XTRA- WINDOORS

Project Documentation

BIS 635: Business Systems Applications

**Submitted By: Group 7**

VAISHNAVI PEDDU

RAVI BHAGYAPALLI

NA LI

TURKEY ALSOHAIBANI

ANOOP MADIPALLI

**INSTRUCTOR:**

Dr. Zhenyu Huang

**Background:**

One of the most widely used services in the United States is Storage rooms. The business makes good profits as few people prefer to use these services to store goods. The storage rooms are spread over a large area and will be under surveillance continuously round the clock. This facility is economical and highly secured as it avoids paying a huge rental amount for storing goods by leasing an apartment.

One of the companies in Mt Pleasant, Michigan named AAA Lockup Mini-Storage is a small business that provides storage facility. There are 350 storage units with 7 different sizes with variation in prices. The storage rooms can be leased in monthly basis or even for a long term depending on the requirement. Graig Preffer, the owner of this company is still following the traditional way of using papers in running a business. The company has a 24 hour surveillance over CCTV to ensure safety and security. The company also has few added services like providing padlocks and keys system that enhances security.

**Business Problems:**

After looking into the business and its functions, it is found that AAA Lockup Mini-Storage Company is using outdated system to enter the details of the customer. The owner is still using paper work to enter the leasing details. Few of the business problems are listed below:

* It is hard to track the leasing details.
* Search would be too long to retrieve any information from previous records.
* Has a lot of paper work that should be done prior to signing the lease.
* Checking the availability of the rooms is manual.
* Piling up of papers and space occupancy is another issue to be considered.
* Customer authentication is one of the difficult tasks.
* Customer has to be present in person for any booking or extending lease.
* There is high chance of data duplication.

**Project description:**

AAA Lockup services is providing storage rooms for people around Mt Pleasant. The main objective of the project is to develop a system that minimizes paper work to the maximum. Various aspects such as usability, ability to render customer needs, economical and technical feasibility will be considered in this system.

The major advantage of this system is that it reduces the paper work thus reducing the physical storage needs and cost involved in maintaining them. The data can be accessed, stored, shared and maintained for future reference. The project focuses on checking the availability of the rooms so that there would not be any overlapping of lease. The application has the flexibility to edit the data entered thus providing greater control of storing error free data. Once this system is implemented the owner can have a better access of data and can make his business more efficient.

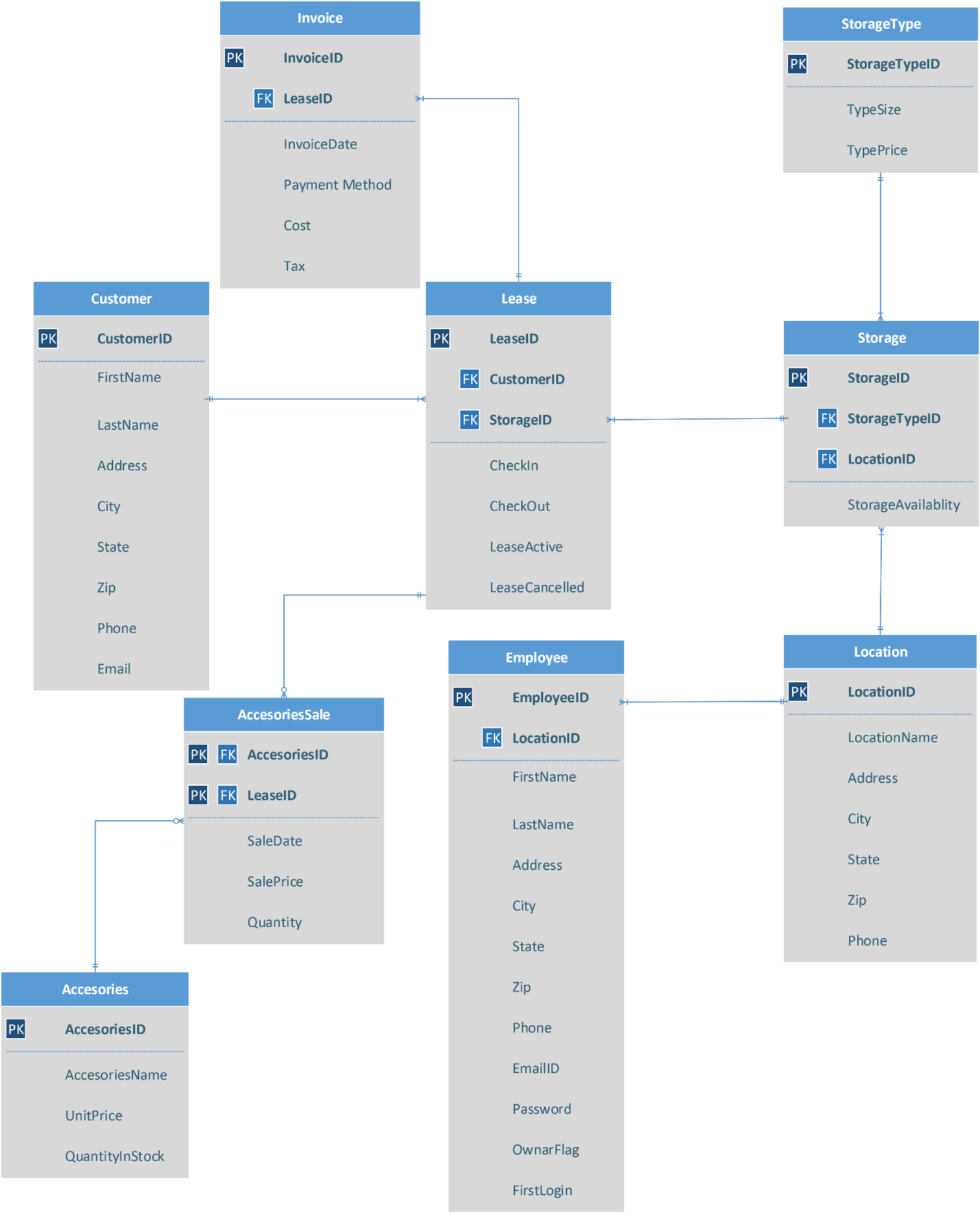
**Project feasibility:**

The proposed system uses Visual basic Development Environment to develop the front end and a Windows database application on the back end of the data. This application runs in windows platform. This application can be accessed only with valid credentials. Security is ensured as the system runs only with valid credentials. The application is user friendly and does not require much training to use it. The application can be executed as a stand-alone application that runs on Windows Operating System.

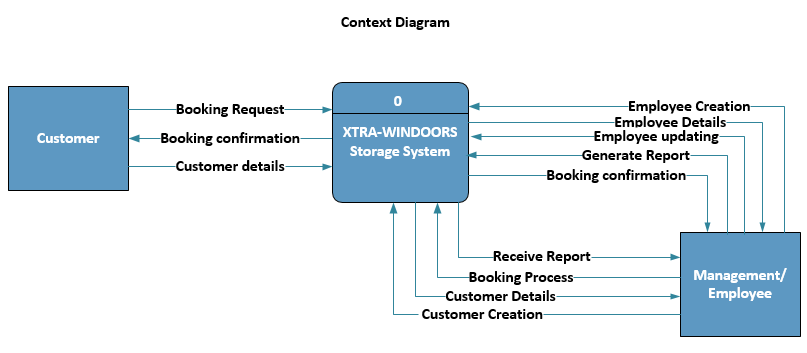
**Section II**

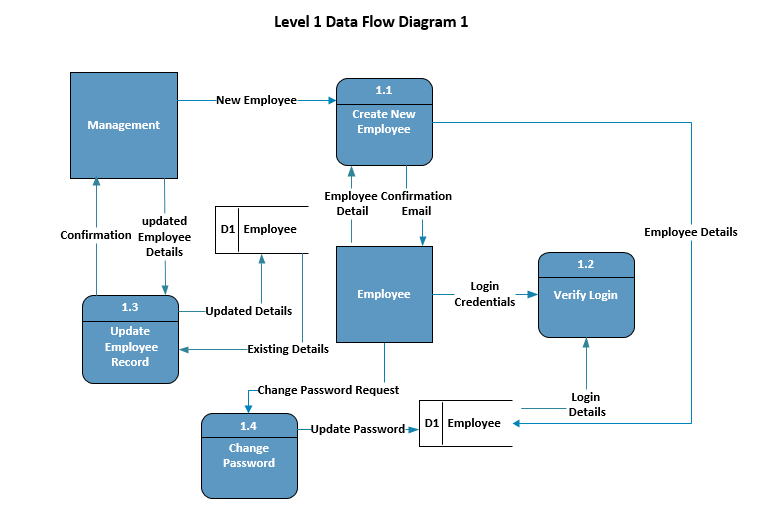
**System Design**

**ERD**



**DFD**

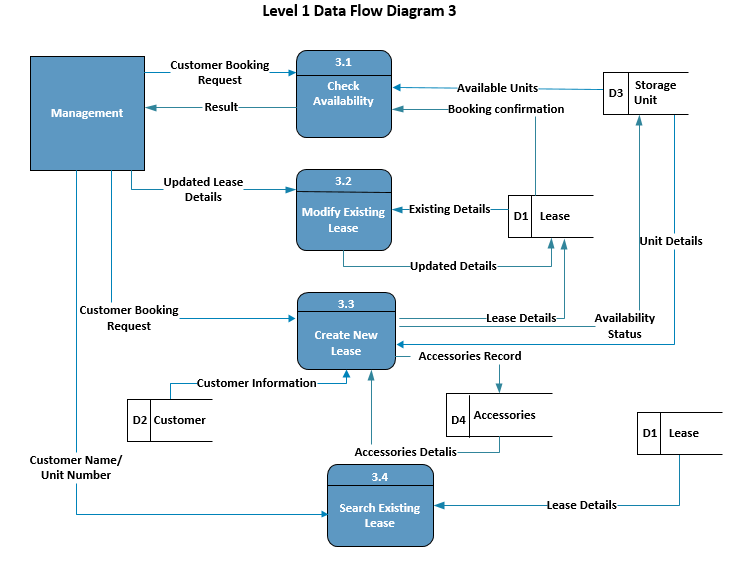
* **Context Diagram**
* **Level 0 Data Flow Diagram**
* **Level 1 Data Flow Diagram 1**

****

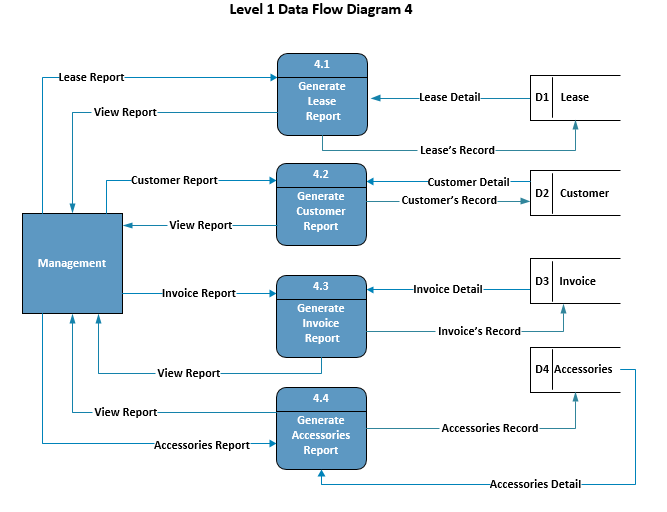
* **Level 1 Data Flow Diagram 2**



* **Level 1 Data Flow Diagram 3**



* **Level 1 Data Flow Diagram 4**



**Section III**

**System Manual**

**Splash Screen:**

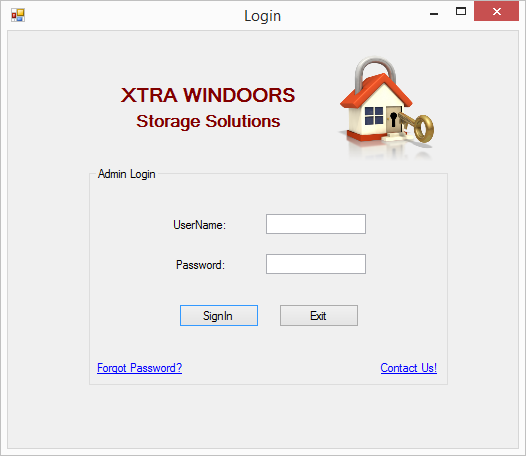
When the employee starts the program, a splash screen pops up with a progress bar and status text indicating the progress of the application start up.



**Figure 1** Splash Screen

**Login Screen:**

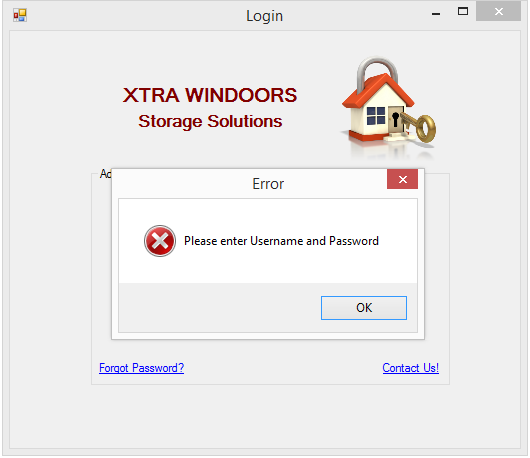
Immediately after the Splash Screen closes, login screen appears



**Figure 2** Login Screen

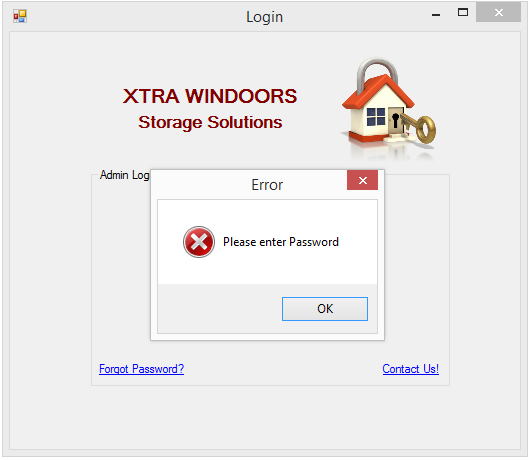
**Login Screen Validations:**

If the employee does not enter Username and Password then a error message appears asking to enter Username and Password.



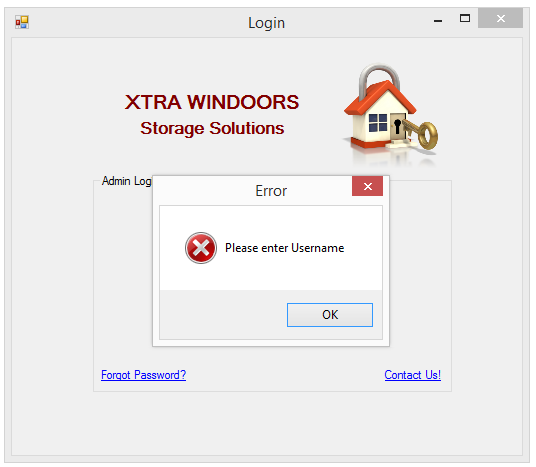
**Figure 3** Login Screen Validation (Enter Username and Password)

If the employee does not enter the password field then the following error message appears on the screen.



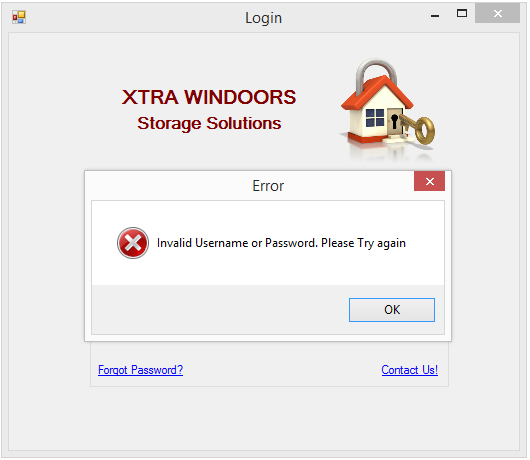
**Figure 4** Login Screen Validation (Enter Password)

If the employee does not enter Username field then the following error message appears on the screen.



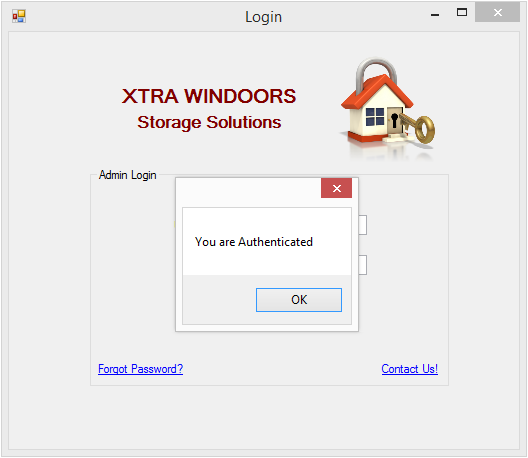
**Figure 5** Login Screen Validation (Enter Username)

If the employee does enter invalid login credentials, then the following error message appears on the screen.



**Figure 6** Login Screen Validation (Invalid Login Credentials)

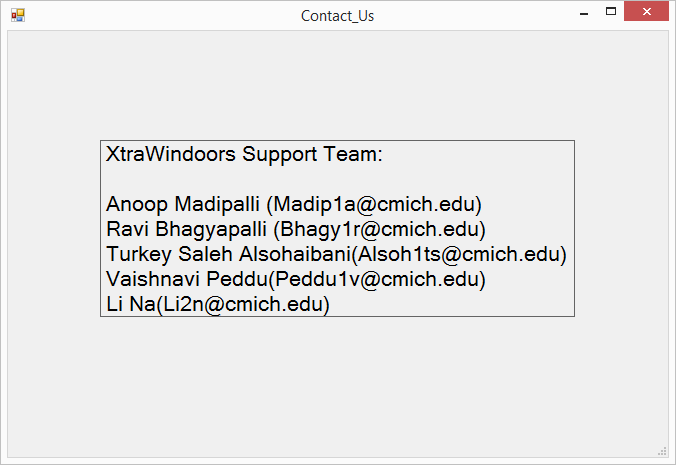
If the employee successfully enter login credentials, then a welcome message appears on the screen.



**Figure 7** Login Screen Validation (Successful Validation of Login Credentials)

**Contact Us Screen:**

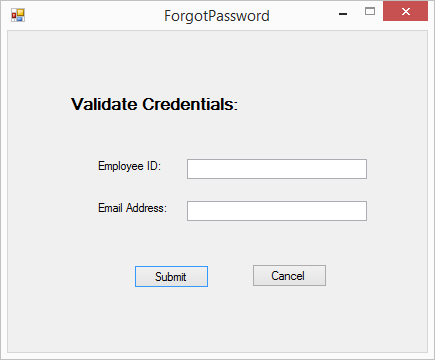
When the employee needs any support before signing in, then they can click on the Contact Us label on the login screen and a new screen opens up with the contact email address of the admins.



**Figure 8** Contact Us Screen

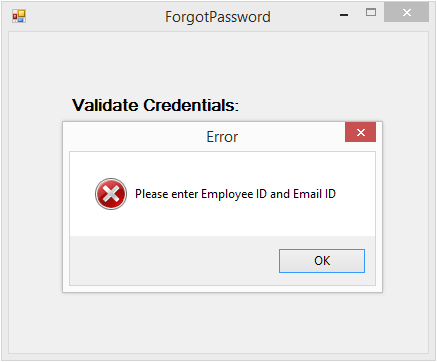
**Forgot Password:**

When the employee forgot his or her password then upon clicking the Forgot Password label they will be re directed to this screen.



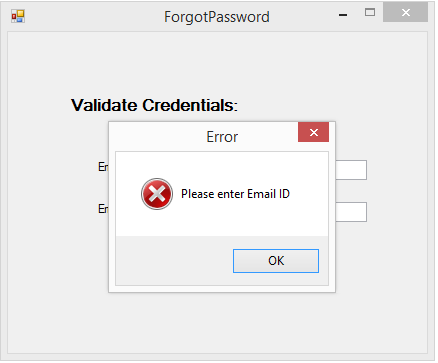
**Figure 9** Forgot password Screen

When Employee does not enter details and click submit button



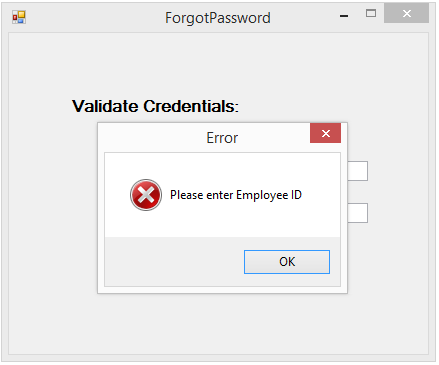
**Figure 10** Forgot Password Validation (Enter Employee Id and email ID)

When employee does not enter email ID and submits the form, the following message appears on screen



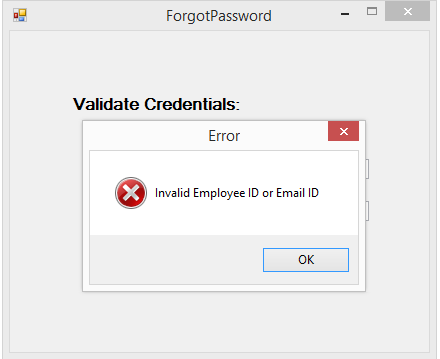
**Figure 11** Forgot Password Validations (Enter Email ID)

When employee does not enter employee ID and submits the form, the following message appears on screen



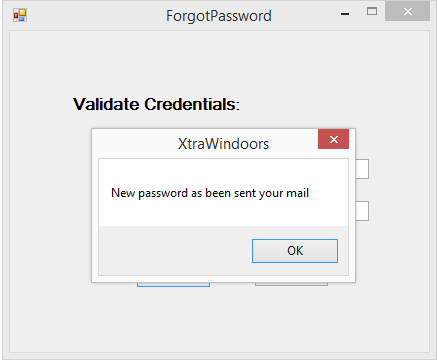
**Figure 12** Forgot Password Validations (Enter Employee ID)

When employee enters invalid employee ID or email ID and submits the form, the following message appears on screen



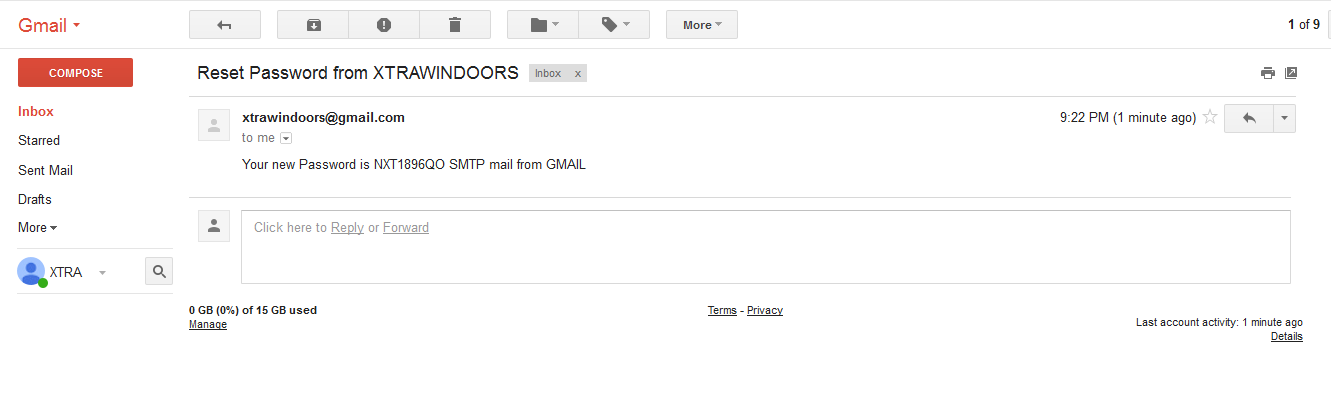
**Figure 13** Forgot Password Validations (Invalid Details Provided)

When employee enters valid details and submits the form, the following message appears on screen



**Figure 14** Forgot Password Validations (New Password Mail Confirmation)

Email with employee new password sent to employee authorized email address

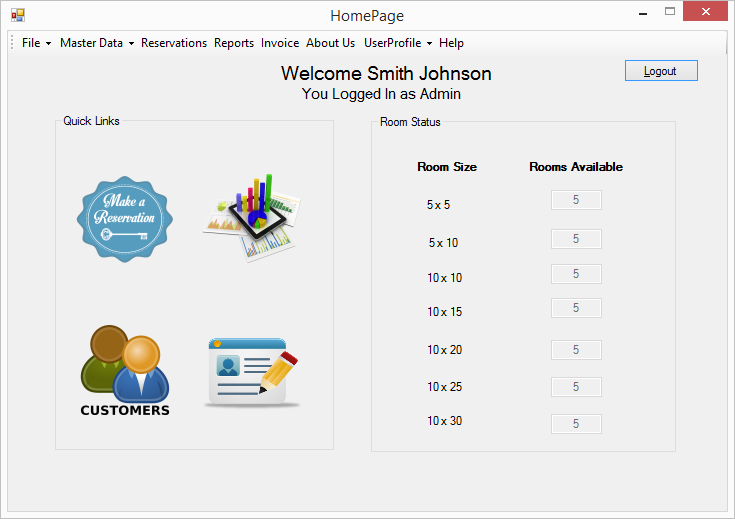


**Figure 15** Email with New Password Details

**Home Screen:**

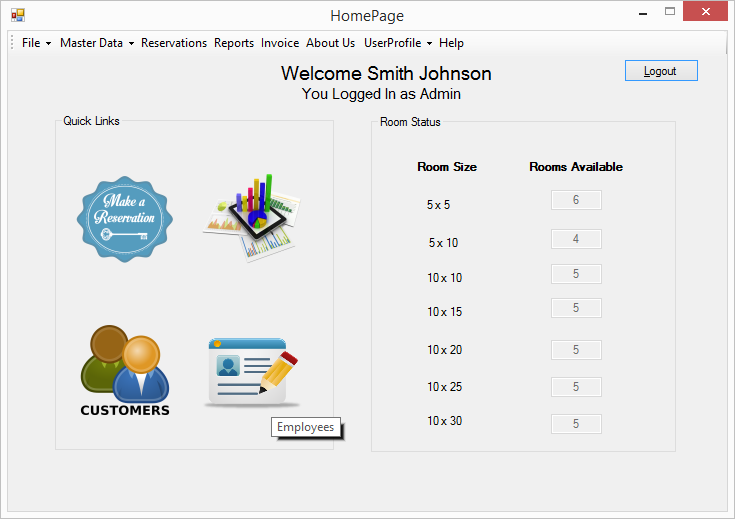
When the employee successfully login, Home Page Screen appears. The full name of the employee and the position will be displayed on the top of the screen just below menu tool strip bar. Quick links are provides with tooltip text function (Hover) and while clicking on any of them the employee would be re-directed to respective Screen.

The employee can also navigate into various screens with the help of menu bar at the top of the screen.



**Figure 16** Home Page Screen

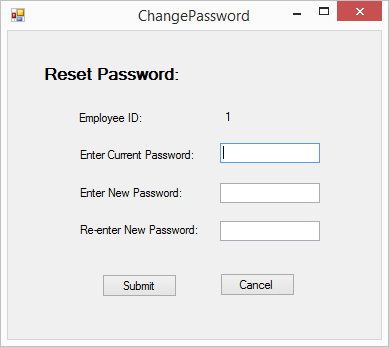
On the other side of the screen, summary of available number of rooms will be displayed automatically based on the room size. If any room is booked, automatically the number would get updated accordingly.



**Figure 17** Home Page Screen with updated available room status

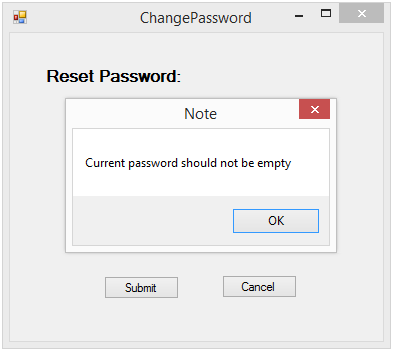
**Change Employee Login Password:**

Upon Successful login, the employees can change their password by clicking on User Profile and then Change Password options available in the Menu tool strip of the Home Page Screen. The employee ID will be automatically reflected in the First field (disabled for editing).



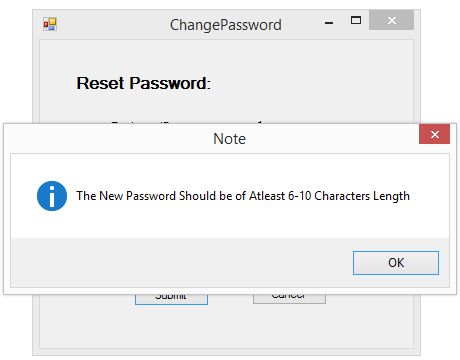
**Figure 18** Change Password Screen

When the employee does not type the current password field and moves to new password field then the following message appears on the screen.



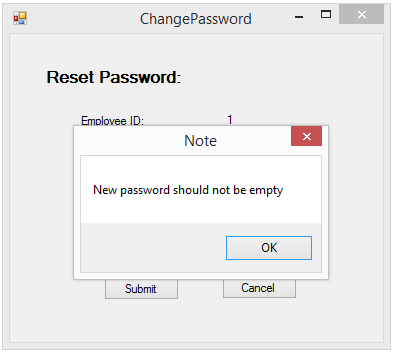
**Figure 19** Validation for leaving current password field empty

The password should be minimum of 6-10 characters length.



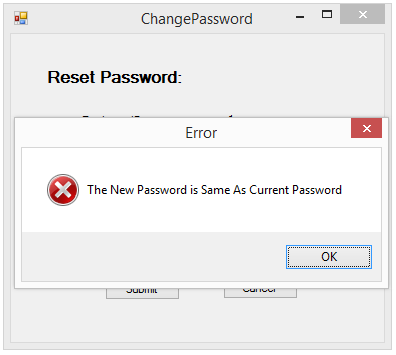
**Figure 20** Validation stating the length of the new password

If the employee leaves the new password field empty and moves to the next field then the following message appears on the screen.



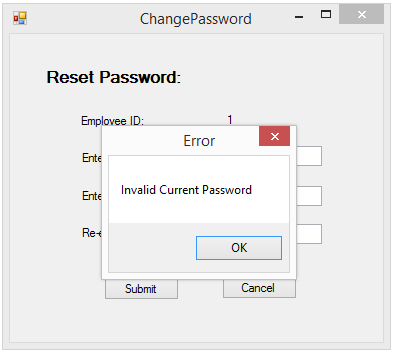
**Figure 21** Validation for leaving the new password field empty

If the employee tries to use the same old password then an error message appears on the screen as indicated below.



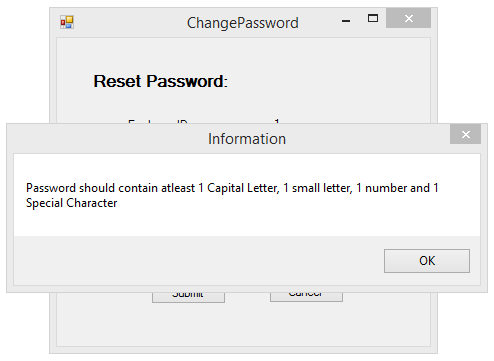
**Figure 22** Validation indicating both current password and new password are same

If the employee provides invalid old password then a message appears on the screen as indicated below.



**Figure 23** Validation indicating invalid current password

Message indicating the rules of creating new password.



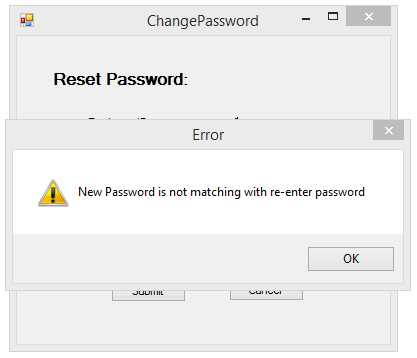
**Figure 24** Validation indicating the configuration of new password

Validation message indicating that the re-enter password should not be empty while changing password.



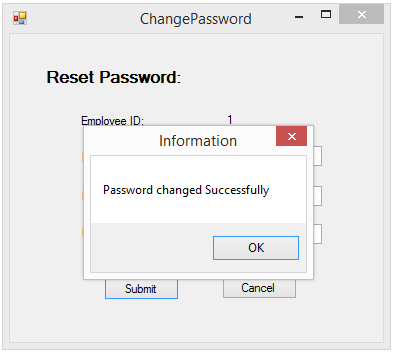
**Figure 25** Validation indicating re-enter password field left empty

Message indicating new password is not matching with re-enter password.



**Figure 26** Validation indicating mismatch of new password and re-enter password

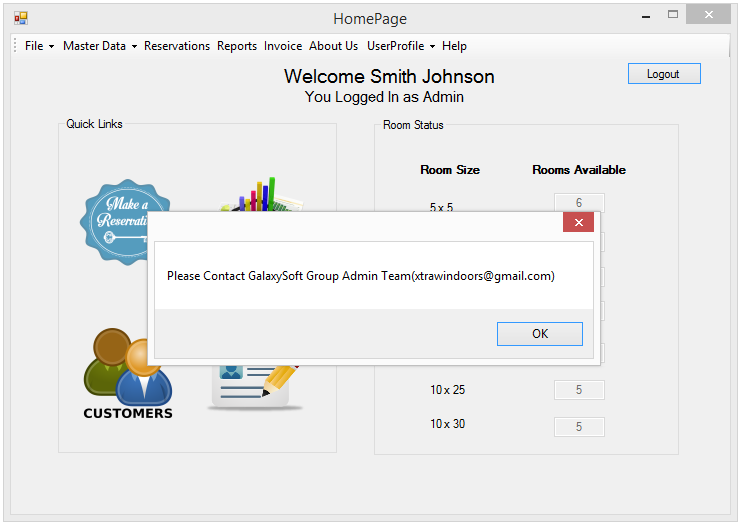
Information provided to the employee stating that the password has been changed successfully.



**Figure 27** Confirmation of successful password change

**Help Screen:**

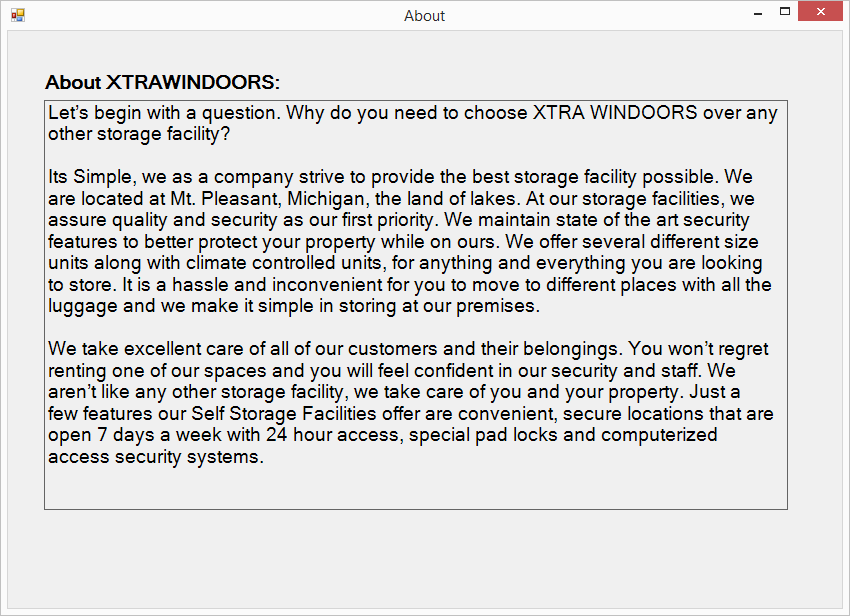
When clicked on the Help option on the menu bar, a message box pops up providing admin email address as shown below.



**Figure 28** Help Screen

**About US Screen:**

When clicked on the About Us option on the menu bar the following screen appears.



**Figure 29** About Us Screen

**Coding:**

**Splash Screen Code:**

Public Class SplashScreen

Private Sub Timer1\_Tick(ByVal sender As System.Object, e As System.EventArgs) Handles Timer1.Tick

Timer1.Start()

ProgressBar.Increment(1)

If ProgressBar.Value = 25 Then

Status.Text = "Retrieving Records..."

End If

If ProgressBar.Value = 50 Then

Status.Text = "Updating Records..."

End If

If ProgressBar.Value = 96 Then

Status.Text = "Completed!!"

End If

If ProgressBar.Value = 100 Then

Timer1.Stop()

Me.Hide()

LoginForm.Show()

End If

End Sub

Private Sub SplashScreen1\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

Status.Text = "Loading Details..."

End Sub

End Class

**Login Screen Code:**

Imports System.Data.OleDb

Imports System.Net.Mail

Public Class LoginForm

' Dim cn As New OleDb.OleDbConnection

' Dim cmd As New OleDb.OleDbCommand

' Dim str As String

Dim EmpID As String

Dim path = System.Windows.Forms.Application.StartupPath

Private Sub SignInButton\_Click(sender As System.Object, e As System.EventArgs) Handles SignInButton.Click

Dim connection As New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|\XtraWindoors.accdb")

Dim command As New OleDbCommand("SELECT [EmpID], [FirstName], [LastName], [Position] FROM [Employee] WHERE [EmpID] = ? AND [password] = ?", connection)

Dim usernameParam As New OleDbParameter("username", Me.EmployeeID.Text)

Dim passwordParam As New OleDbParameter("password", Me.UserPass.Text)

command.Parameters.Add(usernameParam)

command.Parameters.Add(passwordParam)

command.Connection.Open()

Dim reader As OleDbDataReader = command.ExecuteReader()

'If (reader.Read() = True) Then

If reader.HasRows Then

MessageBox.Show("You are Authenticated")

reader.Read()

Dim HomePageScreen As New HomePageScreen

'Dim Strin As String

'Strin = reader.GetString(0)

''reader.GetString(0)

EmpID = Me.EmployeeID.Text

HomePageScreen.SetUserData(EmpID, reader.GetString(1) + " " + reader.GetString(2), reader.GetString(3))

HomePageScreen.Show()

EmployeeID.Text = ""

UserPass.Text = ""

Me.Hide()

ElseIf String.IsNullOrEmpty(EmployeeID.Text) And String.IsNullOrEmpty(UserPass.Text) Then

MsgBox("Please enter Username and Password", MsgBoxStyle.Critical, "Error")

ElseIf String.IsNullOrEmpty(EmployeeID.Text) Then

MsgBox("Please enter Username", MsgBoxStyle.Critical, "Error")

EmployeeID.Text = ""

ElseIf String.IsNullOrEmpty(UserPass.Text) Then

MsgBox("Please enter Password", MsgBoxStyle.Critical, "Error")

UserPass.Text = ""

Else

MsgBox("Invalid Username or Password. Please Try again", MsgBoxStyle.Critical, "Error")

EmployeeID.Text = ""

UserPass.Text = ""

End If

command.Connection.Close()

End Sub

Private Sub ExitButton\_Click(sender As System.Object, e As System.EventArgs) Handles ExitButton.Click

Application.Exit()

End Sub

Private Sub ContactUsLink\_LinkClicked(sender As System.Object, e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles ContactUsLink.LinkClicked

Contact\_Us.Show()

End Sub

Private Sub ForgotPasswordLink\_LinkClicked(sender As System.Object, e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles ForgotPasswordLink.LinkClicked

Me.Hide()

ForgotPassword.Show()

End Sub

Private Sub LoginForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

End Sub

End Class

**Contact Us Form Code:**

Public Class Contact\_Us

Private Sub BookingForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

LoginForm.Show()

End Sub

End Class

**Forgot Password Code:**

Imports System.Net.Mail

Imports System.Data.OleDb

Public Class ForgotPassword

Dim str1 As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|\XtraWindoors.accdb"

Dim cn1 As New OleDb.OleDbConnection(str1)

Dim empID As String

Dim empMailID As String

Private Function GetNewPassword() As String

Try

Dim empID = Trim(TxtEmpID.Text)

Dim newPswd = String.Format("NXT{0}QO", Convert.ToInt32(empID) + 1895)

Dim cmd2 As New OleDb.OleDbCommand(String.Format("Update Employee set [Password] ='{0}' where [EmpID] = {1}", newPswd, empID), cn1)

cmd2.ExecuteNonQuery()

Return newPswd

Catch ex As Exception

MsgBox(ex.ToString)

Return String.Empty

End Try

End Function

Private Sub SendMail()

Try

Dim SmtpServer As New SmtpClient()

Dim mail As New MailMessage()

SmtpServer.Credentials = New Net.NetworkCredential("xtrawindoors@gmail.com", "GalaxySoft")

SmtpServer.Port = 587

SmtpServer.EnableSsl = True

SmtpServer.Host = "smtp.gmail.com"

mail = New MailMessage()

mail.From = New MailAddress("xtrawindoors@gmail.com")

mail.To.Add(empMailID)

mail.Subject = "Reset Password from XTRAWINDOORS"

mail.Body = String.Format("Your new Password is {0} SMTP mail from GMAIL", GetNewPassword())

SmtpServer.Send(mail)

MsgBox("New password as been sent your mail")

Me.Hide()

LoginForm.Show()

Catch ex As Exception

MsgBox(ex.ToString)

End Try

End Sub

Private Sub BtnSubmit\_Click(sender As System.Object, e As System.EventArgs) Handles BtnSubmit.Click

If String.IsNullOrEmpty(TxtEmpID.Text) And String.IsNullOrEmpty(Txtemailadd.Text) Then

MsgBox("Please enter Employee ID and Email ID", MsgBoxStyle.Critical, "Error")

ElseIf String.IsNullOrEmpty(TxtEmpID.Text) Then

MsgBox("Please enter Employee ID", MsgBoxStyle.Critical, "Error")

ElseIf String.IsNullOrEmpty(Txtemailadd.Text) Then

MsgBox("Please enter Email ID", MsgBoxStyle.Critical, "Error")

Else

empID = Trim(TxtEmpID.Text)

empMailID = Trim(Txtemailadd.Text)

Dim command As New OleDbCommand("Select \* from Employee where EmpID=" & empID & " and EmailID='" & empMailID & "'", cn1)

command.Connection.Open()

Dim reader As OleDbDataReader = command.ExecuteReader()

If reader.HasRows Then

SendMail()

Else

MsgBox("Invalid Employee ID or Email ID", MsgBoxStyle.Critical, "Error")

End If

End If

End Sub

Private Sub BtnCancel\_Click(sender As System.Object, e As System.EventArgs) Handles BtnCancel.Click

Me.Hide()

LoginForm.Show()

End Sub

Private Sub BookingForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

LoginForm.Show()

End Sub

End Class

**Home Page Screen Code:**

Imports System.Data.OleDb

Public Class HomePageScreen

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim str As String

Dim EmployeeID As Integer

Public Sub SetUserData(EmpID As Integer, myUserName As String, myUserDesig As String)

LblUserName.Text = String.Format("Welcome {0}", myUserName)

LblUserDesig.Text = String.Format("You Logged In as {0}", myUserDesig)

EmployeeID = EmpID

End Sub

Private Sub HelpLabel\_Click(sender As System.Object, e As System.EventArgs) Handles HelpLabel.Click

MessageBox.Show("Please Contact GalaxySoft Group Admin Team(xtrawindoors@gmail.com)")

End Sub

Private Sub ReportsLabel\_Click(sender As System.Object, e As System.EventArgs) Handles ReportsLabel.Click

Me.Hide()

ReportForm.Show()

End Sub

Private Sub ReportsPictureBox\_Click(sender As System.Object, e As System.EventArgs) Handles ReportsPictureBox.Click

Me.Hide()

ReportForm.Show()

End Sub

Private Sub EmployeePictureBox\_Click(sender As System.Object, e As System.EventArgs) Handles EmployeePictureBox.Click

Me.Hide()

EmployeeForm.Show()

End Sub

Private Sub EmployeeMasterLabel\_Click(sender As System.Object, e As System.EventArgs) Handles EmployeeMasterLabel.Click

Me.Hide()

EmployeeForm.Show()

End Sub

Private Sub CustomersPictureBox\_Click(sender As System.Object, e As System.EventArgs) Handles CustomersPictureBox.Click

Me.Hide()

CustomerForm.Show()

End Sub

Private Sub CustomerMasterLabel\_Click(sender As System.Object, e As System.EventArgs) Handles CustomerMasterLabel.Click

Me.Hide()

CustomerForm.Show()

End Sub

Private Sub ReservationsHover(sender As System.Object, e As System.EventArgs) Handles ReservationsPictureBox.MouseHover

Dim Tooltip As ToolTip = New ToolTip()

Tooltip.SetToolTip(ReservationsPictureBox, "Reservations")

End Sub

Private Sub ReportsHover(sender As System.Object, e As System.EventArgs) Handles ReportsPictureBox.MouseHover

Dim Tooltip As ToolTip = New ToolTip()

Tooltip.SetToolTip(ReportsPictureBox, "Reports")

End Sub

Private Sub EmployeeHover(sender As System.Object, e As System.EventArgs) Handles EmployeePictureBox.MouseHover

Dim Tooltip As ToolTip = New ToolTip()

Tooltip.SetToolTip(EmployeePictureBox, "Employees")

End Sub

Private Sub CustomersHover(sender As System.Object, e As System.EventArgs) Handles CustomersPictureBox.MouseHover

Dim Tooltip As ToolTip = New ToolTip()

Tooltip.SetToolTip(CustomersPictureBox, "Customers")

End Sub

Private Sub HomePageScreen\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

Timer.Start()

End Sub

Private Sub AboutLabel\_Click(sender As System.Object, e As System.EventArgs) Handles AboutLabel.Click

About.Show()

End Sub

Private Sub FillRomStatus()

Dim AL As New Collection

AL.Add(StatusTextBox1, "1")

AL.Add(StatusTextBox2, "2")

AL.Add(StatusTextBox3, "3")

AL.Add(StatusTextBox4, "4")

AL.Add(StatusTextBox5, "5")

AL.Add(StatusTextBox6, "6")

AL.Add(StatusTextBox7, "7")

str = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|\XtraWindoors.accdb"

cn = New OleDb.OleDbConnection(str)

cn.Open()

Dim dt As New DataTable("Storage")

Dim cmd As New OleDb.OleDbCommand("Select StorageTypeID, COUNT(StorageID) from [Storage] where StorageTypeID is not null and [StorageAvaiablity] = Yes group by StorageTypeID", cn)

Dim dr As OleDb.OleDbDataReader

dr = cmd.ExecuteReader

While dr.Read()

AL(dr(0)).Text = dr(1).ToString

End While

cn.Close()

End Sub

Private Sub Timer\_Tick(sender As System.Object, e As System.EventArgs) Handles Timer.Tick

FillRomStatus()

End Sub

Public Sub ChangePasswordToolStripMenuItem\_Click(sender As System.Object, e As System.EventArgs) Handles ChangePasswordToolStripMenuItem.Click

ChangePassword.GetEmpID(EmployeeID)

ChangePassword.Show()

End Sub

Private Sub HomePageScreen\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

LoginForm.Show()

End Sub

Private Sub ExitLabel\_Click\_1(sender As System.Object, e As System.EventArgs) Handles ExitLabel.Click

Me.Hide()

LoginForm.Show()

End Sub

Private Sub ReservationLabel\_Click(sender As System.Object, e As System.EventArgs) Handles ReservationLabel.Click

Me.Hide()

BookingForm.Show()

End Sub

Private Sub ReservationsPictureBox\_Click(sender As System.Object, e As System.EventArgs) Handles ReservationsPictureBox.Click

Me.Hide()

BookingForm.Show()

End Sub

Private Sub HomeButton\_Click(sender As System.Object, e As System.EventArgs) Handles HomeButton.Click

Me.Hide()

LoginForm.Show()

End Sub

Private Sub ToolStripLabel1\_Click(sender As System.Object, e As System.EventArgs) Handles InvoiceToolStripLabel.Click

Me.Hide()

SaleInvoiceForm.Show()

End Sub

End Class

**Change Password Screen Code:**

Imports System.Data.OleDb

Imports System.Text.RegularExpressions

Public Class ChangePassword

Public EmployeeID As Integer

Public Password As String

Dim str1 As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|\XtraWindoors.accdb"

Dim cn1 As New OleDb.OleDbConnection(str1)

Public Sub GetEmpID(EmpID As Integer)

EmployeeID = EmpID

End Sub

Private Sub ChangePassword\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

'ChangePassword\_user()

EmpIDlabel.Text = EmployeeID

End Sub

Public Sub ChangePassword\_user()

cn1.Open()

If TxtPass.TextLength < 6 Then

MsgBox("The New Password Should be of Atleast 6-10 Characters Length", MsgBoxStyle.Information, "Note")

TxtPass.Text = ""

TxtRePass.Text = ""

TxtCPass.Text = ""

ElseIf TxtPass.Text = TxtRePass.Text Then

Dim dt As New DataTable("Storage")

Dim cmd As New OleDb.OleDbCommand("Select \* from Employee where EmpID=" & Trim(EmployeeID) & " and Password='" & Trim(TxtCPass.Text) & "'", cn1)

Password = TxtPass.Text

Dim dr As OleDb.OleDbDataReader

dr = cmd.ExecuteReader

Dim intstatus As Integer

If dr.HasRows() Then

Dim cmd2 As New OleDb.OleDbCommand(String.Format("Update Employee set [Password] ='{0}' where [EmpID] = {1}", Trim(TxtPass.Text), Trim(EmployeeID)), cn1)

intstatus = cmd2.ExecuteNonQuery()

MessageBox.Show("Password changed Successfully", "Information")

Me.Hide()

End If

End If

cn1.Close()

End Sub

Private Sub BtnSubmit\_Click(sender As System.Object, e As System.EventArgs) Handles BtnSubmit.Click

ChangePassword\_user()

End Sub

Private Sub ChangePassword\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

HomePageScreen.Show()

End Sub

Private Sub TxtCPass\_Validating(sender As System.Object, e As System.ComponentModel.CancelEventArgs) Handles TxtCPass.Validating

If TxtCPass.Text.Trim().Length = 0 Then

MessageBox.Show("Current password should not be empty", "Note")

End If

If TxtCPass.Text <> Password Then

MessageBox.Show("Invalid Current Password", "Error")

End If

End Sub

Private Sub TxtPass\_Validating(sender As System.Object, e As System.ComponentModel.CancelEventArgs) Handles TxtPass.Validating

Dim MatchNumberPattern As String = "(?=^.{6,10}$)(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[!@#$%^&amp;\*()\_+}{&quot;:;'?/&gt;.&lt;,])(?!.\*\s).\*$"

If TxtPass.Text.Trim <> "" Then

If Not Regex.IsMatch(TxtPass.Text, MatchNumberPattern) Then

MessageBox.Show("Password should contain atleast 1 Capital Letter, 1 small letter, 1 number and 1 Special Character", "Information")

TxtPass.Text = ""

End If

End If

If TxtPass.Text.Trim().Length = 0 Then

MessageBox.Show("New password should not be empty", "Note")

End If

If TxtPass.Text = TxtCPass.Text Then

MsgBox("The New Password is Same As Current Password", MsgBoxStyle.Critical, "Error")

TxtPass.Text = ""

End If

End Sub

Private Sub TxtRePass\_Validating(sender As System.Object, e As System.ComponentModel.CancelEventArgs) Handles TxtRePass.Validating

If TxtRePass.Text.Trim().Length = 0 Then

MessageBox.Show("Re-enter password should not be empty", "Note")

End If

If TxtPass.Text <> TxtRePass.Text Then

MsgBox("New Password is not matching with re-enter password", MsgBoxStyle.Exclamation, "Error")

TxtRePass.Text = ""

End If

End Sub

Private Sub BtnCancel\_Click(sender As System.Object, e As System.EventArgs) Handles BtnCancel.Click

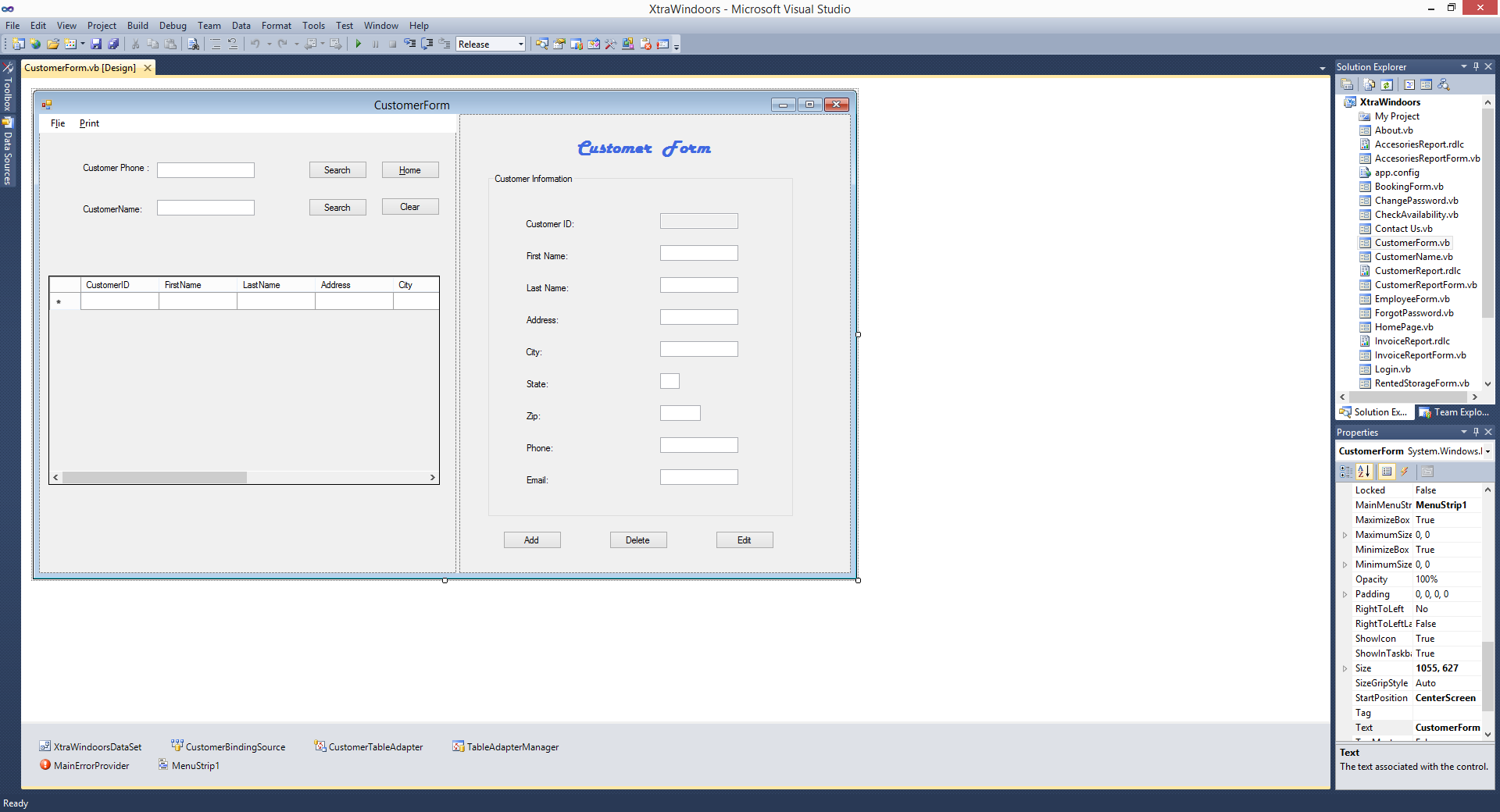
Me.Hide()

End Sub

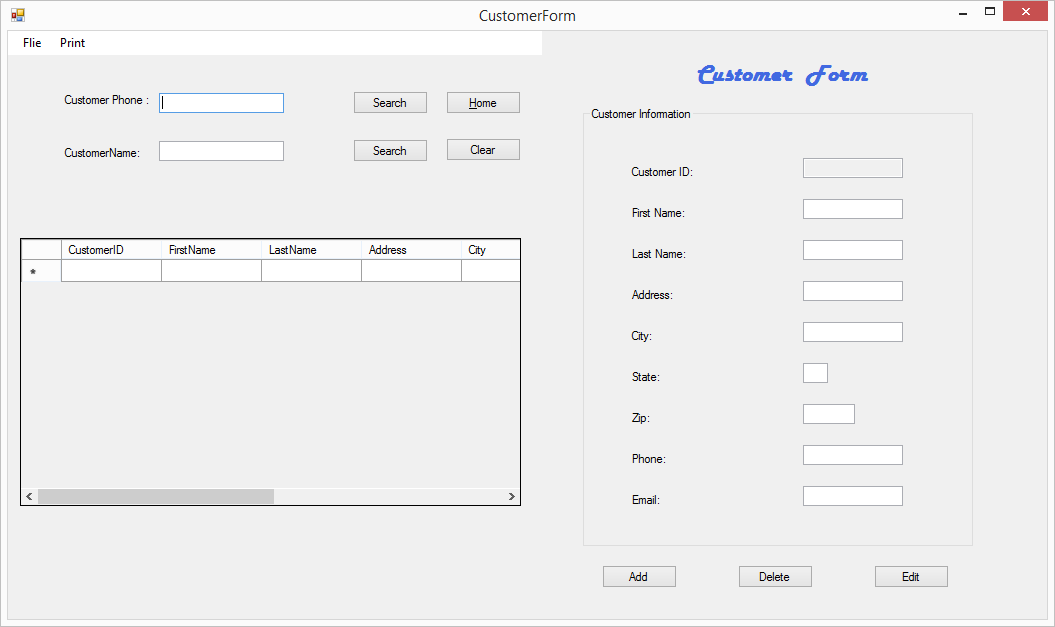
End Class

**Customer form**

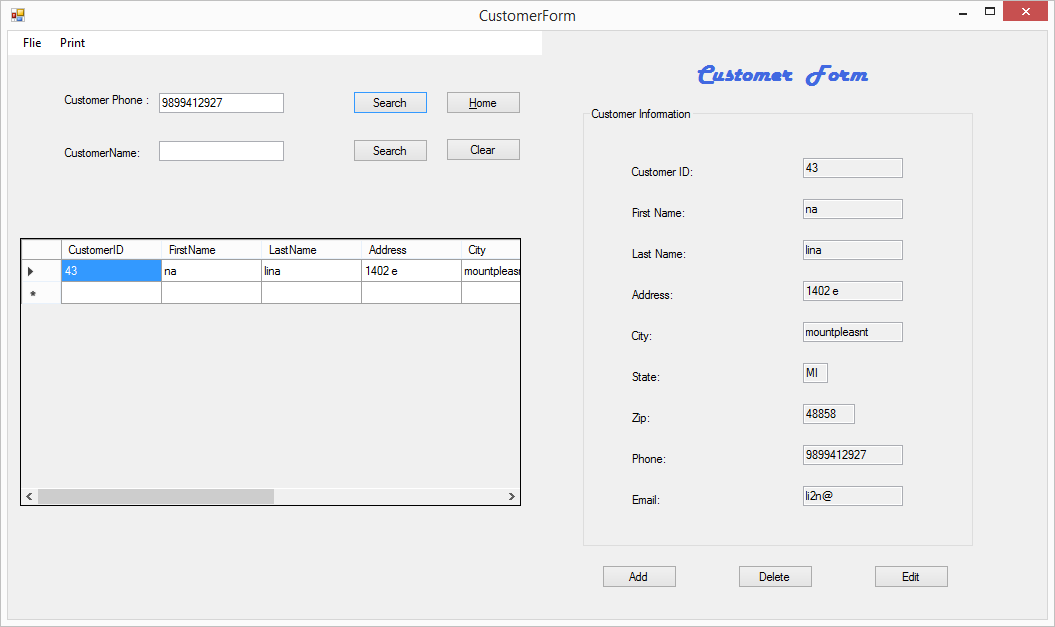
**Figure 1** Design view:



**Figure 2** Excusive view:



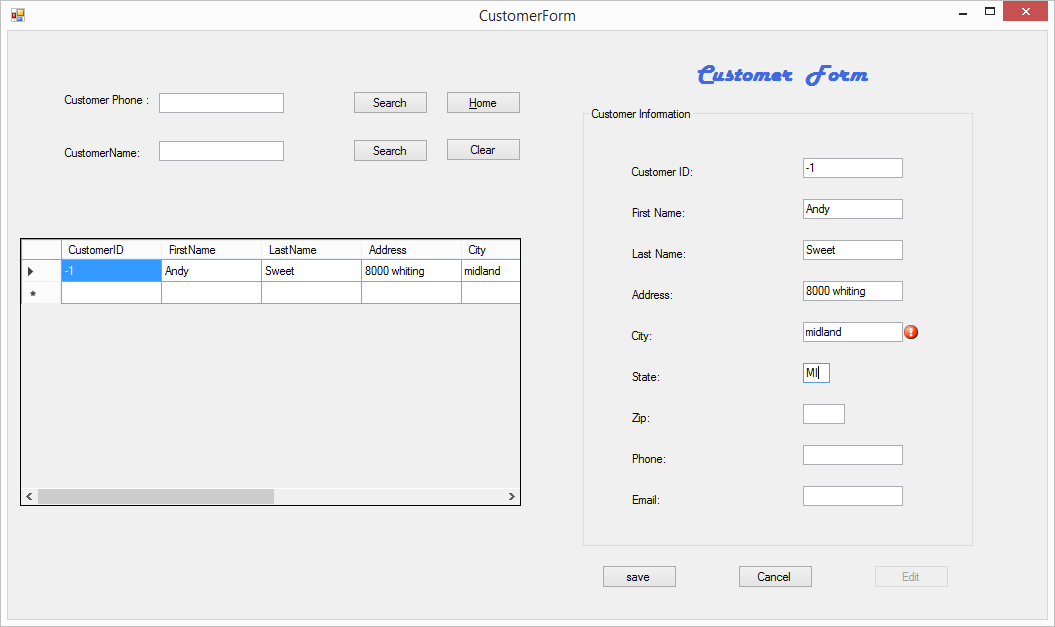
**Figure 2. 1** Search by “Customer phone ” :



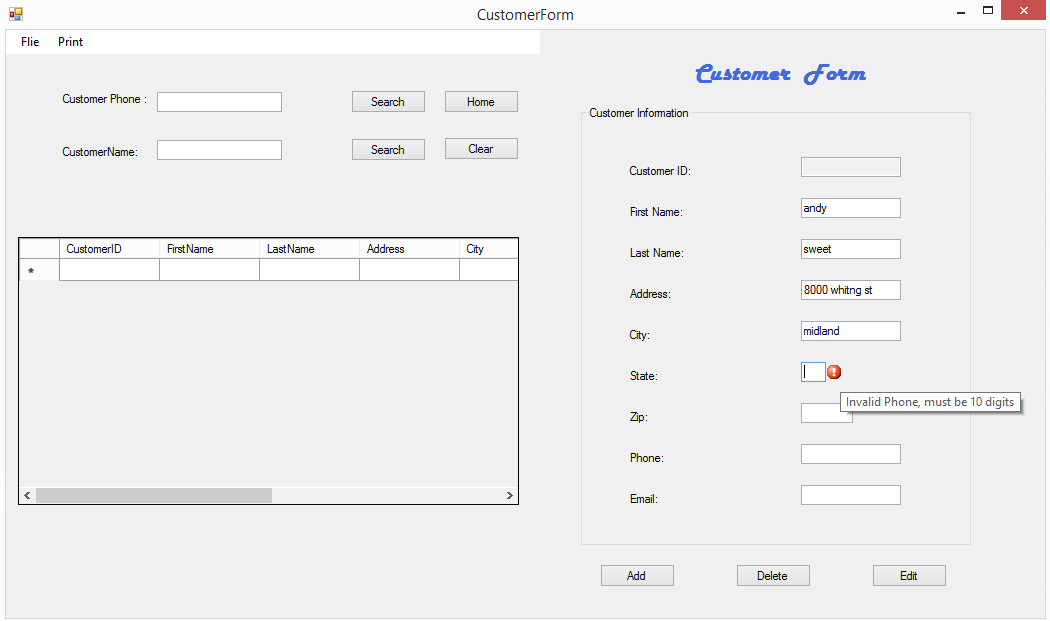
**Figure 2. 1** Search by “customer name”:



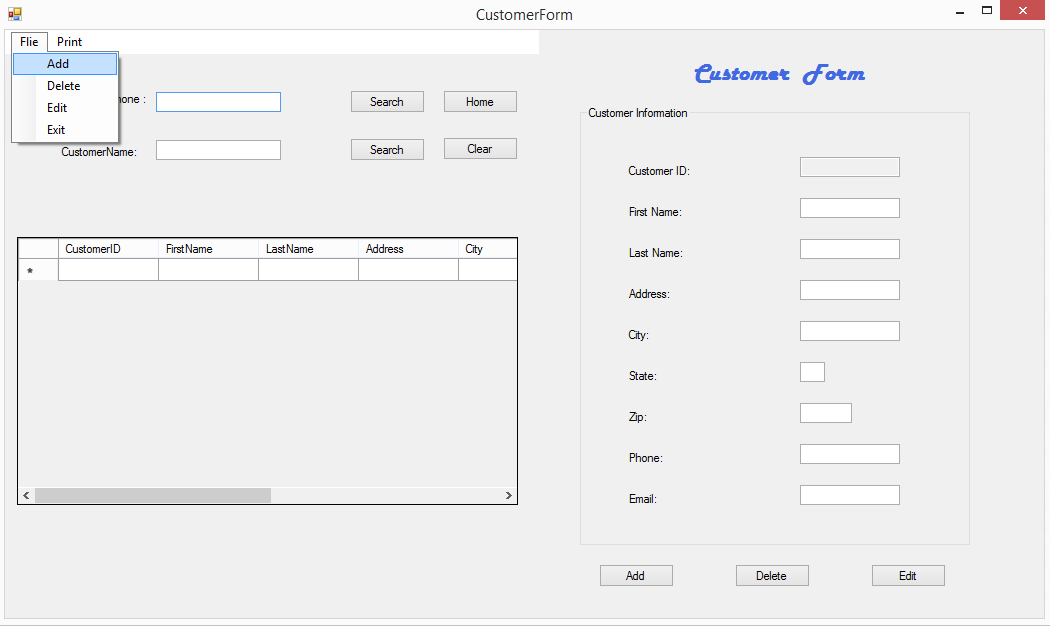
**Figure 2. 3** Add button clicked, and the validating information shows on city:



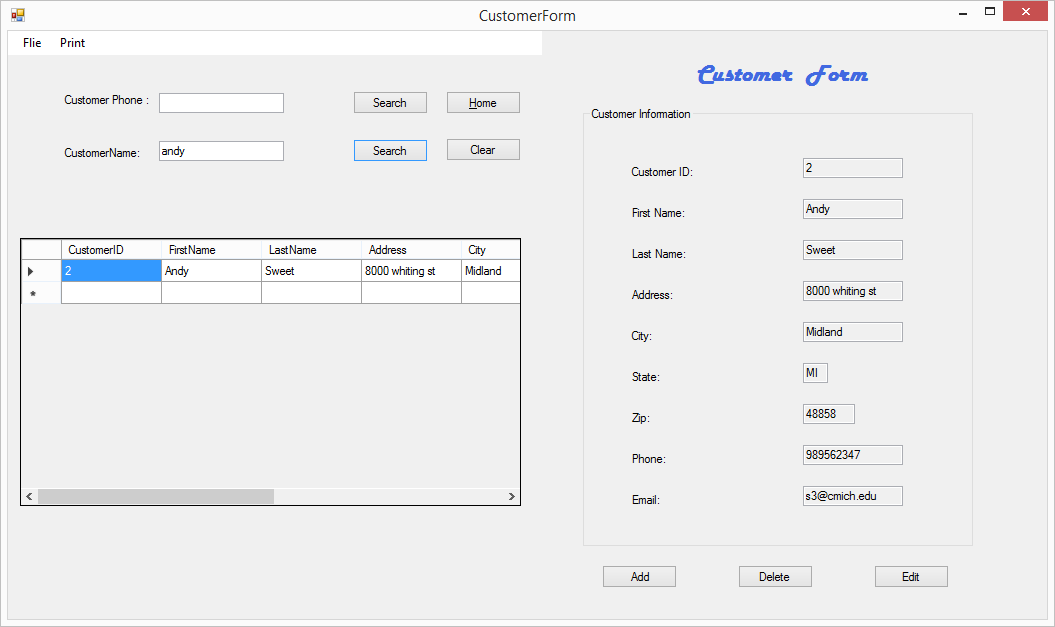
**Figure 2. 3** Add button clicked, and the validating information shows on state:



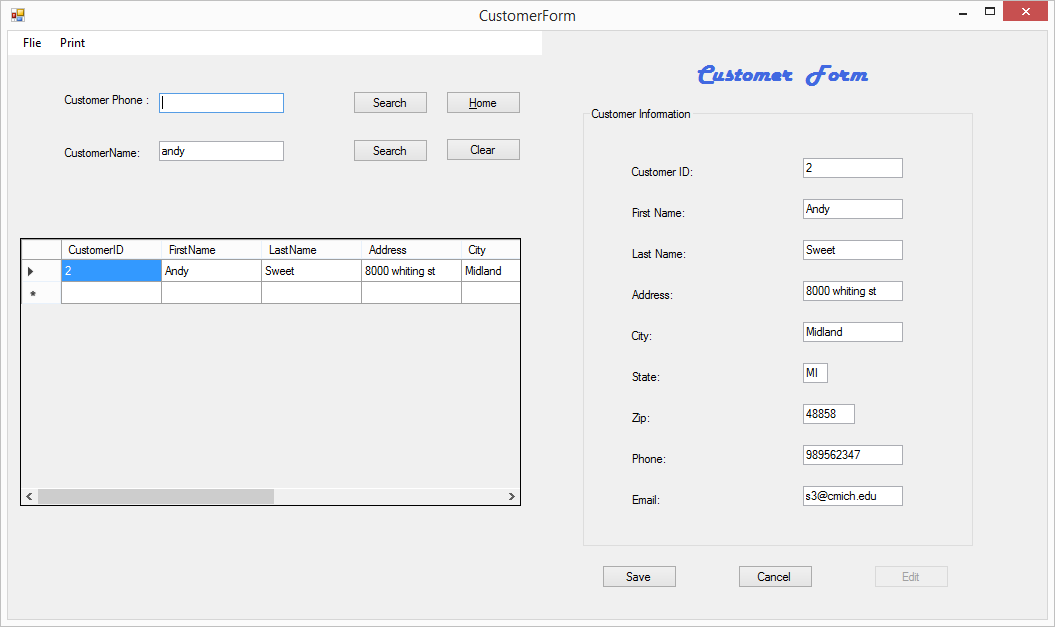
**Figure 2. 4** Add from the menu button:



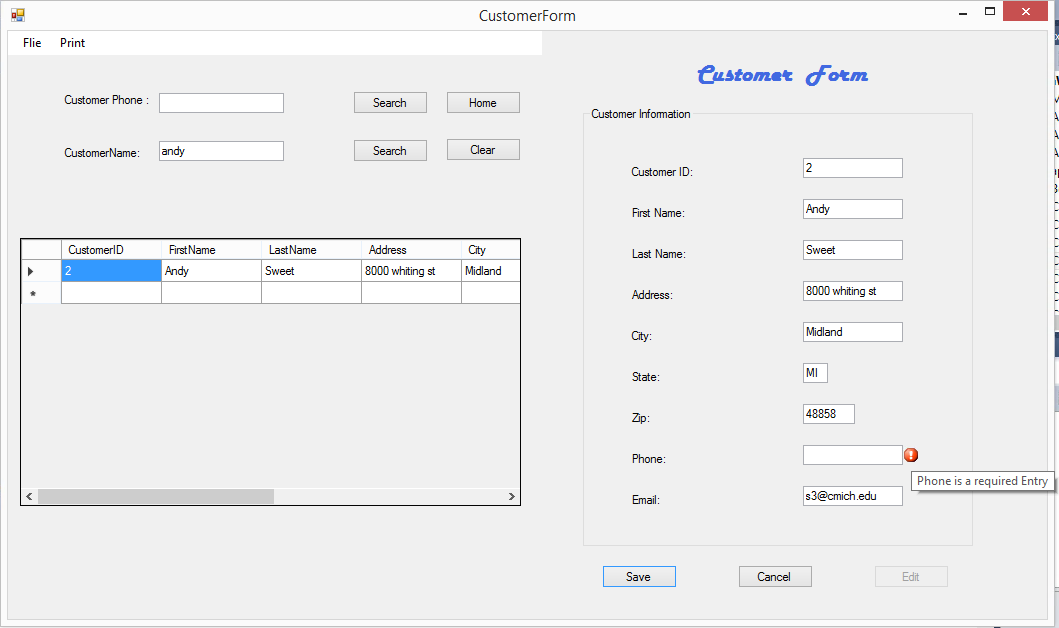
**Figure 2. 5** Check the search by name:



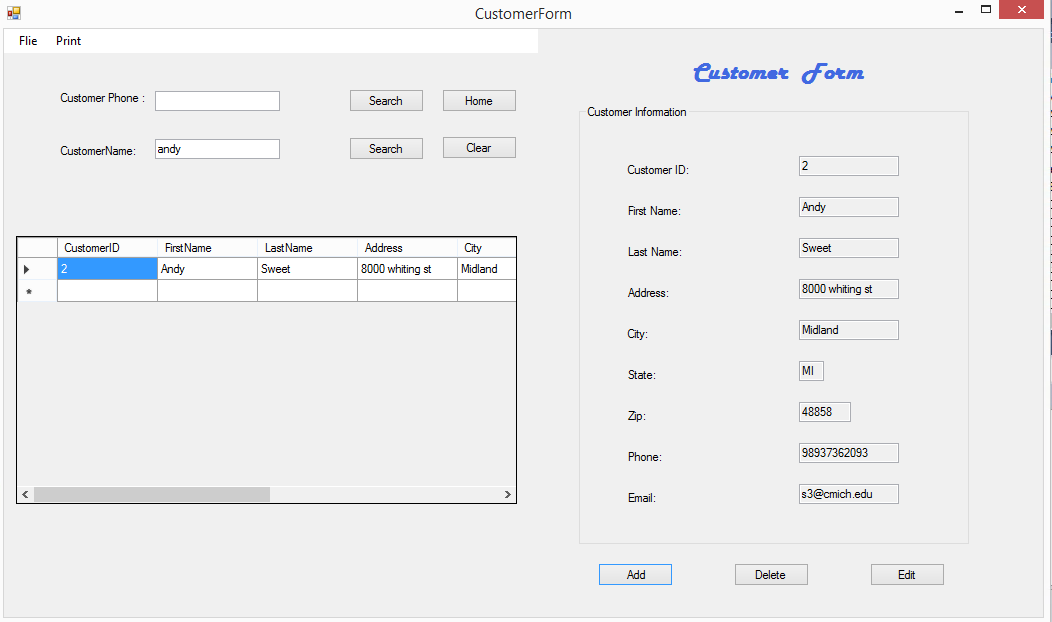
**Figure 3. 1** Edit Button clicked:



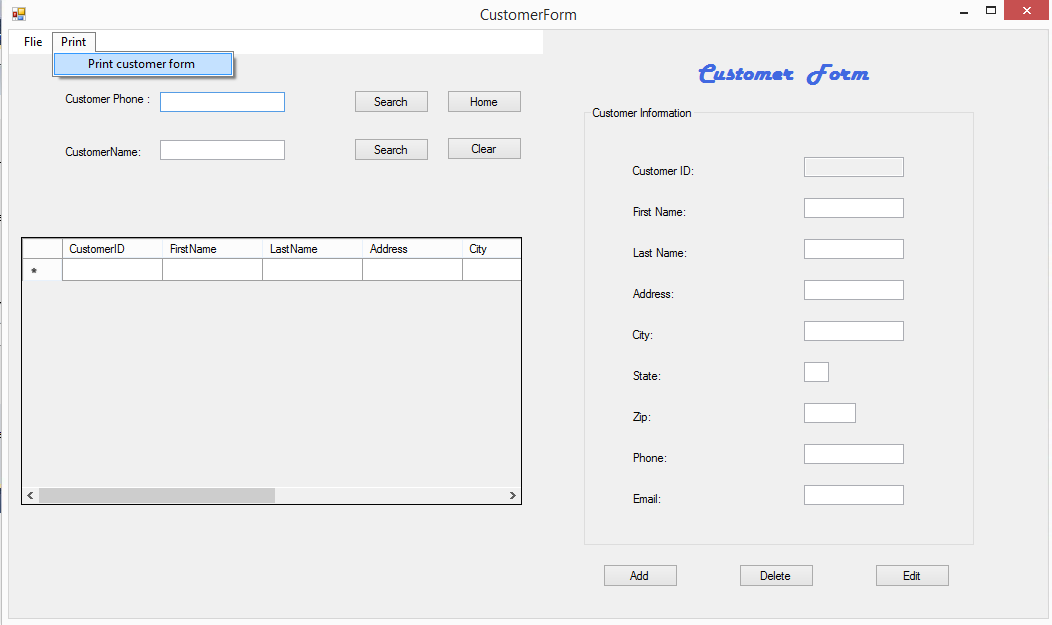
**Figure 3.2** Changed phone number:



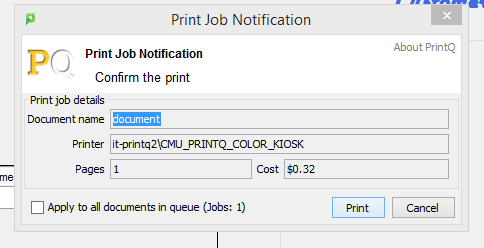
**Figure 3.3** After save button clicked:



**Figure 4.0** Print button from the menu button:



**Figure 4.1**



**Customer form Code**

'Program Name: XtraWindoors Rental System

'Programmer: Group 7

'Date: 12/10/2015

'Class: Customer Form

Public Class CustomerForm

' Module-level declarations.

Private AxtrawindoorsDateSet As New XtraWindoorsDataSet

Private ACustomertableAdapter As New XtraWindoorsDataSetTableAdapters.CustomerTableAdapter

Private WithEvents ACustomerBindingSource As BindingSource

Private AddingBoolean As Boolean = False

Private EditingBoolean As Boolean = False

Private ClosingBoolean As Boolean = False

Private Sub CustomerBindingNavigatorSaveItem\_Click(sender As System.Object, e As System.EventArgs)

Me.Validate()

Me.CustomerBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.XtraWindoorsDataSet)

End Sub

Private Sub CustomerForm\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

''TODO: This line of code loads data into the 'AAAStorageDataSet.Customer' table. You can move, or remove it, as needed.

'Me.CustomerTableAdapter.Fill(Me.AAAStorageDataSet.Customer)

MainErrorProvider.Clear()

End Sub

Private Sub SetControlsReadOnly(ByVal ValueBoolean As Boolean)

CustomerIDTextBox.ReadOnly = ValueBoolean

PhoneTextBox.ReadOnly = ValueBoolean

FirstNameTextBox.ReadOnly = ValueBoolean

LastNameTextBox.ReadOnly = ValueBoolean

AddressTextBox.ReadOnly = ValueBoolean

CityTextBox.ReadOnly = ValueBoolean

StateTextBox.ReadOnly = ValueBoolean

ZipTextBox.ReadOnly = ValueBoolean

EmailTextBox.ReadOnly = ValueBoolean

End Sub

Private Sub SetButtonForEdit()

AddSaveButton.Text = "Save"

DeleteCancelButton.Text = "Cancel"

EditButton.Enabled = False

End Sub

Private Sub ResetButtonafterEdit()

AddSaveButton.Text = " Add"

DeleteCancelButton.Text = "Delete"

EditButton.Enabled = True

End Sub

Private Sub ExitProc()

Dim ADialogResult As DialogResult

ADialogResult = MessageBox.Show("Do you Want to exit?", "Exit", MessageBoxButtons.OKCancel, MessageBoxIcon.Question)

If ADialogResult = Windows.Forms.DialogResult.OK Then

Me.Close()

End If

End Sub

Private Sub ExitToolStripMenuItem\_Click(sender As System.Object, e As System.EventArgs)

Call ExitProc()

End Sub

Private Sub ClearButton\_Click(sender As System.Object, e As System.EventArgs) Handles ClearButton.Click

Try

CustomerBindingSource.Filter = ""

CustomerDataGridView.DataSource = Nothing

CustomerDataGridView.Refresh()

Dim ctl As Control

For Each ctl In GroupBox1.Controls

If TypeOf ctl Is TextBox Then

ctl.Text = ""

End If

Next

Dim ctl1 As Control

For Each ctl1 In SplitContainer1.Panel1.Controls

If TypeOf ctl1 Is TextBox Then

ctl1.Text = ""

End If

Next

'Me.CustomerTableAdapter.Fill(Me.AAAStorageDataSet.Customer)

Catch ex As Exception

System.Windows.Forms.MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub CustomerDataGridView\_CellContentClick(sender As System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs) Handles CustomerDataGridView.CellContentClick

'TODO: This line of code loads data into the 'AAAStorageDataSet.Customer' table. You can move, or remove it, as needed.

Me.CustomerTableAdapter.Fill(Me.XtraWindoorsDataSet.Customer)

'retrieve the Orders information for the grid.

Dim CustomerIDString As String

Static GridInitializedBoolean As Boolean = False

'retrieve the ID of the selected store.

CustomerIDString = CustomerDataGridView.CurrentCell.Value.ToString

'initialize the grid's binding.

If Not GridInitializedBoolean Then

'bind and format the grid

'OrdersDataGridView.DataSource = OrderBindingSource

'SetUpGridCoumns2()

'GridInitializedBoolean = True

'OrdersDataGridView.AllowUserToAddRows = False

'OrdersDataGridView.AllowUserToDeleteRows = False

'OrdersDataGridView.ReadOnly = True

End If

SetControlsReadOnly(False)

'filter the grid's data.

'OrderBindingSource.Filter = "CustomerID = '" & CustomerIDString & "'"

End Sub

Dim WithEvents mPrintDocument As New Printing.PrintDocument

Dim mPrintBitMap As Bitmap

Private Sub m\_PrintDocument\_PrintPage(ByVal sender As Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles mPrintDocument.PrintPage

' Draw the image centered.

Dim lWidth As Integer = e.MarginBounds.X + (e.MarginBounds.Width - mPrintBitMap.Width) \ 2

Dim lHeight As Integer = e.MarginBounds.Y + (e.MarginBounds.Height - mPrintBitMap.Height) \ 2

e.Graphics.DrawImage(mPrintBitMap, lWidth, lHeight)

' There's only one page.

e.HasMorePages = False

End Sub

Private Sub PrintToolStripMenuItem\_Click(sender As System.Object, e As System.EventArgs)

' Copy the form's image into a bitmap.

mPrintBitMap = New Bitmap(Me.Width, Me.Width)

Dim lRect As System.Drawing.Rectangle

lRect.Width = Me.Width

lRect.Height = Me.Width

Me.DrawToBitmap(mPrintBitMap, lRect)

' Make a PrintDocument and print.

mPrintDocument = New Printing.PrintDocument

mPrintDocument.Print()

End Sub

Private Sub EditToolStripMenuItem\_Click(sender As System.Object, e As System.EventArgs)

'Allow editing the current record.

EditingBoolean = True

SetControlsReadOnly(False)

SetButtonForEdit()

End Sub

Private Sub DeleteToolStripMenuItem\_Click(sender As System.Object, e As System.EventArgs)

'Delete the current record after coonfirming or cancel and addd or Edit

Dim DeleteDialogResult As DialogResult

Try

If DeleteCancelButton.Text = "Delete" Then

'delete records

DeleteDialogResult = MessageBox.Show("Delete this record?", "Confirm Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If DeleteDialogResult = Windows.Forms.DialogResult.Yes Then

CustomerBindingSource.RemoveCurrent()

CustomerTableAdapter.Update(XtraWindoorsDataSet.Customer)

End If

Else

'Cancel button is clicked

CustomerBindingSource.CancelEdit()

AddingBoolean = False

EditingBoolean = False

SetControlsReadOnly(True)

ResetButtonafterEdit()

End If

Catch ex As Exception

MessageBox.Show("unable to complete the delete/cancel operation:" & ex.Message, "Delete/Cancel", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Private Sub AddToolStripMenuItem\_Click(sender As System.Object, e As System.EventArgs)

If AddSaveButton.Text = "Add" Then

'Add Button is clicked

Dim ctl As Control

For Each ctl In GroupBox1.Controls

If TypeOf ctl Is TextBox Then

ctl.Text = ""

End If

Next

With CustomerBindingSource

.EndEdit()

.AddNew()

End With

AddingBoolean = True

SetControlsReadOnly(False)

SetButtonForEdit()

AddSaveButton.Text = "save"

MainErrorProvider.Clear()

Else

'save button is Clicked

Try

If CustomerIDTextBox.Text = String.Empty Then

MainErrorProvider.SetError(CustomerIDTextBox, "CustomerID is a required Entry")

ElseIf LastNameTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(LastNameTextBox, "Last Name is a required Entry")

ElseIf FirstNameTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(FirstNameTextBox, " First Name is a required Entry.")

ElseIf AddressTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(AddressTextBox, "Address is a required Entry")

ElseIf CityTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(CityTextBox, "City is a required Entry")

ElseIf StateTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(StateTextBox, "State is a required Entry")

ElseIf ZipTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(ZipTextBox, "zip is a required Entry")

ElseIf PhoneTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(PhoneTextBox, "Phone is a required Entry")

ElseIf EmailTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(EmailTextBox, "Email is a required Entry")

PhoneTextBox.Focus()

Else

CustomerBindingSource.EndEdit()

'Update data to database

CustomerTableAdapter.Update(XtraWindoorsDataSet.Customer)

AddingBoolean = False

EditingBoolean = False

SetControlsReadOnly(True)

ResetButtonafterEdit()

End If

Catch ex As Exception

MessageBox.Show("DataBase Operation Error :" & ex.Message, "Adding Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End If

End Sub

Private Sub Search2Button\_Click(sender As System.Object, e As System.EventArgs) Handles Search2Button.Click

Try

CustomerDataGridView.DataSource = CustomerBindingSource

Me.CustomerTableAdapter.FillByFirstName(Me.XtraWindoorsDataSet.Customer, CustomerNameTextBox1.Text)

SetControlsReadOnly(True)

Catch ex As Exception

MessageBox.Show(" There is no record exist !!")

End Try

End Sub

Private Sub Search1Button\_Click(sender As System.Object, e As System.EventArgs) Handles Search1Button.Click

Try

CustomerDataGridView.DataSource = CustomerBindingSource

Me.CustomerTableAdapter.FillByphone(Me.XtraWindoorsDataSet.Customer, CustomerPhoneTextBox1.Text)

SetControlsReadOnly(True)

Catch ex As System.Exception

MessageBox.Show(" There is no record exist !!")

End Try

End Sub

Private Sub DeleteCancelButton\_Click\_1(sender As System.Object, e As System.EventArgs) Handles DeleteCancelButton.Click

'Delete the current record after coonfirming or cancel and addd or Edit

Dim DeleteDialogResult As DialogResult

Try

If DeleteCancelButton.Text = "Delete" Then

'delete records

DeleteDialogResult = MessageBox.Show("Delete this record?", "Confirm Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If DeleteDialogResult = Windows.Forms.DialogResult.Yes Then

CustomerBindingSource.RemoveCurrent()

CustomerTableAdapter.Update(XtraWindoorsDataSet.Customer)

End If

Else

'Cancel button is clicked

CustomerBindingSource.CancelEdit()

AddingBoolean = False

EditingBoolean = False

SetControlsReadOnly(True)

ResetButtonafterEdit()

End If

Catch ex As Exception

MessageBox.Show("unable to complete the delete/cancel operation:" & ex.Message, "Delete/Cancel", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Private Sub EditButton\_Click\_1(sender As System.Object, e As System.EventArgs) Handles EditButton.Click

'Allow editing the current record.

EditingBoolean = True

SetControlsReadOnly(False)

SetButtonForEdit()

End Sub

Private Sub CustomerIDTextBox\_TextChanged(sender As System.Object, e As System.EventArgs) Handles CustomerIDTextBox.TextChanged

MainErrorProvider.Clear()

End Sub

Private Sub AddSaveButton\_Click(sender As System.Object, e As System.EventArgs) Handles AddSaveButton.Click

If AddSaveButton.Text = "Add" Then

'Add Button is clicked

Dim ctl As Control

For Each ctl In GroupBox1.Controls

If TypeOf ctl Is TextBox Then

ctl.Text = ""

End If

Next

With CustomerBindingSource

.EndEdit()

.AddNew()

End With

AddingBoolean = True

SetControlsReadOnly(False)

SetButtonForEdit()

AddSaveButton.Text = "save"

MainErrorProvider.Clear()

Else

'save button is Clicked

Try

If CustomerIDTextBox.Text = String.Empty Then

MainErrorProvider.SetError(CustomerIDTextBox, "CustomerID is a required Entry")

ElseIf LastNameTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(LastNameTextBox, "Last Name is a required Entry")

ElseIf FirstNameTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(FirstNameTextBox, " First Name is a required Entry.")

ElseIf AddressTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(AddressTextBox, "Address is a required Entry")

ElseIf CityTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(CityTextBox, "City is a required Entry")

ElseIf StateTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(StateTextBox, "State is a required Entry")

ElseIf ZipTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(ZipTextBox, "zip is a required Entry")

ElseIf PhoneTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(PhoneTextBox, "Phone is a required Entry")

ElseIf EmailTextBox.Text = String.Empty Then

MainErrorProvider.Clear()

MainErrorProvider.SetError(EmailTextBox, "Email is a required Entry")

PhoneTextBox.Focus()

Else

CustomerBindingSource.EndEdit()

'Update data to database

CustomerTableAdapter.Update(XtraWindoorsDataSet.Customer)

AddingBoolean = False

EditingBoolean = False

SetControlsReadOnly(True)

ResetButtonafterEdit()

End If

Catch ex As Exception

MessageBox.Show("DataBase Operation Error :" & ex.Message, "Adding Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End If

End Sub

Private Sub HomeButton\_Click(sender As System.Object, e As System.EventArgs) Handles HomeButton.Click

Me.Hide()

HomePageScreen.Show()

End Sub

Private Sub BookingForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

HomePageScreen.Show()

End Sub

Private Sub cityTextBox\_Validating(sender As Object, e As System.ComponentModel.CancelEventArgs) Handles CityTextBox.Validating

Dim citycode As String

citycode = CityTextBox.Text

Dim lengthValid As Boolean = citycode.Length.Equals(5)

If Not lengthValid Then

MainErrorProvider.SetError(CityTextBox, "city is required Entry")

End If

End Sub

Private Sub ZipTextBox\_Validating(sender As Object, e As System.ComponentModel.CancelEventArgs) Handles ZipTextBox.Validating

Dim ZipCode As String

ZipCode = ZipTextBox.Text

Dim lengthValid As Boolean = ZipCode.Length.Equals(5)

If Not lengthValid Then

MainErrorProvider.SetError(ZipTextBox, "Invalid ZIP code, must be 5 digits")

End If

End Sub

Private Sub StateTextBox\_Validating(sender As Object, e As System.ComponentModel.CancelEventArgs) Handles StateTextBox.Validating

Dim StateInput As String

StateInput = StateTextBox.Text

Dim lengthValid As Boolean = StateInput.Length.Equals(2)

If Not lengthValid Then

MainErrorProvider.SetError(StateTextBox, "Invalid State, must be 2 Character")

End If

End Sub

Private Sub PhoneTextBox\_Validating(sender As Object, e As System.ComponentModel.CancelEventArgs) Handles PhoneTextBox.Validating

Dim phoCode As String

phoCode = PhoneTextBox.Text

Dim lengthValid As Boolean = phoCode.Length.Equals(5)

If Not lengthValid Then

MainErrorProvider.SetError(PhoneTextBox, "Invalid Phone, must be 10 digits")

End If

End Sub

Private Sub EmailTextBox\_Validating(sender As Object, e As System.ComponentModel.CancelEventArgs) Handles EmailTextBox.Validating

Dim Email As String

Email = EmailTextBox.Text

Dim lengthValid As Boolean = Email.Length.Equals(5)

If Not lengthValid Then

MainErrorProvider.SetError(EmailTextBox, "Invalid email, must be @")

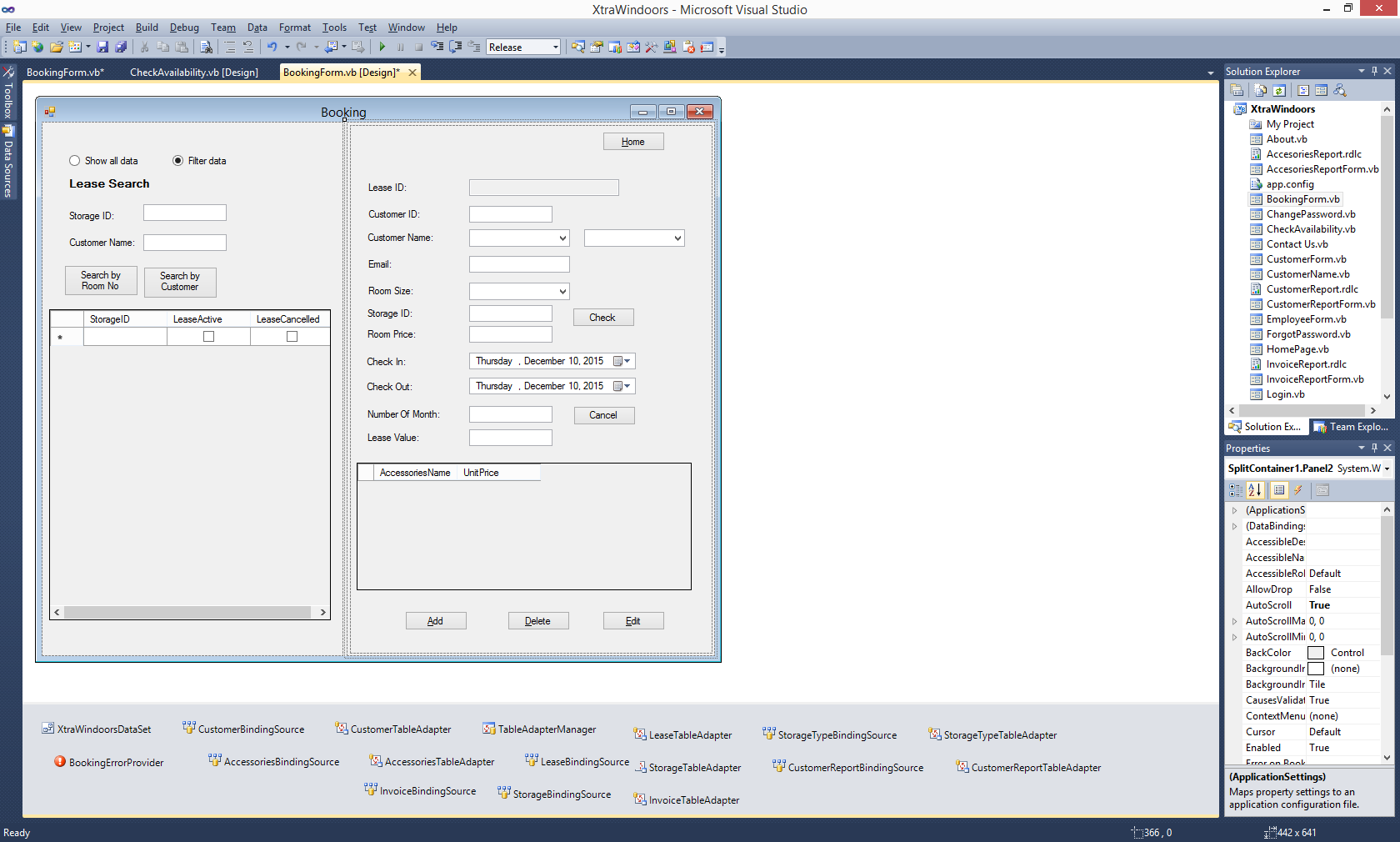
End If

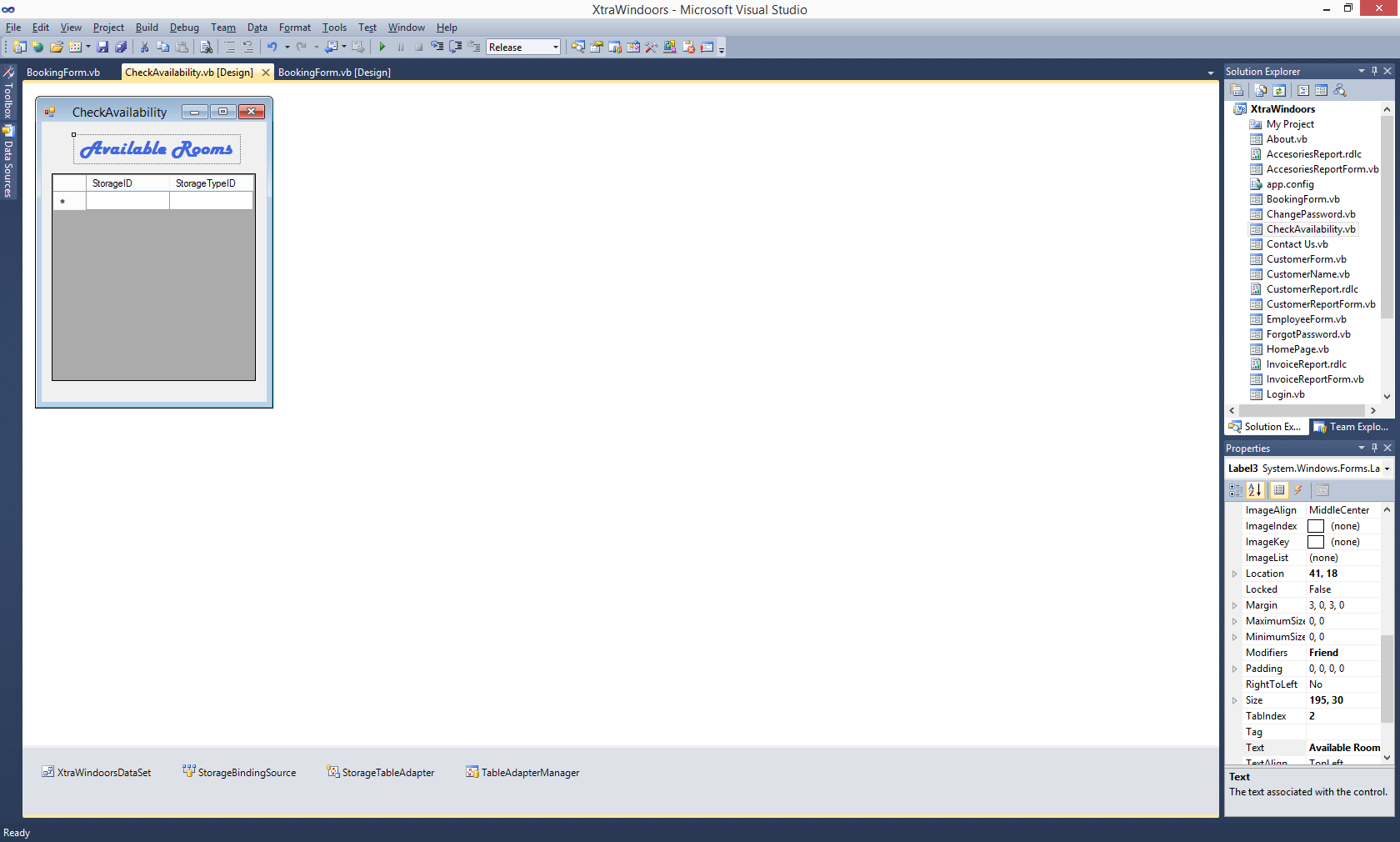
End Sub

End Class

**Booking & Room Availability Check Form:**

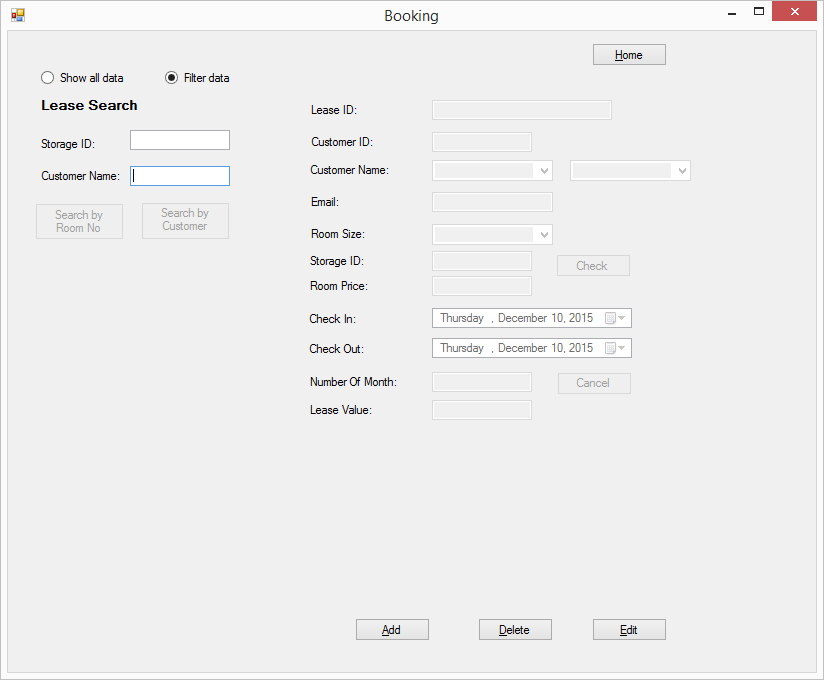
**Figure 1** Booking Form Design view

****

**Figure 2** Check Availability Form Design view****

**Booking Form:**

Once we navigate from the home screen to the Reservations, the Booking Page loads as shown below.



*Fig 1: Booking Form*

**Search for existing lease:**

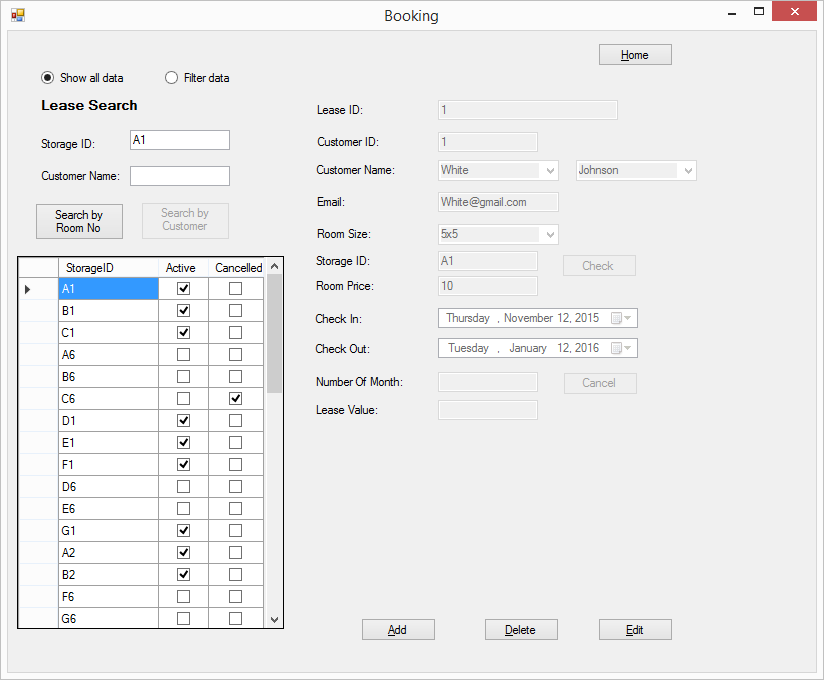
The employee can choose to:

1. **Select from all the list of existing leases**
2. **Search based on criteria**

The employee can search for an existing lease based on two criteria:

1. **Storage ID Number** (or) by
2. **Customer First Name**.

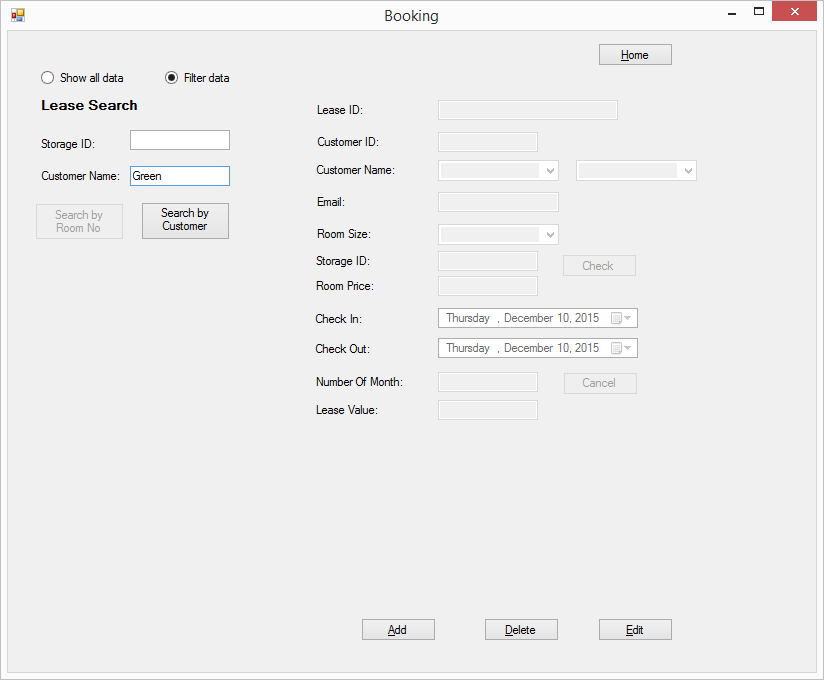
**Select from entire list:**



*Fig 2: Show all existing leases*

**Search based on criteria:**

Based on the field in which the employee types the values the respective *Search button*gets enabled.

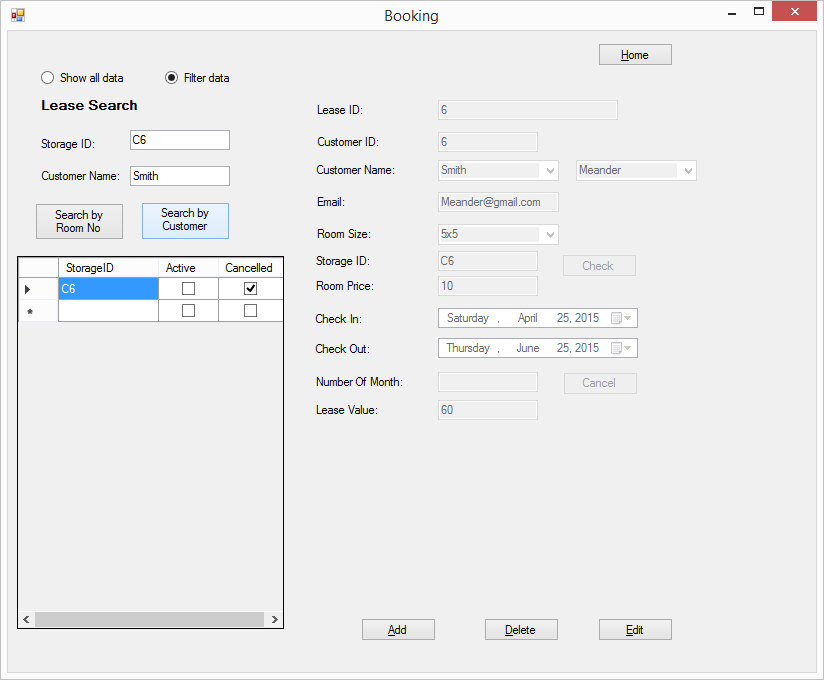


*Fig 3:Filter based on criteria*

Once the search is performed based on a criteria, the grid below displays the list of bookings both active and inactive leases in the system.

The grid also displays the current status of the lease and if it is cancelled.

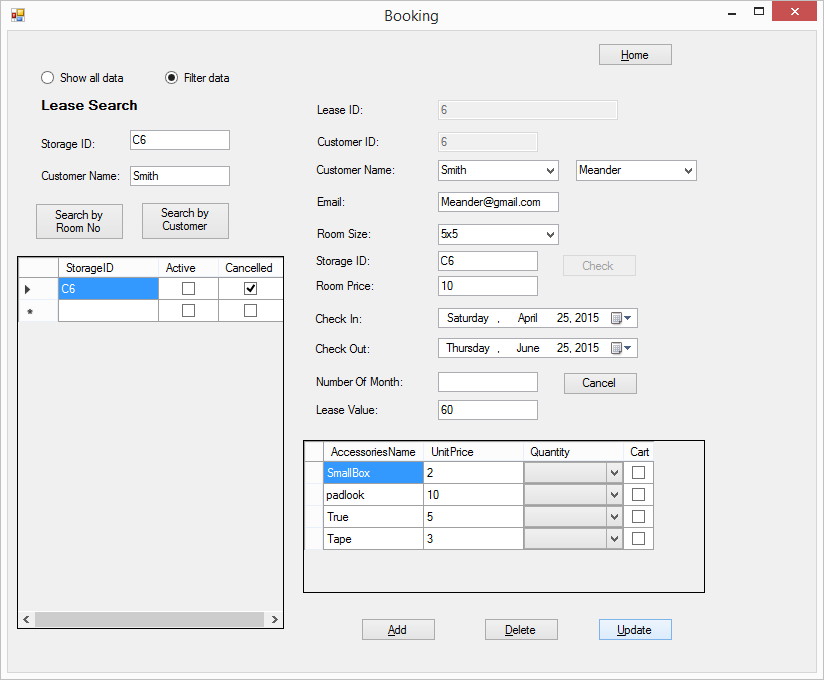
On clicking a particular row in the grid the complete details of the lease associated with it will be displayed on the right half of the screen.



*Fig 4: Searching result based on Customer First Name*

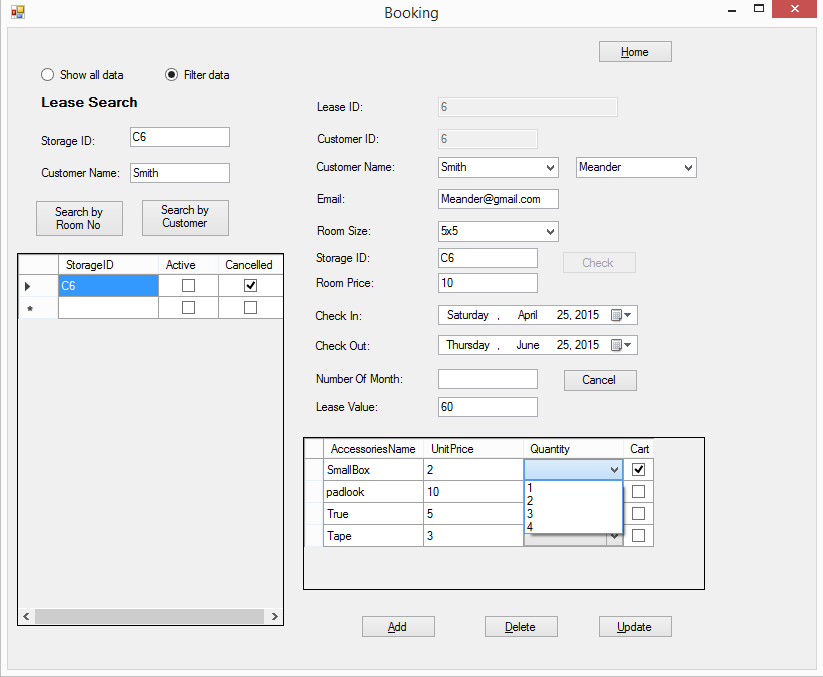
**Modifying an existing lease:**

After retrieving an existing lease from the database, the employee can modify the lease by clicking the *Edit Button* which makes the details into edit mode.



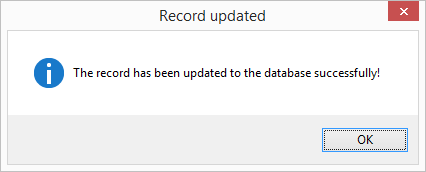
*Fig 5: Edit button is clicked*

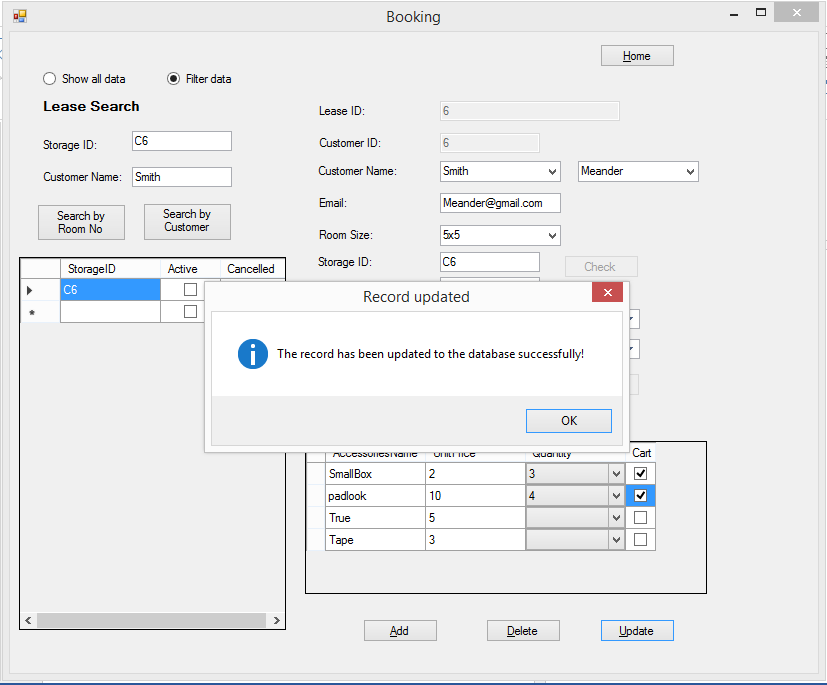
The employee can add accesories as shown below:

****

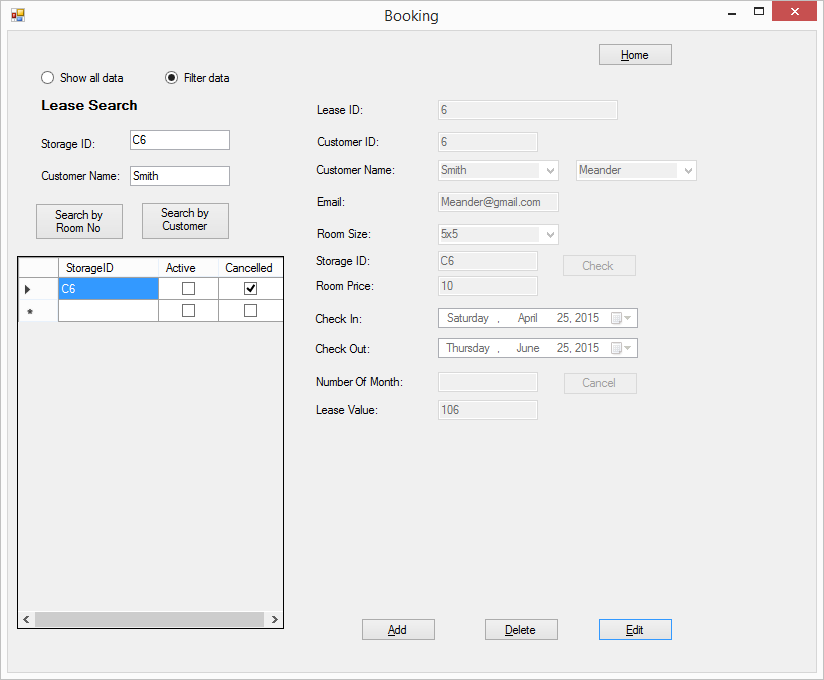
*Fig 6: Editing the existing lease*

On clicking the *Update button* the system shows a pop-up that the record has been updated in the database.



****

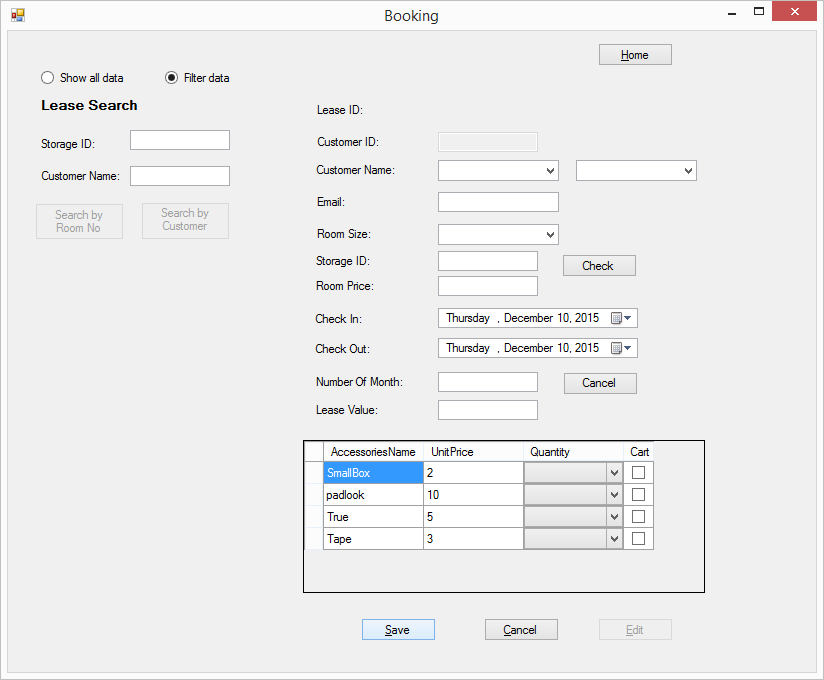
*Fig 7: Update button is clicked*



*Fig 8: Ok button is clicked on the pop-up*

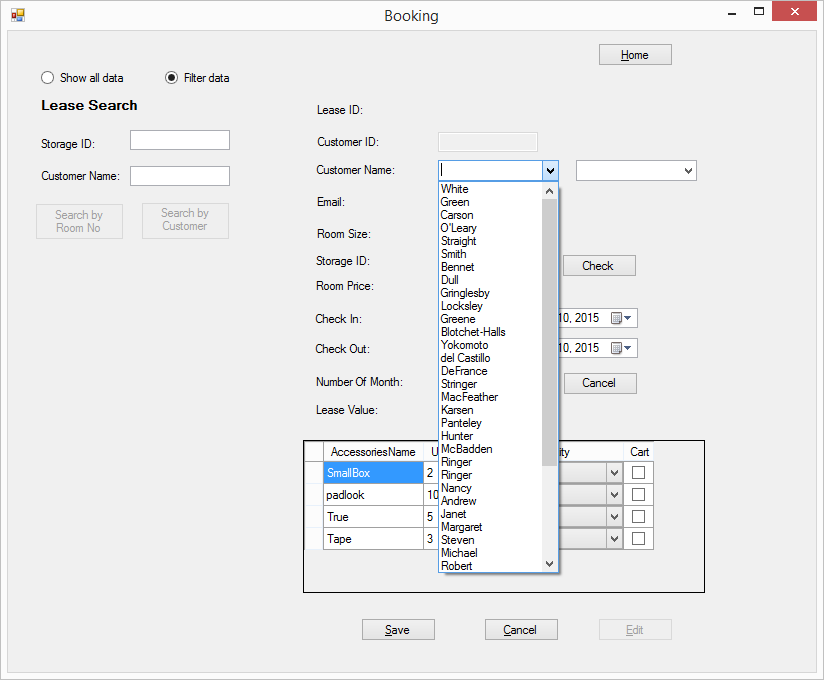
**Adding a New Lease:**

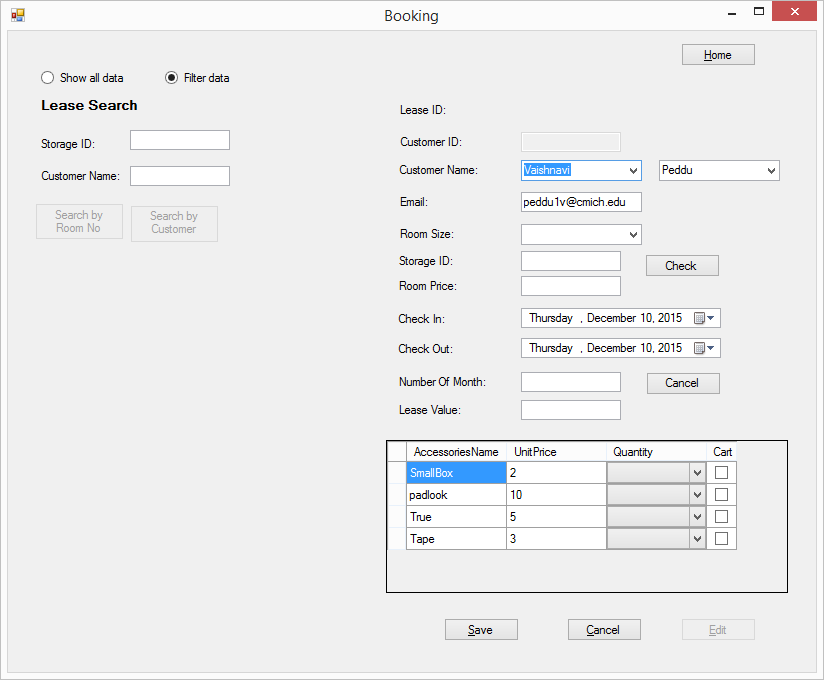
The Employee can add a new lease by clicking on the *Add* button . Once the *Add button* is clicked an empty booking form is displayed as shown below.



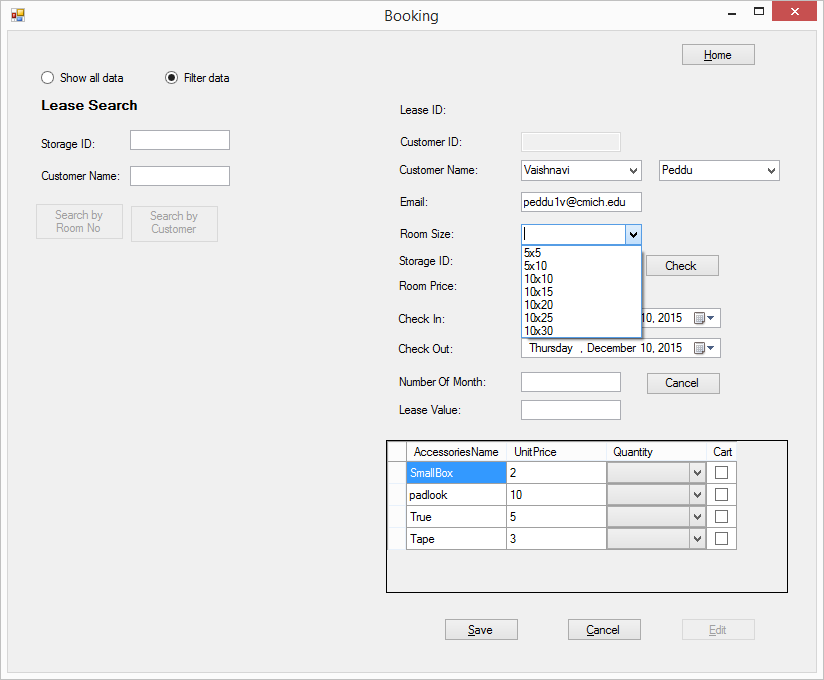
*Fig 9: Add button is clicked*

The Employee can fill out the form from the dynamically- bound drop down boxes as shown below.



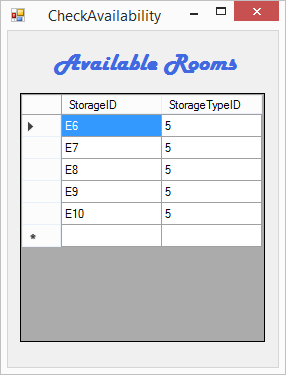


*Fig 10: Employee can either select the first name or last name of the customer*

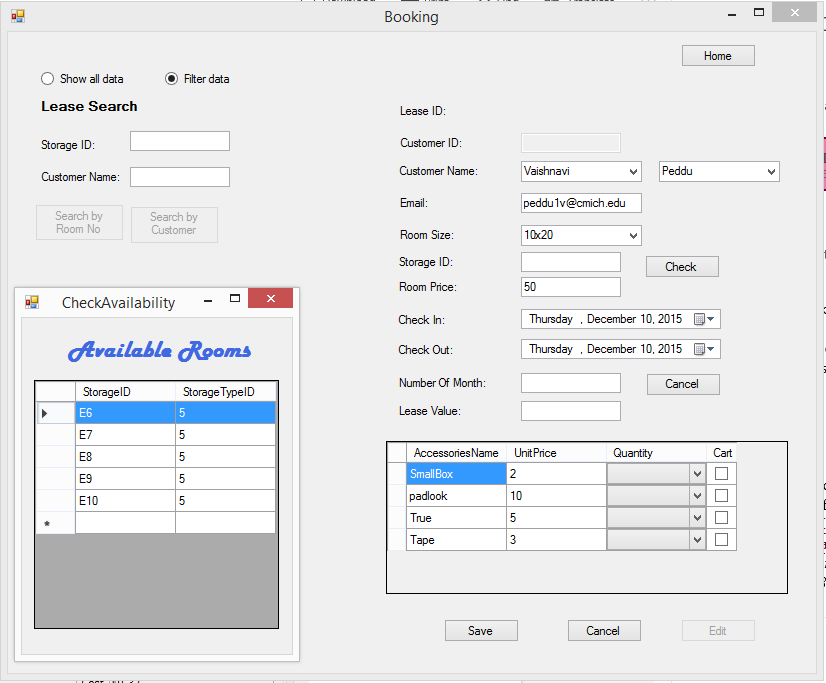


*Fig 10: The Room Size can be selected*

Once the *Room Size* is selected, the **Check Button** should be clicked which displays the Available Rooms for the selected Room Size as shown below.



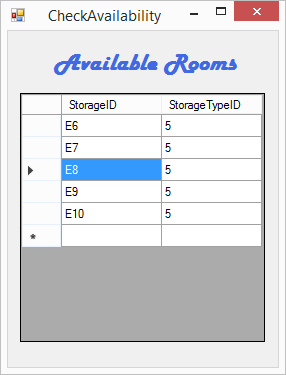
*Fig 11:Pop-up of Available Room of Size 10x20*



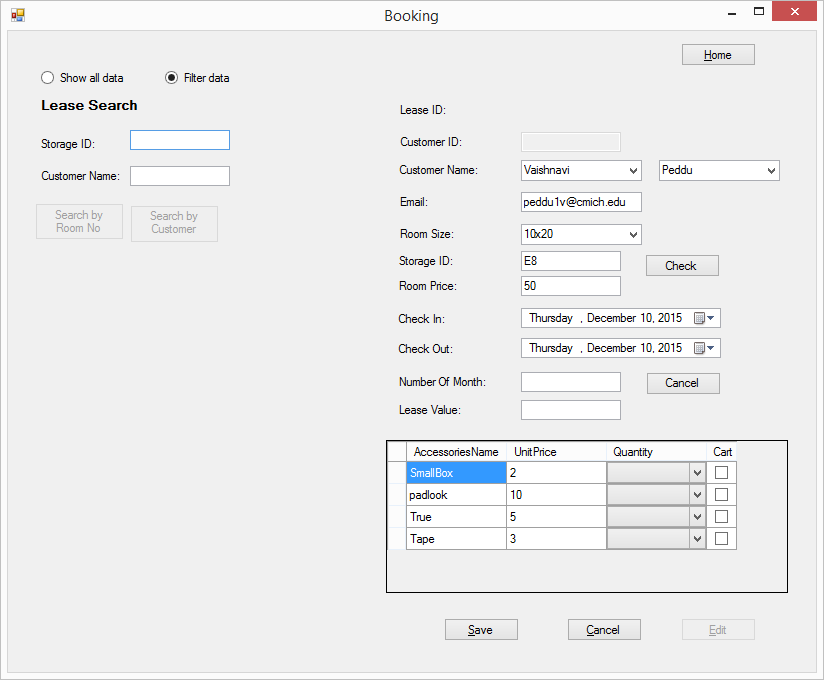
*Fig 12:Check button is clicked*

The employee can select from the displayed list of available storages by clicking on the Storage ID as shown below.

Once the Storage ID is selected the number automatically gets populated into the Booking Form.

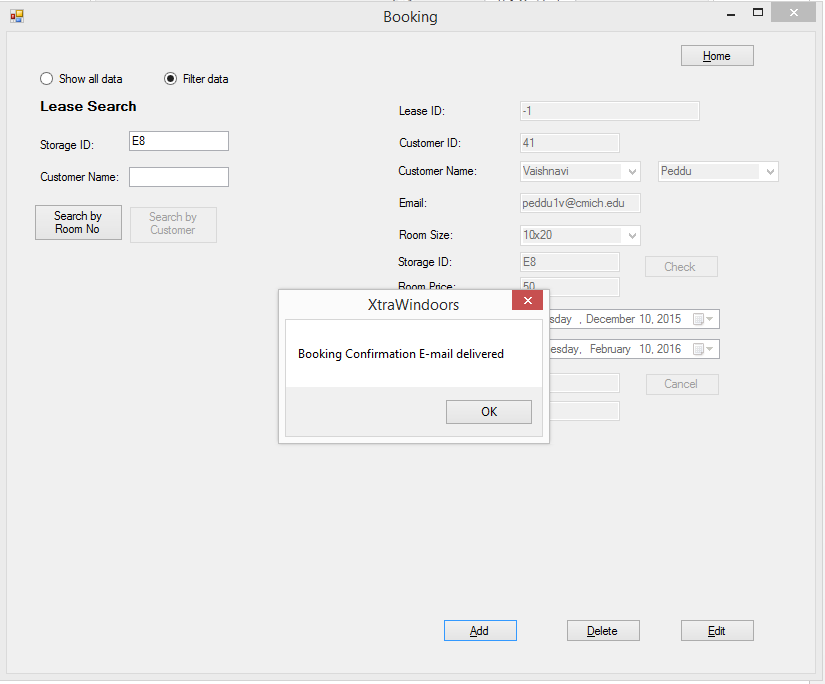


*Fig 13:Selecting a Storage ID from the displayed list*

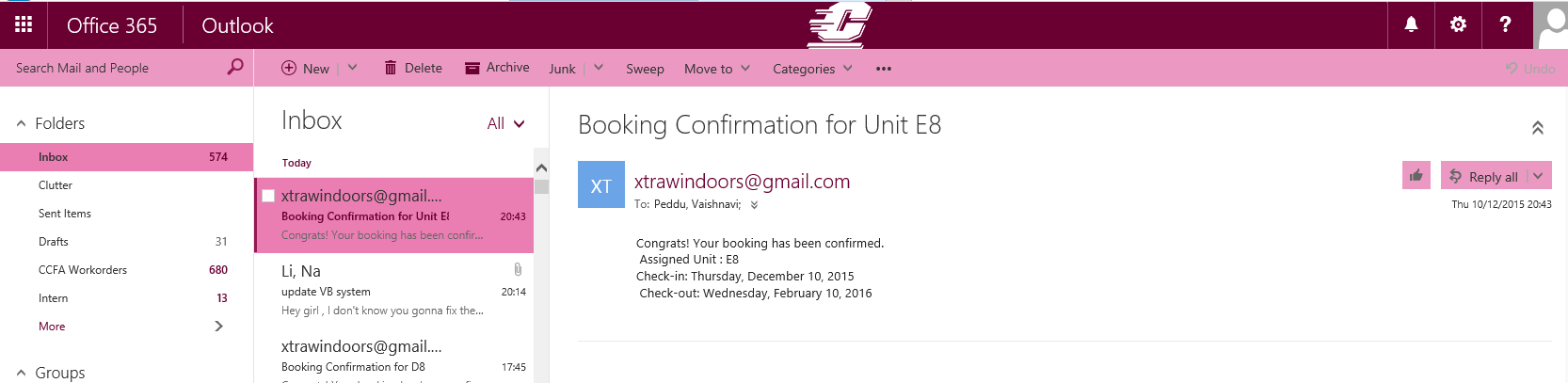


*Fig 14:Storage Id populated in the form*

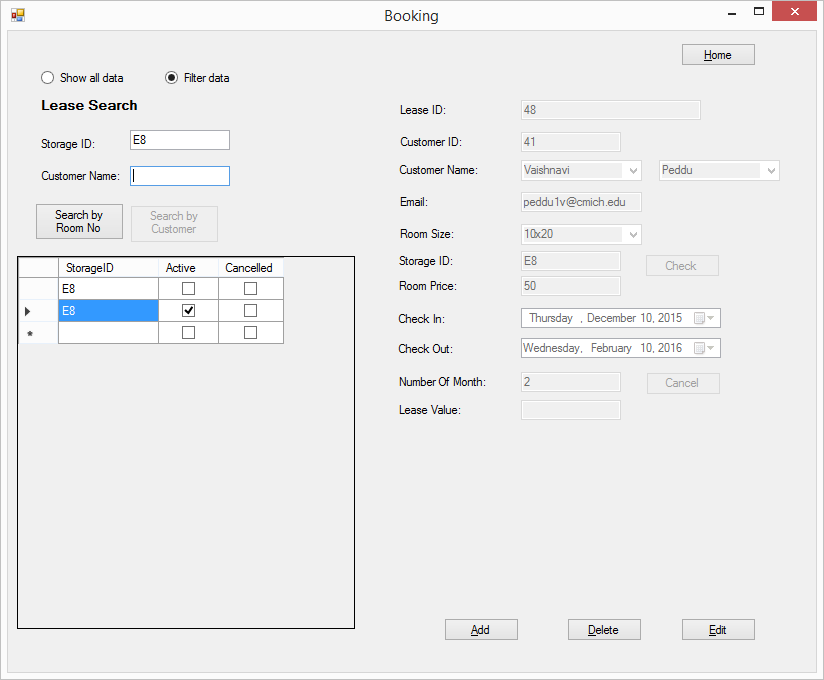
On clicking the *Save button,* the new lease is generated and saved to the database. The customer received the *Booking Confirmation E-mail* after successful booking



*Fig 15:Save button is clicked*



*Fig 16:E-mail received by the customer*

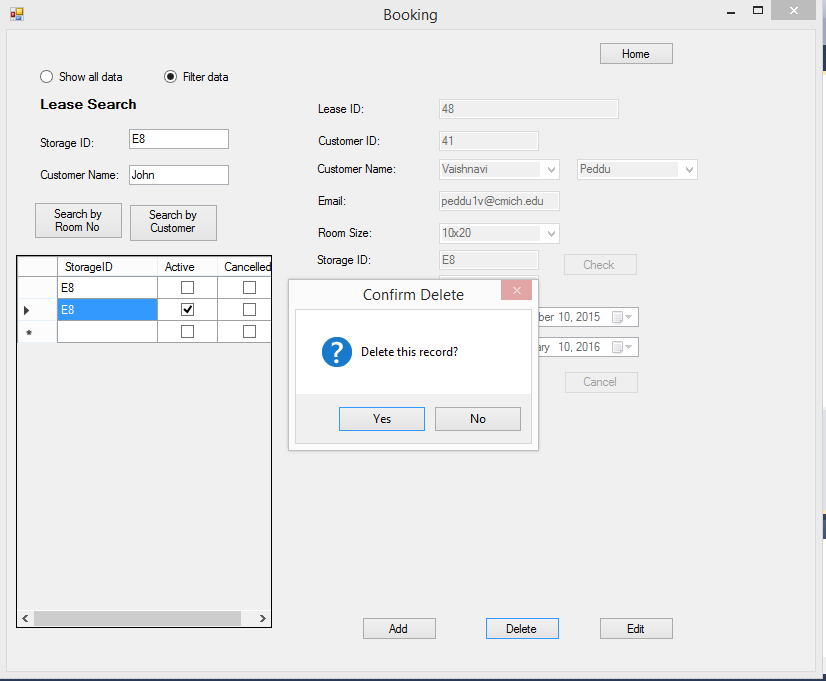


*Fig 17:Search result of the newly booked lease*

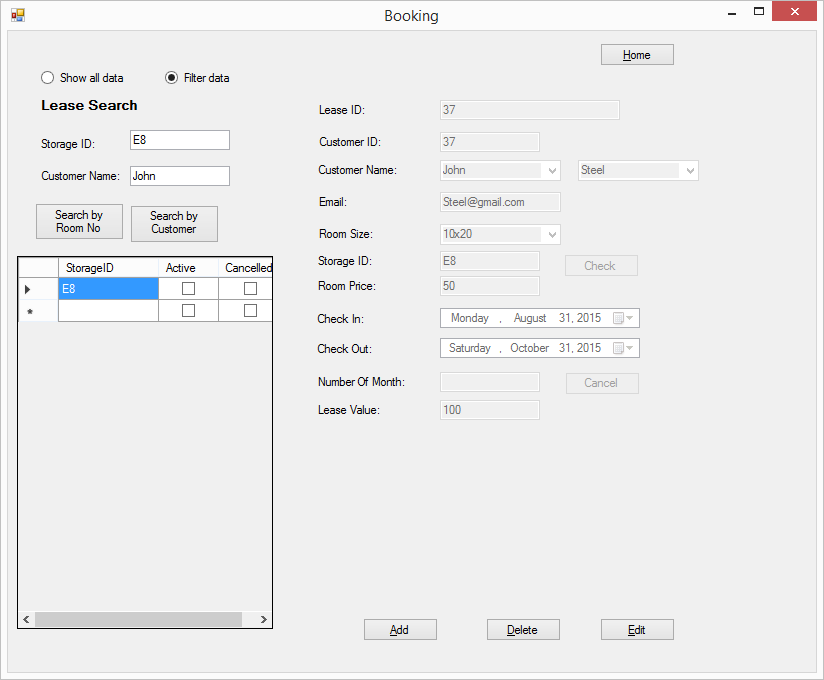
**Delete Existing Lease:**

In case if the customer decides not to reserve the lease, the employee can *Delete* the lease by clicking the *Delete Lease button.* It results in a dialogue as shown below.

On clicking on *Yes* the record will be deleted from the database.



*Fig 18:Delete a lease*



*Fig 18:Search result after successfully delteing the lease*

**Booking Form Code:**

'Program Name: XtraWindoors Rental System

'Programmer: Group 7

'Date: 12/10/2015

'Class: Booking Form

'Final Copy

Imports System.Net.Mail

Public Class BookingForm

Private AddingBoolean As Boolean = False

Private EditingBoolean As Boolean = False

Private ClosingBoolean As Boolean = False

Private Sub CustomerBindingNavigatorSaveItem\_Click(sender As System.Object, e As System.EventArgs)

Me.Validate()

Me.CustomerBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.XtraWindoorsDataSet)

End Sub

Private Sub BookingForm\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

Me.InvoiceTableAdapter.Fill(Me.XtraWindoorsDataSet.Invoice)

Me.AccessoriesTableAdapter.Fill(Me.XtraWindoorsDataSet.Accessories)

Me.StorageTableAdapter.Fill(Me.XtraWindoorsDataSet.Storage)

Me.StorageTypeTableAdapter.Fill(Me.XtraWindoorsDataSet.StorageType)

Me.CustomerTableAdapter.Fill(Me.XtraWindoorsDataSet.Customer)

'bind the form controls

ClearPanel()

SetControlsReadOnly(False)

SetValues()

Dim Qty As New DataGridViewComboBoxColumn()

With Qty

'.Width = 30

.HeaderText = "Quantity"

.Name = "Qty"

.MaxDropDownItems = 4

.Items.Add("1")

.Items.Add("2")

.Items.Add("3")

.Items.Add("4")

End With

AccesoriesDataGridView.Columns.Add(Qty)

Dim chk As New DataGridViewCheckBoxColumn()

AccesoriesDataGridView.Columns.Add(chk)

With chk

.HeaderText = "Cart"

.Name = "chk"

.Width = 30

End With

AccesoriesDataGridView.Rows(2).Cells(0).Value = True

LeaseDataGridView.Columns(1).HeaderText = "Active"

LeaseDataGridView.Columns(1).Width = 60

LeaseDataGridView.Columns(2).HeaderText = "Cancelled"

LeaseDataGridView.Columns(2).Width = 65

StorageIDTextBox.Text = ""

StorageIDTextBox.Focus()

End Sub

Private Sub SearchIDButton\_Click(sender As System.Object, e As System.EventArgs) Handles SearchIDButton.Click

Try

LeaseDataGridView.Visible = True

'TODO: This line of code loads data into the 'XtraWindoorsDataSet.Lease' table. You can move, or remove it, as needed.

'Me.LeaseBindingSource.DataSource = XtraWindoorsDataSet.Lease

Me.LeaseTableAdapter.FillByStorageID(Me.XtraWindoorsDataSet.Lease, StorageIDTextBox.Text)

Me.CustomerTableAdapter.Fill(Me.XtraWindoorsDataSet.Customer)

Me.CustomerBindingSource.Filter = "CustomerID = '" & CustomerIDTextBox.Text & "'"

Me.StorageTypeTableAdapter.FillByStorageID(Me.XtraWindoorsDataSet.StorageType, StorageIDTextBox.Text)

SetControlsReadOnly(False)

FirstNameTextBox.Text = FirstNameComboBox.Text

StorageIDTextBox.Text = StorageIDTextBox1.Text

Catch ex As System.Exception

MessageBox.Show("No record found")

ClearPanel()

'System.Windows.Forms.MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub SearchCustomerButton\_Click(sender As System.Object, e As System.EventArgs) Handles SearchCustomerButton.Click

Try

LeaseDataGridView.Visible = True

'TODO: This line of code loads data into the 'XtraWindoorsDataSet.Lease' table. You can move, or remove it, as needed.

Me.LeaseTableAdapter.FillByFirstName(Me.XtraWindoorsDataSet.Lease, FirstNameTextBox.Text)

Me.StorageTypeTableAdapter.FillByFirstName(Me.XtraWindoorsDataSet.StorageType, FirstNameTextBox.Text)

Me.CustomerTableAdapter.Fill(Me.XtraWindoorsDataSet.Customer)

Me.CustomerBindingSource.Filter = "CustomerID = '" & CustomerIDTextBox.Text & "'"

Catch ex As System.Exception

MessageBox.Show("No record found")

ClearPanel()

End Try

SetControlsReadOnly(False)

End Sub

Private Sub FirstNameTextBox\_TextChanged(sender As Object, e As System.EventArgs) Handles FirstNameTextBox.TextChanged

If String.IsNullOrWhiteSpace(FirstNameTextBox.Text) Then

SearchCustomerButton.Enabled = False

Else

SearchCustomerButton.Enabled = True

End If

End Sub

Private Sub StorageIDTextBox\_TextChanged(sender As Object, e As System.EventArgs) Handles StorageIDTextBox.TextChanged

If String.IsNullOrWhiteSpace(StorageIDTextBox.Text) And ShowRadioButton.Checked = True Then

LeaseDataGridView.DataSource = LeaseBindingSource

LeaseBindingSource.Filter = Nothing

Me.LeaseTableAdapter.Fill(Me.XtraWindoorsDataSet.Lease)

ElseIf String.IsNullOrWhiteSpace(StorageIDTextBox.Text) Then

SearchIDButton.Enabled = False

Else

SearchIDButton.Enabled = True

End If

End Sub

Private Sub NumberOfMonthTextBox\_TextChanged(sender As Object, e As System.EventArgs) Handles NumberOfMonthTextBox.TextChanged

Try

If String.IsNullOrWhiteSpace(NumberOfMonthTextBox.Text) = False And String.IsNullOrWhiteSpace(TypePriceTextBox.Text) = False Then

CostTextBox.Text = TypePriceTextBox.Text \* NumberOfMonthTextBox.Text

CheckOutDateTimePicker.Value = CheckOutDateTimePicker.Value.AddMonths(NumberOfMonthTextBox.Text)

Else

CostTextBox.Text = ""

CheckOutDateTimePicker.Value = System.DateTime.Today

End If

Catch ex As System.Exception

System.Windows.Forms.MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub FilterRadioButton\_CheckedChanged(sender As Object, e As System.EventArgs) Handles FilterRadioButton.CheckedChanged

ClearPanel()

LeaseDataGridView.Visible = False

'ClearGrid()

End Sub

Private Sub ShowRadioButton\_CheckedChanged(sender As Object, e As System.EventArgs) Handles ShowRadioButton.CheckedChanged

ClearPanel()

LeaseDataGridView.Visible = True

ShowGrid()

End Sub

Private Sub LeaseDataGridView\_CellClick(sender As Object, e As System.Windows.Forms.DataGridViewCellEventArgs) Handles LeaseDataGridView.CellClick

Try

Dim i As Integer

i = LeaseDataGridView.CurrentRow.Index

Me.StorageIDTextBox.Text = Me.LeaseDataGridView.Item(0, i).Value

Me.StorageIDTextBox1.Text = Me.StorageIDTextBox.Text

'TODO: This line of code loads data into the 'XtraWindoorsDataSet.Lease' table. You can move, or remove it, as needed.

'Me.LeaseTableAdapter.FillByStorageID(Me.XtraWindoorsDataSet.Lease, StorageIDTextBox.Text)

Me.CustomerTableAdapter.Fill(Me.XtraWindoorsDataSet.Customer)

Me.CustomerBindingSource.Filter = "CustomerID = '" & CustomerIDTextBox.Text & "'"

Me.StorageTypeTableAdapter.FillByStorageID(Me.XtraWindoorsDataSet.StorageType, StorageIDTextBox.Text)

Catch ex As System.Exception

System.Windows.Forms.MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub LeaseDataGridView\_DataBindingComplete(sender As Object, e As System.Windows.Forms.DataGridViewBindingCompleteEventArgs) Handles LeaseDataGridView.DataBindingComplete

Try

Dim i As Integer

i = LeaseDataGridView.CurrentRow.Index

Me.StorageIDTextBox.Text = Me.LeaseDataGridView.Item(0, i).Value

Me.StorageIDTextBox1.Text = Me.LeaseDataGridView.Item(0, i).Value

Catch ex As Exception

End Try

End Sub

Private Function GridViewRow() As Object

Throw New NotImplementedException

End Function

Private Sub AddSaveButton\_Click(sender As System.Object, e As System.EventArgs) Handles AddSaveButton.Click

If AddSaveButton.Text = "&Add" Then

'Add Button is clicked

ClearPanel()

AddingBoolean = True

SetControlsReadOnly(True)

SetButtonForEdit()

AddSaveButton.Text = "&Save"

CustomerBindingSource.Filter = Nothing

StorageTypeBindingSource.Filter = Nothing

btnCheck.Enabled = True

btnCancelLease.Enabled = True

LeaseIDTextBox.Visible = False

Else

Try

Dim ctl1 As Control

For Each ctl1 In SplitContainer1.Panel2.Controls

If TypeOf ctl1 Is TextBox Then

If IsNothing(ctl1) Then

BookingErrorProvider.SetError(ctl1, "Manditory Field")

End If

ElseIf TypeOf ctl1 Is ComboBox Then

If IsNothing(ctl1) Then

BookingErrorProvider.SetError(ctl1, "Manditory Field")

End If

End If

Next

Dim q, c, Acctotal As Int32

Acctotal = 0

For Each row As DataGridViewRow In AccesoriesDataGridView.Rows

If row.Cells("chk").Value = True Then

q = row.Cells("Qty").Value

c = row.Cells(1).Value

If q = 0 Then

Acctotal = Acctotal + (1 \* c).ToString

Else

Acctotal = Acctotal + (q \* c).ToString

End If

End If

Next

If String.IsNullOrWhiteSpace(CostTextBox.Text) Then

CostTextBox.Text = 0

End If

CostTextBox.Text = Acctotal + CostTextBox.Text

LeaseBindingSource.EndEdit()

btnCancelLease.Enabled = False

btnCheck.Enabled = False

LeaseIDTextBox.Visible = True

Dim ANewLeaseRow As DataRow = XtraWindoorsDataSet.Lease.NewRow

ANewLeaseRow("CustomerID") = FirstNameComboBox.SelectedValue

ANewLeaseRow("StorageID") = StorageIDTextBox1.Text

ANewLeaseRow("LeaseActive") = 1

ANewLeaseRow("CheckIn") = CheckInDateTimePicker.Value

ANewLeaseRow("CheckOut") = CheckOutDateTimePicker.Value

ANewLeaseRow("LeaseCancelled") = 0

XtraWindoorsDataSet.Lease.Rows.Add(ANewLeaseRow)

'declare a variable to add a row to the dataset

'Update data to database

LeaseTableAdapter.Update(XtraWindoorsDataSet.Lease)

AddingBoolean = False

EditingBoolean = False

SetControlsReadOnly(False)

ResetButtonafterEdit()

Try

Dim SmtpServer As New SmtpClient()

Dim mail As New MailMessage()

SmtpServer.Credentials = New Net.NetworkCredential("xtrawindoors@gmail.com", "GalaxySoft")

SmtpServer.Port = 587

SmtpServer.EnableSsl = True

SmtpServer.Host = "smtp.gmail.com"

mail = New MailMessage()

mail.From = New MailAddress("xtrawindoors@gmail.com")

mail.To.Add(EmailTextBox1.Text)

mail.Subject = "Booking Confirmation for Unit " & StorageIDTextBox1.Text

mail.Body = String.Format("Congrats! Your booking has been confirmed." & vbNewLine & " Assigned Unit : {0} " & vbNewLine & "Check-in: {1}" & vbNewLine & " Check-out: {2}", StorageIDTextBox1.Text, CheckInDateTimePicker.Text, CheckOutDateTimePicker.Text)

SmtpServer.Send(mail)

MsgBox("Booking Confirmation E-mail delivered")

Catch ex As Exception

MsgBox(ex.ToString)

End Try

Catch ex As Exception

MessageBox.Show("DataBase Operation Error :" & ex.Message, "Adding Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End If

End Sub

Private Sub DeleteCancelButton\_Click(sender As System.Object, e As System.EventArgs) Handles DeleteCancelButton.Click

'Delete the current record after coonfirming or cancel and add or Edit

Dim DeleteDialogResult As DialogResult

Try

If DeleteCancelButton.Text = "&Delete" Then

'delete records

DeleteDialogResult = MessageBox.Show("Delete this record?", "Confirm Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If DeleteDialogResult = Windows.Forms.DialogResult.Yes Then

LeaseBindingSource.RemoveCurrent()

LeaseTableAdapter.Update(XtraWindoorsDataSet.Lease)

End If

Else

'Cancel button is clicked

LeaseBindingSource.CancelEdit()

AddingBoolean = False

EditingBoolean = False

SetControlsReadOnly(False)

btnCancelLease.Enabled = False

LeaseIDTextBox.Visible = True

btnCheck.Enabled = False

ResetButtonafterEdit()

ClearPanel()

End If

Catch ex As Exception

MessageBox.Show("unable to complete the delete/cancel operation:" & ex.Message, "Delete/Cancel", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Private Sub ClearPanel()

Dim ctl1 As Control

For Each ctl1 In SplitContainer1.Panel2.Controls

If TypeOf ctl1 Is TextBox Then

ctl1.Text = ""

ElseIf TypeOf ctl1 Is ComboBox Then

ctl1.Text = ""

End If

Next

End Sub

Private Sub SetControlsReadOnly(ByVal ValueBoolean As Boolean)

Dim ctl1 As Control

For Each ctl1 In SplitContainer1.Panel2.Controls

If TypeOf ctl1 Is TextBox Then

ctl1.Enabled = ValueBoolean

ElseIf TypeOf ctl1 Is ComboBox Then

ctl1.Enabled = ValueBoolean

ElseIf TypeOf ctl1 Is DateTimePicker Then

ctl1.Enabled = ValueBoolean

ElseIf TypeOf ctl1 Is DataGridView Then

ctl1.Visible = ValueBoolean

End If

CustomerIDTextBox.Enabled = False

LeaseIDTextBox.Enabled = False

Next

End Sub

Private Sub ResetButtonafterEdit()

AddSaveButton.Text = "&Add"

DeleteCancelButton.Text = "&Delete"

EditButton.Enabled = True

End Sub

Private Sub EditButton\_Click(sender As System.Object, e As System.EventArgs) Handles EditButton.Click

EditingBoolean = True

btnCancelLease.Enabled = True

SetControlsReadOnly(True)

'SetButtonForEdit()

If EditButton.Text = "&Edit" Then

'Edit Button is clicked

EditingBoolean = True

SetControlsReadOnly(True)

EditButton.Text = "&Update"

'SetButtonForEdit()

Else

Try

Dim ctl1 As Control

For Each ctl1 In SplitContainer1.Panel2.Controls

If TypeOf ctl1 Is TextBox Then

If IsNothing(ctl1) Then

BookingErrorProvider.SetError(ctl1, "Manditory Field")

End If

ElseIf TypeOf ctl1 Is ComboBox Then

If IsNothing(ctl1) Then

BookingErrorProvider.SetError(ctl1, "Manditory Field")

End If

End If

Next

Dim q, c, Acctotal As Int32

Acctotal = 0

For Each row As DataGridViewRow In AccesoriesDataGridView.Rows

If row.Cells("chk").Value = True Then

q = row.Cells("Qty").Value

c = row.Cells(1).Value

If q = 0 Then

Acctotal = Acctotal + (1 \* c).ToString

Else

Acctotal = Acctotal + (q \* c).ToString

End If

End If

Next

If String.IsNullOrWhiteSpace(CostTextBox.Text) Then

CostTextBox.Text = 0

End If

CostTextBox.Text = Acctotal + CostTextBox.Text

LeaseBindingSource.EndEdit()

btnCancelLease.Enabled = False

XtraWindoorsDataSet.Tables("Lease").Rows(0).Item("StorageID") = StorageIDTextBox1.Text

XtraWindoorsDataSet.Tables("Lease").Rows(0).Item("CheckIn") = CheckInDateTimePicker.Value

XtraWindoorsDataSet.Tables("Lease").Rows(0).Item("CheckOut") = CheckOutDateTimePicker.Value

'Update data to database

LeaseBindingSource.EndEdit()

LeaseTableAdapter.Update(XtraWindoorsDataSet.Lease)

MessageBox.Show("The record has been updated to the database successfully!", "Record updated", MessageBoxButtons.OK, MessageBoxIcon.Information)

AddingBoolean = False

EditingBoolean = False

SetControlsReadOnly(False)

EditButton.Text = "&Edit"

Catch ex As Exception

MessageBox.Show("DataBase Operation Error :" & ex.Message, "Adding Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End If

End Sub

Private Sub SetButtonForEdit()

AddSaveButton.Text = "&Save"

DeleteCancelButton.Text = "&Cancel"

EditButton.Enabled = False

End Sub

Private Sub SetValues()

With FirstNameComboBox

.DataSource = CustomerBindingSource

.DisplayMember = "FirstName"

.ValueMember = "CustomerID"

.DataBindings.Add("text", CustomerBindingSource, "FirstName", False, DataSourceUpdateMode.Never)

.SelectedIndex = -1

End With

With LastNameComboBox

.DataSource = CustomerBindingSource

.DisplayMember = "LastName"

.ValueMember = "CustomerID"

.DataBindings.Add("text", CustomerBindingSource, "LastName", False, DataSourceUpdateMode.Never)

.SelectedIndex = -1

End With

With TypeSizeComboBox

.DataSource = StorageTypeBindingSource

.DisplayMember = "TypeSize"

.ValueMember = "StorageTypeID"

.DataBindings.Add("text", StorageTypeBindingSource, "TypeSize", False, DataSourceUpdateMode.Never)

.SelectedIndex = -1

End With

End Sub

Private Sub ClearGrid()

CustomerBindingSource.Filter = ""

LeaseDataGridView.DataSource = Nothing

'LeaseDataGridView.Refresh()

End Sub

Private Sub ShowGrid()

LeaseDataGridView.DataSource = LeaseBindingSource

Me.LeaseTableAdapter.Fill(Me.XtraWindoorsDataSet.Lease)

LeaseDataGridView.Columns(1).HeaderText = "Active"

LeaseDataGridView.Columns(1).Width = 50

LeaseDataGridView.Columns(2).HeaderText = "Cancelled"

LeaseDataGridView.Columns(2).Width = 55

End Sub

Private Sub btnCheck\_Click(sender As System.Object, e As System.EventArgs) Handles btnCheck.Click

If TypeSizeComboBox.SelectedIndex = -1 Then

Else

CheckAvailability.GetStorageID(TypeSizeComboBox.SelectedValue)

CheckAvailability.Show()

End If

End Sub

Private Sub HomeButton\_Click(sender As System.Object, e As System.EventArgs) Handles HomeButton.Click

Me.Close()

HomePageScreen.Show()

End Sub

Private Sub BookingForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

HomePageScreen.Show()

End Sub

Private Sub btnCancelLease\_Click(sender As System.Object, e As System.EventArgs) Handles btnCancelLease.Click

Dim DeleteDialogResult As DialogResult

DeleteDialogResult = MessageBox.Show("Cancel current Lease? Click 'Yes' to delete, 'No' to cancel.", "Delete Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If DeleteDialogResult = Windows.Forms.DialogResult.Yes Then

CheckOutDateTimePicker.Value = System.DateTime.Now

MessageBox.Show("The Lease has been cancelled!", "Fine updated", MessageBoxButtons.OK, MessageBoxIcon.Information)

If String.IsNullOrWhiteSpace(CostTextBox.Text) Then

CostTextBox.Text = 0

End If

CostTextBox.Text = 50 + CostTextBox.Text

XtraWindoorsDataSet.Tables("Lease").Rows(0).Item("LeaseCancelled") = 1

'Update data to database

LeaseBindingSource.EndEdit()

LeaseTableAdapter.Update(XtraWindoorsDataSet.Lease)

End If

End Sub

End Class

**Check Availability Form Code:**

'Program Name: XtraWindoors Rental System

'Programmer: Group 7

'Date: 12/10/2015

'Class: Check Availability Form

Public Class CheckAvailability

Dim StorageID As Integer

Private Sub CheckAvailability\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

End Sub

Public Sub GetStorageID(StorId As Integer)

StorageID = StorId

End Sub

Private Sub StorageBindingNavigatorSaveItem\_Click(sender As System.Object, e As System.EventArgs)

Me.Validate()

Me.StorageBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.XtraWindoorsDataSet)

End Sub

Private Sub BookingForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

BookingForm.Show()

Me.StorageBindingSource.Filter = Nothing

End Sub

Private Sub CheckAvailability\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

Me.StorageBindingSource.Filter = "StorageTypeID = '" & StorageID & "'"

Me.StorageTableAdapter.FillByAvailability(Me.XtraWindoorsDataSet.Storage)

End Sub

Private Sub StorageDataGridView\_CellContentClick(sender As System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs) Handles StorageDataGridView.CellContentClick

Me.Hide()

Dim StorageID As String

StorageID = StorageDataGridView.CurrentCell.Value

BookingForm.StorageIDTextBox1.Text = StorageID

End Sub

End Class

**Employee form:**

Once we navigate from the home screen to the Employee form, the given image appears.

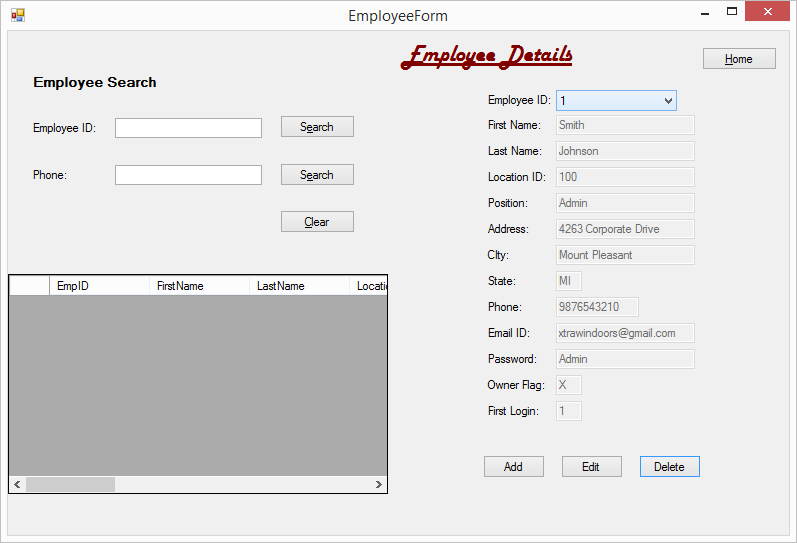


Fig 1: Employee form

This form helps to enter the details of the employee. The entered details can be saved and retrieved whenever necessary.

**Adding employee:**

For adding an employee, we need to click on the Add button at the right bottom of the screen. By clicking on the button, the given screen appears.

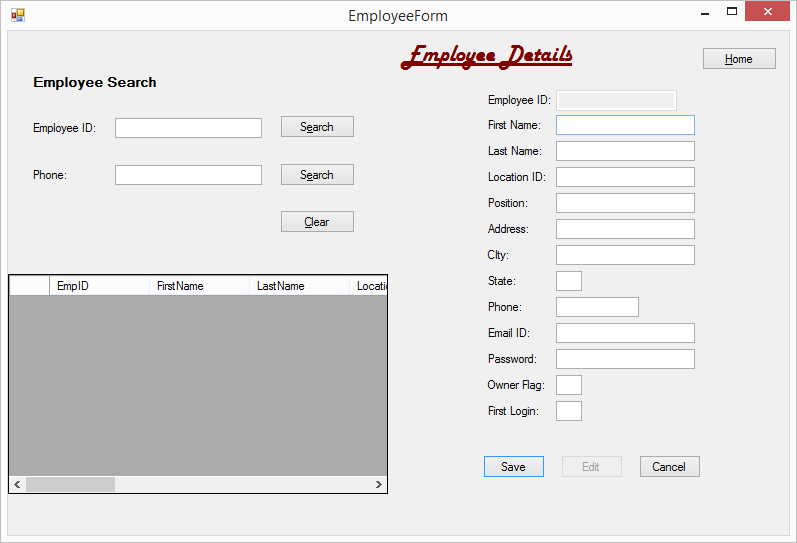


Fig 2: Adding employees

The given field names should be entered respectively and the Save button should be clicked.

**Saving employee details:**

The given screen appears while we are trying to enter the details of the employee and saving.

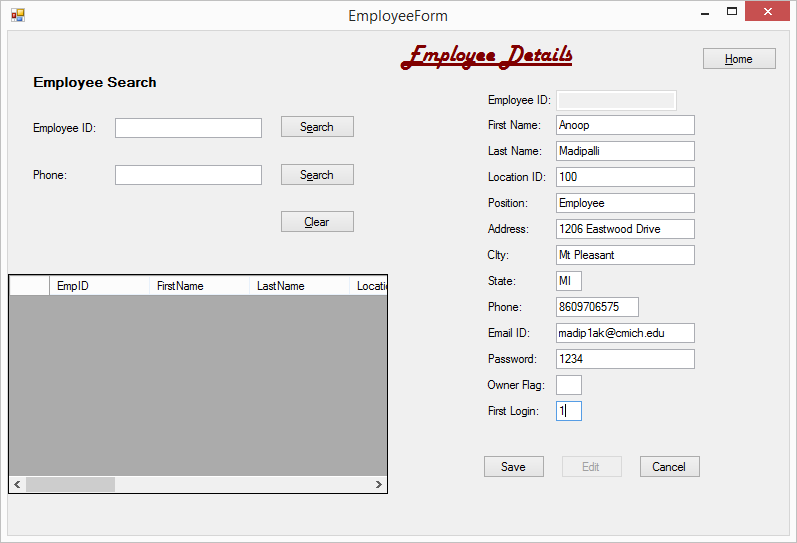


Fig 3: Entering the details of the employee

By clicking on the save button the given screen appears.



Fig: 4

A new password will be sent to the employee to the email mentioned.

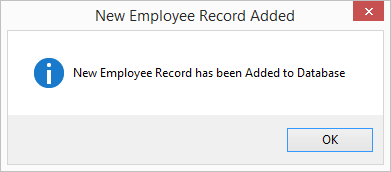


Fig: 5

This is the confirmation message box stating that a new employee has been created.

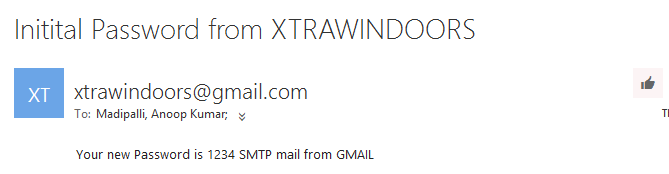


Fig: 6

This is the confirmation email sent to the employee. This password will be used by the employee to make conformation that the email has been received by the actual employee.

**Retrieving employee details:**

An employee ID will be automatically generated and with the help of this ID we can retrieve the details of the employee. The search option of the left top of the screen will help to get the details of the newly saved employee.

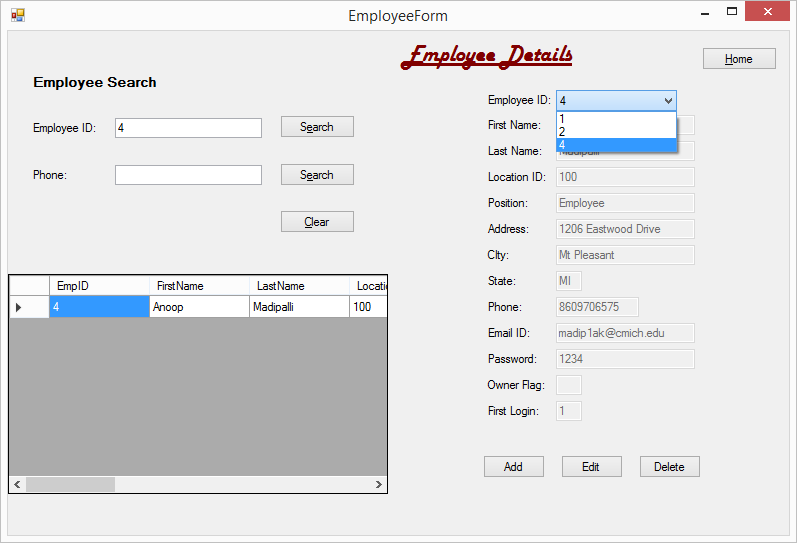


Fig: 7

The saved employee has Employee ID 4 that is automatically generated. This will help to search the employee details.

The employee details can be searched even with the help of phone number.

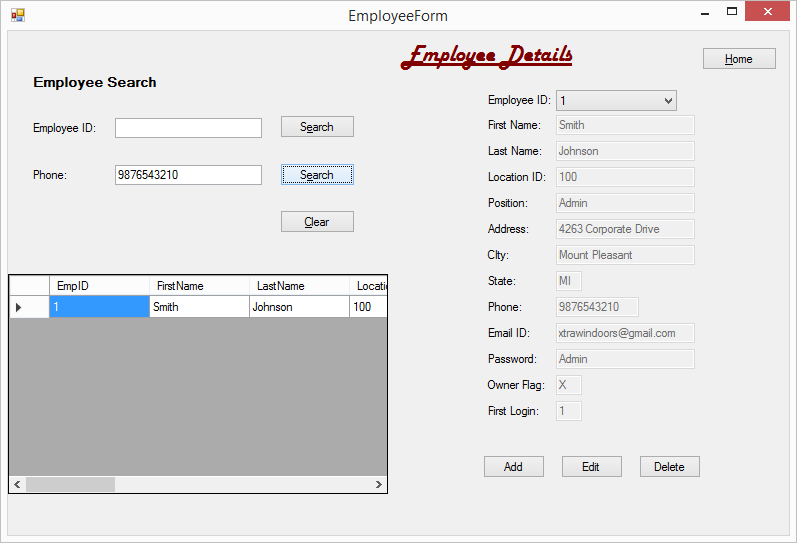


Fig: 8

The phone number of the person can help to retrieve details of the employee.

**Editing details:**

By clicking the edit button, the detail in the fields can be edited.



Fig: 9

Once the edit button is clicked, the data in the fields can be edited.

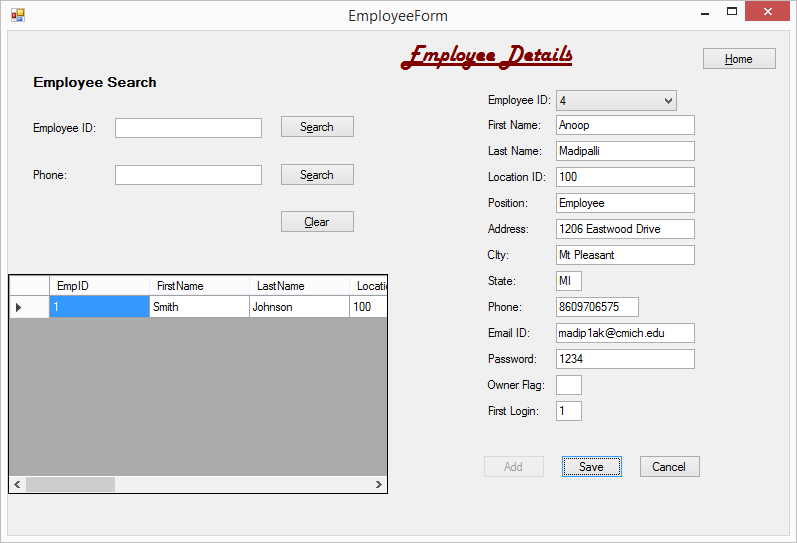


Fig 10: The fields can be edited.

The details of the employee are changed and can be seen in the detail entered by comparing it with Fig 10.

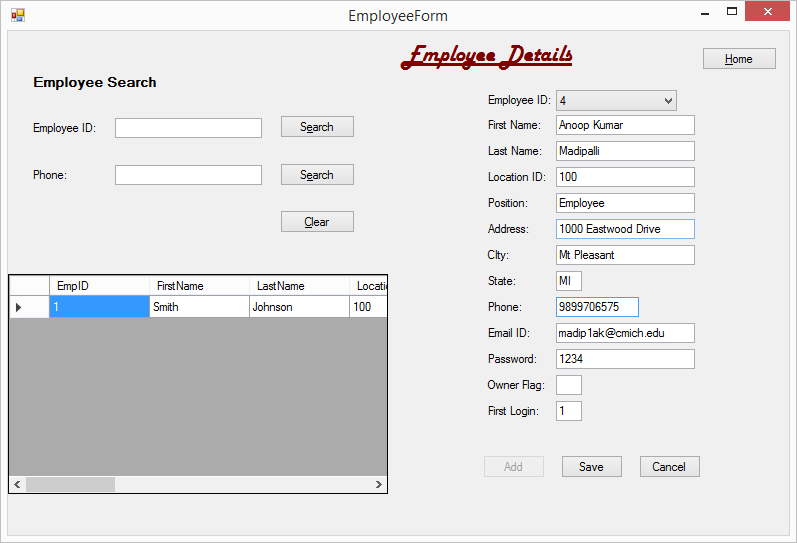


Fig 11: Edited details of the employee

By clicking the save button, the employee details will be saved again and can be retrieved. The given screen appears once the record is updated.

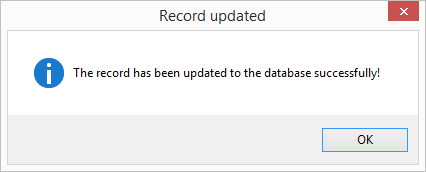


Fig 12: Confirm message of updated records.

**Delete records:**

This option helps to delete the records of the employee. A pop up screen appears asking for conformation to delete the record. Action is resumed after choosing the option.

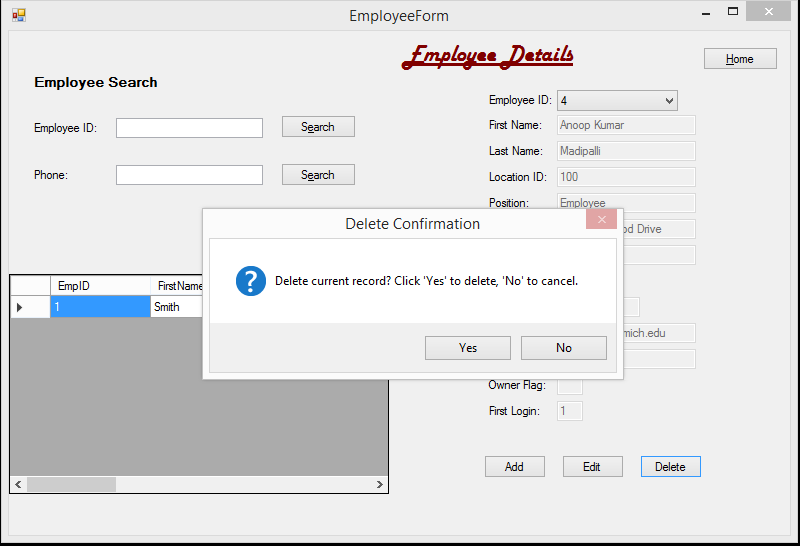


Fig: 13

The below given screen appears after deleting the record.

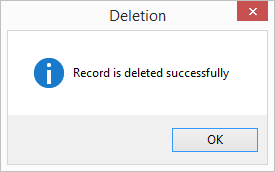


Fig: 14

The employee details cannot be retrieved from the database as the details are deleted.

**Going back to Home page:**

Once the functions are completed, in order to go to the home page, clicking the home button will navigate to the main page.

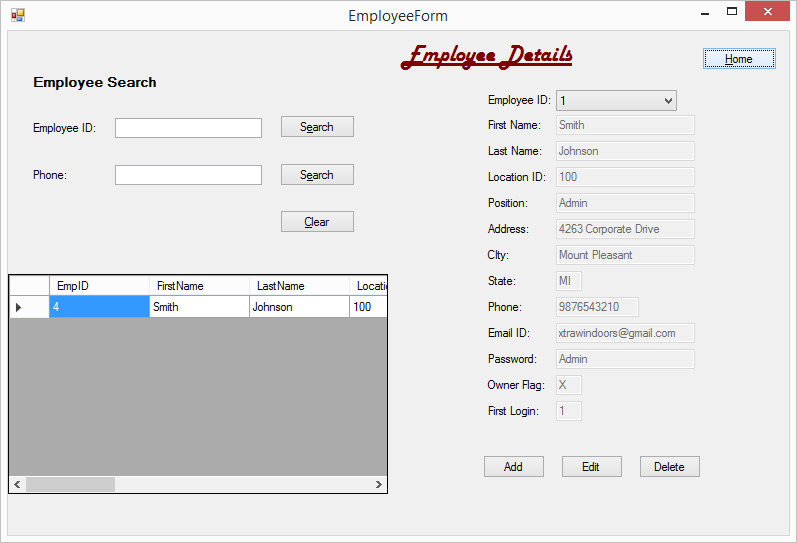


Fig 15:

Once the home button is clicked the given page appears on the screen.

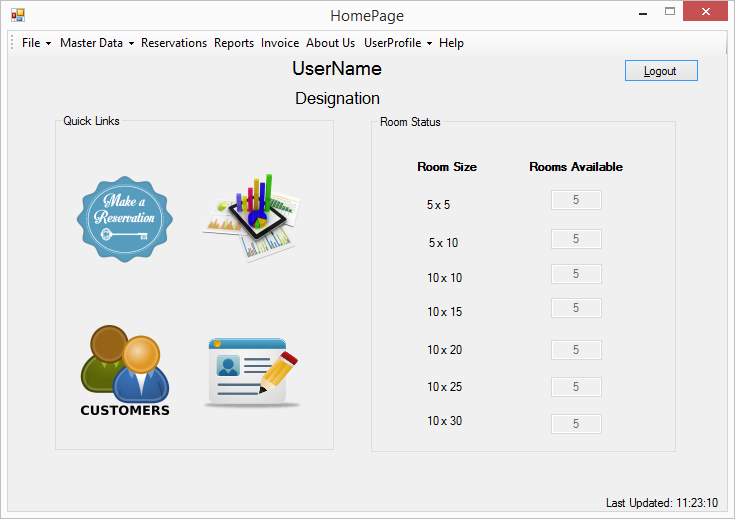


Fig 16: Home page

**Coding:**

'Program Name: XtraWindoors Rental System

'Programmer: Group 7

'Date: 12/10/2015

'Class: Employee Form

Imports System.Net.Mail

Public Class EmployeeForm

Private AXtraWindoorsDataSet As New XtraWindoorsDataSet

Private AEmployeeTableAdapter As New XtraWindoorsDataSetTableAdapters.EmployeeTableAdapter

Private ALocationTableAdapter As New XtraWindoorsDataSetTableAdapters.LocationTableAdapter

Private AEmployeeBindingSource As New BindingSource

Private AEmployeeGridBindingSource As New BindingSource

Private AddFlag As Boolean

Private EditFlag As Boolean

'Public Property EnforceConstraints As Boolean

Private Sub EmployeeForm\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

'TODO: This line of code loads data into the 'XtraWindoorsDataSet.Employee' table. You can move, or remove it, as needed.

'Me.EmployeeTableAdapter.Fill(Me.XtraWindoorsDataSet.Employee)

'EnforceConstraints = False

Try

AEmployeeTableAdapter = New XtraWindoorsDataSetTableAdapters.EmployeeTableAdapter

AXtraWindoorsDataSet = New XtraWindoorsDataSet

AEmployeeTableAdapter.Fill(AXtraWindoorsDataSet.Employee)

AEmployeeBindingSource = New BindingSource

AEmployeeGridBindingSource = New BindingSource

With AEmployeeBindingSource

.DataSource = AXtraWindoorsDataSet

.DataMember = "Employee"

End With

With EmpIDComboBox

.DataSource = AEmployeeBindingSource

.DisplayMember = "EmpID"

.DataBindings.Add("text", AEmployeeBindingSource, "EmpID", False, DataSourceUpdateMode.Never)

End With

FirstNameTextBox.DataBindings.Add("text", AEmployeeBindingSource, "FirstName")

LastNameTextBox.DataBindings.Add("text", AEmployeeBindingSource, "LastName")

LocationIDTextBox.DataBindings.Add("text", AEmployeeBindingSource, "LocationID")

PositionTextBox.DataBindings.Add("text", AEmployeeBindingSource, "Position")

AddressTextBox.DataBindings.Add("text", AEmployeeBindingSource, "Address")

CItyTextBox.DataBindings.Add("text", AEmployeeBindingSource, "City")

StateTextBox.DataBindings.Add("text", AEmployeeBindingSource, "State")

PhoneTextBox.DataBindings.Add("text", AEmployeeBindingSource, "Phone")

EmailIDTextBox.DataBindings.Add("text", AEmployeeBindingSource, "EmailID")

PasswordTextBox.DataBindings.Add("text", AEmployeeBindingSource, "Password")

OwnerFlagTextBox.DataBindings.Add("text", AEmployeeBindingSource, "OwnerFlag")

FirstLoginTextBox.DataBindings.Add("text", AEmployeeBindingSource, "FirstLogin")

SetFieldsforReadonly()

Catch ex As Exception

MessageBox.Show("Error: " & ex.Message)

End Try

End Sub

Private Sub Add\_SaveButton\_Click(sender As System.Object, e As System.EventArgs) Handles Add\_SaveButton.Click

EmpIDComboBox.SelectedIndex = -1

If Add\_SaveButton.Text = "Add" Then

Add\_SaveButton.Text = "Save"

Delete\_CancelButton.Text = "Cancel"

Edit\_SaveButton.Text = "Edit"

Edit\_SaveButton.Enabled = False

AddFlag = True

EditFlag = False

SetFieldsforEditing()

ElseIf Add\_SaveButton.Text = "Save" Then

Add\_SaveButton.Text = "Add"

Delete\_CancelButton.Text = "Delete"

Edit\_SaveButton.Text = "Edit"

Edit\_SaveButton.Enabled = True

'SetFieldsforReadonly()

If FirstLoginTextBox.Text = "1" Then

Try

Dim SmtpServer As New SmtpClient()

Dim mail As New MailMessage()

SmtpServer.Credentials = New \_

Net.NetworkCredential("xtrawindoors@gmail.com", "GalaxySoft")

SmtpServer.Port = 587

SmtpServer.EnableSsl = True

SmtpServer.Host = "smtp.gmail.com"

mail = New MailMessage()

mail.From = New MailAddress("xtrawindoors@gmail.com")

mail.To.Add(EmailIDTextBox.Text)

mail.Subject = "Initital Password from XTRAWINDOORS"

mail.Body = String.Format("Your new Password for Your Email account {0} is {1} SMTP mail from GMAIL", EmailIDTextBox.Text, PasswordTextBox.Text)

SmtpServer.Send(mail)

MsgBox("New password as been sent your mail")

Catch ex As Exception

MsgBox(ex.ToString)

End Try

End If

Dim ANewEmployeeRow As DataRow = AXtraWindoorsDataSet.Employee.NewRow

Dim ANewLocationRow As DataRow = AXtraWindoorsDataSet.Location.NewRow

'ANewEmployeeRow("EmpID") = Convert.ToInt32(EmpIDComboBox.Text) 'EmployeeIDTextBox.Text

ANewEmployeeRow("LocationID") = LocationIDTextBox.Text

ANewEmployeeRow("FirstName") = FirstNameTextBox.Text

ANewEmployeeRow("LastName") = LastNameTextBox.Text

ANewEmployeeRow("Position") = PositionTextBox.Text

ANewEmployeeRow("Address") = AddressTextBox.Text

ANewEmployeeRow("City") = CItyTextBox.Text

ANewEmployeeRow("State") = StateTextBox.Text

ANewEmployeeRow("Phone") = PhoneTextBox.Text

ANewEmployeeRow("EmailID") = EmailIDTextBox.Text

ANewEmployeeRow("Password") = PasswordTextBox.Text

ANewEmployeeRow("OwnerFlag") = OwnerFlagTextBox.Text

ANewEmployeeRow("FirstLogin") = FirstLoginTextBox.Text

ANewLocationRow("LocationID") = LocationIDTextBox.Text

Try

ALocationTableAdapter.Update(AXtraWindoorsDataSet.Location)

AXtraWindoorsDataSet.Employee.Rows.Add(ANewEmployeeRow)

AEmployeeTableAdapter.Update(AXtraWindoorsDataSet.Employee)

AEmployeeTableAdapter.Fill(AXtraWindoorsDataSet.Employee)

MessageBox.Show("New Employee Record has been Added to Database", "New Employee Record Added", MessageBoxButtons.OK, MessageBoxIcon.Information)

Catch ex As Exception

MessageBox.Show("Add Record Error:" & ex.Message, "Add Record Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

AEmployeeTableAdapter.Fill(AXtraWindoorsDataSet.Employee)

End Try

SetFieldsforReadonly()

End If

End Sub

Private Sub Edit\_SaveButton\_Click(sender As System.Object, e As System.EventArgs) Handles Edit\_SaveButton.Click

If Edit\_SaveButton.Text = "Edit" Then

Add\_SaveButton.Text = "Add"

Add\_SaveButton.Enabled = False

Edit\_SaveButton.Text = "Save"

Delete\_CancelButton.Text = "Cancel"

EditFlag = True

AddFlag = False

SetFieldsforEditing()

ElseIf Edit\_SaveButton.Text = "Save" Then

Edit\_SaveButton.Text = "Edit"

Delete\_CancelButton.Text = "Delete"

Add\_SaveButton.Enabled = True

Try

AEmployeeBindingSource.EndEdit()

AEmployeeTableAdapter.Update(AXtraWindoorsDataSet.Employee)

''Refresh dataset from physical Database

Dim i As Integer = AEmployeeBindingSource.Position

AEmployeeTableAdapter.Fill(AXtraWindoorsDataSet.Employee)

AEmployeeBindingSource.Position = i

MessageBox.Show("The record has been updated to the database successfully!", "Record updated", MessageBoxButtons.OK, MessageBoxIcon.Information)

Catch ex As Exception

MessageBox.Show("Update Record error:" & ex.Message, "Update Record Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

AEmployeeTableAdapter.Fill(AXtraWindoorsDataSet.Employee)

End Try

SetFieldsforReadonly()

End If

End Sub

Private Sub Delete\_CancelButton\_Click(sender As System.Object, e As System.EventArgs) Handles Delete\_CancelButton.Click

If Delete\_CancelButton.Text = "Delete" Then

If MessageBox.Show("Delete current record? Click 'Yes' to delete, 'No' to cancel.", "Delete Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question) = Windows.Forms.DialogResult.Yes Then

If AEmployeeBindingSource.Count > 0 Then

Try

AEmployeeBindingSource.RemoveCurrent()

AEmployeeTableAdapter.Update(AXtraWindoorsDataSet.Employee)

AEmployeeTableAdapter.Fill(AXtraWindoorsDataSet.Employee)

MessageBox.Show("Record is deleted successfully", "Deletion", MessageBoxButtons.OK, MessageBoxIcon.Information)

Add\_SaveButton.Text = "Add"

Edit\_SaveButton.Text = "Edit"

Add\_SaveButton.Enabled = True

Edit\_SaveButton.Enabled = True

SetFieldsforReadonly()

Catch ex As Exception

MessageBox.Show("Delete record error: " & ex.Message, "Delete Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

AEmployeeTableAdapter.Fill(AXtraWindoorsDataSet.Employee)

End Try

End If

End If

ElseIf Delete\_CancelButton.Text = "Cancel" Then

If MessageBox.Show("Unsaved Data will be lost. Do you want to Cancel?", "Cancel?", MessageBoxButtons.OKCancel, MessageBoxIcon.Question) = Windows.Forms.DialogResult.OK Then

Add\_SaveButton.Text = "Add"

Edit\_SaveButton.Text = "Edit"

Add\_SaveButton.Enabled = True

Edit\_SaveButton.Enabled = True

Delete\_CancelButton.Text = "Delete"

SetFieldsforReadonly()

Dim i As Integer = AEmployeeBindingSource.Position

AEmployeeTableAdapter.Fill(AXtraWindoorsDataSet.Employee)

AEmployeeBindingSource.Position = i

End If

End If

End Sub

Private Sub SplitContainer1\_Panel1\_Paint(sender As System.Object, e As System.Windows.Forms.PaintEventArgs) Handles SplitContainer1.Panel1.Paint

End Sub

Sub SetFieldsforEditing()

If AddFlag = True Then

EmpIDComboBox.DropDownStyle = ComboBoxStyle.Simple

EmpIDComboBox.Enabled = False

EmpIDComboBox.Text = ""

LocationIDTextBox.Text = ""

FirstNameTextBox.Text = ""

LastNameTextBox.Text = ""

PositionTextBox.Text = ""

AddressTextBox.Text = ""

CItyTextBox.Text = ""

StateTextBox.Text = ""

PhoneTextBox.Text = ""

EmailIDTextBox.Text = ""

PasswordTextBox.Text = ""

OwnerFlagTextBox.Text = ""

FirstLoginTextBox.Text = ""

Else

End If

FirstNameTextBox.Enabled = True

LocationIDTextBox.Enabled = True

LastNameTextBox.Enabled = True

PositionTextBox.Enabled = True

AddressTextBox.Enabled = True

CItyTextBox.Enabled = True

StateTextBox.Enabled = True

PhoneTextBox.Enabled = True

EmailIDTextBox.Enabled = True

PasswordTextBox.Enabled = True

OwnerFlagTextBox.Enabled = True

FirstLoginTextBox.Enabled = True

End Sub

Sub SetFieldsforReadonly()

EmpIDComboBox.DropDownStyle = ComboBoxStyle.DropDownList

EmpIDComboBox.Enabled = True

LocationIDTextBox.Enabled = False

FirstNameTextBox.Enabled = False

LastNameTextBox.Enabled = False

PositionTextBox.Enabled = False

AddressTextBox.Enabled = False

CItyTextBox.Enabled = False

StateTextBox.Enabled = False

PhoneTextBox.Enabled = False

EmailIDTextBox.Enabled = False

PasswordTextBox.Enabled = False

OwnerFlagTextBox.Enabled = False

FirstLoginTextBox.Enabled = False

End Sub

Private Sub EMPIDSearchButton\_Click(sender As System.Object, e As System.EventArgs) Handles EMPIDSearchButton.Click

Try

Me.EmployeeTableAdapter.FillByEmpID(Me.XtraWindoorsDataSet.Employee, EmployeeIDSearchTextBox.Text)

Catch ex As Exception

MessageBox.Show(" There is no record exist !!")

End Try

End Sub

Private Sub EmpPhoneSearchButton\_Click(sender As System.Object, e As System.EventArgs) Handles EmpPhoneSearchButton.Click

Try

Me.EmployeeTableAdapter.FillByPhone(Me.XtraWindoorsDataSet.Employee, PhoneSearchTextBox.Text)

Catch ex As Exception

MessageBox.Show(" There is no record exist !!")

End Try

End Sub

Private Sub ClearSearchButton\_Click(sender As System.Object, e As System.EventArgs) Handles ClearSearchButton.Click

PhoneSearchTextBox.Clear()

EmployeeIDSearchTextBox.Clear()

EmployeeDataGridView.Refresh()

End Sub

Private Sub HomeButton\_Click(sender As System.Object, e As System.EventArgs) Handles HomeButton.Click

Me.Hide()

HomePageScreen.Show()

End Sub

Private Sub BookingForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

HomePageScreen.Show()

End Sub

Private Sub StateTextBox\_Validating(sender As Object, e As System.ComponentModel.CancelEventArgs) Handles StateTextBox.Validating

Dim StateInput As String

StateInput = StateTextBox.Text

Dim lengthValid As Boolean = StateInput.Length.Equals(2)

If Not lengthValid Then

MainErrorProvider.SetError(StateTextBox, "Invalid State, must be 2 Character")

End If

End Sub

Private Sub PhoneTextBox\_Validating(sender As Object, e As System.ComponentModel.CancelEventArgs) Handles StateTextBox.Validating

Dim phoCode As String

phoCode = PhoneTextBox.Text

Dim lengthValid As Boolean = phoCode.Length.Equals(5)

If Not lengthValid Then

MainErrorProvider.SetError(StateTextBox, "Invalid Phone, must be 10 digits")

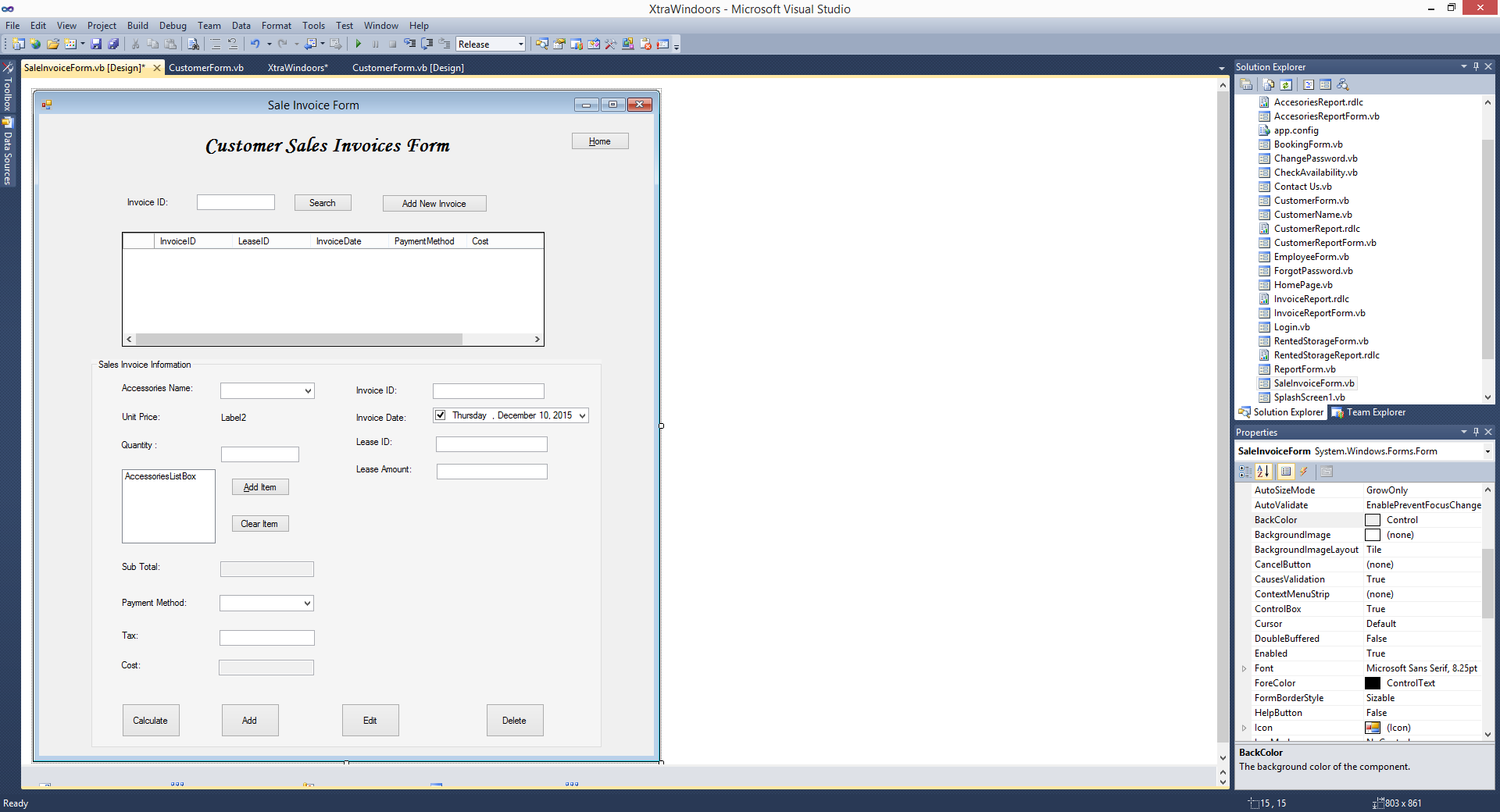
End If

End Sub

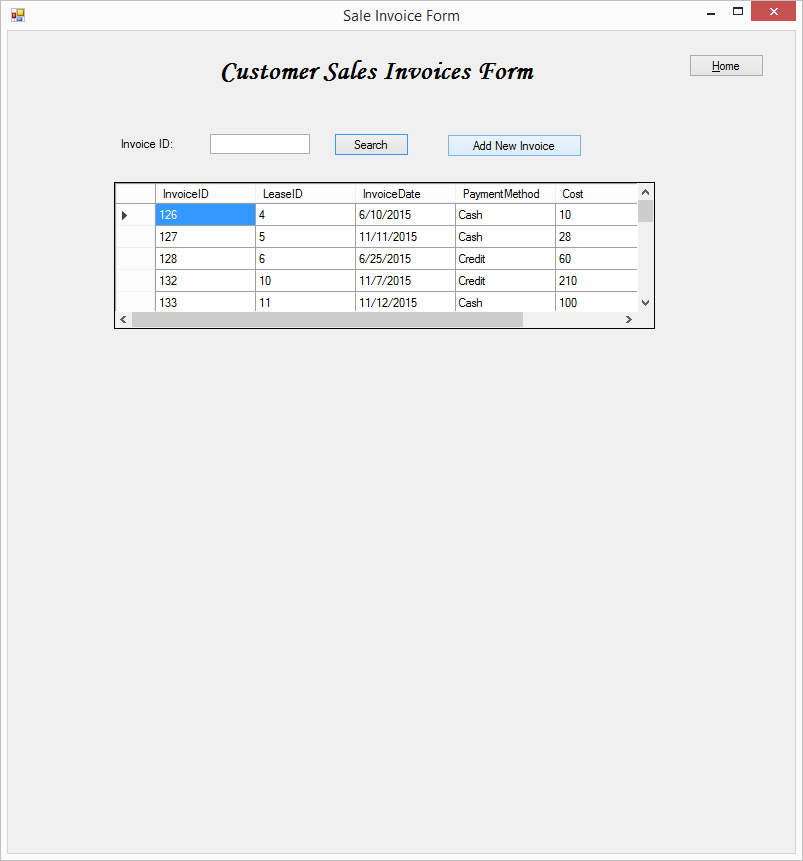
End Class

**Sales Invoice**

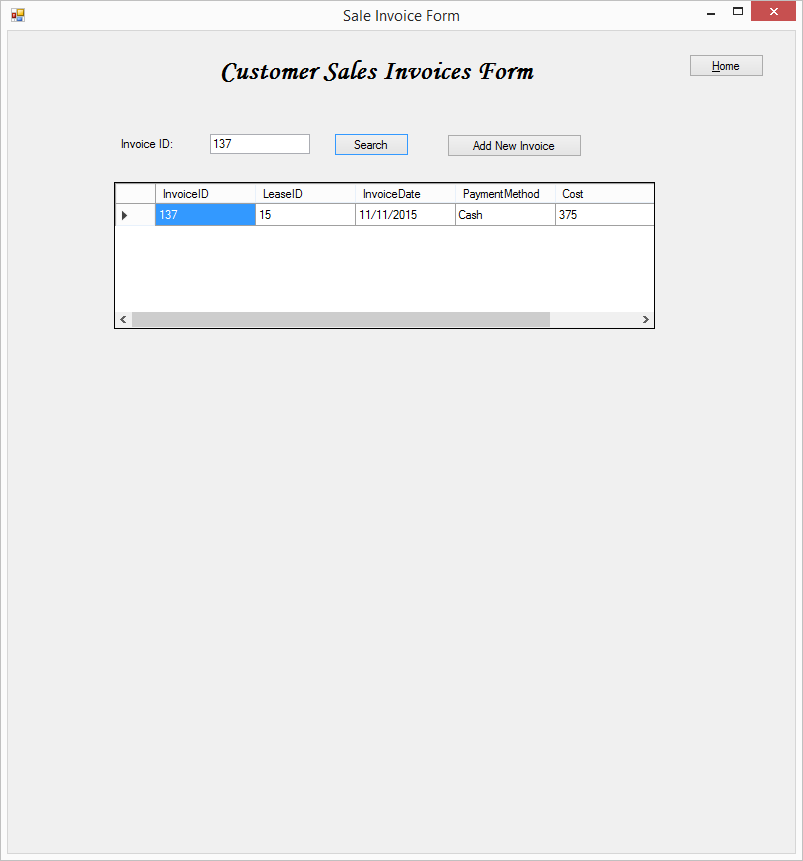
**Figure1.0 Design view**



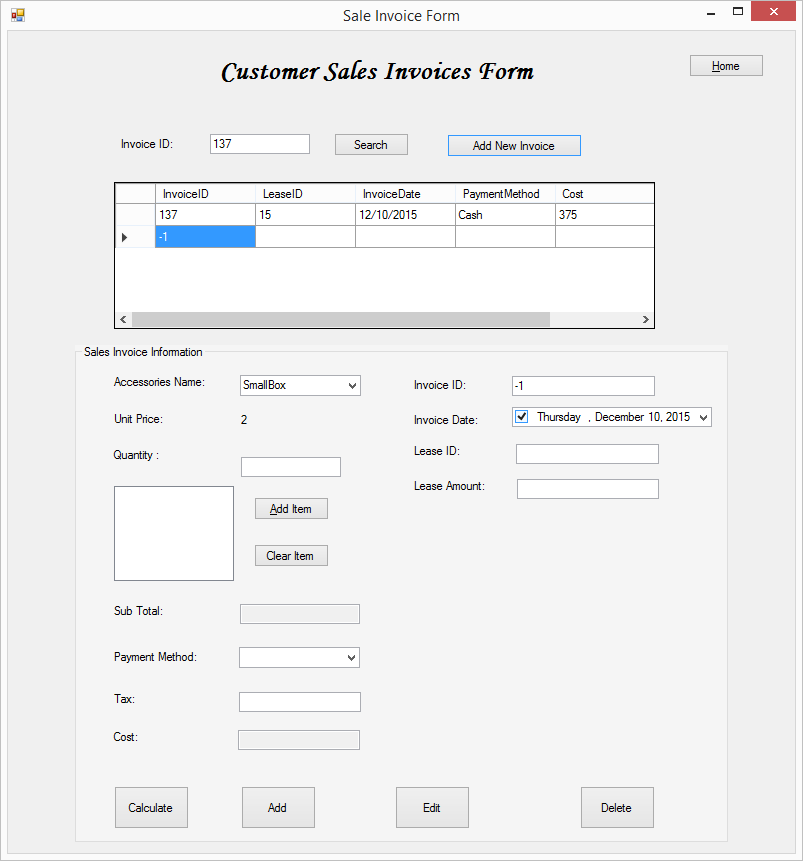
**Figure1.1 Excusive view**



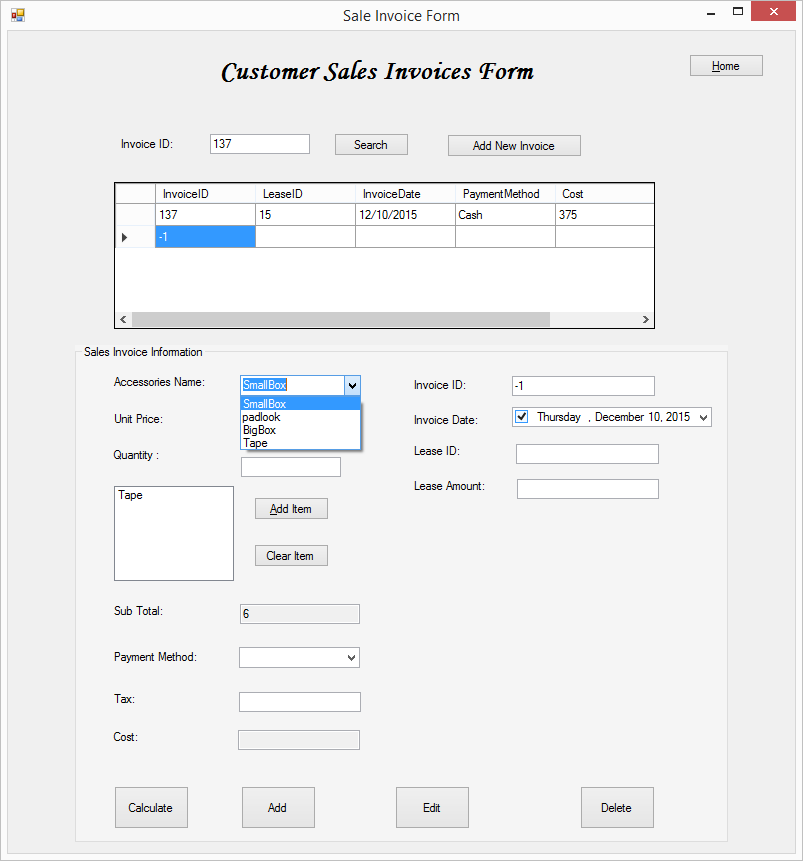
**Figure1.2. looking for the sales invoice “invoice ID 137” by Search on the sales invoice ID, enter the information 137 , and clicked on search button.**

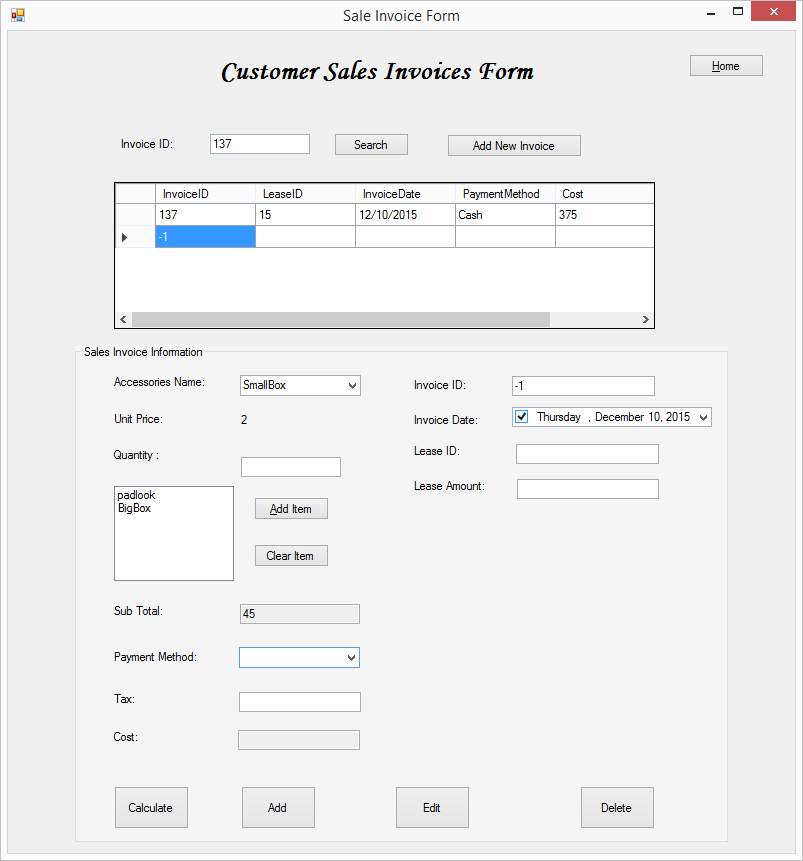


**Figure2.0. Add a new invoice by click on the “Add new invoice “button.**

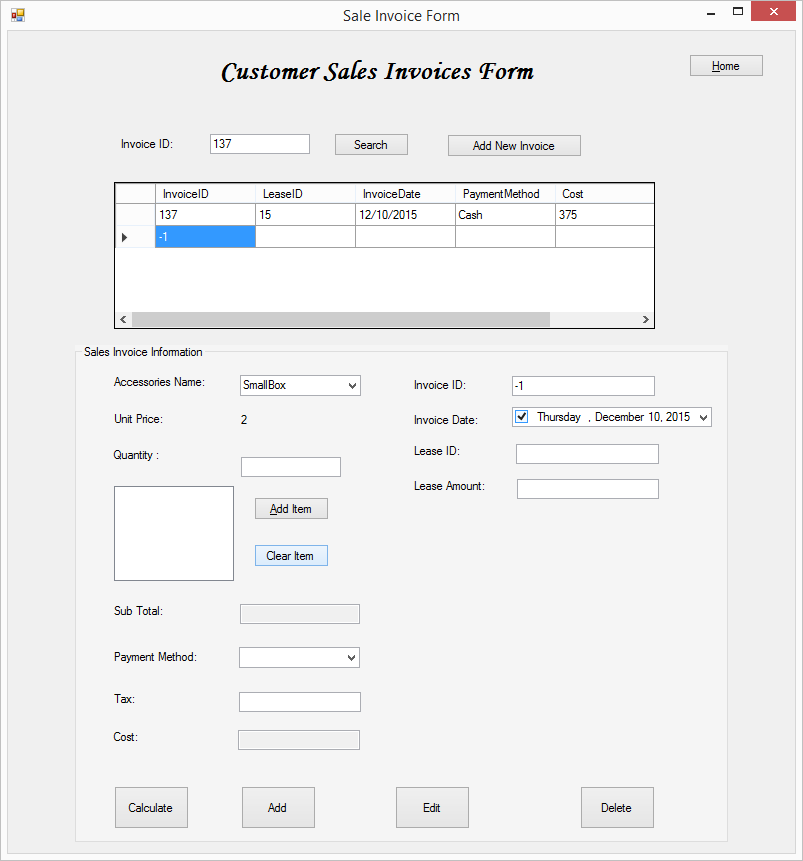


**Figure2.1. Add item button clicked, add 2 pad back, and 5 big box.**

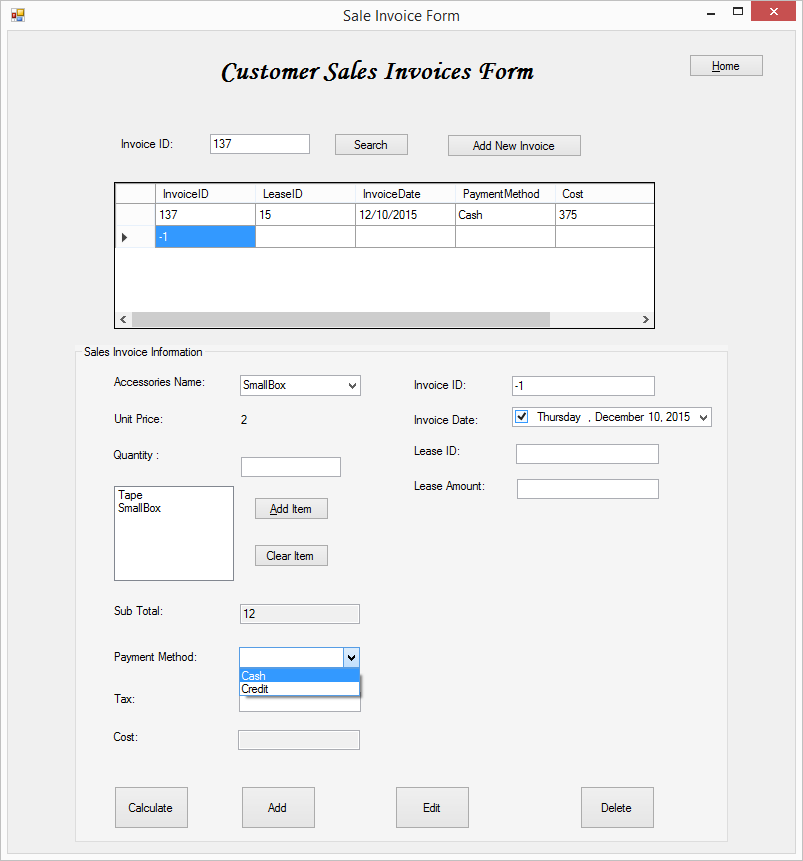




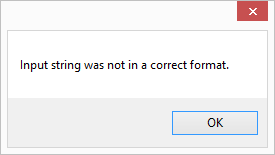
**Figure2.2. Clear item button clicked, it cleared 2 pad back, and 5 big box.**

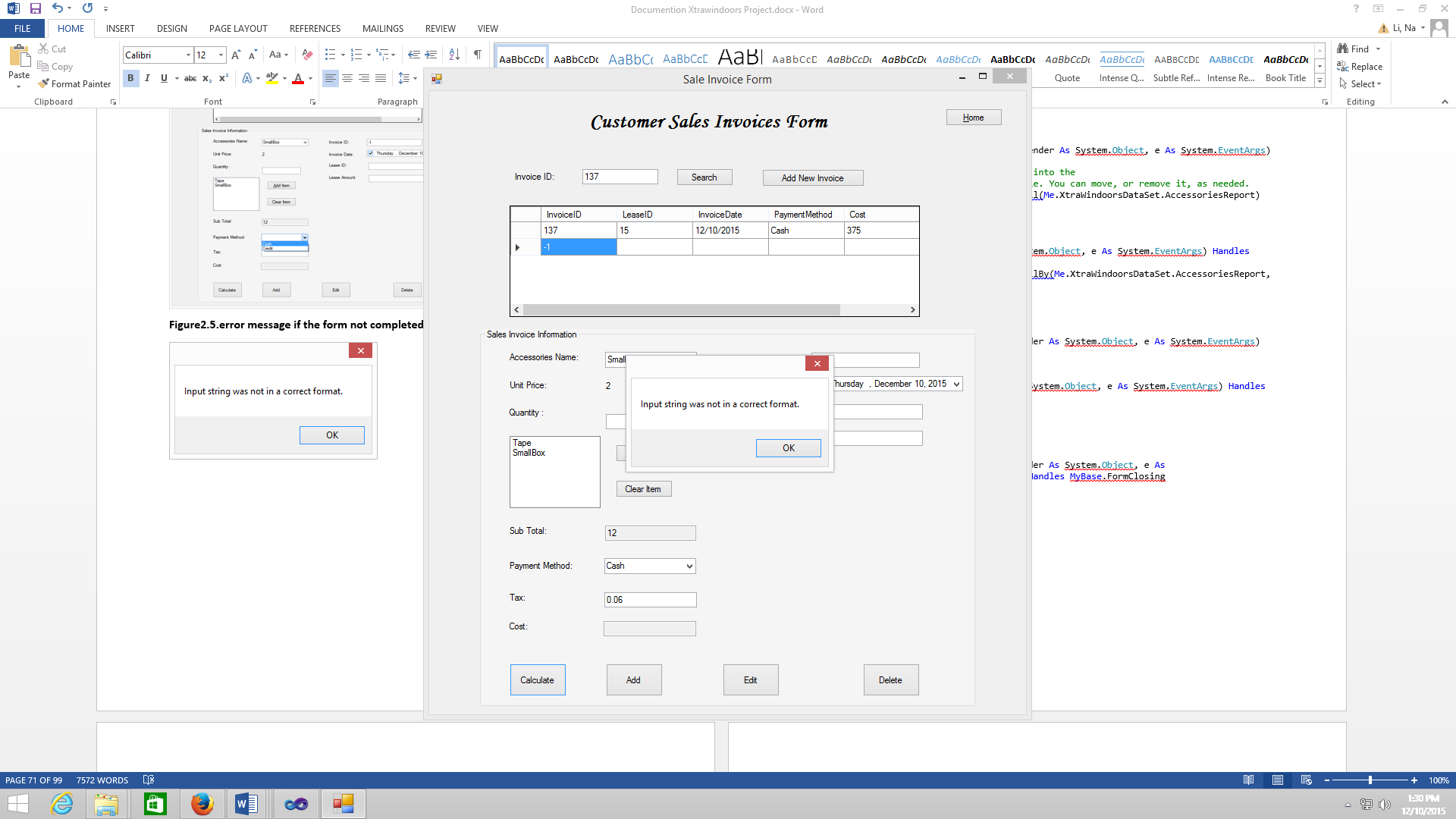


**Figure2.4.Payment selected from the drop down**

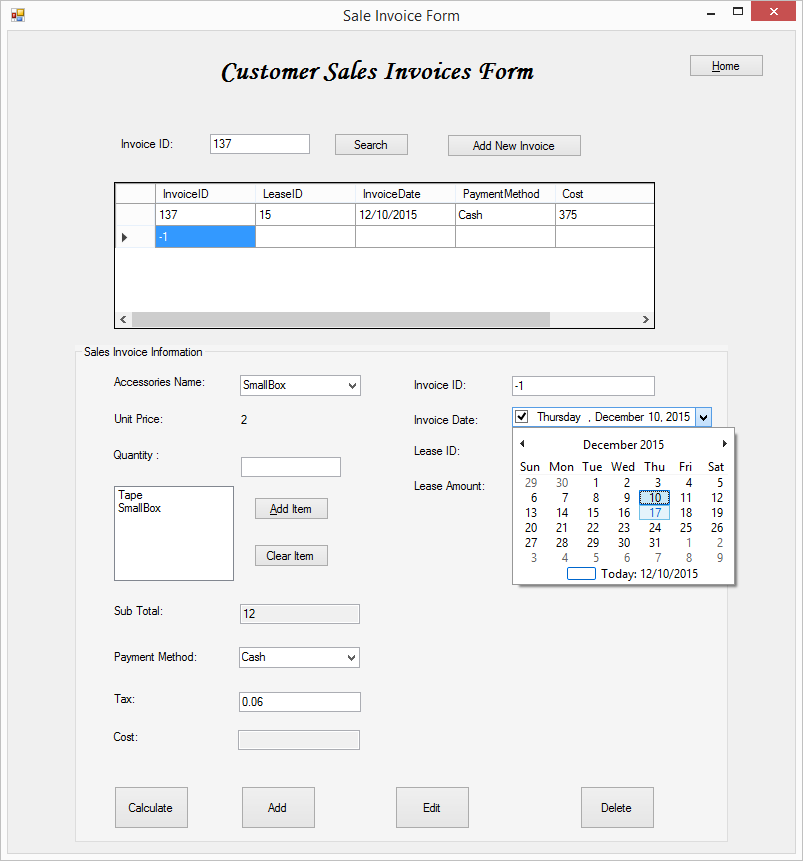


**Figure2.5.error message if the form not completed in a correct way.**

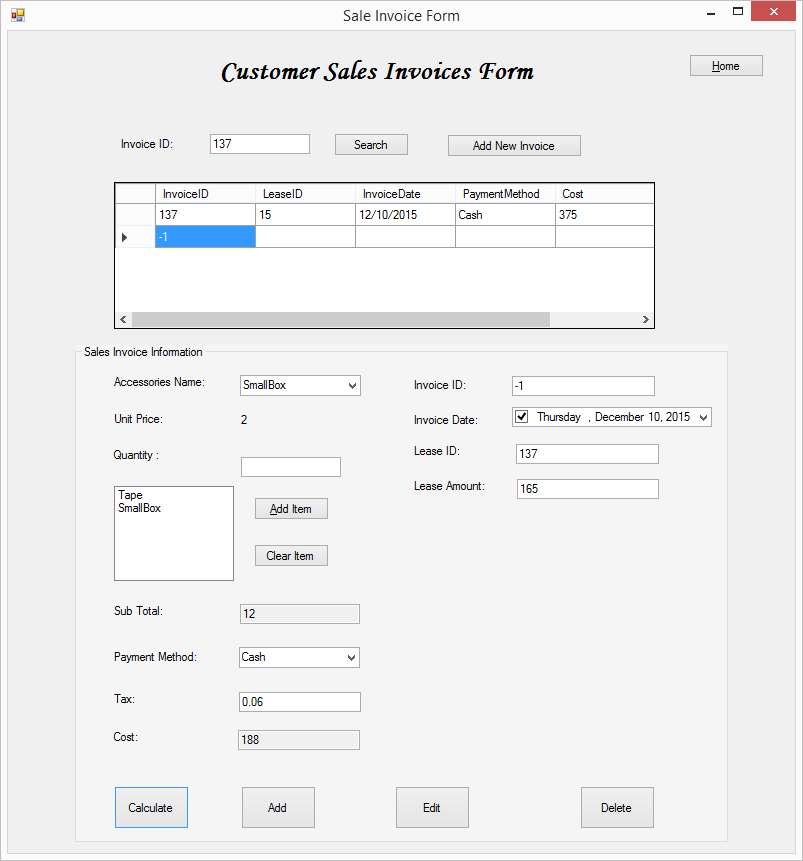




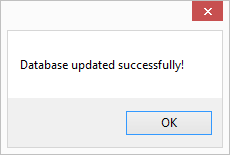
**Figure2.6. Date has selected.**



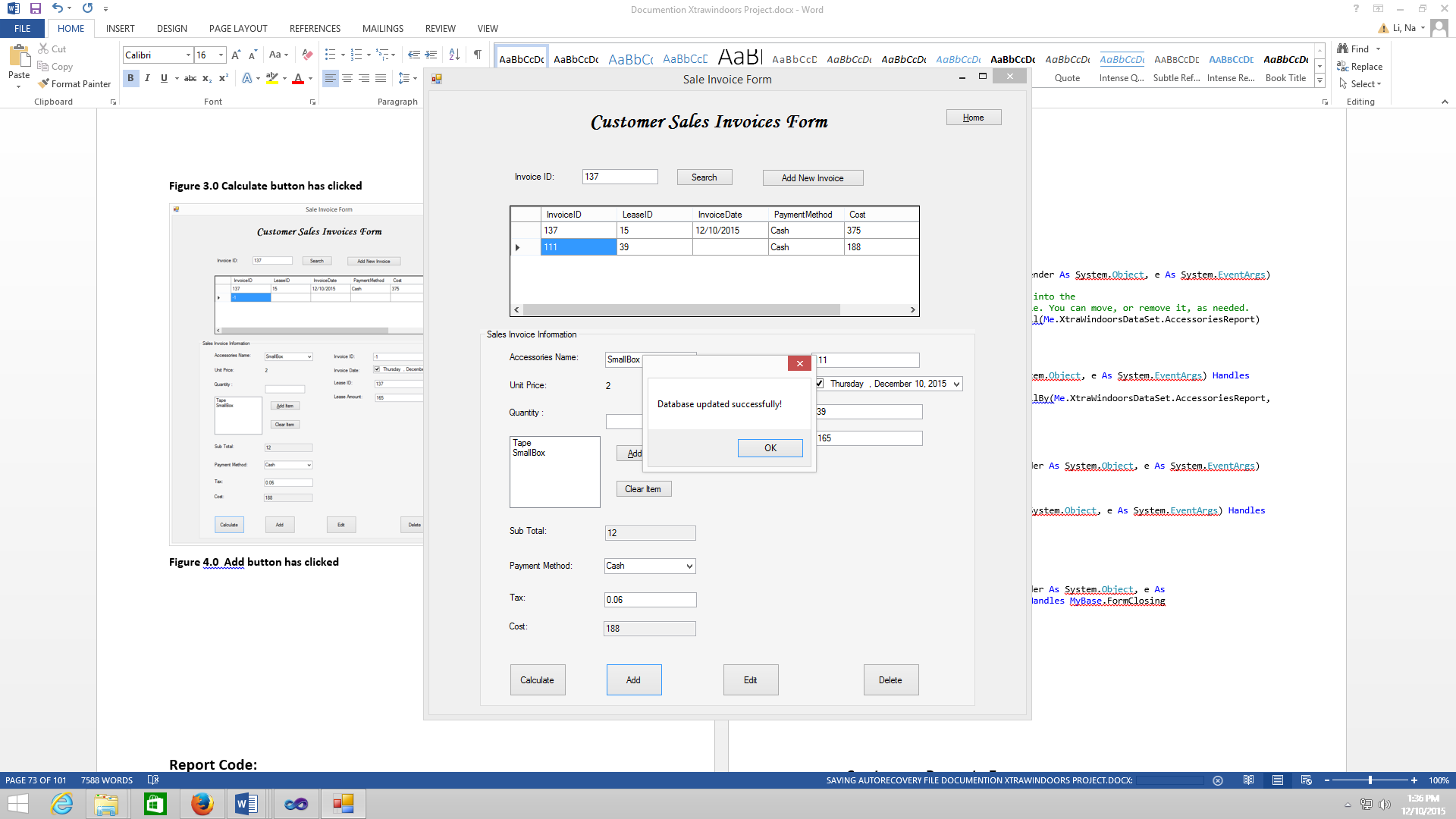
**Figure 3.0 Calculate button has clicked**



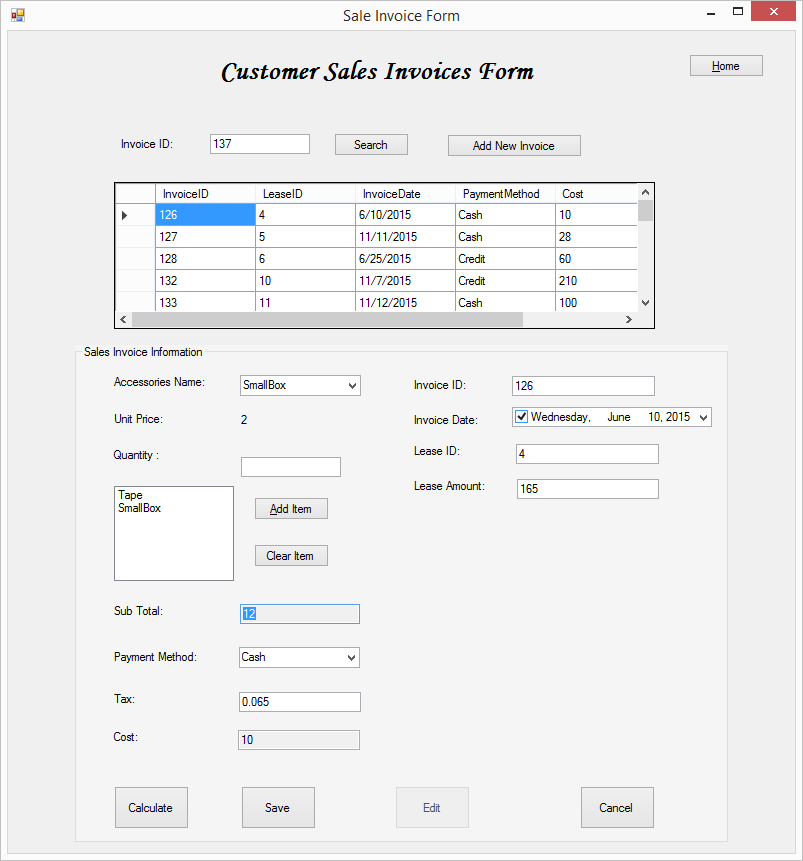
**Figure 4.0 Add button has clicked**



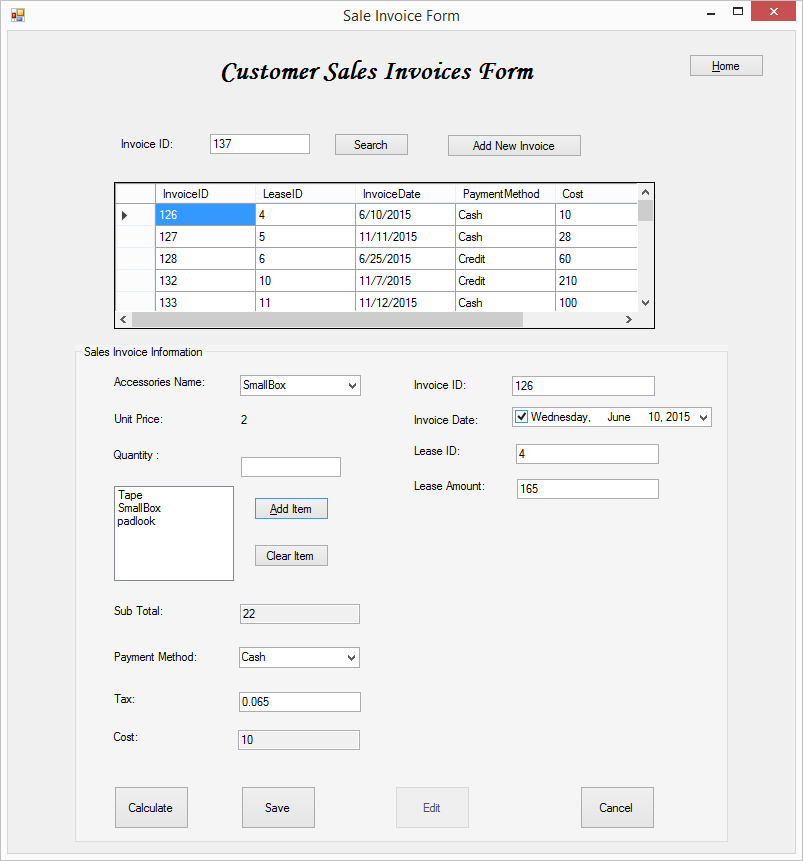
**Figure 4.1**



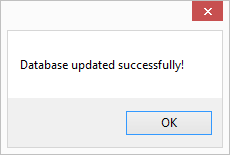
**Figure 5.0 Edit button has clicked, we can make a change here. Add one more pad lock.**



**Figure 5.1**



**Figure 5.2**



**Sales invoices coding:**

'Program Name: XtraWindoors Rental System

'Programmer: Group 7

'Date: 12/10/2015

'Class: Sale Invoice Form

Public Class SaleInvoiceForm

Private SubTotaDecimal, TaxDecimal, TotalAmountInteger, SubAmountLeaseDecimal As Decimal

Private AddingBoolean As Boolean = False

Private EditingBoolean As Boolean = False

Private Sub InvoiceBindingNavigatorSaveItem\_Click(sender As System.Object, e As System.EventArgs)

Me.Validate()

Me.InvoiceBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.XtraWindoorsDataSet)

End Sub

Private Sub SaleInvoiceForm\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

'TODO: This line of code loads data into the 'XtraWindoorsDataSet.Invoice' table. You can move, or remove it, as needed.

Me.InvoiceTableAdapter.Fill(Me.XtraWindoorsDataSet.Invoice)

'TODO: This line of code loads data into the 'XtraWindoorsDataSet.Accessories' table. You can move, or remove it, as needed.

Me.AccessoriesTableAdapter.Fill(Me.XtraWindoorsDataSet.Accessories)

End Sub

Private Sub NewButton\_Click(sender As System.Object, e As System.EventArgs) Handles NewButton.Click

GroupBox1.Visible = True

InvoiceDateDateTimePicker.Value = Today

SetControlsReadOnly(False)

Me.InvoiceBindingSource.EndEdit()

Me.InvoiceBindingSource.AddNew()

End Sub

Private Sub CalculateButton\_Click(sender As System.Object, e As System.EventArgs) Handles CalculateButton.Click

Try

SubAmountLeaseDecimal = Decimal.Parse(SubAmountTextBox.Text)

TaxDecimal = Decimal.Parse(TaxTextBox.Text)

TotalAmountInteger = Convert.ToInt32((SubTotaDecimal + SubAmountLeaseDecimal) \* (1 + TaxDecimal))

CostTextBox.Text = TotalAmountInteger.ToString

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub AddSaveButton\_Click(sender As System.Object, e As System.EventArgs) Handles AddSaveButton.Click

Try

Me.Validate()

Me.InvoiceBindingSource.EndEdit()

'AccessoriesBindingSource.EndEdit()

Me.InvoiceTableAdapter.Update(Me.XtraWindoorsDataSet.Invoice)

' Me.TableAdapterManager.UpdateAll(Me.XtraWindoorsDataSet)

MessageBox.Show("Database updated successfully!")

InvoiceTableAdapter.Fill(XtraWindoorsDataSet.Invoice)

SetValues(True)

InvoiceDateDateTimePicker.Enabled = False

Catch ex As Exception

MessageBox.Show("Unable to complete the add/save operation: " & ex.Message, "Add/Save", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Private Sub SetValues(Switch As Boolean)

InvoiceIDTextBox.ReadOnly = Switch

LeaseIDTextBox.ReadOnly = Switch

SubAmountTextBox.ReadOnly = Switch

' CostTextBox.ReadOnly = Switch

TaxTextBox.ReadOnly = Switch

QuantityTextBox.ReadOnly = Switch

PaymentMethodComboBox.Enabled = Not (Switch)

AccessoriesNameComboBox.Enabled = Not (Switch)

AccessoriesListBox.Enabled = Not (Switch)

End Sub

Private Sub EditButton\_Click(sender As System.Object, e As System.EventArgs) Handles EditButton.Click

'Allow editing the current record.

EditingBoolean = True

SetControlsReadOnly(False)

SetButtonForEdit()

' SetValues(True)

End Sub

Private Sub SetButtonForEdit()

AddSaveButton.Text = "Save"

DeleteCancelButton.Text = "Cancel"

EditButton.Enabled = False

End Sub

Private Sub ResetButtonafterEdit()

AddSaveButton.Text = " Add"

DeleteCancelButton.Text = "Delete"

EditButton.Enabled = True

End Sub

Private Sub DeleteCancelButton\_Click(sender As System.Object, e As System.EventArgs) Handles DeleteCancelButton.Click

Dim DeleteDialogResult As DialogResult

Try

If DeleteCancelButton.Text = "Delete" Then

'delete records

DeleteDialogResult = MessageBox.Show("Delete this record?", "Confirm Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If DeleteDialogResult = Windows.Forms.DialogResult.Yes Then

InvoiceBindingSource.RemoveCurrent()

InvoiceTableAdapter.Update(XtraWindoorsDataSet.Invoice)

End If

Else

'Cancel button is clicked

InvoiceBindingSource.CancelEdit()

AddingBoolean = False

EditingBoolean = False

SetControlsReadOnly(True)

ResetButtonafterEdit()

End If

Catch ex As Exception

MessageBox.Show("unable to complete the delete/cancel operation:" & ex.Message, "Delete/Cancel", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Private Sub AddItemButton\_Click(sender As System.Object, e As System.EventArgs) Handles AddItemButton.Click

AccessoriesListBox.Items.Add(AccessoriesNameComboBox.Text)

Dim UnitpriceDecimal, PriceDecimal As Decimal

Dim QuantityInteger As Integer

Try

UnitpriceDecimal = Decimal.Parse(UnitPriceLabel3.Text)

QuantityInteger = Integer.Parse(QuantityTextBox.Text)

PriceDecimal = UnitpriceDecimal \* QuantityInteger

SubTotaDecimal += PriceDecimal

PriceTextBox.Text = SubTotaDecimal.ToString

AccessoriesNameComboBox.SelectedIndex = -1

QuantityTextBox.Clear()

UnitPriceLabel3.Text = ""

Me.AccessoriesTableAdapter.Fill(Me.XtraWindoorsDataSet.Accessories)

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub SetControlsReadOnly(ByVal ValueBoolean As Boolean)

InvoiceIDTextBox.ReadOnly = ValueBoolean

LeaseIDTextBox.ReadOnly = ValueBoolean

QuantityTextBox.ReadOnly = ValueBoolean

PaymentMethodComboBox.Enabled = True

InvoiceDateDateTimePicker.Enabled = True

SubAmountTextBox.ReadOnly = ValueBoolean

TaxTextBox.ReadOnly = ValueBoolean

AccessoriesListBox.Enabled = True

AccessoriesNameComboBox.Enabled = True

End Sub

Private Sub DeleteItemButton\_Click(sender As System.Object, e As System.EventArgs) Handles DeleteItemButton.Click

SubTotaDecimal = 0

PriceTextBox.Clear()

AccessoriesListBox.Items.Clear()

End Sub

Private Sub SearchButton\_Click(sender As System.Object, e As System.EventArgs) Handles SearchButton.Click

Try

InvoiceDataGridView.DataSource = InvoiceBindingSource

Me.InvoiceTableAdapter.FillByInvoiceID(Me.XtraWindoorsDataSet.Invoice, InvoiceTextBox.Text)

SetControlsReadOnly(True)

Catch ex As System.Exception

MessageBox.Show(" There is no record exist !!")

End Try

End Sub

Private Sub HomeButton\_Click(sender As System.Object, e As System.EventArgs) Handles HomeButton.Click

Me.Hide()

HomePageScreen.Show()

End Sub

Private Sub BookingForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

HomePageScreen.Show()

End Sub

End Class

**Report Code:**

**1.Accessories Report Form**

'Program Name: XtraWindoors Rental System

'Programmer: Group 7

'Date: 12/10/2015

'Class: Accessories Report Form

Public Class AccesoriesReportForm

Private Sub AccesoriesReportForm\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

'TODO: This line of code loads data into the 'XtraWindoorsDataSet.AccessoriesReport' table. You can move, or remove it, as needed.

Me.AccessoriesReportTableAdapter.Fill(Me.XtraWindoorsDataSet.AccessoriesReport)

Me.ReportViewer1.RefreshReport()

End Sub

Private Sub Button1\_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click

Me.AccessoriesReportTableAdapter.FillBy(Me.XtraWindoorsDataSet.AccessoriesReport, DateTimePicker1.Text)

Me.ReportViewer1.RefreshReport()

End Sub

Private Sub DateTextBox\_TextChanged(sender As System.Object, e As System.EventArgs)

End Sub

Private Sub HomeButton\_Click(sender As System.Object, e As System.EventArgs) Handles HomeButton.Click

Me.Hide()

HomePageScreen.Show()

End Sub

Private Sub BookingForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

HomePageScreen.Show()

End Sub

End Class

**2. Customer Report Form**

'Program Name: XtraWindoors Rental System

'Programmer: Group 7

'Date: 12/10/2015

'Class: Customer Report Form

Public Class CustomerReportForm

Private Sub CustomerReportForm\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

'TODO: This line of code loads data into the 'XtraWindoorsDataSet.CustomerReport' table. You can move, or remove it, as needed.

Me.CustomerReportTableAdapter.Fill(Me.XtraWindoorsDataSet.CustomerReport)

Me.ReportViewer1.RefreshReport()

End Sub

Private Sub HomeButton\_Click(sender As System.Object, e As System.EventArgs) Handles HomeButton.Click

Me.Hide()

HomePageScreen.Show()

End Sub

Private Sub BookingForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

HomePageScreen.Show()

End Sub

End Class

**3. Rented Storage Report Form**

'Program Name: XtraWindoors Rental System

'Programmer: Group 7

'Date: 12/10/2015

'Class: Rented Storage Report Form

Public Class RentedStorageForm

Private Sub RentedStorageForm\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

'TODO: This line of code loads data into the 'XtraWindoorsDataSet.RentedStorageReport' table. You can move, or remove it, as needed.

Me.RentedStorageReportTableAdapter.Fill(Me.XtraWindoorsDataSet.RentedStorageReport)

Me.ReportViewer1.RefreshReport()

End Sub

Private Sub HomeButton\_Click(sender As System.Object, e As System.EventArgs) Handles HomeButton.Click

Me.Hide()

HomePageScreen.Show()

End Sub

Private Sub BookingForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

HomePageScreen.Show()

End Sub

End Class

**4. Customer Name for Invoice Report**

'Program Name: XtraWindoors Rental System

'Programmer: Group 7

'Date: 12/10/2015

'Class: Customer last Name or Lease ID for Invoice Report

Public Class CustomerName

Private Sub Button1\_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click

If LeaseIDComboBox.SelectedIndex = -1 And LastNameComboBox.SelectedIndex = -1 Then

MessageBox.Show("Please select customer last name or Lease ID", "Select Customer", MessageBoxButtons.OK, MessageBoxIcon.Information)

Else

InvoiceReportForm.InvoiceReportTableAdapter.FillBy(InvoiceReportForm.XtraWindoorsDataSet.InvoiceReport, LastNameComboBox.Text, LeaseIDComboBox.Text)

InvoiceReportForm.ReportViewer1.RefreshReport()

InvoiceReportForm.Show()

LastNameComboBox.SelectedIndex = -1

LeaseIDComboBox.SelectedIndex = -1

Me.Hide()

End If

End Sub

Private Sub LastNameTextBox\_TextChanged(sender As System.Object, e As System.EventArgs)

End Sub

Private Sub CustomerName\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

'TODO: This line of code loads data into the 'XtraWindoorsDataSet.InvoiceReport' table. You can move, or remove it, as needed.

Me.InvoiceReportTableAdapter.Fill(Me.XtraWindoorsDataSet.InvoiceReport)

LastNameComboBox.SelectedIndex = -1

LeaseIDComboBox.SelectedIndex = -1

End Sub

Private Sub HomeButton\_Click(sender As System.Object, e As System.EventArgs) Handles HomeButton.Click

Me.Hide()

HomePageScreen.Show()

End Sub

Private Sub BookingForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

HomePageScreen.Show()

End Sub

End Class

**5. Invoice Report Form**

'Program Name: XtraWindoors Rental System

'Programmer: Group 7

'Date: 12/10/2015

'Class: Invoice Report Form

Public Class InvoiceReportForm

Private Sub InvoiceReportForm\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

'TODO: This line of code loads data into the 'XtraWindoorsDataSet.InvoiceReport' table. You can move, or remove it, as needed.

Me.InvoiceReportTableAdapter.Fill(Me.XtraWindoorsDataSet.InvoiceReport)

Me.ReportViewer1.RefreshReport()

End Sub

Private Sub HomeButton\_Click(sender As System.Object, e As System.EventArgs) Handles HomeButton.Click

Me.Hide()

HomePageScreen.Show()

End Sub

Private Sub BookingForm\_FormClosing(sender As System.Object, e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Me.Hide()

HomePageScreen.Show()

End Sub

End Class

* **SQL Statement**

**Accessories Report:**

SELECT Accessories.AccessoriesID, Accessories.AccessoriesName,

Accessories.QuantityInStock, SUM(AccessoriesSale.Quantity) AS QuantitySale,

AccessoriesSale.SaleDateTime

FROM (Accessories INNER JOIN

AccessoriesSale ON Accessories.AccessoriesID = AccessoriesSale.AccessoriesID)

GROUP BY Accessories.AccessoriesID, Accessories.AccessoriesName,

Accessories.QuantityInStock, AccessoriesSale.Quantity, AccessoriesSale.SaleDateTime

HAVING (AccessoriesSale.SaleDateTime = ?)

**Invoice Report:**

SELECT Customer.FirstName, Customer.LastName, Invoice.InvoiceID, Invoice.InvoiceDate,

Invoice.Cost, Invoice.Tax, Lease.LeaseID, Lease.CustomerID, Lease.StorageID,

Lease.CheckIn, Lease.CheckOut, StorageType.TypeSize

FROM ((((Customer INNER JOIN

Lease ON Customer.CustomerID = Lease.CustomerID) INNER JOIN

Invoice ON Lease.LeaseID = Invoice.LeaseID) INNER JOIN

Storage ON Lease.StorageID = Storage.StorageID) INNER JOIN

StorageType ON Storage.StorageTypeID = StorageType.StorageTypeID)

WHERE (Customer.LastName = ?) AND (Lease.LeaseID = ?)

**Rented Storage Report:**

SELECT Location.LocationID, StorageType.TypeSize,

COUNT(Storage.StorageAvaiablity) AS RentedStorage

FROM ((Location INNER JOIN

Storage ON Location.LocationID = Storage.LocationID) INNER JOIN

StorageType ON Storage.StorageTypeID = StorageType.StorageTypeID)

WHERE (Storage.StorageAvaiablity = False)

GROUP BY Storage.StorageAvaiablity, StorageType.TypeSize, Location.LocationID

**Customer Report:**

SELECT Customer.CustomerID, Customer.FirstName, Customer.LastName,

Customer.Email, Lease.CheckIn, Lease.CheckOut, Lease.LeaseActive,

Lease.LeaseCancelled, Invoice.PaymentMethod, Invoice.Cost

FROM ((Customer INNER JOIN

Lease ON Customer.CustomerID = Lease.CustomerID) INNER JOIN

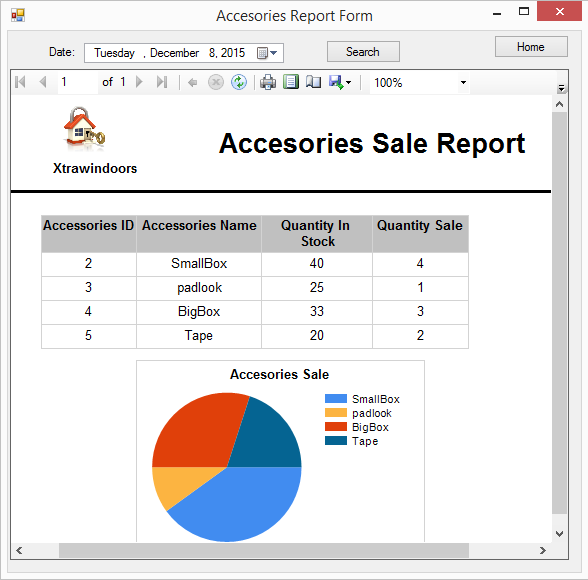
Invoice ON Lease.LeaseID = Invoice.LeaseID)

* **Execute:**

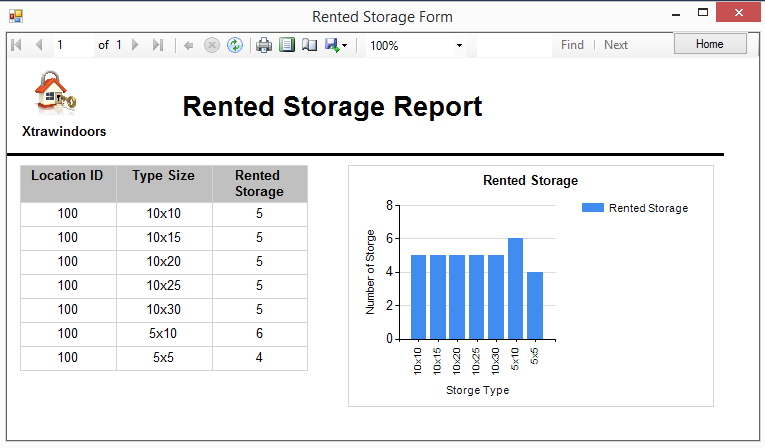
To help our client to track his business, we create 4 report Accessories, Rented Storage, Invoice, Customer Report.



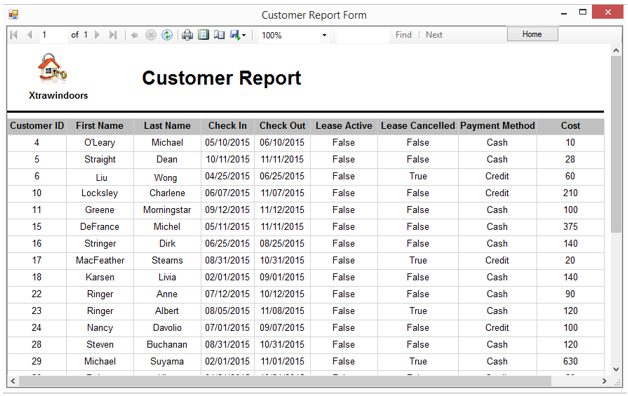
First report is accessories report in this report we display how many unit he sales for each accessories and how many he has in inventory. Also he can select particular day to display the data in this day. We add also pie chart to make the report more useful.



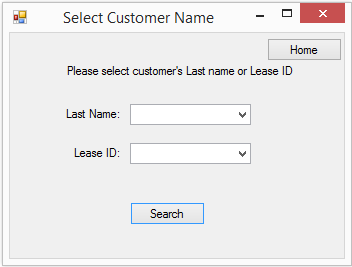
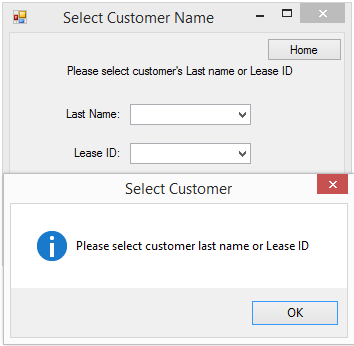
The second report is rented Storage. In this report display the number of storage that rented in report day. We group the data by location and storage type. Also we display the data in chart.

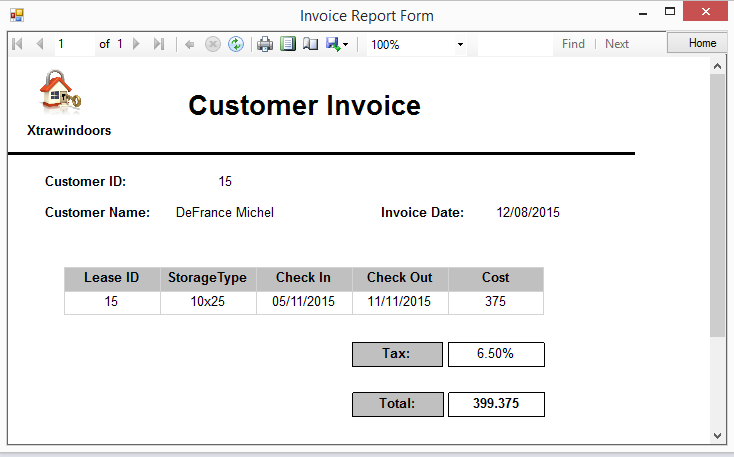


The third report is customer report. In this report we display details about customer lease and payment.



The fourth report is invoice. In case if the customer need invoice paper, we design is report to display lease detail and the total cost after tax. The first screen ask to select customer last name or lease ID to display the invoice. If the user does not select any thang and click search bottom; the user will get message error.





**Section IV**

**Project Journals**

**Minutes of Meeting 1**

Date: 09/08/2015

**Objective of the meeting:**

Introducing and getting to know each member share with contact information. Each of members brainstorms the topic idea and share with each other.

**Members present:**

Anoop Madipalli

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

**Meeting conclusion:**

Discussion about member’s background and group Name with “GalaxySoft ” and group project with storage room idea. And, Na LI will be the secretary and in charge of this project journal.

**Meeting Notification:**

Meeting holds on Friday

Time: 11:00 am -1:00 pm

Location: Library

**Minutes of Meeting 2**

Date: 09/11/2015

**Objective of the meeting:**

Discussion about the business scenario with each member, and created the interview questions form.

**Members present:**

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

Three of us made for the Interview with AAA lockup Company. And, important notes and voice record had made. We also, scheduled the time for the next meeting.

**Meeting Notification:**

Meeting holds on Friday

Time: 12:00 am -2:00 pm

Location: Library

**Minutes of Meeting 3**

Date: 09/18/2015

**Objective of the meeting:**

Discussion about the interview with AAA lockup Company, and shared with rest of members. The dead line of the proposal and the presentation discussed as a group.

**Members present:**

Anoop Madipalli

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

Project proposal has been divided with each member. (E.g Lina;background , Turkey: problem description, Ravi and Anoop system description,Vashnavi : feasibility . Everyone will present with his or her own part. Project Name: XTRA- Windoors.

**Meeting Notification:**

Meeting holds on Friday

Time: 12:00 am -2:00 pm

Location: Library

**Minutes of Meeting 4**

Date: 09/25/2015

**Objective of the meeting:**

Project two, has been discussed in this meeting. Each member has to contribute an idea of what we plan to do in this project as final decision, and given suggestions. Change the meeting time.

**Members present:**

Anoop Madipalli

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

Project proposal has been divided with each member. Everyone will present with his or her own part, Na Li and Vashnavi peddu took the task list and the critical path, Turkey took the ERD, and Anoop and Ravi took the DFD.

Meeting set on Friday 10 to 12.

**Meeting Notification:**

Meeting holds on Friday

Time: 10:00 am -12:00 pm

Location: Library

**Minutes of Meeting 5**

Date: 10/02/2015

**Objective of the meeting:**

Each of the members has aware of the main idea of the Visual Basic project, and work with the DFD and ERD. Second, deign for our project logo. Changed the meeting time and date.

**Members present:**

Anoop Madipalli

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

The visual Basic design forms plan has done, and ERD draft has done.

We did draw the project logo.

**Meeting Notification:**

Meeting holds on Thursday

Time: 8:00 am - 10:00 pm

Location: Library

**Minutes of Meeting 6**

Date: 10/09/2015

**Objective of the meeting:**

In this meeting, we will finish the Task list and Grant Chart, and Change the ERD and DFD errors.

**Members present:**

Anoop Madipalli

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

Finished the Task list and Grant Chart, and the DFD context and Diagram level 0.

**Meeting Notification:**

Meeting holds on Thursday

Time: 8:00 am - 10:00 pm

Location: Library

**Minutes of Meeting 7**

Date: 10/16/2015

**Objective of the meeting:**

Check the errors and change the ERD, and the DFD, Task list and Grant Chart everything before putting in in the Microsoft Project.

**Members present:**

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

Finalized the ERD, and the DFD, Task list and Grant Chart everything. And discuss with each member of the project two as whole

**Meeting Notification:**

Meeting holds on Thursday

Time: 8:00 am - 10:00 pm

Location: Library

**Minutes of Meeting 8**

Date: 10/22/2015

**Objective of the meeting:**

This meeting is to complete the whole project, and finalized the entire task list, grant chart, ERD and DFD is in correct format.

**Members present:**

Anoop Madipalli

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

ERD had changed by Turkey and Vashnavi, and DFD had changed by NA LI and Turkey. NA Li made the combination and submission.

**Meeting Notification:**

Meeting holds on Thursday

Time: 8:00 am - 10:00 pm

Location: Library

**Minutes of Meeting 9**

Date: 11/06/2015

**Objective of the meeting:**

Discuss the project three, and verify the project target and assign individual works.

**Members present:**

Anoop Madipalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

Group decided to split the whole project in to 7 parts:

Turkey: Database, report

Na LI: Customer form, sales invoices

Vashnavi: Booking form

Ravi: login form, Main form

Anoop: employee form

**Meeting Notification:**

Meeting holds on Thursday

Time: 8:00 am - 10:00 pm

Location: Library

**Minutes of Meeting 10**

Date: 11/13/2015

**Objective of the meeting:**

Accomplish the database as been discussed in the last meeting.

**Members present:**

Anoop Madipalli

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

Group members following the project requirement list, we need to including all the functions and start with VB program.

**Meeting Notification:**

Meeting holds on Thursday

Time: 8:00 am - 10:00 pm

Location: Library

**Minutes of Meeting 11**

Date: 10/20/2015

**Objective of the meeting:**

Chang the database suggested by Ravi Bhagyapalli. Create new customer form, and booking forms.

**Members present:**

Anoop Madipalli

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

The whole group cannot argue with Ravi Bhagyapalli, so we have to agree to change the database. Since we all had finished the customer form, and booking form, and report. Now, we have to start over again.

**Meeting Notification:**

Meeting holds on Monday

Time: 9:30 pm - 11:50 pm

Location: Grawn

**Minutes of Meeting 12**

Date: 10/30/2015

**Objective of the meeting:**

Accomplish everyone’s individual form and collect all the works.

**Members present:**

Na Li

Vaishnavi peddu

**Meeting conclusion:**

Employee form, Customer form, sale invoice, booking form, and report had integrated.

**Meeting Notification:**

Meeting holds on Thursday

Time: 8:00 am - 10:00 pm

Location: Ronan

**Minutes of Meeting 13**

Date: 12/01/2015

**Objective of the meeting:**

Fix the errors, and integrated with the login form and homepage.

**Members present:**

Anoop Madipalli

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

Fix the errors of the booking and sales invoices and Homepage.

**Meeting Notification:**

Meeting holds on Thursday

Time: 2:00 am - :00 pm

Location: Ronan

**Minutes of Meeting 13**

Date: 12/01/2015

**Objective of the meeting:**

Prettify the interface. Add background and accomplish the Power point Slides.

**Members present:**

Anoop Madipalli

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

Testing the system, and make sure it works well. And, finished the Power point slides.

**Meeting Notification:**

Meeting holds on Wednesday

Time: 2:00 am - 4:00 pm

Location: Rona

**Minutes of Meeting 14**

Date: 12/08/2015

**Objective of the meeting:**

Fix the problems that we had in the presentation, and correct the DFD.

**Members present:**

Anoop Madipalli

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

We decided to finish the last work by split it into few parts:

Anoop: Employee form

Ravi: login and main form

Turkey: ERD, and DFD corrections, and report documentation with code.

Na LI: Project Journals, Customer form, and sales invoices documentation with code.

Vashnavi: Cover page, context, business need, and booking documentation with code.

**Meeting Notification:**

Meeting holds on Tuesday

Time: 2:00 am - 4:00 pm

Location: Ronan

**Minutes of Meeting 15**

Date: 12/10/2015

**Objective of the meeting:**

Submission

**Members present:**

Anoop Madipalli

Ravi Bhagyapalli

Na Li

Turkey Alsohaibani

Vaishnavi peddu

**Meeting conclusion:**

Finished the completion works, and submitted it to the Blackboard.

Thanks for professor Dr. Huang‘s teaching. We had learned so much from your class.

**Meeting Notification:**

Meeting holds on Thursday

Time: 2:00 am - 4:00 pm

Location: Ronan