# 

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20201183 | Mayar Mohamed Hamed | mayarmohamedhamed12345@gmail.com | 01024079836 |
| 20200595 | Nadeen Badr Eldin | Nadeenbadr0@gmail.com | 01122062048 |
| 20200095 | Amira Ahmed Ibrahim | am27335102@gmail.com | 01114989003 |
| 20201218 | Walaa Soudy Ibrahim | Walaasoudy36@gmail.com | 01150324662 |

**Answer the following questions:**

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

S (single responsibility principle):

Achieved in all classes (owner, Slots, vehicle, Display, calculate income, calculate fees) since they have only 1 main function that they do.

O (open / closed principle):

Not achieved

L (liskov substitution Principle):

Achieved since we did the park in function in a different class than the park out

They both use vehicles and slots but their functions are different so we created 2 different classes for each function

I (interface segregation principle):

Achieved

Class (website) is the interface and classes (park out, park in) implement from it

And the customer view only the stuff that he needs about his car (park in, park out, fees)

And the owner only view the stuff that’s related to manage the garage (income, number of slots, garage techniques)

D (dependency inversion principle):

Achieved since that high-level modules not depended on low-level modules; both are depend on abstractions and that’s achieved in (park in, first park, Best Park

**-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 there is design pattern in my class :-

1 – strategy in interface (website) , class(bestplace) and class(Firstpark).