CarSells

This is a technical report for the software platform to show all vehicles available to be sell in the company.

As follows some user stories are defined:
- As manager, I want to add new engines, so what I can create different vehicles with those engines.
- As manager, I want to add new vehicles as cars, trucks, yacht and motorcycles, so what the clients could see all available products.
- As user, I want to watch all vehicles in the catalog, so what I could choose the vehicle better works for my needs.
- As a manager, I want to calculate vehicle consumption, so what the clients could understand vehicle performance.
- As a manager, I want the system validate user type, so what menu just shows specific functionalities.

Also, after talking with the client next business rules could be defined:
- User just could log in and see vehicles list. Managers, could see engines, vehicles, and add new both engines and vehicles.
- Every vehicle has a chassis, and only type A or B are valid.
- For every vehicle, chassis, model, gas consumption, and year, should be showed in the system.
- For each engine, name, type, potency, and weight should be persisted and showed.
- Gas consumption is based on next equation:
$$1.1 * engine.potency + 0.2 * engine.weight - (0.3 \text{ if } A \text{ or } 0.5 \text{ if } B)$$

## *CRC Cards*

| Engine | |
|---|---|
| Provide information to understand vehicle potency and consumption | Vehicle<br>Entry Catalog |

| Vehicle | |
|---|---|
| - Provide information about vehicles in order to give better insights to clients<br>- Auto calculate vehicle consumption based on engine and chassis information | Engine<br>Entry Catalog |

| Catalog Entry | |
|---|---|
| - validate user's role in order to define functionalities<br>- show a menu with different functionalities for all the type of users<br>- for each menu option, provide full functionality | Engine<br>Vehicle |

All vehicle types as car, motorcycle, truck and yacht, are under vehicles super class, so it is not necessary to create cards for those classes cos' they are just concrete versions.
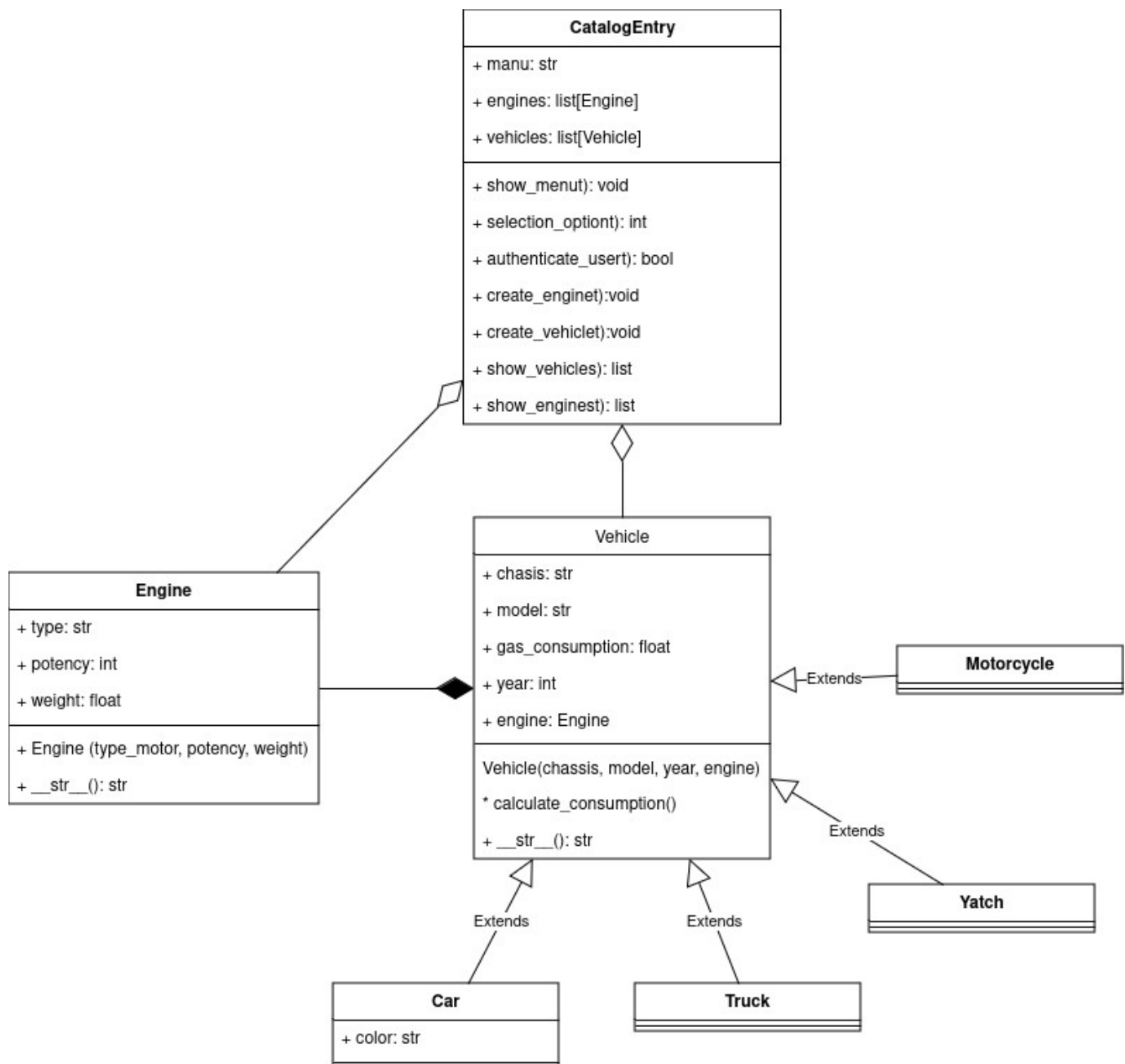
### *Activity Diagrams*

1. Validate user
2. Create engine
3. Create vehicle
4. Show menu and receive option

### *Sequence Diagrams*

*- Create Vehicle*

### *Class Diagram new Version*

## CatalogEntry

+ manu: str

+ engines: list[Engine]

+ vehicles: list[Vehicle]

+ show_menut): void

+ selection_optiont): int

+ authenticate_usert): bool

+ create_enginet):void

+ create_vehiclet):void

+ show_vehicles): list

+ show_enginest): list

## Engine

+ type: str

+ potency: int

+ weight: float

+ Engine (type_motor, potency, weight)

+ __str__(): str

## Vehicle

+ chasis: str

+ model: str

+ gas_consumption: float

+ year: int

+ engine: Engine

Vehicle(chassis, model, year, engine)

* calculate_consumption()

+ __str__(): str

## Motorcycle

Extends

## Yatch

Extends

## Car

+ color: str

Extends

## Truck

Extends

---

*State Diagrams*

**VehicleState**

**CatalogEntryState**

*Code Improvements*

A class called CatalogEntry is created t have a better management of menu and user's interactions. Also, each class will be separated in different modules, except by child classes of vehicles.
In addition, doctrings and type hint had been added to the code.