Object Oriented Analysis and Design

Section: L

Group :6

Members:

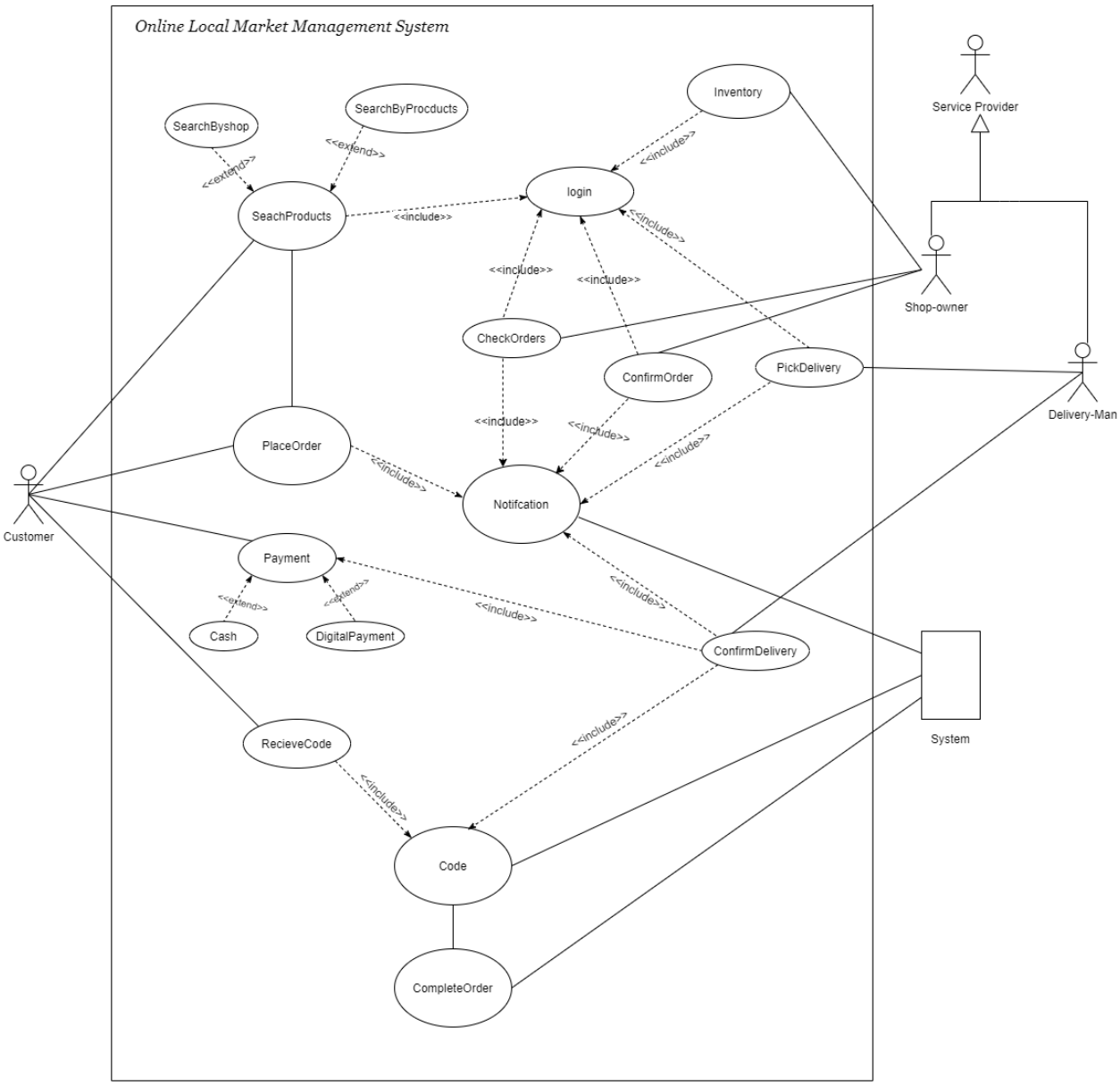
Name	ID
1. **Shakib Sadat Shanto**	**20-43074-1**
2. **Zishan Ahmed**	**20-42085-1**
3. **Jemima Tarafdar Jenee**	**19-41724-3**
4. **Sadman Saqib**	**20-43435-1**
5. **Fazle Rabbi Fahad**	**20-42717-1**

Local Market Management System

Case scenario:

In the Local Market Management system customers can buy their daily groceries from the local store in their area, sitting at their home. This system will help all the customers and shop owners to connect with each other in order to do business. To use the app users have to create their account by providing the useful information (Name, Address, Phone, NID) and have to login their account.
In this system three actors. Customer, Shop-owner and the delivery boy. The shop owner and delivery boy both are service providers and the customer is the consumer. Customers may feel the necessity to compare the price of the products .So customers can search by the product name and can compare the price of the particular product from different stores and order them. Or if a customer has his own preference about shop then he can simply browse to that store and can buy the product from there. Customers can take the order by delivery system or they can pick the order by their own from the shop. A shopkeeper can only get his shop to online when he fills his inventory with all products. When customers confirm their order from their end a simple notification will notify the shop owners about the ordered products. The order will be confirmed only when the shop owner will accept the order. After the final confirmation of the order, the app system will notify all the local delivery boys about the order. The delivery boy has to confirm the pick in order to deliver the goods. After the delivery the customers can do payment both in cash or digitally. After the payment, to close the order the system requires a code which will be sent to the customer's mobile by the system and the delivery boy has to write the code in the required place. If the code matches then the order will be marked as closed.

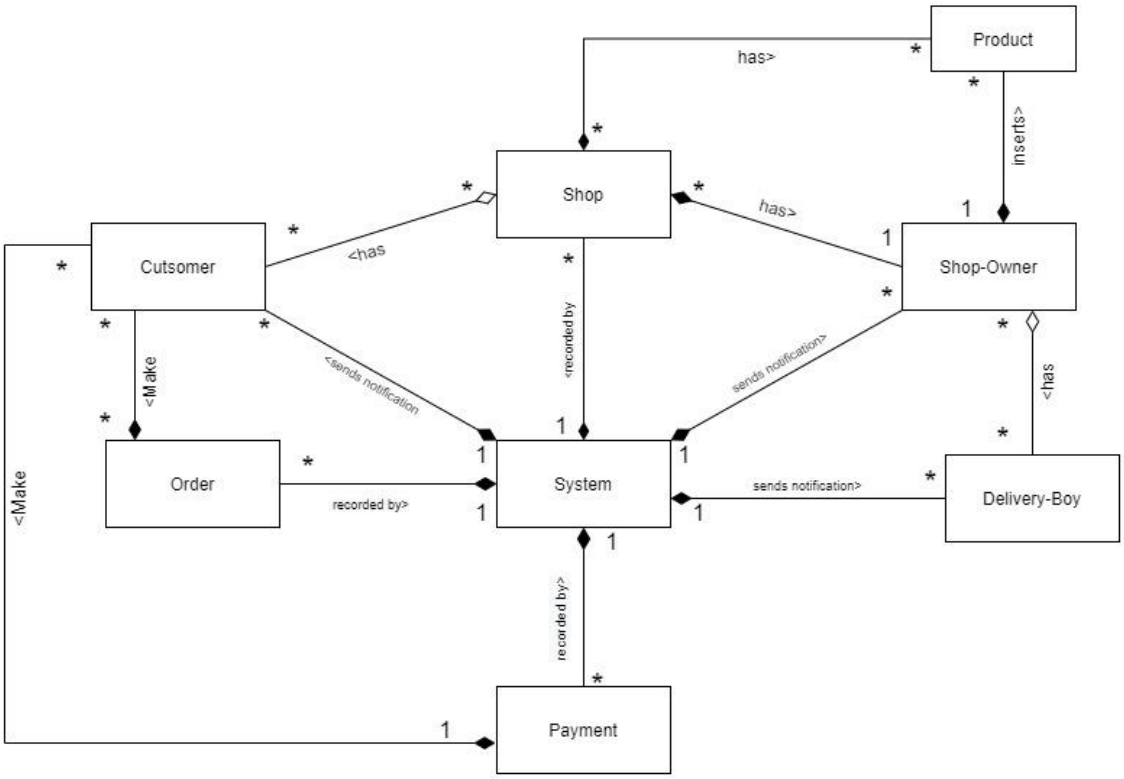
Use case Diagram:



Class-Diagram Scenario:

In the local shop management system A **Customer** class has five attributes cusname, cusPhoneNo , cusAddress, cusId and orderconfirmation. A customer may make more than one order from many shops, and **Order** class has inherited the Customer class. Shops are owned by the shopowners. A shopowner may own more than one shop. In **Shopowner** class there are three attributes , shopPin, sOwnerName, sOwnerPhoneNo, and the attributes are private. In every shop there are many products which are ordered by the customers. **Shop** class has inherited the Shopowner class and Product class. **Product** class has its own attributes. In the system there is a class called **System** class which handles everything. Every class in the management system are linked to the system class. After every successful order it will assign a delivery boy and after any successful payment it will mark the order as completed. In the System class there are seven attributes and every one of them is private. When any order is made the system notifies the every deliveryboy around the area, the one who accepts first, take the order for delivery. A **DeliveryBoy** class has three attributes , dBoyName, dBoyPhoneNo, dBoyPin. After the delivery boy delivers the product to the customer, the customer pays the bill. A customer can do payment once at a time. A **Payment** class has its own attributes which are private and inherits Customer class.

Online Local Market Management System



Class Notation

Customer
- cusName : String - cusID : String{unique & set by System} - cusPhoneNo : String - cusAddress : String - orderConfirmation : boolean
+getCustomerID() : String +setCustomerName(cusName: String) : void +getCustomeName() : String +setCustomePhn(cusPhoneNo: String) : void +getCustomePhn() : String +setCustomerAddress(cusAddress: String) : void +getCustomeAddress() : String +placeOrder(orderConfirmation:boolean): boolean +getOrderDetails(oderID: String): void

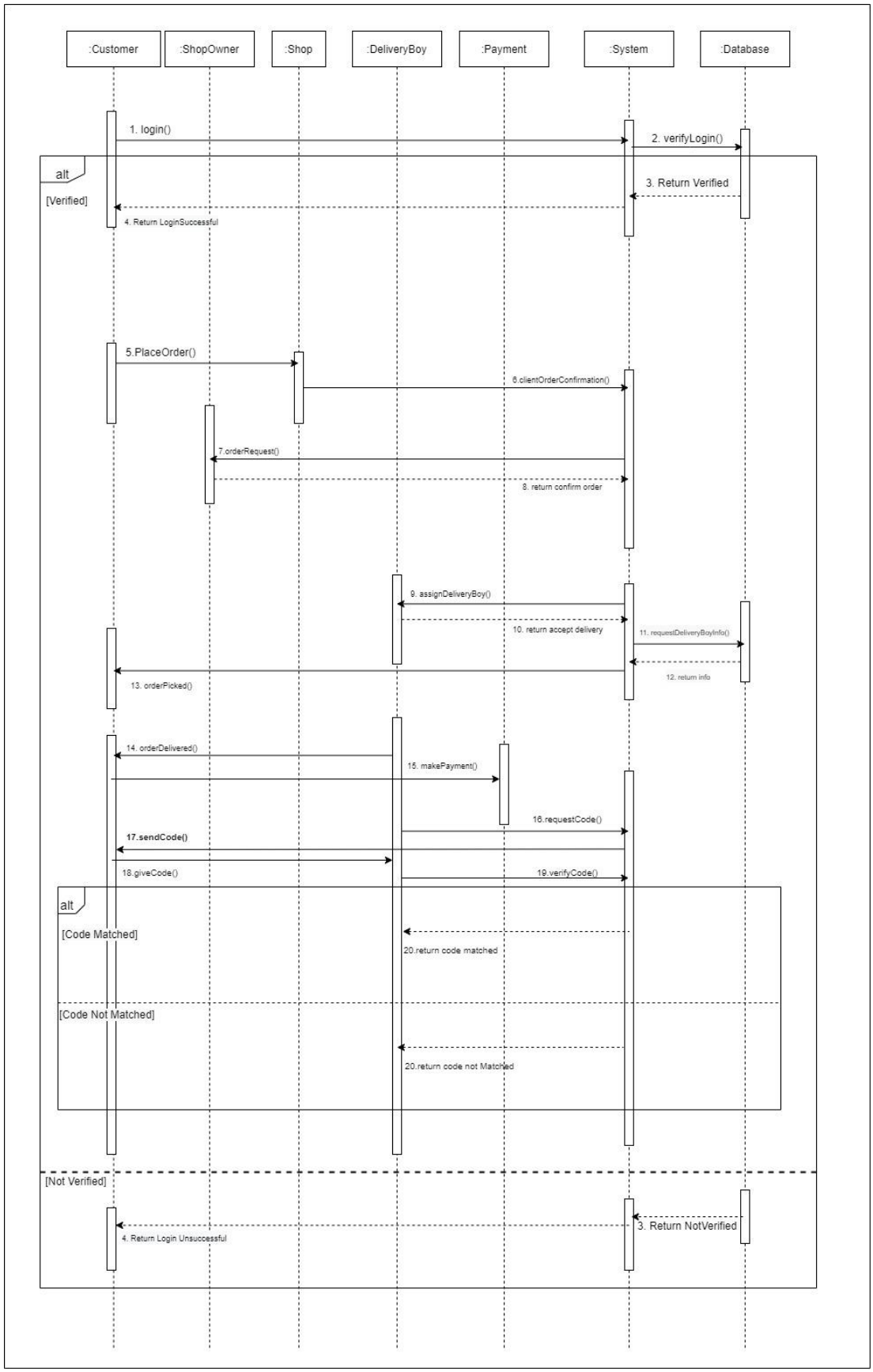
ShopOwner
- sOwnerName : String - sOwnerPhoneNo : String {set by Shop-Owner} - shopPin : String {unique & set by System}
+setShopOwnerName(sOwnerName:String) : void +getShopOwnerName() : String +setSOwnerPhone(sOwnerPhoneNo:String) : void +setSOwnerPhone () : String +getShopPin () : String +addProduct(pName: String, price:double, quantity:int):void +removeProduct(productid: String):void +checkOrder(orderID: String) :void

DeliveryBoy
- dBoyName : String - dBoyPhoneNo : String - dBoyPin : String {unique & set By System}
+setDBoyName(dBoyName:String) : void +getDBoyName() : String +setDBoyPhn(dBoyPhone:String) : void +getDBoyPhn() : String +getDBoyPin() : String +pickDelivery(oderID: String):void +cancelDelivery(oderID: String):void +paymentConfirmation (code: String): String


System
- code : String
- notification : String
- shopList[] : Shop
- shopOwnerList []: ShopOwner
- customerList[] : Customer
- deliveryBoyList[] : DeliveryBoy
- orderList[] : Order
+setCode(String : code) :void
+getCode() : String
+setShopPin(String : shopPin): void
+setDBoyPin(String : deliveryBoyPin): void
+setCusID (cusID : String) : void
+setProductID (productID : String) : void
+setOrderID(String : orderID): void
+addShop(shopName:String) : void
+removeShop(shopName:String) : void
+seachShop(shopName:String) : Shop
+addShopOwner(sOwnerName:String) : void
+removeCustomer(cusID:String) : void
+addDeliveryBoy(dBoyName: String): void
+removeDeliveryBoy(dBoyName: String): void
+searchDeliveryBoy(dBoyName: String): DeliveryBoy
+confirmOder(String: code): void
+sendNotification(String: oderID): String
Shop
- shopName : String {set by Shop-Owner}
- shopPhoneNo : String {set by Shop-Owner}
- shopAddress : String {set by Shop-Owner}
- inventory[]: Product {set by Shop-Owner}
- shopCart[] : Product {set by Customer}
+setShopName(shopName:String) : void
+getShopName() : String
+setShopPhone(shopPhoneNo:String) : void
+getShopPhone() : String
+addProductToInventory(pName: String):void
+removeProductFromInventory(pName: String):void
+seachProduct(pName:String) : Product
+addProductToCart(pName: String):void
+removeProductFromCart(pName: String):void
+showAllProducts(): void
+showShopDetails(): void
Order
- orderID : String {unique & set by system }
- amount : double
+getOrderID() : String
+placeOrder(orderID: String) : void
+amountCalculation(orderID: String) : double
+cancelOrder(orderID: String) : void
+showOrderDetails(orderID: String): void
Product
- productName : String{set by Shop-Owner}
- productID : String{unique & set by System}
- price : double {set by Shop-Owner}
-quantity : int {set by Shop-Owner}
+setProductName(productName:String) : void
+getProductName() : String
+setProductPrice(price: double) : void
+getProductPrice(): double
+getProductID(): String
+showProductDetails() : void
Payment
- paymentType : String
+setPaymentType (paymentType: String) : void
+getPaymentType () : String
+confirmPayment (orderID: String): String

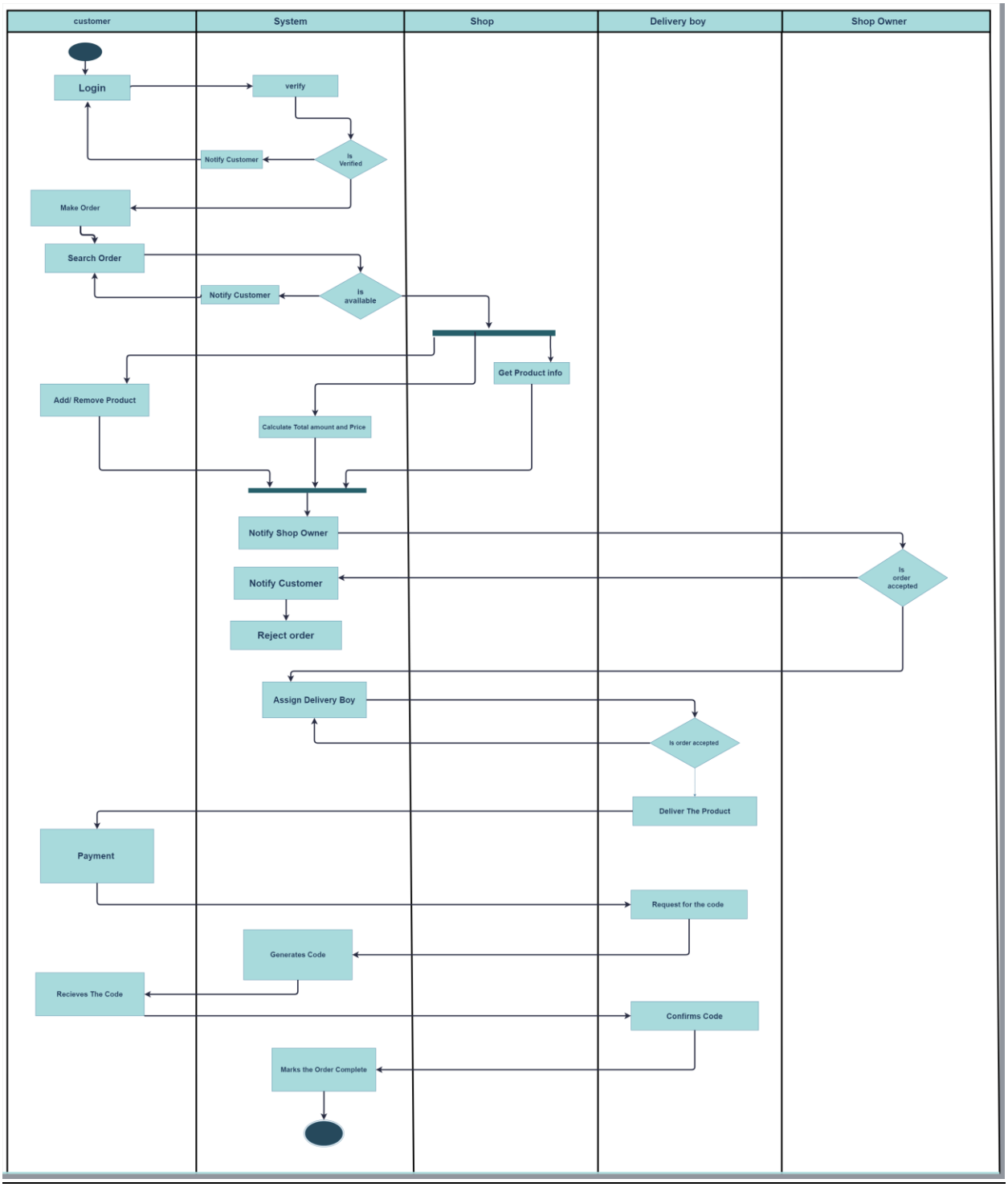
Sequence diagram scenario:

In a local manage system a customer must login in hte sysstem before he can place order. A customer have to lodin with his password. The password wil be verified by the system if the customer is authentic or not. For every unseccesful login the system will show a login error. After the successful login the customer can place the oders by searching products or direct browsing the store for place a order. When cutomer will confirm the order from his end the system eill notify the shop owners about the product.Only the order will be finalized when the shop owner will accept the order. After accepting the order the system will automatically notify all the delivery boy about hte order around the area. When a delivery boy will accept the prder, the system will automatically assign him for the order to deliver and a notifcation will be sent to the customer. When the delivery boy will deliver the product . The customer will complete his payment. After the successful payment the delivery boy will request a code confirmation code to the system in order to close the order. After system veryfied the code the order will be automatically be marked as a complete order.


:Customer
:ShopOwner
:Shop
:DeliveryBoy
:Payment
:System
:Database
1. login()
2. verifyLogin()
alt
[Verified]
3. Return Verified
4. Return LoginSuccessful
5.PlaceOrder()
6.clientOrderConfirmation()
7.orderRequest()
8. return confirm order
9. assignDeliveryBoy()
10. return accept delivery
11. requestDeliveryBoyInfo()
12. return info
13. orderPicked()
14. orderDelivered()
15. makePayment()
16.requestCode()
17.sendCode()
18.giveCode()
19.verifyCode()
alt
[Code Matched]
20.return code matched
[Code Not Matched]
20.return code not Matched
[Not Verified]
3. Return NotVerified
4. Return Login Unsuccessful

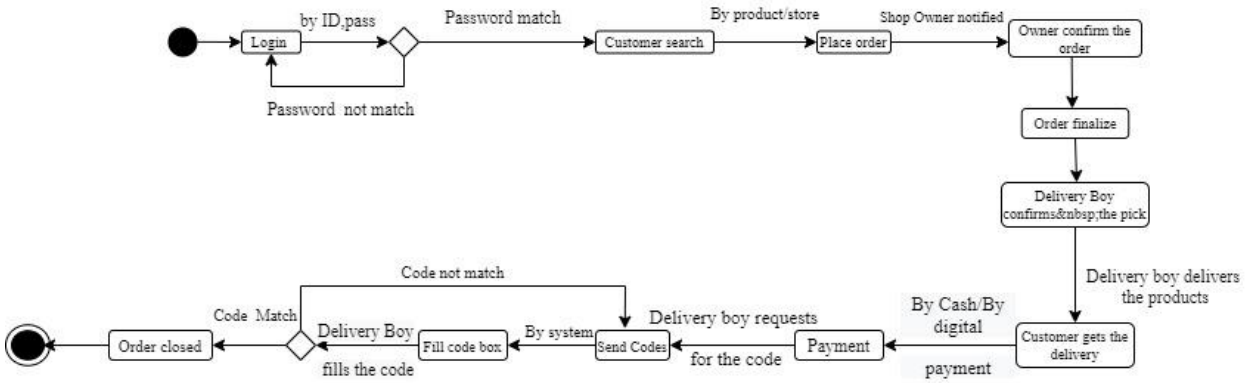
Activity Diagram Scenario:

In a Local shop management system when a customer login into the system by password, the system verifies the password. For every incorrect login the system notifies the customer and redirect him to login again. After the successful login the customer is able to place the order. For placing the order the customer can search in the system by product or by direct browsing the specific store. While searching the product a customer can get information of products for the shop. After customer confirms the order from his end the system instantly notifies the shop owners, an order will be finalized only when the shop owner will accept the order. If a shop owner reject the order, system will notify the customer and reject the particular products in the order. After finalizing the order a notification will be sent to the delivery boys around his area. When the delivery boy accepts the order, the system will assign him for the delivery. After delivering the product the customer has to complete his payment. After the payment in order to close the order the delivery boy will request a code to the system, the system will send the code to the customer. Customer has to shoe the code to the delivery boy and he will put the code in to the code verification box. If the code matches the order will be marked as closed and if not the system will again send the code to the customer for verifying the code.


customer
System
Shop
Delivery boy
Shop Owner
Login
verify
Is Verified
Notify Customer
Make Order
Search Order
Notify Customer
Is available
Get Product info
Add/ Remove Product
Calculate Total amount and Price
Notify Shop Owner
Is order accepted
Notify Customer
Reject order
Assign Delivery Boy
Is order accepted
Deliver The Product
Payment
Request for the code
Generates Code
Recieves The Code
Confirms Code
Marks the Order Complete

State chart diagram Scenario:

In the local Market Management System, customer objects have to login to the system by customer ID and password. If it matches customer can search any product or store. But if it does not match system will take him to the login page. After searching product or store, he can place an order. The system notifies the shop owner and owner to confirm the order. The will be finalized by the owner. A delivery boy confirms the pick when he gets notified by the system. Delivery boy delivers the products. After getting delivery customer does his payment with cash or digital payment. The delivery boy requests the code to the customer. The customer gets a code by system. The delivery boy fills the code. If the code matches order closed. But if the code does not match system will send another code to the customer.



COCOMO (COnstructive COst MOdel)

In this project :-

- Software Project Type: Semi-detached
- SLOC (Source Lines Of Code) : 5000
- P (Project Complexity) : 1.12
- Coefficient<Effort factor> : 3.0
- T (SLOC Dependent Coefficient) : 0.35

1. Effort/PM(Person-months needed for Project)

Coefficient$_{\text{<Effort Factor>}}$*(SLOC/1000)^P

= 3.0 * (5000/1000)^1.12

= 18.195

2. Development time /DM

2.50*(PM)^T

= 2.50*(18.195)^0.35

= 6.901

3. Required Number Of People/ ST

PM/DM

= 18.195 /6.901

= 2.63

Contribution:

1. **Shakib Sadat Shanto** – Use case
2. **Zishan Ahmed** – Class diagram & COCOMO
3. **Jemima Tarafdar Jenee** – Sequence Diagram
4. **Sadman Saqib** – Activity Diagram
5. **Fazle Rabbi Fahad** – State chart Diagram