

Project Title: Desktop Application for Canteen Management
And Ordering for Academic Institution

By:

Fateen Ahmed

Abdullah Jamal Abdullah Al Balushi

Hamed Amaad Hamed Al Manthari

Ibrahim Salim Humaid Salim Al Busaidi

Acknowledgement

Thanks, and gratitude is expressed to our teachers as well as our college dean Dr. Ahmed Al Balushi for giving us an opportunity to work on this project to develop an application for use in the canteen of universities. And special thanks to our Integrated Project II teachers Ms. Nejila and Ms. Shahida for their support throughout the development of this application and their guidance through this semester. They helped us in doing research and learning different techniques to implement in our application, which also increased our knowledge in the vb.net programming language and database.

Abstract

The following report reviews and implements the working of an online canteen management and ordering system that allows the customers to register online, browse and select food from the e-menu list, and place orders digitally by simply selecting the food that the customer desires. The choice of ordered food will display on the screen near the waiter who prepares the dish. The report discusses about the various features of the application such as the management options offered to the staff and the ordering options that are presented to the students and customers. The user interface and wireframes are designed and documented. The software application is developed through Visual Studio Development Environment and programmed in visual basic and SQL. Moreover, design frameworks such as Guna and .Net are utilized to improve the graphics of the application. The report will also include a brief introduction to the project. The literature review covers the information about the existing systems and their drawbacks. A discussion of the observed outcomes, marketing and commercialization, and future project additions are documented as well.

Table of Contents

Acknowledgement.....	2
Abstract	3
Project Introduction.....	5
Literature Review	6
Introduction	6
I. Cashless Canteen Management System Journal	6
Project Overview.....	6
Limitations of existing system	6
Need for the New System	7
Advantages of the New System	7
II. Canteen Management System by DAccess	7
Project Overview.....	7
Limitations of existing system	7
Need for the New System	8
Advantages of the New System	8
III. Online Canteen Management Journal	8
Project Overview.....	8
Limitations of existing system	8
Need for the New System	9
Advantages of the New System	9
Design and Implementation.....	10
Wireframes and Prototyping	10
Dashboard.....	11
Login and Load Up Page	11
Management Features	11
Ordering Features	11
Code.....	11
Results and Discussion.....	12
Marketing and Commercialization	19
Conclusions and Recommendations.....	20
Future Work	21
References	22
Appendix	23
All Wireframes	23
Source Code.....	25

Project Introduction

College canteen is often crowded with students during their break time, resulting in a long queue that students will spend a long time waiting in the queue and a large portion of their break time is wasted. Order N Eat (ONE) is a software application that is usable for students as well as the canteen staff. ONE allows students to use their personal computers or the university computer systems to place an order, and for the canteen staff, it is possible to manage the different essential systems to organize the work flow of the canteen. Using an application like ONE will make the students spend less time in the canteen queues, thus increasing the time they can spend for working on their courses or winding down from a long day at the university. The easier management it provides for the staff also help in reducing the queue times since the flow of orders is faster due to the efficient organization. The languages used to develop this program are vb.net, as well as SQL database to provide tables to the application, and some of the table have automation that record the purchase sale made by the user. This paper will discuss the application and its core functions, and explain the design and implementation of the program, and the marketing strategy used to advertise the application, and finally the future improvements that can be introduced later to the application in the future.

Literature Review

Introduction

The application provides an important service to academic institutions; therefore, it is vital that the application proves to be relevant in solving the problems it proposes. As a result, research and review prove to be essential activities that need to be conducted in order to implement the project successfully. This section of the report elaborates on the research conducted on existing techniques and applications used for canteen management. Moreover, it investigates the limitations of such techniques or applications and states the new features of the proposed application that aim to provide a better option to the users.

I. Cashless Canteen Management System Journal

Project Overview

This report suggests an online website for managing orders for a canteen at a particular institution. The report states that the purpose for developing software to manage a canteen is due to a persistent problem of long queues at canteens. This problem leads to disruption of class timings, receiving of wrong orders, and payment related issues. The project mainly focusses on addressing the problems related to payment. An online website is developed that displays the menu, and lets the students select their order and pay through card. The website is developed by means of JavaScript and its libraries. Therefore, the developed software solves cash related issues such as the student not having the correct amount in cash or the canteen not having the appropriate balance. (Kumar, et al., 2020)

Limitations of existing system

The development process of the application involves the linking of various libraries and program modules as demonstrated by the journal. This leads to difficulties in debugging and updating the program. Moreover, the software's primary target is to solve payment related problems; consequently, other aspects of the software lack key features such as inventory organization. Also, the user interface of the application is basic. (Kumar, et al., 2020)

Need for the New System

As canteens are a common site of interaction for students, it is important that the application provides a modern user interface that is familiar to use. Furthermore, the software needs to provide all the basic management tools to the canteen staff. In addition, a student's view of the application, assists the canteen staff while adding or updating new items into the menu. As commercial ordering applications continue to improve their usability, it is significant to update the management and ordering applications for canteens to provide a similar user experience. (Kumar, et al., 2020)

Advantages of the New System

The proposed software provides the managing and ordering facilities of canteens in a single user-friendly application. When compared to similar existing software, the proposed application is easier to update and debug, as it is programmed by using minimal libraries and languages. The application is designed as a modern tool such that the student can easily familiarize with the user interface. As the software is a desktop application, it requires lesser loading time when compared to a website. (Kumar, et al., 2020)

II. Canteen Management System by DAccess

Project Overview

Every day, technology advances, bringing new innovations and revolutionary projects to the market. This section describes the limits of the existing system, the necessity for the new system, and the benefits of the new system over the prior hardware system.

Limitations of existing system

The existing system uses outdated, fixed technology that is difficult to update. It also employs distinct devices for ordering and management. Furthermore, the large design takes up more space than it should. Moreover, the system installation is difficult to administer. Additionally, because the system is based on outdated technology, it operates slowly and inefficiently. In addition, because the ordering devices rely on a continual connection to the management devices, if the ordering devices lose connection to the management devices, the canteen's system would operate slowly.

Need for the New System

Because the old system does not allow for easy frequent updates, the new system must allow for easy regular updates. Furthermore, because it employs distinct devices for ordering and management systems, a new system to supply both systems in all devices is required to simplify the system. Because the system also takes up more space than it should, a new system that takes up less space is required. The market also favors the need for a new system that is simple to install. A new system that is fast and efficient is also required. To boost system efficiency, a new system must be dependable and the devices must be capable of working independently.

Advantages of the New System

The new system introduces new technologies that enables easy frequent updates that can increase system performance. It also provides all of the system's features, such as ordering and management, to every device in the system. The canteen is able to use whichever Windows device it wants, whether it's a tablet, laptop, or something else. Furthermore, the system is incredibly simple to set up. Only Windows devices should be used to install the program. Furthermore, because it runs on the Windows operating system, the system delivers quick and efficient service.

III. Online Canteen Management Journal

Project Overview

Online canteen system is a project offering online ordering system for the customers. This project uses an online payment system; however, paperwork is used for all the data, files and billing system in the canteen. Managing the files and billing system manually could be very time consuming which reduces the efficiency of the canteen management.

Limitations of existing system

Many of the existing systems are not utilizing automation to organize and manage the canteens, which slows down the process of storing data because of manual observation. Depending on manual observation to manage the canteen will consume a lot of time, which can be used for something more important while leaving the management and storing data to automation. It is also important to move away from paperwork because it reduces waste production and helps our environment.

Need for the New System

To create a better and more efficient system, automation can greatly help to reduce the time spent on inventory management, analysis of sales and predicting of preferred food by customers, and many other processes. automation will also shift from using paper billing system and files to digital bills through emails and storing the data digitally.

Advantages of the New System

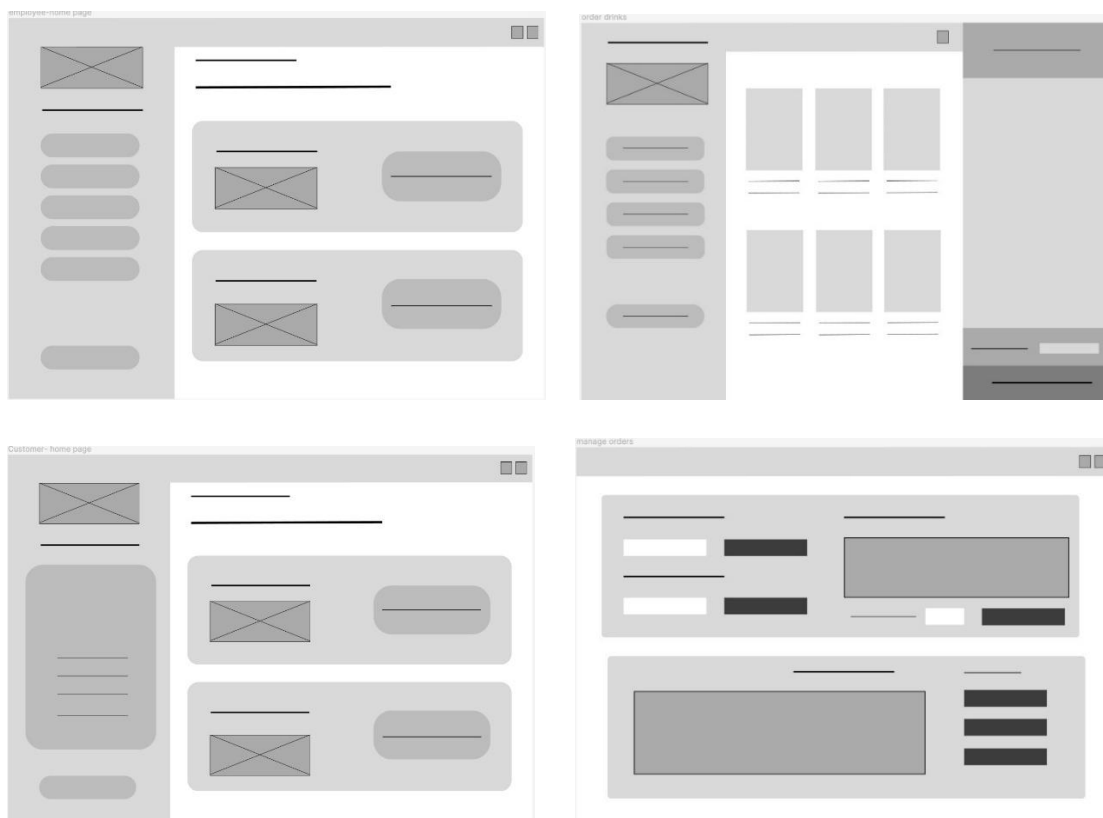
The new automated system will bring many advantages to the canteen business. The first advantage is the time efficiency that automation provides, replacing manual system with a system that performs everything digitally will reduce the unnecessary time and money wasted on paperwork. It will also reduce the waste production which helps our environment. Another advantage is that the bills for the customer can be preserved longer since it will be digital through e-mail or the application itself, while on the other hand, paper bills can be damaged or lost very quickly. Moreover, in an automated system, it is easier to record the favorite food item that is ordered by the customers, by analyzing and observing the data. These advantages will help making the management of the canteen easier and more efficient.

Design and Implementation

The desktop application is developed using the programming language Visual Basic. The code development process is executed and compiled by the means of the Microsoft's Visual Studio IDE. In addition, the controls provided by Visual Basic, an extended framework is installed and applied. The framework termed as Guna provides latest controls and features and provides modern looking graphic components for the application. Furthermore, in order to design and implement the database for the application a popular query language known as SQL is utilized that is embedded into Visual Studio.

The preliminary design and user interface is carried out using the Figma graphic editor. The design and user interface are prototyped based on principles such as easy usability, accessibility and consistency. The initial wireframes are as follows:

Wireframes and Prototyping



Once the design is confirmed, the various controls and features of the application are programmed.

Dashboard

First, a dashboard is implemented for both the use cases. The first case, that is for the staff members, a menu of management is designed. The menu contains features such as Manage Bulk Orders, Manage Inventory, Manage Sales and an Overview Page. These features are presented as buttons that launch the specific Windows Form when a click event is registered. For the second case, that is for student, a profile tab is introduced instead of the menus. The ordering section is common to both the staff and student.

Login and Load Up Page

A login page for both the student and staff is developed. The accounts for the users are stored in a database. However, a feature to add new accounts to the database is provided for the users. The Load Up page is designed to start the execution of the program.

Management Features

Each management task is divided into a separate form, with a dedicated database. The staff members can add, delete, search and update for the required elements in these forms. Additionally, a view of the database is also provided for the staff members so that the staff is aware of the elements of the database.

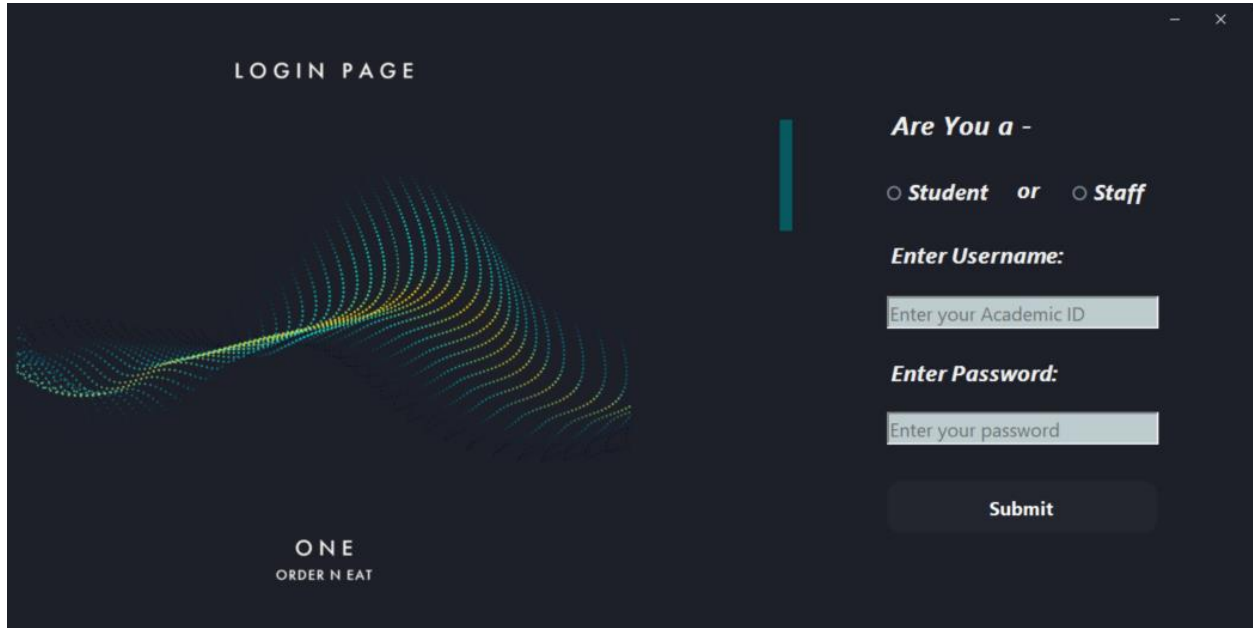
Ordering Features

The Ordering section is divided into two sub sections of Food and Drinks. The sections consist of a vast menu of the items that are available for ordering. Students can select their way of payment and order the items in this section.

Code

The code is provided in the Appendix section.

Results and Discussion



LOGIN PAGE

Are You a -

☐ **Student** or ☐ **Staff**

Enter Username:

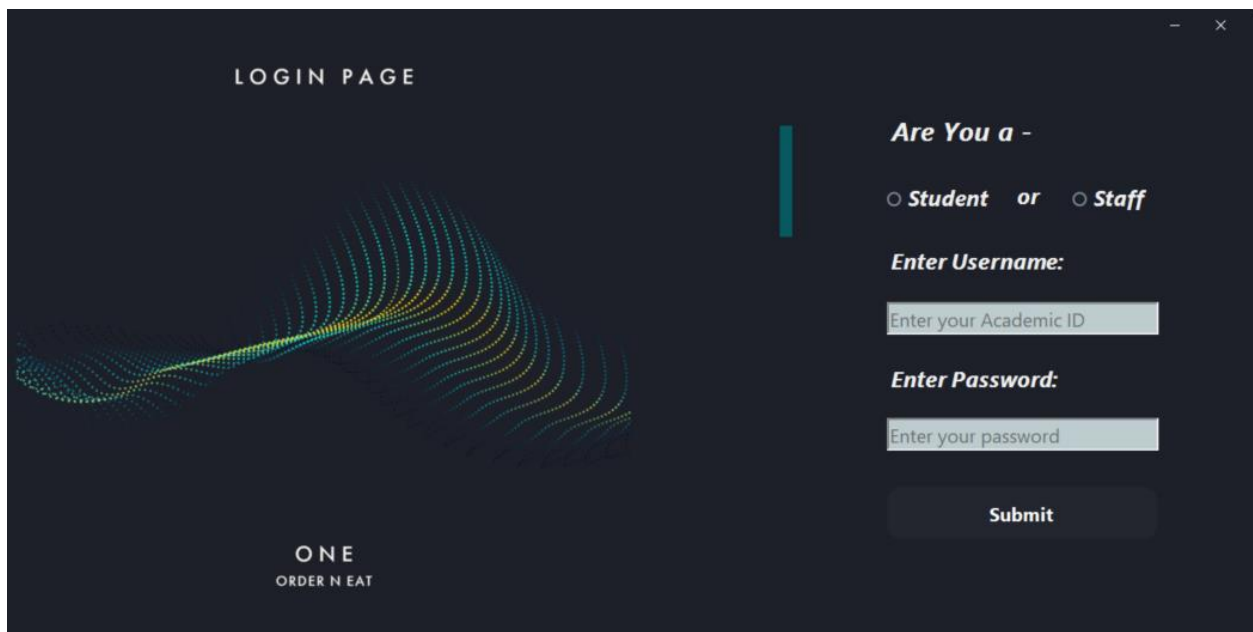
Enter your Academic ID

Enter Password:

Enter your password

Submit

ONE
ORDER N EAT



LOGIN PAGE

Are You a -

☐ **Student** or ☐ **Staff**

Enter Username:

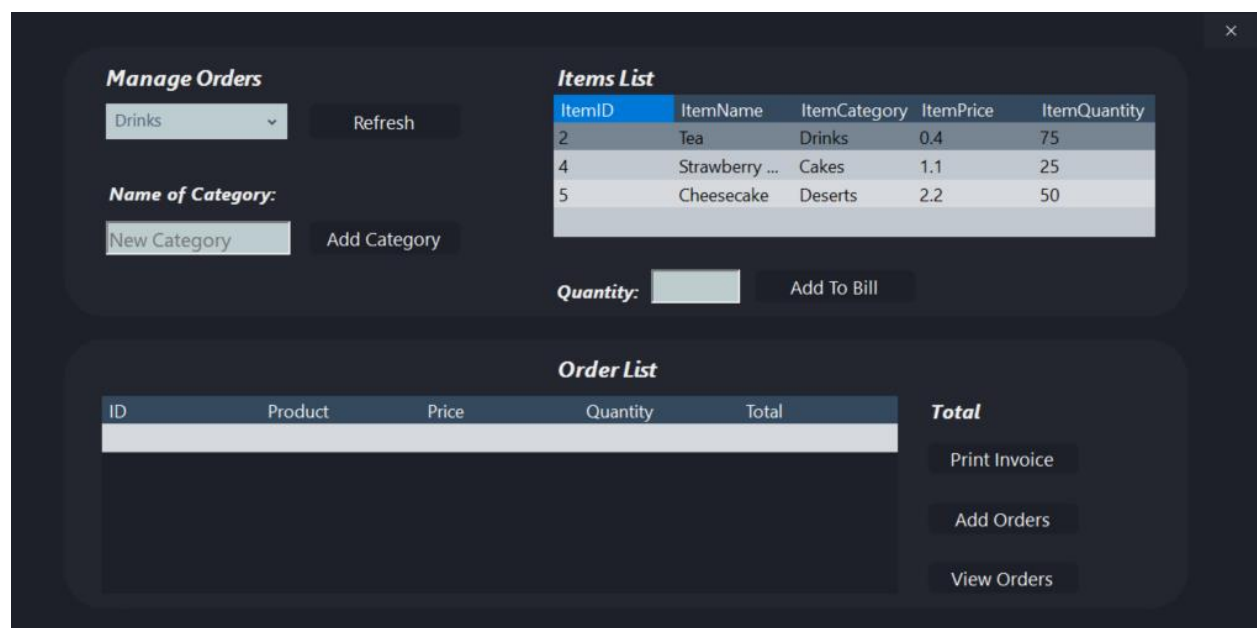
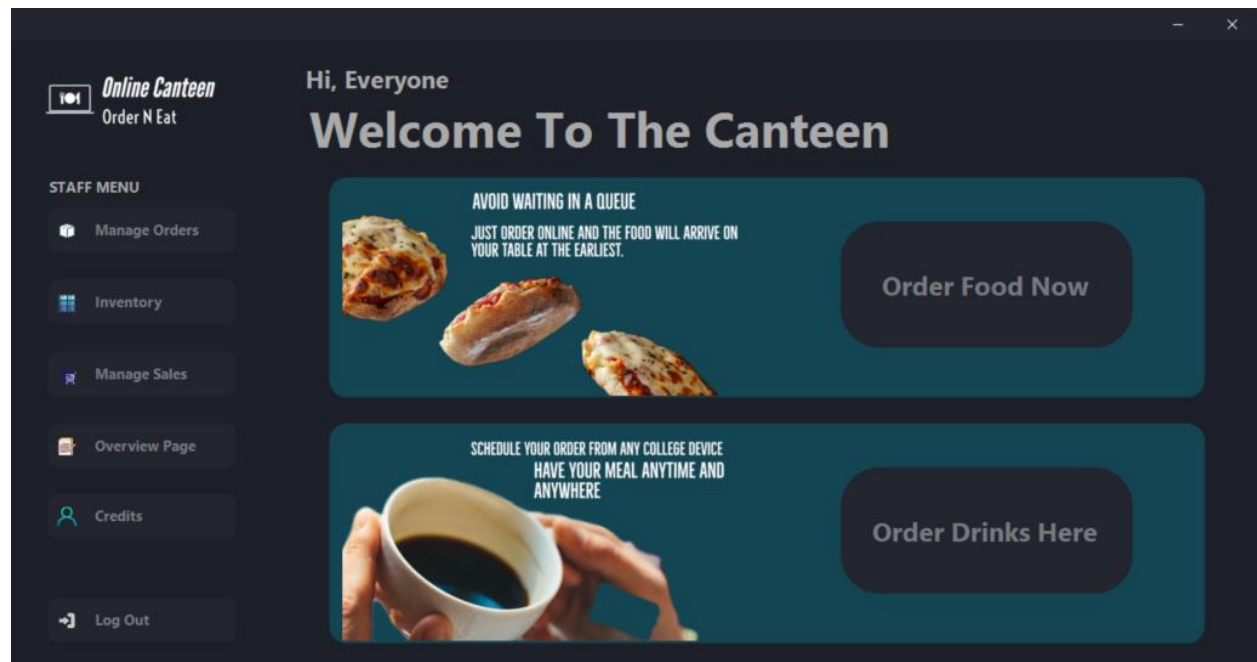
Enter your Academic ID

Enter Password:

Enter your password

Submit

ONE
ORDER N EAT



Inventory Management

Name of Category:

New Category

Add Category

Enter Name of Item:

Enter Category:

Drinks

Enter Price:

Price

Enter Quantity:

Number

Add

Edit

Delete

Reset

Inventory Overview

ItemID	ItemName	ItemCategory	ItemPrice	ItemQuantity
2	Tea	Drinks	0.4	75
4	Strawberry Muffin	Cakes	1.1	25
5	Cheesecake	Deserts	2.2	50

Overview

Refresh

Orders Table

OrderID	OrderDate	OrderAmount
8	24/04/2022	2.2
9	08/05/2022	1.1

Sales Table

Id	name	date	sale
14	ibrahim19061...	19/05/2022	89
20	emp101	... 19/05/2022	0.5

Items Table

ItemID	ItemName	ItemCategory	ItemPrice	ItemQuantity
2	Tea	Drinks	0.4	75
4	Strawberry Muffin	Cakes	1.1	25
5	Cheesecake	Deserts	2.2	50

Avoid waiting in a queue.
Just order online and the food will arrive on
your table at the earliest.

Breakfast

Salads

Bits And Bites

Soups

Return Back

Choose Payment Method

Cash On Pickup


VISA

Your Cart

1 Sweet Corn Soup 0.5 OMR
1 Dal Soup 0.5 OMR

Total1 OMR

Proceed to checkout



AVOID WAITING IN A QUEUE
JUST ORDER ONLINE AND THE FOOD WILL ARRIVE ON
YOUR TABLE AT THE EARLIEST

Breakfast

Salads

Bits And Bites

Soups

Return Back

Choose Payment Method

☒ Cash On Pickup

☐ VISA

On Pickup

Place Order


Your Cart

1 Sweet Corn Soup 0.5 OMR

1 Dal Soup 0.5 OMR

Total1 OMR

Proceed to checkout



Breakfast

Salads

Bits And Bites

Soups

Return Back

Choose Payment Method

☒ Cash On Pickup

☐ VISA

On Pickup

Place Order

Order Successful

Thank you for ordering

Total : 1 OMR


OK

Your Cart

1 Sweet Corn Soup 0.5 OMR
1 Dal Soup 0.5 OMR

Total1 OMR

Proceed to checkout




Hot Coffee

Cold Coffee


Soda

Juice


Return Back




■ Espresso - Cold
0.25 OMR




■ Espresso Macchiato - Cold
0.25 OMR




■ Americano - Cold
0.5 OMR



■ Caffe Latte - Cold
0.5 OMR



■ Cappuccino - Cold
0.75 OMR




■ Caffe Mocha - Cold
0.75 OMR

Your Cart

Total0 OMR

Proceed to checkout



SCHEDULE YOUR ORDER FROM ANY COLLEGE DEVICE
HAVE YOUR MEAL ANYTIME AND ANYWHERE


Hot Coffee

Cold Coffee


Soda

Juice


Return Back




■ Espresso
0.25 OMR




■ Espresso Macchiato
0.25 OMR




■ Americano
0.5 OMR



■ Caffe Latte
0.5 OMR



■ Cappuccino
0.75 OMR



■ Caffe Mocha
0.75 OMR

Your Cart

Total 0 OMR

Proceed to checkout

Sales Management

Name

Date

Sale

Sale ID
22

Add

Remove

Total Sales

17.95

Sales Overview

Id	name	date	sale
22	emp101	5/19/2022	2.25
23	Salim	5/19/2022	8.2
24	Mohammed	5/19/2022	4.25
25	ibrahim190611	5/19/2022	3.25

The application was fully developed and worked efficiently without any errors. All the exceptions were handled accordingly. The program was tested for multiple cases and it was noted that the program handled all the tasks successfully.

Marketing and Commercialization

The application goes by the commercial name of *One*. The *One* is an abbreviation for Order N Eat. In order to promote the application, various marketing strategies are adopted. The primary focus for marketing is given to institutions like universities, colleges, schools and corporates. The tool used for marketing is social media platforms, as it provides a larger audience and reasonable budget for advertisement. The social media platforms also enable the developers to communicate with the institutions by collecting timely feedbacks on the application and when introducing new features for the application. In this way, developers can improve the user experience constantly. Furthermore, another marketing strategy that will be used is to conduct workshops and seminars. This will help to create awareness about the benefits of the product to a large number of institutions. Moreover, an introductory free trial can be arranged for the institutions that are willing to try the application. Thus, the mentioned marketing methods are required to commercialize the application termed as *One*.

Conclusions and Recommendations

In summary, an online management and ordering desktop application for canteens, cafés, and other food establishments was successfully developed using visual basic. The project's ultimate goal was met with success. A market analysis was conducted and the user interface was designed. Login and sign-up system were designed for students and admins separately for security purposes. A system for managing and ordering items was created. A stock management system was implemented and a system for sales management was also developed. An overview form was developed that displayed all the management details. The project's objectives have been met with success. A project introduction was written. Literature reviews comparing previous systems to the new system were conducted. The project's design and implementation were addressed. Discussion of the results was delivered. The project's marketing and commercialization were examined. It is widely acknowledged that adequate training for both workers and management is essential before any meaningful computer-based information management can be implemented in any business. Therefore, the data entry staff should be properly trained in computer data entry techniques. The staff should also understand how to properly maintain computer gear and applications to avoid losing data and overspending.

Future Work

There are many improvements that can be made to this canteen application. One of the possible improvements is adding a notifications tab to review the newly added orders and their details, this will help the staff by making it easier to track the orders received. Moreover, an estimated delivery time could be added for each item, this gives a brief idea about the time the order will take. Another improvement could be adding a delivery system to the application and provide delivery to any place on the campus. Automation of the application could be applied more broadly on the different management system included for the staff menu, as it is only applied for calculating the sales and total amount of sales currently, it can also be added to keep track of the amount sold of each item to find a favorite item which is the most common in students' orders.

Implementing the application for mobile is also a great improvement that is essential these days since most of the students use their mobiles more often than their personal computers. This will also make it possible to use different methods to order, pay, or collect the order such as using QR codes or using SMS messages to receive receipts for orders along emails and notifications in the application itself.

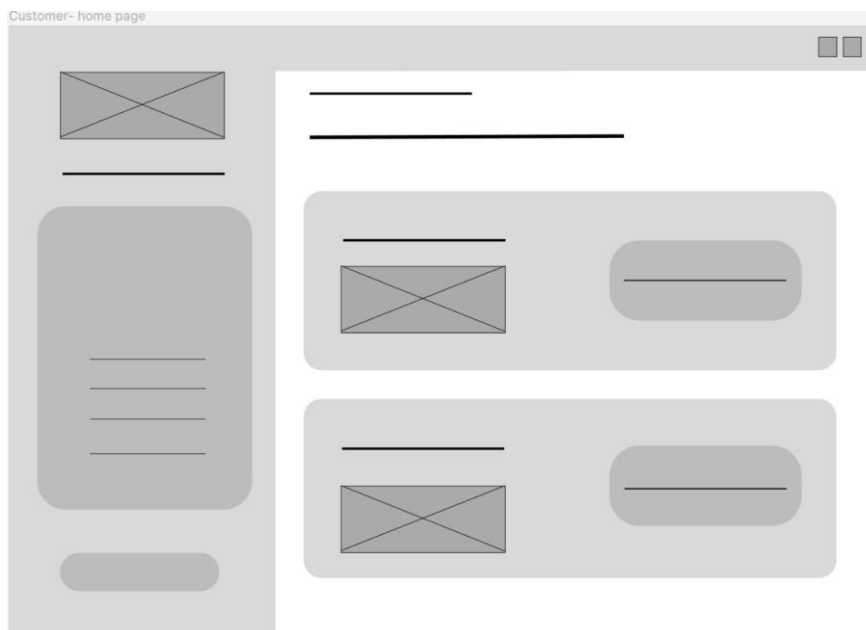
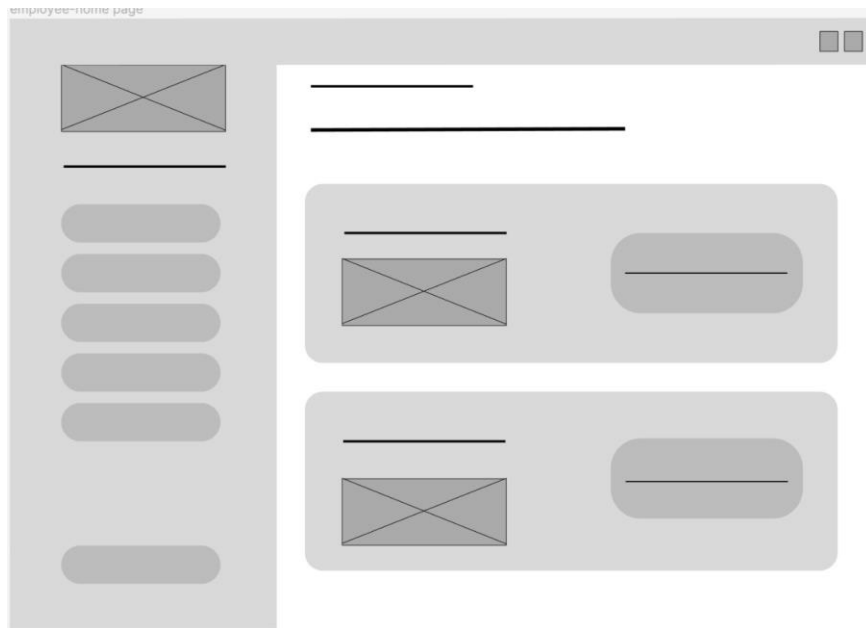
Another improvement that can be made is introducing a pick-up or delivery time for the students to choose from during the hours they are available in campus, they could specify a place on the university to be delivered to at the specified time, or pick up the order personally at that time. These are the improvements that are prioritized currently, and further improvements may be considered in the future.

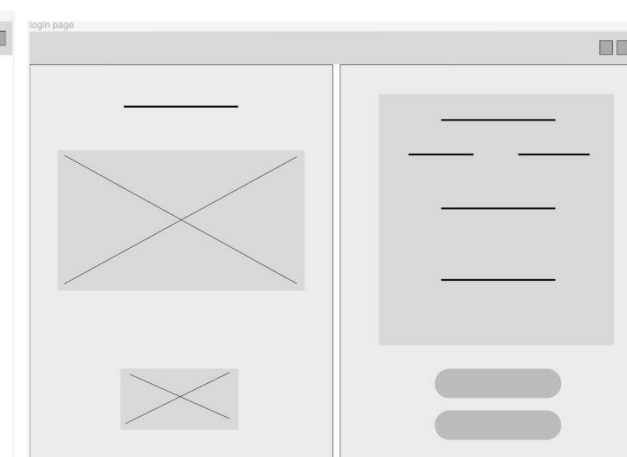
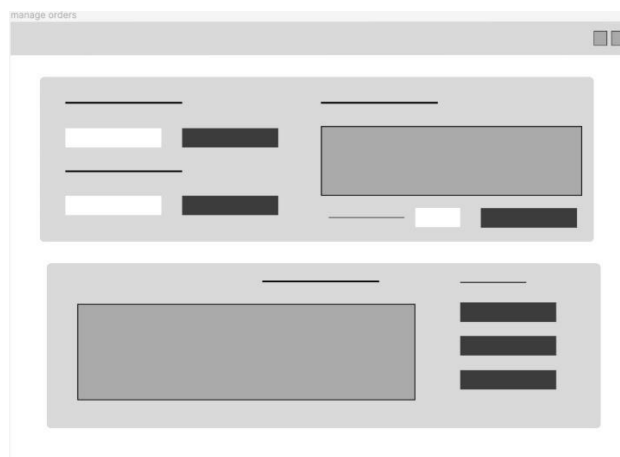
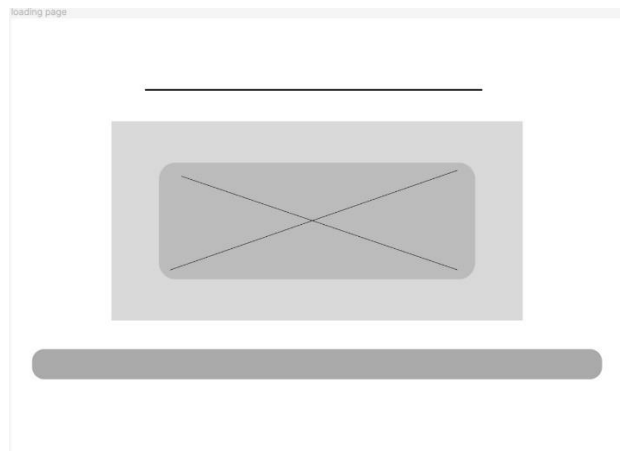
References

- Guru99. 2022. What is VB.Net? Introduction, History, Features, Advantages, Disadvantages. [online] Available at: <<https://www.guru99.com/vb-net-introduction-features.html>> [Accessed 19 May 2022].
- M. A., Kumar, S. R., Nair, S. S. & S, R. K., 2020. Cashless Canteen Management System. *International Journal of Innovative Technology and Exploring Engineering*, 9(7), pp. 579-582.
- Samples.jblearning.com. 2022. [online] Available at: <https://samples.jblearning.com/0763724785/ch02_bronson.pdf> [Accessed 19 May 2022].
- Tutorialspoint.com. 2022. VB.Net Programming Tutorial. [online] Available at: <<https://www.tutorialspoint.com/vb.net/index.htm>> [Accessed 19 May 2022].
- UserGuiding. 2022. 20 Fundamental UX Design Principles a Designer Has to Live By. [online] Available at: <<https://userguiding.com/blog/ux-design-principles/>> [Accessed 19 May 2022].

Appendix

All Wireframes





Source Code

LoginPage

```
Imports System.Data
Imports System.Data.SqlClient
Module getName
    Private strName As String
    Public Property getSName() As String
        Get
            Return strName
        End Get
        Set(ByVal sName As String)
            strName = sName
        End Set
    End Property
End Module
Public Class LoginPage
    Public checkLogin As Boolean = False

    Private Sub LoginSubmitButton_Click(sender As Object, e As EventArgs) Handles
LoginSubmitButton.Click

        If RadioButtonStaff.Checked = True Then

            If TextBoxUserName.Text = "" Or TextBoxPassword.Text = "" Then
                MessageBox.Show("Enter Username and Password")
                checkLogin = False
            Else

                Dim con As SqlConnection = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\smart\OneDrive\Desktop\cante
en-updated-v4\can-update1\canteen\StaffLogin.mdf;Integrated Security=True")
                Dim cmd As SqlCommand = New SqlCommand("select * from StaffLogin
where UserID = '" + TextBoxUserName.Text + "' and Password = '" +
TextBoxPassword.Text + "'", con)
                Dim sda As SqlDataAdapter = New SqlDataAdapter(cmd)
                Dim dt As DataTable = New DataTable()
                sda.Fill(dt)

                If (dt.Rows.Count > 0) Then
                    MessageBox.Show("Successfull Login")
                    Dim page = New CanteenDashboard
                    checkLogin = True
                    page.Show()
                    Me.Hide()
                Else
                    MessageBox.Show("Incorrect Username or Password" & vbCrLf & "If
You Are New Staff Please Create An Account")
                    TextBoxUserName.Text = ""
                    TextBoxPassword.Text = ""
                    checkLogin = False
                End If
            End If
        End If
    End Sub
End Class
```

```

ElseIf RadioButtonStudent.Checked = True Then

    CanteenDashboard.BtnManageOrders.Visible = False
    CanteenDashboard.btnInventory.Visible = False

    If TextBoxUserName.Text = "" Or TextBoxPassword.Text = "" Then
        MessageBox.Show("Enter Username and Password")
        checkLogin = False
    Else

        Dim con As SqlConnection = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\smart\OneDrive\Desktop\cante
en-updated-v4\can-update1\canteen\StudentsLogin.mdf;Integrated Security=True")
        Dim cmd As SqlCommand = New SqlCommand("select * from StudentLogin
where Username = ' + TextBoxUserName.Text + ' and Password = ' +
TextBoxPassword.Text + '", con)
        Dim sda As SqlDataAdapter = New SqlDataAdapter(cmd)
        Dim dt As DataTable = New DataTable()
        sda.Fill(dt)

        If (dt.Rows.Count > 0) Then
            MessageBox.Show("Successfull Login")

            Dim page = New CanteenDashboard
            page.studentBanner.Visible = True
            page.Label2.Text = "Hi There"
            page.BannerName.Text = "Welcome,"
            page.Label4.Text = TextBoxUserName.Text
            page.lblMenu.Visible = False
            page.btnInventory.Visible = False
            page.btnManageSales.Visible = False
            page.btnNotification.Visible = False
            page.btnReportPage.Visible = False
            page.BtnManageOrders.Visible = False
            checkLogin = True
            page.Show()
            Me.Hide()
        Else
            MessageBox.Show("Incorrect Username or Password" & vbCrLf & "If
You Are A New Sudent Please Create An Account")
            TextBoxUserName.Text = ""
            TextBoxPassword.Text = ""
            checkLogin = False
        End If

    End If
Else
    MessageBox.Show("Please Select the Choice - Are you a Student or Staff")
End If

If checkLogin = False Then
    btnCreateAccount.Visible = True
End If
End Sub

```

```

Private Sub btnCreateAccount_Click(sender As Object, e As EventArgs) Handles
btnCreateAccount.Click

    LoginSubmitButton.Visible = False
    LabelUsername.Text = "Enter New Username: "
    LabelPassword.Text = "Enter New Password: "

    If RadioButtonStaff.Checked = True Then

        If TextBoxUserName.Text = "" Or TextBoxPassword.Text = "" Then
            MessageBox.Show("Please Enter the New Username and Password")
            checkLogin = False
        Else

            Dim con As SqlConnection = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\smart\OneDrive\Desktop\cante
en-updated-v4\can-update1\canteen\StaffLogin.mdf;Integrated Security=True")
            con.Open()
            Dim cmd As SqlCommand = New SqlCommand("insert into StaffLogin
values('" & TextBoxUserName.Text & "',''" & TextBoxPassword.Text & "')", con)
            cmd.ExecuteNonQuery()
            con.Close()
            MessageBox.Show("Account Successfully Created")
            LoginSubmitButton.Visible = True
        End If

    ElseIf RadioButtonStudent.Checked = True Then

        If TextBoxUserName.Text = "" Or TextBoxPassword.Text = "" Then
            MessageBox.Show("Please Enter New Username and Password")
            checkLogin = False
        Else

            Dim con As SqlConnection = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\smart\OneDrive\Desktop\cante
en-updated-v4\can-update1\canteen\StudentsLogin.mdf;Integrated Security=True")
            con.Open()
            Dim cmd As SqlCommand = New SqlCommand("insert into StudentLogin
values('" & TextBoxUserName.Text & "',''" & TextBoxPassword.Text & "')", con)
            cmd.ExecuteNonQuery()
            con.Close()
            MessageBox.Show("Account Successfully Created")
            LoginSubmitButton.Visible = True
        End If

    Else
        MessageBox.Show("Please Select the Choice - Are you a Student or Staff")
    End If

End Sub

Private Sub Guna2ControlBox1_Click(sender As Object, e As EventArgs) Handles
Guna2ControlBox1.Click
    Application.Exit()
End Sub

Private Sub TextBoxUserName_TextChanged(sender As Object, e As EventArgs)
Handles TextBoxUserName.TextChanged
    getSName = TextBoxUserName.Text

```

```

End Sub

Private Sub LoginPage_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub
End Class

```

Dashboard

```
Public Class CanteenDashboard
```

```

    Private Sub Guna2Button1_Click(sender As Object, e As EventArgs) Handles
btnSignout.Click
        Me.Hide()
        LoginPage.Show()

```

```
End Sub
```

```

    Private Sub Guna2Button2_Click(sender As Object, e As EventArgs) Handles
BtnManageOrders.Click

        Orders.Show()

```

```
End Sub
```

```

    Private Sub Guna2Button3_Click(sender As Object, e As EventArgs) Handles
btnInventory.Click

        InventoryForm.Show()

```

```
End Sub
```

```

    Private Sub btnOrderFood_Click(sender As Object, e As EventArgs) Handles
btnOrderFood.Click
        OrderFood.Show()

```

```
End Sub
```

```

    Private Sub btnOrderDrinks_Click(sender As Object, e As EventArgs) Handles
btnOrderDrinks.Click
        OrderDrinks.Show()

```

```
End Sub
```

```

    Private Sub Guna2ControlBox1_Click(sender As Object, e As EventArgs) Handles
Guna2ControlBox1.Click
        Application.Exit()

```

```
End Sub
```

```

    Private Sub btnManageSales_Click(sender As Object, e As EventArgs) Handles
btnManageSales.Click

```

```

        SaleManagement.Show()
End Sub

```

```

Private Sub btnReportPage_Click(sender As Object, e As EventArgs) Handles
btnReportPage.Click

    Overview.Show()
End Sub

Private Sub btnNotification_Click(sender As Object, e As EventArgs) Handles
btnNotification.Click
    Developers.Show()
End Sub

Public Sub increase(ListBox As ListBox, Total As TextBox, Price As Label,
Quantity As TextBox, Item As CheckBox)
    Quantity.Text = Val(Quantity.Text) + 1
    Total.Text = Val(Total.Text) + Val(Price.Text) & " OMR"
    ListBox.Items.Remove(Val(Quantity.Text) - 1 & " " & Item.Text & " " &
Val(Price.Text) * Val(Quantity.Text) - Val(Price.Text) & " OMR")
    ListBox.Items.Add(Val(Quantity.Text) & " " & Item.Text & " " &
Val(Price.Text) * Val(Quantity.Text) & " OMR")
End Sub

Public Sub decrease(ListBox As ListBox, Total As TextBox, Price As Label,
Quantity As TextBox, Item As CheckBox)
    If Not Val(Quantity.Text) <= 1 Then
        Quantity.Text = Val(Quantity.Text) - 1
        Total.Text = Val(Total.Text) - Val(Price.Text) & " OMR"
        ListBox.Items.Remove(Val(Quantity.Text) + 1 & " " & Item.Text & " " &
Val(Price.Text) * Val(Quantity.Text) + Val(Price.Text) & " OMR")
        ListBox.Items.Add(Val(Quantity.Text) & " " & Item.Text & " " &
Val(Price.Text) * Val(Quantity.Text) & " OMR")
    End If
End Sub

Public Sub checkingAnItem(Item As CheckBox, Panel As Panel, Total As TextBox,
ListBox As ListBox, Quantity As TextBox, Price As Label)
    If Item.Checked Then
        Panel.Visible = True
        Total.Text = Val(Total.Text) + Val(Price.Text) & " OMR"
        ListBox.Items.Add(Val(Quantity.Text) & " " & Item.Text & " " &
Val(Price.Text) * Val(Quantity.Text) & " OMR")
    Else
        Panel.Visible = False
        Total.Text = Val(Total.Text) - Val(Price.Text) * Val(Quantity.Text) & "
OMR"
        ListBox.Items.Remove(Val(Quantity.Text) & " " & Item.Text & " " &
Val(Price.Text) * Val(Quantity.Text) & " OMR")
        Quantity.Text = "1"
    End If
End Sub

Public Sub QuantityVisibility(Quantity As TextBox, Panel As Panel)
    Quantity.Text = "1"
    Panel.Visible = False
End Sub

Private Sub TimerToday_Tick(sender As Object, e As EventArgs) Handles
TimerToday.Tick

```

```

        lblTime.Text = TimeOfDay.ToString("h:mm:ss tt")
    End Sub

```

```
End Class
```

Check-Out

```

Imports System.Data.SqlClient
Public Class FoodCheckout
    Dim con = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\smart\OneDrive\Desktop\cante
en-updated-v4\can-update1\canteen\Management1.mdf;Integrated Security=True")
    Private Sub FormCheckout_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        ComboBoxVISA.Visible = False
        LabelVisa.Visible = False
        TextBoxVISA.Visible = False
        BtnOrder.Visible = False
    End Sub

    Private Sub RadioButtonVISA_CheckedChanged(sender As Object, e As EventArgs)
Handles RadioButtonVISA.CheckedChanged 'sub for making ComboBoxVISA visible and
ButtonOrder invisible when RadioButtonVISA is checked
        If RadioButtonVISA.Checked Then
            ComboBoxVISA.Visible = True
            BtnOrder.Visible = False
        End If
    End Sub

    Private Sub ComboBoxVISA_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles ComboBoxVISA.SelectedIndexChanged 'when ComboBoxVISA selected index is
changed
        If ComboBoxVISA.SelectedItem = "Online" Then 'if Online is selected, make
LabelVisa, TextBoxVISA and ButtonOrder visible
            LabelVisa.Visible = True
            TextBoxVISA.Visible = True
            BtnOrder.Visible = True
        ElseIf ComboBoxVISA.SelectedItem = "On Pickup" Then 'if On Delivery is
selected, make only Button Order visible
            BtnOrder.Visible = True
            LabelVisa.Visible = False
            TextBoxVISA.Visible = False
        Else 'if nothing is selected, make everything invisible
            LabelVisa.Visible = False
            TextBoxVISA.Visible = False
            BtnOrder.Visible = False
        End If
    End Sub

    Private Sub RadioButtonCash_CheckedChanged(sender As Object, e As EventArgs)
Handles RadioButtonCash.CheckedChanged 'when RadioButtonCash is checked
        If RadioButtonCash.Checked Then 'if it is checked, make ComboBoxVISA
selected item into nothing, make it invisible, and make ButtonOrder visible
            ComboBoxVISA.SelectedItem = Nothing

```

```

        ComboBoxVISA.Visible = False
        BtnOrder.Visible = True
    End If
End Sub

Private Sub BtnOrder_Click(sender As Object, e As EventArgs) Handles
    BtnOrder.Click
    If RadioButtonCash.Checked OrElse ComboBoxVISA.SelectedItem = "On Pickup"
    OrElse TextBoxVISA.Text Like "#### #### #### ####" Then 'check if for every valid
    condition
        MessageBox.Show("Thank you for ordering" & vbCrLf & vbCrLf & "Total : "
        & OrderFood.TextBoxTotal.Text, "Order Successful")
    ElseIf Not TextBoxVISA.Text Like "#### #### #### ####" Then 'else if check
    if credit card number is invalid
        MessageBox.Show("Invalid Card Number" & vbCrLf & vbCrLf & "Card Number
        Should Be 16 Digits With A Space Between Every 4 Digits" & vbCrLf & vbCrLf &
        "Example: 1111 2222 3333 4444", "Order Unsuccessful")
    Else 'else
        MessageBox.Show("Invalid Payment Method", "Order Unsuccessful")
    End If
    Dim query As String = "insert into SalesTable values('" & getSName & "','" &
    DateTime.Today & "','" & Val(OrderFood.TextBoxTotal.Text) & ")"
    con.open()
    Dim cmd = New SqlCommand(query, con)
    cmd.ExecuteNonQuery()
    con.close()
End Sub

Private Sub Guna2ControlBox1_Click(sender As Object, e As EventArgs)
    Me.Close()
End Sub
End Class

```

Order – Students

```

Public Class OrderFood
    Private currentForm As Form = Nothing
    Private Sub openChildForm(childForm As Form) 'sub for Docking the
    childForm(s)/Forms in place of PanelChildForm
        If currentForm Is childForm AndAlso childForm.Visible = True Then 'if the
        same child form is called through a button, make all forms invisible (like close)
            Breakfast.Visible = False
            Salads.Visible = False
            BitAndBites.Visible = False
            Soups.Visible = False
            FoodCheckout.Visible = False
        Else 'if not, put the form that is called through a button in PanelChildForm
            currentForm = childForm
            childForm.TopLevel = False
            childForm.FormBorderStyle = FormBorderStyle.None
            childForm.Dock = DockStyle.Fill
            PanelChildForm.Controls.Add(childForm)
            PanelChildForm.Tag = childForm
            childForm.BringToFront()
        End If
    End Sub
End Class

```

```

        childForm.Show()
    End If
End Sub

Private Sub changeButtonColor(button As Button) 'sub for changing the color of
button when pressed
    If button.BackColor = Color.FromArgb(25, 28, 37) Then 'if the button that is
pressed was the last button pressed beforehand, change it back to the normal color
        button.BackColor = Color.FromArgb(35, 38, 47)
    Else 'if not, change all colors to normal then make the pressed button light
gray
        ButtonBreakfast.BackColor = Color.FromArgb(35, 38, 47)
        ButtonSalads.BackColor = Color.FromArgb(35, 38, 47)
        ButtonBits.BackColor = Color.FromArgb(35, 38, 47)
        ButtonSoups.BackColor = Color.FromArgb(35, 38, 47)
        ButtonCheckout.BackColor = Color.FromArgb(35, 38, 47)
        button.BackColor = Color.FromArgb(25, 28, 37)
    End If
End Sub

Private Sub ButtonChicken_Click(sender As Object, e As EventArgs) Handles
ButtonBreakfast.Click
    openChildForm(Breakfast) 'calling openChildForm sub for FormChicken when
ButtonChicken is pressed to insert the form FormChicken in the place of
PanelChildForm
    changeButtonColor(ButtonBreakfast) 'calling openChildForm sub for
FormChicken when ButtonChicken is pressed
End Sub

Private Sub ButtonMcsaver_Click(sender As Object, e As EventArgs) Handles
ButtonSalads.Click
    openChildForm(Salads)
    changeButtonColor(ButtonSalads)
End Sub

Private Sub ButtonDesserts_Click(sender As Object, e As EventArgs) Handles
ButtonBits.Click
    openChildForm(BitAndBites)
    changeButtonColor(ButtonBits)
End Sub

Private Sub ButtonCold_Click(sender As Object, e As EventArgs) Handles
ButtonSoups.Click
    openChildForm(Soups)
    changeButtonColor(ButtonSoups)
End Sub

Private Sub FormMcDonalds_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
    TextBoxTotal.Text = "0 OMR" 'default value of TextBoxTotal
End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles
ButtonCheckout.Click
    If TextBoxTotal.Text = "0 OMR" Then 'if TextBoxTotal is empty
        MessageBox.Show("Your Cart Is Empty", "Empty Cart") 'print message that
the cart is empty
    Else 'if not

```



```

        openChildForm(FoodCheckout) 'open FormCheckout in place of
PanelChildForm
        changeButtonColor(ButtonCheckout)
    End If
End Sub

Private Sub BtnClose_Click(sender As Object, e As EventArgs) Handles
BtnClose.Click
    Me.Close()
End Sub
End Class

```

Order -Staff

```

Imports System.Data.SqlClient
Imports System.Windows.Forms.Form
Public Class Orders

    Dim con = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\smart\OneDrive\Desktop\cante
en-updated-v4\can-update1\canteen\Management1.mdf;Integrated Security=True")

    Private Sub displayItem()
        con.open()
        Dim query = "select * from ItemTable"
        Dim cmd = New SqlCommand(query, con)
        Dim adapter = New SqlDataAdapter(cmd)
        Dim builder = New SqlCommandBuilder(adapter)
        Dim ds = New DataSet()
        adapter.Fill(ds)
        ItemsDGV.DataSource = ds.Tables(0)
        con.close()
    End Sub

    Private Sub CheckCategory()
        con.open()
        Dim query = "select * from ItemTable where ItemCategory='" &
comboBoxtypes.SelectedValue.ToString() & "'"
        Dim cmd = New SqlCommand(query, con)
        Dim adapter = New SqlDataAdapter(cmd)
        Dim builder = New SqlCommandBuilder(adapter)
        Dim ds = New DataSet()
        adapter.Fill(ds)
        ItemsDGV.DataSource = ds.Tables(0)
        con.close()
    End Sub

    Private Sub fillCategory()
        con.open()
        Dim cmd = New SqlCommand("select * from CategoryTable", con)
        Dim adapter = New SqlDataAdapter(cmd)
        Dim table = New DataTable()
        adapter.Fill(table)
        comboBoxtypes.DataSource = table
        comboBoxtypes.DisplayMember = "CategoryName"
        comboBoxtypes.ValueMember = "CategoryName"
    End Sub
End Class

```

```

        con.close()
    End Sub

    Private Sub Orders_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        displayItem()
        fillCategory()
    End Sub

    Private Sub comboboxtypes_SelectionChangeCommitted(sender As Object, e As
EventArgs) Handles comboboxtypes.SelectionChangeCommitted
        CheckCategory()
    End Sub

    Dim productsName As String
    Dim i = 0, price, quantity
    Dim grdTotal = 0

    Private Sub Guna2Button4_Click(sender As Object, e As EventArgs) Handles
Guna2Button4.Click
        If key = 0 Then
            MessageBox.Show("Select A Item")
        ElseIf quantity > stock Then
            MessageBox.Show("Not Enough Stock Available")
        Else
            Dim rownum As Integer = OrderDGV.Rows.Add()
            Dim total = Convert.ToInt32(txtQuantity.Text) * price
            i = i + 1
            OrderDGV.Rows.Item(rownum).Cells("Column1").Value = i
            OrderDGV.Rows.Item(rownum).Cells("Column2").Value = productsName
            OrderDGV.Rows.Item(rownum).Cells("Column3").Value = price
            OrderDGV.Rows.Item(rownum).Cells("Column4").Value = txtQuantity.Text
            OrderDGV.Rows.Item(rownum).Cells("Column5").Value = total
            grdTotal = grdTotal + total
            LabelTotal.Text = "OMR " + Convert.ToString(grdTotal)
            txtQuantity.Text = ""
            key = 0

        End If
    End Sub

    Private Sub Guna2Button1_Click(sender As Object, e As EventArgs) Handles
Guna2Button1.Click
        PrintPreviewDialog1.Show()

    End Sub

    Private Sub addInvoice()
        con.open()
        Dim query = "insert into OrderTable values('" &
DateTime.Today.Date.ToString("D") & "', '" & grdTotal & "')"
        Dim cmd As SqlCommand
        cmd = New SqlCommand(query, con)
        cmd.ExecuteNonQuery()
        MessageBox.Show("Invoice Added Successfully")
        con.close()
    End Sub

```

```

    Private Sub PrintDocument1_PrintPage(sender As Object, e As
Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
    e.Graphics.DrawString("College Canteen", New Font("Times New Roman", 22),
Brushes.Teal, 335, 35)
    e.Graphics.DrawString("Invoice", New Font("Times New Roman", 18),
Brushes.Teal, 390, 60)
    Dim bm As New Bitmap(Me.OrderDGV.Width, Me.OrderDGV.Height)
    OrderDGV.DrawToBitmap(bm, New Rectangle(0, 0, Me.OrderDGV.Width,
Me.OrderDGV.Height))
    e.Graphics.DrawImage(bm, 0, 90)
    e.Graphics.DrawString("Total Amount " + grdTotal.ToString(), New Font("Times
New Roman", 15), Brushes.Teal, 320, 580)
    e.Graphics.DrawString("-----***** Thanks For Ordering *****-----", New
Font("Times New Roman", 15), Brushes.Teal, 200, 600)

    End Sub

    Private Sub Guna2Button6_Click(sender As Object, e As EventArgs) Handles
Guna2Button6.Click
    addInvoice()
    End Sub

    Private Sub Guna2Button3_Click(sender As Object, e As EventArgs) Handles
Guna2Button3.Click
    Dim win = New OrderList
    win.Show()
    Me.Hide()
    End Sub

    Private Sub Guna2ControlBox1_Click(sender As Object, e As EventArgs) Handles
Guna2ControlBox1.Click
    Me.Close()

    End Sub

    Dim key = 0, stock
    Private Sub ItemsDGV_CellMouseClicked(sender As Object, e As
DataGridViewCellMouseEventArgs) Handles ItemsDGV.CellMouseClicked
    Dim row As DataGridViewRow = ItemsDGV.Rows(e.RowIndex)
    productsName = row.Cells(1).Value.ToString

    If productsName = "" Then
        key = 0
        stock = 0
    Else
        key = Convert.ToInt32(row.Cells(0).Value.ToString)
        stock = Convert.ToInt32(row.Cells(4).Value.ToString)
        price = Convert.ToSingle(row.Cells(3).Value.ToString)
    End If
    End Sub
End Class

```

Sale Management

```
Imports System.Data.SqlClient
Public Class SaleManagement
    Dim con = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\smart\OneDrive\Desktop\cante
en-updated-v4\can-update1\canteen\Management1.mdf;Integrated Security=True")
    Private Sub fillSaleID()
        con.open()
        Dim cmd = New SqlCommand("select * from SalesTable", con)
        Dim adapter = New SqlDataAdapter(cmd)
        Dim table = New DataTable()
        adapter.Fill(table)
        cbxsales.DataSource = table
        cbxsales.DisplayMember = "Id"
        cbxsales.ValueMember = "Id"
        con.close()
    End Sub
    Private Sub fillSaleTotal()
        con.open()
        Dim cmd = New SqlCommand("select sum(sale) from SalesTable", con)
        Dim sum As Double = cmd.ExecuteScalar()
        txttotal.Text = sum
        con.close()
    End Sub
    Private Sub displaySales()
        con.open()
        Dim query = "select * from SalesTable"
        Dim cmd = New SqlCommand(query, con)
        Dim adapter = New SqlDataAdapter(cmd)
        Dim builder = New SqlCommandBuilder(adapter)
        Dim ds = New DataSet()
        adapter.Fill(ds)
        SalesDGV.DataSource = ds.Tables(0)
        con.close()
    End Sub
    Private Sub SaleManagement_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        txtdatesales.Text = Date.Today
        displaySales()
        fillSaleID()
        fillSaleTotal()
    End Sub
    Private Sub btnsaleadd_Click(sender As Object, e As EventArgs) Handles
btnsaleadd.Click
        If Not IsNumeric(txtsale.Text) Then
            MessageBox.Show("Invalid Sale input.")
        Else
            con.open()
            Dim query = "insert into SalesTable values('" & txtnamesales.Text &
"', '" & DateTime.Today & "', '" & txtsale.Text & "')"
            Dim cmd = New SqlCommand(query, con)
            cmd.ExecuteNonQuery()
            con.close()
            displaySales()
            fillSaleID()
            fillSaleTotal()
        End If
    End Sub
End Class
```

```

End Sub

Private Sub btnsaleremove_Click(sender As Object, e As EventArgs) Handles
btnsaleremove.Click
    con.open()
    Dim query = "delete from SalesTable where Id = '" & cbxsales.SelectedValue &
    """"
    Dim cmd = New SqlCommand(query, con)
    cmd.ExecuteNonQuery()
    con.close()
    displaySales()
    fillSaleID()
    fillSaleTotal()
End Sub
End Class

```

Inventory

```
Imports System.Data.SqlClient
```

```
Public Class InventoryForm
```

```

    Dim con = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\smart\OneDrive\Desktop\cante
en-updated-v4\can-update1\canteen\Management1.mdf;Integrated Security=True")

```

```

    Private Sub Guna2ControlBox1_Click(sender As Object, e As EventArgs) Handles
Guna2ControlBox1.Click
        Me.Close()
    End Sub

```

```

    Private Sub Guna2Button1_Click(sender As Object, e As EventArgs) Handles
Guna2Button1.Click
        If CategoryAddToTable.Text = "" Then
            MessageBox.Show("Enter the Category")
        Else
            con.open()
            Dim query = "insert into CategoryTable values('" &
CategoryAddToTable.Text & "')"
            Dim cmd As SqlCommand
            cmd = New SqlCommand(query, con)
            cmd.ExecuteNonQuery()
            MessageBox.Show("Category Added")
            con.close()
            CategoryAddToTable.Text = ""
            fillCategory()
        End If
    End Sub

```

```
End Sub
```

```

Private Sub reset()
    comboboxCategory.SelectedIndex = 0
    txtboxItemName.Text = ""
    txtboxItemPrice.Text = ""
    txtboxQuantity.Text = ""
End Sub

```

```

Private Sub fillCategory()
    con.open()
    Dim cmd = New SqlCommand("select * from CategoryTable", con)
    Dim adapter = New SqlDataAdapter(cmd)
    Dim table = New DataTable()
    adapter.Fill(table)
    comboboxCategory.DataSource = table
    comboboxCategory.DisplayMember = "CategoryName"
    comboboxCategory.ValueMember = "CategoryName"
    con.close()
End Sub

Private Sub ResetItems_Click(sender As Object, e As EventArgs) Handles
ResetItems.Click
    reset()
End Sub

Private Sub Form3_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    fillCategory()
    displayItem()
End Sub

Private Sub displayItem()
    con.open()
    Dim query = "select * from ItemTable"
    Dim cmd = New SqlCommand(query, con)
    Dim adapter = New SqlDataAdapter(cmd)
    Dim builder = New SqlCommandBuilder(adapter)
    Dim ds = New DataSet()
    adapter.Fill(ds)
    ItemsDGV.DataSource = ds.Tables(0)
    con.close()
End Sub

Private Sub AddIetm_Click(sender As Object, e As EventArgs) Handles
AddIetm.Click
    If comboboxCategory.SelectedIndex = -1 Or txtboxItemName.Text = "" Or
txtboxItemPrice.Text = "" Or txtboxQuantity.Text = "" Then
        MessageBox.Show("Please Enter All the Fields")
    Else
        con.open()
        Dim query = "insert into ItemTable values('" & txtboxItemName.Text &
"', '" & comboboxCategory.SelectedValue.ToString() & "', '" & txtboxItemPrice.Text &
"', '" & txtboxQuantity.Text & "')"
        Dim cmd As SqlCommand
        cmd = New SqlCommand(query, con)
        cmd.ExecuteNonQuery()
        MessageBox.Show("Item Added into Inventory")
        con.close()
        reset()
        displayItem()
        'CategoryAddToTable.Text = ""
        'fillCategory()
    End If
End Sub

Dim key = 0

```

```

Private Sub ItemsDGV_CellMouseClicked(sender As Object, e As
DataGridViewCellMouseEventArgs) Handles ItemsDGV.CellMouseClicked
    Dim row As DataGridViewRow = ItemsDGV.Rows(e.RowIndex)
    txtboxItemName.Text = row.Cells(1).Value.ToString
    comboboxCategory.SelectedValue = row.Cells(2).Value.ToString
    txtboxItemPrice.Text = row.Cells(3).Value.ToString
    txtboxQuantity.Text = row.Cells(4).Value.ToString

    If txtboxItemName.Text = "" Then
        key = 0
    Else
        key = Convert.ToInt32(row.Cells(0).Value.ToString)
    End If
End Sub

Private Sub DeleteItem_Click(sender As Object, e As EventArgs) Handles
DeleteItem.Click
    If key = 0 Then
        MessageBox.Show("Please Select The Field To Be Deleted")
    Else
        con.open()
        Dim query = "delete from ItemTable where ItemID=" & key & ""
        Dim cmd As SqlCommand
        cmd = New SqlCommand(query, con)
        cmd.ExecuteNonQuery()
        MessageBox.Show("Item Deleted Successfully")
        con.close()
        reset()
        displayItem()
    End If
End Sub

Private Sub EditItem_Click(sender As Object, e As EventArgs) Handles
EditItem.Click
    If comboboxCategory.SelectedIndex = -1 Or txtboxItemName.Text = "" Or
txtboxItemPrice.Text = "" Or txtboxQuantity.Text = "" Then
        MessageBox.Show("Please Enter All the Fields")
    Else
        con.open()
        Dim query = "update ItemTable set ItemName='" & txtboxItemName.Text &
"',ItemCategory='" & comboboxCategory.SelectedValue.ToString() & "',ItemPrice=" &
txtboxItemPrice.Text & ",ItemQuantity=" & txtboxQuantity.Text & " where ItemID=" &
key & ""
        Dim cmd As SqlCommand
        cmd = New SqlCommand(query, con)
        cmd.ExecuteNonQuery()
        MessageBox.Show("Item Updated")
        con.close()
        reset()
        displayItem()
        'CategoryAddToTable.Text = ""
        'fillCategory()
    End If
End Sub
End Class

```

Food Sample 1

```
Public Class Juice
    Private Sub ButtonQ1Inc_Click(sender As Object, e As EventArgs) Handles
ButtonQ1Inc.Click
        CanteenDashboard.increase(OrderDrinks.ListBoxItems,
OrderDrinks.TextBoxTotal, PriceApple, TextBoxQ1, CheckBoxApple)
    End Sub

    Private Sub ButtonQ1Dec_Click(sender As Object, e As EventArgs) Handles
ButtonQ1Dec.Click
        CanteenDashboard.decrease(OrderDrinks.ListBoxItems,
OrderDrinks.TextBoxTotal, PriceApple, TextBoxQ1, CheckBoxApple)
    End Sub

    Private Sub CheckBoxEspresso_CheckedChanged(sender As Object, e As EventArgs)
Handles CheckBoxApple.CheckedChanged
        CanteenDashboard.checkingAnItem(CheckBoxApple, Panel1,
OrderDrinks.TextBoxTotal, OrderDrinks.ListBoxItems, TextBoxQ1, PriceApple)
    End Sub

    Private Sub ButtonQ2Inc_Click(sender As Object, e As EventArgs) Handles
ButtonQ2Inc.Click
        CanteenDashboard.increase(OrderDrinks.ListBoxItems,
OrderDrinks.TextBoxTotal, PriceOrange, TextBoxQ2, CheckBoxOrange)
    End Sub

    Private Sub ButtonQ2Dec_Click(sender As Object, e As EventArgs) Handles
ButtonQ2Dec.Click
        CanteenDashboard.decrease(OrderDrinks.ListBoxItems,
OrderDrinks.TextBoxTotal, PriceOrange, TextBoxQ2, CheckBoxOrange)
    End Sub

    Private Sub CheckBoxMacchiato_CheckedChanged(sender As Object, e As EventArgs)
Handles CheckBoxOrange.CheckedChanged
        CanteenDashboard.checkingAnItem(CheckBoxOrange, Panel2,
OrderDrinks.TextBoxTotal, OrderDrinks.ListBoxItems, TextBoxQ2, PriceOrange)
    End Sub

    Private Sub ButtonQ3Inc_Click(sender As Object, e As EventArgs) Handles
ButtonQ3Inc.Click
        CanteenDashboard.increase(OrderDrinks.ListBoxItems,
OrderDrinks.TextBoxTotal, PriceWatermelon, TextBoxQ3, CheckBoxWatermelon)
    End Sub

    Private Sub ButtonQ3Dec_Click(sender As Object, e As EventArgs) Handles
ButtonQ3Dec.Click
        CanteenDashboard.decrease(OrderDrinks.ListBoxItems,
OrderDrinks.TextBoxTotal, PriceWatermelon, TextBoxQ3, CheckBoxWatermelon)
    End Sub
End Class
```



```
Private Sub CheckBoxAmericano_CheckedChanged(sender As Object, e As EventArgs) Handles CheckBoxWatermelon.CheckedChanged
    CanteenDashboard.checkingAnItem(CheckBoxWatermelon, Panel3,
    OrderDrinks.TextBoxTotal, OrderDrinks.ListBoxItems, TextBoxQ3, PriceWatermelon)
End Sub
```

```
Private Sub ButtonQ4Inc_Click(sender As Object, e As EventArgs) Handles ButtonQ4Inc.Click
    CanteenDashboard.increase(OrderDrinks.ListBoxItems,
    OrderDrinks.TextBoxTotal, PriceLemon, TextBoxQ4, CheckBoxLemon)
End Sub
```

```
Private Sub ButtonQ4Dec_Click(sender As Object, e As EventArgs) Handles ButtonQ4Dec.Click
    CanteenDashboard.decrease(OrderDrinks.ListBoxItems,
    OrderDrinks.TextBoxTotal, PriceLemon, TextBoxQ4, CheckBoxLemon)
End Sub
```

```
Private Sub CheckBoxlatte_CheckedChanged(sender As Object, e As EventArgs) Handles CheckBoxLemon.CheckedChanged
    CanteenDashboard.checkingAnItem(CheckBoxLemon, Panel4,
    OrderDrinks.TextBoxTotal, OrderDrinks.ListBoxItems, TextBoxQ4, PriceLemon)
End Sub
```

```
Private Sub ButtonQ5Inc_Click(sender As Object, e As EventArgs) Handles ButtonQ5Inc.Click
    CanteenDashboard.increase(OrderDrinks.ListBoxItems,
    OrderDrinks.TextBoxTotal, PricePineapple, TextBoxQ5, CheckBoxPineapple)
End Sub
```

```
Private Sub ButtonQ5Dec_Click(sender As Object, e As EventArgs) Handles ButtonQ5Dec.Click
    CanteenDashboard.decrease(OrderDrinks.ListBoxItems,
    OrderDrinks.TextBoxTotal, PricePineapple, TextBoxQ5, CheckBoxPineapple)
End Sub
```

```
Private Sub CheckBoxCappuccino_CheckedChanged(sender As Object, e As EventArgs) Handles CheckBoxPineapple.CheckedChanged
    CanteenDashboard.checkingAnItem(CheckBoxPineapple, Panel5,
    OrderDrinks.TextBoxTotal, OrderDrinks.ListBoxItems, TextBoxQ5, PricePineapple)
End Sub
```

```
Private Sub ButtonQ6Inc_Click(sender As Object, e As EventArgs) Handles ButtonQ6Inc.Click
    CanteenDashboard.increase(OrderDrinks.ListBoxItems,
    OrderDrinks.TextBoxTotal, PricePear, TextBoxQ6, CheckBoxPear)
End Sub
```

```
Private Sub ButtonQ6Dec_Click(sender As Object, e As EventArgs) Handles ButtonQ6Dec.Click
    CanteenDashboard.decrease(OrderDrinks.ListBoxItems,
    OrderDrinks.TextBoxTotal, PricePear, TextBoxQ6, CheckBoxPear)
```

```

End Sub

Private Sub CheckBoxMocha_CheckedChanged(sender As Object, e As EventArgs)
Handles CheckBoxPear.CheckedChanged
    CanteenDashboard.checkingAnItem(CheckBoxPear, Panel6,
OrderDrinks.TextBoxTotal, OrderDrinks.ListBoxItems, TextBoxQ6, PricePear)
End Sub

Private Sub HotCoffee_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    CanteenDashboard.QuantityVisibility(TextBoxQ1, Panel1)
    CanteenDashboard.QuantityVisibility(TextBoxQ2, Panel2)
    CanteenDashboard.QuantityVisibility(TextBoxQ3, Panel3)
    CanteenDashboard.QuantityVisibility(TextBoxQ4, Panel4)
    CanteenDashboard.QuantityVisibility(TextBoxQ5, Panel5)
    CanteenDashboard.QuantityVisibility(TextBoxQ6, Panel6)
End Sub

Private Sub Guna2ControlBox1_Click(sender As Object, e As EventArgs)
    Me.Close()
End Sub
End Class

```

Overview

```

Imports System.Data.SqlClient
Public Class Overview
    Dim con = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\smart\OneDrive\Desktop\cante
en-updated-v4\can-update1\canteen\Management1.mdf;Integrated Security=True")

Private Sub displaySalesTable()
    con.open()
    Dim query = "select * from SalesTable"
    Dim cmd = New SqlCommand(query, con)
    Dim adapter = New SqlDataAdapter(cmd)
    Dim builder = New SqlCommandBuilder(adapter)
    Dim ds = New DataSet()
    adapter.Fill(ds)
    gridSales.DataSource = ds.Tables(0)
    con.close()
End Sub
Private Sub displayItemsTable()
    con.open()
    Dim query = "select * from ItemTable"
    Dim cmd = New SqlCommand(query, con)
    Dim adapter = New SqlDataAdapter(cmd)
    Dim builder = New SqlCommandBuilder(adapter)
    Dim ds = New DataSet()
    adapter.Fill(ds)
    gridItems.DataSource = ds.Tables(0)
    con.close()

```

```

End Sub
Private Sub displayOrderTable()
    con.open()
    Dim query = "select * from OrderTable"
    Dim cmd = New SqlCommand(query, con)
    Dim adapter = New SqlDataAdapter(cmd)
    Dim builder = New SqlCommandBuilder(adapter)
    Dim ds = New DataSet()
    adapter.Fill(ds)
    gridOrders.DataSource = ds.Tables(0)
    con.close()
End Sub
Private Sub Overview_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    displaySalesTable()
    displayItemsTable()
    displayOrderTable()
End Sub

Private Sub btnrefresh_Click(sender As Object, e As EventArgs) Handles
btnrefresh.Click
    displaySalesTable()
    displayItemsTable()
    displayOrderTable()
End Sub
End Class

```