**Ide Aplikasi**

Aplikasi Rental HP dengan fitur register login, pada saat register user memasukan nama, nomor hp, no ktp, dan juga alamat tujuannya agar aman lalu user dapat login dan dapat memilih hp yang akan disewa dan berapa lama selanjutnya ada fitur untuk melihat transaksi lampau dan juga export transaksi transaksi tersebut

**Penjelasan Design Pattern**

* **Singleton**

Tujuan penggunaan Singleton yaitu untuk memastikan bahwa hanya ada 1 Instance dari suatu class yang ada. Biasanya digunakan karena terkait dengan resource seperti suatu database atau file. Singleton juga memberikan kemampuan untuk mengakses Object dari bagian manapun dalam program

* **State**

Tujuan menggunakan state pattern yaitu untuk menentukan behavior atau bagaimana program bekerja pada suatu state tertentu. Dalam state pattern kita disarankan untuk membuat satu class baru untuk setiap state yang ada dan meletakan bagaimana suatu behavior dalam state tersebut.

* **Decorator**

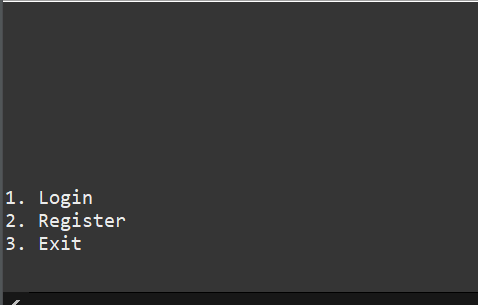
Decorator merupakan salah satu structural design pattern yang membuat kita dapat menambahkan behavior baru pada sebuah object dengan menempatkan object tersebut dalam sebuah wrapper atau pembungkus.

* **Prototype**

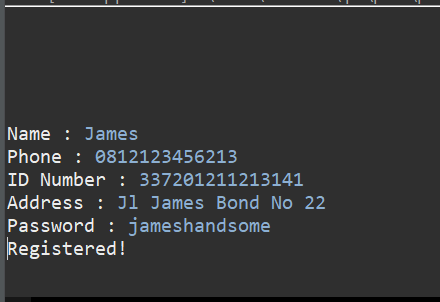
Prototype adalah salah satu creational design pattern yang memungkinkan kita dalam mengcopy object yang sudah ada tanpa membuat code kita memiliki ketergantungan dengan class tersebut.

**Screenshots**

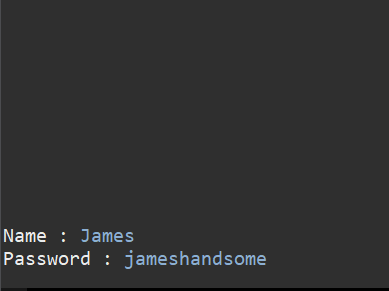
* Main Menu (Before Login)



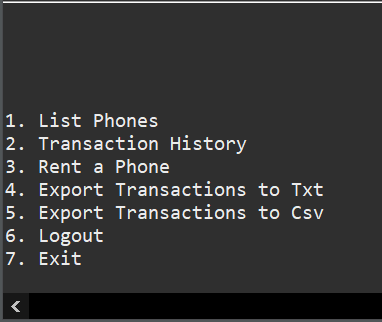
* Register



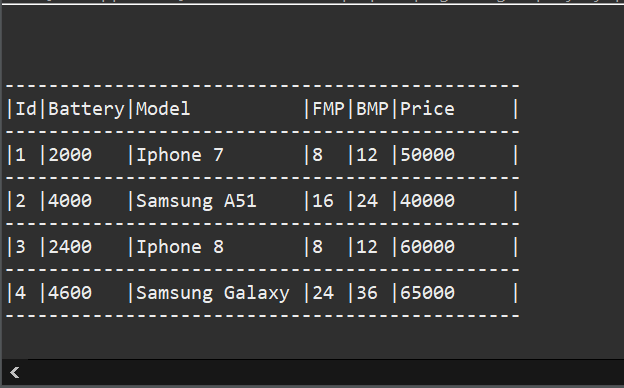
* Login



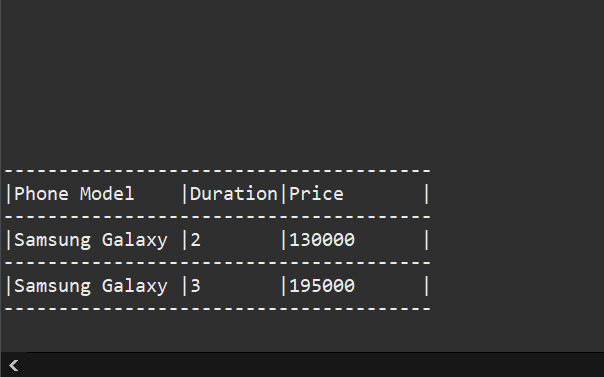
* Main Menu (After Login)



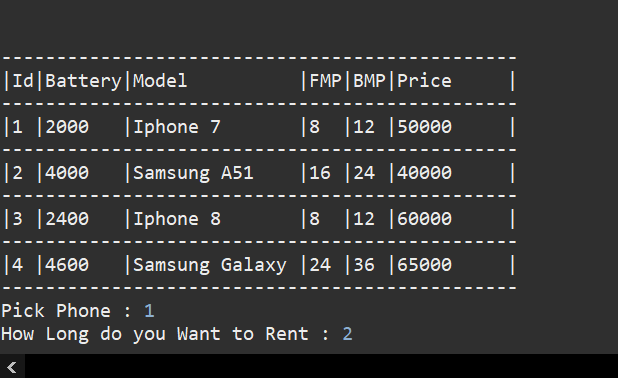
* List Phones



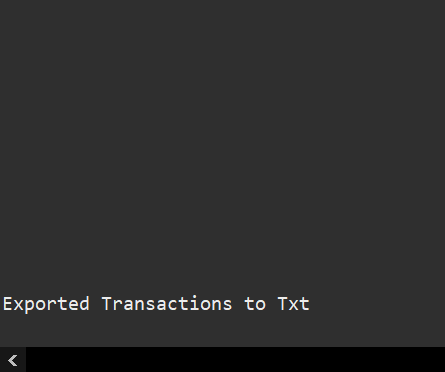
* Transaction History

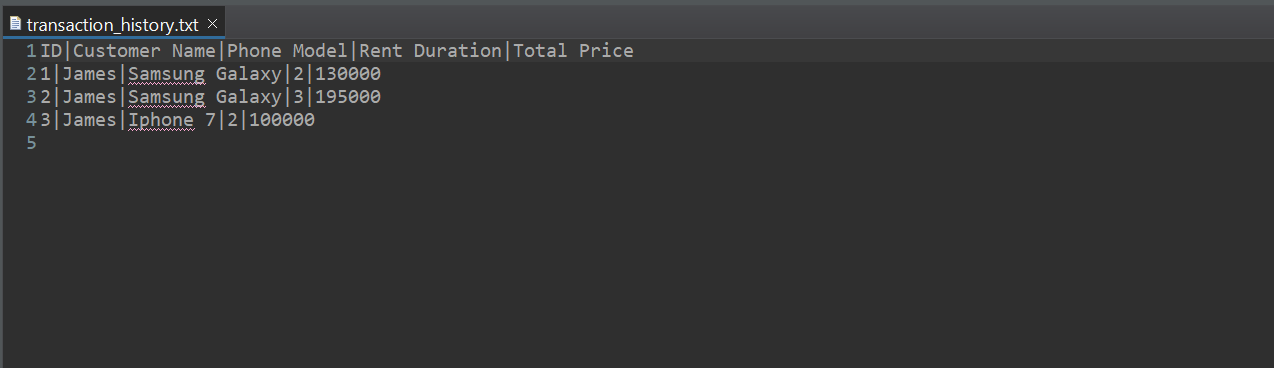


* Rent a Phone

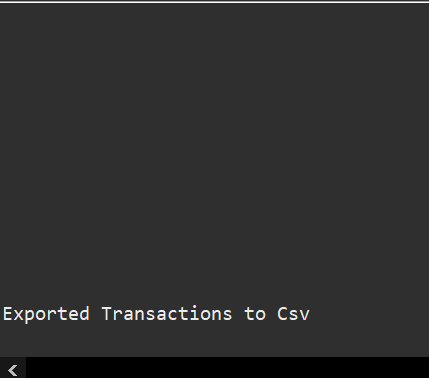


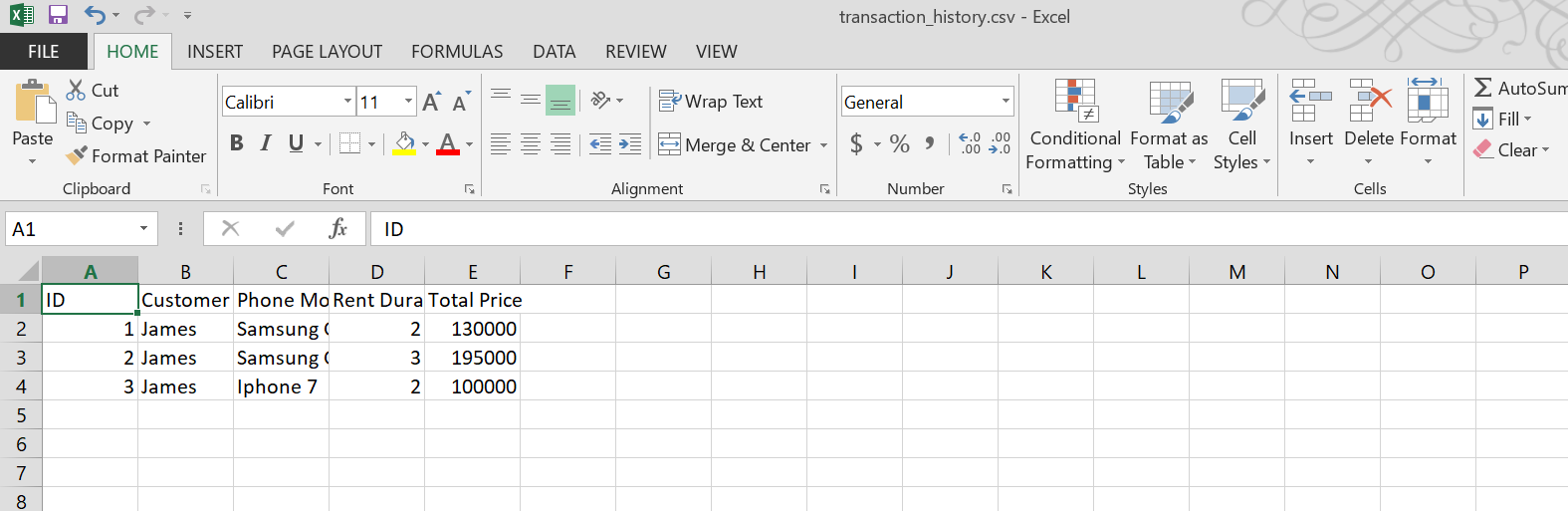
* Export Transactions to Txt





* Export Transactions to Csv





**References**

Singleton Pattern : <https://refactoring.guru/design-patterns/singleton>

State Pattern : <https://refactoring.guru/design-patterns/state>

Decorator Pattern : <https://refactoring.guru/design-patterns/decorator>

Prototype Pattern : <https://refactoring.guru/design-patterns/prototype>