**Mobile Shop Management System**



Session: 2022 – 2026

**Submitted by:**

Gul-e-Zahra 2022-CS-75

**Supervised by:**

Prof. Dr. Muhammad Awais Hassan

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

 **Department of Computer Science**

University of Engineering and Technology, Lahore

Table of Contents

Youtube’s Link3

Project Overview3

Users of Application3

Intended Functionality3

Comparison between Object Oriented Programming and Procedural Programming4

Design Pattern Implementation5

Class Details6

CRC Diagram7

Conclusion9

* **Youtube’s link:**

https://youtu.be/jnJpat-CQMw

* **Project Overview**

My project deals with the management of a mobile shop, an application which provide an elaborative interface for all the people associated with this application in any term like clients, suppliers, owner etc. In this application, the owner can see all details of his\her business, clients can go through all the desired models of mobile phone and furthermore.

* **Users of Application** 
  1. Admin: The primary users of the system are admins. They will use the system to manage their sales, customer information, and other shop-related tasks.
  2. Customers: These are the core users of the application. It provides them with an easy interface to purchase devices as per their desire.
  3. Employees: These users are updated by admins and admins maintain their record. They can see their profile using the application.
* **Intended Functionality**

|  |  |  |
| --- | --- | --- |
| **Admin** | Maintain stock | Admin can maintain stock which includes the authority to add a new item, delete any existing item and update stock of the item as well. |
| Update credentials | Admin can update credential of employees which includes changing their name, pin code and also the salary. |
| View customers | Admin can also view all customers. |
| Maintain employees | Maintaining their credentials and salary. |
| **Customer** | Availability | He can check the availability of device. |
| Purchase | Purchase any device |
| Change pin code | Customer can change his pin code. |
| Change name | Customer can also change his name. |

* **Comparison between Object Oriented Programming and Procedural Programming:**

| **Advantage** | **Object-Oriented Programming (OOP)** | **Procedural Programming** |
| --- | --- | --- |
| Modularity | OOP promotes modularity by organizing code into objects, making it easier to manage and reuse. | Procedural programming lacks inherent modularity, which can lead to code duplication. |
| Reusability | Objects and classes can be reused in different parts of the program, reducing development time. | Code reusability is limited to functions and can be more challenging to implement. |
| Encapsulation | Encapsulation hides internal details, providing a clear interface and reducing code dependencies. | Data and functions can be dispersed and less encapsulated, leading to potential issues. |
| Code Organization | OOP provides a structured way of organizing code, making it easier to understand and maintain. | Procedural programming may lack a clear structure, making code maintenance more complex. |
| Abstraction | Abstraction simplifies complex systems, focusing on essential features and hiding unnecessary details. | Procedural programming does not emphasize abstraction explicitly. |
| Inheritance | Inheritance allows for hierarchical relationships between classes, promoting code reuse. | Procedural programming does not support direct inheritance, which can lead to code duplication. |
| Polymorphism | Polymorphism enables objects of different types to be treated as objects of a common type. | Polymorphism is limited, as functions operate on specific data types. |
| Maintainability and Scalability | OOP facilitates code maintenance and scalability through modular and organized code structure. | Procedural programming can become less maintainable and scalable as codebase grows. |
| Simulates Real-World Concepts | OOP models real-world objects, behaviors, and relationships, making code more intuitive. | Procedural programming may not directly map to real-world concepts, requiring more effort. |
| Team Collaboration and Software Development | OOP allows for better team collaboration through the use of classes and objects. | Procedural programming can make collaboration challenging due to its linear nature. |

* **Design Pattern Implementation:**

1. **BL(Business Layer)**

This project utilizes the BL Design Pattern in such a way that all the BL classes contains the Business logic functions and all the variables are declared in it and all getter( ) and setter( ) functions are also included in this layer.

1. **DL(Data Layer)**

This project utilizes the DL Design Pattern in such a way that the DL classes contain all the Lists, and all functions related to Lists like AddingDataToList( ) etc. and all the CRUD functions of classes.

1. **UI(User Interface Layer)**

This project utilizes the UI Design Pattern in such a way that all Displaying Functions and Data Input Functions are included in UI classes.

* **Class Details:**

1. **Person**
   1. username: The unique username chosen by the user.
   2. password: The password associated with the user account.
   3. role: Defines the role or type of user (e.g., employee, manager, or customer).

**Methods**

1. validatePassword(password): Validates the provided password against the stored password for the user account.
2. validateName(name): Ensure that correct name is entered.
3. validateRole(password): Validates the input role according to system.
4. isUserAlreadyPresent(userName, password): Ensure that repetitive data must not enter in the list.
5. toString(): Returns string value of all attributes define in class

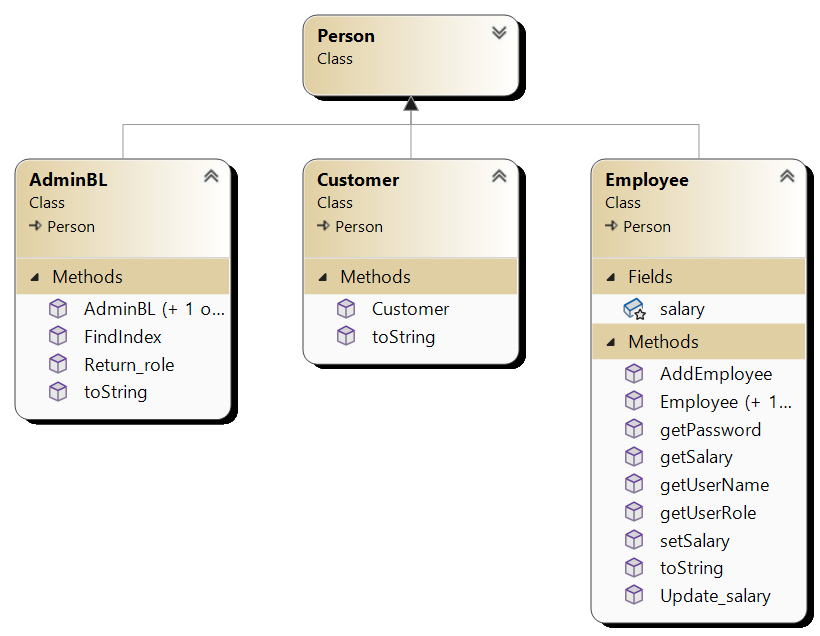
.

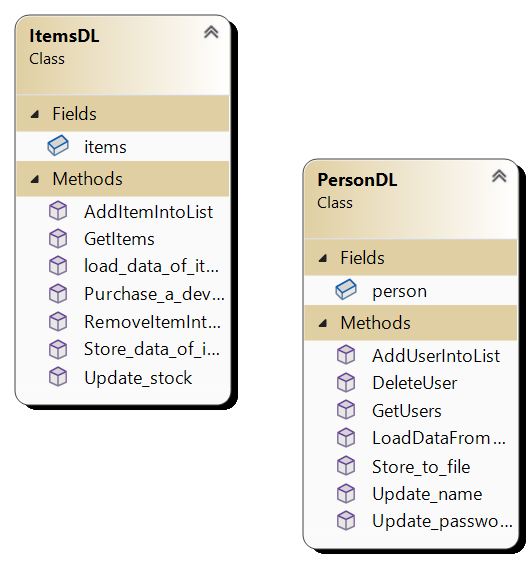
1. **Items**
2. itemName: The name or title of the item.
3. itemPrice: The price of the item.
4. itemQuantity: The quantity or stock available for the item.

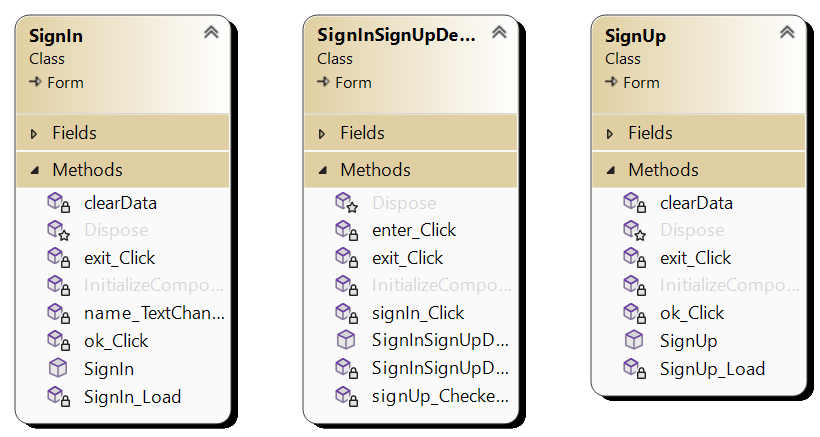
**Methods**

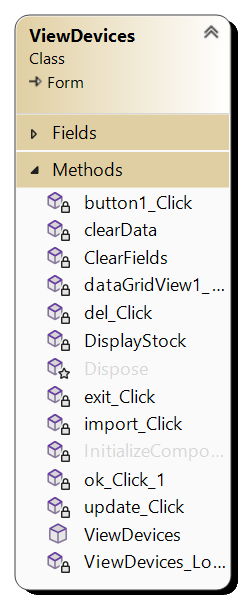
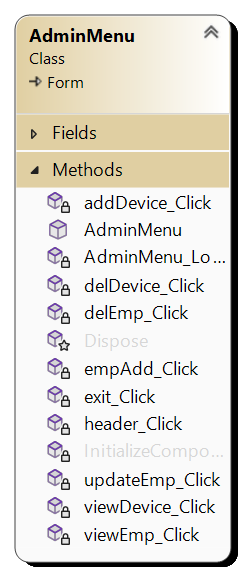
Additionally, the "Item" class may have methods to perform various operations related to items, such as:

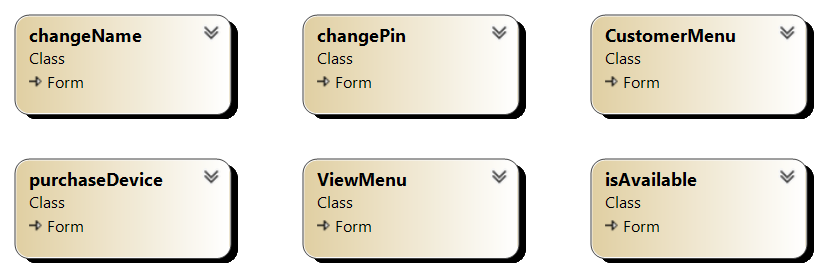
* 1. addAnItem(): Adds the specified quantity of the item to the inventory.
  2. deleteAnItem(): Removes the specified quantity of the item from the inventory.
  3. updateQuantity(): Updates multiple details of the item simultaneously.
  4. checkStockAvailability(): Checks if the item is currently in stock.
  5. getTotalValue(): Calculates the total value of the item based on its price and quantity.
* **CRC Diagram**









**Conclusion:**

The mobile shop management system project successfully applied the principles of encapsulation, inheritance, and polymorphism in the context of object-oriented programming. It achieved modular and reusable code through encapsulating data and functionality within classes, established relationships between classes through inheritance, and allowed objects of different classes to be treated uniformly through polymorphism. Challenges included designing a flexible class hierarchy and ensuring proper encapsulation, while lessons learned revolved around creating reusable and extensible class structures and effectively utilizing inheritance and polymorphism.