correct way and I see how I got all the ideas more clear, well , but this workshop is heavy and it affected me

Between more workshops like this, I will improve my logic coding, I still have errors that took me one hour to find the error and the error are like identation, cycles and more.