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)  - let user apply filters to find vehicles | Vehicles  Engines  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.