TECHNICAL DOCUMENTATION

TABLE OF CONTENTS

Hardware/Software Requirements ... 1

Hardware Requirements ... 1

Software Requirements ... 1

Record, File and Data Structures Used ... 2

1. Category_Table ... 2
2. Items_Table ... 3
3. CustomerBill_Table ... 4
4. BillDetails_Table ... 5
5. Userlogin ... 6

Data Dictionary ... 7

1. Category_Table ... 7

2. Items_Table ... 7

3. CustomerBill_Table ... 8

4. BillDetails_Table ... 8

UserLogin ... 9

Menu ... 10

For Receptionist ... 10

For Administrator ... 11

Entity- Relationship (E-R) Diagram ... 12

Relationships ... 12

Data Flow Diagram ... 13

Flowcharts ... 14

Log-in ... 14

Food Items ... 16

Bill(Adding Food Item) ... 20

Reporting ... 21

Maintenance ... 23

Program Listing ... 25

Main Form(MDI Form) ... 25

Log-in Screen ... 27

Password Changer Screen 27

Modify Food Items Screen 29

Add new Item Screen 30

Edit Food Items Screen 31

Add Category 33

Select Table Screen 34

Bill Screen 35

Add Item to bill 36

Calculate Screen 38

Daily Sales Report 39

Monthly Sales Report 40

Daily Sales Report 41

Monthly Sales Report 42

Bill Receipt 44

Validator 45

Possible Adaptive Maintenance 46

Code Listing 46

Main 46

Food Items 49

Bill 55

Password 62

Reports 64

Hardware/Software Requirements

Hardware Requirements

The recommended hardware devices and their capacities required for the billing software are:

- LCD Monitor: It is the main output device which displays the result of processing and the data that he user has entered using input device. LCD Monitor is recommended because it is common and cheap nowadays and is ergonomically beneficial compared to CRT.

- Input Devices: Standard QWERTY Keyboard is required to enter the data to the software and Mouse to select the menu items in the software.

- Processor: 1.6 Giga Hertz or higher which can smoothly run the Operating System(Windows 7 preferred)

- Hard-Disk: At least 20 GB(Giga Byte) to store the Operating system and backups of the software(databases)

- CD R/W: The software will be installed from CD. Writable CD Drive is required for the backups of the software data files to be performed.

- Portable USB Drive: Pen drive or USB Drive for the backing up the data files with greater flexibility.

- Printer: A Printer is required to print Customer Invoices and other Reports. Dot Matrix printer will be suitable to print out the customer bill and Laser Printer will be suitable for Reports.

Software Requirements

- Microsoft Operating System (O.S): As the software is created using Microsoft technology (Microsoft Visual Basic), any version of Windows released after and including Windows XP (Service Pack 3) will work. Windows 7 is recommended.

- Microsoft Access 2007: Database is implemented using Microsoft Access 2007 so it is required.

- .NET Framework 4.0: It is required to convert the Compiled program (which is in Microsoft Intermediate language) into Machine Language so that the program can execute.

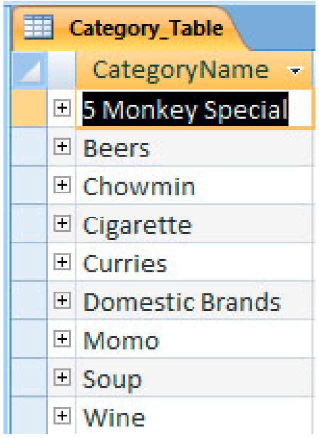
Record, File and Data Structures Used

The tables of the database which is used to stores the record required and generated by the program are shown below. The purpose of the fields are shown in data dictionary

1. Category_Table

This table stores the Category Name under which food items fall. Each one is unique

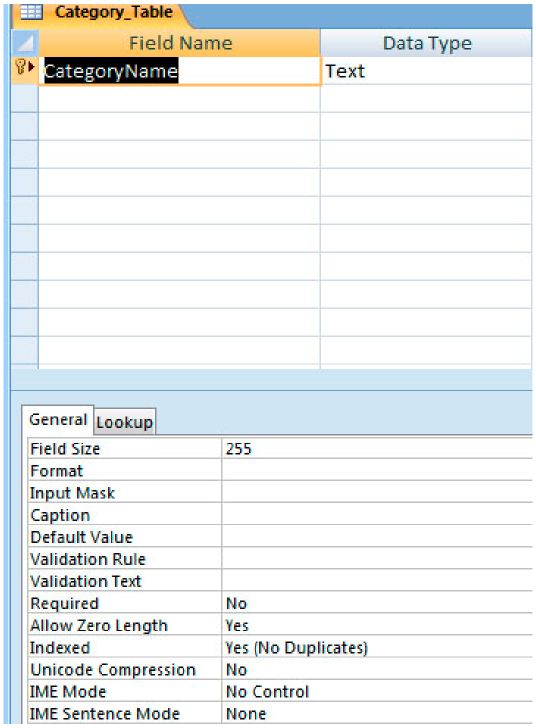
Datasheet View



Category_Table

CategoryName
5 Monkey Special
Beers
Chowmin
Cigarette
Curries
Domestic Brands
Momo
Soup
Wine

Design View



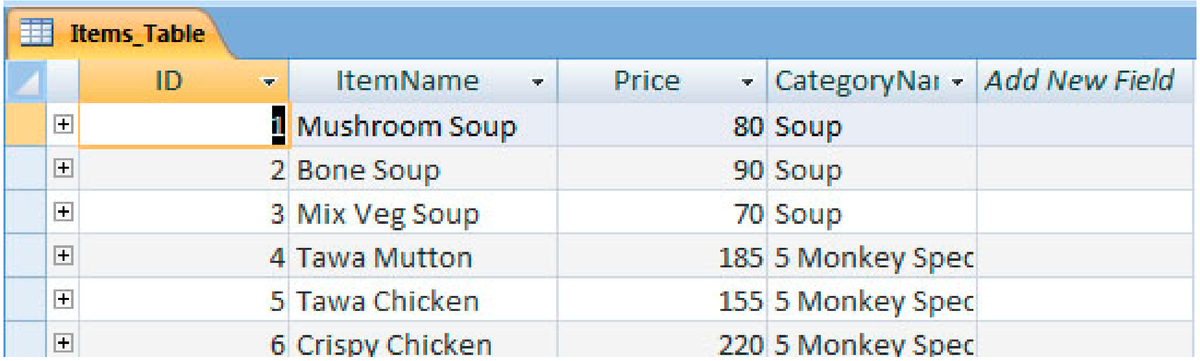
Field Name	Data Type
CategoryName	Text

Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None

2. Items_Table

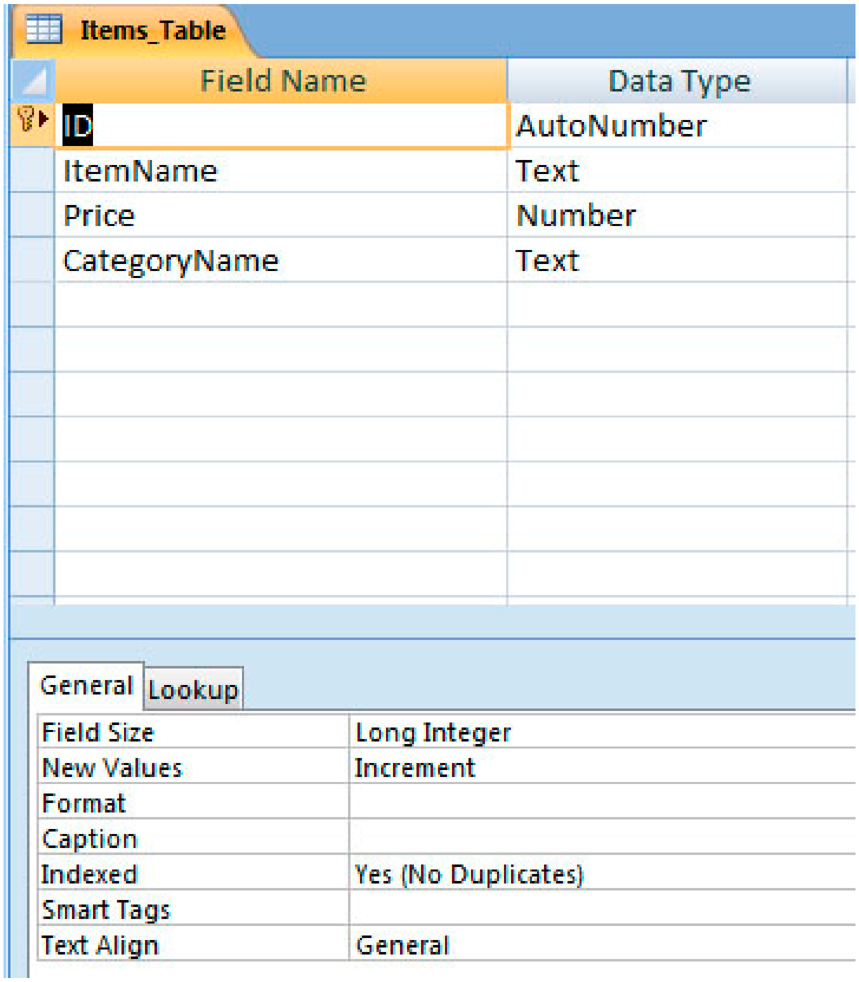
This table stores the details of each and every food items

Datasheet View



ID	ItemName	Price	CategoryNar	Add New Field
1	Mushroom Soup	80	Soup	
2	Bone Soup	90	Soup	
3	Mix Veg Soup	70	Soup	
4	Tawa Mutton	185	5 Monkey Spec	
5	Tawa Chicken	155	5 Monkey Spec	
6	Crispy Chicken	220	5 Monkey Spec	

Design View



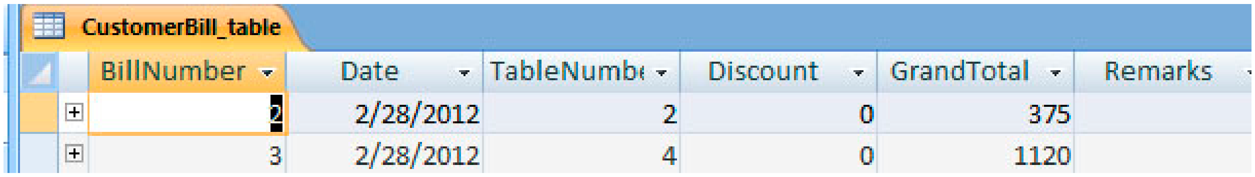
Field Name	Data Type
ID	AutoNumber
ItemName	Text
Price	Number
CategoryName	Text

Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)
Smart Tags	
Text Align	General

3. CustomerBill_Table

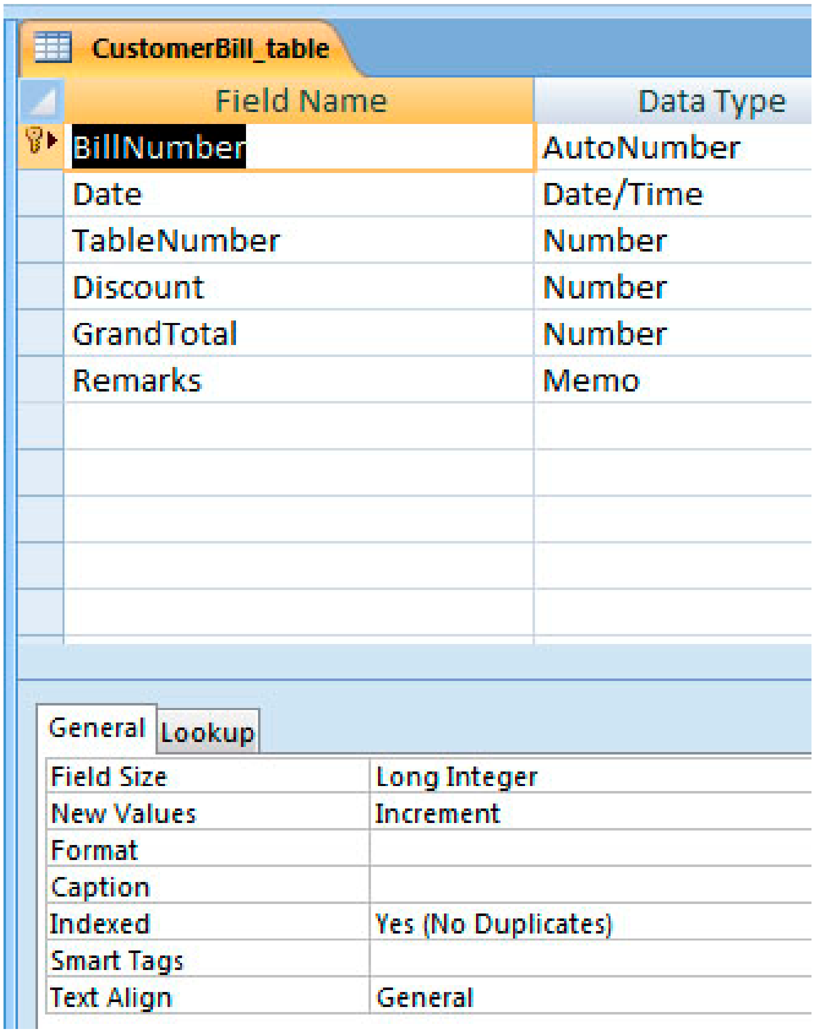
This table stores the details of every receipt / bill for which the customer makes the payment.

Datasheet View



BillNumber	Date	TableNumb	Discount	GrandTotal	Remarks
2	2/28/2012	2	0	375	
3	2/28/2012	4	0	1120	

Design View



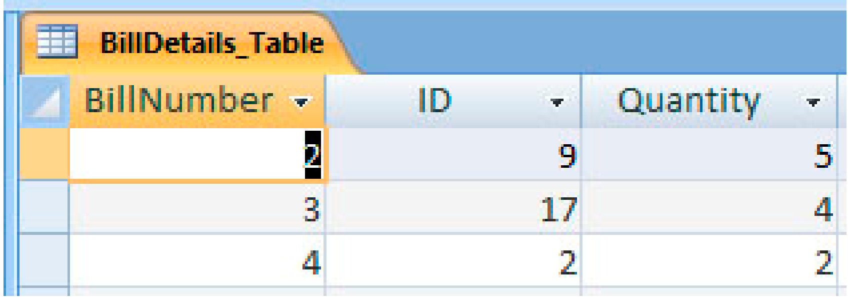
Field Name	Data Type
BillNumber	AutoNumber
Date	Date/Time
TableNumber	Number
Discount	Number
GrandTotal	Number
Remarks	Memo

Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)
Smart Tags	
Text Align	General

4. BillDetails_Table

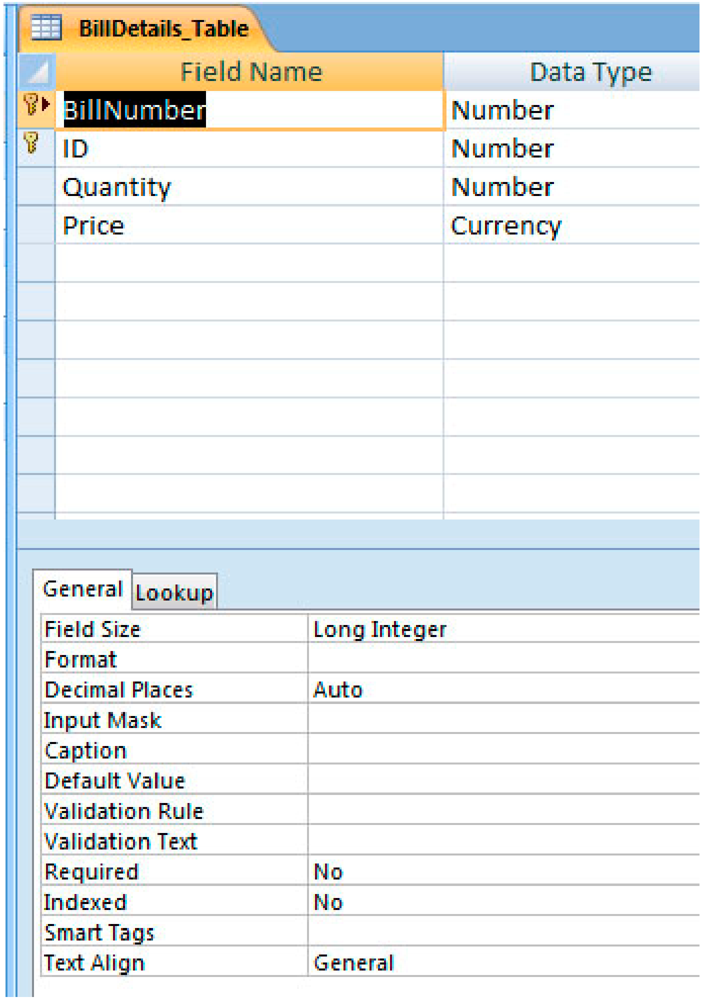
This table stores the details of food item ordered by customer of particular bill number.

Datasheet View



BillNumber	ID	Quantity
2	9	5
3	17	4
4	2	2

Design View



BillDetails_Table

Field Name	Data Type
BillNumber	Number
ID	Number
Quantity	Number
Price	Currency

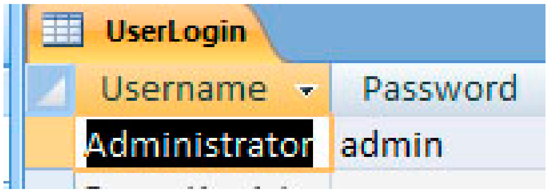
General Lookup

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Indexed	No
Smart Tags	
Text Align	General

5. Userlogin

This table stores the log-in information (Username and password)

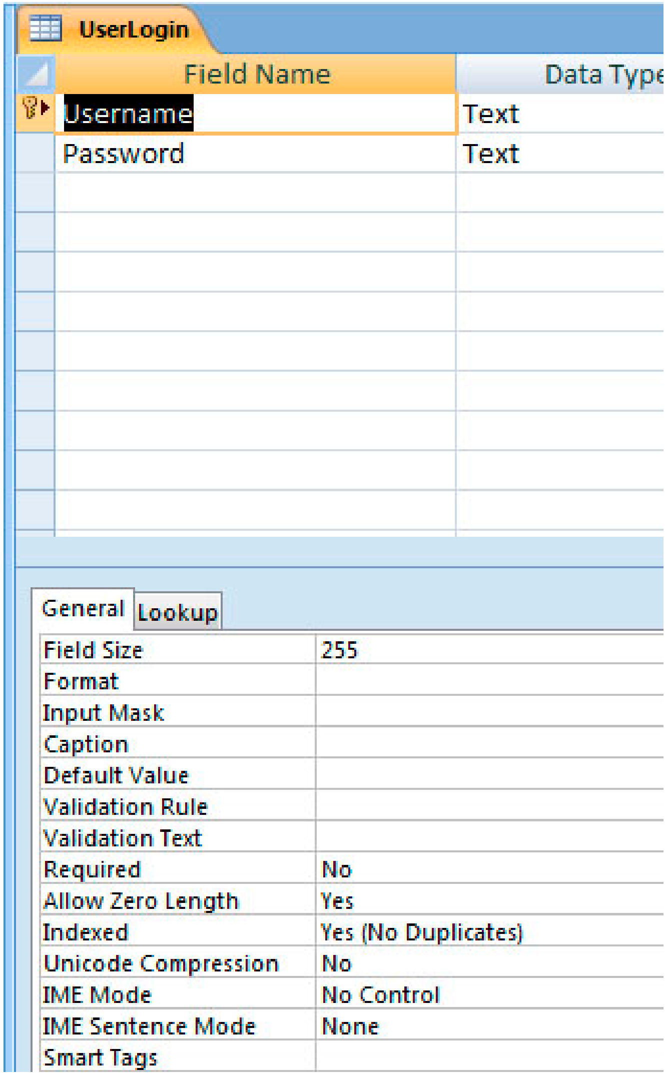
Datasheet View



UserLogin

Username	Password
Administrator	admin

Design View



UserLogin

Field Name	Data Type
Username	Text
Password	Text

Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

Data Dictionary

[P.K-> Primary Key; F.K-> Foreign Key]

1. Category_Table

Field Name	Data Type	Size(Bytes)	Description	Validation	Example
Category Name(P.K)	String	30	Name of Food Category	Unique	Chowmin
Size of a Record		**30 Bytes**			

2. Items_Table

Field Name	Data Type	Size(Bytes)	Description	Validation	Example

ID(P.K)	Integer	2	Unique Identification of each food item, System Generated	Unique	1
Item Name	String	40	Name of Food Item	Presence Check	Chicken Chowmin
Price	Currency	8	Selling Price of Food Item	Number Only	125
Category Name(F.K)	String	30	Name of Food Category	Presence Check	Chowmin
Size of a Record		**80 Bytes**			

3. CustomerBill_Table

Field Name	Data Type	Size(Bytes)	Description	Validation	Example
Bill Number	Integer	2	Unique Identification of each receipt,Automatically Generated	Unique	1
Date	Date	8	Date when the transaction took place.	Presence Check	2/28/2012
Table Number	Integer	2	Table Number of the bill .	Numbers Only	5
Discount	Currency	8	Discount Amount of the particular bill.	Numbers Only	145
Grand Total	Currency	8	Amount that Customer paid.	Numbers Only	500
Remarks	String	50	Additional Detail about the transaction.		Was Owner
Size of a Record		**78 Bytes**			

4. BillDetails_Table

Field Name	Data Type	Size(Bytes)	Description	Validation	Example
Bill Number(P.K)	Integer	2	Receipt Number	Number Only	1
ID(P.K, F.K)	Integer	2	Identifier of Food Item	Presence Check	1
Quantity	Integer	2	Quantity of Food item of given ID	Number Only	3
Price	Currency	8	Selling Price of selected Food item	Number Only	125

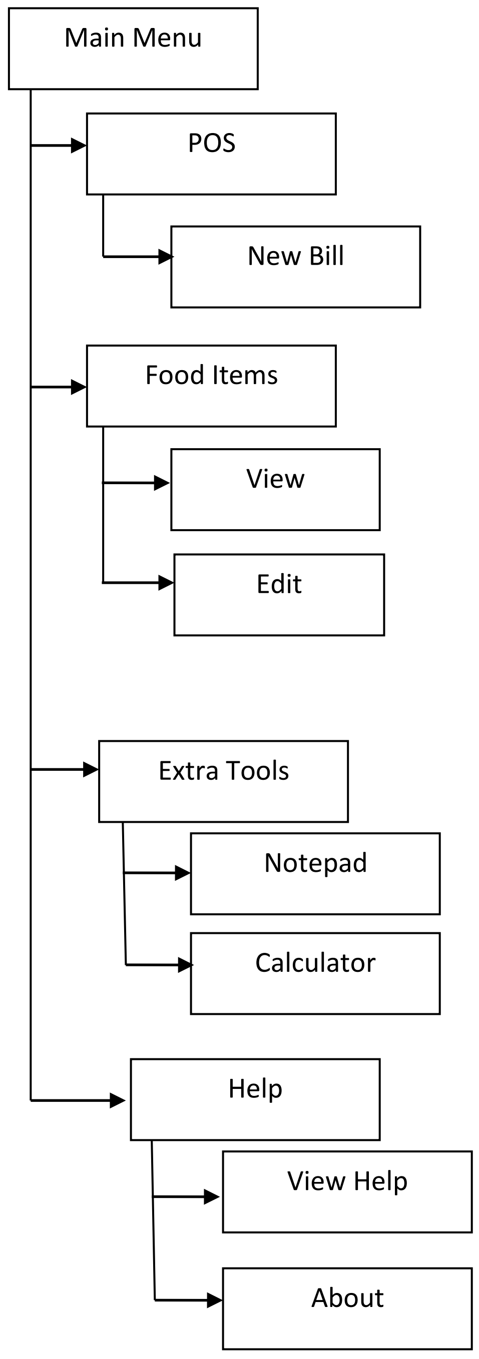
Size of a Record	**14 Bytes**

UserLogin

Field Name	Data Type	Size(Bytes)	Description	Validation	Example
Username	String	15	Username through which user gets access to system	Presence Check	Administrator, Receptionist
Password	String	20	Password of the Username	Presence Check, Length Check	Admin,recep
Size of a Record			**35 Bytes**		

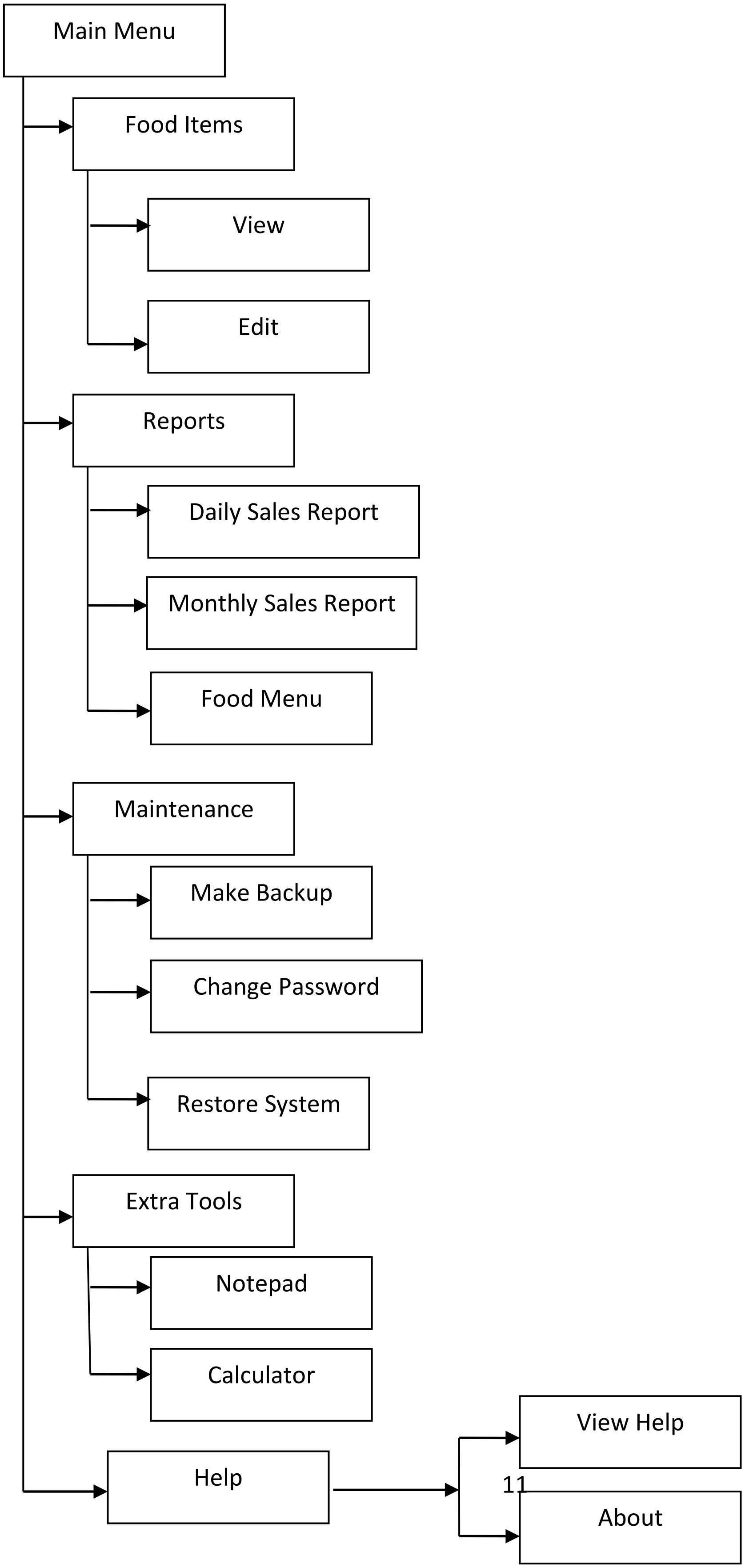
Menu

For Receptionist

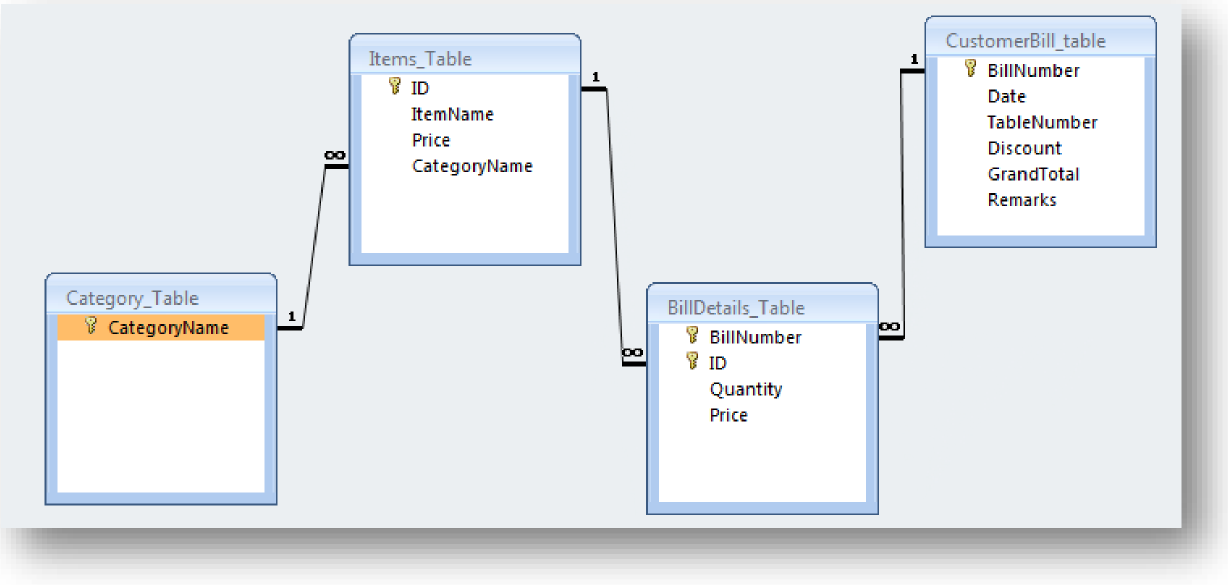


-

For Administrator



Entity- Relationship (E-R) Diagram



Relationships

Category_Table **Items_Table**

Category_Table			**Items_Table**
CategoryName	1	1	CategoryName

Attributes: Enforced, Cascade Updates, Cascade Deletes
RelationshipType: One-To-Many

Items_Table **BillDetails_Table**

Items_Table			**BillDetails_Table**
ID	1	∞	ID

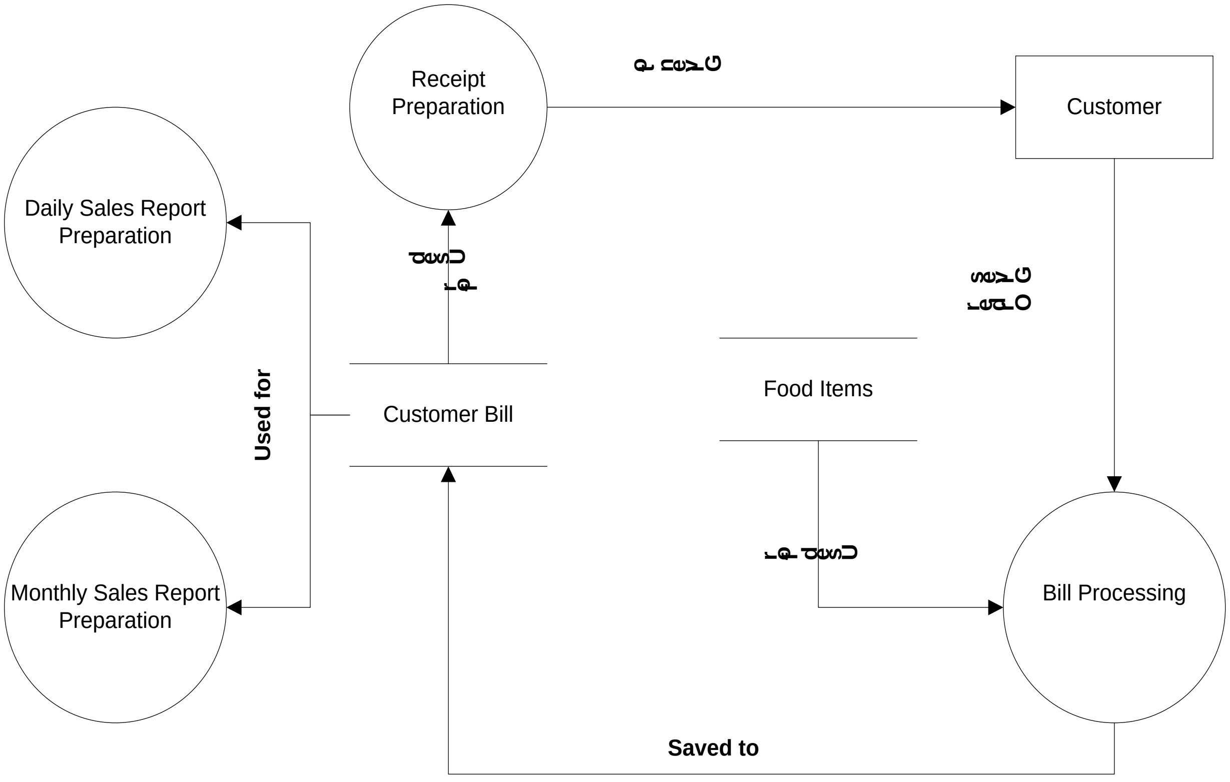
Attributes: Enforced, Cascade Updates, Cascade Deletes
RelationshipType: One-To-Many

CustomerBill_Table **BillDetails_Table**

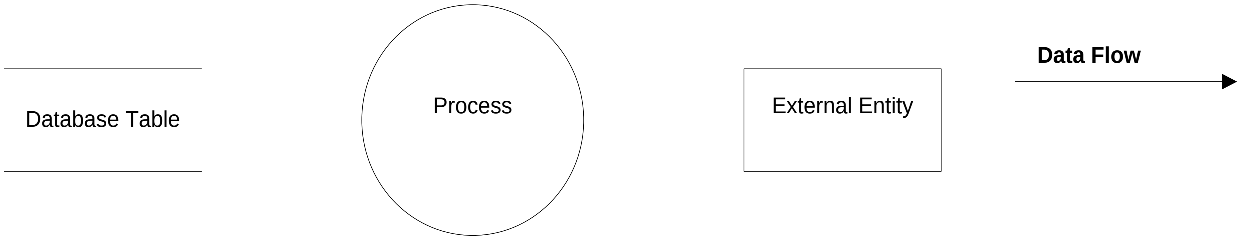
CustomerBill_Table			**BillDetails_Table**
BillNumber	1	∞	BillNumber

Attributes: Enforced, Cascade Updates, Cascade Deletes
RelationshipType: One-To-Many

Data Flow Diagram



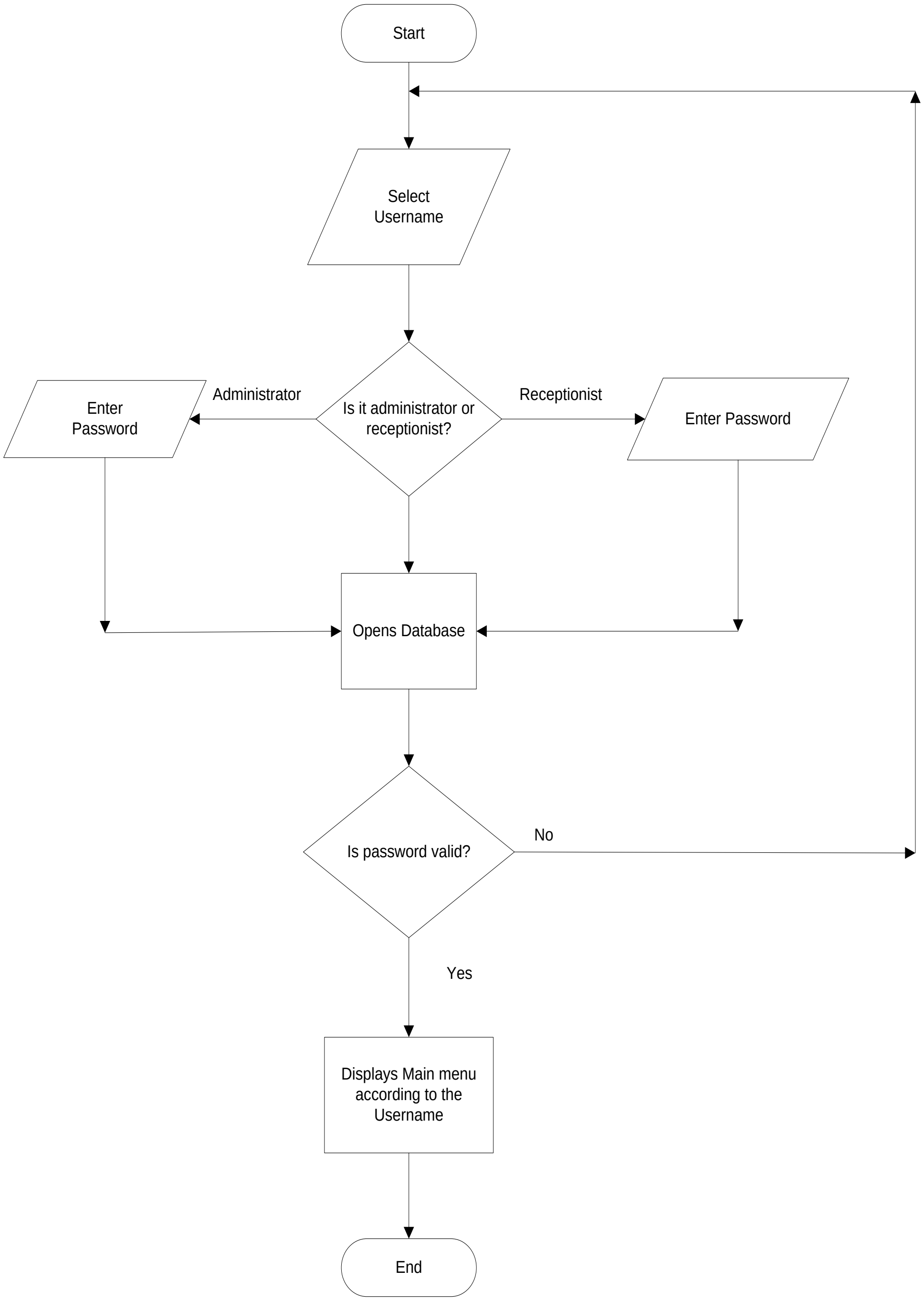
Key:



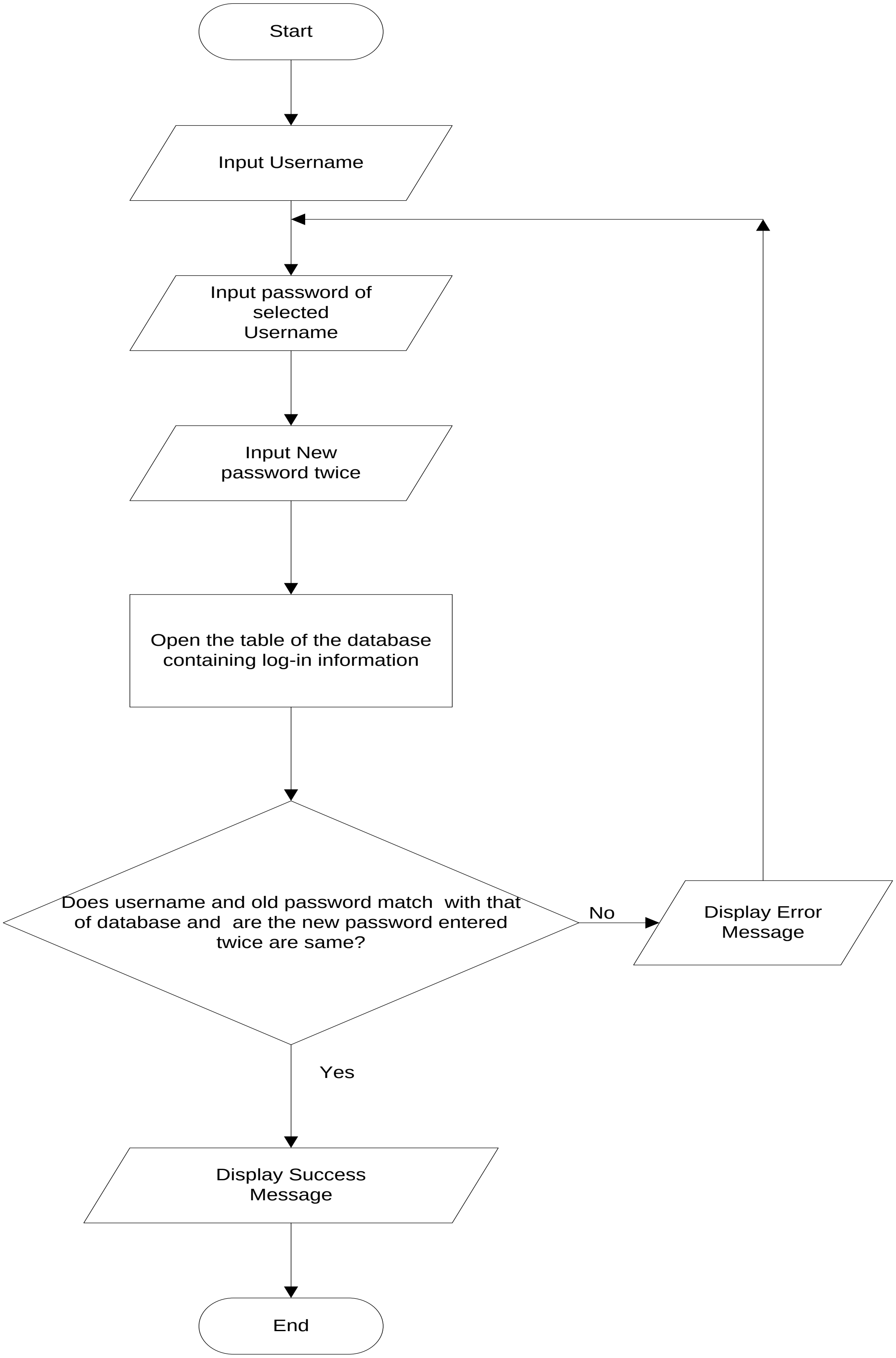
Flowcharts

Log-in

Log-in System

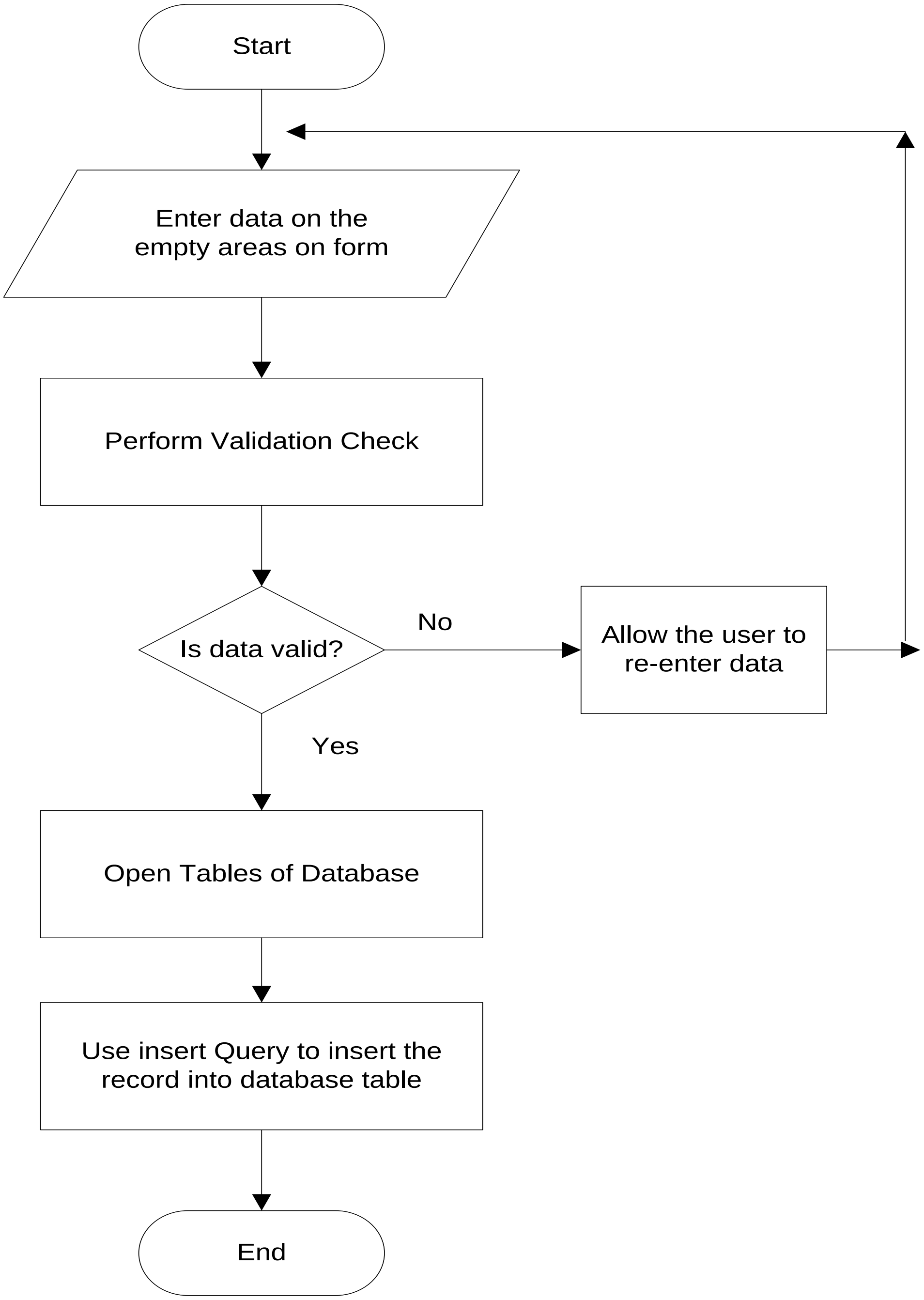


Change Password

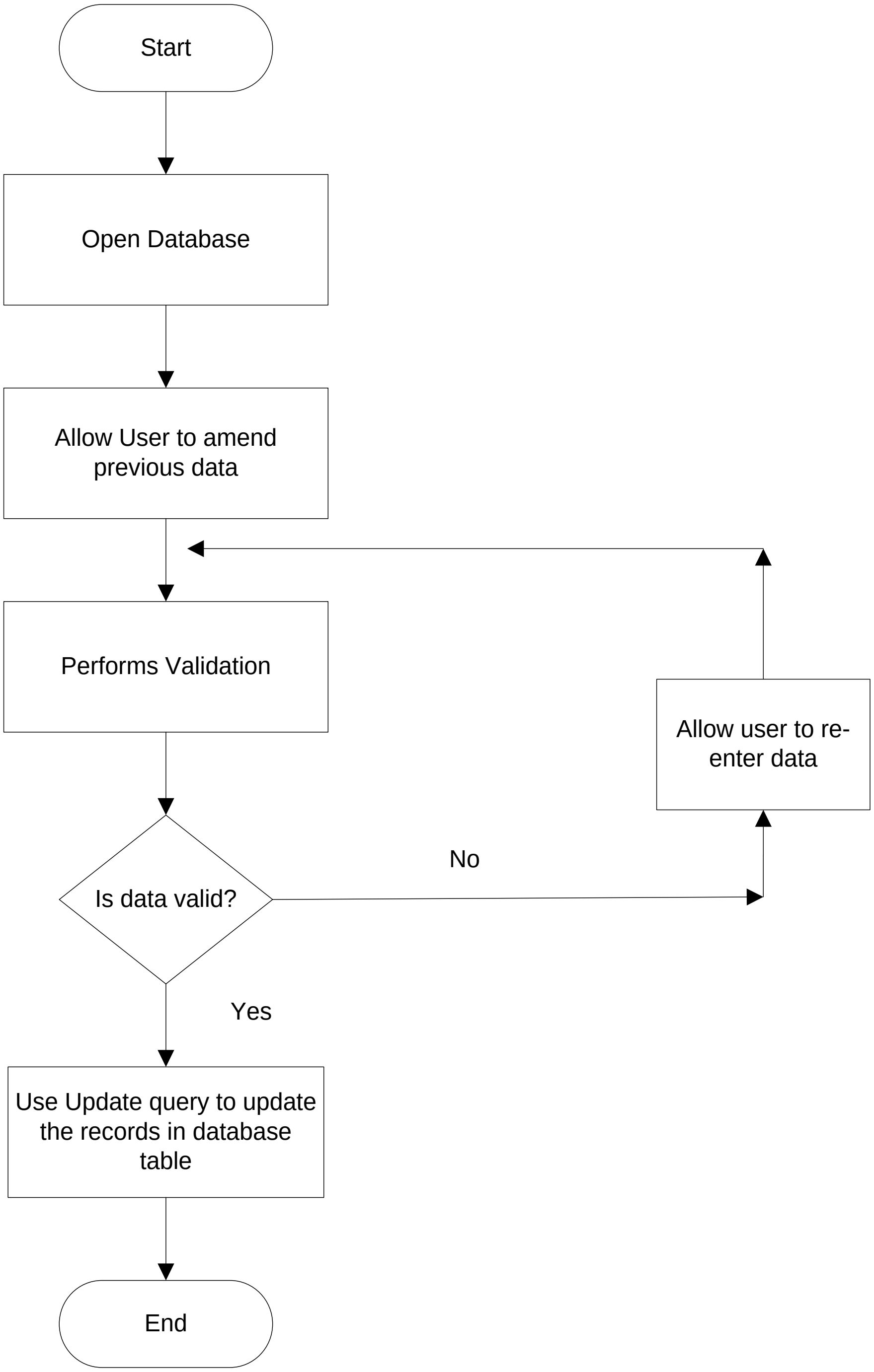

Start
Input Username
Input password of selected Username
Input New password twice
Open the table of the database containing log-in information
Does username and old password match with that of database and are the new password entered twice are same?
No
Display Error Message
Yes
Display Success Message
End

Food Items

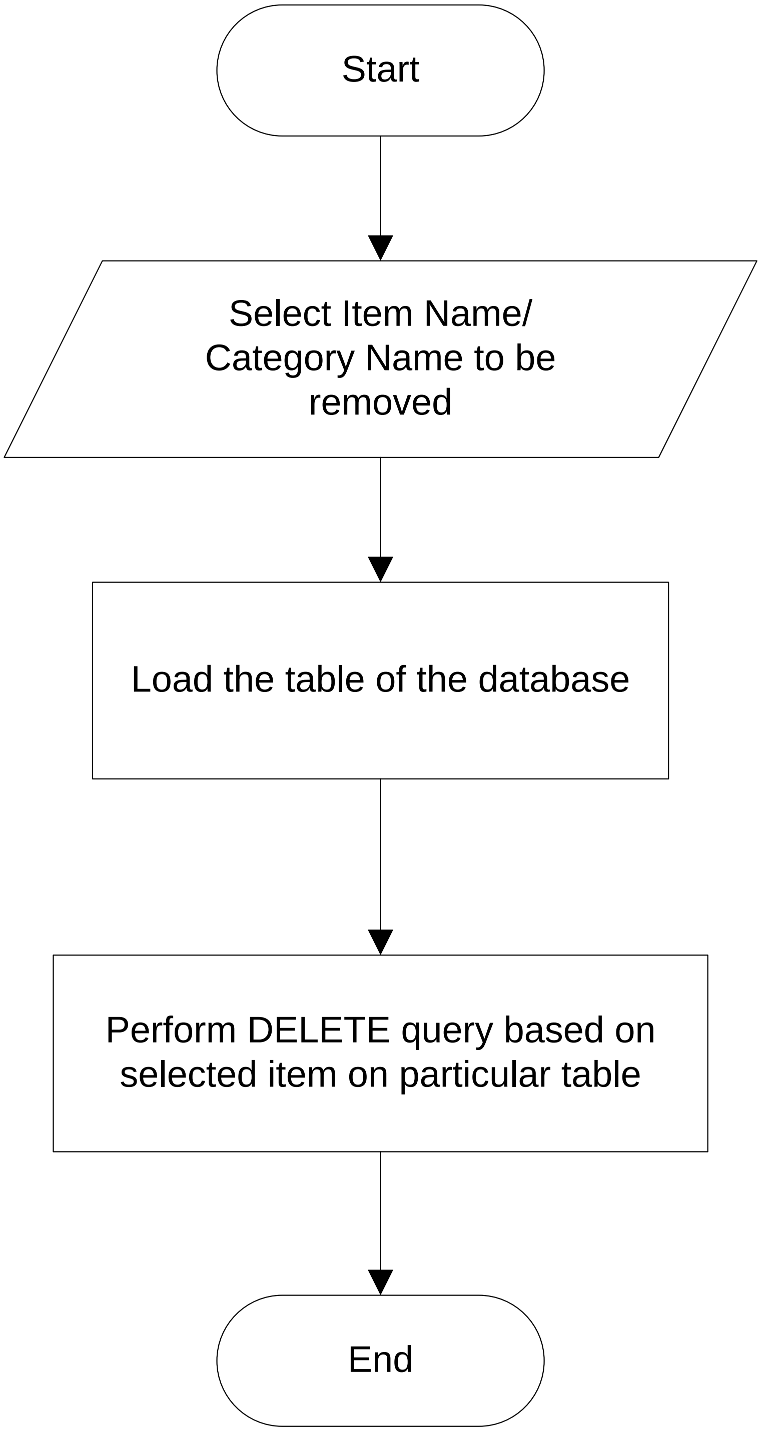
Add Food Item/Category


Start
Enter data on the empty areas on form
Perform Validation Check
Is data valid?
No
Allow the user to re-enter data
Yes
Open Tables of Database
Use insert Query to insert the record into database table
End

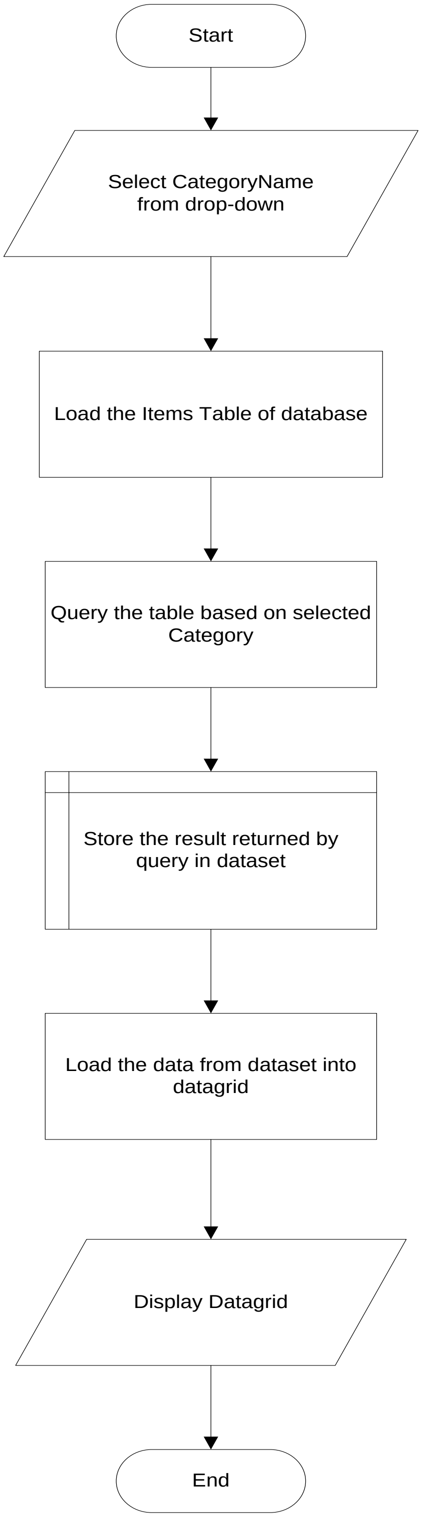
Edit Food Items


Start
Open Database
Allow User to amend previous data
Performs Validation
Allow user to re-enter data
Is data valid?
No
Yes
Use Update query to update the records in database table
End

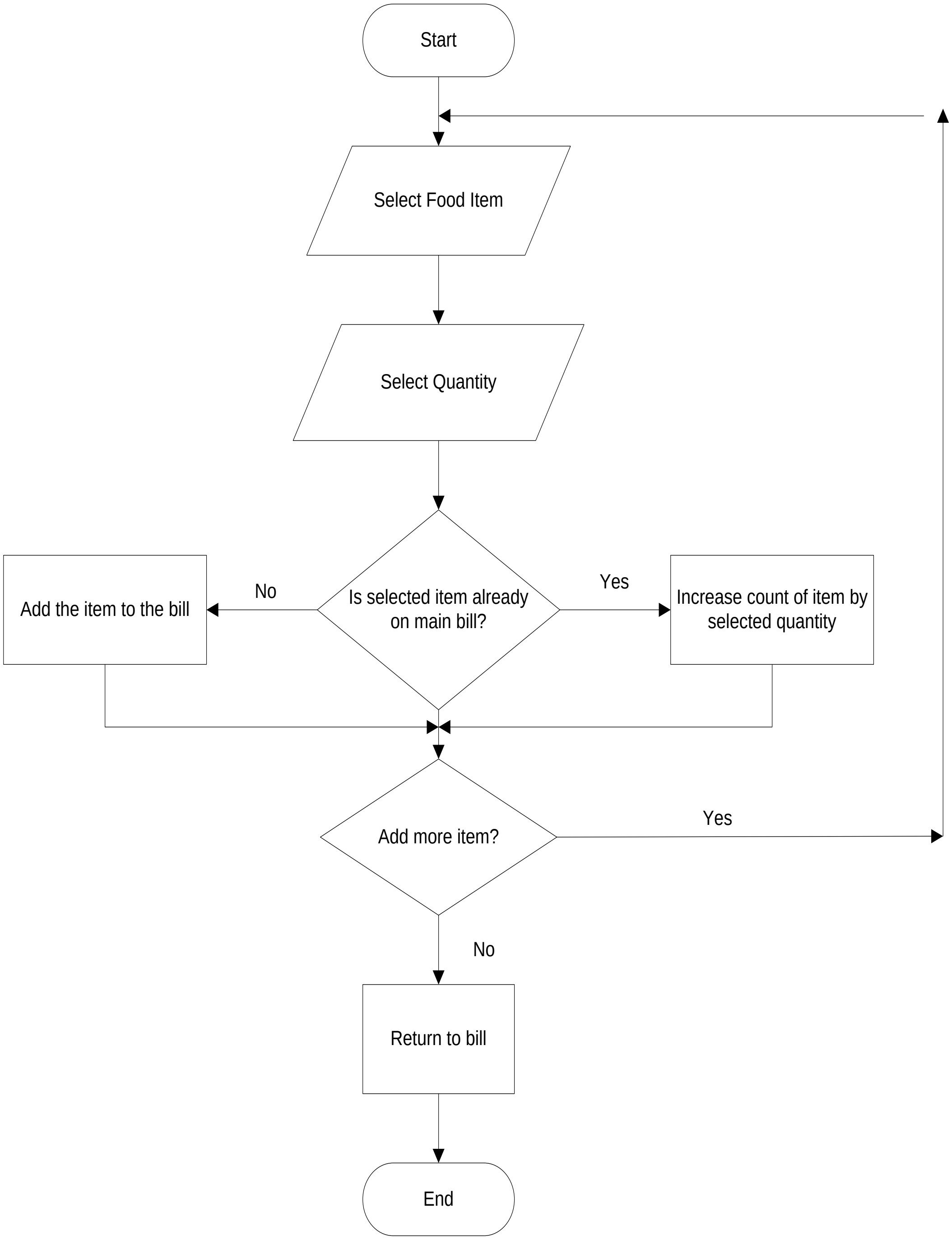
Remove Food Item/ Category


Start
Select Item Name/ Category Name to be removed
Load the table of the database
Perform DELETE query based on selected item on particular table
End

View Food Items

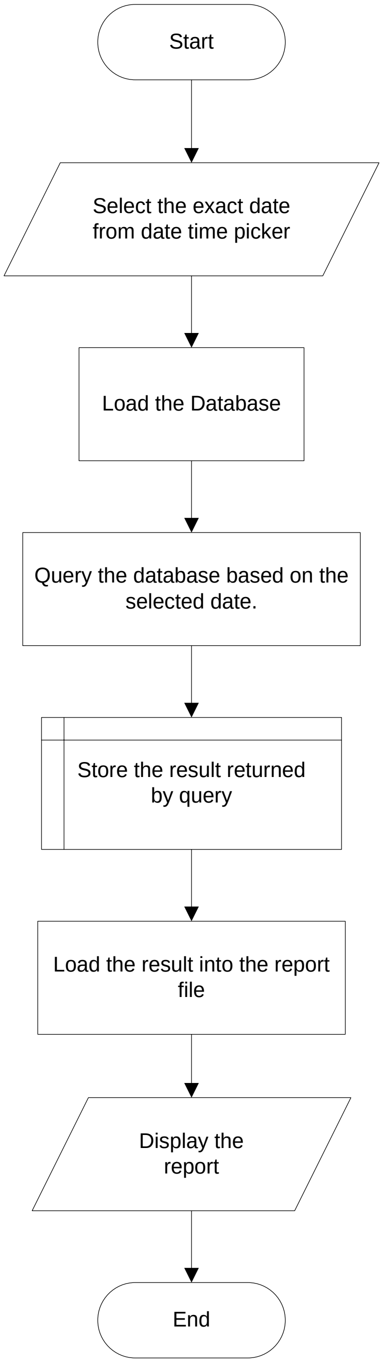

Start
Select CategoryName from drop-down
Load the Items Table of database
Query the table based on selected Category
Store the result returned by query in dataset
Load the data from dataset into datagrid
Display Datagrid
End

Bill(Adding Food Item)

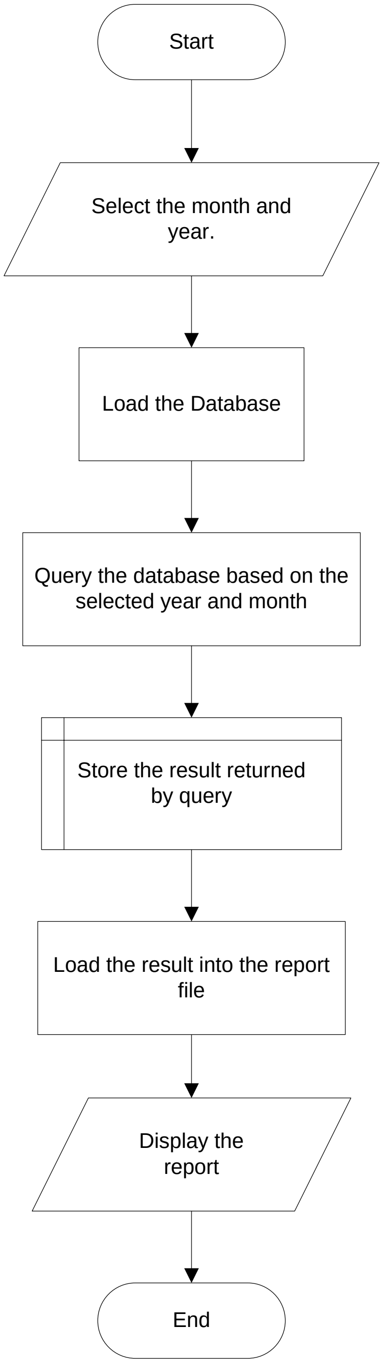


Reporting

Daily Sales Report

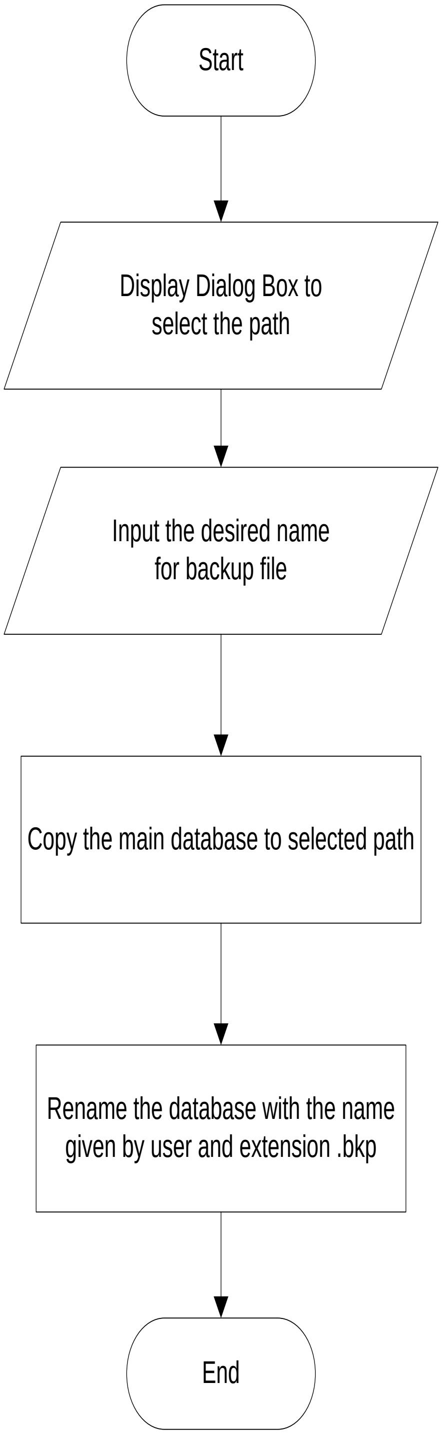

Start
Select the exact date from date time picker
Load the Database
Query the database based on the selected date.
Store the result returned by query
Load the result into the report file
Display the report
End

Monthly Sales Report

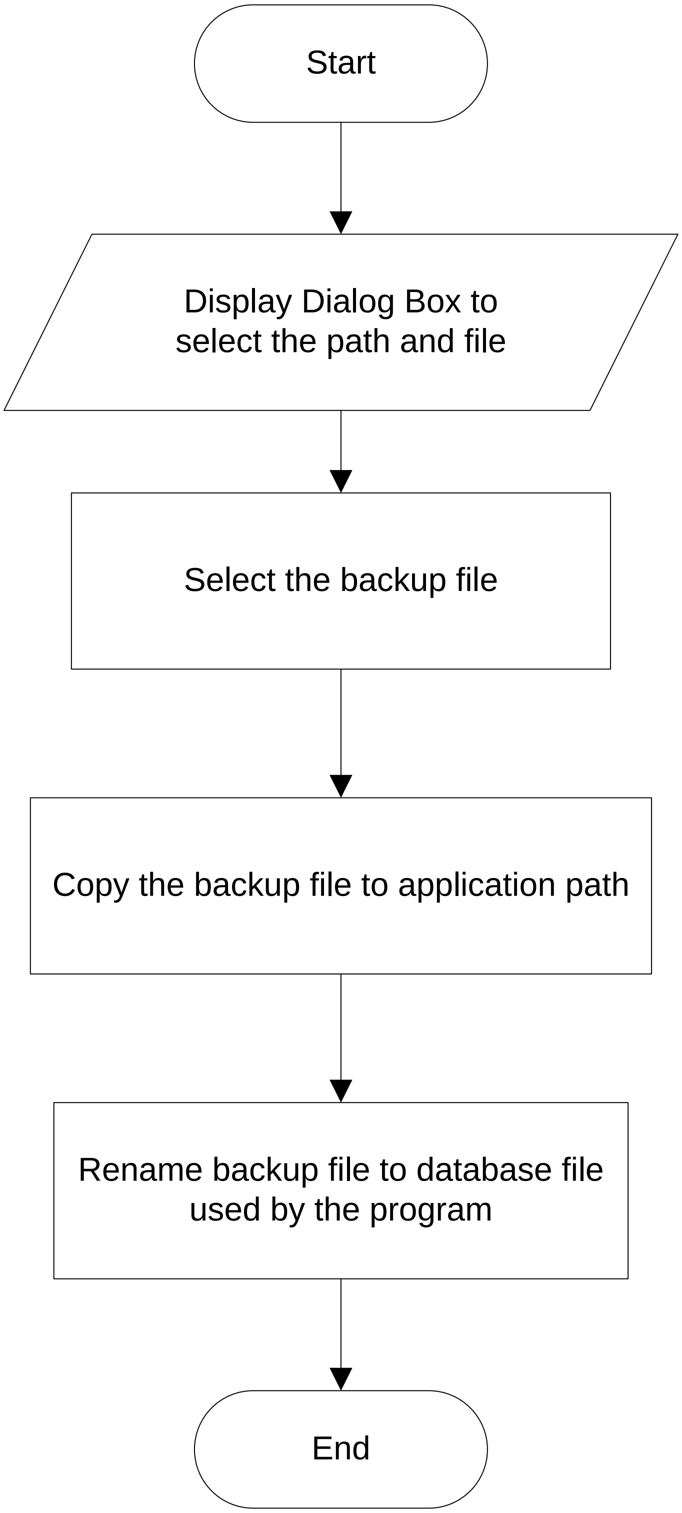

Start
Select the month and year.
Load the Database
Query the database based on the selected year and month
Store the result returned by query
Load the result into the report file
Display the report
End

Maintenance

Backup


Start
Display Dialog Box to select the path
Input the desired name for backup file
Copy the main database to selected path
Rename the database with the name given by user and extension .bkp
End

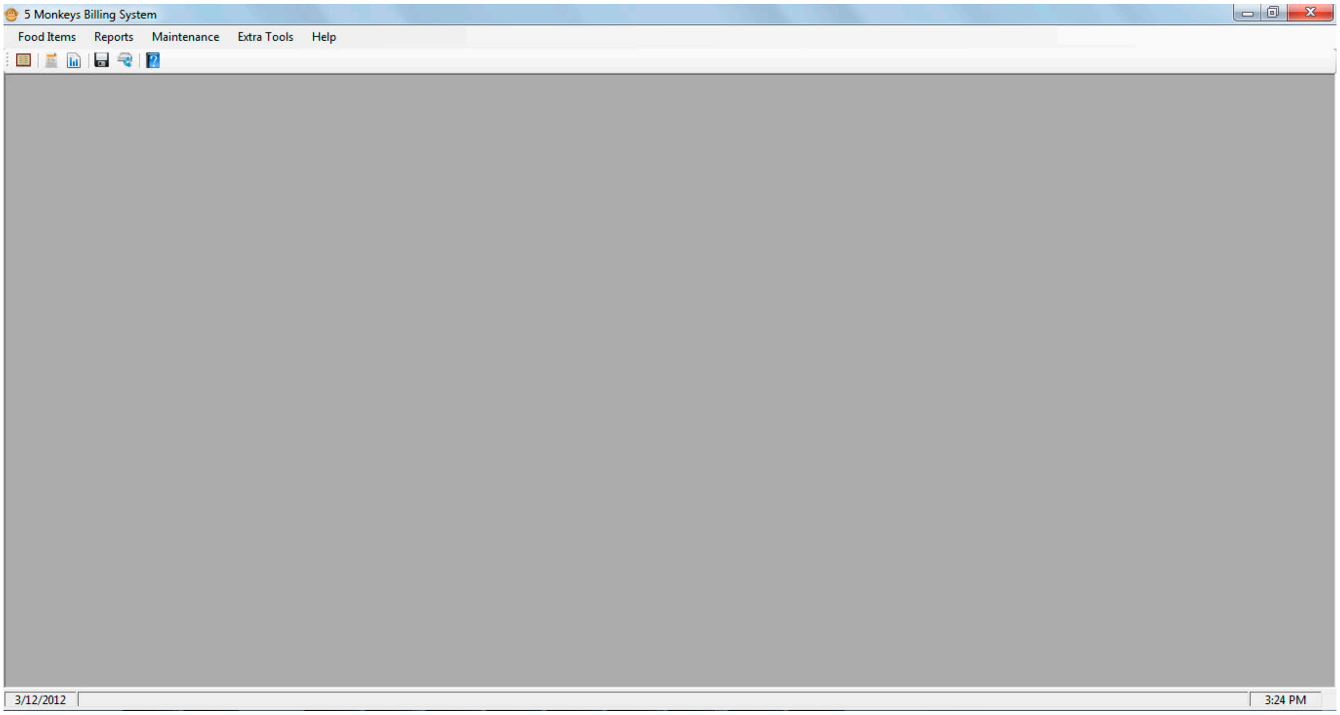
Restore


Start
Display Dialog Box to select the path and file
Select the backup file
Copy the backup file to application path
Rename backup file to database file used by the program
End

Program Listing

The initial design (mock screen, which shows the purpose of each control in the form) and the final design (forms made in VB 2010) along with the methods used in each form are shown below:

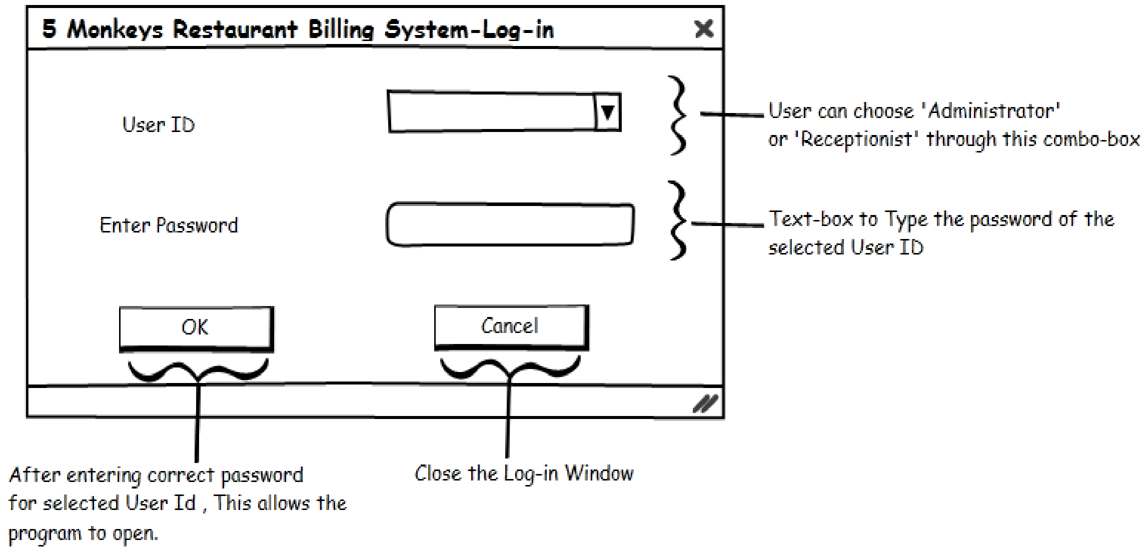
Main Form(MDI Form)



Methods	Purpose
Public Sub Main_Load	Displays log-in form and manages the settings for menu hierarchy according to selected username
Private Sub EditFoodItems()	Displays Edit Food Item form as a dialog box
Private Sub CreateNewBill()	Displays form to create a new bill
Private Sub MakeBackup()	Creates a backupfile of the system
Private Sub RestoreSystem()	Restores the system to the selected backup file
Private Sub Timer1_Tick	Updates the time(hour and minutes) in the status bar
Private Sub DisplayFoodMenu()	Displays Food Menu Report
Private Sub DisplayDailySales()	Displays daily sales report
Private Sub DisplayMonthlySales()	Displays Monthly Sales Report
Private Sub ViewFoodItems()	Displays View Food Items form
Private Sub mnuMaintenanceChangePassword_Click()	Displays form for changing the password of the system

Private Sub mnuFoodItemsView_Click()	Calls ViewFoodItem( ) sub procedure
Private Sub btnViewMenu_Click()	Calls ViewFoodItem( ) sub procedure
Private Sub mnuFoodItemsEdit_Click()	Calls EditFoodItems() sub procedure
Private Sub mnuPOSNewBill_Click()	Calls CreateNewBill() sub procedure
Private Sub btnNewBill_Click()	Calls CreateNewBill() sub procedure
Private Sub mnuExtraCalculator_Click()	Displays System Calculator
Private Sub mnuExtraNotepad_Click()	Displays System Notepad
Private Sub mnuMaintenanceBackup_Click()	Calls MakeBackup() sub procedure
Private Sub RestoreToolStripMenuItem_Click()	Calls RestoreSystem() sub procedure
Private Sub btnBackup_Click()	Calls MakeBackup() sub procedure
Private Sub btnRestore_Click()	Calls RestoreSystem() sub procedure
Private Sub AddNewBillToolStripMenuItem_Click()	Calls CreateNewBill() sub procedure
Private Sub EditFoodItemsToolStripMenuItem_Click()	Calls EditFoodItems() sub procedure
Private Sub FoodMenuToolStripMenuItem_Click()	Calls DisplayFoodMenu() sub procedure
Private Sub mnuReportsDaily_Click()	Calls DisplayDailySales() sub procedure
Private Sub btnMonthlySales_Click()	Calls DisplayMonthlySales() sub procedure
Private Sub BackupToolStripMenuItem_Click()	Calls MakeBackup() sub procedure
Private Sub RestoreToolStripMenuItem_Click()	Calls RestoreSystem() sub procedure
Private Sub DailySalesReportToolStripMenuItem_Click	Calls DisplayDailySales() sub procedure
Private Sub mnuReportsMonthly_Click()	Calls DisplayMonthlySales() sub procedure
Private Sub MonthlySalesReportToolStripMenuItem_Click()	Calls DisplayMonthlySales() sub procedure
Private Sub btnDailySales_Click()	Calls DisplayDailySales() sub procedure
Private Sub mnuHelpAbout_Click	Displays About Dialog box

Log-in Screen

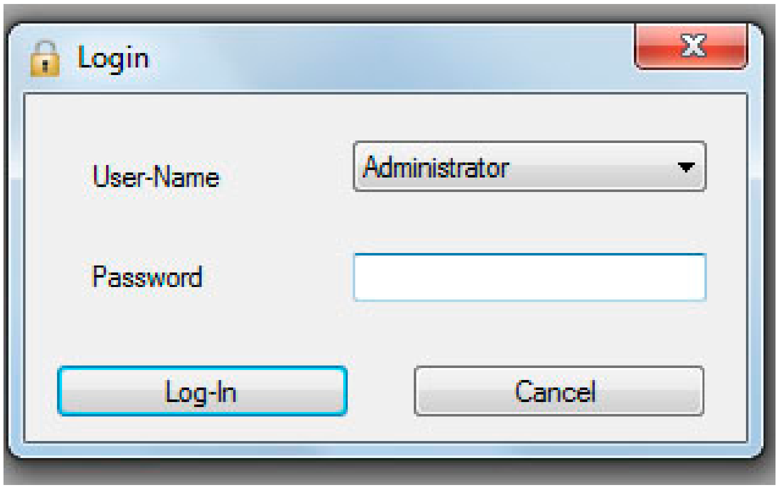


User can choose 'Administrator' or 'Receptionist' through this combo-box

Text-box to Type the password of the selected User ID

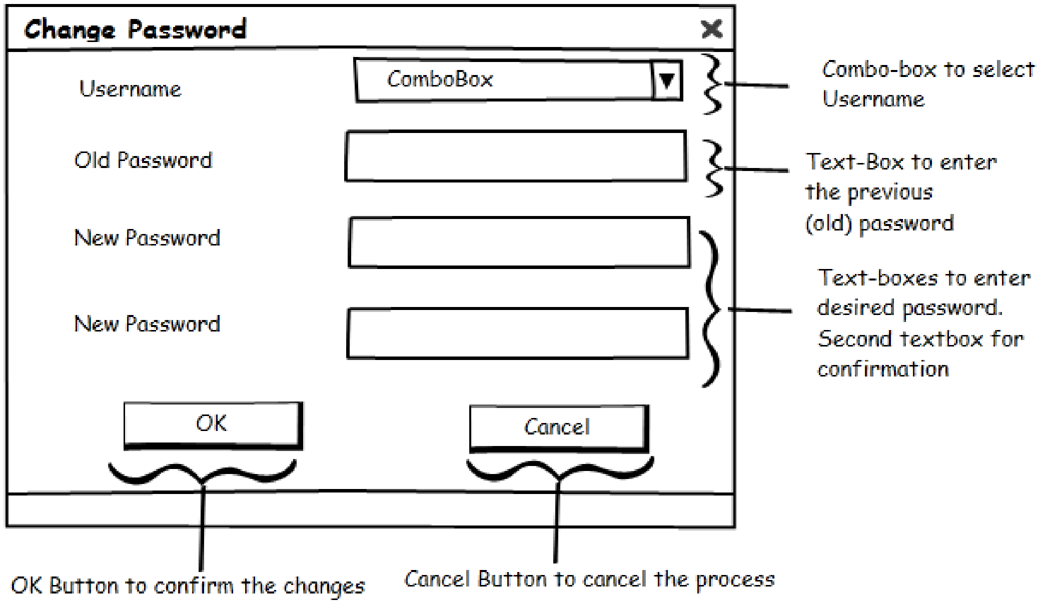
After entering correct password for selected User Id , This allows the program to open.

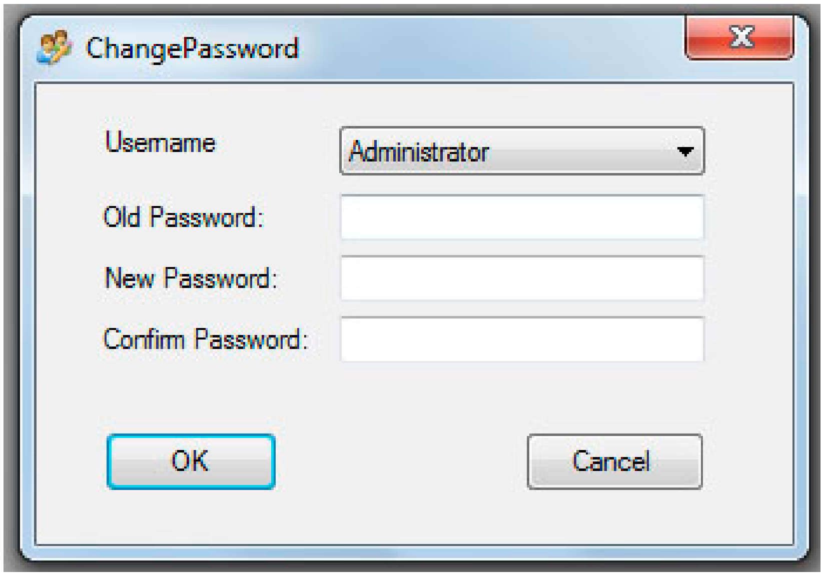
Close the Log-in Window



Methods	Purpose
Private Sub Login_Load()	Loads the user name in Combo-box from database
Private Sub btnLogIn_Click()	Stores the entered password in tag property of login form
Private Sub btnCancel_Click()	Closes the form

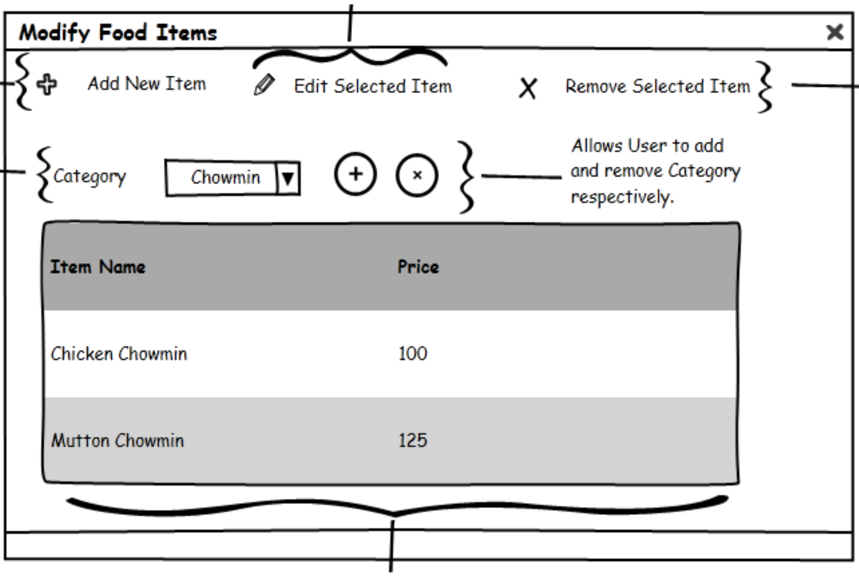
Password Changer Screen


Change Password
Username
ComboBox
Old Password
New Password
New Password
OK
Cancel
Combo-box to select Username
Text-Box to enter the previous (old) password
Text-boxes to enter desired password. Second textbox for confirmation
OK Button to confirm the changes
Cancel Button to cancel the process



Methods	Purpose
Private Sub btnOk_Click()	Performs necessary validations and changes the password of the selected user in database table
Private Sub btnCancel_Click()	Closes the form
Private Sub ChangePassword_Load	Loads the user name in Combo-box from database

Modify Food Items Screen



Item Name	Price
Chicken Chowmin	100
Mutton Chowmin	125

It displays new window allowing to edit the name and price of selected food item.

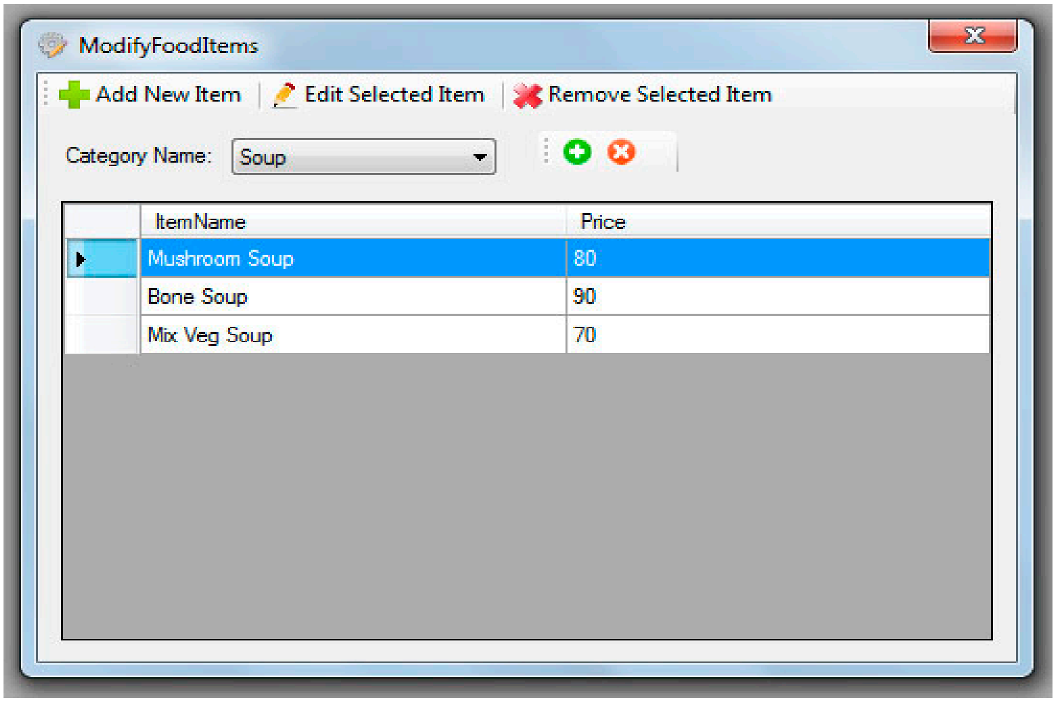
It displays new window which allows user to add new food item to selected category.

It removes the food item selected in the table.

Allows user to select the category name from the combo-box.

Allows User to add and remove Category respectively.

Displays the name and price of food item which lies in the selected category.

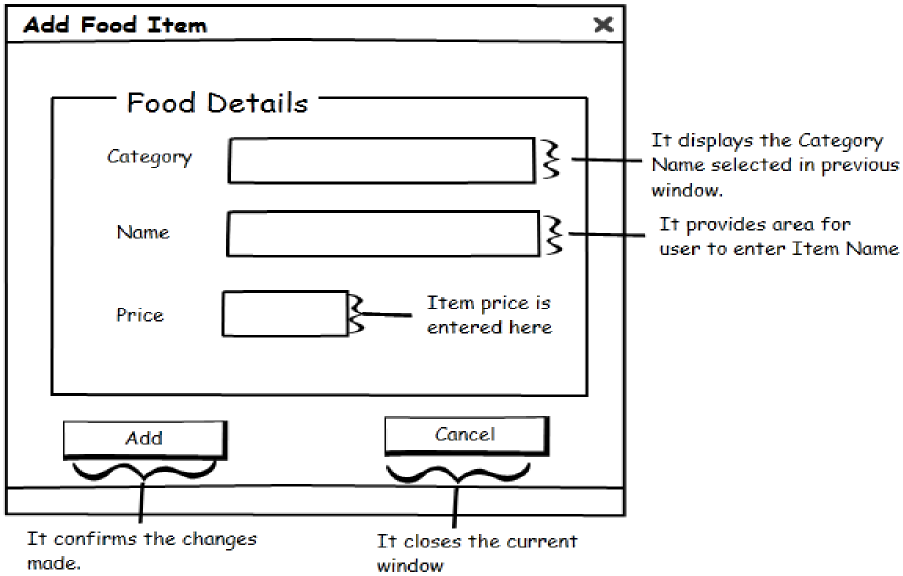


Category Name: Soup

	ItemName	Price
▶	Mushroom Soup	80
	Bone Soup	90
	Mix Veg Soup	70

Methods	Purpose
Private Sub ModifyFoodItems_Load()	Loads the food items and their prices in data grid
Private Sub cboCategoryName_SelectedValueChanged()	Displays the food items and their prices of the chosen category
Private Sub btnAddNewItem_Click()	Opens AddFoodItem form
Private Sub btnRemoveSelectedItem_Click()	Removes the selected Food Item
Private Sub btnEditSelectedItem_Click()	Opens EditSelectedItem form
Private Sub btnAddCategory_Click()	Opens AddCategory form
Private Sub btnDeleteCategory_Click()	Removes the slected category and updates the database

Add new Item Screen



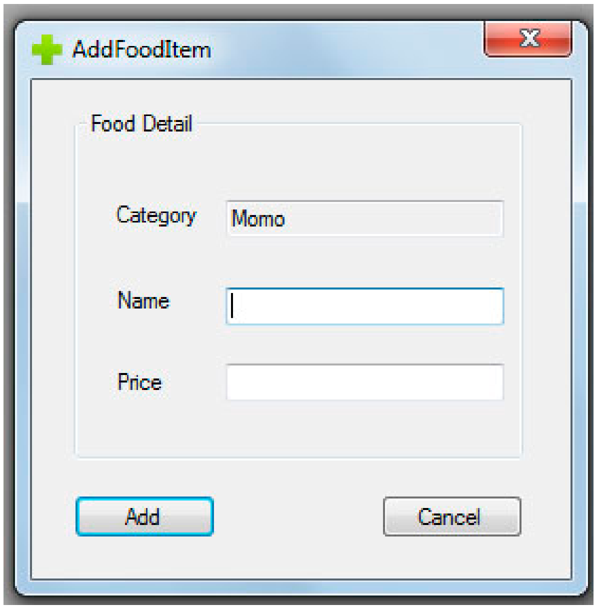
It displays the Category Name selected in previous window.

It provides area for user to enter Item Name

Item price is entered here

It confirms the changes made.

It closes the current window



Food Detail

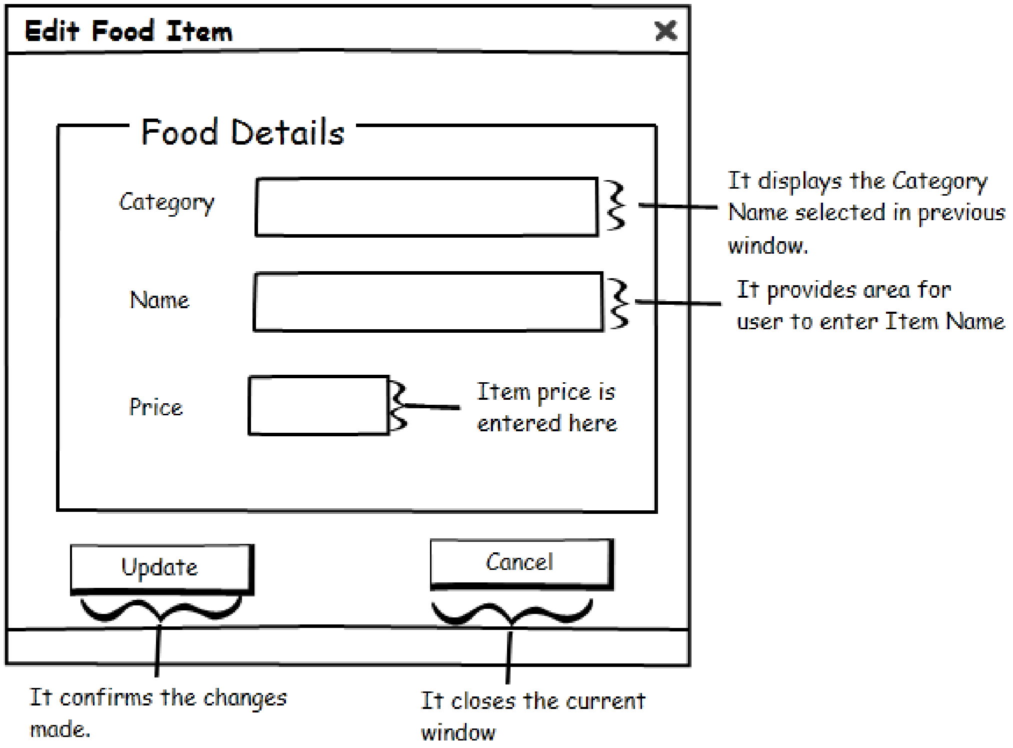
Category: Momo

Name:

Price:

Methods	Purpose
Private Sub btnAdd_Click()	Adds the entered item name and price to the table in database
Private Function IsValidData()	Performs validation on entered data using the Validator Class
Private Sub AddFoodItem_Load()	Displays the category selected in the text-box of this form when the form loads
Private Sub btnCancel_Click()	Closes the form

Edit Food Items Screen



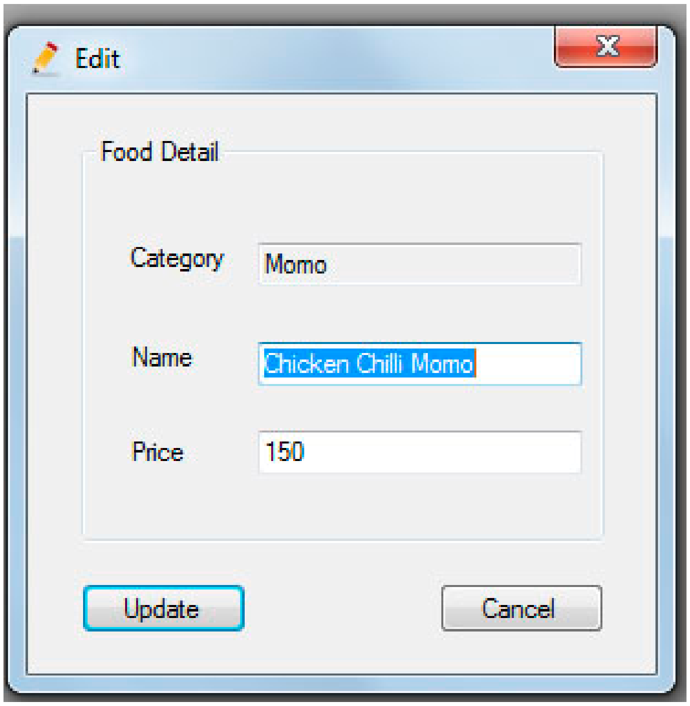
It displays the Category Name selected in previous window.

It provides area for user to enter Item Name

Item price is entered here

It confirms the changes made.

It closes the current window



Edit

Food Detail

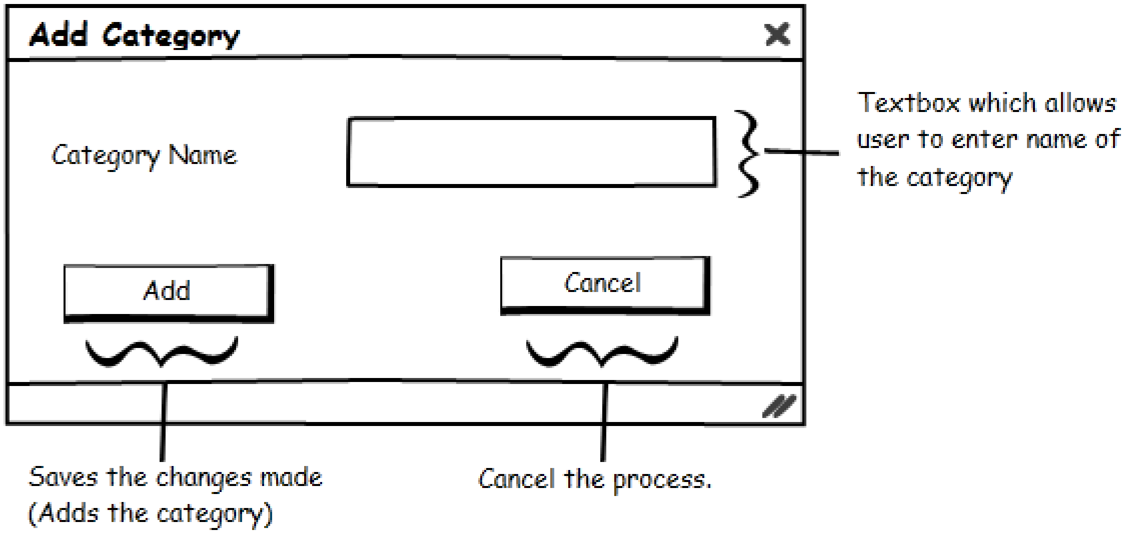
Category: Momo

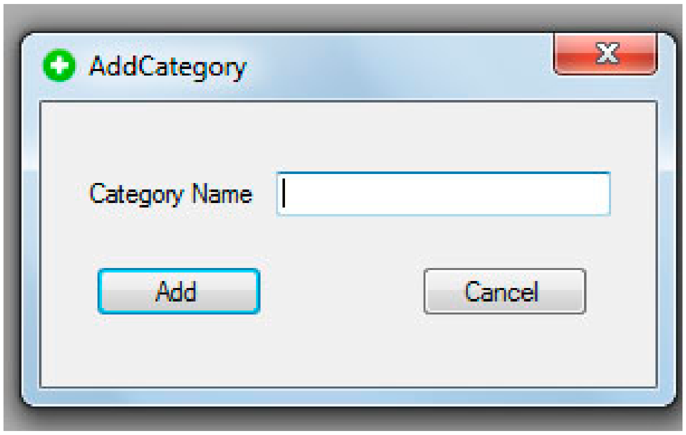
Name: Chicken Chilli Momo

Price: 150

Methods	Purpose
Private Sub EditSelectedItem_Load()	fills the category name text box with selected category
Public Sub FillItemInfo()	Fill the remaining text-box with the information of selected food item
Private Function IsValidData() As Boolean	Performs validation text using the validator class.
Private Sub txtUpdate_Click()	Updates the information about food items in database table
Private Sub txtCancel_Click()	Closes the form

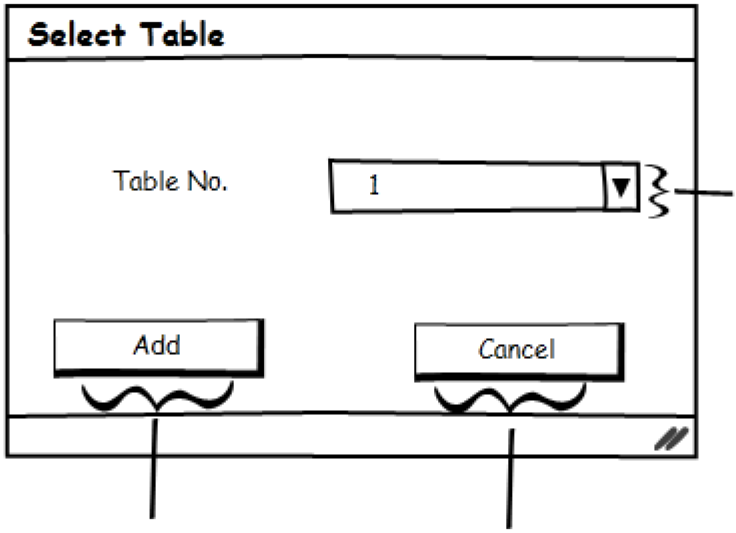
Add Category


Add Category
Category Name
Add
Cancel
Textbox which allows user to enter name of the category
Saves the changes made (Adds the category)
Cancel the process.



Methods	Purpose
Private Sub btnAdd_Click()	Validates the category name and adds it to the database table.
Private Sub btnCancel_Click()	Close the form

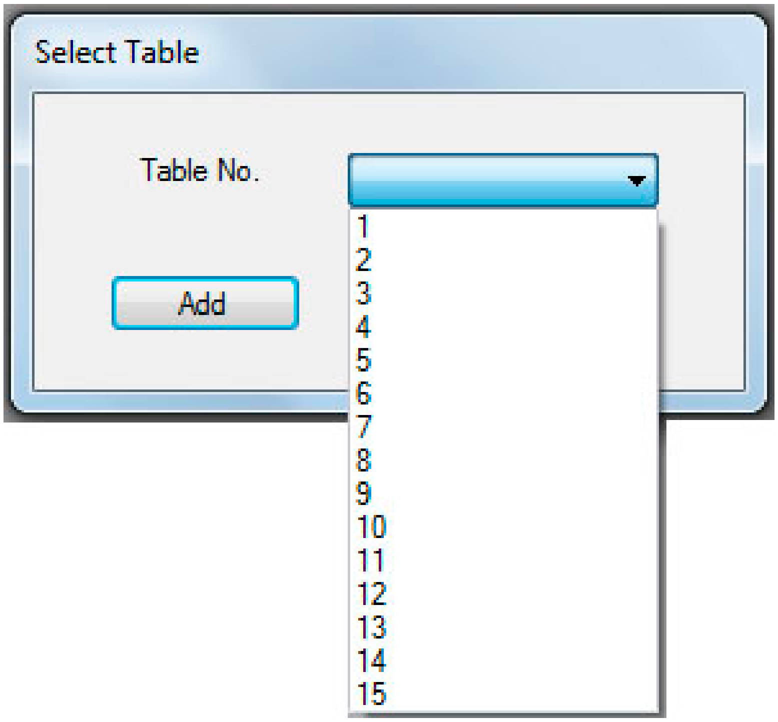
Select Table Screen



Comobox that allows Receptionist to select the table number from the dropdown list.

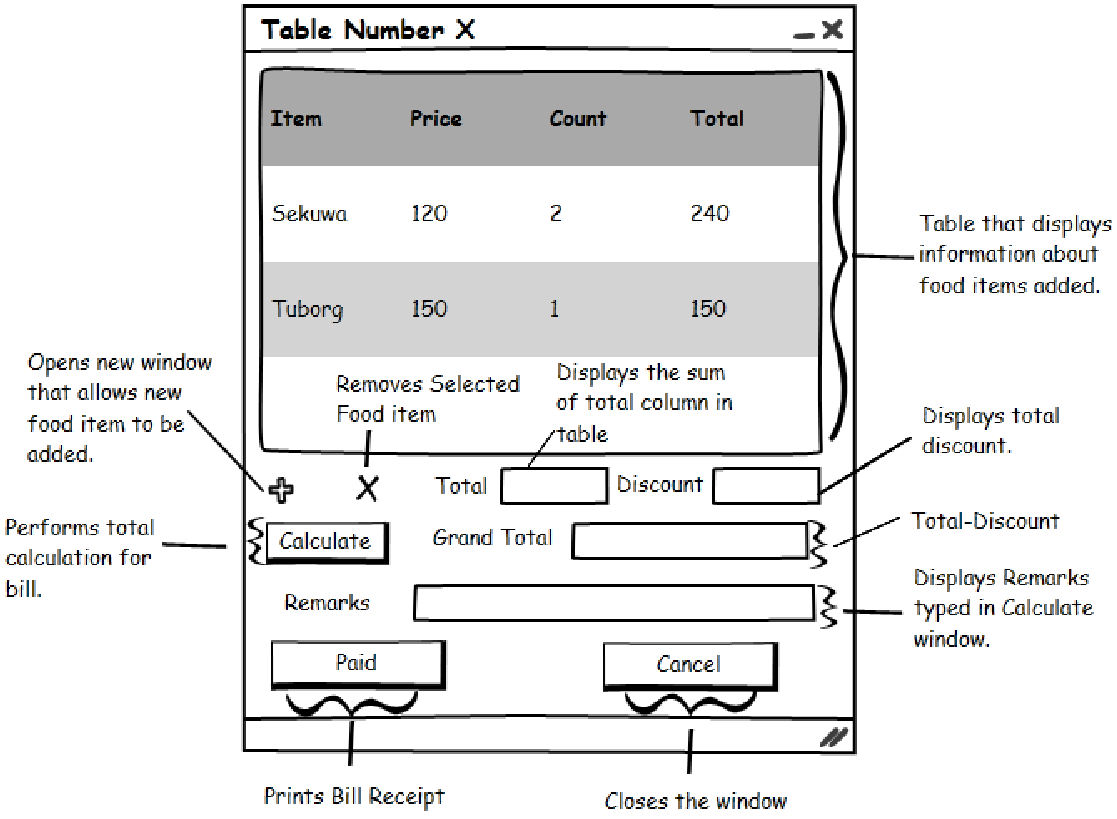
Creates a billing window for selected table number

Cancels the Process



Methods	Purpose
Private Sub btnAdd_Click()	Creates a new bill form with the name of selected table

Bill Screen



Item	Price	Count	Total
Sekuwa	120	2	240
Tuborg	150	1	150

Table that displays information about food items added.

Opens new window that allows new food item to be added.

Removes Selected Food item

Displays the sum of total column in table

Displays total discount.

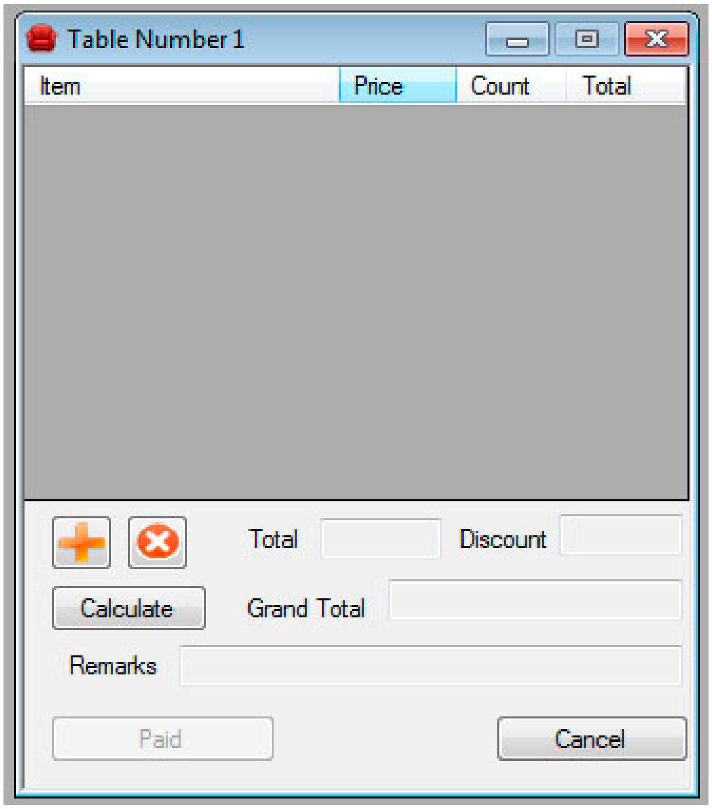
Performs total calculation for bill.

Total-Discount

Displays Remarks typed in Calculate window.

Prints Bill Receipt

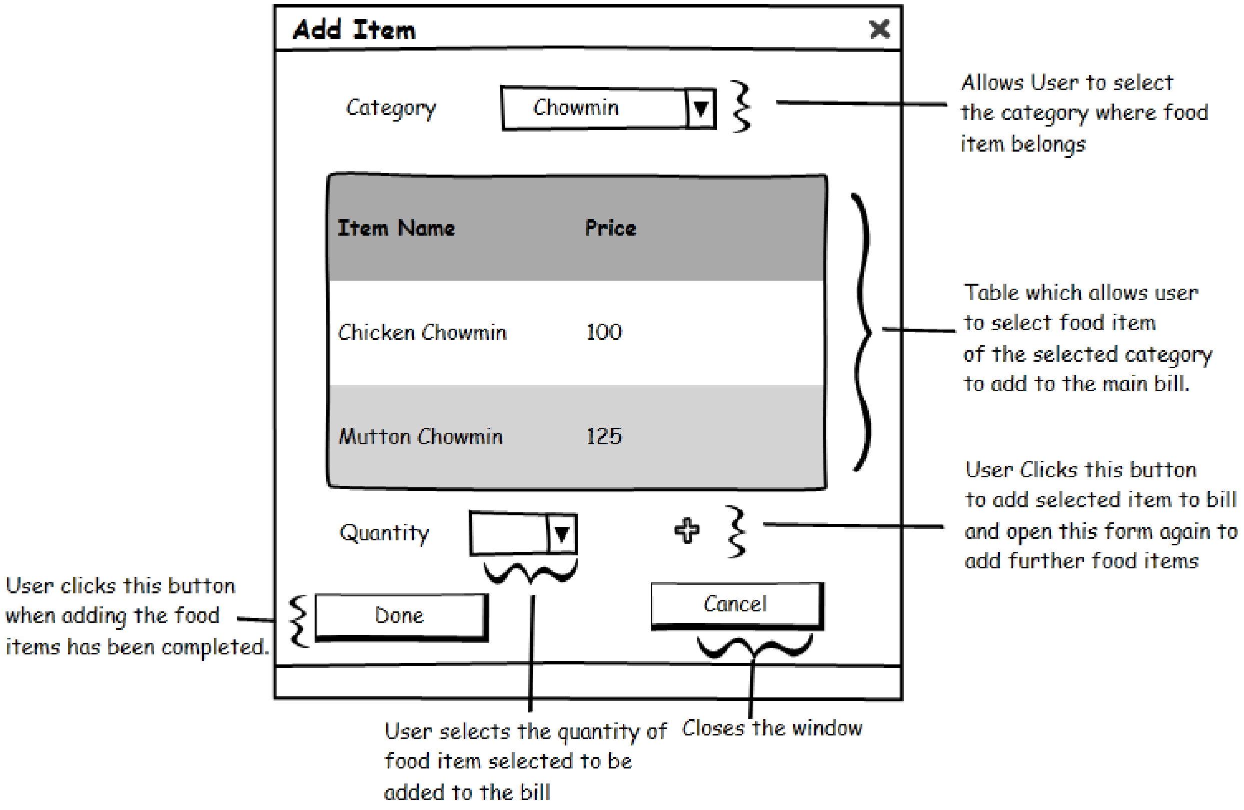
Closes the window



Methods	Purpose
Private Sub btnAddItem_Click()	Opens Additem form and adds the selected item to the main bill
Private Sub btnRemoveItem_Click()	Removes the selected item from the main bill
Private Sub btnCalculate_Click()	Opens Calculate form and performs calculation based on data entered in calculate form and displays the result in main bill
Private Sub btnPaid_Click()	Updates the database tables and displays the receipt bill for printing
Private Sub btnCancel_Click()	Closes the form
Private Sub Bill_Activated()	Displays the purpose of the form in the status bar

Add Item to bill

This form is displayed when user clicks + button in above form



Item Name	Price
Chicken Chowmin	100
Mutton Chowmin	125

Allows User to select the category where food item belongs

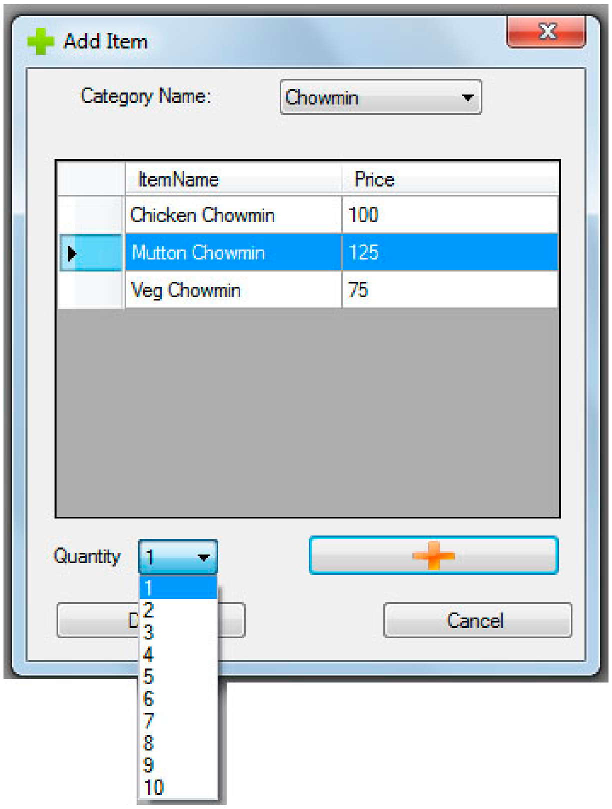
Table which allows user to select food item of the selected category to add to the main bill.

User Clicks this button to add selected item to bill and open this form again to add further food items

User clicks this button when adding the food items has been completed.

User selects the quantity of food item selected to be added to the bill

Closes the window



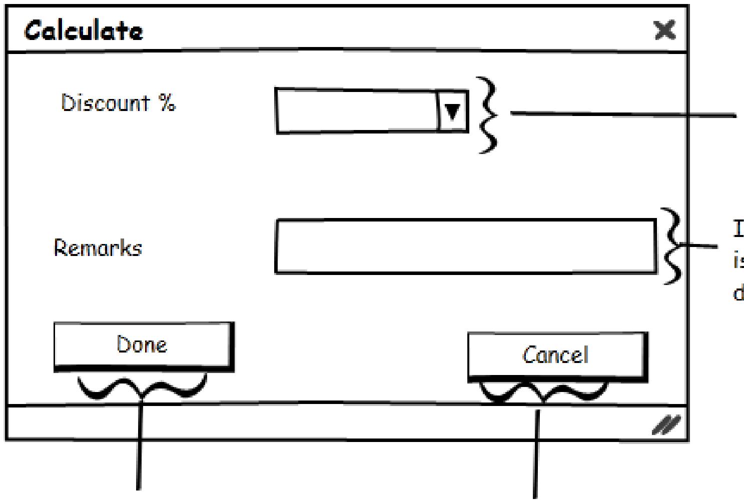
Category Name: Chowmin

	ItemName	Price
	Chicken Chowmin	100
▶	Mutton Chowmin	125
	Veg Chowmin	75

Quantity 1

Methods	Purpose
Private Sub ViewFoodItems_Load()	To load the food items name, their prices and their category from database
Private Sub cboCategoryName_SelectedValueChanged()	When user chooses another category, the items shown in data grid is changed
Private Sub btnCancel_Click()	Closes the form
Private Sub btnAdd_Click()	Transfers the name , price and quantity of selected item into main bill
Private Sub btnDone_Click()	Returns the control to calling form(main bill)

Calculate Screen

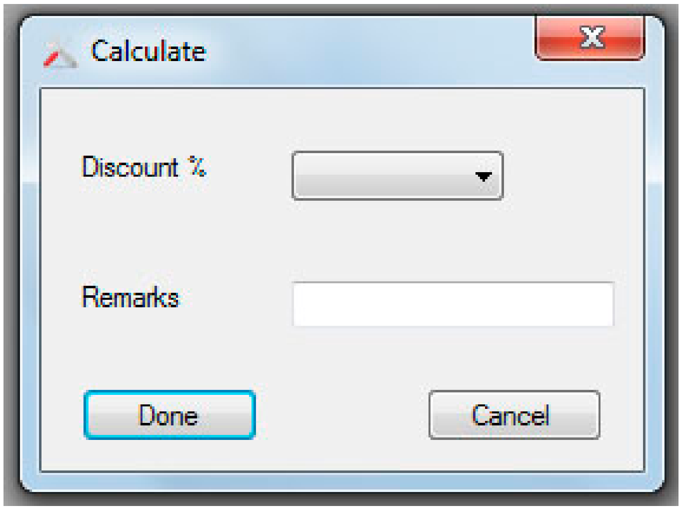


User selects the discount percentage for the customer.

In some cases, remark is required to clarify the discount %.

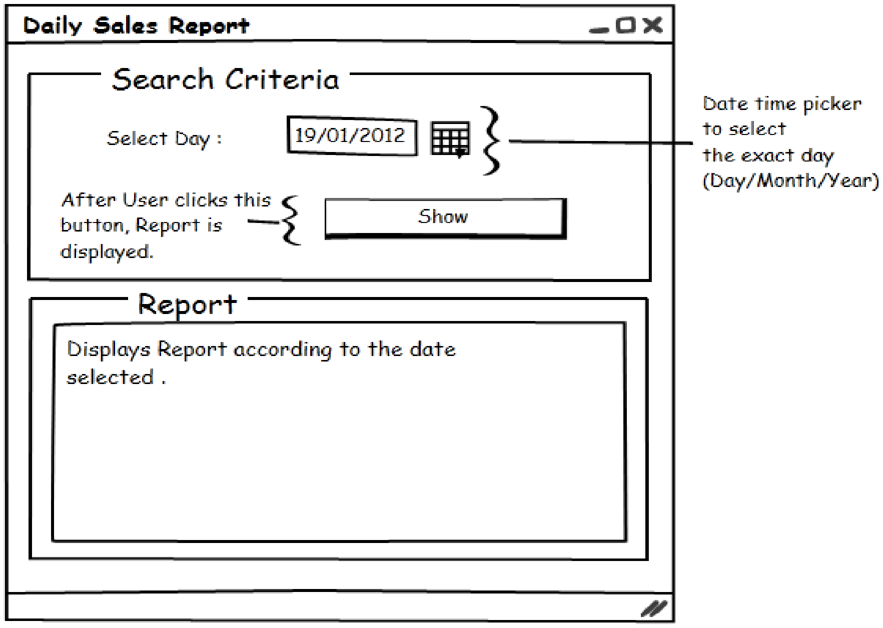
It clicked after therequired field(Discount% & Remarks) has been filled.

Button that closes the Calculate window.



Methods	Purpose
Private Sub btnDone_Click()	Validates the combo-box and returns the control to calling form
Private Sub Cancel_Click()	Closes the form

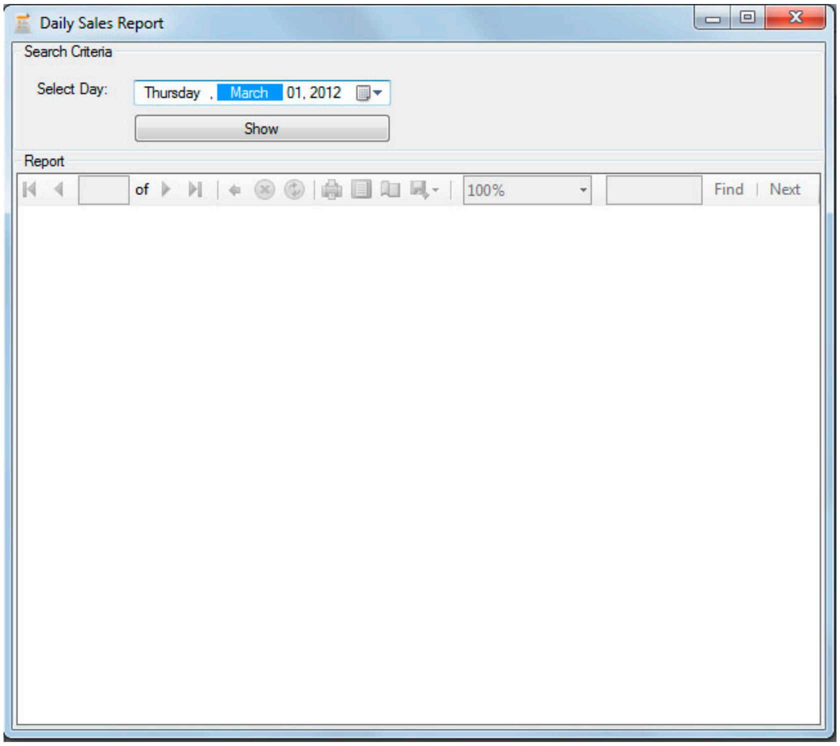
Daily Sales Report



Date time picker to select the exact day (Day/Month/Year)

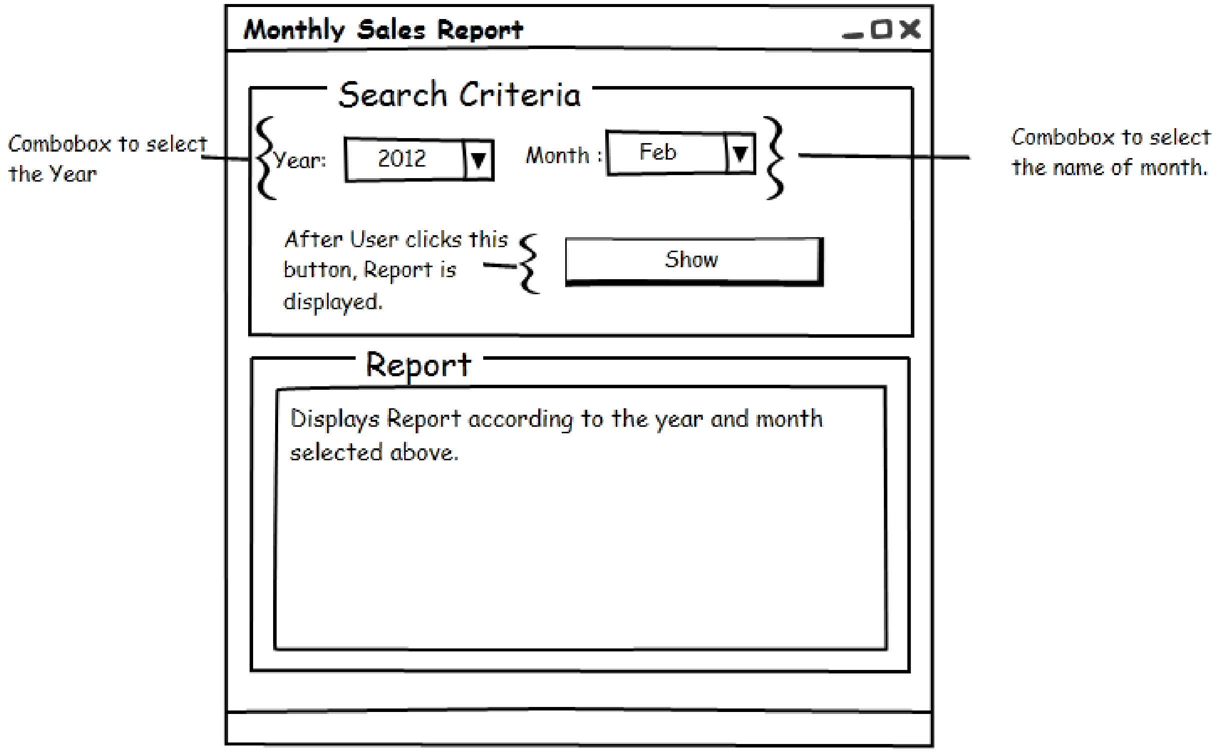
After User clicks this button, Report is displayed.

Displays Report according to the date selected .



Methods	Purpose
Private Sub btnShow_Click()	Displays the report according to selected date

Monthly Sales Report

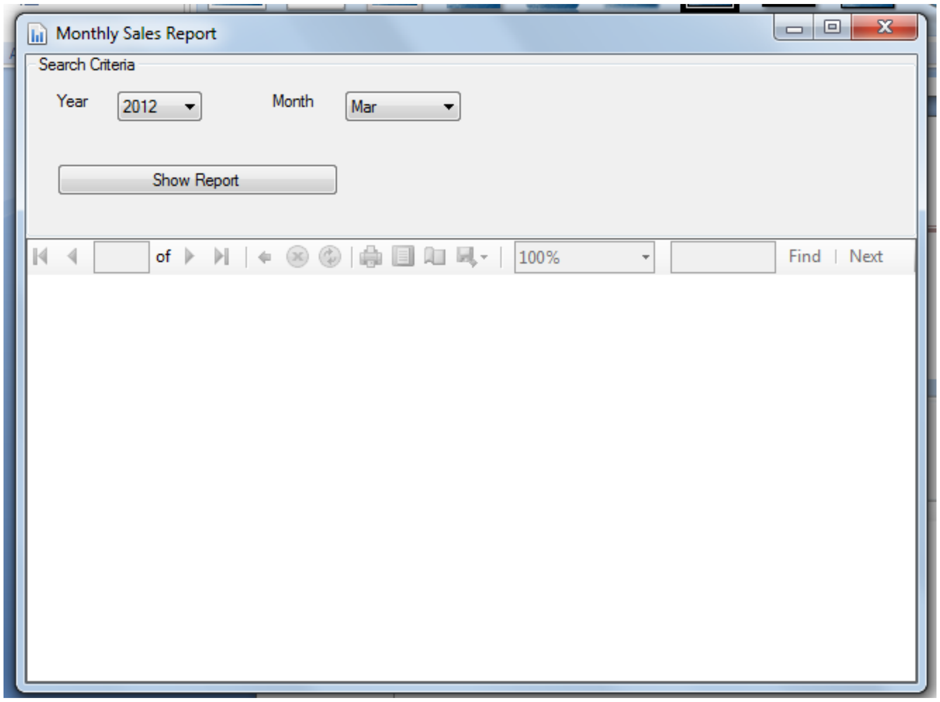


Combobox to select the Year

Combobox to select the name of month.

After User clicks this button, Report is displayed.

Displays Report according to the year and month selected above.



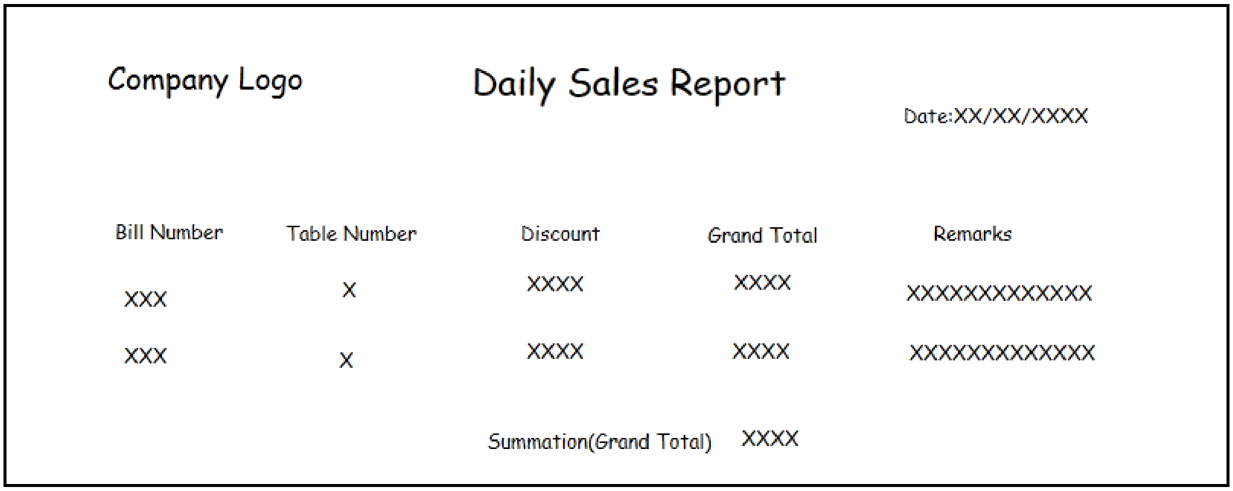
Year 2012

Month Mar

Methods	Purpose
Private Sub btnShow_Click()	Displays the monthly sales report according to the

	selected year and month

Daily Sales Report



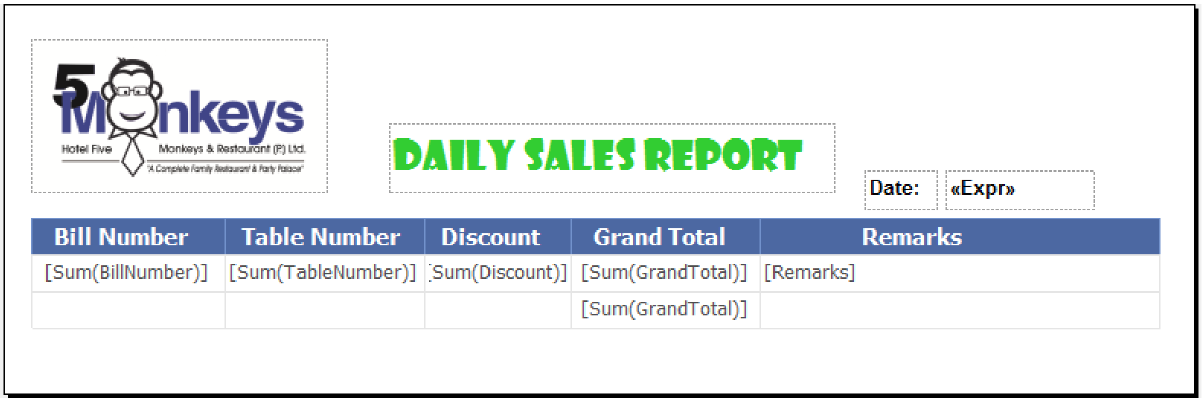
Company Logo

Daily Sales Report

Date:XX/XX/XXXX

Bill Number	Table Number	Discount	Grand Total	Remarks
XXX	X	XXXX	XXXX	XXXXXXXXXXXXX
XXX	X	XXXX	XXXX	XXXXXXXXXXXXX

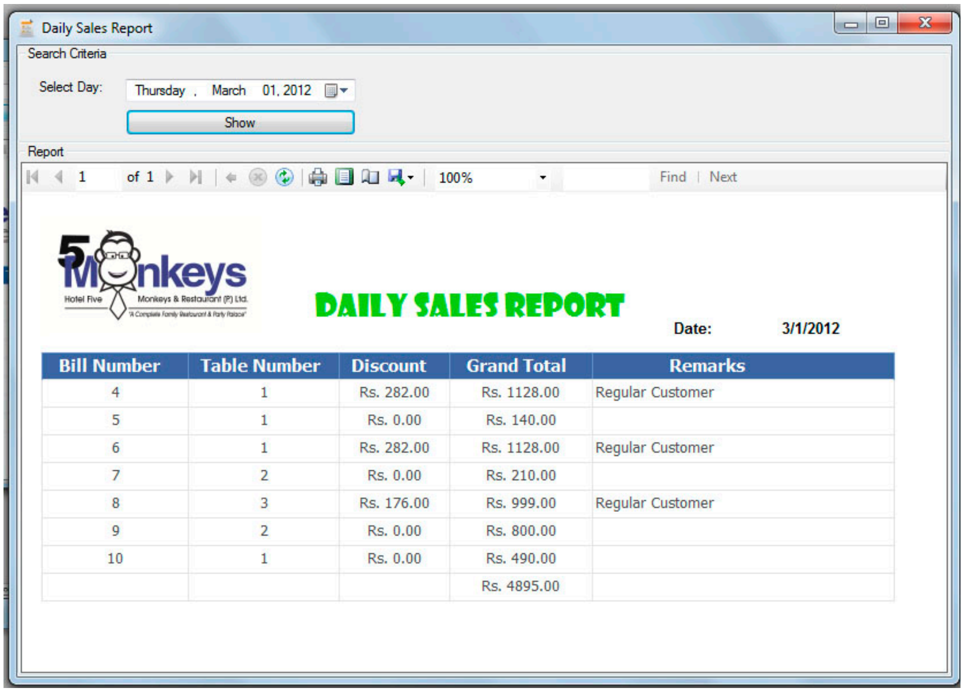
Summation(Grand Total) XXXX



DAILY SALES REPORT

Date: «Expr»

Bill Number	Table Number	Discount	Grand Total	Remarks
[Sum(BillNumber)]	[Sum(TableNumber)]	[Sum(Discount)]	[Sum(GrandTotal)]	[Remarks]
			[Sum(GrandTotal)]	



Search Criteria

Select Day: Thursday , March 01, 2012

Report

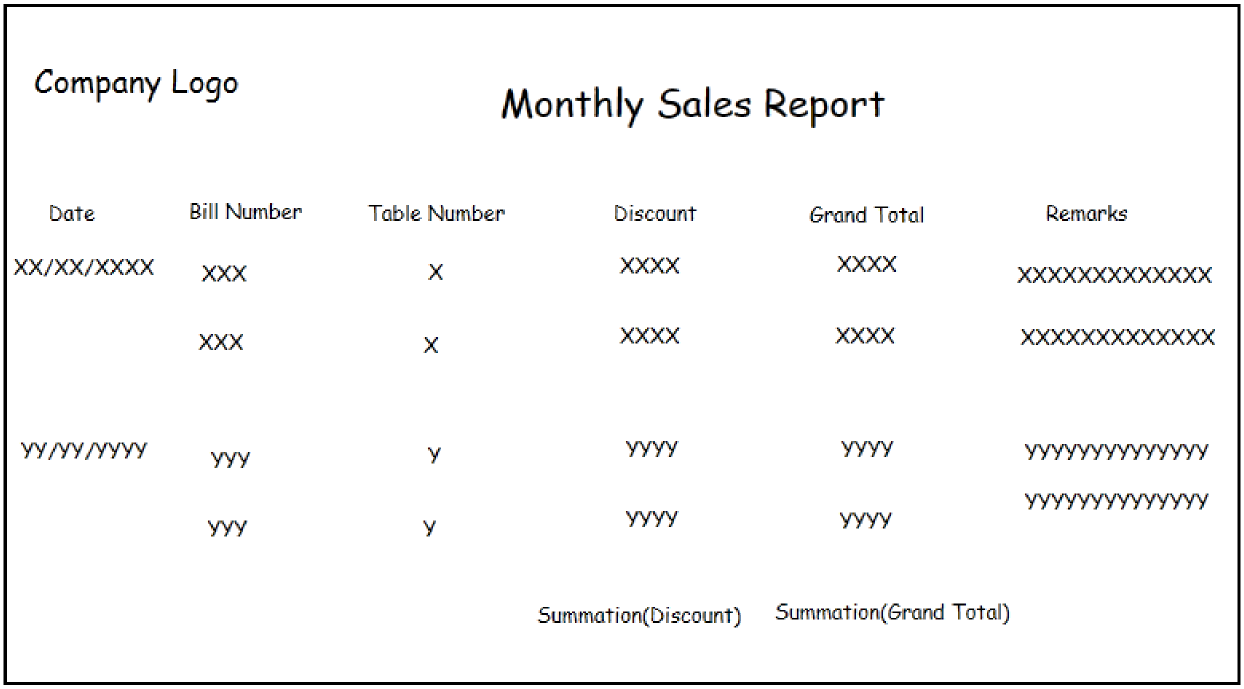
5 Monkeys
Hotel Five
Monkeys & Restaurant (P) Ltd.
"A Complete Family Restaurant & Party Palace"

DAILY SALES REPORT

Date: 3/1/2012

Bill Number	Table Number	Discount	Grand Total	Remarks
4	1	Rs. 282.00	Rs. 1128.00	Regular Customer
5	1	Rs. 0.00	Rs. 140.00	
6	1	Rs. 282.00	Rs. 1128.00	Regular Customer
7	2	Rs. 0.00	Rs. 210.00	
8	3	Rs. 176.00	Rs. 999.00	Regular Customer
9	2	Rs. 0.00	Rs. 800.00	
10	1	Rs. 0.00	Rs. 490.00	
			Rs. 4895.00	

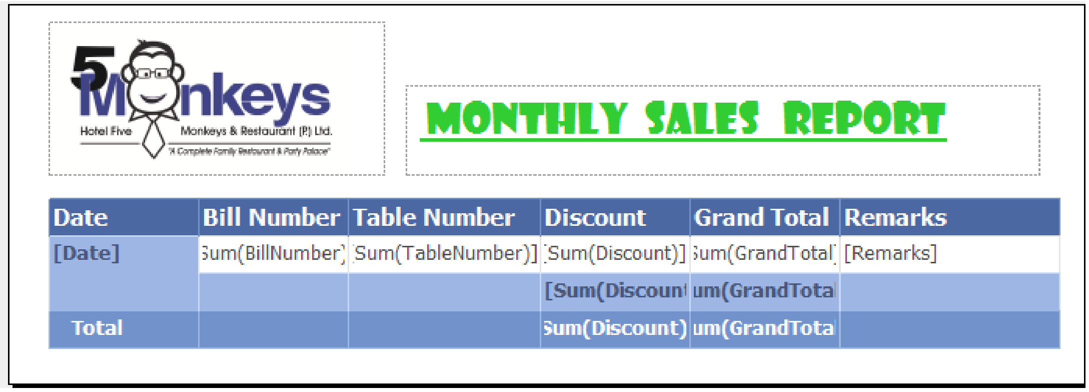
Monthly Sales Report



Company Logo

Monthly Sales Report

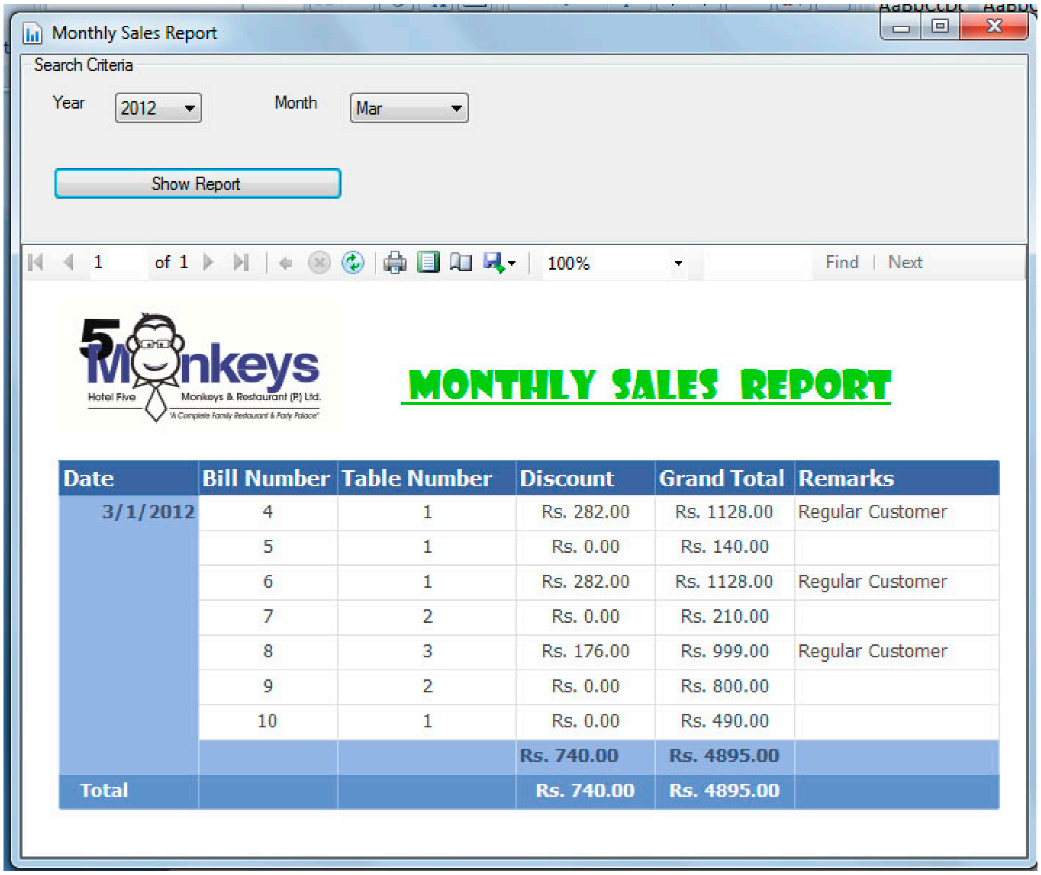
Date	Bill Number	Table Number	Discount	Grand Total	Remarks
XX/XX/XXXX	XXX	X	XXXX	XXXX	XXXXXXXXXXXXX
	XXX	X	XXXX	XXXX	XXXXXXXXXXXXX
YY/YY/YYYY	yyy	y	yyyy	yyyy	yyyyyyyyyyyyyy
	yyy	y	yyyy	yyyy	yyyyyyyyyyyyyy
			Summation(Discount)	Summation(Grand Total)	



5 Monkeys
Hotel Five
Monkeys & Restaurant (P) Ltd.
"A Complete Family Restaurant & Party Palace"

MONTHLY SALES REPORT

Date	Bill Number	Table Number	Discount	Grand Total	Remarks
[Date]	Sum(BillNumber)	Sum(TableNumber)]	Sum(Discount)]	Sum(GrandTotal]	[Remarks]
			[Sum(Discount	um(GrandTota	
Total			Sum(Discount)	um(GrandTota	



Year 2012

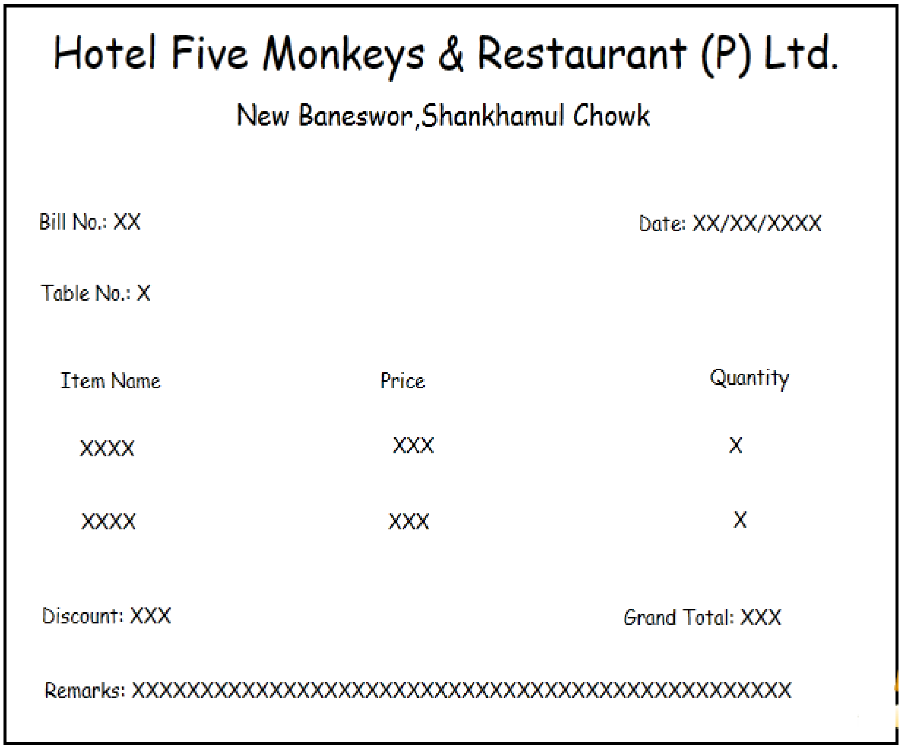
Month Mar

Show Report
1 of 1
100%
Find | Next
5 Monkeys
Hotel Five
Monkeys & Restaurant (P) Ltd.
"A Complete Family Restaurant & Party Palace"

MONTHLY SALES REPORT

Date	Bill Number	Table Number	Discount	Grand Total	Remarks
3/1/2012	4	1	Rs. 282.00	Rs. 1128.00	Regular Customer
	5	1	Rs. 0.00	Rs. 140.00	
	6	1	Rs. 282.00	Rs. 1128.00	Regular Customer
	7	2	Rs. 0.00	Rs. 210.00	
	8	3	Rs. 176.00	Rs. 999.00	Regular Customer
	9	2	Rs. 0.00	Rs. 800.00	
	10	1	Rs. 0.00	Rs. 490.00	
			Rs. 740.00	Rs. 4895.00	
Total			Rs. 740.00	Rs. 4895.00	

Bill Receipt



Hotel Five Monkeys & Restaurant (P) Ltd.

New Baneswor,Shankhamul Chowk

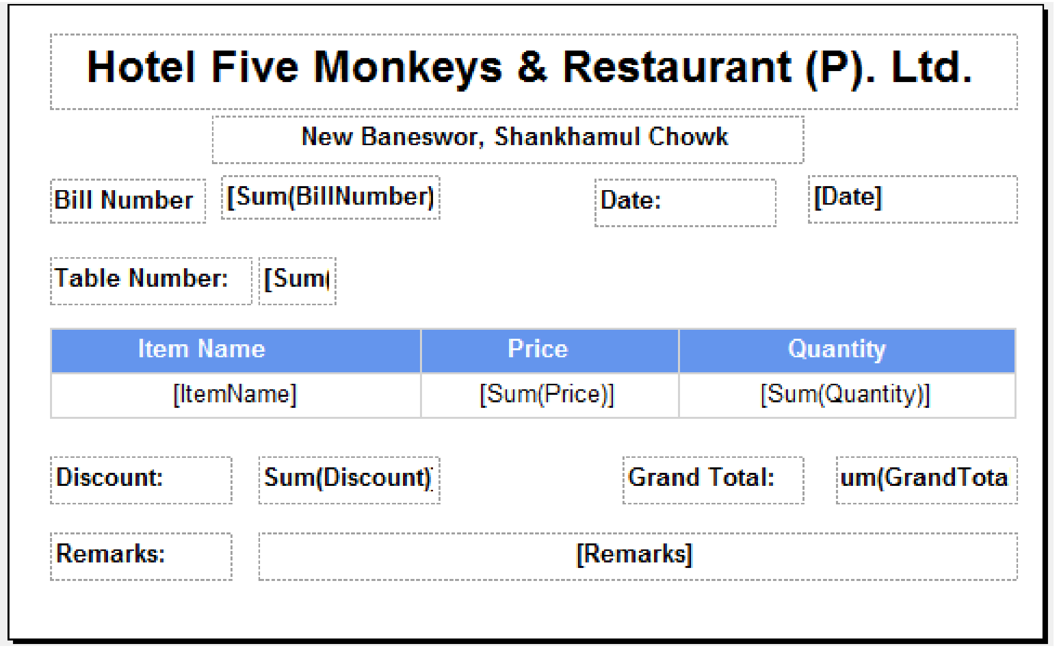
Bill No.: XX Date: XX/XX/XXXX

Table No.: X

Item Name	Price	Quantity
XXXX	XXX	X
XXXX	XXX	X

Discount: XXX Grand Total: XXX

Remarks: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



Hotel Five Monkeys & Restaurant (P). Ltd.

New Baneswor, Shankhamul Chowk

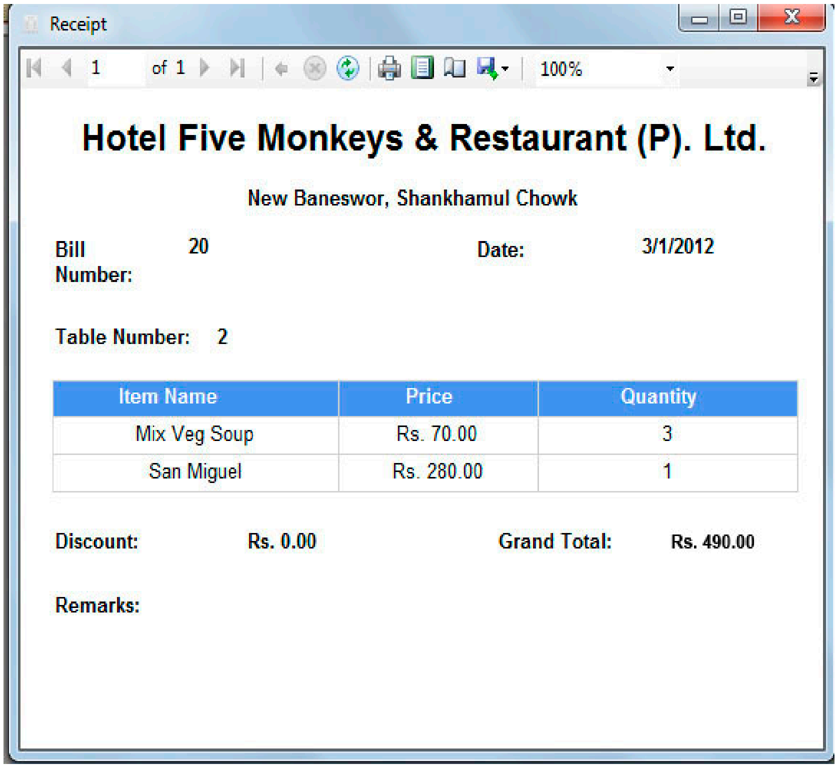
Bill Number [Sum(BillNumber) Date: [Date]

Table Number: [Sum(

Item Name	Price	Quantity
[ItemName]	[Sum(Price)]	[Sum(Quantity)]

Discount: Sum(Discount) Grand Total: um(GrandTota

Remarks: [Remarks]



Hotel Five Monkeys & Restaurant (P). Ltd.

New Baneswor, Shankhamul Chowk

Bill Number: 20 Date: 3/1/2012

Table Number: 2

Item Name	Price	Quantity
Mix Veg Soup	Rs. 70.00	3
San Miguel	Rs. 280.00	1

Discount: Rs. 0.00 Grand Total: Rs. 490.00

Remarks:

Methods	Purpose
Private Sub Receipt_Load()	Displays the customer receipt for food

Validator

Methods	Purpose
Public Shared Function IsPresent(ByVal textbox As TextBox) As Boolean	Checks if the text-box is empty
Public Shared Function IsDecimal(ByVal textbox As TextBox) As Boolean	Checks if the value entered in text-box is decimal
Public Shared Function IsDecimal(ByVal textbox As TextBox) As Boolean	Checks if the number entered in text-box lies between given interval
Public Shared Function LengthCheck(ByVal textbox As TextBox) As Boolean	Checks if the length of string is suitable or not

Possible Adaptive Maintenance

To further improve the system, the future work and updates could include:

- Using SQL Server as DBMS instead of Access as it provides more features which are suitable for business scale applications.
- Feature allowing individual share holders to keep track of their (individual) profit.
- Feature of keeping track of bar stock after each transaction has been made.
- Include Stock Control system to keep track of kitchen stock.
- Ability of automatically backing up the system in internal hard-disk and USB drive (if inserted) after certain interval.
- Ability of sending that backup file to an email or cloud storage automatically.
- Feature to take employee attendance and keep track of their salary.

Code Listing

Main

```
Imports System.IO


Public Class Main


    Public Sub Main_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Load



        '**************************************
        '*     Code Handling Log in form      *
        '**************************************
       TryAgain:
        Try
            '***code that executes if user clicks OK button in login form***

            'create a new instance of the class login form
            Dim loginForm As New Login

            ' Displays login form as a dialog box and stores the result it returns
            Dim result As DialogResult = loginForm.ShowDialog()

            'if user clickcs cancel button in login form the form is terminated
            If result = Windows.Forms.DialogResult.Cancel Then
                End
            End If

            'create a string which stores the password entered in the loginform
            Dim enteredPassword As String = loginForm.Tag
            'Database code to retreive the original password in the database

            Dim myTableAdapter As New
            billing_systemDataSetTableAdapters.UserLoginTableAdapter
            Dim myDataTable As DataTable =
            myTableAdapter.GetDataByUsername(loginForm.cboUsername.Text)

            'create a string which stores the original password in the database
            Dim originalPassword As String = myDataTable.Rows(0).Item(1)

            'compares entered password and orignial password
            If enteredPassword <> originalPassword Then
                    MessageBox.Show("Invalid password", "Error", MessageBoxButtons.OK,
                    MessageBoxIcon.Error)
                GoTo TryAgain

            End If


            'enables and disables certain features according to the user selected
            Dim selectedUsername As String = myDataTable.Rows(0).Item(0)
            Select Case selectedUsername
                Case "Receptionist"
                    Me.mnuMaintenance.Visible = False
                    Me.mnuReports.Visible = False
                    Me.btnBackup.Visible = False
                    Me.btnDailySales.Visible = False
                    Me.btnRestore.Visible = False
                    Me.btnMonthlySales.Visible = False
                    Me.ToolStripSeparator3.Visible = False
                    Me.ToolStripSeparator4.Visible = False
                    Me.RestoreToolStripMenuItem.Visible = False
                    Me.BackupToolStripMenuItem.Visible = False
                    Me.ReportsToolStripMenuItem.Visible = False
                    Exit Select
                Case "Administrator"
                    Me.mnuPOS.Visible = False
                    Me.mnuTables.Visible = False
                    Me.ToolStripSeparator1.Visible = False
                    Me.btnNewBill.Visible = False
                    Me.AddNewBillToolStripMenuItem.Visible = False
                    Me.mnuPOSNewBill.ShortcutKeys = Nothing
            End Select
        Catch ex As Exception
            'handles error
              MessageBox.Show("Invalid password", "Error", MessageBoxButtons.OK,
              MessageBoxIcon.Error)
            GoTo TryAgain
        End Try
    End Sub


    Private Sub mnuMaintenanceChangePassword_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles mnuMaintenanceChangePassword.Click
        ' displays change password form
        Dim passwordChanger As New ChangePassword
        passwordChanger.ShowDialog()
```

```
End Sub

Private Sub ViewFoodItems()
    'creates new instance of viewfooditems form and displays it
    Dim myViewFoodItems As New ViewFoodItems
    myViewFoodItems.ShowDialog()
End Sub

Private Sub mnuFoodItemsView_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFoodItemsView.Click
    'calls sub procedure that dispalys the food item categorically
    Me.ViewFoodItems()
End Sub

Private Sub btnViewMenu_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnViewMenu.Click
    'calls sub procedure that dispalys the food item categorically
    Me.ViewFoodItems()
End Sub

Private Sub EditFoodItems()
    'creates an instance of modify food items form
    Dim myModifyFoodItems As New ModifyFoodItems
    'displays the modifyfood item
    ModifyFoodItems.ShowDialog()
End Sub

Private Sub mnuFoodItemsEdit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFoodItemsEdit.Click
    'calls sub procedure that displays form to edit food items
    Me.EditFoodItems()
End Sub

Private Sub CreateNewBill()
    'displays the form that allows the user to select the table number
    Dim selectTable As New frmTables
    selectTable.ShowDialog()
End Sub

Private Sub mnuPOSNewBill_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuPOSNewBill.Click
    'calls sub procedure that creates and handles new bill
    Me.CreateNewBill()
End Sub

Private Sub btnNewBill_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNewBill.Click
    'call sub procedure that creates and handles new bill
    Me.CreateNewBill()
End Sub

Private Sub mnuExtraCalculator_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuExtraCalculator.Click
    'opens notepad
    Process.Start("notepad.exe")
End Sub
```

```
    Private Sub mnuExtraNotepad_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuExtraNotepad.Click
        'opens system calculator
        Process.Start("calc.exe")
    End Sub

    Private Sub MakeBackup()
        'sets the title of backup dialog box
        SaveFileDialog.Title = "Make Backup"

        'opens the make backup dialog box and performs backup if user specifies the file
       name and its path
        If SaveFileDialog.ShowDialog = Windows.Forms.DialogResult.OK Then

            Try
                'stores the path of original database in a string
                Dim portfolioPath As String = My.Application.Info.DirectoryPath
                'creates a file name specified in make backup dialog box
                File.Create(SaveFileDialog.FileName).Close()
                'copies the original database
                FileCopy(portfolioPath & "\billing_system.accdb",SaveFileDialog.FileName)
                'displays successful backup message
                MessageBox.Show("Backup file created", "Successful", MessageBoxButtons.OK,
              MessageBoxIcon.Informatio)
            Catch ex As Exception
                'handles the error by displaying error message
              MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
              MessageBoxIcon.Error)
            End Try
        End If
    End Sub

    Private Sub mnuMaintenanceBackup_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuMaintenanceBackup.Click
        'calls sub procedure that displays create backup form

        Me.MakeBackup()
    End Sub
```

Food Items

View Food Items

```
Public Class ViewFoodItems

       '**************************************
       '*   Code that displays food items    *
       '**************************************


       Private Sub ViewFoodItems_Load(ByVal sender As System.Object, ByVal e As
       System.EventArgs) Handles MyBase.Load
              ' This line of code loads data into the 'Billing_systemDataSet.Items_Table'
              table. You can move, or remove it, as needed.
```

```
               Me.Items_TableTableAdapter.Fill(Me.Billing_systemDataSet.Items_Table)
              ' This line of code loads data into the
              'Billing_systemDataSet.Category_Table' table. You can move, or remove it,
              as needed.
              Me.Category_TableTableAdapter.Fill(Me.Billing_systemDataSet.Category_Table)
              'fills the combobox with category names
              Me.Category_TableTableAdapter.Fill(Me.Billing_systemDataSet.Category_Table)


       End Sub

       Private Sub cboCategoryName_SelectedValueChanged(ByVal sender As Object, ByVal e
       As System.EventArgs) Handles cboCategoryName.SelectedValueChanged
               'displays the the items of category chosen
               Try

                     Me.Items_TableTableAdapter.FillByCategoryName(Me.Billing_systemDataS
                     et.Items_Table, cboCategoryName.Text)
               Catch ex As System.Exception
                     System.Windows.Forms.MessageBox.Show(ex.Message)
               End Try
           End Sub
End Class
```

Modify Food Items

```
Public Class ModifyFoodItems

       '***********************************************
       '*   Code that allows modifying food items    *
       '***********************************************

    Private Sub ModifyFoodItems_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        'This line of code loads data into the  'Billing_systemDataSet.Category_Table'
        table. You can move, or remove it, as needed.
        Me.Category_TableTableAdapter.Fill(Me.Billing_systemDataSet.Category_Table)
        'fills the combobox with category names
        Me.Category_TableTableAdapter.Fill(Me.Billing_systemDataSet.Category_Table)
    End Sub




    Private Sub cboCategoryName_SelectedValueChanged(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles cboCategoryName.SelectedValueChanged
        'displays the the items of category chosen
        Try

              Me.Items_TableTableAdapter.FillByCategoryName(Me.Billing_systemDataSet.Item
              s_Table, cboCategoryName.Text)
        Catch ex As System.Exception
            System.Windows.Forms.MessageBox.Show(ex.Message)
        End Try
    End Sub

    Private Sub btnAddNewItem_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnAddNewItem.Click
        Dim myAddNewItem As New AddFoodItem
        If myAddNewItem.ShowDialog = Windows.Forms.DialogResult.OK Then
              'shows the updated data after the addition in the datagrid
              Me.Items_TableTableAdapter.FillByCategoryName(Me.Billing_systemDataSet.Item
              s_Table, cboCategoryName.Text)
        End If
    End Sub


Private Sub btnRemoveSelectedItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRemoveSelectedItem.Click
  ' make sure a single item is being selected
        If DGV.SelectedRows.Count = 0 Then
            Exit Sub
        End If

        ' get the itemname of the item
        Dim itemName As String = DGV.SelectedRows(0).Cells(0).Value

        Try
            ' remove the item
            Items_TableTableAdapter.DeleteByItemName(itemName)
            'shows the updated data after the addition in the datagrid

              Me.Items_TableTableAdapter.FillByCategoryName(Me.Billing_systemDataSet.Item
              s_Table, cboCategoryName.Text)

        Catch ex As Exception
            MsgBox(ex.Message, MsgBoxStyle.Critical Or MsgBoxStyle.OkOnly, "Error")
        End Try
    End Sub

    Private Sub btnEditSelectedItem_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnEditSelectedItem.Click
        'make sure an item is selected
        If DGV.SelectedRows.Count = 0 Then
            Exit Sub
        End If

        ' get the food item name
        Dim itemName As String = DGV.SelectedRows(0).Cells(0).Value

        ' create the edit window
        Dim EditItemWindow As New EditSelectedItem

        ' fill the window with information
        EditItemWindow.FillItemInfo(itemName)

        If EditItemWindow.ShowDialog = Windows.Forms.DialogResult.OK Then

             'shows the updated data after the addition in the datagrid
```

```
              Me.Items_TableTableAdapter.FillByCategoryName(Me.Billing_systemDataSet.Item
              s_Table, cboCategoryName.Text)
        End If
    End Sub



    Private Sub btnAddCategory_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnAddCategory.Click
        'create an instace of add category table and displays it
        Dim myAddCategory As New AddCategory
        myAddCategory.ShowDialog()
        'shows the updated category name
        Me.Category_TableTableAdapter.Fill(Me.Billing_systemDataSet.Category_Table)
        'selects the category name recently added
        Me.cboCategoryName.SelectedIndex = cboCategoryName.Items.Count - 1
    End Sub

    Private Sub btnDeleteCategory_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnDeleteCategory.Click
        'creates a new table adapter
        Dim TA As New billing_systemDataSetTableAdapters.Category_TableTableAdapter
        'deletes the category name selected
        TA.Delete(cboCategoryName.Text)
        'displays the updated information
        Me.Category_TableTableAdapter.Fill(Me.Billing_systemDataSet.Category_Table)
    End Sub
End Class
```

Add Food Item

```
'Used to import the namespace that contains the validator class
Imports WindowsApplication1.Validator
Public Class AddFoodItem

       '********************************************
       '*   Code that is used to add food items    *
       '********************************************


    Private Sub txtAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnAdd.Click
        If IsValidData() Then
            Try
             'create a new table adapter of items table
              Dim myTableAdapter As New
              billing_systemDataSetTableAdapters.Items_TableTableAdapter
             'inserts the values entered in textobox to the database
              myTableAdapter.Insert(txtFoodName.Text, CDec(txtPrice.Text),
              txtCategory.Text)
              'returns result to the calling form as OK
              Me.DialogResult = Windows.Forms.DialogResult.OK
            Catch ex As Exception
              MessageBox.Show(ex.Message, "Error")
            End Try
        Else
            Exit Sub
        End If
    End Sub

    'Performs validation text using the validator class
    Private Function IsValidData() As Boolean
       Return IsPresent(txtFoodName) AndAlso IsPresent(txtPrice) AndAlso
       IsDecimal(txtPrice) AndAlso IsWithinRange(txtPrice, 10, 1500)
    End Function

    Private Sub AddFoodItem_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        txtCategory.Text = ModifyFoodItems.cboCategoryName.Text
    End Sub

    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCancel.Click
        Me.DialogResult = Windows.Forms.DialogResult.Cancel
    End Sub

End Class
```

Edit Selected Item

```
Imports WindowsApplication1.Validator
Public Class EditSelectedItem

       '***************************************
       '*      Code to edit food items        *
       '***************************************

    Private Sub EditSelectedItem_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        'fills the category name text box with selected category
        txtCategory.Text = ModifyFoodItems.cboCategoryName.Text
    End Sub

    'this method is used to fill the information
    Public Sub FillItemInfo(ByVal itemname As String)
        ItemsTA.FillByItemName(MyDS.Items_Table, itemname)
    End Sub

    'Performs validation text using the validator class
    Private Function IsValidData() As Boolean
       Return IsPresent(txtFoodName) AndAlso IsPresent(txtPrice) AndAlso
       IsDecimal(txtPrice) AndAlso IsWithinRange(txtPrice, 10, 1500)
    End Function

       Private Sub txtUpdate_Click(ByVal sender As System.Object, ByVal e As
       System.EventArgs)Handles btnUpdate.Click
        If IsValidData() Then
            Try
                ' create the adapter
                Dim TA As New billing_systemDataSetTableAdapters.Items_TableTableAdapter

                'insert the item
                BS.EndEdit()
                TA.Update(MyDS.Items_Table)

                'returns result to the calling form as OK
                Me.DialogResult = Windows.Forms.DialogResult.OK
            Catch ex As Exception
                MessageBox.Show(ex.Message, "Error")
            End Try
        Else
            Exit Sub
        End If
    End Sub

    Private Sub txtCancel_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCancel.Click
        'closes the edit dialog box
        Me.DialogResult = Windows.Forms.DialogResult.Cancel

    End Sub

End Class
```

Add Category

```
Public Class AddCategory

       '**************************************
       '*   Code that adds new category      *
       '**************************************

    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnAdd.Click

        Try
            'validates the text entered in the textbox
        If Validator.IsPresent(txtCategoryName) AndAlso
       Validator.LengthCheck(txtCategoryName) Then
                     'stores the text entered in textbox into variable named categoryName
                     Dim categoryName As String = txtCategoryName.Text
                     'creates an instance of table adapter for category table
                     Dim TA As New
                     billing_systemDataSetTableAdapters.Category_TableTableAdapter
                     'insert the entered text into category table
                TA.Insert(categoryName)
        End If
              'stores the name of category
              Me.Tag = txtCategoryName.Text
              'controls the transfer to the calling code
               Me.DialogResult = Windows.Forms.DialogResult.OK
        Catch ex As Exception
              MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
              MessageBoxIcon.Error)
        End Try
    End Sub

    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCancel.Click
        Me.DialogResult = Windows.Forms.DialogResult.Cancel
    End Sub

End Class
```

Bill

Select Table

```
Public Class frmTables

       '**************************************
       '*  Code that allows to select table  *
       '**************************************


    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnAdd.Click
        'create an instance of newbill table
        Dim newBill As New Bill
        'tells that main form is the parent form of newbill
        newBill.MdiParent = Main
        'sets the text property of newbill to the number selected in combobox
        newBill.Text = "Table Number " & cboTableNumber.Text
        newBill.Tag = cboTableNumber.Text

        'if option in combobox is not selected newform is not displayed
        If cboTableNumber.SelectedIndex <> -1 Then
            Main.mnuTables.Visible = True 'adds tables menu to the main form
            newBill.Show()
            Main.LayoutMdi(MdiLayout.TileVertical) 'arranages newform in main form
            'closes show table form
            Me.DialogResult = Windows.Forms.DialogResult.OK
        End If
 End Sub

End Class
```

Bill

```
Public Class Bill


    Private Sub btnAddItem_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnAddItem.Click
```

```
 '***************************************
 '*     Code Handling Add Item Form    *
 '***************************************

 'declare and initialise the variable which later stores the result sent by
additem bill
 Dim result As DialogResult = Nothing
 Try
     Do
         'creates an instance of add item form and displays it
         Dim addItem As New Additem

         'displays additem form
         result = addItem.ShowDialog()

         'if the user clicks ok or cancel button then the control is returned back
         to bill form
         If result = Windows.Forms.DialogResult.Cancel Or result =
         Windows.Forms.DialogResult.OK Then Exit Do

         '***Code that checks if the user selected item already exists in the bill
         form***

         'stores the item name of the selected row in variable named selecteditem
         Dim selectedItem As String = addItem.Tag
       'creates a boolean which is used later to tell if the items selected
       already exists in the bill
         Dim alreadyPresent As Boolean = False
       'Variable that stores the row of the item if it already exists in main bill
         Dim rowNumber As Integer
       'loops through each row and its column to check if selected item is already
       there
         For i As Integer = 0 To DGV.Rows.Count - 1
             If DGV.Rows(i).Cells(1).Value = selectedItem Then
                 alreadyPresent = True
                 rowNumber = i
                 Exit For
             End If
         Next

         'if the item already exists in the bill form then count and total is
         updated
         If alreadyPresent = True Then
             Dim count As Integer = DGV.Rows(rowNumber).Cells(4).Value
             DGV.Rows(rowNumber).Cells(4).Value = count +
             CInt(addItem.cboQuantity.Text)
             DGV.Rows(rowNumber).Cells(5).Value =
             (DGV.Rows(rowNumber).Cells(2).Value) * (count +
             CInt(addItem.cboQuantity.Text))
             Exit Sub

         End If


         '***If the item is not present in bill following lines of code run***

         'creates an instance of items table adapter
```

```
            Dim myTableAdapter As New
            billing_systemDataSetTableAdapters.Items_TableTableAdapter

            'uses table adapter to query the database for the item name sent from
            additem form
            Dim tb = myTableAdapter.GetDataByItemName(addItem.Tag)

            'displays the the item name, and other information in the datagrid
                 DGV.Rows.Add(tb.Rows(0).Item(0), tb.Rows(0).Item(1),
                 tb.Rows(0).Item(2), tb.Rows(0).Item(3),
                 CInt(addItem.cboQuantity.Text), CInt(addItem.cboQuantity.Text) *
                 CInt(tb.Rows(0).Item(2)))

        Loop While result = Windows.Forms.DialogResult.Retry
    Catch ex As Exception
          MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
          MessageBoxIcon.Error)

    End Try

End Sub


Private Sub btnRemoveItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRemoveItem.Click
    '***removes the selected row***
    If DGV.SelectedRows.Count = 0 Then
        Exit Sub
    End If
    'code to remove the selected row from the bill
    DGV.Rows.Remove(DGV.SelectedRows(0))
End Sub

Private Sub btnCalculate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCalculate.Click
    Try
        'displays error message if no item is present in data grid view
        If DGV.Rows.Count < 1 Then
                 MessageBox.Show("No Items Added", "Error", MessageBoxButtons.OK,
                 MessageBoxIcon.Exclamation)
            Exit Sub
        End If

        'activates the paid button
        btnPaid.Enabled = True

        'creates a new instance of calculate form
        Dim calculate As New Calculate
          'displays calculate form and performs further processes based on what user
          clicked in calculate form
        If calculate.ShowDialog = Windows.Forms.DialogResult.OK Then
            'declaring total variable
            Dim total As Integer
            'calculates total by looping through each row
            For i = 0 To DGV.Rows.Count - 1
                total += DGV.Rows(i).Cells(5).Value
            Next
```

```
                'Displays the required information of the bill in the bill form
                txtTotal.Text = total
               Dim discount As Integer = (CDec(calculate.cboDiscountPercentage.Text) /100)
                                                                                   * total
                txtDiscount.Text = discount
                txtGrandTotal.Text = (total - discount).ToString
                txtRemarks.Text = calculate.txtRemarks.Text
            End If
        Catch ex As Exception
            'handles the runtime error
              MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
              MessageBoxIcon.Error)
        End Try

    End Sub

    Private Sub btnPaid_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnPaid.Click


        '**************************************
        '*     Saves the bill record in DB    *
        '**************************************
        Dim MyConnection As OleDb.OleDbConnection = Nothing
        Dim MyTransaction As OleDb.OleDbTransaction = Nothing

        Try


            ' create the connection and  transaction object
              Dim connectionString As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data
              Source=|DataDirectory|\billing_system.accdb;Jet OLEDB:Database
              Password=thenewboston"
            MyConnection = New OleDb.OleDbConnection(connectionString)
            MyConnection.Open()
            MyTransaction = MyConnection.BeginTransaction

            ' insert the new recipt

            Dim SQL As String = "INSERT INTO CustomerBill_table ([Date], [TableNumber],
            Discount, [GrandTotal], Remarks)VALUES        (:0, :1, :2, :3, :4)"
            Dim CMD1 As New OleDb.OleDbCommand
            CMD1.Connection = MyConnection
            CMD1.Transaction = MyTransaction
            CMD1.CommandText = SQL
            CMD1.Parameters.AddWithValue(":0", Now.Date.Date)
            CMD1.Parameters.AddWithValue(":1", CInt(Me.Tag))
            CMD1.Parameters.AddWithValue(":2", CDec(txtDiscount.Text))
            CMD1.Parameters.AddWithValue(":3", CDec(txtGrandTotal.Text))
            CMD1.Parameters.AddWithValue(":4", txtRemarks.Text)
            CMD1.ExecuteNonQuery()
            CMD1.Dispose()

            ' get the id for the recipt
            SQL = "select max([BillNumber]) as MAXID from CustomerBill_Table"
            Dim CMD2 As New OleDb.OleDbCommand
            CMD2.Connection = MyConnection
            CMD2.Transaction = MyTransaction
            CMD2.CommandText = SQL
            Dim ReciptID As Long = CMD2.ExecuteScalar()
            CMD2.Dispose()

            ' insert the details of the receipt
            Dim I As Integer
            For I = 0 To DGV.Rows.Count - 1

                ' get the values from the data grid
                Dim itemID As String = DGV.Rows(I).Cells(0).Value
                Dim quantity As Decimal = DGV.Rows(I).Cells(4).Value
                Dim price As Decimal = DGV.Rows(I).Cells(2).Value


                ' create a command that extracts the food item
                Dim CMD3 As New OleDb.OleDbCommand
                SQL = "insert into BillDetails_Table " & _
                      "([BillNumber],ID,Quantity,Price) " & _
                      "values " & _
                      "(:0      ,:1     ,:2       ,:3   )"
                CMD3.Connection = MyConnection
                CMD3.Transaction = MyTransaction
                CMD3.CommandText = SQL
                CMD3.Parameters.AddWithValue(":0", ReciptID)
                CMD3.Parameters.AddWithValue(":1", itemID)
                CMD3.Parameters.AddWithValue(":2", quantity)
                CMD3.Parameters.AddWithValue(":3", price)


                CMD3.ExecuteNonQuery()
                CMD3.Dispose()

            Next


            ' all well save the changes
            MyTransaction.Commit()

            ' close connection
            MyTransaction.Dispose()
            MyConnection.Close()
            MyConnection.Dispose()



            'Displays the bill for printing
            Dim myreceipt As New Receipt
            myreceipt.Show()
            Me.Close()

        Catch ex As Exception
            '***handles the runtime error

            'if failure occurs previous state of data will be recovered
            If MyTransaction IsNot Nothing Then
                MyTransaction.Rollback()
```

```
            End If
            'closes the connection if it is still open
            If MyConnection IsNot Nothing Then
                If MyConnection.State = ConnectionState.Open Then
                    MyConnection.Close()
                End If
            End If

            MsgBox(ex.Message, MsgBoxStyle.Critical Or MsgBoxStyle.OkOnly, "Error")
        End Try
    End Sub


    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCancel.Click
        Me.Close()
    End Sub

    Private Sub Bill_Activated(ByVal sender As Object, ByVal e As System.EventArgs)
    Handles Me.Activated
        'displays the function of active mdichild form at in status bar
        Main.sbMain.Items(1).Text = "Processes the bill of " & Me.Text
    End Sub

End Class
```

Add Item

```
Public Class Additem

       '***************************************
       '*   Adding food item to main bill    *
       '***************************************


    Private Sub ViewFoodItems_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        'This line of code loads data into the 'Billing_systemDataSet.Category_Table'
        table. You can move, or remove it, as needed.
        Me.Category_TableTableAdapter.Fill(Me.Billing_systemDataSet.Category_Table)
        'fills the combobox with category names
        Me.Category_TableTableAdapter.Fill(Me.Billing_systemDataSet.Category_Table)
    End Sub




    Private Sub cboCategoryName_SelectedValueChanged(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles cboCategoryName.SelectedValueChanged
        'displays the the items of category chosen
        Try

              Me.Items_TableTableAdapter.FillByCategoryName(Me.Billing_systemDataSet.Item
              s_Table, cboCategoryName.Text)
        Catch ex As System.Exception
```

```
              System.Windows.Forms.MessageBox.Show(ex.Message)
        End Try
    End Sub



    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCancel.Click
        'close this form
        Me.Close()
    End Sub

    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnAdd.Click

        '***sends the name of the selected food item to the calling form***'

        'If none data is selected no processing will be done
        If Not DGV1.SelectedRows.Count = 0 Then
              'sets the tag property of the current form to the item name of the
              seleceted row
            Me.Tag = DGV1.SelectedRows(0).Cells(1).Value
            'If users don't select the quantity number then error is displayed
            If cboQuantity.SelectedIndex = -1 Then
              MessageBox.Show("You must select Quantity", "Data Missing",
              MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
              cboQuantity.Focus()
              Exit Sub
            End If
            'returns the control to the calling form
            DialogResult = Windows.Forms.DialogResult.Retry

        End If
    End Sub


    Private Sub btnDone_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnDone.Click
        'returns the control to the calling form
        Me.DialogResult = Windows.Forms.DialogResult.OK
    End Sub
End Class
```

Calculate

```
Public Class Calculate
       '*****************************************
       '* Code that performs bill calculation  *
       '*****************************************


    Private Sub btnDone_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnDone.Click
        'validates the combobox and returns the control to calling form
        If cboDiscountPercentage.SelectedIndex <> -1 Then
```

```
            Me.DialogResult = Windows.Forms.DialogResult.OK
        End If
    End Sub

    Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles Cancel.Click
        'returns the control to calling form
        Me.DialogResult = Windows.Forms.DialogResult.Cancel
    End Sub
End Class
```

Password

Log-in

```
       '**************************************
       '*   Code that handles log-in form    *
       '**************************************



Public Class Login

    Private Sub Login_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
        ' This line of code loads data into the 'Billing_systemDataSet.UserLogin' table.
        You can move, or remove it, as needed.
        Me.UserLoginTableAdapter.Fill(Me.Billing_systemDataSet.UserLogin)
        ' This line of code loads data into the 'Billing_systemDataSet.UserLogin' table
        Me.UserLoginTableAdapter.Fill(Me.Billing_systemDataSet.UserLogin)
    End Sub

    Private Sub btnLogIn_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnLogIn.Click
        'stores the entered password for validation
        Me.DialogResult = Windows.Forms.DialogResult.OK
        Me.Tag = txtPassword.Text
    End Sub

    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCancel.Click
        'closes the form
        Me.DialogResult = Windows.Forms.DialogResult.Cancel
    End Sub

End Class
```

Change Password

```
Public Class ChangePassword
```

```
Private Sub btnOk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnOk.Click
    '**************************************
    '*   Code Handling Password Changer   *
    '**************************************
    Try


        ' validate the new password
        If txtNewPassword.Text = "" Then
          MessageBox.Show("You should enter a password", "Error",
          MessageBoxButtons.OK, MessageBoxIcon.Error)
          txtNewPassword.Focus()
          Exit Sub
        End If

          If txtNewPassword.Text <> txtNewPasswordAgain.Text Then
            MessageBox.Show("The password does not match its confirmation", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error)
            txtNewPassword.Focus()
            Exit Sub
        End If

        ' get the original password
        Dim myTableAdapter As New
        billing_systemDataSetTableAdapters.UserLoginTableAdapter
        Dim myDataTable As DataTable =
        myTableAdapter.GetDataByUsername(cboUsername.Text)
        Dim originalPassword As String = myDataTable.Rows(0).Item(1)

        'check if the old password is correct
        If originalPassword <> txtOldPassword.Text Then
           MessageBox.Show("The old password is incorrect", "Error",
          MessageBoxButtons.OK, MessageBoxIcon.Error)
          txtOldPassword.Focus()
          Exit Sub
        End If

        ' everything went well, so we are going to update the password in the db
          Dim rowChanged As Integer =
        myTableAdapter.UpdatePassword(txtNewPassword.Text, cboUsername.Text)
        If rowChanged = 0 Then
          MessageBox.Show("Password not updated", "Error", MessageBoxButtons.OK,
          MessageBoxIcon.Error)
          Exit Sub
        End If

        Me.DialogResult = Windows.Forms.DialogResult.OK


    Catch ex As Exception
        ' handle the error
        MsgBox(ex.Message, MsgBoxStyle.Critical Or MsgBoxStyle.OkOnly, "Error")
        Exit Sub
    End Try
End Sub
```

```
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCancel.Click
        'user cancels the change
        Me.DialogResult = Windows.Forms.DialogResult.Cancel
    End Sub

    Private Sub ChangePassword_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        'This line of code loads data into the 'Billing_systemDataSet.UserLogin' table.
        You can move, or remove it, as needed.
        Me.UserLoginTableAdapter.Fill(Me.Billing_systemDataSet.UserLogin)
        ' This line of code loads data into the 'Billing_systemDataSet.UserLogin' table.
        Me.UserLoginTableAdapter.Fill(Me.Billing_systemDataSet.UserLogin)
 End Sub

End Class
```

Reports

Daily Sales Report

```
Public Class DailySalesReport
       '*****************************************************
       '*   Code that fills daily sales report to dataset   *
       '*****************************************************


    Private Sub DailySalesReport_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
    'This line of code loads data into the 'billing_systemDataSet.CustomerBill_table'
    table. You can move, or remove it, as needed.
    Me.CustomerBill_tableTableAdapter.Fill(Me.billing_systemDataSet.CustomerBill_table)
    End Sub

    Private Sub btnShow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnShow.Click
    'This line of code loads data into the 'billing_systemDataSet.CustomerBill_table'
    table. You can move, or remove it, as needed.

Me.CustomerBill_tableTableAdapter.FillByDate(Me.billing_systemDataSet.CustomerBill_table,
DateTimePicker1.Value.Date)
        Me.rvDaily.RefreshReport()
    End Sub

End Class
```

Monthly Sales Report

```
Public Class MonthlySalesReport
       '*****************************************************
       '*   Code that fills monthly sales report to dataset  *
       '*****************************************************
```

```
    Private Sub btnShow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnShow.Click
        'checks whether user has selected year and month or not
        If cboYear.SelectedIndex = -1 OrElse cboMonth.SelectedIndex = -1 Then
        MessageBox.Show("Year and Month must be selected", "Error", MessageBoxButtons.OK,
       MessageBoxIcon.Exclamation)
            Exit Sub
        End If
        ' This line of code loads data into the
       'billing_systemDataSet.CustomerBill_table' table. You can move, or remove it, as
       needed.
       Me.CustomerBill_tableTableAdapter.FillByYearandMonth(Me.billing_systemDataSet.Cust
       omerBill_table, cboYear.Text, cboMonth.SelectedIndex + 1)
       Me.rvMonthly.RefreshReport()
    End Sub

End Class
```

Receipt

```
Public Class Receipt

    Private Sub Receipt_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
        Dim MyConnection As OleDb.OleDbConnection = Nothing
       Dim connectionString As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data
       Source=|DataDirectory|\billing_system.accdb;Jet OLEDB:Database
       Password=thenewboston"
       MyConnection = New OleDb.OleDbConnection(connectionString)
       MyConnection.Open()
       ' get the id for the recipt
       Dim Sql As String = "select max([BillNumber]) as MAXID from CustomerBill_Table"
       Dim CMD2 As New OleDb.OleDbCommand
       CMD2.Connection = MyConnection
       CMD2.CommandText = Sql
       Dim ReciptID As Integer = CMD2.ExecuteScalar()
       CMD2.Dispose()
       MyConnection.Close()

      ' This line of code loads data into the 'billing_systemDataSet.Receipt' table. You
      can move, or remove it, as needed.
        Me.ReceiptTableAdapter.Fill(Me.billing_systemDataSet.Receipt, ReciptID)
        Me.rvReceipt.RefreshReport()
    End Sub

End Class
```

Food Menu

```
       '******************************************************
       '*    Code that fills food menu report to dataset     *
       '******************************************************
```

```
Public Class FoodMenu

    Private Sub FoodMenu_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
       ' This line of code loads data into the
       'billing_systemDataSet.CustomerBillForReport' table. You can move, or remove it,
       as needed.
       Me.Items_TableTableAdapter.Fill(Me.billing_systemDataSet.Items_Table)
       Me.ReportViewer1.RefreshReport()
    End Sub
End Class
```