**Does your class diagram respect or violate SOLID principles? Justify your answer.**

It respects it.

**1- Single responsibility Principle:**

* We delegated the functions that operate the arrival and departure time for the new class (CalcTime).
* We delegated the function (park in) for the new class (ParkIn).
* We delegated the function (park out) for the new class (Parkout).
* We delegated the function (set and get the configuration) for the new class (Configuration).

**2- Open close Principle:**

We made an interface (ParkWay) to (BestFit) and (FirstFit) and delegated function (park way) to a new class (Configuration) so that when we modify the system and add a new way to park in the only modification is to add the class itself which implement from (ParkWay) and add a new condition to class (Configuration).

**3- Interface segregation principle:**

Code to an interface, we made any function that deals with the two ways of parking in to deal with the interface (ParkWay).

**Does your class diagram contain any design pattern(s), if yes name it and list the names of the classes involved in such pattern(s)**

It does.

**Strategy Pattern:**

We made an interface (ParkWay) to (BestFit) and (FirstFit) and delegated function (parkin) to a new class (Configuration) so that when we modify the system and add a new way to park in the only modification is to add the class itself which implement from (ParkWay) and add a new condition to class (Configuration).