Pengajian Sains Pengkomputeran,
Kolej Pengajian Pengkomputeran, Informatik dan Matematik,
Universiti Teknologi MARA (UiTM) Cawangan Pahang,
Kampus Raub

CSC301 – VISUAL PROGRAMMING

GROUP PROJECT

Project Title:

DATO’S BAKERY ORDERING SYSTEM

Group Members		
Student ID	**Name**	**Tel**
2022478924	MUHAMMAD AIDIEL BIN MOHAMAD HUSSIN	016-291 3835
2022605596	MUHAMMAD NAZHAN BIN ROZAINI	010-296 6259
2022877512	NURIN IMAN BINTI MASNGOT	017-366 9682
2022843782	NUR SABRINA BINTI ABD MANAP	011-5197 3635

Submission Date: 15th JULY 2024

Table of Contents

1. **Introduction**
   - 1.1 Background Company 1
   - 1.2 Problem Statement 2
   - 1.3 Objectives 2
2. **Flowchart**
   - 2.1 Admin 3
   - 2.1 Customer 4
3. **Interfaces (GUI)**
   - 3.1 Admin 5 - 8
   - 3.2 Customer 8 - 10
4. **Source Code**
   - 4.1 Login 11 - 12
   - 4.2 Register 12 – 14
   - 4.3 Product 14 - 16
   - 4.4 Cart 16 - 20
   - 4.5 Receipt 20 - 21
   - 4.6 Account 21 - 24
   - 4.7 Order Log 25 - 30
   - 4.8 Product Admin 30 - 35
   - 4.9 Users 35 - 43
5. **Database Table**
   - 5.1 Table Customer 44
   - 5.2 Table Order 44
   - 5.3 Table Product 44
6. **Conclusion** 45
7. **References** 46

1. Introduction

1.1 Background Company

The Dato's Bakery Shop is concerned about both the challenges faced by our employees and the needs of our customers. The entrepreneur has proactively enhanced the business in several areas with the goal of continuously producing high-quality products to promote healthy competition. The proprietor and the staff at Dato's Bakery Shop have proposed developing an online buy service as a means of offering an online purchasing system that will facilitate employee work and help the firm grow. Customers won't need to physically visit the store because they will be able to place orders using this method from wherever they are right now.

The goal of Dato's Bakery Shop is to increase its reach, and this method supports that goal. As a result, a more effective method for handling customers, their orders, and records in the store needs to be established. Nevertheless, a lot of inefficiencies, mistakes, and delays have happened when filling orders because the bakery still processes orders manually. The staff has been burdened by these concerns.

A system is also required to guarantee Dato's Bakery Shop functions effectively and efficiently because it has several branches. To effectively fulfil these duties, maintain seamless operations, and provide customers with outstanding customer service, each branch requires an effective group of employees. Dato's Bakery understands the value of utilizing digital technologies to remain competitive in today's market, in line with the fundamentals of the Fourth Industrial Revolution. The bakery can utilize technology to improve operations, improve customer experiences, and future-proof its business model due to the thoughtful implementation of an online ordering system.

1.2 Problem Statement

Although Dato's Bakery Shop's existing system functions well and gets along well with staff members, there are a few issues that we have found that could be fixed. The unwillingness of staff members to obtain data is the system's main problem. The orders and the information that each employee receives become inconsistent as a result. It means that orders must be given to certain employees before they can be communicated to the kitchen workers. As a result, it limits communication within the bakery and allows customer service issues that will undoubtedly have negative consequences down in the future.

In order to analyse sales, the staff would also need to manually calculate the orders they received and record the data into their preferred system. This would increase the chances of errors and oversights and make sales reporting laborious. This may cause incomplete or lost records to occur, which could cause orders to be filled incorrectly or slowly.

Since the current system is established and recognizable to our customers, staff members have additional responsibilities because of its reliance on administrators, who must allocate more time to serving customers. hence reducing the staff's potential for productivity.

1.3 Objectives

The objective below highlight the primary goals of implementing this system:

- To provide a more efficient system for customers to place their orders and receive assistance throughout the process.
- To facilitate the recording and relay of order information to bakery staff, optimizing the fulfilment process.
- To empower bakery staff with the ability to review order records, allowing for evaluation of successes and shortcomings in store operations.

2. Flowchart

2.1 Admin

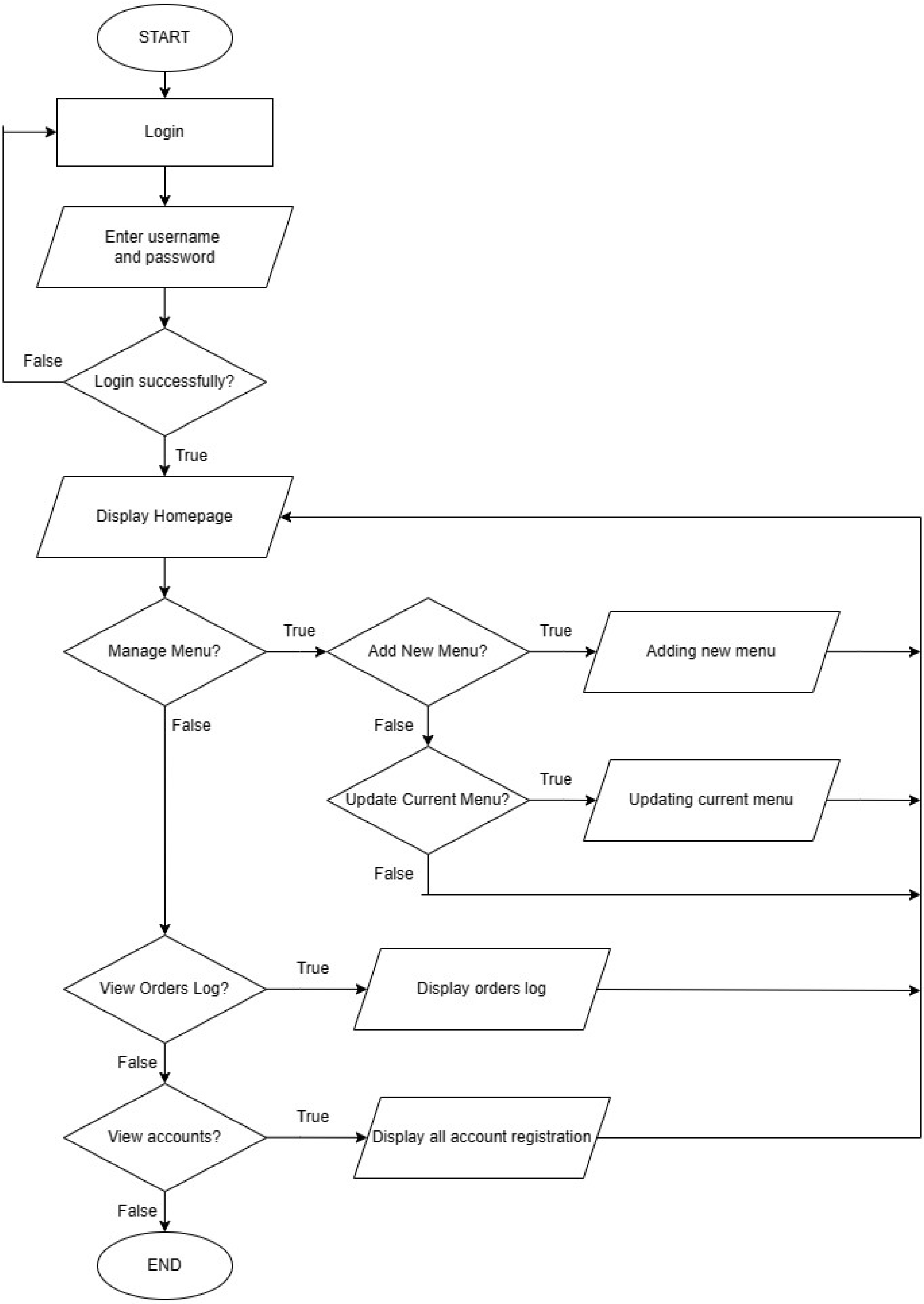


Figure 2.1: Flowchart of Admin processes

. 2.2 Customer

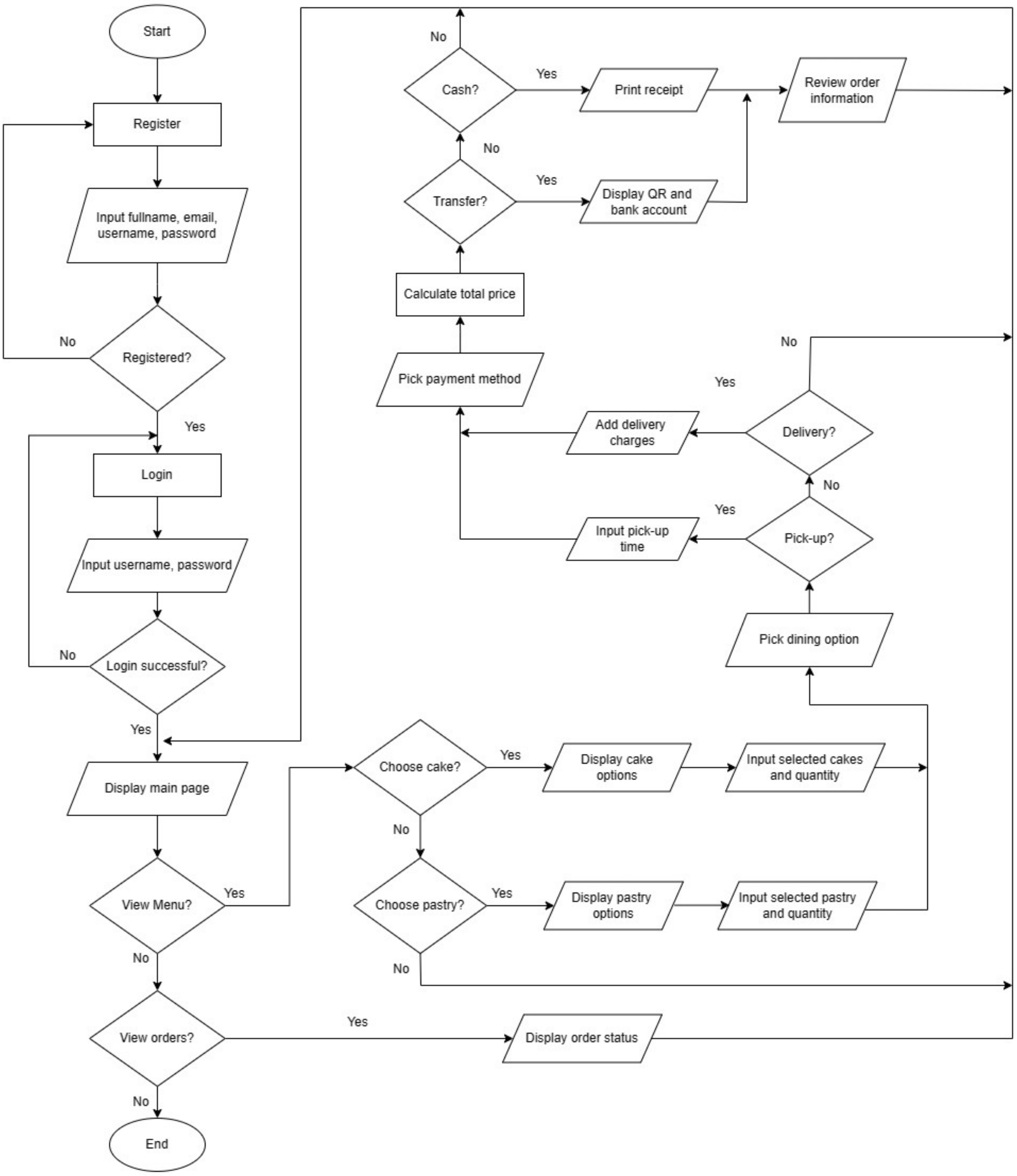


Figure 2.2: Flowchart of Customer processes

3. Proposed System User Interface Design

3.1 Admin

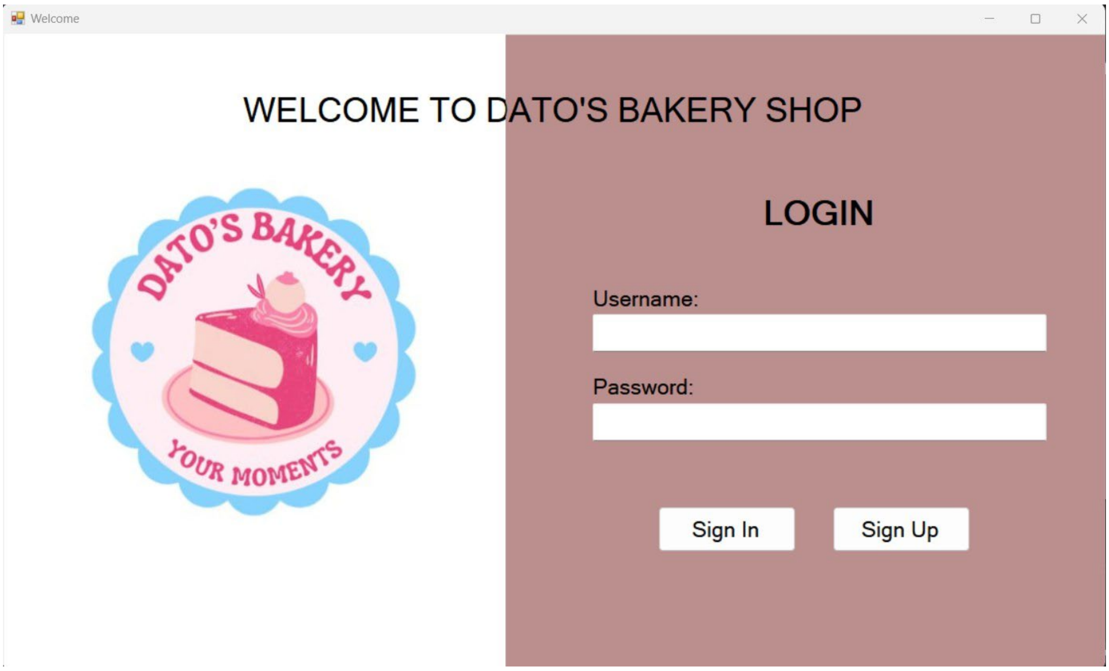


Figure 3.1.1: Login

Figure 3.1.1 shows the login form where users need to enter their unique username and password to access the system. If a user does not have an account yet, they can create a new account by clicking the Sign Up button, which will bring them to the registration form. If the username and password match those in the database, the user can access the system and place an order. For admins, the username and password are already set in the system, and they cannot create a new account.

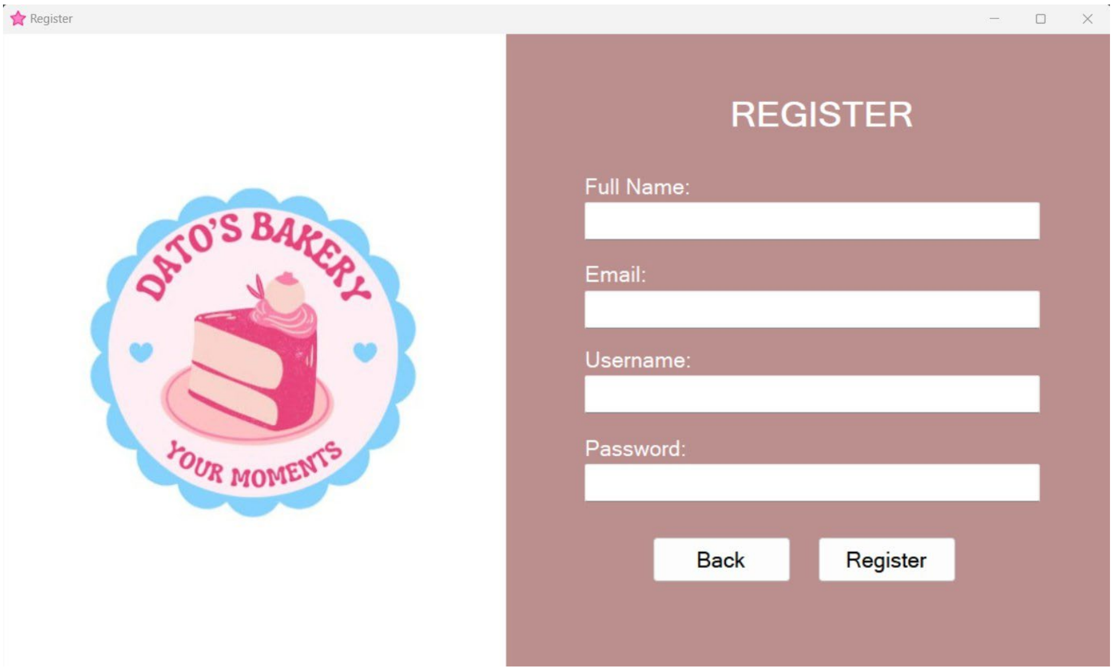
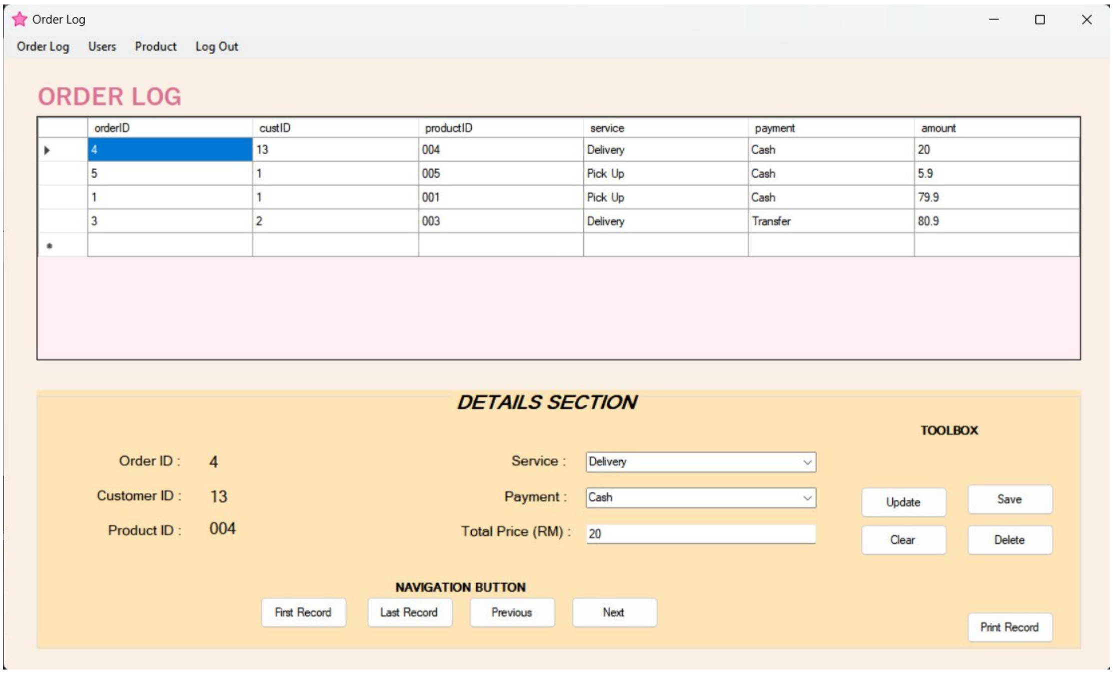


Figure 3.1.2: Register

Figure 3.1.2 shows the registration form where new users can create an account. Users need to fill in the required data to register. After inserting all the data, the user's account will be created, and they will need to go back to the login page by clicking the back button to start a login session.



ORDER LOG

orderID	custID	productID	service	payment	amount
4	13	004	Delivery	Cash	20
5	1	005	Pick Up	Cash	5.9
1	1	001	Pick Up	Cash	79.9
3	2	003	Delivery	Transfer	80.9

DETAILS SECTION

Order ID : 4

Customer ID : 13

Product ID : 004

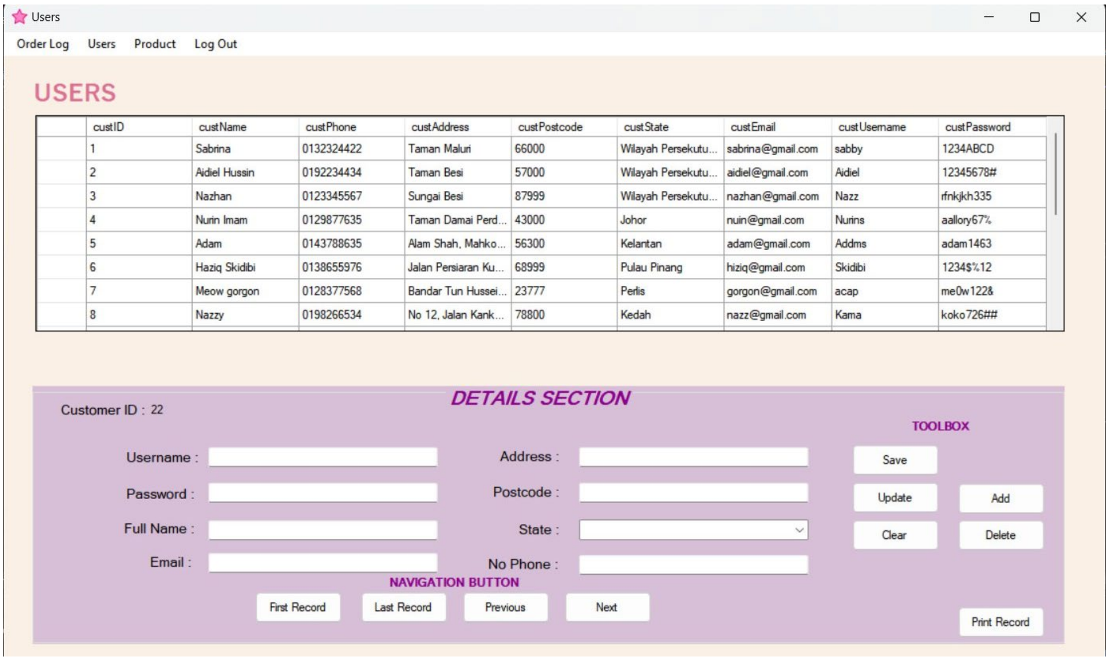
Service : Delivery

Payment : Cash

Total Price (RM) : 20

Figure 3.1.3: Order Log Admin

Figure 3.1.3 shows the order log form where the admin can view the order log retrieved from the database. There are several tools that the admin can use such as save for saving the current data in the details section, update for saving the updated data to the database, clear for clearing the text box form in the details section, delete for deleting the selected records, and print for printing the order log database. The admin can also view the first, last, previous, and next records by clicking the navigation buttons. The admin can also interact with the menu at the top of the form to navigate to another form or log out.



	custID	custName	custPhone	custAddress	custPostcode	custState	custEmail	custUsemame	custPassword
	1	Sabrina	0132324422	Taman Maluri	66000	Wilayah Persekutu...	sabrina@gmail.com	sabby	1234ABCD
	2	Aidiel Hussin	0192234434	Taman Besi	57000	Wilayah Persekutu...	aidiel@gmail.com	Aidiel	12345678#
	3	Nazhan	0123345567	Sungai Besi	87999	Wilayah Persekutu...	nazhan@gmail.com	Nazz	rfnkjkh335
	4	Nurin Imam	0129877635	Taman Damai Perd...	43000	Johor	nuin@gmail.com	Nurins	aallory67%
	5	Adam	0143788635	Alam Shah, Mahko...	56300	Kelantan	adam@gmail.com	Addms	adam1463
	6	Haziq Skidibi	0138655976	Jalan Persiaran Ku...	68999	Pulau Pinang	hiziq@gmail.com	Skidibi	1234$%12
	7	Meow gorgon	0128377568	Bandar Tun Hussei...	23777	Perlis	gorgon@gmail.com	acap	me0w122&
	8	Nazzy	0198266534	No 12, Jalan Kank...	78800	Kedah	nazz@gmail.com	Kama	koko726##

Customer ID : 22

Username :

Password :

Full Name :

Email :

Address :

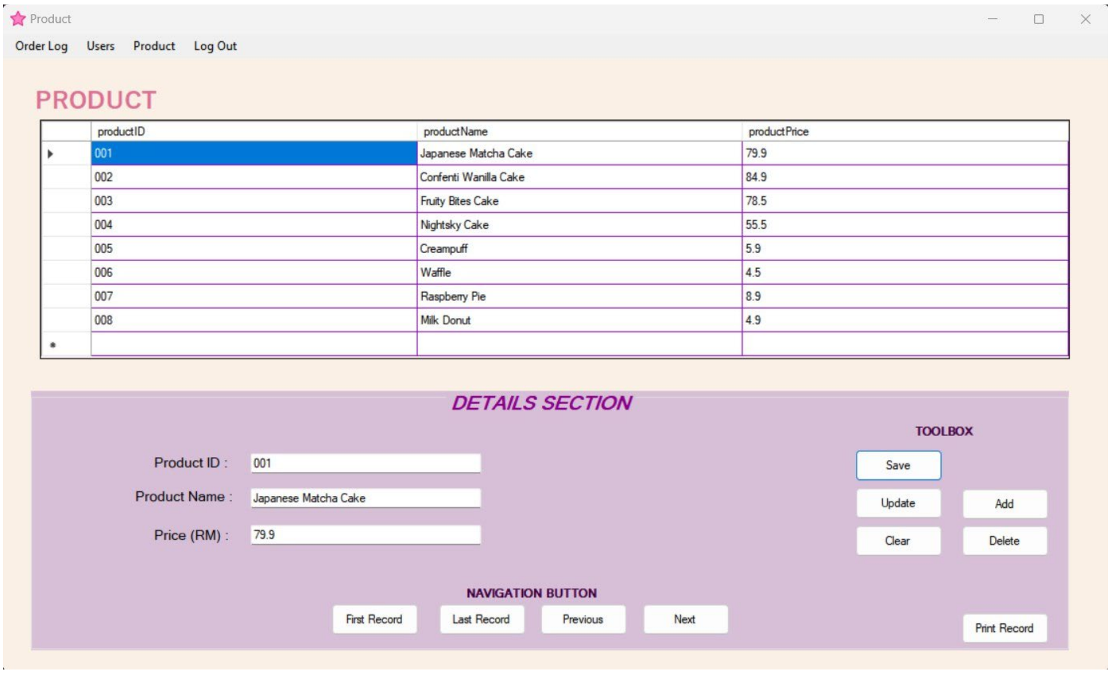
Postcode :

State :

No Phone :

Figure 3.1.4: Users Admin

Figure 3.4 shows the users form where the admin can view the customers that have been registered to Dato's Bakery Ordering System by retrieved from the database. There are several tools that the admin can use such as save for saving the current data in the details section, update for saving the updated data to the database, clear for clearing the text box form in the details section, add for adding a new record, delete for deleting the selected records, and print for printing the order log database. The admin can also view the first, last, previous, and next records by clicking the navigation buttons. The admin can also interact with the menu at the top of the form to navigate to another form or log out.



	productID	productName	productPrice
▸	001	Japanese Matcha Cake	79.9
	002	Confenti Wanilla Cake	84.9
	003	Fruity Bites Cake	78.5
	004	Nightsky Cake	55.5
	005	Creampuff	5.9
	006	Waffle	4.5
	007	Raspberry Pie	8.9
	008	Milk Donut	4.9
*			

Product ID : 001

Product Name : Japanese Matcha Cake

Price (RM) : 79.9

Figure 3.1.5: Product Admin

Figure 3.5 shows the product form where the admin can view the product that have been registered to Dato’s Bakery Ordering System by retrieved from the database. There are several tools that the admin can use such as save for saving the current data in the details section, update for saving the updated data to the database, clear for clearing the text box form in the details section, add for adding a new record, delete for deleting the selected records, and print for printing the order log database. The admin can also view the first, last, previous, and next records by clicking the navigation buttons. The admin can also interact with the menu at the top of the form to navigate to another form or log out.

3.2 Customer

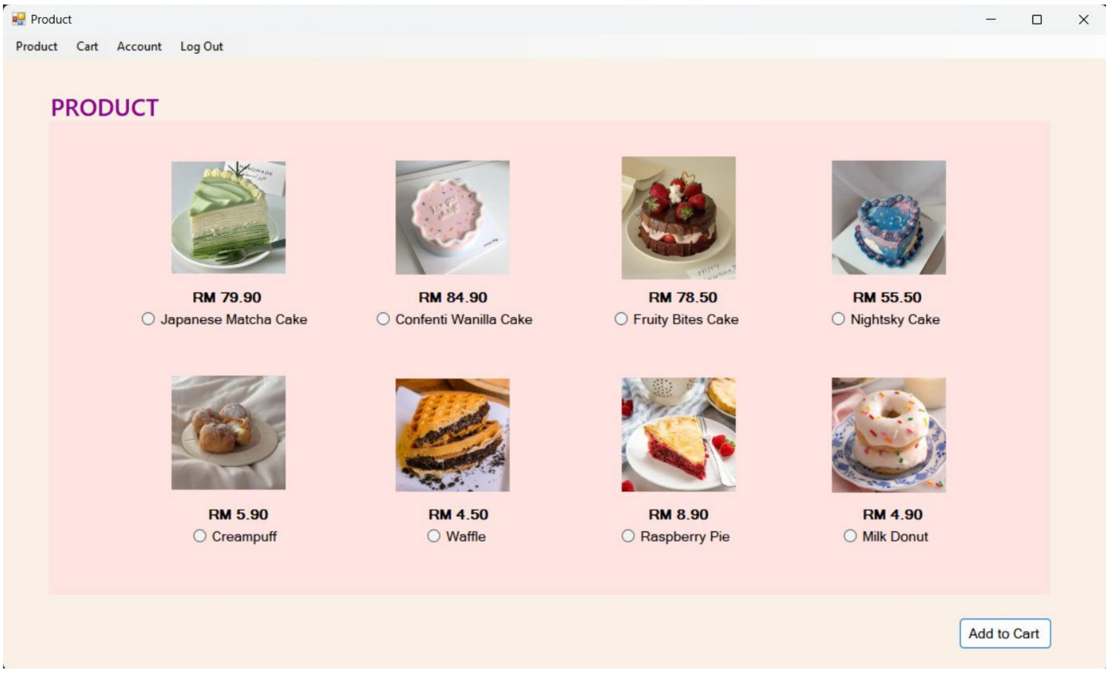


Figure 3.2.1: Product Menu Page

Figure 3.2.1 shows that graphical of product menu. Customers need to click on the radio button if they want to choose the product that want to Add to Cart. After that, customers need to click on “Add to Cart” button to move to the next steps.

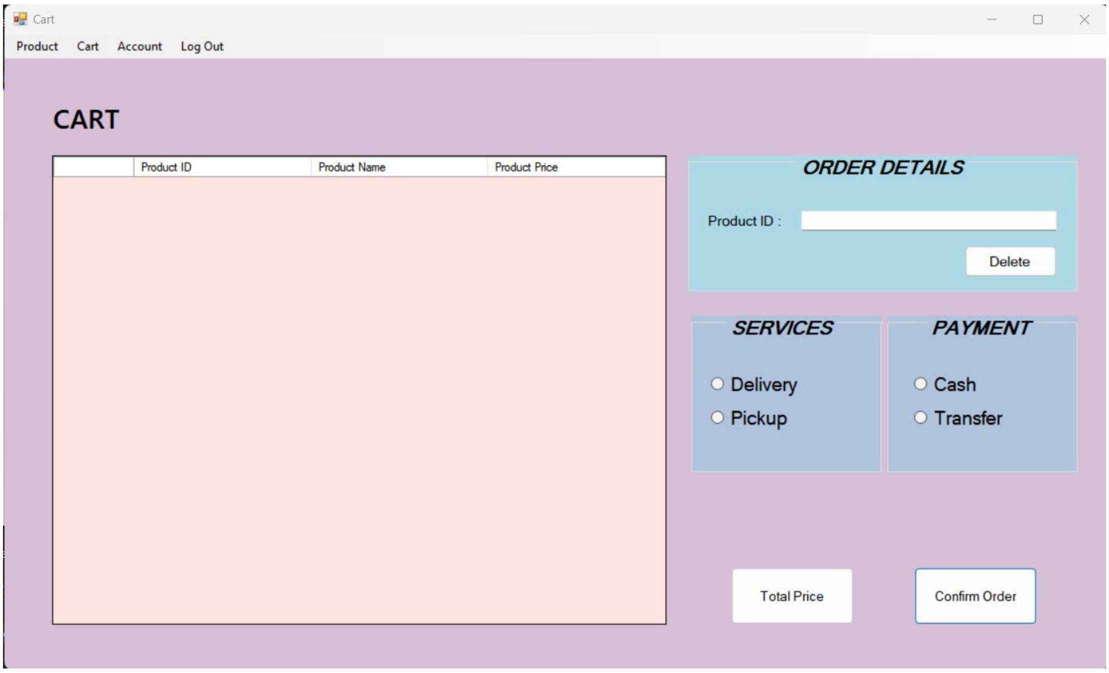
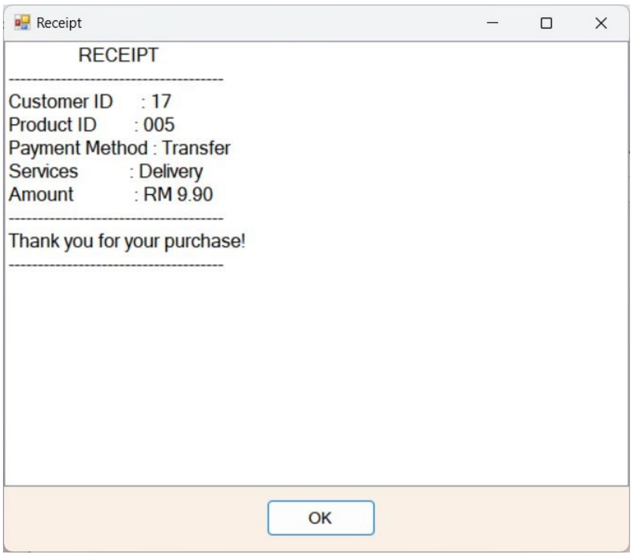


Figure 3.2.2: Cart

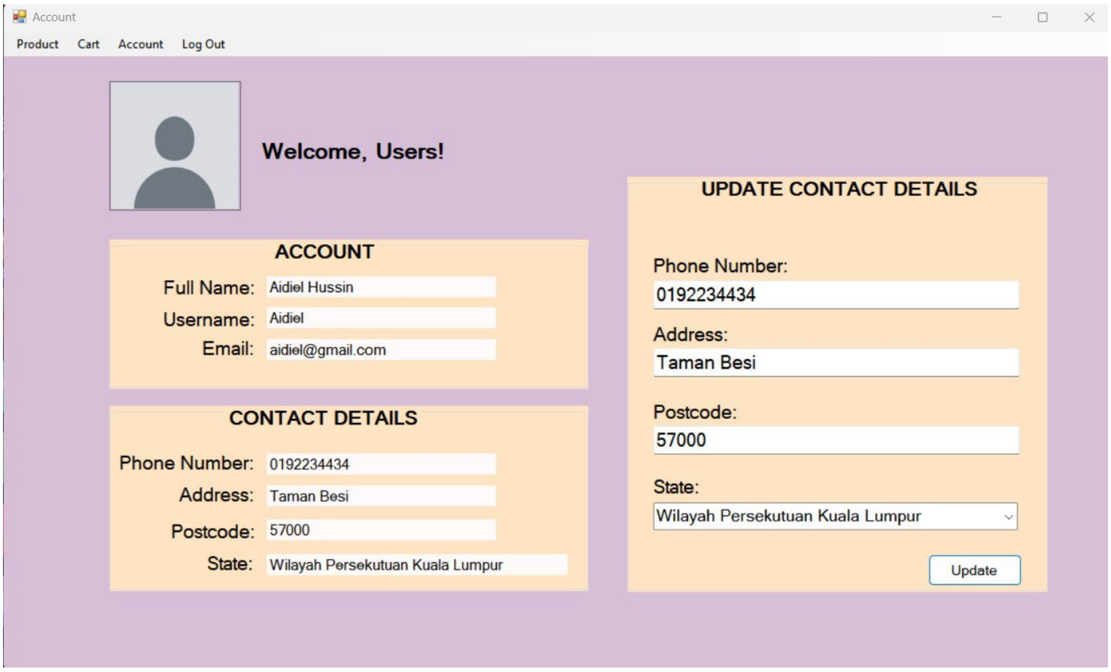
Figure 3.2.2 shows that the cart page after customer had add their product in cart. In the card grid, it will show the table list of products that had been add from the menu page. Customer also can delete their order on order details by searching using product id. For services customers had two option which is delivery and pickup. Same goes to payment, customers need to choose between cash or transfer. Their need to click on radio button to choose. After that, to know total of price order, customers need to click on "Total Price" button. Then, there can confirm their order by clicking "Confirm Order" button.



```
        RECEIPT
------------------------------------
Customer ID    : 17
Product ID     : 005
Payment Method : Transfer
Services       : Delivery
Amount         : RM 9.90
------------------------------------
Thank you for your purchase!
------------------------------------
```

Figure 3.2.3: Receipt

Figure 3.2.3 shows that receipt page after the customer had make a payment. In the receipt shows a customer information of service and payment, which is Customer ID, Product ID, Payment Method, Services and total amount that must pay.



Welcome, Users!

ACCOUNT

Full Name: Aidiel Hussin
Username: Aidiel
Email: aidiel@gmail.com

CONTACT DETAILS

Phone Number: 0192234434
Address: Taman Besi
Postcode: 57000
State: Wilayah Persekutuan Kuala Lumpur

UPDATE CONTACT DETAILS

Phone Number: 0192234434
Address: Taman Besi
Postcode: 57000
State: Wilayah Persekutuan Kuala Lumpur

Figure 3.2.4: Account Information

Figure shows that dashboard for users' account. User can view their account information and their contact details. User also can update their contact details by fulfill the textbox given and then click on "Update" button to update their contact information details.

4. Source Code

4.1 Login

```
Imports System.Data.OleDb

Public Class Login
    Dim connection As New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=D:\uni\mac24 - ogos 24\csc301\VB Project\NEW
CSC301\datosBakery.accdb")

    Private Sub btnSignIn_Click(sender As Object, e As
EventArgs) Handles btnSignIn.Click
        Dim username As String = txtUsername.Text
        Dim password As String = txtPassword.Text

        ' Check if the entered credentials are for admin
        If username = "admin" AndAlso password = "admin1234."
Then
            ' If the credentials match, open the
orderLogAdmin form
            Dim adminForm As New orderLogAdmin()
            adminForm.Show()
            Me.Hide()

       Else
            ' Create the select query
            Dim query As String = "SELECT * FROM tblCustomer
WHERE custUsername = @Username AND custPassword = @Password"

            ' Create the command and add the parameters
            Dim command As OleDbCommand = New
OleDbCommand(query, connection)
            command.Parameters.AddWithValue("@Username",
username)
            command.Parameters.AddWithValue("@Password",
password)

            Try
                ' Open the connection
                connection.Open()

                ' Execute the query and read the data
                Dim reader As OleDbDataReader =
command.ExecuteReader()
                If reader.Read() Then
                    ' If credentials are correct, get the
custID and open the next form
                    Dim custID As Integer =
Convert.ToInt32(reader("custID"))
```

```
                    Dim Account As New Account(custID,
username)
                    Account.Show()
                    Me.Hide()
                Else
                    MessageBox.Show("Invalid username or
password", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                End If

                ' Close the reader and the connection
                reader.Close()
                connection.Close()
            Catch ex As Exception
                MessageBox.Show("Error: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        End If
    End Sub

    Private Sub btnSignUp_Click(sender As Object, e As
EventArgs) Handles btnSignUp.Click
        ' Open the Register form
        Dim registerForm As New Register()
        registerForm.Show()
        Me.Hide()
    End Sub
 End Class
```

4.2 Register

```
Imports System.Data.OleDb

Public Class Register
    Dim connection As New OleDbConnection
    Dim command As OleDbCommand
    Dim sql As String = Nothing

    Private Sub Register_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
        connection.ConnectionString =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\uni\mac24 -
ogos 24\csc301\VB Project\NEW CSC301\datosBakery.accdb"
    End Sub

    Private Sub btnRegister_Click(sender As Object, e As
EventArgs) Handles btnRegister.Click
        ' Get the user input
        Dim fullName As String = txtFullName.Text
        Dim email As String = txtEmail.Text
        Dim username As String = txtUsername.Text
        Dim password As String = txtPassword.Text

        ' Validate the input
        If String.IsNullOrWhiteSpace(fullName) OrElse
String.IsNullOrWhiteSpace(email) OrElse
String.IsNullOrWhiteSpace(username) OrElse
String.IsNullOrWhiteSpace(password) Then
            MessageBox.Show("Please fill all the fields.",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            Return
        End If

        ' Create the insert query
        Dim query As String = "INSERT INTO tblCustomer
(custName, custEmail, custUsername, custPassword) VALUES
(@FullName, @Email, @Username, @Password)"

        ' Create the command and add the parameters
        Dim command As OleDbCommand = New OleDbCommand(query,
connection)
        command.Parameters.AddWithValue("@FullName",
fullName)
        command.Parameters.AddWithValue("@Email", email)
        command.Parameters.AddWithValue("@Username",
username)
        command.Parameters.AddWithValue("@Password",
password)

        Try
            ' Open the connection
            connection.Open()

            ' Execute the query
            command.ExecuteNonQuery()

            ' Show success message
            MessageBox.Show("Registration successful!",
"Success", MessageBoxButtons.OK, MessageBoxIcon.Information)

            ' Clear the textboxes
            txtFullName.Clear()
            txtEmail.Clear()
            txtUsername.Clear()
            txtPassword.Clear()

            Login.Show()
            Me.Hide()
        Catch ex As Exception
            ' Show error message
            MessageBox.Show("Error: " & ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        Finally
            ' Close the connection
            connection.Close()
        End Try
    End Sub

    Private Sub btnBack_Click(sender As Object, e As
EventArgs) Handles btnBack.Click
        Login.Show()
        Me.Hide()
    End Sub
End Class
```

4.3 Product

```
Public Class ProductList
    Private custID As Integer
    Private selectedProductID As String
    Private selectedProductName As String
    Private selectedProductPrice As Decimal
    Private cartPage As Cart = Nothing
    Dim username As String

    Public Sub New(ByVal custID As Integer, ByVal Username As
String)
        ' This call is required by the designer.
        InitializeComponent()
        ' Set the custID
        Me.custID = custID
        Me.username = Username
    End Sub

    Private Sub rdnJMC_CheckedChanged(sender As Object, e As
EventArgs) Handles rdnJMC.CheckedChanged
        If rdnJMC.Checked Then
            selectedProductID = "001"
            selectedProductName = "Japanese Matcha Cake"
            selectedProductPrice = 79.9D
        End If
    End Sub

    Private Sub rdnCWC_CheckedChanged(sender As Object, e As
EventArgs) Handles rdnCWC.CheckedChanged
        If rdnCWC.Checked Then
            selectedProductID = "002"
            selectedProductName = "Confenti Wanilla Cake"
            selectedProductPrice = 84.9D
        End If
    End Sub

    Private Sub rdnFBC_CheckedChanged(sender As Object, e As
EventArgs) Handles rdnFBC.CheckedChanged
        If rdnFBC.Checked Then
            selectedProductID = "003"
            selectedProductName = "Fruity Bites Cake"
            selectedProductPrice = 78.5D
        End If
    End Sub

    Private Sub rdnNC_CheckedChanged(sender As Object, e As
EventArgs) Handles rdnNC.CheckedChanged
        If rdnNC.Checked Then
            selectedProductID = "004"
            selectedProductName = "Nightsky Cake"
            selectedProductPrice = 55.5D
        End If
    End Sub

    Private Sub rdnCreampuff_CheckedChanged(sender As Object,
e As EventArgs) Handles rdnCreampuff.CheckedChanged
        If rdnCreampuff.Checked Then
            selectedProductID = "005"
            selectedProductName = "Creampuff"
            selectedProductPrice = 5.9D
        End If
    End Sub

    Private Sub rdnWaffle_CheckedChanged(sender As Object, e
As EventArgs) Handles rdnWaffle.CheckedChanged
        If rdnWaffle.Checked Then
            selectedProductID = "006"
            selectedProductName = "Waffle"
            selectedProductPrice = 4.5D
        End If
    End Sub

    Private Sub rdnRaspberryPie_CheckedChanged(sender As
Object, e As EventArgs) Handles
rdnRaspberryPie.CheckedChanged
        If rdnRaspberryPie.Checked Then
            selectedProductID = "007"
            selectedProductName = "Raspberry Pie"
            selectedProductPrice = 8.9D
        End If
    End Sub
    Private Sub rdnMilkDonut_CheckedChanged(sender As Object,
e As EventArgs) Handles rdnMilkDonut.CheckedChanged
        If rdnMilkDonut.Checked Then
            selectedProductID = "008"
            selectedProductName = "Milk Donut"
            selectedProductPrice = 4.9D
        End If
    End Sub

    Private Sub btnCart_Click(sender As Object, e As
EventArgs) Handles btnCart.Click
        If cartPage Is Nothing Then
            cartPage = New Cart(custID, username)
        End If

        cartPage.AddToCart(selectedProductID,
selectedProductName, selectedProductPrice)
        cartPage.Show()
        Me.Hide()
    End Sub

    Private Sub LogOutToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles LogOutToolStripMenuItem.Click
        Me.Close()
        Login.Show()
    End Sub


    Private Sub CartToolStripMenuItem_Click(sender As Object,
e As EventArgs) Handles CartToolStripMenuItem.Click
        Dim cartForm As New Cart(custID, username)
        cartForm.Show()
        Me.Hide()
    End Sub

    Private Sub AccountToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles
AccountToolStripMenuItem.Click
        Dim accountForm As New Account(custID, username)
        accountForm.Show()
        Me.Hide()
    End Sub
End Class
```

4.4 Cart

```
Imports System.Data.OleDb
Public Class Cart
    Private custID As Integer
    Dim username As String
    Dim connection As New OleDbConnection
    Dim adapter As New OleDbDataAdapter
    Dim ds As New DataSet
    Dim selectedProductPrice As Double = 0.0

    Public Sub New(ByVal custID As Integer, ByVal Username As
String)
        ' This call is required by the designer.
        InitializeComponent()

        ' Add any initialization after the
InitializeComponent() call.
        InitializeDataGridView()

        ' Set the custID
        Me.custID = custID
        Me.username = Username
    End Sub

    Private Sub Cart_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
        ' Set up your connection string
        connection.ConnectionString =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\uni\mac24 -
ogos 24\csc301\VB Project\NEW CSC301\datosBakery.accdb"
    End Sub

    Private Sub InitializeDataGridView()
        DataGridView1.Columns.Clear()
        DataGridView1.Columns.Add("ProductID", "Product ID")
        DataGridView1.Columns.Add("ProductName", "Product
Name")
        DataGridView1.Columns.Add("ProductPrice", "Product
Price")
    End Sub

    Public Sub AddToCart(productID As String, productName As
String, productPrice As Decimal)
        ' Add the product to the DataGridView
        DataGridView1.Rows.Add(productID, productName,
productPrice)
    End Sub
    Private Sub btnConfirm_Click(sender As Object, e As
EventArgs) Handles btnConfirm.Click
        If DataGridView1.Rows.Count > 0 Then
            Dim paymentMethod As String =
If(RadioCash.Checked, "Cash", "Transfer")
            Dim services As String =
If(radioDelivery.Checked, "Delivery", "Pickup")
```

```
            ' Calculate total price
            Dim totalPrice As Double = CalculateTotalPrice()

            For Each row As DataGridViewRow In
DataGridView1.Rows
                If Not row.IsNewRow Then
                    Dim productId As String =
row.Cells("ProductID").Value.ToString()
                    'Dim amount As Double =
Convert.ToDouble(row.Cells("ProductPrice").Value)'
                    Dim amount As Double = totalPrice

                    Try
                        ' Insert into database

                        InsertOrder(custID, productId,
paymentMethod, services, amount)
                    Catch ex As Exception
                        MessageBox.Show("Error confirming
order: " & ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                        Return
                    End Try
                End If
            Next

            ' Show receipt form with order details-
Convert.ToDouble(DataGridView1.Rows(0).Cells("ProductPrice").
Value)
            Dim receiptForm As New Receipt()
            receiptForm.SetReceiptData(custID,
DataGridView1.Rows(0).Cells("ProductID").Value.ToString(),
paymentMethod, services, totalPrice)
            receiptForm.ShowDialog()
        Else
            MessageBox.Show("No products in the cart to
confirm.")
        End If
    End Sub

    Private Function CalculateTotalPrice() As Double
        Dim totalPrice As Double = 0.0

        For Each row As DataGridViewRow In DataGridView1.Rows
            If Not row.IsNewRow Then
                totalPrice +=
Convert.ToDouble(row.Cells("ProductPrice").Value)
            End If
        Next
        If radioDelivery.Checked Then
            totalPrice += 4.0
        End If

        Return totalPrice
    End Function

    Private Sub InsertOrder(customerId As Integer, productId
As String, paymentMethod As String, services As String,
amount As Double)
        Try
            ' Prepare the INSERT command
            Dim query As String = "INSERT INTO tblOrder
(custID, productID, payment, service, amount) " &
                                  "VALUES (@custID,
@productID, @payment, @service, @amount)"

            ' Create and open connection
            Using cmd As New OleDbCommand(query, connection)
                cmd.Parameters.AddWithValue("@custID",
customerId)
                cmd.Parameters.AddWithValue("@productID",
productId)
                cmd.Parameters.AddWithValue("@payment",
paymentMethod)
                cmd.Parameters.AddWithValue("@service",
services)
                cmd.Parameters.AddWithValue("@amount",
amount)

                connection.Open()
                cmd.ExecuteNonQuery()
            End Using
        Catch ex As Exception
            Throw New Exception("Error inserting order: " &
ex.Message)
        Finally
            connection.Close()
        End Try
    End Sub

    Private Sub btnDelete_Click(sender As Object, e As
EventArgs) Handles btnDelete.Click
        ' Get the product ID from the text box
        Dim productIdToDelete As String = txtProductID.Text

        ' Search for the row with the specified product ID
and remove it
        For Each row As DataGridViewRow In DataGridView1.Rows
```

```
            If Not row.IsNewRow AndAlso
row.Cells("ProductID").Value.ToString() = productIdToDelete
Then
                DataGridView1.Rows.Remove(row)
                Exit For
            End If
        Next
    End Sub

    Private Sub btnTotalPrice_Click(sender As Object, e As
EventArgs) Handles btnTotalPrice.Click
        Dim totalPrice As Double = CalculateTotalPrice()
        MessageBox.Show("Total price: RM " &
totalPrice.ToString("F2"), "Message", MessageBoxButtons.OK,
MessageBoxIcon.Information)
    End Sub


    Private Sub ProductToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles
ProductToolStripMenuItem.Click
        ' Open the Product form
        Dim productListForm As New ProductList(custID,
username)
        productListForm.Show()
        Me.Hide()
    End Sub


    Private Sub LogOutToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles LogOutToolStripMenuItem.Click
        Me.Close()
        Login.Show()
    End Sub

    Private Sub AccountToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles
AccountToolStripMenuItem.Click
        Dim accountForm As New Account(custID, username)
        accountForm.Show()
        Me.Hide()
    End Sub
End Class
```

4.5 Receipt

```
Public Class Receipt
    Public Sub SetReceiptData(customerId As Integer,
productId As String, paymentMethod As String, services As
String, amount As Double)
```

```
        ' Clear listbox and display receipt details
        lstReceipt.Items.Clear()
        lstReceipt.Items.Add("               RECEIPT
")
        lstReceipt.Items.Add("-------------------------------
----------")

        ' Add receipt details with formatting
        lstReceipt.Items.Add($"Customer ID        :
{customerId}")
        lstReceipt.Items.Add($"Product ID           :
{productId}")
        lstReceipt.Items.Add($"Payment Method :
{paymentMethod}")
        lstReceipt.Items.Add($"Services             :
{services}")
        lstReceipt.Items.Add($"Amount              : RM
{amount:F2}")

        ' Add a footer
        lstReceipt.Items.Add("-------------------------------
----------")
        lstReceipt.Items.Add("Thank you for your purchase!")
        lstReceipt.Items.Add("-------------------------------
----------")
    End Sub

    Private Sub btnOK_Click(sender As Object, e As EventArgs)
Handles btnOK.Click
        Me.Close() ' Close the receipt form
    End Sub

End Class
```

4.6 Account

```
Imports System.Data.OleDb

Public Class Account
    Dim connection As New OleDbConnection
    Dim command As OleDbCommand
    Dim sql As String = Nothing
    Dim custID As Integer
    Dim username As String

    Public Sub New(ByVal custID As Integer, ByVal Username As
String)
        ' This call is required by the designer.
        InitializeComponent()
        ' Set the username
        Me.custID = custID
        Me.username = Username
        ' Set the connection string
        connection.ConnectionString =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\uni\mac24 -
ogos 24\csc301\VB Project\NEW CSC301\datosBakery.accdb"

        ' Load the data into the form
        LoadData()
    End Sub

    Private Sub LoadData()
        ' Create the select query
        Dim query As String = "SELECT * FROM tblCustomer
WHERE custUsername = @Username"

        ' Create the command and add the parameter
        Dim command As OleDbCommand = New OleDbCommand(query,
connection)
        command.Parameters.AddWithValue("@Username",
username)

        Try
            ' Open the connection
            connection.Open()

            ' Create an OleDbDataReader to read the data
            Dim reader As OleDbDataReader =
command.ExecuteReader()

            ' If there is a row returned, load the data into
the form
            If reader.Read() Then
                lblFullName.Text =
reader("custName").ToString()
                lblEmail.Text =
reader("custEmail").ToString()
                lblUsername.Text =
reader("custUsername").ToString()
                lblPhoneNumber.Text =
reader("custPhone").ToString()
                lblAddress.Text =
reader("custAddress").ToString()
                lblPostcode.Text =
reader("custPostcode").ToString()
                lblState.Text =
reader("custState").ToString()

            End If
            ' Close the reader and the connection
            reader.Close()
            connection.Close()
        Catch ex As Exception
            ' Show error message
            MessageBox.Show("Error: " & ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub

    Private Sub btnUpdate_Click(sender As Object, e As
EventArgs) Handles btnUpdate.Click
        ' Get the updated details from the textboxes
        Dim phoneNumber As String = txtPhoneNumber.Text
        Dim address As String = txtAddress.Text
        Dim postcode As String = txtPostCode.Text
        Dim state As String = String.Empty

        ' Check if cmbState has a selected item
        If cmbState.SelectedItem IsNot Nothing Then
            state = cmbState.SelectedItem.ToString()
        End If

        ' Check if any field is empty
        If String.IsNullOrWhiteSpace(phoneNumber) OrElse
       String.IsNullOrWhiteSpace(address) OrElse
       String.IsNullOrWhiteSpace(postcode) OrElse
       String.IsNullOrWhiteSpace(state) Then
            MessageBox.Show("Please fill all the fields.",
"Incomplete Information", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
            Return
        End If

        ' Create the update query
        Dim query As String = "UPDATE tblCustomer SET
custPhone = @PhoneNumber, custAddress = @Address,
custPostcode = @Postcode, custState = @State WHERE
custUsername = @Username"

        ' Create the command and add the parameters
        Dim command As OleDbCommand = New OleDbCommand(query,
connection)
        command.Parameters.AddWithValue("@PhoneNumber",
phoneNumber)
        command.Parameters.AddWithValue("@Address", address)
        command.Parameters.AddWithValue("@Postcode",
postcode)
        command.Parameters.AddWithValue("@State", state)
        command.Parameters.AddWithValue("@Username",
username)

        Try
            ' Open the connection
            connection.Open()

            ' Execute the query
            command.ExecuteNonQuery()

            ' Show success message
            MessageBox.Show("Update successful!", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information)

            ' Update the labels with the new values
            lblPhoneNumber.Text = phoneNumber
            lblAddress.Text = address
            lblPostcode.Text = postcode
            lblState.Text = state
        Catch ex As Exception
            ' Show error message
            MessageBox.Show("Error: " & ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        Finally
            ' Close the connection
            connection.Close()
        End Try
    End Sub

    Private Sub LogOutToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles LogOutToolStripMenuItem.Click
        Me.Close()
        Login.Show()
    End Sub

    Private Sub ProductToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles
ProductToolStripMenuItem.Click
        Dim productListForm As New ProductList(custID,
username)
        productListForm.Show()
        Me.Hide()
    End Sub


    Private Sub CartToolStripMenuItem_Click(sender As Object,
e As EventArgs) Handles CartToolStripMenuItem.Click
        Dim cartForm As New Cart(custID, username)
        cartForm.Show()
        Me.Hide()
    End Sub
End Class
```

4.7 Order Log (Admin)

```
Imports System.Data.OleDb
Imports System.Data.SqlClient
Imports System.Drawing.Printing
Imports
DatosBakeryOrderingSystem.datosBakeryDataSetTableAdapters

Public Class orderLogAdmin
    Private connectionString As String =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\uni\mac24 -
ogos 24\csc301\VB Project\NEW CSC301\datosBakery.accdb;"
    Private printTitle As String = "Order Log"
    Private rowIndex As Integer = 0

    Private Sub ProductToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles
ProductToolStripMenuItem.Click
        ProductAdmin.Show()
        Me.Hide()
    End Sub

    Private Sub LogOutToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles LogOutToolStripMenuItem.Click
        Me.Close()
        Login.Show()
    End Sub

    Private Sub orderLogAdmin_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the
'DatosBakeryDataSet1.tblOrder' table. You can move, or remove
it, as needed.

Me.TblOrderTableAdapter1.Fill(Me.DatosBakeryDataSet1.tblOrder
)
        LoadData()

        ' Add event handlers for selection change and double-
click
        AddHandler grdOrder.SelectionChanged, AddressOf
grdOrder_SelectionChanged
        AddHandler grdOrder.CellDoubleClick, AddressOf
grdOrder_CellDoubleClick

        ' Set up PrintPreviewDialog
        PrintPreviewDialog1.Document = PrintDocument1

    End Sub
    Private Sub grdOrder_SelectionChanged(sender As Object, e
As EventArgs) Handles grdOrder.SelectionChanged
        If btnUpdate.Text <> "Cancel" Then
            LoadSelectedRecord()
        End If
    End Sub

    Private Sub btnUpdate_Click(sender As Object, e As
EventArgs) Handles btnUpdate.Click
        If btnUpdate.Text = "Update" Then
            btnUpdate.Text = "Cancel"

        Else
            btnUpdate.Text = "Update"
            ClearInputFields()
        End If
    End Sub
    Private Sub btnSave_Click(sender As Object, e As
EventArgs) Handles btnSave.Click
        If btnUpdate.Text = "Cancel" Then
            UpdateRecord()
        End If
        btnUpdate.Text = "Update"
        LoadData()
    End Sub
    Private Sub btnDelete_Click(sender As Object, e As
EventArgs) Handles btnDelete.Click
        If MessageBox.Show(Me, "Do you want to DELETE?",
"Confirmation", MessageBoxButtons.YesNo,
MessageBoxIcon.Warning) = DialogResult.Yes Then
            DeleteRecord()
            LoadData()
        End If
    End Sub
    Private Sub ClearInputFields()
        lblOrderID.Text = ""
        lblCustomerID.Text = ""
        lblProductID.Text = ""
        cmbService.Text = ""
        cmbPayment.Text = ""
        txtTotalPrice.Text = ""
    End Sub

    Private Sub btnClear_Click(sender As Object, e As
EventArgs) Handles btnClear.Click
        ClearInputFields()
    End Sub

    Private Sub LoadSelectedRecord()
        If grdOrder.CurrentRow IsNot Nothing Then
```

```
            Try

                ' Load the selected record into the input
fields for editing
                lblOrderID.Text =
grdOrder.CurrentRow.Cells("DataGridViewTextBoxColumn1").Value
.ToString()
                lblCustomerID.Text =
grdOrder.CurrentRow.Cells("DataGridViewTextBoxColumn2").Value
.ToString()
                lblProductID.Text =
grdOrder.CurrentRow.Cells("DataGridViewTextBoxColumn3").Value
.ToString()
                cmbService.Text =
grdOrder.CurrentRow.Cells("DataGridViewTextBoxColumn4").Value
.ToString()
                cmbPayment.Text =
grdOrder.CurrentRow.Cells("DataGridViewTextBoxColumn5").Value
.ToString()
                txtTotalPrice.Text =
grdOrder.CurrentRow.Cells("DataGridViewTextBoxColumn6").Value
.ToString()
            Catch ex As Exception
                MessageBox.Show("Error: " & ex.Message)
            End Try
        Else
            MessageBox.Show("No row is selected. Please
select a row first.")
        End If
    End Sub

    Private Sub UpdateRecord()
           cmbService.Text = cmbService.Text.Trim()
        cmbPayment.Text = cmbPayment.Text.Trim()
        txtTotalPrice.Text = txtTotalPrice.Text.Trim()

        'If Not ValidateFields() Then Exit Sub

        Using conn As New OleDbConnection(connectionString)
            Dim cmd As New OleDbCommand("UPDATE tblOrder SET
service = @service, payment = @payment, amount = @amount
WHERE orderID = @orderID", conn)
            cmd.Parameters.AddWithValue("@service",
cmbService.Text)
            cmd.Parameters.AddWithValue("@payment",
cmbPayment.Text)
            cmd.Parameters.AddWithValue("@amount",
txtTotalPrice.Text)
            cmd.Parameters.AddWithValue("@orderID",
lblOrderID.Text)
```

```
            conn.Open()
            cmd.ExecuteNonQuery()
        End Using
    End Sub
    Private Sub DeleteRecord()
        If String.IsNullOrEmpty(lblOrderID.Text) Then
            MessageBox.Show("No order selected. Please select
an order to delete.")
            Exit Sub
        End If

        Using conn As New OleDbConnection(connectionString)
            Dim cmd As New OleDbCommand("DELETE FROM tblOrder
WHERE orderID = @orderID", conn)
            cmd.Parameters.AddWithValue("@orderID",
lblOrderID.Text)
            conn.Open()
            cmd.ExecuteNonQuery()
        End Using
    End Sub

    Private Sub LoadData()
        ' Load your data from the database and display it
        Dim dt As New DataTable()
        Using conn As New OleDbConnection(connectionString)
            Dim cmd As New OleDbCommand("SELECT * FROM
tblOrder", conn)
            conn.Open()
            Using reader As OleDbDataReader =
cmd.ExecuteReader()
                dt.Load(reader)
            End Using
        End Using
        ' grdOrder.DataSource = dt
        TblOrderBindingSource.DataSource = dt
        grdOrder.DataSource = TblOrderBindingSource

    End Sub

    Private Sub grdOrder_CellDoubleClick(sender As Object, e
As DataGridViewCellEventArgs)

        Dim selectedID =
grdOrder.CurrentRow.Cells("DataGridViewTextBoxColumn1").Value
.ToString()
        Dim selectedcust =
grdOrder.CurrentRow.Cells("DataGridViewTextBoxColumn2").Value
.ToString()
        Dim selectedProduct =
grdOrder.CurrentRow.Cells("DataGridViewTextBoxColumn3").Value
```

```
.ToString()
        Dim selectedService =
grdOrder.CurrentRow.Cells("DataGridViewTextBoxColumn4").Value
.ToString()

        MessageBox.Show($"Order ID: {selectedID}, User ID:
{selectedcust}, Product ID: {selectedProduct }, Service :
{selectedService}")
    End Sub

    Private Sub btnFirst_Click(sender As Object, e As
EventArgs) Handles btnFirst.Click
        TblOrderBindingSource.MoveFirst()
    End Sub

    Private Sub btnLast_Click(sender As Object, e As
EventArgs) Handles btnLast.Click
        TblOrderBindingSource.MoveLast()
    End Sub

    Private Sub btnPrevious_Click(sender As Object, e As
EventArgs) Handles btnPrevious.Click
        TblOrderBindingSource.MovePrevious()
    End Sub

    Private Sub btnNext_Click(sender As Object, e As
EventArgs) Handles btnNext.Click
        TblOrderBindingSource.MoveNext()
    End Sub

    Private Sub printDocument1_PrintPage(sender As Object, e
As PrintPageEventArgs) Handles PrintDocument1.PrintPage
        Dim title As String = "Users"
        Dim titleFont As New Font("Arial", 14,
FontStyle.Bold)
        Dim titleHeight As Integer =
e.Graphics.MeasureString(title, titleFont).Height

        ' Draw the title
        e.Graphics.DrawString(title, titleFont,
Brushes.Black, New PointF(0, 0))

        ' Create a bitmap of the DataGridView
        Dim bm As New Bitmap(Me.grdOrder.Width,
Me.grdOrder.Height)
        grdOrder.DrawToBitmap(bm, New Rectangle(0, 0,
Me.grdOrder.Width, Me.grdOrder.Height))

        ' Adjust the position where the DataGridView is drawn
to be below the title
```

```
        Dim dataGridViewPosition As New Point(0, titleHeight
+ 10) ' 10 pixels below the title
        e.Graphics.DrawImage(bm, dataGridViewPosition)
    End Sub

    Private Sub btnPrint_Click(sender As Object, e As
EventArgs) Handles btnPrint.Click
        PrintDialog1.Document = PrintDocument1
        If PrintDialog1.ShowDialog() = DialogResult.OK Then
            PrintPreviewDialog1.Document = PrintDocument1
            PrintPreviewDialog1.ShowDialog()
        End If
    End Sub

    Private Sub UsersToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles UsersToolStripMenuItem.Click
        Users.Show()
        Me.Hide()
    End Sub
End Class
```

4.8 Product (Admin)

```
Imports System.Drawing.Printing
Imports
DatosBakeryOrderingSystem.datosBakeryDataSetTableAdapters
Imports System.Data.OleDb

Public Class ProductAdmin
    Private connectionString As String =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\uni\mac24 -
ogos 24\csc301\VB Project\NEW CSC301\datosBakery.accdb"

    Private Sub btnPrint_Click(sender As Object, e As
EventArgs) Handles btnPrint.Click
        PrintDialog1.Document = PrintDocument1
        If PrintDialog1.ShowDialog() = DialogResult.OK Then
            PrintPreviewDialog1.Document = PrintDocument1
            PrintPreviewDialog1.ShowDialog()
        End If
    End Sub

    Private Sub printDocument1_PrintPage(sender As Object, e
As Printing.PrintPageEventArgs) Handles
PrintDocument1.PrintPage
        Dim title As String = "Products"
        Dim titleFont As New Font("Arial", 14,
FontStyle.Bold)
        Dim titleHeight As Integer =
e.Graphics.MeasureString(title, titleFont).Height
```

```
        ' Draw the title
        e.Graphics.DrawString(title, titleFont,
Brushes.Black, New PointF(0, 0))

        ' Create a bitmap of the DataGridView
        Dim bm As New Bitmap(Me.DataGridView1.Width,
Me.DataGridView1.Height)
        DataGridView1.DrawToBitmap(bm, New Rectangle(0, 0,
Me.DataGridView1.Width, Me.DataGridView1.Height))

        ' Adjust the position where the DataGridView is drawn
to be below the title
        Dim dataGridViewPosition As New Point(0, titleHeight
+ 10) ' 10 pixels below the title
        e.Graphics.DrawImage(bm, dataGridViewPosition)
    End Sub

    Private Sub btnAdd_Click(sender As Object, e As
EventArgs) Handles btnAdd.Click
        If btnAdd.Text = "Add" Then
            btnAdd.Text = "Cancel"
            ClearInputFields()
        Else
            btnAdd.Text = "Add"
            ClearInputFields()
        End If
    End Sub

    Private Sub btnUpdate_Click(sender As Object, e As
EventArgs) Handles btnUpdate.Click
        If btnUpdate.Text = "Update" Then
            btnUpdate.Text = "Cancel"
            LoadSelectedRecord()
        Else
            btnUpdate.Text = "Update"
            ClearInputFields()
        End If
    End Sub

    Private Sub btnSave_Click(sender As Object, e As
EventArgs) Handles btnSave.Click
        If btnAdd.Text = "Cancel" Then
            InsertRecord()
        ElseIf btnUpdate.Text = "Cancel" Then
            UpdateRecord()
        End If
        btnAdd.Text = "Add"
        btnUpdate.Text = "Update"
        LoadData()
    End Sub

    Private Sub btnDelete_Click(sender As Object, e As
EventArgs) Handles btnDelete.Click
        If MessageBox.Show(Me, "Do you want to DELETE?",
"Confirmation",
                           MessageBoxButtons.YesNo,
MessageBoxIcon.Warning) = DialogResult.Yes Then
            DeleteRecord()
            LoadData()
        End If
    End Sub

    Private Sub ClearInputFields()
        ' Clear your input fields here
        txtProductID.Text = ""
        txtProductName.Text = ""
        txtPrice.Text = ""
    End Sub

    Private Sub LoadSelectedRecord()
        If DataGridView1.CurrentRow IsNot Nothing Then
            Try
                ' Debugging output
                Dim selectedProductID =
DataGridView1.CurrentRow.Cells("ProductIDDataGridViewTextBoxC
olumn").Value.ToString()
                Dim selectedProductName =
DataGridView1.CurrentRow.Cells("ProductNameDataGridViewTextBo
xColumn").Value.ToString()
                Dim selectedPrice =
DataGridView1.CurrentRow.Cells("ProductPriceDataGridViewTextB
oxColumn").Value.ToString()



                ' Update the text fields
                txtProductID.Text = selectedProductID
                txtProductName.Text = selectedProductName
                txtPrice.Text = selectedPrice
            Catch ex As Exception
                MessageBox.Show("Error: " & ex.Message)
            End Try
        Else
            MessageBox.Show("No row is selected. Please
select a row first.")
        End If
    End Sub

    Private Sub InsertRecord()
        Using conn As New OleDbConnection(connectionString)
```

```
            Dim cmd As New OleDbCommand("INSERT INTO
tblProduct (productID, productName, productPrice) VALUES
(@productID, @productName, @productPrice)", conn)
            cmd.Parameters.AddWithValue("@productID",
txtProductID.Text)
            cmd.Parameters.AddWithValue("@productName",
txtProductName.Text)
            cmd.Parameters.AddWithValue("@productPrice",
txtPrice.Text)
            conn.Open()
            cmd.ExecuteNonQuery()
        End Using
    End Sub

    Private Sub UpdateRecord()
        Using conn As New OleDbConnection(connectionString)
            Dim cmd As New OleDbCommand("UPDATE tblProduct
SET productName = @productName, productPrice = @productPrice
WHERE productID = @productID", conn)
            cmd.Parameters.AddWithValue("@productName",
txtProductName.Text)
            cmd.Parameters.AddWithValue("@productPrice",
txtPrice.Text)
            cmd.Parameters.AddWithValue("@productID",
txtProductID.Text)
            conn.Open()
            cmd.ExecuteNonQuery()
        End Using
    End Sub

    Private Sub DeleteRecord()
        Using conn As New OleDbConnection(connectionString)
            Dim cmd As New OleDbCommand("DELETE FROM
tblProduct WHERE productID = @productID", conn)
            cmd.Parameters.AddWithValue("@productID",
txtProductID.Text)
            conn.Open()
            cmd.ExecuteNonQuery()
        End Using
    End Sub

    Private Sub LoadData()
        ' Load your data from the database and display it
        Dim dt As New DataTable()
        Using conn As New OleDbConnection(connectionString)
            Dim cmd As New OleDbCommand("SELECT * FROM
tblProduct", conn)
            conn.Open()
            Using reader As OleDbDataReader =
cmd.ExecuteReader()
```

```
                dt.Load(reader)
            End Using
        End Using
        'DataGridView1.DataSource = dt
        TblProductBindingSource.DataSource = dt
        DataGridView1.DataSource = TblProductBindingSource
    End Sub

    Private Sub btnFirst_Click(sender As Object, e As
EventArgs) Handles btnFirst.Click
        TblProductBindingSource.MoveFirst()
    End Sub

    Private Sub btnLast_Click(sender As Object, e As
EventArgs) Handles btnLast.Click
        TblProductBindingSource.MoveLast()
    End Sub

    Private Sub btnPrevious_Click(sender As Object, e As
EventArgs) Handles btnPrevious.Click
        TblProductBindingSource.MovePrevious()
    End Sub

    Private Sub btnNext_Click(sender As Object, e As
EventArgs) Handles btnNext.Click
        TblProductBindingSource.MoveNext()
    End Sub

    Private Sub DataGridView1_SelectionChanged(sender As
Object, e As EventArgs)
        LoadSelectedRecord()
    End Sub

    Private Sub ProductAdmin_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the
'DatosBakeryDataSet1.tblProduct' table. You can move, or
remove it, as needed.

Me.TblProductTableAdapter1.Fill(Me.DatosBakeryDataSet1.tblPro
duct)
        PrintPreviewDialog1.Document = PrintDocument1
        AddHandler DataGridView1.SelectionChanged, AddressOf
DataGridView1_SelectionChanged
        ' In your form's constructor or Load event
        AddHandler DataGridView1.CellDoubleClick, AddressOf
DataGridView1_CellDoubleClick

    End Sub
    Private Sub DataGridView1_CellDoubleClick(sender As
Object, e As DataGridViewCellEventArgs)
        ' Debugging output
        Dim selectedProductID =
DataGridView1.CurrentRow.Cells("ProductIDDataGridViewTextBoxC
olumn").Value.ToString()
        Dim selectedProductName =
DataGridView1.CurrentRow.Cells("ProductNameDataGridViewTextBo
xColumn").Value.ToString()
        Dim selectedPrice =
DataGridView1.CurrentRow.Cells("ProductPriceDataGridViewTextB
oxColumn").Value.ToString()

        ' Displaying values for debugging
        MessageBox.Show($"Selected Product ID:
{selectedProductID}, Product Name: {selectedProductName},
Price: {selectedPrice}")
    End Sub

    Private Sub OrderLogToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles
OrderLogToolStripMenuItem.Click
        orderLogAdmin.Show()
        Me.Hide()
    End Sub

    Private Sub LogOutToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles LogOutToolStripMenuItem.Click
        Me.Close()
        Login.Show()
    End Sub

    Private Sub UsersToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles UsersToolStripMenuItem.Click
        Me.Hide()
        Users.Show()
    End Sub

    Private Sub btnClear_Click(sender As Object, e As
EventArgs) Handles btnClear.Click
        ClearInputFields()
    End Sub
End Class
```

4.9 Users (Admin)

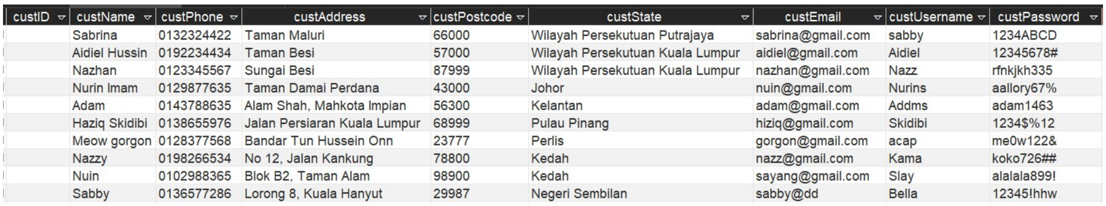
```
Imports System.Data.OleDb
Imports
DatosBakeryOrderingSystem.datosBakeryDataSetTableAdapters

```

```
Public Class Users
    Private connectionString As String =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\uni\mac24 -
ogos 24\csc301\VB Project\NEW CSC301\datosBakery.accdb"
    Private Sub Users_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
        ' Load data into the DataGridView on form load
        LoadData()

        ' Add event handlers for selection change and double-
click
        AddHandler DataGridView1.SelectionChanged, AddressOf
DataGridView1_SelectionChanged
        AddHandler DataGridView1.CellDoubleClick, AddressOf
DataGridView1_CellDoubleClick
    End Sub

    Private Sub btnAdd_Click(sender As Object, e As
EventArgs) Handles btnAdd.Click
        If btnAdd.Text = "Add" Then
            btnAdd.Text = "Cancel"
            ClearInputFields()
        Else
            btnAdd.Text = "Add"
            ClearInputFields()
        End If
    End Sub

    Private Sub btnUpdate_Click(sender As Object, e As
EventArgs) Handles btnUpdate.Click
        If btnUpdate.Text = "Update" Then
            btnUpdate.Text = "Cancel"

        Else
            btnUpdate.Text = "Update"
            ClearInputFields()
        End If
    End Sub

    Private Sub btnSave_Click(sender As Object, e As
EventArgs) Handles btnSave.Click
        If btnAdd.Text = "Cancel" Then
            InsertRecord()
        ElseIf btnUpdate.Text = "Cancel" Then
            UpdateRecord()
        End If
        btnAdd.Text = "Add"
        btnUpdate.Text = "Update"
        LoadData()
    End Sub
    Private Sub btnDelete_Click(sender As Object, e As
EventArgs) Handles btnDelete.Click
        If MessageBox.Show(Me, "Do you want to DELETE?",
"Confirmation", MessageBoxButtons.YesNo,
MessageBoxIcon.Warning) = DialogResult.Yes Then
            DeleteRecord()
            LoadData()
        End If
    End Sub

    Private Sub ClearInputFields()
        lblCustID.Text = "" ' Use the Text property of the
Label control
        txtUsername.Text = ""
        txtEmail.Text = ""
        txtPassword.Text = ""
        txtFullName.Text = ""
        comboState.Text = ""
        txtPostcode.Text = ""
        txtAddress.Text = ""
        txtNoPhone.Text = ""
    End Sub

    Private Sub DataGridView1_SelectionChanged(sender As
Object, e As EventArgs)
        If btnUpdate.Text <> "Cancel" Then
            LoadSelectedRecord()
        End If
    End Sub

    Private Sub DataGridView1_CellDoubleClick(sender As
Object, e As DataGridViewCellEventArgs)

        Dim selectedID =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn1").
Value.ToString()
        Dim selectedUsername =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn8").
Value.ToString()
        Dim selectedEmail =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn7").
Value.ToString()
        Dim selectedPhone =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn3").
Value.ToString()

        MessageBox.Show($"Selected User ID: {selectedID},
Username: {selectedUsername}, Email: {selectedEmail}, Phone :
{selectedPhone}")
```

```
    End Sub

    Private Sub LoadSelectedRecord()
        If DataGridView1.CurrentRow IsNot Nothing Then
            Try
                lblCustID.Text =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn1").
Value.ToString()
                txtUsername.Text =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn8").
Value.ToString()
                txtEmail.Text =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn7").
Value.ToString()
                txtPassword.Text =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn9").
Value.ToString()
                txtFullName.Text =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn2").
Value.ToString()
                comboState.Text =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn6").
Value.ToString()
                txtPostcode.Text =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn5").
Value.ToString()
                txtAddress.Text =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn4").
Value.ToString()
                txtNoPhone.Text =
DataGridView1.CurrentRow.Cells("DataGridViewTextBoxColumn3").
Value.ToString()
            Catch ex As Exception
                MessageBox.Show("Error: " & ex.Message)
            End Try
        Else
            MessageBox.Show("No row is selected. Please
select a row first.")
        End If
    End Sub

    Private Sub InsertRecord()
        ' Trim input fields
        txtUsername.Text = txtUsername.Text.Trim()
        txtEmail.Text = txtEmail.Text.Trim()
        txtPassword.Text = txtPassword.Text.Trim()
        txtFullName.Text = txtFullName.Text.Trim()
        comboState.Text = comboState.Text.Trim()
        txtPostcode.Text = txtPostcode.Text.Trim()
        txtAddress.Text = txtAddress.Text.Trim()
        txtNoPhone.Text = txtNoPhone.Text.Trim()


        If Not ValidateFields() Then Exit Sub

        Using conn As New OleDbConnection(connectionString)
            Dim cmd As New OleDbCommand("INSERT INTO
tblCustomer (custUsername, custEmail, custPassword, custName,
custState, custPostcode, custAddress, custPhone) VALUES
(@custUsername, @custEmail, @custPassword, @custName,
@custState, @custPostcode, @custAddress, @custPhone)", conn)
            cmd.Parameters.AddWithValue("@custUsername",
txtUsername.Text)
            cmd.Parameters.AddWithValue("@custEmail",
txtEmail.Text)
            cmd.Parameters.AddWithValue("@custPassword",
txtPassword.Text)
            cmd.Parameters.AddWithValue("@custName",
txtFullName.Text)
            cmd.Parameters.AddWithValue("@custState",
comboState.Text)
            cmd.Parameters.AddWithValue("@custPostcode",
txtPostcode.Text)
            cmd.Parameters.AddWithValue("@custAddress",
txtAddress.Text)
            cmd.Parameters.AddWithValue("@custPhone",
txtNoPhone.Text)
            conn.Open()
            cmd.ExecuteNonQuery()
        End Using
    End Sub

    Private Function ValidateFields() As Boolean
        If txtUsername.Text.Length > 50 Then
            MessageBox.Show("Username is too long. Maximum 50
characters.")
            Return False
        End If
        If txtEmail.Text.Length > 100 Then
            MessageBox.Show("Email is too long. Maximum 100
characters.")
            Return False
        End If
        If txtPassword.Text.Length > 50 Then
            MessageBox.Show("Password is too long. Maximum 50
characters.")
            Return False
        End If
        If txtFullName.Text.Length > 100 Then
            MessageBox.Show("Full Name is too long. Maximum
100 characters.")
            Return False
        End If
        If comboState.Text.Length > 50 Then
            MessageBox.Show("State is too long. Maximum 50
characters.")
            Return False
        End If
        If txtPostcode.Text.Length > 10 Then
            MessageBox.Show("Postcode is too long. Maximum 10
characters.")
            Return False
        End If
        If txtAddress.Text.Length > 255 Then
            MessageBox.Show("Address is too long. Maximum 255
characters.")
            Return False
        End If
        If txtNoPhone.Text.Length > 20 Then
            MessageBox.Show("Phone number is too long.
Maximum 20 characters.")
            Return False
        End If

        Return True
    End Function

    Private Sub UpdateRecord()
        ' Trim input fields
        txtUsername.Text = txtUsername.Text.Trim()
        txtEmail.Text = txtEmail.Text.Trim()
        txtPassword.Text = txtPassword.Text.Trim()
        txtFullName.Text = txtFullName.Text.Trim()
        comboState.Text = comboState.Text.Trim()
        txtPostcode.Text = txtPostcode.Text.Trim()
        txtAddress.Text = txtAddress.Text.Trim()
        txtNoPhone.Text = txtNoPhone.Text.Trim()

        If Not ValidateFields() Then Exit Sub

        Using conn As New OleDbConnection(connectionString)
            Dim cmd As New OleDbCommand("UPDATE tblCustomer
SET custUsername = @custUsername, custEmail = @custEmail,
custPassword = @custPassword, custName = @custName, custState
= @custState, custPostcode = @custPostcode, custAddress =
@custAddress, custPhone = @custPhone WHERE custID = @custID",
conn)
            cmd.Parameters.AddWithValue("@custUsername",
txtUsername.Text)
            cmd.Parameters.AddWithValue("@custEmail",
txtEmail.Text)
            cmd.Parameters.AddWithValue("@custPassword", 
txtPassword.Text)
            cmd.Parameters.AddWithValue("@custName", 
txtFullName.Text)
            cmd.Parameters.AddWithValue("@custState", 
comboState.Text)
            cmd.Parameters.AddWithValue("@custPostcode", 
txtPostcode.Text)
            cmd.Parameters.AddWithValue("@custAddress", 
txtAddress.Text)
            cmd.Parameters.AddWithValue("@custPhone", 
txtNoPhone.Text)
            cmd.Parameters.AddWithValue("@custID", 
lblCustID.Text)
            conn.Open()
            cmd.ExecuteNonQuery()
        End Using
    End Sub

    Private Sub DeleteRecord()
        If String.IsNullOrEmpty(lblCustID.Text) Then
            MessageBox.Show("No customer selected. Please 
select a customer to delete.")
            Exit Sub
        End If

        Using conn As New OleDbConnection(connectionString)
            Dim cmd As New OleDbCommand("DELETE FROM 
tblCustomer WHERE custID = @custID", conn)
            cmd.Parameters.AddWithValue("@custID", 
lblCustID.Text)
            conn.Open()
            cmd.ExecuteNonQuery()
        End Using
    End Sub

    Private Sub LoadData()
        ' Load your data from the database and display it
        Dim dt As New DataTable()
        Using conn As New OleDbConnection(connectionString)
            Dim cmd As New OleDbCommand("SELECT * FROM 
tblCustomer", conn)
            conn.Open()
            Using reader As OleDbDataReader = 
cmd.ExecuteReader()
                dt.Load(reader)
            End Using
        End Using
        'DataGridView1.DataSource = dt
        TblCustomerBindingSource1.DataSource = dt
        DataGridView1.DataSource = TblCustomerBindingSource1
    End Sub

    Private Sub btnClear_Click(sender As Object, e As
EventArgs) Handles btnClear.Click
        ClearInputFields()
    End Sub

    Private Sub btnFirst_Click(sender As Object, e As
EventArgs) Handles btnFirst.Click
        TblCustomerBindingSource1.MoveFirst()
    End Sub

    Private Sub btnLast_Click(sender As Object, e As
EventArgs) Handles btnLast.Click
        TblCustomerBindingSource1.MoveLast()
    End Sub

    Private Sub btnPrevious_Click(sender As Object, e As
EventArgs) Handles btnPrevious.Click
        TblCustomerBindingSource1.MovePrevious()
    End Sub

    Private Sub btnNext_Click(sender As Object, e As
EventArgs) Handles btnNext.Click
        TblCustomerBindingSource1.MoveNext()
    End Sub

    Private Sub OrderLogToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles
OrderLogToolStripMenuItem.Click
        orderLogAdmin.Show()
        Me.Hide()
    End Sub

    Private Sub ProductToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles
ProductToolStripMenuItem.Click
        ProductAdmin.Show()
        Me.Hide()
    End Sub

    Private Sub LogOutToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles LogOutToolStripMenuItem.Click
        Me.Close()
        login.Show()
    End Sub

    Private Sub btnPrint_Click(sender As Object, e As
EventArgs) Handles btnPrint.Click
        PrintDialog1.Document = PrintDocument1
        If PrintDialog1.ShowDialog() = DialogResult.OK Then
            PrintPreviewDialog1.Document = PrintDocument1
            PrintPreviewDialog1.ShowDialog()
        End If

    End Sub

    Private Sub printDocument1_PrintPage(sender As Object, e
As Printing.PrintPageEventArgs) Handles
PrintDocument1.PrintPage
        Dim title As String = "Users"
        Dim titleFont As New Font("Arial", 14,
FontStyle.Bold)
        Dim titleHeight As Integer =
e.Graphics.MeasureString(title, titleFont).Height

        ' Draw the title
        e.Graphics.DrawString(title, titleFont,
Brushes.Black, New PointF(0, 0))

        ' Create a bitmap of the DataGridView
        Dim bm As New Bitmap(Me.DataGridView1.Width,
Me.DataGridView1.Height)
        DataGridView1.DrawToBitmap(bm, New Rectangle(0, 0,
Me.DataGridView1.Width, Me.DataGridView1.Height))

        ' Adjust the position where the DataGridView is drawn
to be below the title
        Dim dataGridViewPosition As New Point(0, titleHeight
+ 10) ' 10 pixels below the title
        e.Graphics.DrawImage(bm, dataGridViewPosition)
    End Sub
End Class
```

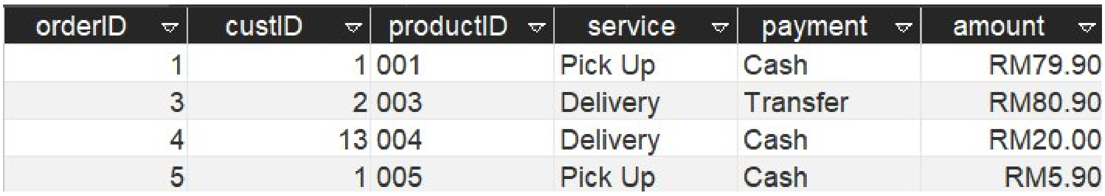
5. Database Tables’ Print Screens

5.1 Table Customer



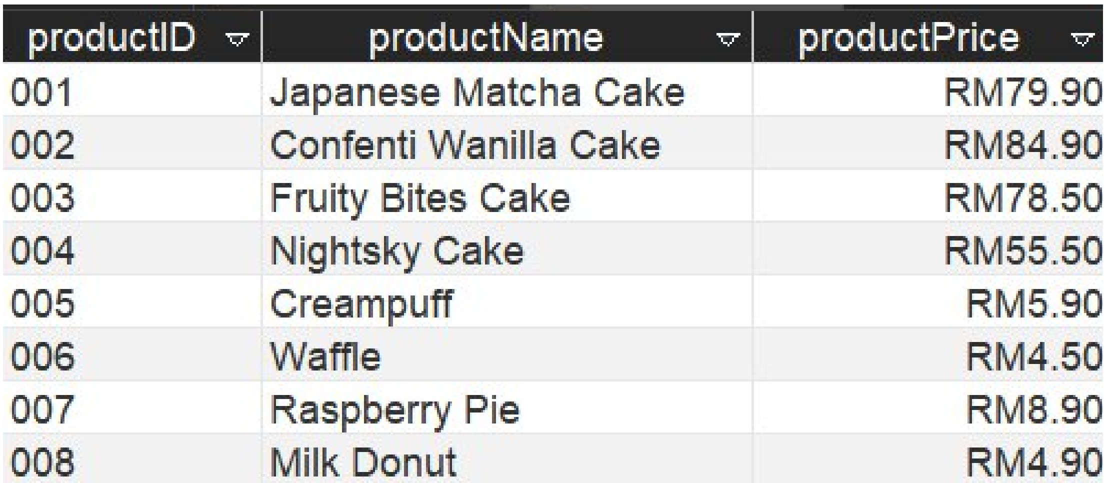
custID	custName	custPhone	custAddress	custPostcode	custState	custEmail	custUsername	custPassword
	Sabrina	0132324422	Taman Maluri	66000	Wilayah Persekutuan Putrajaya	sabrina@gmail.com	sabby	1234ABCD
	Aidiel Hussin	0192234434	Taman Besi	57000	Wilayah Persekutuan Kuala Lumpur	aidiel@gmail.com	Aidiel	12345678#
	Nazhan	0123345567	Sungai Besi	87999	Wilayah Persekutuan Kuala Lumpur	nazhan@gmail.com	Nazz	rfnkjkh335
	Nurin Imam	0129877635	Taman Damai Perdana	43000	Johor	nuin@gmail.com	Nurins	aallory67%
	Adam	0143788635	Alam Shah, Mahkota Impian	56300	Kelantan	adam@gmail.com	Addms	adam1463
	Haziq Skidibi	0138655976	Jalan Persiaran Kuala Lumpur	68999	Pulau Pinang	hiziq@gmail.com	Skidibi	1234$%12
	Meow gorgon	0128377568	Bandar Tun Hussein Onn	23777	Perlis	gorgon@gmail.com	acap	me0w122&
	Nazzy	0198266534	No 12, Jalan Kankung	78800	Kedah	nazz@gmail.com	Kama	koko726##
	Nuin	0102988365	Blok B2, Taman Alam	98900	Kedah	sayang@gmail.com	Slay	alalala899!
	Sabby	0136577286	Lorong 8, Kuala Hanyut	29987	Negeri Sembilan	sabby@dd	Bella	12345!hhw

5.2 Table Order



orderID	custID	productID	service	payment	amount
1	1	001	Pick Up	Cash	RM79.90
3	2	003	Delivery	Transfer	RM80.90
4	13	004	Delivery	Cash	RM20.00
5	1	005	Pick Up	Cash	RM5.90

5.3 Table Product



productID	productName	productPrice
001	Japanese Matcha Cake	RM79.90
002	Confenti Wanilla Cake	RM84.90
003	Fruity Bites Cake	RM78.50
004	Nightsky Cake	RM55.50
005	Creampuff	RM5.90
006	Waffle	RM4.50
007	Raspberry Pie	RM8.90
008	Milk Donut	RM4.90

6.0 Conclusion

Dato's Bakery Shop understands the importance of supporting both its employees and customers through proactive improvements. By introducing an online ordering system, the bakery aims to make ordering more convenient for customers while also streamlining operations and boosting efficiency. This move not only helps the bakery expand its customer base but also addresses current challenges caused by manual order processing. Embracing digital technology is a strategic step for Dato's Bakery to remain competitive in a fast-changing market, ensuring smooth operations across its various locations and ultimately enhancing customer satisfaction. By integrating technology thoughtfully, Dato's Bakery Shop is committed to maintaining its high standards and preparing for future growth opportunities in today's rapidly advancing digital landscape.

7.0 References

7.1 YouTube

- K S, (2019). How To Create A Database (Table) In Windows Visual Studio. Retrieved on 10th July 2024 from https://www.youtube.com/watch?v=YWE5OjBaa98.
- B L 4 C K Gaming, (2019). How to Connect MS ACCESS Database to Visual Studio and Login Form. Retrieved on 10th July 2024 from https://www.youtube.com/watch?v=8clalCxTd79s.

7.2 Internet

- Anonymous, (2023). Connect to an Access database in .NET Framework applications. Retrieved on 11th July 2024 from https://learn.microsoft.com/en-us/visualstudio/data-tools/connect-to-data-in-an-access-database-windows-forms?view=vs-2022.
- JohnSaunders, (2013). How to Display data in datagridview from access database. Retrieved on 11th July 2024 from https://stackoverflow.com/questions/15149491/how-to-display-data-in-datagridview-from-access-database.