SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering
Program: BTech, Sem III
A.Y. 2022-23
Course: Object Oriented Programming

Project Report

SUPERMARKET BILLING SYSTEM AND INVENTORY MANAGEMENT

Group Members:
1. Chinmayi Desai - E014
2. Heer Dhandhukia - E015
3. Sarthak Girish - E020

Introduction/Problem Statement:
The project is on Supermarket Billing system and inventory management. It is an application by which fast billing, stock maintenance, management of price and discounts of items, etc. can be done in an efficient manner. The project is done in C++ programming language using the concepts of OOP and file handling.

Application/Usefulness of problem statement chosen:
We have used the following Object-Oriented Programming concepts in our project:
1. Classes and Objects
   - Class Customer_Details
   - Class Items
   - Class Admin
2. Inheritance
3. File handling

Functionalities:

- string readData(): It basically asks the customer its details which include Name, Contact number, Address and Payment method.
- int selectCategory(): It allows the customer to select the categories from the list in the supermarket.
- void displayData(): It displays the customer details and items selected in the Invoice.
- void update_record(): It updates the units of selected items by making changes in the csv files of the respective categories. It also shows if the selected item id Out of Stock or if insufficient items are present.
- float selectItems(): It allows customer to select the items and also decide the number of units they want to purchase. It also calculates the total amount.
- void calcDiscount(): It calculates the discounts based on the total amount of shopping done, and also displays the amount in the Invoice.
- void create_account(): It creates an account for the Admin by setting a valid username and password which must accept all the given conditions.
- void display_record(): It displays the data of the csv files created for different categories.
- void display_bnh(): It displays records of “Beauty & Hygiene” category.
- void display_bv(): It displays records of “Beverages” category.
- void display_dr(): It displays records of “Dairy” category.
- void display_fnv(): It displays records of “Fruits & Vegetables” category.
- void display_sn(): It displays records of “Snacks” category.
- void addItems(): It adds units to a particular item and updates the csv files.
- void viewCategories(): It lets the admin view the database of a single category or all categories at a time.
- void updateCategories(): It lets the admin update the database by adding units by selecting a particular item of a category.
- void login(): It is for the Admin to login into the system. It gives 3 tries to log into the system after which it goes back to the original menu.

Contribution of Group members:

- CHINMAYI:
  - displayData() function [class Customer_Details]
  - update_record() function [class Items]
  - selectItems() function [class Items]
  - create_account() function [class Admin]
  - display_record() function [class Admin]
  - display_bnh() function [class Admin]
  - display_bv() function [class Admin]
  - display_dr() function [class Admin]
  - display_fnv() function [class Admin]
  - display_sn() function [class Admin]
  - addItems() function [class Admin]
  - updateCategories() function [class Admin]
  - login() function [class Admin]

- HEER:
  - readData() function [class Customer_Details]
  - displayData() function [class Customer_Details]
  - selectItems() function [class Items]
  - create_account() function [class Admin]
  - login() function [class Admin]
  - viewCategories() function [class Admin]

- SARTHAK:
  - selectCategory() function [class Customer_Details]
  - update_record() function [class Items] – {partly}
  - selectItems() function [class Items]
  - create_account() function [class Admin]
  - calcDiscount() function [class Items]
  - updateCategories() function [class Admin]

Source Code:

```
#include <iostream>
#include <map>
#include <fstream>
#include <string>
using namespace std;

class Customer_Details
{
protected :

  string name, contact_no, address, pay_method, bank_name;
  float amount = 0;

public :

  string readData()
  {
    cout << "Enter customer name : ";
    getline (cin >> ws, name);
    cout << "\nEnter customer contact no. : ";
    cin >> contact_no;
    cout << "\nEnter customer address : ";
    getline (cin >> ws, address);
    while (1)
    {
      cout << "\nPayment method" << endl;
      cout << "Enter   'OP' for Online payment\n\t'COD' for Cash On Delivery"
<< endl;
      cout << "Enter payment method : ";
      cin >> pay_method;
      if (pay_method == "OP" || pay_method == "op")
      {
        pay_method = "Online Payment";
        cout << "Enter bank name : ";
        cin >> bank_name;
```

```
      break;
     }
     else if (pay_method == "COD" || pay_method == "cod")
     {
      pay_method = "Cash On Delivery";
      break;
     }
     else
     {
      cout << "Please enter the correct choice!";
     }
    }
    return pay_method;
   }

   int selectCategory ()
   {
    int category;
    cout << "1 -> Beauty & Hygiene\n2 -> Beverages\n3 -> Dairy\n4 -> Fruits &
Vegetables\n5 -> Snacks" << endl;
    cout << "Enter desired category : ";
    cin >> category;
    return category;
   }

   void displayData ()
   {
    cout << " CUSTOMER NAME : " << name << endl;
    cout << "|--------------------------------------|" << endl;
    cout << " CUSTOMER CONTACT NO. : " << contact_no << endl;
    cout << "|--------------------------------------|" << endl;
    cout << " CUSTOMER ADDRESS : " << address << endl;
    cout << "|--------------------------------------|" << endl;
    cout << " PAYMENT METHOD : " << pay_method << endl;
    cout << "|--------------------------------------|" << endl;
    if (pay_method == "Online Payment")
    {
```

```
    cout << " BANK NAME : " << bank_name << endl;
    cout << "|--------------------------------------|" << endl;
   }
   cout << "Item\tRate\tUnits\tCost" << endl;
   ifstream fin; //Sales.csv stores list of items selected by the customer
   string data;
   fin.open ("Sales.csv");
   while (fin.peek() != EOF)
   {
    getline (fin, data, ',');
    cout << data << "\t";
   }
   fin.close();
   int result = remove ("Sales.csv"); //Deletes the file Sales.csv after the bill has
been displayed
  }
};

class Items : public Customer_Details
{
private :

  int choice, units_needed;

public :

  void update_record (string filename, map <int, string> item_list, int choice, int
units_needed, int units_present[])
  {
   fstream fout;
   fout.open (filename, ios :: out); //Re-writes the files after the customer has
selected a certain number of units for a particular item
   for(int i = 0; i < 5; i++)
   {
    if (choice == i + 1)
    {
     if (units_present[i] == 0)
```

```
        {
          cout << "Out of stock" << endl;
        }
        else if (units_present [i] < units_needed)
        {
          cout << "Insufficient items" << endl;
        }
        if (units_present[i] == 0 || units_present[i] < units_needed)
        {
          for (int i = 0; i < 5; i++)
          {
            fout << i + 1 << ',' << item_list[i + 1] << ',' << units_present [i] << '\n';
          }
        }
        else
        {
          for (int i = 0; i < 5; i++)
          {
            if (choice == i + 1)
            {
              units_present [i] -= units_needed;
            }
            fout << i + 1 << ',' << item_list[i + 1] << ',' << units_present [i] << '\n';
          }
        }
      }
    }
  }

  float selectItems (int category, int bnh_units[], int bv_units[], int dr_units[], int
fnv_units[], int sn_units[], ofstream& f)
  {
    if (category == 1)
    {
        while (1)
        {
          map <int, float> bnh_cost;
```

```
        bnh_cost [1] = 296;
        bnh_cost [2] = 120;
        bnh_cost [3] = 150;
        bnh_cost [4] = 395;
        bnh_cost [5] = 314;
        map <int, string> bnh_item;
        bnh_item [1] = "Soap";
        bnh_item [2] = "Serum";
        bnh_item [3] = "HairOil";
        bnh_item [4] = "Perfume";
        bnh_item [5] = "Shampoo";
        cout << "BEAUTY & HYGIENE" << endl;
        cout << "1 -> Soap\t\tRs. 296" << endl;
        cout << "2 -> Serum\t\tRs. 120" << endl;
        cout << "3 -> HairOil\t\tRs. 150" << endl;
        cout << "4 -> Perfume\t\tRs. 395" << endl;
        cout << "5 -> Shampoo\t\tRs. 314" << endl;
        cout << "\nSelect item number : ";
        cin >> choice;
        cout << "Enter number of units : ";
        cin >> units_needed;
        update_record ("Beauty & Hygiene.csv", bnh_item, choice, units_needed,
bnh_units);
        if (bnh_units[choice - 1] > units_needed && bnh_units[choice - 1] != 0)
        {
          f.open ("Sales.csv", ios :: app); //Appends a new row containing item
name, rate, units purchased and cost in file Sales.csv
          f << bnh_item[choice] << ',' << bnh_cost[choice] << ',' << units_needed
<< ',' << bnh_cost[choice] * units_needed << endl;
          f.close();
          amount += units_needed * bnh_cost [choice];
        }
        char item_choice;
        cout << "Continue shopping from this category?";
        cin >> item_choice;
        if (item_choice == 'n')
        break;
```

```
        }
    }
    else if (category == 2)
    {
        while (1)
        {
          map <int, float> bv_cost;
          bv_cost [1] = 228;
          bv_cost [2] = 300;
          bv_cost [3] = 33;
          bv_cost [4] = 101;
          bv_cost [5] = 20;
          map <int, string> bv_item;
          bv_item [1] = "Tea";
          bv_item [2] = "Coffee";
          bv_item [3] = "Drink";
          bv_item [4] = "Juice";
          bv_item [5] = "Water";
          cout << "BEVERAGES" << endl;
          cout << "1 -> Tea\t\tRs. 228" << endl;
          cout << "2 -> Coffee\t\tRs. 300" << endl;
          cout << "3 -> Drink\t\tRs. 33" << endl;
          cout << "4 -> Juice\t\tRs. 101" << endl;
          cout << "5 -> Water\t\tRs. 20" << endl;
          cout << "\nSelect item number : ";
          cin >> choice;
          cout << "Enter number of units : ";
          cin >> units_needed;
          update_record ("Beverages.csv", bv_item, choice, units_needed,
bv_units);
          if (bv_units[choice - 1] > units_needed && bv_units[choice - 1] != 0)
          {
            f.open ("Sales.csv", ios :: app);
            f << bv_item[choice] << ',' << bv_cost[choice] << ',' << units_needed
<< ',' << bv_cost[choice] * units_needed << endl;
            f.close();
            amount += units_needed * bv_cost [choice];
```

```
      }
      char item_choice;
      cout << "Continue shopping from this category?";
      cin >> item_choice;
      if (item_choice == 'n')
      break;
    }
}
else if (category == 3)
{
    while (1)
    {
      map <int, float> dr_cost;
      dr_cost [1] = 133;
      dr_cost [2] = 120;
      dr_cost [3] = 90;
      dr_cost [4] = 98;
      dr_cost [5] = 2023;
      map <int, string> dr_item;
      dr_item [1] = "Cheese";
      dr_item [2] = "Yogurt";
      dr_item [3] = "Paneer";
      dr_item [4] = "Butter";
      dr_item [5] = "Ghee";
      cout << "DAIRY" << endl;
      cout << "1 -> Cheese\t\tRs. 133" << endl;
      cout << "2 -> Yogurt\t\tRs. 120" << endl;
      cout << "3 -> Paneer\t\tRs. 90" << endl;
      cout << "4 -> Butter\t\tRs. 98" << endl;
      cout << "5 -> Ghee\t\tRs. 2023" << endl;
      cout << "\nSelect item number :\t";
      cin >> choice;
      cout << "Enter number of units: ";
      cin >> units_needed;
      update_record ("Dairy.csv", dr_item, choice, units_needed, dr_units);
      if (dr_units[choice - 1] > units_needed && dr_units[choice - 1] != 0)
      {
```

```
            f.open ("Sales.csv", ios :: app);
            f << dr_item[choice] << ',' << dr_cost[choice] << ',' << units_needed <<
',' << dr_cost[choice] * units_needed << endl;
            f.close();
            amount += units_needed * dr_cost [choice];
          }
          char item_choice;
          cout << "Continue shopping from this category?";
          cin >> item_choice;
          if (item_choice == 'n')
          break;
        }
    }
    else if (category == 4)
    {
       while (1)
       {
          map <int, float> fnv_cost;
          fnv_cost [1] = 27;
          fnv_cost [2] = 38;
          fnv_cost [3] = 15;
          fnv_cost [4] = 96;
          fnv_cost [5] = 162;
          map <int, string> fnv_item;
          fnv_item [1] = "Onion";
          fnv_item [2] = "Potato";
          fnv_item [3] = "Tomato";
          fnv_item [4] = "Banana";
          fnv_item [5] = "Apple";
          cout << "FRUITS & VEGETABLES" << endl;
          cout << "1 -> Onion\t\tRs. 27" << endl;
          cout << "2 -> Potato\t\tRs. 38" << endl;
          cout << "3 -> Tomato\t\tRs. 15" << endl;
          cout << "4 -> Banana\t\tRs. 96" << endl;
          cout << "5 -> Apple\t\tRs. 162" << endl;
          cout << "\nSelect item number :\t";
          cin >> choice;
```

```
        cout << "Enter number of units : ";
        cin >> units_needed;
        update_record ("Fruits & Vegetables.csv", fnv_item, choice,
units_needed, fnv_units);
        if (fnv_units[choice - 1] > units_needed && fnv_units[choice - 1] != 0)
        {
          f.open ("Sales.csv", ios :: app);
          f << fnv_item[choice] << ',' << fnv_cost[choice] << ',' << units_needed
<< ',' << fnv_cost[choice] * units_needed << endl;
          f.close();
          amount += units_needed * fnv_cost [choice];
        }
        char item_choice;
        cout << "Continue shopping from this category?";
        cin >> item_choice;
        if (item_choice == 'n')
        break;
       }
    }
    else if (category == 5)
    {
     while (1)
     {
       map <int, float> sn_cost;
       sn_cost [1] = 82;
       sn_cost [2] = 249;
       sn_cost [3] = 200;
       sn_cost [4] = 20;
       sn_cost [5] = 10;
       map <int, string> sn_item;
       sn_item [1] = "Bread";
       sn_item [2] = "Cereal";
       sn_item [3] = "Candy";
       sn_item [4] = "Chips";
       sn_item [5] = "Maggie";
       cout << "SNACKS" << endl;
       cout << "1 -> Bread\t\tRs. 82" << endl;
```

```
        cout << "2 -> Cereal\t\tRs. 249" << endl;
        cout << "3 -> Candy\t\tRs. 200" << endl;
        cout << "4 -> Chips\t\tRs. 20" << endl;
        cout << "5 -> Maggi\t\tRs. 10" << endl;
        cout << "\nSelect item number :\t";
        cin >> choice;
        cout << "Enter number of units : ";
        cin >> units_needed;
        update_record ("Snacks.csv", sn_item, choice, units_needed, sn_units);
        if (bnh_units[choice - 1] > units_needed && sn_units[choice - 1] != 0)
        {
          f.open ("Sales.csv", ios :: app);
          f << sn_item[choice] << ',' << sn_cost[choice] << ',' << units_needed <<
',' << sn_cost[choice] * units_needed << endl;
          f.close();
          amount += units_needed * sn_cost [choice];
        }
        char item_choice;
        cout << "Continue shopping from this category?";
        cin >> item_choice;
        if (item_choice == 'n')
        break;
       }
     }
     else
     {
       cout << "You selected the wrong category number!" << endl;
     }
     return amount;
   }

   void calcDiscount ()
   {
     float discount;
     float topay;
     cout << endl;
     cout << " TOTAL : Rs." << amount << endl;
```

```
    cout << "|---------------------------------------|" << endl;
    if (amount <= 100)
    {
      cout << "  DISCOUNT RECEIVED = No discount" << endl;
      cout << "|---------------------------------------|" << endl;
      topay = amount;
    }
    else
   {
      if(amount > 100 && amount <= 200)
      {
        cout << " DISCOUNT RECEIVED = 5%" << endl;
        discount = (amount * 5) / 100;
      }
      else if(amount > 200 && amount <= 400)
      {
        cout << " DISCOUNT RECEIVED = 10%" << endl;
        discount = (amount * 10) / 100;
      }
      else if(amount > 400 && amount <= 800)
      {
        cout<< " DISCOUNT RECEIVED = 20%" << endl;
        discount = (amount * 20) / 100;
      }
      else
      {
        cout<< " DISCOUNT RECEIVED = 25%" << endl;
        discount = (amount * 25) / 100;
      }
      topay = amount - discount;
    }
      cout << "|---------------------------------------|" << endl;
      cout << "  Amount to be paid : Rs." << topay << endl;
      cout << endl;
   }
};
```

```
class Admin
{

private :

  string username, password;
  int admin_choice, admin_choice1, admin_choice2;

public :

  void create_account()
  {
    while (1)
    {
      cout << "USERNAME" << endl;
      cout << "Username should have minimum 6 and maximum 10 characters"
<< endl;
      cout << "Enter username : ";
      cin >> username;
      if (username.length() >= 6 && username.length() <= 10)
      {
        cout << "Username Accepted!" << endl;
        break;
      }
      else if (username.length() < 6)
      {
        cout << "Username entered has less than 6 characters" << endl;
        cout << "Enter username again" << endl;
      }
      else if (username.length() > 10)
      {
        cout << "Username entered has more than 10 characters" << endl;
        cout << "Enter username again" << endl;
      }
    }
    while (1)
    {
```

```
    cout << "PASSWORD" << endl;
    cout << "Password should have exactly 2 special characters, atmost 3
numeric characters and should have 8 or more characters" << endl;
    cout << "Enter password : ";
    cin >> password;
    int sc_count = 0, n_count = 0;
    for (int i = 0; i < password.length(); i++)
    {
     char c = password.at(i);
     if (c == '!' || c == '@' || c == '#' || c == '$' || c == '%' || c == '&' || c == '*')
     {
      sc_count++;
     }
     if (isdigit(c))
     {
      n_count++;
     }
    }
    if (sc_count == 2 && n_count <= 3 && password.length() >= 8)
    {
     cout << "Password accepted!" << endl;
     break;
    }
    else
    {
     if (sc_count != 2)
     {
      cout << "Password entered does not have exactly 2 special characters" <<
endl;
      cout << "Enter password again" << endl;
     }
     if (n_count > 3)
     {
      cout << "Password entered has more than 3 numeric characters" << endl;
      cout << "Enter password again" << endl;
     }
     if (password.length() < 8)
```

```
      {
        cout << "Password entered has less than 8 characters" << endl;
        cout << "Enter password again" << endl;
      }
    }
  }
}
void display_record (string filename)
{
  cout << "ItemNo.\tName\tUnits" << endl;
  ifstream fin;
  string data;
  fin.open (filename);
  while (fin.peek() != EOF)
  {
    getline (fin, data, ',');
    cout << data << "\t";
  }
  fin.close();
}

void display_bnh()
{
  cout << "\nBEAUTY & HYGIENE" << endl;
  display_record ("Beauty & Hygiene.csv");
}

void display_bv()
{
  cout << "\nBEVERAGES" << endl;
  display_record ("Beverages.csv");
}

void display_dr()
{
  cout << "\nDAIRY" << endl;
  display_record ("Dairy.csv");
```

```
  }

  void display_fnv()
  {
    cout << "\nFRUITS & VEGETABLES" << endl;
    display_record ("Fruits & Vegetables.csv");
  }

  void display_sn()
  {
    cout << "\nSNACKS" << endl;
    display_record ("Snacks.csv");
  }

  void addItems (string filename, map <int, string> item_list, int choice, int
units_added, int units_present[])
  {
    fstream fout;
    fout.open (filename, ios :: out); //Re-writes the files with the added units for a
particular item
    for (int i = 0; i < 5; i++)
    {
      if (choice == i + 1)
      {
        units_present [i] += units_added;
      }
      fout << i + 1 << ',' << item_list[i + 1] << ',' << units_present [i] << '\n';
    }
  }

  void viewCategories()
  {
    cout << "[1] View Beauty & Hygiene category\n[2] View Beverages
category\n[3] View Dairy category\n[4] View Fruits & Vegetables
category\n[5] View Snacks category\n[Any other number] View all categories"
<< endl;
    cout << "Enter choice : ";
```

```
cin >> admin_choice1;
switch (admin_choice1)
{
case 1 :
  {
    display_bnh();
    break;
  }
case 2 :
  {
    display_bv();
    break;
  }
case 3 :
  {
    display_dr();
    break;
  }
 case 4 :
  {
    display_fnv();
    break;
  }
  case 5 :
  {
    display_sn();
    break;
  }
  default :
  {
    display_bnh();
    display_bv();
    display_dr();
    display_fnv();
    display_sn();
    break;
  }
```

```
    }
  }

  void updateCategories(int bnh_units[], int bv_units[], int dr_units[], int
fnv_units[], int sn_units[])
  {
    cout << "[1] Update Beauty & Hygiene category\n[2] Update Beverages
category\n[3] Update Dairy category\n[4] Update Fruits & Vegetables
category\n[5] Update Snacks category" << endl;
    cout << "Enter choice : ";
    cin >> admin_choice1;
    int units_added;
    switch (admin_choice1)
    {
    case 1 :
      {
        map <int, string> bnh_item;
        bnh_item [1] = "Soap";
        bnh_item [2] = "Serum";
        bnh_item [3] = "HairOil";
        bnh_item [4] = "Perfume";
        bnh_item [5] = "Shampoo";
        display_bnh();
        cout << "\nEnter choice : ";
        cin >> admin_choice2;
        cout << "Enter no. of units to be added : ";
        cin >> units_added;
        addItems ("Beauty & Hygiene.csv", bnh_item, admin_choice2,
units_added, bnh_units);
        break;
      }
     case 2 :
       {
         map <int, string> bv_item;
         bv_item [1] = "Tea";
         bv_item [2] = "Coffee";
         bv_item [3] = "Drink";
```

```
          bv_item [4] = "Juice";
          bv_item [5] = "Water";
          display_bv();
          cout << "\nEnter choice : ";
          cin >> admin_choice2;
          cout << "Enter no. of units to be added : ";
          cin >> units_added;
          addItems ("Beverages.csv", bv_item, admin_choice2, units_added,
bv_units);
          break;
         }
        case 3 :
         {
          map <int, string> dr_item;
          dr_item [1] = "Cheese";
          dr_item [2] = "Yogurt";
          dr_item [3] = "Paneer";
          dr_item [4] = "Butter";
          dr_item [5] = "Ghee";
          display_dr();
          cout << "\nEnter choice : ";
          cin >> admin_choice2;
          cout << "Enter no. of units to be added : ";
          cin >> units_added;
          addItems ("Dairy.csv", dr_item, admin_choice2, units_added, dr_units);
          break;
         }
         case 4 :
         {
          map <int, string> fnv_item;
          fnv_item [1] = "Onion";
          fnv_item [2] = "Potato";
          fnv_item [3] = "Tomato";
          fnv_item [4] = "Banana";
          fnv_item [5] = "Apple";
          display_fnv();
          cout << "\nEnter choice : ";
```

```
          cin >> admin_choice2;
          cout << "Enter no. of units to be added : ";
          cin >> units_added;
          addItems ("Fruits & Vegetables.csv", fnv_item, admin_choice2,
units_added, fnv_units);
          break;
         }
         case 5 :
         {
          map <int, string> sn_item;
          sn_item [1] = "Bread";
          sn_item [2] = "Cereal";
          sn_item [3] = "Candy";
          sn_item [4] = "Chips";
          sn_item [5] = "Maggie";
          display_sn();
          cout << "\nEnter choice : ";
          cin >> admin_choice2;
          cout << "Enter no. of units to be added : ";
          cin >> units_added;
          addItems ("Snacks.csv", sn_item, admin_choice2, units_added,
sn_units);
          break;
         }
         }
      }

     void login(int bnh_units[], int bv_units[], int dr_units[], int fnv_units[], int
sn_units[])
     {
       string user, pass;
       cout << "LOGIN" << endl;
       cout << "Username : ";
       cin >> user;
       cout << "Password : ";
       cin >> pass;
       int i = 1;
```

```
while (i != 3)
{
 if (username == user && password == pass)
 {
  cout << "Successfully logged in!" << endl;
  cout << "[1] View Inventory\n[2] Update Units" << endl;
  cout << "Enter choice : ";
  cin >> admin_choice;
  switch (admin_choice)
  {
  case 1 :
   {
    viewCategories();
    break;
   }
  case 2 :
   {
    updateCategories(bnh_units, bv_units, dr_units, fnv_units, sn_units);
    break;
   }
  }
  break;
 }
 else
 {
  int choice;
  cout << "Either the username or the password is wrong" << endl;
  cout << "[1] Username\n[2] Password\n[3] Both" << endl;
  cout << "Which one do you want to re-enter?";
  cin >> choice;
  switch (choice)
  {
  case 1 :
   {
    cout << "Re-enter username : ";
    cin >> user;
    break;
```

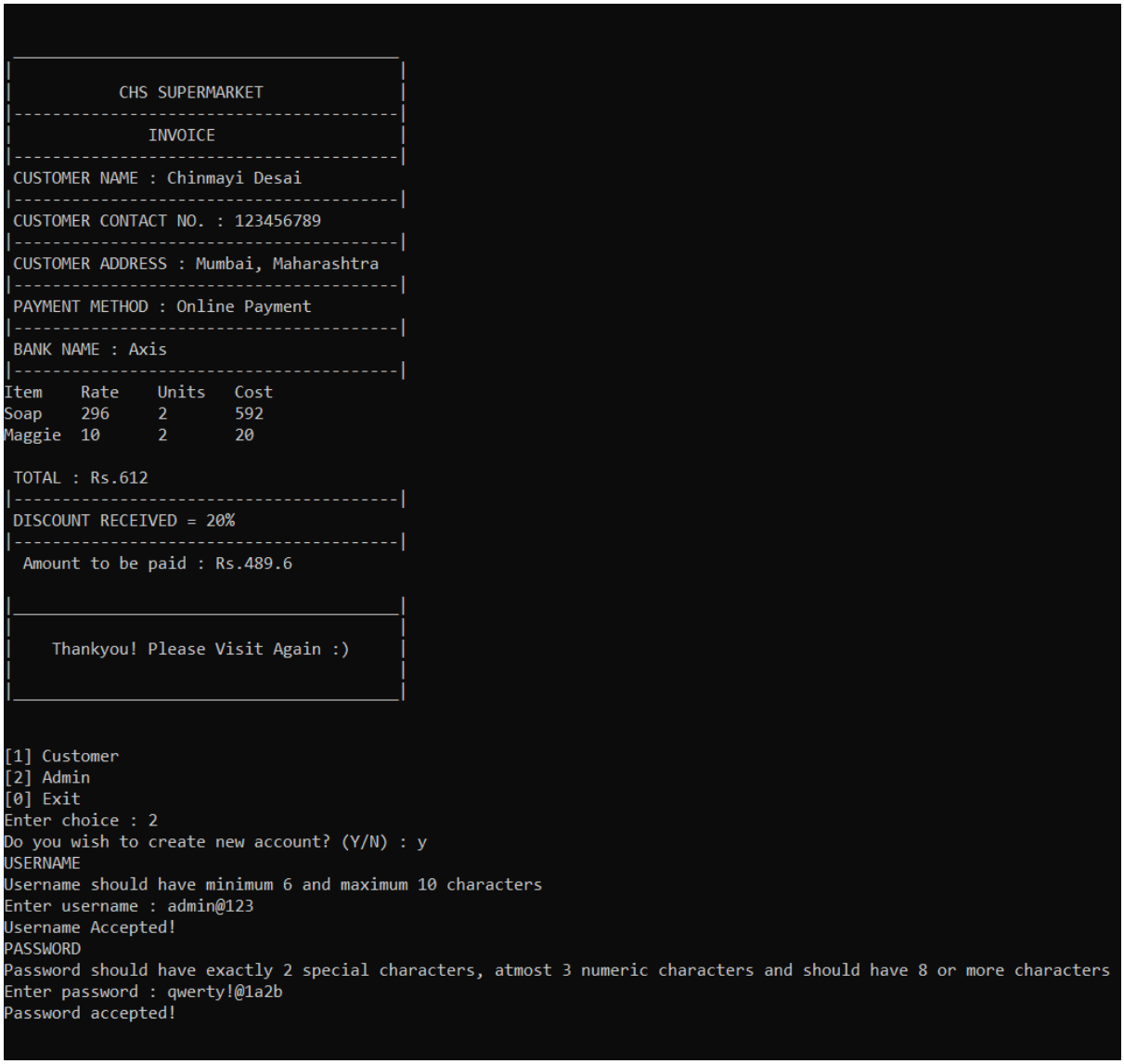
```
            }
          case 2 :
            {
              cout << "Re-enter password : ";
              cin >> pass;
              break;
            }
          case 3 :
            {
              cout << "Re-enter username : ";
              cin >> user;
              cout << "Re-enter password : ";
              cin >> pass;
            }
          }
        }
        i++;
      }
    }
};

int main ()
{
  int bnh_units[] = {20, 20, 20, 20, 20};
  int bv_units[] = {20, 20, 20, 20, 20};
  int dr_units[] = {20, 20, 20, 20, 20};
  int fnv_units[] = {20, 20, 20, 20, 20};
  int sn_units[] = {20, 20, 20, 20, 20};

  ofstream f;

  Admin a;
  while (1)
  {
    cout << endl;
    cout << "[1] Customer" << endl;
    cout << "[2] Admin" << endl;
```

```
    cout << "[0] Exit" << endl;
    int choice;
    cout << "Enter choice : ";
    cin >> choice;
    switch (choice)
    {
    case 1 :
      {
        Customer_Details cd;
        fstream newFile;
        newFile.open ("Sales.csv"); //creates a new file Sales.csv to store list of
items selected by the customer
        cd.readData();
        Items it;
        while (1)
        {
          int category = cd.selectCategory ();
          it.selectItems(category, bnh_units, bv_units, dr_units, fnv_units, sn_units,
f);
          char choice;
          cout << "Do you wish to add more items to your cart? (Y/N) :";
          cin >> choice;
          if (choice == 'N' || choice == 'n')
          {
          break;
          }
        }
        cout << endl;
        cout << endl;
        cout << " _____"  << endl;
        cout << "|                                      |" << endl;
        cout << "|          CHS SUPERMARKET             |" << endl;
        cout << "|--------------------------------------|" << endl;
        cout << "|             INVOICE                  |" << endl;
        cout << "|--------------------------------------|" << endl;
        cd.displayData ();
        it.calcDiscount();
```

```
        cout << "|_____|"  << endl;
        cout << "|                                    |" << endl;
        cout << "|    Thankyou! Please Visit Again :)     |" << endl;
        cout << "|                                    |" << endl;
        cout << "|_____|"  << endl;
        break;
      }
    case 2:
      {
        char choice;
        cout << "Do you wish to create new account? (Y/N) : ";
        cin >> choice;
        if (choice == 'y' || choice == 'Y')
        {
          a.create_account();
        }
        else
        {
          a.login(bnh_units, bv_units, dr_units, fnv_units, sn_units);
        }
        break;
      }
    case 0:
      {
        cout << "Exit application" << endl;
        exit(0);
      }
    default :
      cout << "Entered wrong choice" << endl;
   }
   }

  return 0;
}
```

Output:

```
[1] Customer
[2] Admin
[0] Exit
Enter choice : 1
Enter customer name : Chinmayi Desai

Enter customer contact no. : 123456789

Enter customer address : Mumbai, Maharashtra

Payment method
Enter   'OP' for Online payment
        'COD' for Cash On Delivery
Enter payment method : op
Enter bank name : Axis
1 -> Beauty & Hygiene
2 -> Beverages
3 -> Dairy
4 -> Fruits & Vegetables
5 -> Snacks
Enter desired category : 1
BEAUTY & HYGIENE
1 -> Soap               Rs. 296
2 -> Serum              Rs. 120
3 -> HairOil            Rs. 150
4 -> Perfume            Rs. 395
5 -> Shampoo            Rs. 314

Select item number : 1
Enter number of units : 2
Continue shopping from this category?n
Do you wish to add more items to your cart? (Y/N) :y
1 -> Beauty & Hygiene
2 -> Beverages
3 -> Dairy
4 -> Fruits & Vegetables
5 -> Snacks
Enter desired category : 5
SNACKS
1 -> Bread              Rs. 82
2 -> Cereal             Rs. 249
3 -> Candy              Rs. 200
4 -> Chips              Rs. 20
5 -> Maggi              Rs. 10

Select item number :    5
Enter number of units : 2
Continue shopping from this category?n
Do you wish to add more items to your cart? (Y/N) :n
```


CHS SUPERMARKET
INVOICE
CUSTOMER NAME : Chinmayi Desai
CUSTOMER CONTACT NO. : 123456789
CUSTOMER ADDRESS : Mumbai, Maharashtra
PAYMENT METHOD : Online Payment
BANK NAME : Axis
Item Rate Units Cost
Soap 296 2 592
Maggie 10 2 20
TOTAL : Rs.612
DISCOUNT RECEIVED = 20%
Amount to be paid : Rs.489.6
Thankyou! Please Visit Again :)
[1] Customer
[2] Admin
[0] Exit
Enter choice : 2
Do you wish to create new account? (Y/N) : y
USERNAME
Username should have minimum 6 and maximum 10 characters
Enter username : admin@123
Username Accepted!
PASSWORD
Password should have exactly 2 special characters, atmost 3 numeric characters and should have 8 or more characters
Enter password : qwerty!@1a2b
Password accepted!

```
[1] Customer
[2] Admin
[0] Exit
Enter choice : 2
Do you wish to create new account? (Y/N) : n
LOGIN
Username : admin@123
Password : qwerty!@1a2b
Successfully logged in!
[1] View Inventory
[2] Update Units
Enter choice : 1
[1] View Beauty & Hygiene category
[2] View Beverages category
[3] View Dairy category
[4] View Fruits & Vegetables category
[5] View Snacks category
[Any other number] View all categories
Enter choice : 5

SNACKS
ItemNo. Name    Units
1       Bread   20
2       Cereal  20
3       Candy   20
4       Chips   20
5       Maggie  18


[1] Customer
[2] Admin
[0] Exit
Enter choice : 2
Do you wish to create new account? (Y/N) : n
LOGIN
Username : admin@123
Password : qwerty!@1a2b
Successfully logged in!
[1] View Inventory
[2] Update Units
Enter choice : 1
[1] View Beauty & Hygiene category
[2] View Beverages category
[3] View Dairy category
[4] View Fruits & Vegetables category
[5] View Snacks category
[Any other number] View all categories
Enter choice : 6
```

```
BEAUTY & HYGIENE
ItemNo. Name    Units
1       Soap    18
2       Serum   20
3       HairOil 20
4       Perfume 20
5       Shampoo 20

BEVERAGES
ItemNo. Name    Units
1       Tea     20
2       Coffee  20
3       Drink   20
4       Juice   20
5       Water   20

DAIRY
ItemNo. Name    Units
1       Cheese  20
2       Yogurt  20
3       Paneer  20
4       Butter  20
5       Ghee    20

FRUITS & VEGETABLES
ItemNo. Name    Units
1       Onion   20
2       Potato  20
3       Tomato  20
4       Banana  20
5       Apple   20

SNACKS
ItemNo. Name    Units
1       Bread   20
2       Cereal  20
3       Candy   20
4       Chips   20
5       Maggie  18


[1] Customer
[2] Admin
[0] Exit
Enter choice : 2
Do you wish to create new account? (Y/N) : n
LOGIN
Username : admin@123
Password : qwerty!@1a2b
Successfully logged in!
[1] View Inventory
[2] Update Units
```

```
Enter choice : 2
[1] Update Beauty & Hygiene category
[2] Update Beverages category
[3] Update Dairy category
[4] Update Fruits & Vegetables category
[5] Update Snacks category
Enter choice : 2

BEVERAGES
ItemNo. Name    Units
1       Tea     20
2       Coffee  20
3       Drink   20
4       Juice   20
5       Water   20

Enter choice : 4
Enter no. of units to be added : 21


[1] Customer
[2] Admin
[0] Exit
Enter choice : 2
Do you wish to create new account? (Y/N) : n
LOGIN
Username : admin@123
Password : qwerty!@1a2b
Successfully logged in!
[1] View Inventory
[2] Update Units
Enter choice : 1
[1] View Beauty & Hygiene category
[2] View Beverages category
[3] View Dairy category
[4] View Fruits & Vegetables category
[5] View Snacks category
[Any other number] View all categories
Enter choice : 2

BEVERAGES
ItemNo. Name    Units
1       Tea     20
2       Coffee  20
3       Drink   20
4       Juice   41
5       Water   20
```

```
[1] Customer
[2] Admin
[0] Exit
Enter choice : 0
Exit application

Process returned 0 (0x0)   execution time : 144.954 s
Press any key to continue.
```