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

It respects it.

**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 (ParkIn) to (Best Fit) and (First Come First Served) and delegated function (parkin) from (ParkingSystem) to a new class (CurrentConfiguration) 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 (ParkIn), and add a new condition to class (CurrentConfiguration).

**3- Interface segregation principle:**

We have separated vehicle specification from vehicle because its functions and variables are responded of dimensions which are used in different places from the functions in class vehicle. Same thing for (ParkingSlot) and (ParkingSlotSpecifications).

**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 (ParkIn) to (Best Fit) and (First Come First Served) and delegated function (parkin) from (ParkingSystem) to a new class (CurrentConfiguration) 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 (ParkIn), and add a new condition to class (CurrentConfiguration).