1. It respects the SOLID principles as:

* Every class do one thing and only one thing (Single Responsibility)
* The system is open to add new features and closed to refactor the old classes as if you want to add new functionality, you can add a new class with functionality or even a new park in method, or add new parking slots
* We separated all interfaces as every class that inherits from any interface uses all of it’s methods and attributes
* There is no class is tightly closed to any other class as if we edited in any class, then we don’t have to edit in other classes.

1. The classes follow design patters concepts as:

* **Singleton:** as the code creates a single object from every class.
* **Strategy:** as the classes of functionalities inherits from abstractions, which group them under the umbrella of a system functionality.
* **Abstractor Factory**: abstracting class from larger classes to make the bigger classes simpler and the abstracted classes easy to change.

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200631 | Yasser Mohamed Emam Mohamed | YasserEmam2002@gmail.com | 01091720988 |
| 20200046 | Ahmed Mohamed Majeed El-Zeedy | Zeidy44zeidy44@gmail.com | 01022150693 |
| 20201002 | Ahmed Ayman Sayed Farag | Ahmedayman778770@gmail.com | 01141763762 |
| 20200394 | Kareem Mohamed Kadrey |  | 01148406302 |