

Dairouty Mobiles' System

This application's aim is to create a management system for Dairouty Mobiles Shop. The user is first given the option to log in as an admin or as a seller (cases of incorrect login details are handled).

The user is also given the option to change their password but only after entering their email and username in order to receive an OTP to verify the user's identity.

If the user is a seller they get to enter all the customer's details as well as their purchases with options to save and print the customer's invoice.

The admin is given the option to view the billing system as well as add to it, remove from it or print it. In addition, they can view the shop's inventory to check for any details or to add new products or update any quantities. As an admin, the user also has a right to view all customers and their orders and to edit or remove any details.

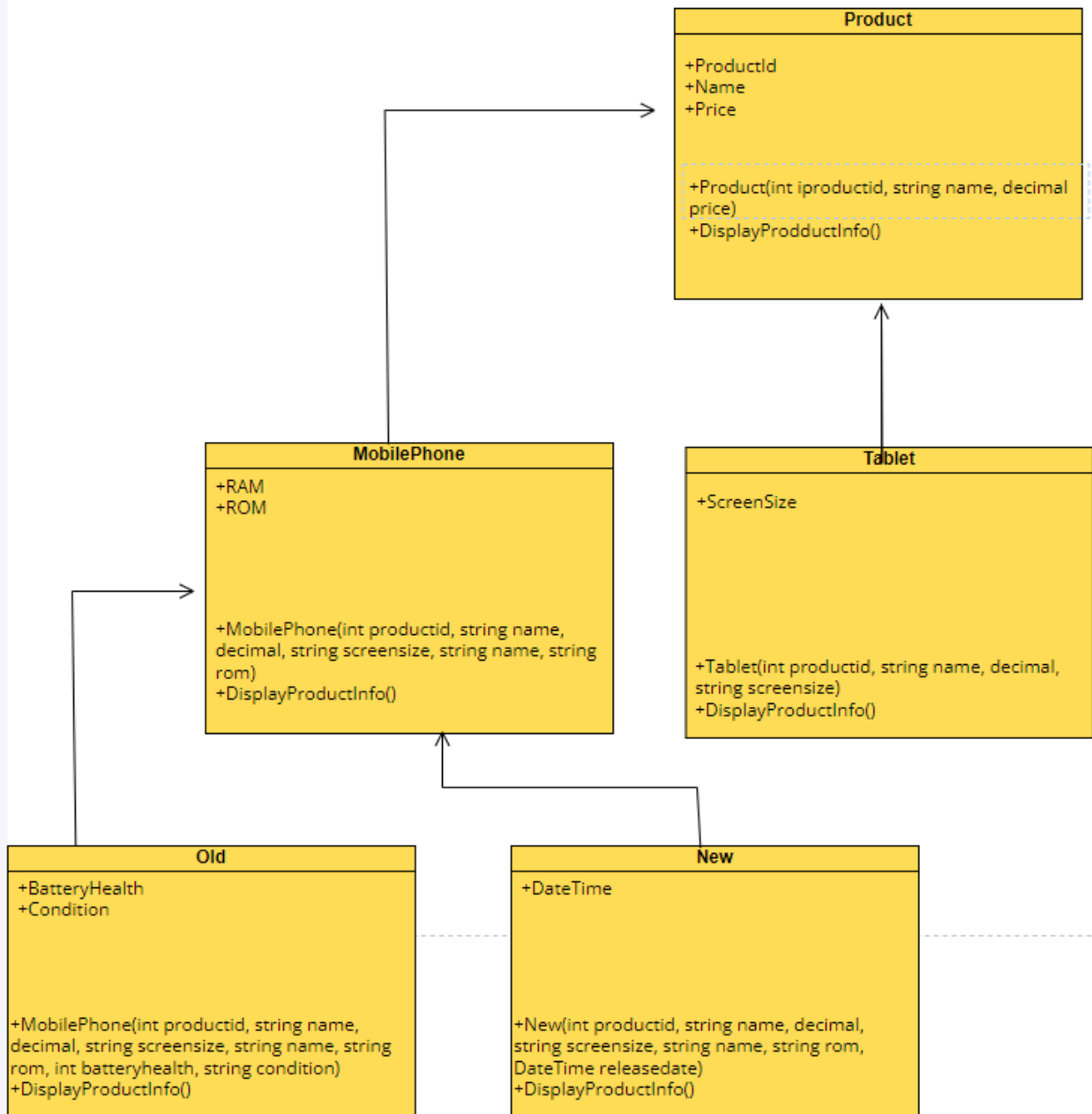
As is it is a GUI application, there are multiple graphic buttons used to make it user-friendly, for example: the buttons present on all forms, allowing the user to access the shop's social media accounts as well as its location.

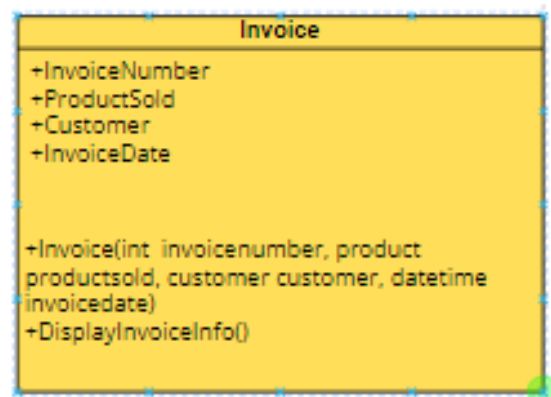
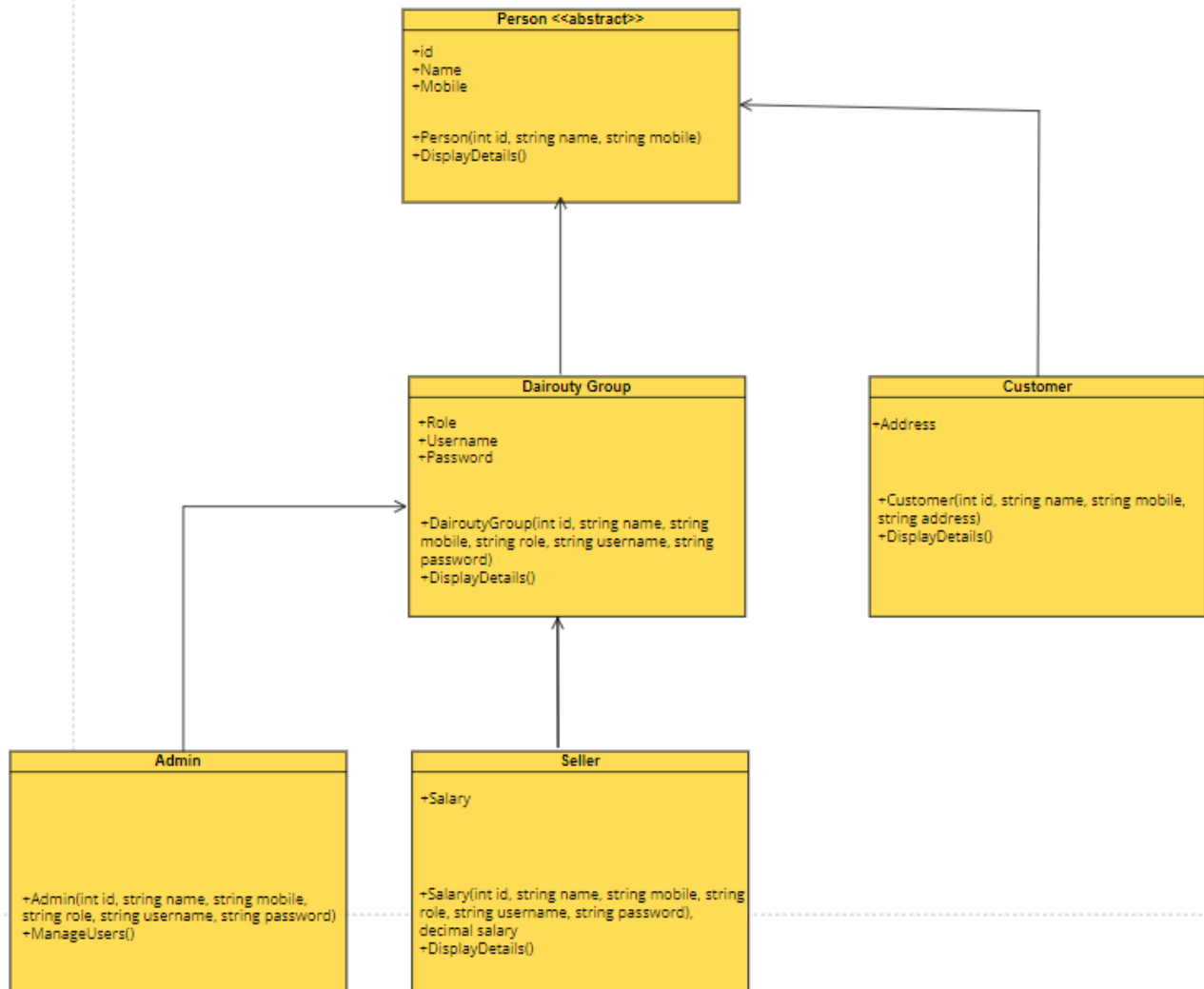
Counters are also present for stock, customers, profit and fees.

System's Requirements Analysis:

The system receives an ID(int), Name(string), Mobile(string), Role(string), Username(string), Password(string), Salary(decimal), ProductID(int), Product Name(int), Price(decimal), ScreenSize(string), RAM(string), ROM(string), Address(string) & Quantity

- Logging in requires Role, Username and Password (in some cases email and OTP as well).
- Adding or removing products as a seller or admin requires the customer's personal details as well as the products Name, Price, RAM, ROM and Screensize.
- The counters present on the form "Home" also requires the products' information as well as the customer's and the seller's.
- All navigation buttons require details of the forms themselves.
- The system is also required to update any changes on the database.





```

namespace Dairouty_Mobiles
{
    0 references
    internal static class Classes
    {
        [STAThread]
        0 references
        static void Main()
        {
            ApplicationConfiguration.Initialize();
            Application.Run(new Login());
        }
    }

    5 references
    public abstract class Person
    {
        1 reference
        public int Id { get; set; }
        2 references
        public string Name { get; set; }
        2 references
        public string Mobile { get; set; }

        2 references
        public Person(int id, string name, string mobile)
        {
            Id = id;
            Name = name;
            Mobile = mobile;
        }

        7 references
        public virtual void DisplayDetails()
        {
            Console.WriteLine($"Name: {Name}, Mobile: {Mobile}");
        }
    }
}

```

```

7 references
public abstract class Product
{
    1 reference
    public int ProductId { get; set; }
    2 references
    public string Name { get; set; }
    2 references
    public decimal Price { get; set; }

    2 references
    public Product(int productId, string name, decimal price)
    {
        ProductId = productId;
        Name = name;
        Price = price;
    }

    9 references
    public virtual void DisplayProductInfo()
    {
        Console.WriteLine($"Product Name: {Name}, Price: {Price}");
    }
}

```

```

public class Customer : Person
{
    2 references
    public string Address { get; set; }
    0 references
    public Customer(int id, string name, string mobile, string address) : base(id, name, mobile)
    {
        Address = address;
    }

    4 references
    public override void DisplayDetails()
    {
        base.DisplayDetails();
        Console.WriteLine($"Address: {Address}");
    }
}

```

```

public class DairoutyGroup : Person
{
    2 references
    public string Role { get; set; }
    2 references
    public string Username { get; set; }
    2 references
    public string Password { get; set; }

    2 references
    public DairoutyGroup(int id, string name, string mobile, string role, string username, string password) : base(id, name, mobile)
    {
        Role = role;
        Username = username;
        Password = password;
    }

    5 references
    public override void DisplayDetails()
    {
        base.DisplayDetails();
        Console.WriteLine($"Role: {Role}, Username: {Username}, Password: {Password}");
    }
}

```

```

1 reference
public class Seller : DairoutyGroup
{
    2 references
    public decimal Salary { get; set; }
    0 references
    public Seller(int id, string name, string mobile, string role, string username, string password, decimal salary) : base(id, name, mobile, role, username, password)
    {
        Salary = salary;
    }
    5 references
    public override void DisplayDetails()
    {
        base.DisplayDetails();
        Console.WriteLine($"Salary: {Salary}");
    }
}

1 reference
public class Admin : DairoutyGroup
{
    0 references
    public Admin(int id, string name, string mobile, string role, string username, string password) : base(id, name, mobile, role, username, password)
    {
    }
    0 references
    public void ManageUsers()
    {
        Admin_Sellers admin_Sellers = new Admin_Sellers();
        admin_Sellers.Show();
    }
}

```

```

1 reference
public class Tablet : Product
{
    2 references
    public string ScreenSize { get; set; }
    0 references
    public Tablet(int productId, string name, decimal price, string screenSize): base(productId, name, price)
    {
        ScreenSize = screenSize;
    }
    4 references
    public override void DisplayProductInfo()
    {
        base.DisplayProductInfo();
        Console.WriteLine($"Screen Size: {ScreenSize}");
    }
}

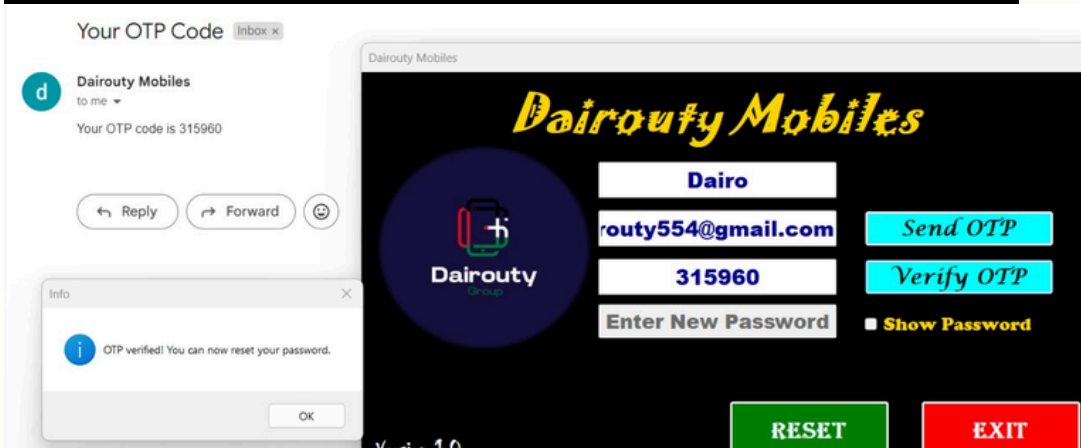
5 references
public class MobilePhone : Product
{
    2 references
    public string RAM { get; set; }
    2 references
    public string ROM { get; set; }
    2 references
    public MobilePhone(int productId, string name, decimal price, string ram, string rom): base(productId, name, price)
    {
        RAM = ram;
        ROM = rom;
    }
    8 references
    public override void DisplayProductInfo()
    {
        base.DisplayProductInfo();
        Console.WriteLine($"RAM: {RAM}, ROM: {ROM}");
    }
}

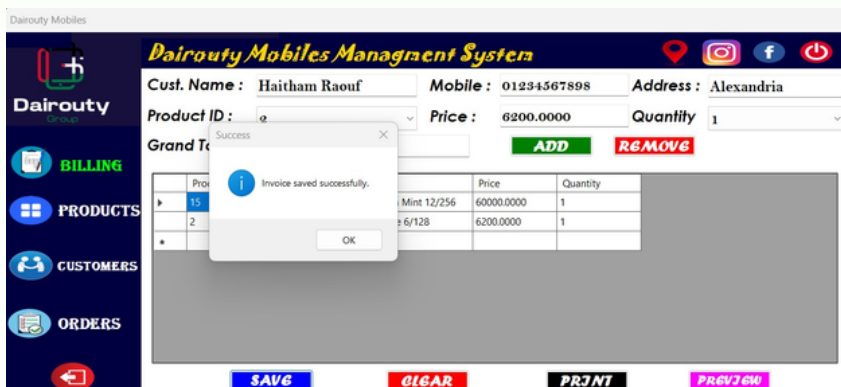
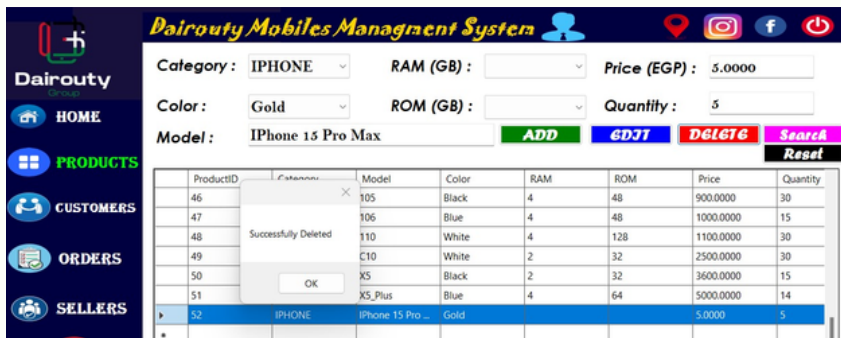
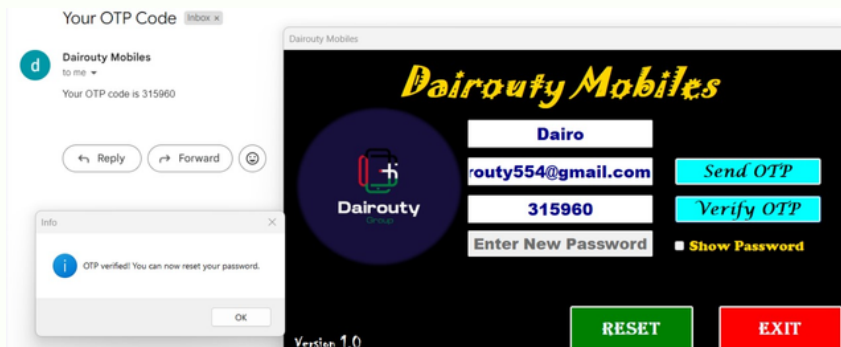
```

```

1 reference
public class Invoice
{
    2 references
    public int InvoiceNumber { get; set; }
    2 references
    public Product ProductSold { get; set; }
    2 references
    public Customer Customer { get; set; }
    2 references
    public DateTime InvoiceDate { get; set; }
    0 references
    public Invoice(int invoiceNumber, Product productSold, Customer customer, DateTime invoiceDate)
    {
        InvoiceNumber = invoiceNumber;
        ProductSold = productSold;
        Customer = customer;
        InvoiceDate = invoiceDate;
    }
    0 references
    public void DisplayInvoiceInfo()
    {
        Console.WriteLine($"Invoice Number: {InvoiceNumber}, Sale Date: {InvoiceDate}");
        ProductSold.DisplayProductInfo();
        Customer.DisplayDetails();
    }
}

```





HOME

PRODUCTS

CUSTOMERS

ORDERS

SELLERS

Dairouty Mobiles Management System

Total Stock Count :
896

Total Customers Count :
6

Today's Orders Count and Money Gained :
1 (66200.0000 EGP)

Total Sellers Count and Salary Fees :
1 (12500.0000 EGP)

أكبر بكتير مع الديروطي
خصم 10%

REFRESH

BILLING

PRODUCTS

CUSTOMERS

ORDERS

Dairouty Mobiles Management System

Invoice No. : 7Customer Name : Haitham Raouf

Invoice Date : 28/05/2024 00:00:00Grand Total : 66200.0000

EDITDELETESearchReset

Invoice_Number	Invoice_Date	Customer_Name	Grand_Total
1	26/05/2024	Tamer Farid	15000.0000
2	26/05/2024	Ahmed Fouad	17500.0000
3	24/05/2024	Zeina Mohamed	40000.0000
4	24/05/2024	Tamer Farid	1500.0000
5	27/05/2024	Ali	11400.0000
6	27/05/2024	Ayman	16550.0000
7	28/05/2024	Haitham Raouf	66200.0000

Dairouty Mobiles

Invoice

CUSTOMER NAME : Haitham Raouf

CUSTOMER MOBILE : 01234567898

CUSTOMER ADDRESS : Alexandria

PRODUCTS >

SAMSUNG Galaxy-S24-Ultra Mint 12/25660000.0000 EGP x1

SAMSUNG Galaxy-A04s Blue 6/1286200.0000 EGP x1

GRAND TOTAL : 66200.0000 EGP

SELLER SIGNATURE :


```

}
string connectionString = "Data Source=DAIRO\\SQLEXPRESS;Initial Catalog=Dairouty Mobiles\\;Integrated Security=True;Encri
private DataTable invoiceTable;
public Seller_Billing()
{
    InitializeComponent();
    invoiceTable = new DataTable();
    invoiceTable.Columns.Add("ProductID", typeof(string));
    invoiceTable.Columns.Add("Product", typeof(string));
    invoiceTable.Columns.Add("Price", typeof(decimal));
    invoiceTable.Columns.Add("Quantity", typeof(int));
    dataGridView1.DataSource = invoiceTable;
    dataGridView1.Columns["Product"].Width = 300;
    ProductIdsList();
    Prods.SelectedIndexChanged += QuantityList;
}
private void ProductIdsList()
{
    Prods.Items.Clear();

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        SqlCommand command = new SqlCommand("SELECT ProductID FROM [Dairouty Products] WHERE Quantity > 0", connection);
        SqlDataReader reader = command.ExecuteReader();

        while (reader.Read())
        {

```

```

using (SqlConnection connection = new SqlConnection(connectionString))
{
    connection.Open();
    SqlCommand command = new SqlCommand("INSERT INTO [Dairouty Customers] (Invoice_Number, Invoice_Date, Customer_Name, Customer_ID,
command.Parameters.AddWithValue("@InvoiceDate", DateTime.Today);
command.Parameters.AddWithValue("@InvoiceNumber", invnumber);
command.Parameters.AddWithValue("@CustomerID", id);
command.Parameters.AddWithValue("@CustomerName", customerName);
command.Parameters.AddWithValue("@CustomerMobile", customerMobile);
command.Parameters.AddWithValue("@CustomerAddress", customerAddress);
command.Parameters.AddWithValue("@GrandTotal", grandTotal);

    foreach (DataRow row in invoiceTable.Rows)
    {
        string productID = row["ProductID"].ToString();
        int quantity = Convert.ToInt32(row["Quantity"]);
        for (int i = 0; i < quantity; i++)
        {
            productIDsBuilder.Append(productID);
            productIDsBuilder.Append(" ");
        }
        SqlCommand updateCommand = new SqlCommand("UPDATE [Dairouty Products] SET Quantity = Quantity - @Quantity WHERE ProductID = ",
updateCommand.Parameters.AddWithValue("@Quantity", quantity);
updateCommand.Parameters.AddWithValue("@ProductID", productID);
updateCommand.ExecuteNonQuery();
    }
    string productIDs = productIDsBuilder.ToString().TrimEnd(' ');
    command.Parameters.AddWithValue("@Prods", productIDs);
}

```

```

}
private void PrintPage(object sender, PrintPageEventArgs e)
{
    Graphics g = e.Graphics;
    float pageWidth = e.PageBounds.Width;
    float pageHeight = e.PageBounds.Height;
    Image logo = Image.FromFile("");
    //Image logo = Image.FromFile(@"D:\Prog\C#\Project\Dairouty Mobiles\Photos\Pictur1.jpg");
    float logoWidth = pageWidth / 4;
    float logoHeight = (logoWidth / logo.Width) * logo.Height;
    float logoX = 10;
    float logoY = 10;
    g.DrawImage(logo, logoX, logoY, logoWidth, logoHeight);
    SizeF dairoutySize = g.MeasureString("Dairouty Mobiles", new Font("Jokerman", 24));
    float centerX_Dairouty = (pageWidth - dairoutySize.Width) / 2;
    SizeF invoiceSize = g.MeasureString("Invoice", new Font("Forte", 20, FontStyle.Italic));
    float centerX_Invoice = (pageWidth - invoiceSize.Width) / 2;
    float headerY = Math.Max(logoY + logoHeight + 10, dairoutySize.Height + 20);
    float invoiceY = headerY + dairoutySize.Height + 10;
    g.DrawString("Dairouty Mobiles", new Font("Jokerman", 24, FontStyle.Bold), new SolidBrush(Color.Navy), centerX_Dairouty, headerY);
    g.DrawString("Invoice", new Font("Forte", 24, FontStyle.Italic), new SolidBrush(Color.Navy), centerX_Invoice, invoiceY);
    float lineY = Math.Max(headerY + dairoutySize.Height + 10, invoiceY + invoiceSize.Height + 10) + 10;
    g.DrawLine(Pens.Black, 1, lineY, pageWidth, lineY);
}

```