CS-201 FINAL SEM PROJECT

Team Members:-

Om Kumar Thakur (201952223)

Shashwat Mishra (201951141)

Anavya Upadhyay (201951023)

Prahlad Ghosalya (201951115)

Project : Smart Billing Software

How the idea came in our mind !

Lack of the systematic billing system .

Many of the shops don’t have a proper and organized billing system which sometimes result in the tedious task of manual bill calculation leading to the waste of time and resources. Due to this there are chances of ambiguity between customer and shopkeeper related to the bill.

Our Solution !!

To make the system hassle free and time efficient , we thought of a solution that could make the system fast, smooth and transparent. So, we came up with this idea of the “Smart Billing Software”, which is a billing platform which enables both the customer as well as the shop owner to maintain and carry on their transaction on this platform.

- Saving in labor due to self-service system.

- Low cost of operation.
- Freedom of selection.
- Shopping is very easy and quick

Program Code:-

```
/**
 * @author Om Kumar Thakur, Shashwat Mishra, Anavya Upadhyay and Prahalad Ghosalya
 *
 * Implementing the Smart Billing Software, using the OOPS features in JAVA
 */

package firstJavaProgram;
import java.util.*;


interface Operation  //Interface
{

public void login_staff();
public void login_customer();
public void menu_display();
public void  Add(int code,String name,int quant,int price);
public void  Search(Billing_System []obj,int size);
public void Edit(Billing_System obj[],int size);
public void Delete(Billing_System []obj,int size);
public void bill_calc(Billing_System obj[],int size);
public void bill_display(Billing_System obj[],int size);
}



class Billing_System implements Operation //Super Class
{

// Code of each item brought
// Quanity of each item
// Price of each item
// Name of each item


        private int code,quantity,price;
        private String name;
```

```
    Billing_System() //Non Parameterized Constructor
       {
             this.code=0;
             this.quantity=0;
             this.price=0;
             this.name="";
       }


   public void login_staff()  //Login for Staff_Members
   {
    Scanner sc = new Scanner(System.in);


       System.out.println("\n\n\n\n\n\n\n\n");
       System.out.println("
<<<< Staff Login Credentials >>>>                                           ");
       System.out.println("
___________________________________                 ");
       System.out.println("\n\n");
       System.out.print("
Name : ");

       String uname = sc.nextLine();
       System.out.print("\n
UID  : ");
       String pass = sc.nextLine();
       System.out.print("\n\n\n\n\n");
       System.out.print("\n\n\n\n\n\n\n...........................................
..................................................................................
..................................................................................
............................\n\n");

  }



   public void login_customer() //Login for Customers
   {
    Scanner sc = new Scanner(System.in);

    System.out.println("\n\n\n\n\n\n\n\n");
       System.out.println("
<<<< Customer Login Credentials >>>>                                           ");
       System.out.println("
___________________________________                 ");
       System.out.println("\n\n");
       System.out.print("
Name      : ");

       String uname = sc.nextLine();
       System.out.print("\n
Phone No. : ");
       String phone = sc.nextLine();
       System.out.print("\n\n\n\n\n");
       System.out.print("\n\n\n\n\n\n\n...........................................
..................................................................................
..................................................................................
............................\n\n");

  }

 // Method for menu Display to choose add, delete and edit item

 public void menu_display()
 {


    System.out.print("\n\n\n\n\n\n\n
BILLING SYSTEM   ");
    System.out.print("\n
________________");
    System.out.print("\n\n\n
Main Menu");
System.out.print("\n\n\n\n\n
1. Add Goods");
    System.out.print("\n
2. Search");
    System.out.print("\n
3. Edit Goods");
    System.out.print("\n
4. Delete Goods");
    System.out.print("\n
5. Calculate Bill");
    System.out.print("\n
6. Display Bill");
    System.out.print("\n
7. Customer's Feedback");
    System.out.print("\n
8. Exit");
    System.out.print("\n\n\n\n
Please enter the choice : ");

 }



// Method for adding items to the cart

public void  Add(int code,String name,int quant,int price)
 {

   this.code=code;
   this.name=name;
   this.quantity=quant;
   this.price=price;

 }




 public void Search(Billing_System obj[],int n) // Method for searching the item
 {

   int f=0;
   Scanner sc=new Scanner(System.in);

   if(obj[0].code==0) // If no item has been added then cart is empty
   System.out.print("
NO ITEM ADDED !!");

    else // If present
    {
     System.out.print("\n\n\n\n\n
************* SEARCH MENU *************");
     System.out.print("\n\n\n\n\n
1.By Code");
     System.out.print("\n
2.By Name");
     System.out.print("\n\n\n
Enter option 1 or 2 : ");

     int option=sc.nextInt(); // Entering Option to search item by code or name

    if(option==1)  // Searching for the item by its code
    {
         System.out.print("\n
Enter the code of the item : ");
      int codes=sc.nextInt();
      int j=n-1;


    while(j>-1)   // Checking for item
    {

    if(codes==obj[j].code)
     {
     f++;
     break;
     }
     j--;

    }

    if(f>0)   // If present
    {
      System.out.println("\n\n\n\n\n\n\n\n");
      System.out.println("                                                                            Code
Name                               Quantity                                Price\n\n");
      System.out.println("
"+obj[j].code+"                                             "+obj[j].name+"
"+obj[j].quantity+"                                         "+obj[j].price);
    }

    else
     System.out.println("
************ Item not found  ********");

    }//if condition for option 1 ends


    if(option==2)  // Searching for the item by its name
    {
     System.out.print("\n
Enter the name of the item : ");
     sc.nextLine();
     String names=sc.nextLine();
     int j=n-1;

    while(j>-1)
    {
      if(names.equalsIgnoreCase(obj[j].name))
      {
        f++;
      break;
      }
     j--;
    }

    if(f>0)
    {
      System.out.println("\n\n\n\n\n\n\n\n");
      System.out.println("                                                                            Code
Name                           Quantity                         Price\n\n");
      System.out.println("
"+obj[j].code+"                              "+obj[j].name+"
"+obj[j].quantity+"                              "+obj[j].price);
    }

     else
      System.out.println("
************ Item not found  ********");

      }  //if condition for option 2 ends

    //menu_display();

    }//else ends

  }// search function ends


public void Delete(Billing_System obj[],int n) //Method to delete any item present
    {

     //Delete the item by either using their code or name
       //
System.out.print("\n\n\n\n\n\n\n\n\n.............................................
..................................................................................
.........................................\n\n");

         int f=0;
         Scanner sc=new Scanner(System.in);

         System.out.print("\n\n\n\n\n
************* DELETE MENU **************");
         System.out.print("\n\n\n\n\n
1.By Code");
         System.out.print("\n
2.By Name");
```

```
          System.out.print("\n\n\n
Enter option 1 or 2 : ");

          int option=sc.nextInt();

         if(option==1)
         {
             System.out.print("\n
Enter the code of the item : ");
          int codes=sc.nextInt();
          int j=n-1;


         while(j>-1)
         {

         if(codes==obj[j].code)
         {
          f++;
          break;
         }
          j--;
        }

         if(f>0)
        {
               obj[j].code=0;
               obj[j].name=" ";
               obj[j].quantity=0;
               obj[j].price=0;

             bill_display(obj,n);
             //menu_display();
         }

         else if(f==0)
         {
               System.out.print("
Item not found !!");
                    //menu_display();
         }

       }//if condition for option 1 ends



    if(option==2)
       {
          System.out.print("\n
Enter the name of the item : ");
          sc.nextLine();
          String names=sc.nextLine();

           int j=n-1;

         while(j>-1)
        {
         if(names.equalsIgnoreCase(obj[j].name))
         {
               f++;
           break;
         }
           j--;
       }

        if(f>0)
         {
               obj[j].code=0;
               obj[j].name=" ";
               obj[j].quantity=0;
               obj[j].price=0;

         }

         else
         System.out.println("
************Item not found********");

          }  //if condition for option 2 ends

}   //Delete() ends




   // Method to edit the details of the item present

   public void Edit(Billing_System obj[],int n)
    {

      Scanner sc=new Scanner(System.in);


      System.out.print("\n\n\n\n\n\n\n
Do you want to edit the record ? YES/NO : ");
      String choice =sc.nextLine();

     if(choice.equalsIgnoreCase("YES"))
     {
       System.out.print("\n\n
Enter the code of the quantity : ");

       int codes=sc.nextInt();
       int index= get_index(obj,codes,n);

      if(index>=0 && index<n)
      {

       System.out.print("\n\n\n
1.Edit Quantity");
       System.out.print("\n
2.Edit Price (Not Available for Customers)");
       System.out.print("\n\n\n
Enter option (1 or 2) : ");

       int op=sc.nextInt();  //Takes Option 1 or 2
```

```
        if(op==1)
        {
       System.out.print("\n\n
Enter new Quantity : ");
       int quaantity=sc.nextInt();
       obj[index].quantity=quaantity;
       System.out.println("\n\n\n\n\n\n
***************** Item Edited and Detail is updated ******************
");
        }


      if(op==2)
       {
       System.out.print("\n
Enter new Price : ");
       int priice=sc.nextInt();
       obj[index].price=priice;
       System.out.println("\n\n\n
**************** Item Edited and Detail is updated ***************
");
       }

     }//if ends

    }//outer if ends


    else if(choice.equalsIgnoreCase("NO"))
     {
          System.out.println("\n
No detail is to be updated !!\n\n");
          //menu_display();
     }

     else
     {
      System.out.println("\n
Invalid Code !!");
      //menu_display();
     }

} //Edit method ends




// Method to return the index of the searched item

private int get_index(Billing_System obj[],int codes,int n)
 {

int j=n-1;

 while(j>-1)
 {

   if(codes==obj[j].code)
```

```
    return(j);

    j--;
  }
   return -1;

 } //get_index() ends



  // Method to calculate the bill of the item present

  public void bill_calc(Billing_System obj[],int n)
  {

    System.out.println("\n\n\n\n\n\n\n\n\n\n\n");
    System.out.println("
******************* BILL CALCULATION ************************");
    System.out.println("\n\n\n\n
Code                               Name                               Quantity
Price\n\n");
    int total=0;

    for(int j=0;j<n;j++)
    {
        if(obj[j].code>0)
        System.out.println("
"+obj[j].code+"                               "+obj[j].name+"
"+obj[j].quantity+"                               "+obj[j].price);
        total+=obj[j].price;
    }

   System.out.println("\n\n\n\n
GRAND TOTAL : "+total+"\n\n");

   //menu_display();

  } //bill_calc() ends



   // Method to display bill of the item present

   public void bill_display(Billing_System obj[],int n)
   {
      //System.out.print("\n\n\n\n..............................................
..................................................................................
........................................\n\n");

      System.out.println("\n\n\n\n\n\n\n");
      System.out.println("
******************* INVOICE BILL ************************");
      System.out.println("\n\n\n\n\n
Code                               Name                               Quantity
Price\n\n");

      int total=0;
```

```
       for(int j=0;j<n;j++)
       {
             if(obj[j].code !=0)
             {
              System.out.println("
"+obj[j].code+"                              "+obj[j].name+"
"+obj[j].quantity+"                              "+obj[j].price);
             total+=obj[j].price;
             }
       }
         //menu_display();

  } // bill_display() ends

} // Super Class ends




class Staff_emp extends Billing_System
{

  Billing_System sf=new Billing_System();

// Creating array of Object staff[] of each item for storing the details like
code,name,quantity and price

  Billing_System st[]=new Billing_System[100];
  Scanner sc=new Scanner(System.in);

  int size=0;        // The number of items bought

  Staff_emp()
  {
   sf.login_staff();
  }

  private void add_item()
  {
            System.out.print("\n\n\n\n\n\n\n
**************  ADD ARTICLES  **************\n\n\n");

              System.out.print("\n\n\n
Enter the code of item : ");
               int code=sc.nextInt();
               sc.nextLine();

               System.out.print("
Enter the name of item : ");
               String name=sc.nextLine();

               System.out.print("
Enter quantity of item : ");
               int quant=sc.nextInt();
               System.out.print("
Enter price of item    : ");
               int price=sc.nextInt();

        st[size]=new Billing_System();
        st[size].Add(code,name,quant,price);

        size++;

        sc.nextLine();
        System.out.print("\n\n\n
Do you want to add more items? (YES/NO) : ");
        String option=sc.nextLine();

       if(option.equalsIgnoreCase("YES"))
              add_item();

          else if(option.equalsIgnoreCase("NO"))
          {
               menu_driven();
          }

  } //add_item() ends



  private void searching()
  {

      sf.Search(st,size);

      System.out.print("\n\n\n\n\n\n
Do you want to search more items? (YES/NO) : ");
        String option=sc.nextLine();

        if(option.equalsIgnoreCase("YES"))
          searching();

        else if(option.equalsIgnoreCase("NO"))
         {
         menu_driven();
         }

  } //searching() ends



  private void delete()
   {
      sf.Delete(st,size);

        System.out.print("\n\n\n\n\n\n\n
Do you want to delete more items? (YES/NO) : ");
        String option=sc.nextLine();

        if(option.equalsIgnoreCase("YES"))
        delete();

        else if(option.equalsIgnoreCase("NO"))
```

```
         {
         menu_driven();
         }

   } //delete()ends


  private void edit()
   {
   sf.Edit(st,size);

      System.out.print("\n\n\n\n\n\n\n
Do you want to edit more items? (YES/NO) : ");
        String option=sc.nextLine();

        if(option.equalsIgnoreCase("YES"))
            edit();

        else if(option.equalsIgnoreCase("NO"))
         {
         menu_driven();
         }

   } //edit() ends


  private void display()
   {
   sf.bill_display(st,size);
   menu_driven();

   } //display() ends


  private void calc()
   {
   sf.bill_calc(st,size);
   menu_driven();

   } //calc() ends


public void menu_driven()  // Method to perform all the task as the staff of shop
       {
      //Starting of the main flow of the program

      Scanner sc=new Scanner(System.in);

      sf.menu_display();

      int ch= sc.nextInt();
      System.out.print("\n\n\n\n\n\n\n");

System.out.print("\n\n\n\n........................................................
..................................................................................
..........................................................................\n\n");
```

```
      while(ch>=1 && ch<=8)
      {

        switch(ch)
        {

        case 1:
                 {
                   add_item();
                    break;
                  }

        case 2: {
                   searching();
                   break;
                 }

        case 3: {
                   edit();
                   break;
                 }

        case 4: {
                  delete();
                  break;
                 }

        case 5: {
                   calc();
                  break;
                 }

        case 6: {
                   display();
                 break;
                  }

        case 7: {
                System.out.println("\n\n\n
....................... EXIT .......................\n");
                System.exit(0);
             break;
              }

        case 8: {
                   System.out.println("\n\n\n
........................ EXIT .......................\n");
                   System.exit(0);
                 break;
                 }

       default: {
                 System.out.println("\n
Please enter the correct choice");
                 break;
                   }
```

```
          } //switch ends

        }//while ends

      } //function ends


  } //Staff (SubClass) ends







class Consumer extends Billing_System
{

      Billing_System sf=new Billing_System();

// Creating array of Object cons[] of each item for storing the details like
code,name,quantity and price

Billing_System cons[]=new Billing_System[100];

        Scanner sc=new Scanner(System.in);

        int size=0;


        Consumer()
        {
        sf.login_customer();
        }


        private void add_item()
        {
                  System.out.print("\n\n\n\n\n\n\n
**************  ADD ARTICLES  **************\n\n\n");

                    System.out.print("\n\n\n
Enter the code of item : ");
                     int code=sc.nextInt();
                     sc.nextLine();

                     System.out.print("
Enter the name of item : ");
                     String name=sc.nextLine();

                     System.out.print("
Enter quantity of item : ");
                     int quant=sc.nextInt();

                     System.out.print("
Enter price of item    : ");
```

```
                      int price=sc.nextInt();

               cons[size]=new Billing_System();
               cons[size].Add(code,name,quant,price);

               size++;

               sc.nextLine();
               System.out.print("\n\n\n
Do you want to add more items? (YES/NO) : ");
               String option=sc.nextLine();

              if(option.equalsIgnoreCase("YES"))
                     add_item();

                 else if(option.equalsIgnoreCase("NO"))
                 {
                      menu_driven();
                 }

         } //add_item() ends



         private void searching()
         {

             sf.Search(cons,size);

             System.out.print("\n\n\n\n\n\n
Do you want to search more items? (YES/NO) : ");
               String option=sc.nextLine();

               if(option.equalsIgnoreCase("YES"))
                 searching();

               else if(option.equalsIgnoreCase("NO"))
                {
                menu_driven();
                }

         } //searching() ends



         private void delete()
          {
             sf.Delete(cons,size);

               System.out.print("\n\n\n\n\n\n\n
Do you want to delete more items? (YES/NO) : ");
               String option=sc.nextLine();

               if(option.equalsIgnoreCase("YES"))
               delete();

               else if(option.equalsIgnoreCase("NO"))
                {
                menu_driven();
                }

         } //delete() ends



        private void edit()
         {
         sf.Edit(cons,size);

            System.out.print("\n\n\n\n\n\n\n
Do you want to edit more items? (YES/NO) : ");
               String option=sc.nextLine();

               if(option.equalsIgnoreCase("YES"))
                    edit();

               else if(option.equalsIgnoreCase("NO"))
                {
                menu_driven();
                }

         } //edit() ends



        private void display()
         {
         sf.bill_display(cons,size);
         menu_driven();

         } //display() ends


        private void calc()
         {
         sf.bill_calc(cons,size);
         menu_driven();

         } //calc() ends



        private void feedback()
         {

             System.out.print("\n\n\n\n\n\n\n
Do you want to give the feedback (YES/NO)  : ");
             String ch=sc.nextLine();


             if(ch.equalsIgnoreCase("YES"))
             {
           System.out.print("\n\n\n
Write to us : ");
              String feed=sc.nextLine();
              System.out.println("\n\n\n
THANK YOU FOR YOUR FEEBACK !!!\n\n\n");
              menu_driven();
             }

             if(ch.equalsIgnoreCase("NO"))
             {
                   System.out.println("\n\n\n\n\n\n
HAVE A GOOD DAY :) \n\n");
                   menu_driven();
             }

         }



public void menu_driven() // Method to perform all the task as the Customer
              {
             //Starting of the main flow of the program

             Scanner sc=new Scanner(System.in);

             sf.menu_display();

             int ch= sc.nextInt();
             System.out.print("\n\n\n\n\n\n\n\n");

System.out.print("\n\n\n\n........................................................
..................................................................................
..........................................................................\n\n");



             while(ch>=1 && ch<=8)
             {

               switch(ch)
               {

               case 1:
                        {
                          add_item();
                           break;
                         }

               case 2: {
                          searching();
                          break;
                       }

               case 3: {
                          edit();
                          break;
                       }

               case 4: {
                         delete();
                         break;
                       }

               case 5: {
                          calc();
                         break;
                       }

               case 6: {
                          display();
                       break;
                        }

               case 7: {
                       feedback();
                     break;
                     }

               case 8: {
                          System.out.println("\n\n\n
........................ EXIT .......................\n");
                          System.exit(0);
                       break;
                       }

              default: {
                       System.out.println("\n
Please enter the correct choice");
                       break;
                          }

                 } //switch ends

               }//while ends

             } //function ends

     }//Consumer (SubClass)  ends




    public class SuperMarket_Billing_System
    {
     public static void main(String[] args)
      {

       Billing_System sf=new Billing_System();
       Scanner sc = new Scanner(System.in);

       System.out.println("\n\n\n\n
>>>>>>>>>>>>>>>>   WELCOME TO THE  XYZ SUPERMARKET   <<<<<<<<<<<<<<<<");
       System.out.println("\n
**************************************");
       System.out.print("\n\n\n\n\n
Do you want to continue as Staff or Customer (Staff/Customer)   : ");

       String identity=sc.nextLine();

       if(identity.equalsIgnoreCase("Staff"))
       {
```

```
    Staff_emp obj1=new Staff_emp();
    obj1.menu_driven();
    }

    if(identity.equalsIgnoreCase("Customer"))
    {
    Consumer obj2=new Consumer();
     obj2.menu_driven();
    }

   }//main method ends
}   //main class ends
```

--------------------- **Code Completion** -------------------------------

Key Features of OOPS implemented :-

The code has been done for calculating and storing the bill per customer and and it is accessible by the staff of the shop also .

Inheritence :-

In this project , we have implemented the inheritance by inherting the features of the Billing System **(SuperClass)** using the Staff_emp and Customer class **(Subclasses).**

Polymorphism :-

Here ,we tried to implement this features by implementing the Operation **(interface)** in the Billing_System **(Superclass)** since it's an abstract class and the two **(subclasses)** Staff_emp and Customer class are using the features of the interface in their own way.

Abstraction :-

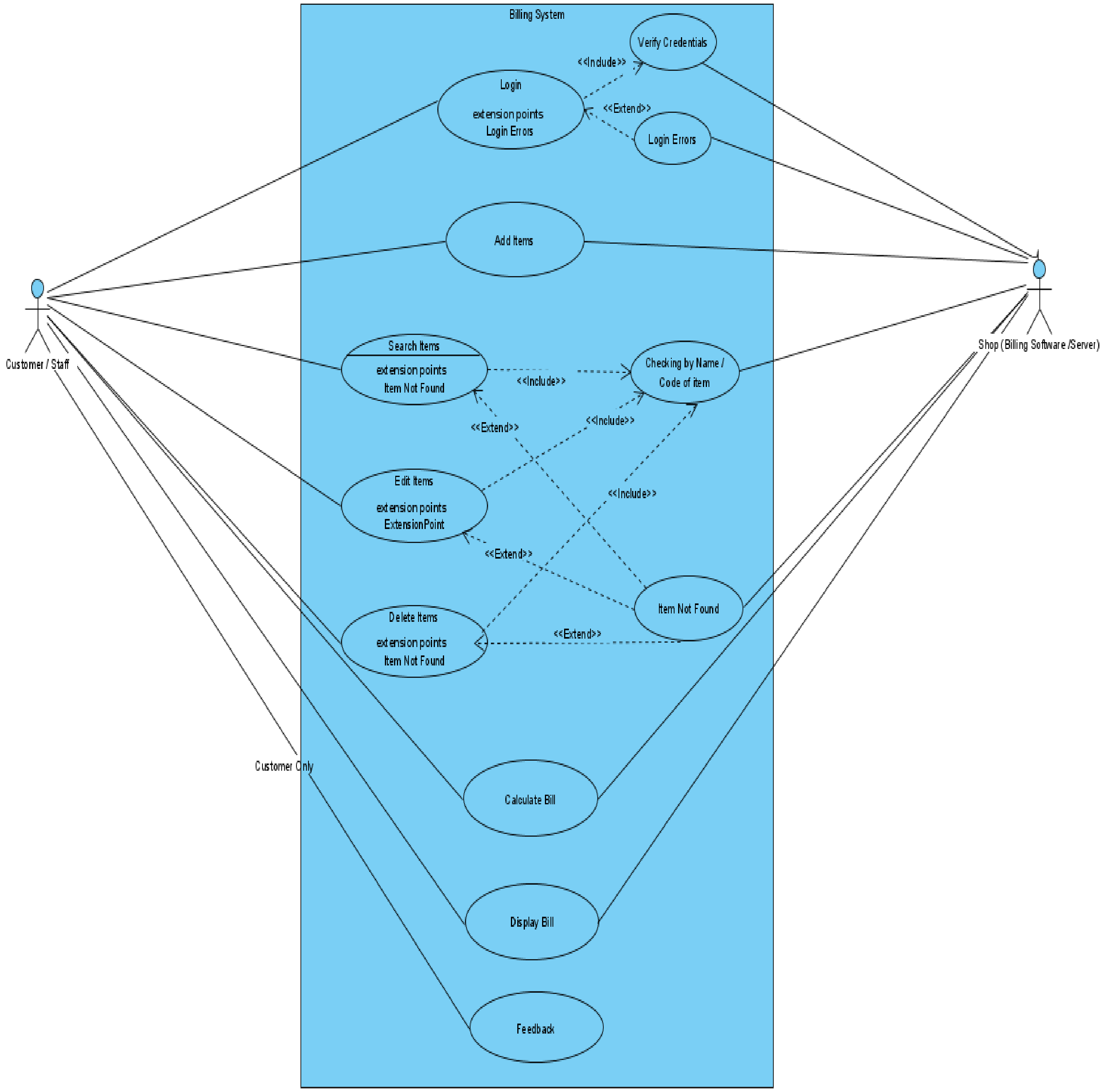
We have used this feature in our code by creating an array of object in the **(subclasses)** for each item without knowing the inside details of the item object. And we also have used some of the behaviours of the object without knowing the actual implementation of those.

Encapsulation:-

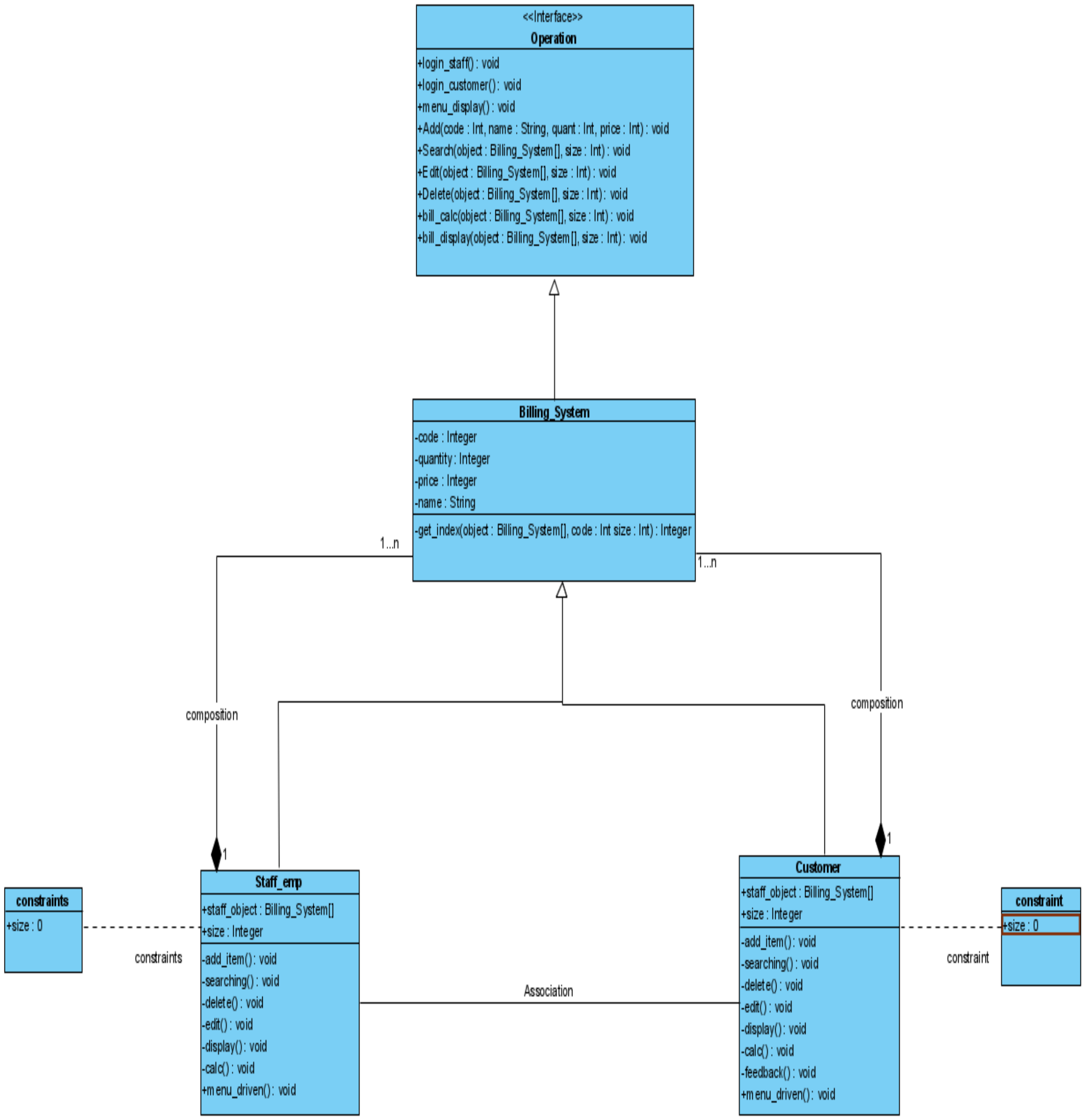
This feature has been implemented by the creation of the objects of the superclass and using their features.

UML Diagram of the Project

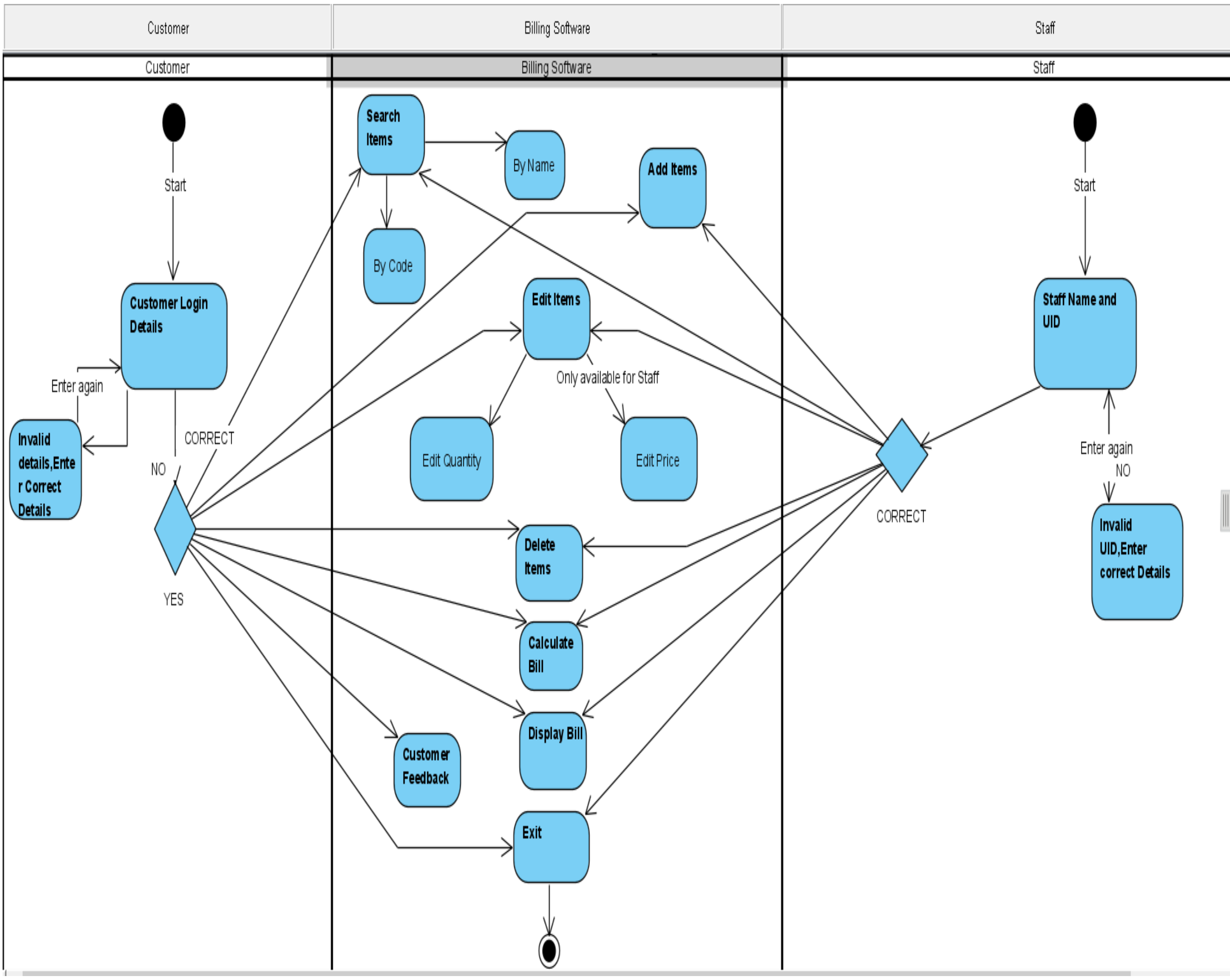
- ## User Case Diagram



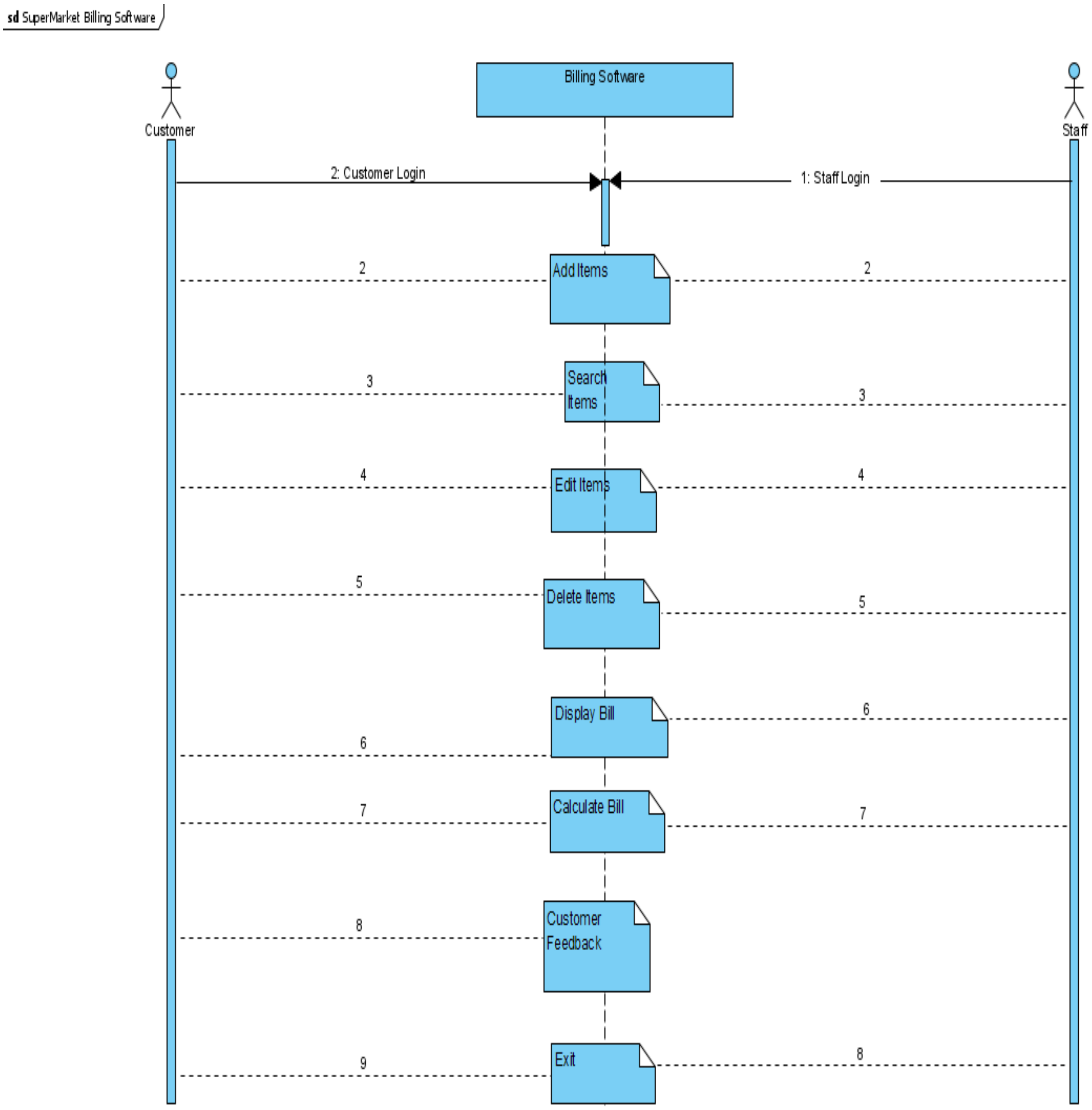
- **<u>Class Diagram</u>**



- **<u>Activity Diagram</u>**


Customer
Billing Software
Staff
Customer
Billing Software
Staff
Start
Customer Login Details
Enter again
Invalid details,Enter Correct Details
CORRECT
NO
YES
Search Items
By Name
By Code
Add Items
Edit Items
Only available for Staff
Edit Quantity
Edit Price
Delete Items
Calculate Bill
Display Bill
Customer Feedback
Exit
Start
Staff Name and UID
Enter again
NO
Invalid UID,Enter correct Details
CORRECT

- **Sequence Diagram**


sd SuperMarket Billing Software
Customer
Billing Software
Staff
2: Customer Login
1: Staff Login
2
Add Items
2
3
Search Items
3
4
Edit Items
4
5
Delete Items
5
Display Bill
6
6
7
Calculate Bill
7
8
Customer Feedback
9
Exit
8