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

Yes, it does.

**1- Single responsibility Principle:**

We delegated the functions that operates the arrival and departure time for the class (Vehicle) to a new class (Time).

**2- Open close Principle:**

We made an interface (SearchType) to (BestFit) and (FirstSlot) for the two configurations, to make it up for modification.

**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 (SearchType)

**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)**

Yes, it does.

**Strategy Pattern:**

We made an interface (SearchType) to (BestFit) and (FirstSlot) and delegated the configuration check from (ParkingSystem) 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 (SearchType), and add a new condition to class (Configuration).