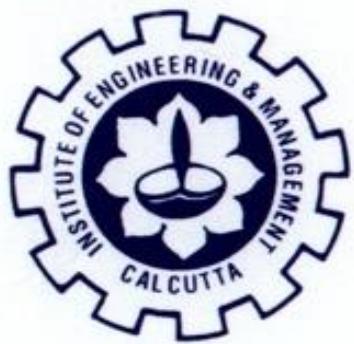


xBookShop

AN AUTOMATED BOOKSHOP
SOLUTION



Soham Banerjee

ROLL NO. C3-96

2ND YEAR, DEPT. OF COMPUTER SCIENCE AND ENGG.

INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA

Contents

Acknowledgements.....	2
Introduction	3
Design and Testing Environment	3
UI/UX Design.....	3
Application Screens.....	4
Login Screen.....	4
Home Screen.....	4
Billing Screen.....	5
Inventory Screen.....	5
Users Screen.....	6
Catalog Screen	6
Database Design.....	7
Splash Screen (\Forms\SplashView.frm).....	8
Source Code	8
Screenshot	10
Login Screen (\Forms\LoginView.frm)	11
Souce Code.....	11
Screenshots.....	13
Home Screen (\Forms\HomeView.frm).....	15
Source Code	15
Screenshots.....	22
Billing Screen (\Forms\BillingView.frm).....	24
Source Code	24
Screenshots.....	30
Bill Print Screen (\Forms\PrintView.frm).....	35
Source Code	35
Screenshots.....	36
Inventory Screen (\Forms\InventoryView.frm)	37
Source Code	37
Screenshots.....	59
Catalog Screen (\Forms\CatalogView.frm).....	62
Source Code	62
Screenshots.....	63
Users Screen (\Forms\UserView.frm).....	63
Source Code	63

Screenshots.....	72
modGDIplusResize (\lib\modGDIplusResize.bas)	75
Description	75
Source Code	75
modUser (\lib\modUser.bas)	81
Description	81
Source Code	81
modDatabase (\lib\modDatabase.bas)	82
Description	82
Source Code	82
Roadmap	83
Discussion.....	83
References	83

Acknowledgements

I would like to express my heartfelt gratitude to Prof. Subhabrata Sengupta for assigning me this project. It was a wonderful experience working on this project and I have gained a lot of knowledge in development of software applications and also improved my technical skill regarding the Win32 API.

I would like to thank Mr. Pinaki Karmakar, Ms. Arpita Ghosh and other faculty members, along with our respected Head of The Department of Computer Science and Engineering, Mrs. Debika Bhattacharya for extending their support and co-operation at all times.

Mention must be made of my friends for their encouragement and support, without whom completing this project would have been a distant reality. My heartfelt thanks goes out to them.

Last but not the least, it is only when you start working on software development that you truly appreciate the awesome tool that GitHub is. Thank you, Linus Torvalds!

This project uses the Roboto™ and Holo™ open typefaces, which are the intellectual property of Google Inc. and licensed under the GNU GPL.

This project uses the OpenFrameset layout framework which is licensed from <http://www.edgemean.net>. © 2009 Ed Wilk, Edgemean.net. All Rights Reserved.

BookShop

A COMPLETE BOOKSHOP AUTOMATION SOLUTION

Introduction

xBookShop is a Book Store Automation portal. xBookShop is designed with fluidity and ease of use in mind. xBookShop provides an option to replace pen-and-paper based Inventory Management with a fully computerized system, using a relational database backend. This is beneficial as it does away with clerical and human errors and also leads to savings in the long run. Also, xBookShop provides an integrated billing and payment portal, where customer receipts can be generated and printed.

xBookShop is designed using an event-driven approach, with more emphasis given on the day-to-day workflow in an actual book retail outlet. Most book stores have the following basic needs:

- To view a list of all available titles in-house.
- To be able to add and remove items from the list.
- To be able to update the stocks of each item in-house.
- To generate bills and receipts for customer copy.

xBookShop takes care of all these basic scenarios and also adds useful and essential information wherever relevant. We also find that in many stores, the store owner has access to all of the store documents, whereas the employees are not authorized to have that level of access. Keeping this situation in mind, xBookShop is built from the ground up to be a multi-user system, with access-protection being enforced among user accounts.

Design and Testing Environment

xBookShop was designed using Microsoft Visual Basic 6.0 and an RDBMS back-end implemented in a Microsoft Access Database. The Database Connection was implemented using the Microsoft ActiveX Data Object controls. xBookShop was tested extensively on Windows 8.1 (Update 1) and Windows 7 (SP1), and is designed to be compatible with all versions of Windows from Windows XP and up.

UI/UX Design

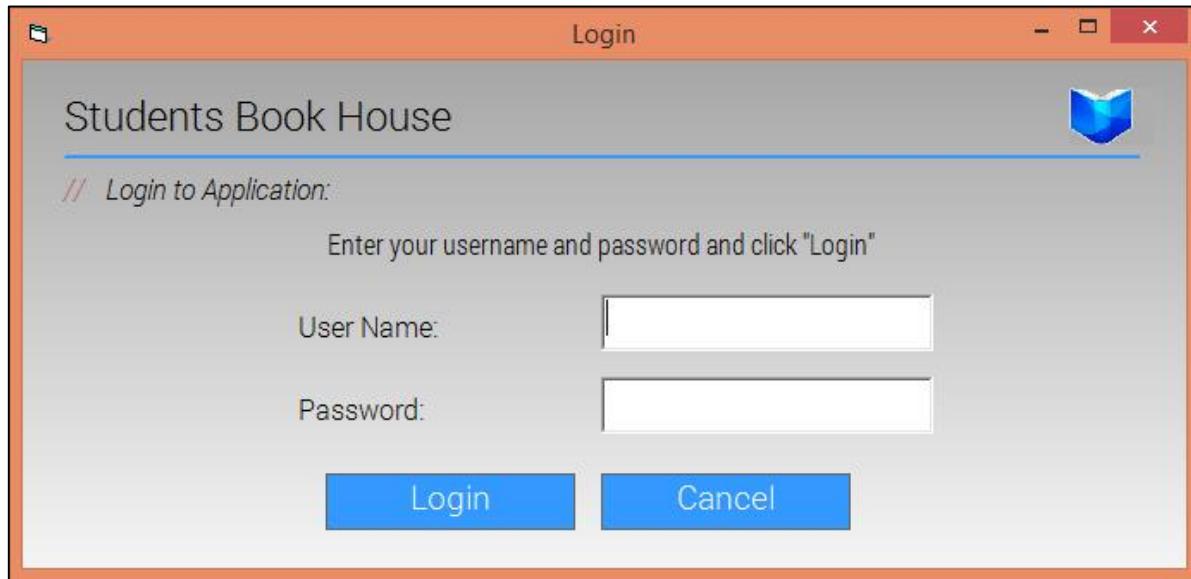
xBookShop is conformant to the design guidelines laid down by Microsoft in its Metro/Modern UI specification. The UI is designed to be contemporary and minimalist, with focus being on the content and data with minimal distractions. xBookShop uses the Roboto open font family by Google as its Primary UX font. It uses Holo Blue as consistent color scheme throughout all app screens.

The various applications views are designed to be complementary rather than contrasting. In line with the minimalist design, application screens fade in and out of view as and when they are required. Thus, at any given moment, there is only one application window on the desktop, which gives the user an immersive experience without any distractions or unnecessary information.

Some design elements such as the Shop Name, Shop Logo and Main Divider remain constant through the various application screens. This gives a sense of branding and also provides an aesthetic way to denote the relationship between various app views.

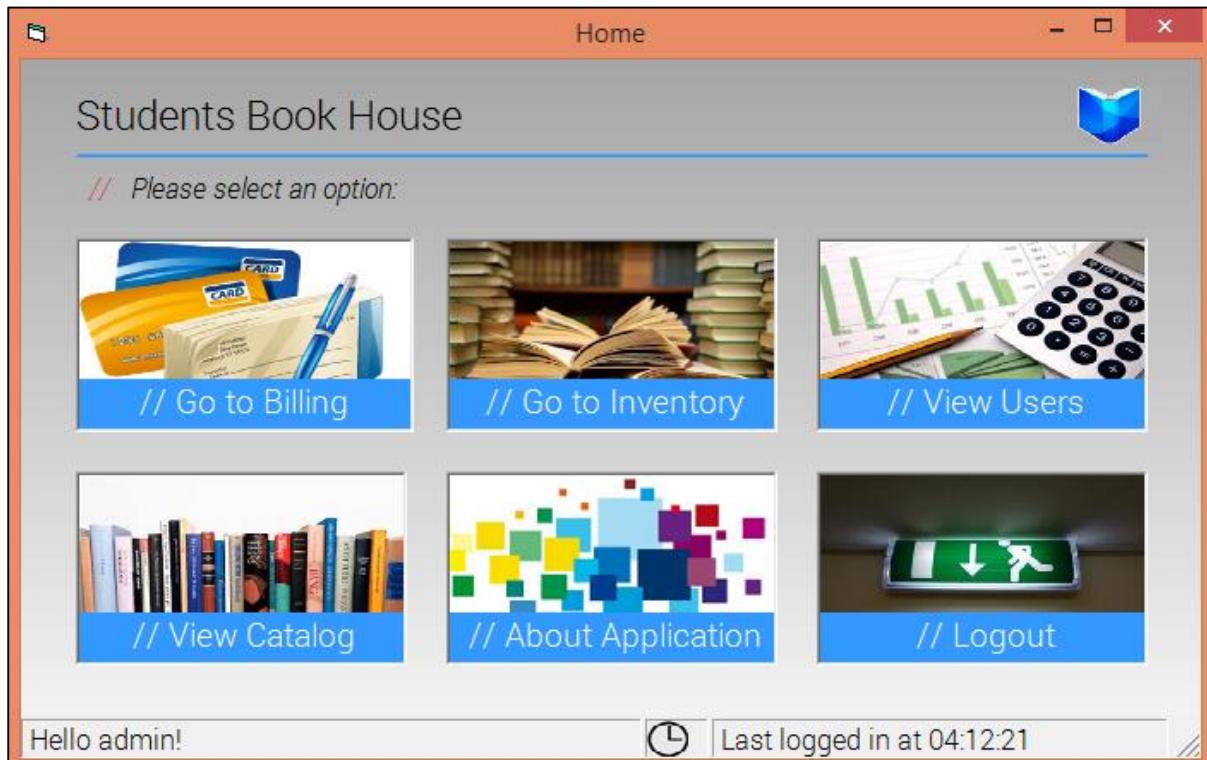
Application Screens

Login Screen



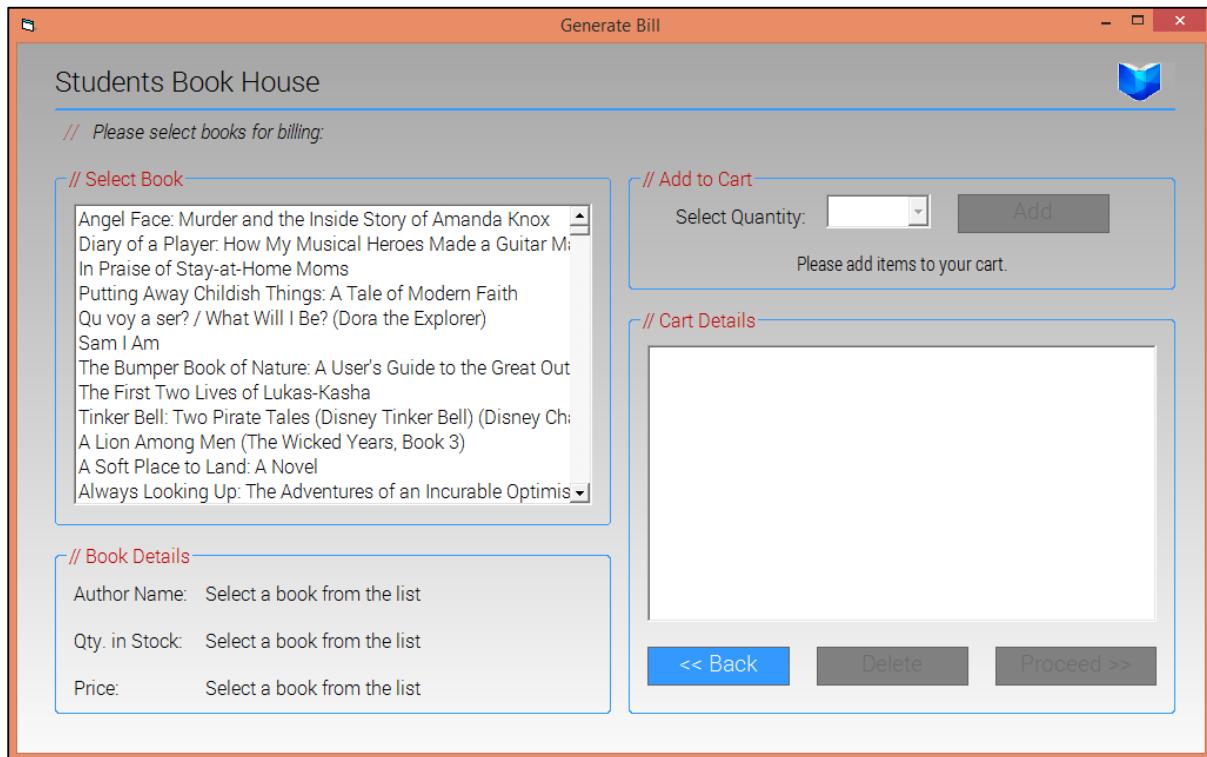
The login screen showing the prompt for user name and password.

Home Screen



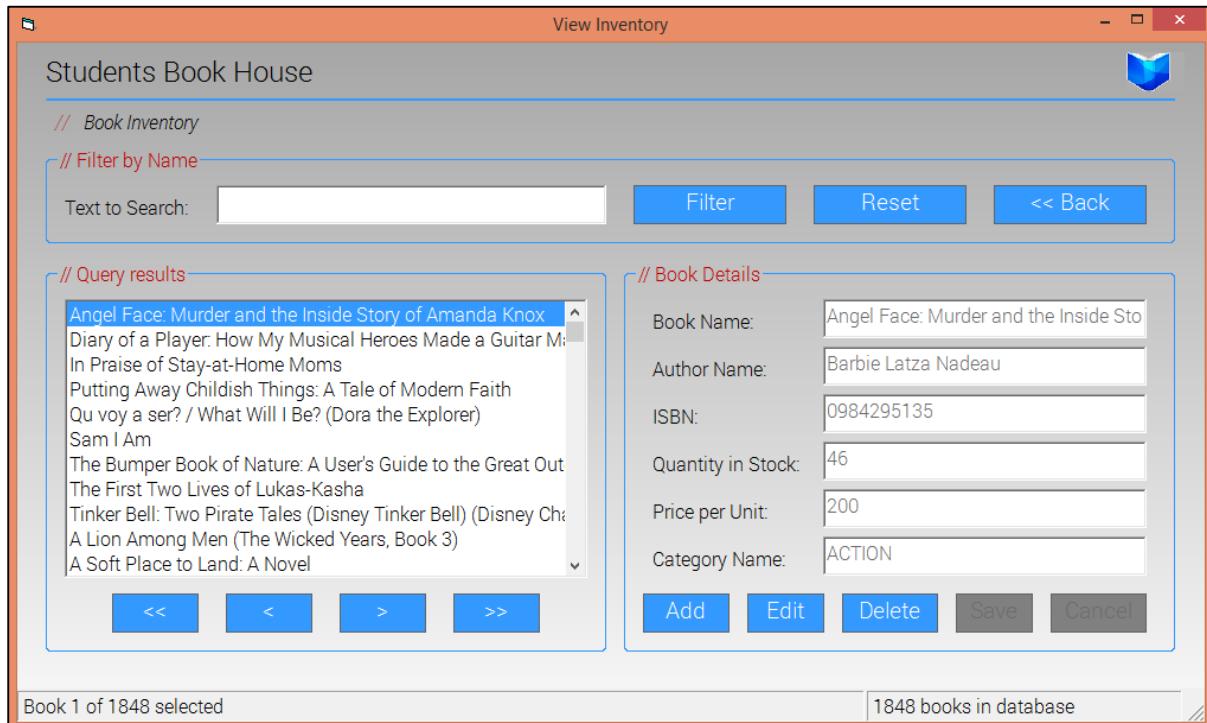
The initial or Home view of the application, after login. A simple gradient has been used as the app view background throughout all app screens.

Billing Screen



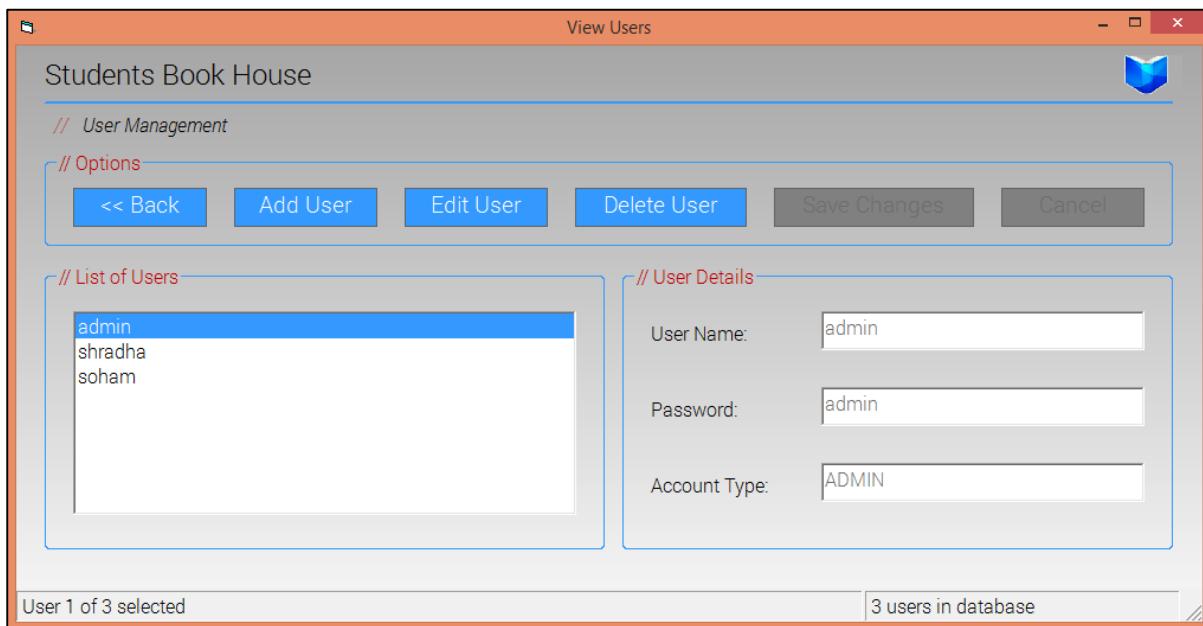
The billing screen to generate invoices. Note that the disabled controls are greyed out and are dynamically enabled as and when required.

Inventory Screen



The inventory screen, containing options for inserting, deleting and updating current stocks of books as well as book details. The disabled controls are grayed out.

Users Screen



The user management portal. Only an administrator can add or remove other users.

Catalog Screen

The screenshot shows a Windows application window titled "View Catalog". The title bar also includes the text "Students Book House" and a blue book icon. The main area displays a table of books with columns: BOOKNAME, AUTHORNAME, CATEGORY, ISBN, PUBLISHER, BINDING, and MRP. The table lists 15 books, such as "Angel Face: Murder" by Barbie Latza Nadeau, "Diary of a Player" by Brad Paisley and Da, and "In Praise of Stay-at-Home Parents" by Laura Schlessinger. A message at the top of the table says "Use Arrow Keys to navigate". At the bottom, there is a blue button labeled "<< Home".

Use Arrow Keys to navigate						
BOOKNAME	AUTHORNAME	CATEGORY	ISBN	PUBLISHER	BINDING	MRP
► Angel Face: Murder	Barbie Latza Nadeau	ACTION	0984295135	21st Century Books	Paper Back	200
Diary of a Player: How to Be a Woman	Brad Paisley and Da	ACTION	1451625529	21st Century Books	Hard Cover	400
In Praise of Stay-at-Home Parents	Laura Schlessinger	ACTION	0061690295	21st Century Books	Hard Cover	500
Putting Away Childhood	Marcus J. Borg	ACTION	0061888141	21st Century Books	Audio Cassette	100
Qu'voy a ser? / What's Next?	Unknown Author	ACTION	1416933662	21st Century Books	Audio Cassette	200
Sam I Am	Unknown Author	ACTION	0439439671	21st Century Books	Audio Cassette	300
The Bumper Book of Stories	Neil Sheehan	ACTION	0679422846	21st Century Books	Hard Cover	400
The First Two Lives	Unknown Author	ACTION	0141300574	21st Century Books	Audio Cassette	500
Tinker Bell: Two Piracies	Unknown Author	ACTION	0736426817	21st Century Books	Audio Cassette	100
A Lion Among Men	Unknown Author	ACTION	0060548924	21st Century Books	Hard Cover	200
A Soft Place to Land	Unknown Author	ACTION	1416558691	21st Century Books	Audio Cassette	300
Always Looking Up	Unknown Author	ACTION	1401303382	Hyperion	Audio Cassette	400

The catalog screen, showing a list of all books and prices in tabular form.

Database Design

The heart of xBookShop is in its Database. xBookShop is designed to work with either a local or remote data source. In its current iteration, xBookShop references a database stored at db\BookDB.mdb with respect to the current working directory. This can be configured to any absolute or relative path, or any ODBC data source.

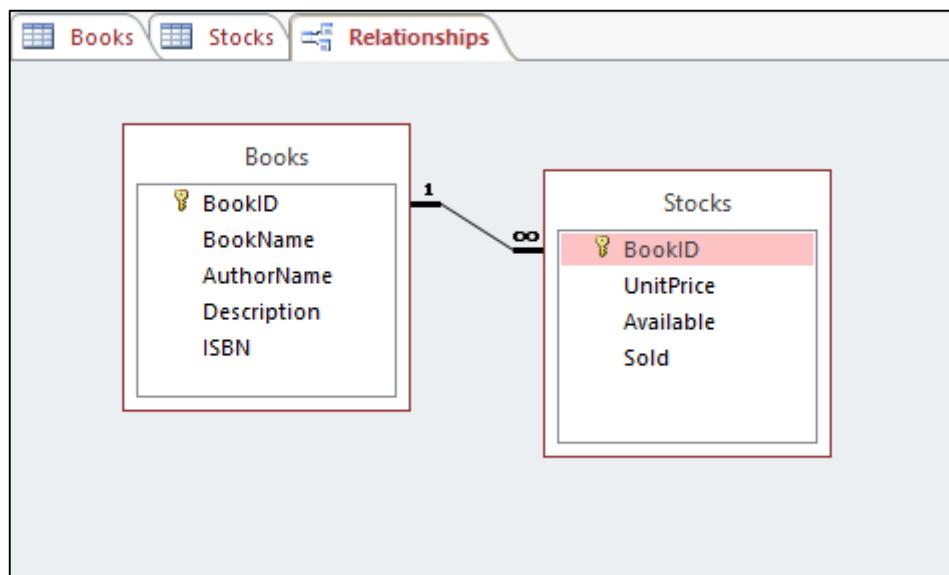
The database has 3 main tables:

- Books – Keeps a track of the individual books in the store.
- Stocks – Keeps a track of the available stock of each book, prices and sales.
- Users – Keeps a track of the users recognized in the storefront, and account types.

Since we are using an RDBMS system, we have enforced a Left Join upon Books and Stocks, so that a Book and its Stock are directly linked.

The screenshot shows the Microsoft Access interface with the 'Books' table selected. The table has columns: BookID, BookName, AuthorName, Description, ISBN, and a 'Click to Add' button. Three rows are present:

BookID	BookName	AuthorName	Description	ISBN	Click to Add
1	Book 1	Author 1	1234	1234	
*	UnitPrice	Available	Sold	Click to Add	
	500	24	11		
*	0	0	0		
2	Book 2	Author 2	1234	4567	
*	UnitPrice	Available	Sold	Click to Add	
	200	30	18		
*	0	0	0		
3	Book 3	Author 3			
*	UnitPrice	Available	Sold	Click to Add	
	0	0	0		
*	(New)				



The database can be password-protected and encrypted to ensure data security and integrity.

Form Design

Splash Screen (\Forms\SplashView.frm)

Source Code

```
Dim dShort As Integer
Dim dLong, dLong2, dLong3 As Integer
Dim animCount, animCount2, animCount3 As Integer
Dim doneAnim, animStarted2, animStarted3 As Boolean
Dim loadTime As Long

Private Sub Form_Load()
    doneAnim = False
    animCount = 0
    dShort = 100
    dLong = 300
    dLong2 = 300
    dLong3 = 300

    animStarted2 = False
    animStarted3 = False
    Timer1.Enabled = True
    Label3.Left = 0
    Label3.Visible = False
    Label2.Left = 0
    Label2.Visible = False
    Label1.Left = 0
    Label1.Visible = False
    Label4.FontBold = True

    loadTime = 0
    initUser
    initDB
End Sub

Private Sub Timer1_Timer()
    If animCount = 0 Then
        Label1.Visible = True
        Label1.Left = 0
        animCount = animCount + 1
    End If
    If animCount > 0 And animCount <= 8 Then
        Label1.Left = Label1.Left + dLong
        animCount = animCount + 1
        dLong = dLong + 10
    End If
    If animCount > 8 And animCount <= 25 Then
        Label1.Left = Label1.Left + dShort
        animCount = animCount + 1
    End If
End Sub
```

```
If animCount > 25 And animCount <= 38 Then
    dLong = dLong - 10
    Label1.Left = Label1.Left + dLong
    animCount = animCount + 1
End If
If animCount > 38 Then
    Label1.Visible = False
    Label1.Left = 0
    animCount = 0
    dLong = 300
End If

animCount2 = (animCount - 7)
If animCount2 < 0 Then animCount2 = animCount2 + 39

If animCount2 = 0 Then
    Label2.Visible = True
    Label2.Left = 0
    dLong2 = 300
    animStarted2 = True
End If
If animCount2 > 0 And animCount2 <= 8 And animStarted2 Then
    Label2.Left = Label2.Left + dLong2
    dLong2 = dLong2 + 20
End If
If animCount2 > 8 And animCount2 <= 25 And animStarted2 Then
    Label2.Left = Label2.Left + dShort
End If
If animCount2 > 25 And animCount2 <= 38 And animStarted2 Then
    dLong2 = dLong2 - 10
    Label2.Left = Label2.Left + dLong2
End If
If animCount2 > 38 Then
    Label2.Visible = False
    Label2.Left = 0
End If

animCount3 = (animCount - 16)
If animCount3 < 0 Then animCount3 = animCount3 + 39

If animCount3 = 0 Then
    Label3.Visible = True
    Label3.Left = 0
    dLong3 = 300
    animStarted3 = True
End If
If animCount3 > 0 And animCount3 <= 8 And animStarted3 Then
    Label3.Left = Label3.Left + dLong3
    dLong3 = dLong3 + 5
End If
If animCount3 > 8 And animCount3 <= 25 And animStarted3 Then
```

```

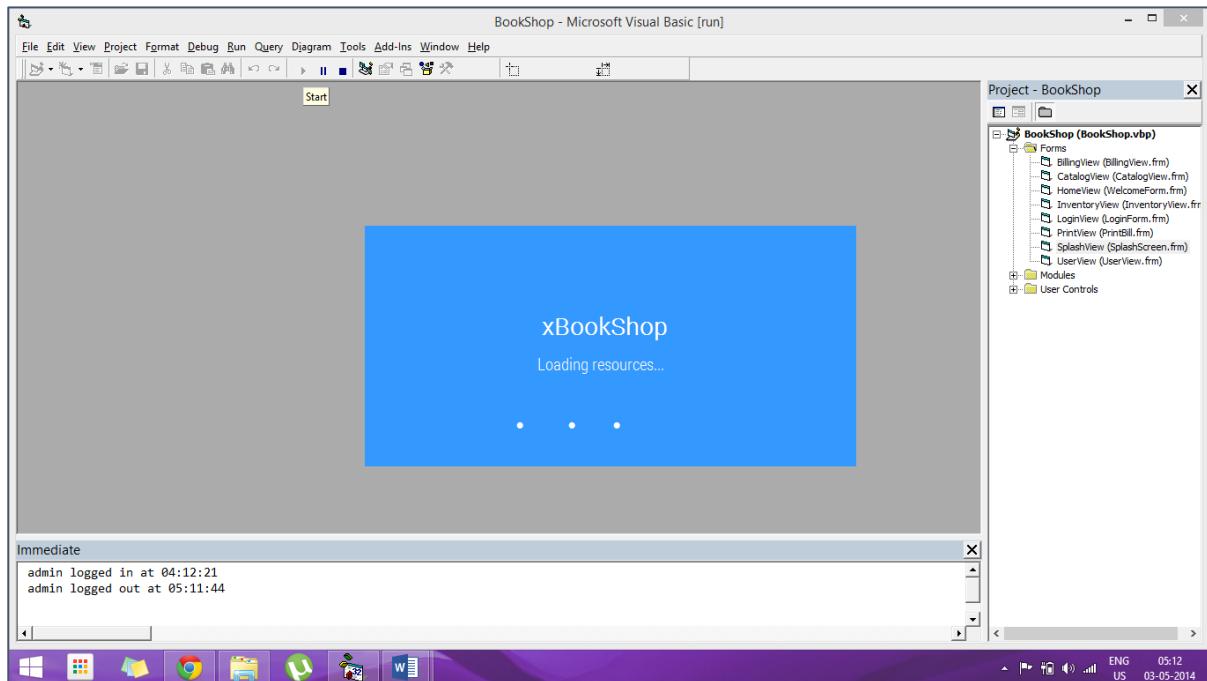
Label3.Left = Label3.Left + dShort
End If
If animCount3 > 25 And animCount3 <= 38 And animStarted3 Then
    dLong3 = dLong3 - 5
    Label3.Left = Label3.Left + dLong3
End If
If animCount3 > 38 Then
    Label3.Visible = False
    Label3.Left = 0
End If

loadTime = loadTime + Timer1.Interval
If loadTime = 1000 Then MessageBar.Caption = "Loading resources..."
If loadTime = 2500 Then MessageBar.Caption = "Initializing database connection..."
If loadTime = 4500 Then MessageBar.Caption = "All done!"

If loadTime > 5000 Then
    Timer1.Enabled = False
    Unload Me
    LoginView.Show
End If
End Sub

```

Screenshot



The initial loading screen of the application. The application caches all required graphics and resources while the Splash Screen is showing. It also initializes all database connections.

Login Screen (\Forms\LoginView.frm)

Source Code

```
Dim Token As Long
Dim C As Long
Dim exitVal As Integer

Dim login As ADODB.Recordset

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdCancel_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    cmdLogin.BackColor = &H8000000D
    cmdCancel.BackColor = &HC0&
    cmdLogin.FontItalic = False
    cmdCancel.FontItalic = True
End Sub

Private Sub cmdLogin_Click()
    Timer1.Enabled = False

    If txtUsername.Text = "" Then
        MessageBar.Caption = "ERROR: Username cannot be empty!"
        txtUsername.SetFocus
        Timer1.Enabled = True
        Exit Sub
    ElseIf txtPassword.Text = "" Then
        MessageBar.Caption = "ERROR: Password cannot be empty!"
        txtPassword.SetFocus
        Timer1.Enabled = True
        Exit Sub
    Else
        Set login = New ADODB.Recordset
        login.CursorType = adOpenDynamic
        login.CursorLocation = adUseClient
        login.LockType = adLockOptimistic
        login.Open "Select * from Users where UserName=''" & txtUsername.Text & "'", conn,
        login.CursorType, login.LockType, adCmdUnknown

        If login.EOF Then
            MessageBar.Caption = "ERROR: No such user exists! Please check for spelling
errors."
            txtUsername.SetFocus
            Timer1.Enabled = True
            Exit Sub
        Else
            If login.Fields("Password") = txtPassword.Text Then
                Me.Hide
            End If
        End If
    End If
End Sub
```

```
        HomeView.Show
        usrLogin.login.Fields("UserName"), login.Fields("Password"),
login.Fields("UserType")
        txtUsername.Text = ""
        txtPassword.Text = ""

    Else
        MessageBar.Caption = "ERROR: Wrong password! Please check for
spelling/capitalization errors."
        txtPassword.SetFocus
        Timer1.Enabled = True
        Exit Sub
    End If
End If
End If
End Sub

Private Sub cmdLogin_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    cmdLogin.BackColor = &HC0&
    cmdCancel.BackColor = &H8000000D
    cmdLogin.FontItalic = True
    cmdCancel.FontItalic = False
End Sub

Private Sub Form_Activate()
    txtUsername.SetFocus
    exitVal = vbNo
End Sub

Private Sub Form_Initialize()
    Token = InitGDIPlus
    C = Me.BackColor
    If C < 0 Then C = GetSysColor(C - &H80000000)
End Sub

Private Sub Form_Load()
    ShopLogo.Picture = LoadPictureGDIPlus(App.Path & "\Images\logo.png", 100, 80, &HADADAD,
True)

    cmdLogin.BackColor = &H8000000D
    cmdCancel.BackColor = &H8000000D

    Timer1.Enabled = False
    exitVal = vbNo
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    cmdLogin.BackColor = &H8000000D
    cmdCancel.BackColor = &H8000000D
    cmdLogin.FontItalic = False
    cmdCancel.FontItalic = False
```

End Sub

```

Private Sub Form_Unload(Cancel As Integer)
    exitVal = MsgBox("Are you sure you want to exit the application?", vbYesNo +
    vbDefaultButton2 + vbQuestion + vbApplicationModal, "Confirm Exit")
    If exitVal = vbYes Then
        FreeGDIPlus Token
    End
    Else
        Cancel = 1
    End If
End Sub

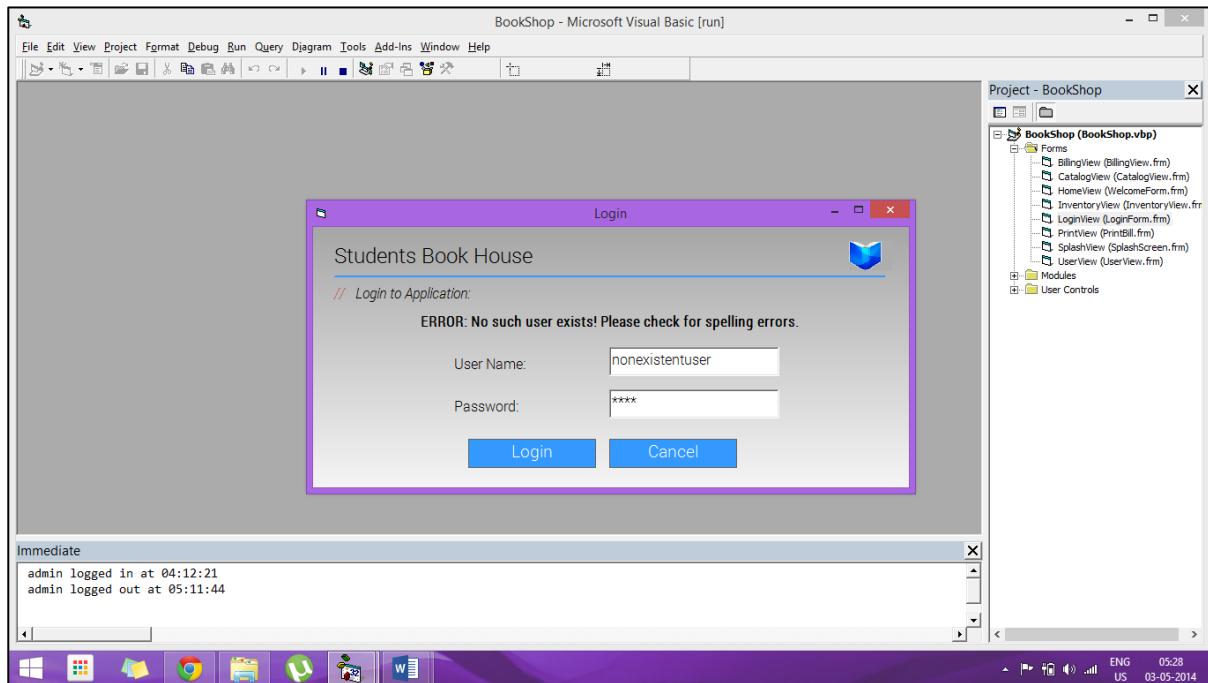
Private Sub Timer1_Timer()
    MessageBar.FontBold = Not MessageBar.FontBold
End Sub

Private Sub txtUsername_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = 13 Then Call cmdLogin_Click
End Sub

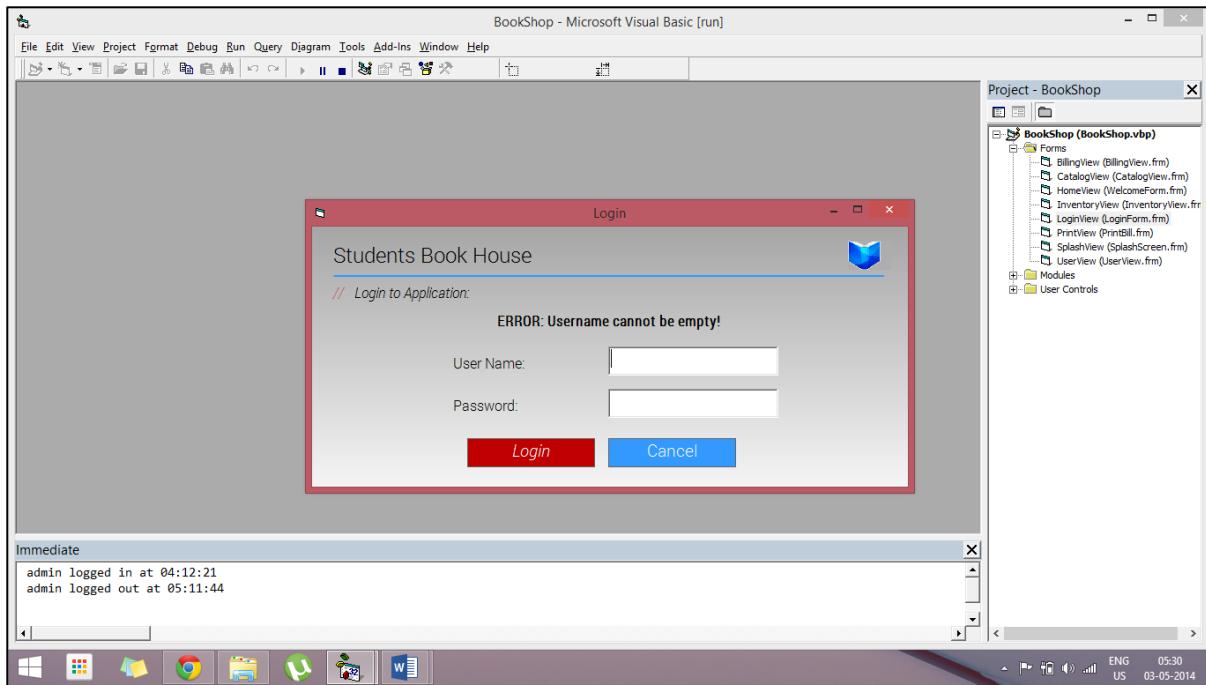
Private Sub txtPassword_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = 13 Then Call cmdLogin_Click
End Sub

```

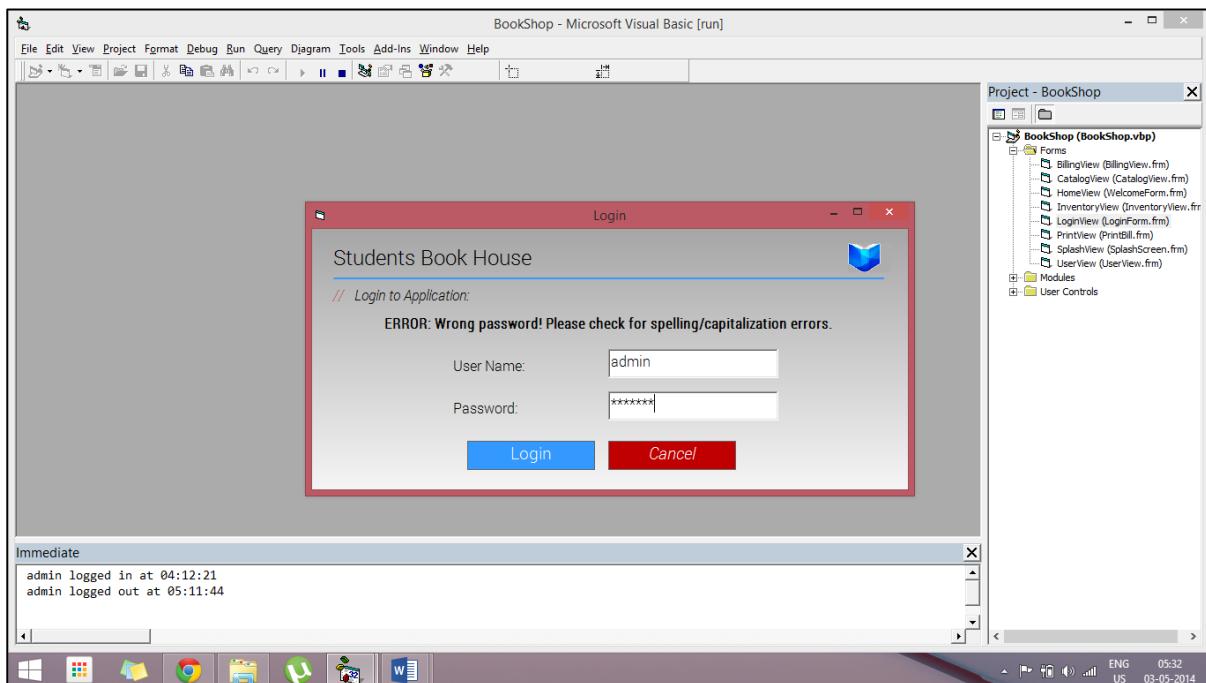
Screenshots



The login screen showing an error for a non-existent user login. An animated error message is flashed inline instead of using a MsgBox, as it is less obtrusive.



Login screen showing error dialog for empty username.



Login screen showing error dialog for wrong password entered.

Home Screen (\Forms\HomeView.frm)

Source Code

```
Dim Token As Long
Dim C As Long
Dim exitVal As Integer

Private Sub AboutIcon_Click()
    MsgBox "xBookShop" & vbCrLf & "Developed by Soham Banerjee" & vbCrLf & "2nd Year,
CSE Dept." & vbCrLf & "IEM Kolkata" & vbCrLf & "www.xeqt.in" & vbCrLf & vbCrLf
& "Licensed to Students Book House" & vbCrLf & "All Rights Reserved", vbApplicationModal
+ vbOKOnly + vbInformation, "About xBookShop"
End Sub

Private Sub AboutOption_Click()
    MsgBox "xBookShop" & vbCrLf & "Developed by Soham Banerjee" & vbCrLf & "2nd Year,
CSE Dept." & vbCrLf & "IEM Kolkata" & vbCrLf & "www.xeqt.in" & vbCrLf & vbCrLf
& "Licensed to Students Book House" & vbCrLf & "All Rights Reserved", vbApplicationModal
+ vbOKOnly + vbInformation, "About xBookShop"
End Sub

Private Sub BillingIcon_Click()
    Me.Hide
    BillingView.Show
End Sub

Private Sub BillingOption_Click()
    Me.Hide
    BillingView.Show
End Sub

Private Sub CatalogOption_Click()
    Me.Hide
    CatalogView.Show
End Sub

Private Sub CatalogIcon_Click()
    Me.Hide
    CatalogView.Show
End Sub

Private Sub ExitIcon_Click()
    Unload Me
End Sub

Private Sub ExitOption_Click()
    Unload Me
End Sub

Private Sub Form_Activate()
    StatusView.Panels(1).Text = "Hello " & username & "!"
```

```
    StatusView.Panels(3).Text = "Last logged in at " & lastLogin
    StatusView.Panels(2).Picture = LoadPictureGDIPlus(App.Path & "\Images\clock.png", 30,
25, C, True)
End Sub

Private Sub Form_Initialize()
    Token = InitGDIPlus
    C = Me.BackColor
    If C < 0 Then C = GetSysColor(C - &H80000000)
End Sub

Private Sub Form_Load()
    ShopLogo.Picture = LoadPictureGDIPlus(App.Path & "\Images\logo.png", 100, 80, &HADADAD,
True)
    BillingIcon.Picture = LoadPictureGDIPlus(App.Path & "\Images\billing.png", 400, 300,
vblWhite, True)
    InventoryIcon.Picture = LoadPictureGDIPlus(App.Path & "\Images\inventory.jpg", 400,
300, C, True)
    ReportsIcon.Picture = LoadPictureGDIPlus(App.Path & "\Images\reports.jpg", 400, 300, C,
True)
    CatalogIcon.Picture = LoadPictureGDIPlus(App.Path & "\Images\catalog.jpg", 400, 300, C,
True)
    AboutIcon.Picture = LoadPictureGDIPlus(App.Path & "\Images\about.png", 500, 300, C,
True)
    ExitIcon.Picture = LoadPictureGDIPlus(App.Path & "\Images\exit.jpg", 500, 300, C, True)

    BillingOption.BackColor = &H8000000D
    InventoryOption.BackColor = &H8000000D
    ReportsOption.BackColor = &H8000000D
    CatalogOption.BackColor = &H8000000D
    AboutOption.BackColor = &H8000000D
    ExitOption.BackColor = &H8000000D

    exitVal = vbNo
End Sub

Private Sub Form_Unload(Cancel As Integer)
    exitVal = MsgBox("Are you sure you want to log out?", vbYesNo + vbDefaultButton2 +
vbQuestion + vbApplicationModal, "Confirm Exit")
    If exitVal = vbYes Then
        usrLogout
        LoginView.Show
        FreeGDIPlus Token
        Unload Me
    Else
        Cancel = 1
    End If
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BillingOption.BackColor = &H8000000D
```

```
InventoryOption.BackColor = &H8000000D
ReportsOption.BackColor = &H8000000D
CatalogOption.BackColor = &H8000000D
AboutOption.BackColor = &H8000000D
ExitOption.BackColor = &H8000000D

BillingOption.FontItalic = False
InventoryOption.FontItalic = False
ReportsOption.FontItalic = False
CatalogOption.FontItalic = False
AboutOption.FontItalic = False
ExitOption.FontItalic = False

End Sub

Private Sub BillingOption_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BillingOption.BackColor = &H80&
    InventoryOption.BackColor = &H8000000D
    ReportsOption.BackColor = &H8000000D
    CatalogOption.BackColor = &H8000000D
    AboutOption.BackColor = &H8000000D
    ExitOption.BackColor = &H8000000D

    BillingOption.FontItalic = True
    InventoryOption.FontItalic = False
    ReportsOption.FontItalic = False
    CatalogOption.FontItalic = False
    AboutOption.FontItalic = False
    ExitOption.FontItalic = False

End Sub

Private Sub BillingIcon_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BillingOption.BackColor = &H80&
    InventoryOption.BackColor = &H8000000D
    ReportsOption.BackColor = &H8000000D
    CatalogOption.BackColor = &H8000000D
    AboutOption.BackColor = &H8000000D
    ExitOption.BackColor = &H8000000D

    BillingOption.FontItalic = True
    InventoryOption.FontItalic = False
    ReportsOption.FontItalic = False
    CatalogOption.FontItalic = False
    AboutOption.FontItalic = False
    ExitOption.FontItalic = False

End Sub

Private Sub InventoryIcon_Click()
    If usertype = "ADMIN" Then
        Me.Hide
    End If

```

```
    InventoryView.Show
Else
    MsgBox "Only administrators can edit the inventory. Please login with an
administrator account.", vbApplicationModal + vbOKOnly + vbInformation + vbDefaultButton1,
"Administrator Account Required"
End If
End Sub

Private Sub InventoryOption_Click()
If usertype = "ADMIN" Then
    Me.Hide
    InventoryView.Show
Else
    MsgBox "Only administrators can edit the inventory. Please login with an
administrator account.", vbApplicationModal + vbOKOnly + vbInformation + vbDefaultButton1,
"Administrator Account Required"
End If
End Sub

Private Sub InventoryOption_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    BillingOption.BackColor = &H8000000D
    InventoryOption.BackColor = &H80&
    ReportsOption.BackColor = &H8000000D
    CatalogOption.BackColor = &H8000000D
    AboutOption.BackColor = &H8000000D
    ExitOption.BackColor = &H8000000D

    BillingOption.FontItalic = False
    InventoryOption.FontItalic = True
    ReportsOption.FontItalic = False
    CatalogOption.FontItalic = False
    AboutOption.FontItalic = False
    ExitOption.FontItalic = False
End Sub

Private Sub InventoryIcon_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    BillingOption.BackColor = &H8000000D
    InventoryOption.BackColor = &H80&
    ReportsOption.BackColor = &H8000000D
    CatalogOption.BackColor = &H8000000D
    AboutOption.BackColor = &H8000000D
    ExitOption.BackColor = &H8000000D

    BillingOption.FontItalic = False
    InventoryOption.FontItalic = True
    ReportsOption.FontItalic = False
    CatalogOption.FontItalic = False
    AboutOption.FontItalic = False
    ExitOption.FontItalic = False
End Sub
```

End Sub

```
Private Sub ReportsIcon_Click()
    If usertype = "ADMIN" Then
        Me.Hide
        Userview.Show
    Else
        MsgBox "Only administrators can edit users. Please login with an administrator account.", vbApplicationModal + vbOKOnly + vbInformation + vbDefaultButton1, "Administrator Account Required"
    End If
End Sub
```

```
Private Sub ReportsOption_Click()
    If usertype = "ADMIN" Then
        Me.Hide
        Userview.Show
    Else
        MsgBox "Only administrators can edit users. Please login with an administrator account.", vbApplicationModal + vbOKOnly + vbInformation + vbDefaultButton1, "Administrator Account Required"
    End If
End Sub
```

```
Private Sub ReportsOption_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BillingOption.BackColor = &H8000000D
    InventoryOption.BackColor = &H8000000D
    ReportsOption.BackColor = &H80&
    CatalogOption.BackColor = &H8000000D
    AboutOption.BackColor = &H8000000D
    ExitOption.BackColor = &H8000000D

    BillingOption.FontItalic = False
    InventoryOption.FontItalic = False
    ReportsOption.FontItalic = True
    CatalogOption.FontItalic = False
    AboutOption.FontItalic = False
    ExitOption.FontItalic = False
End Sub
```

```
Private Sub ReportsIcon_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BillingOption.BackColor = &H8000000D
    InventoryOption.BackColor = &H8000000D
    ReportsOption.BackColor = &H80&
    CatalogOption.BackColor = &H8000000D
    AboutOption.BackColor = &H8000000D
    ExitOption.BackColor = &H8000000D

    BillingOption.FontItalic = False
```

```
    InventoryOption.FontItalic = False
    ReportsOption.FontItalic = True
    CatalogOption.FontItalic = False
    AboutOption.FontItalic = False
    ExitOption.FontItalic = False
End Sub

Private Sub CatalogOption_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BillingOption.BackColor = &H8000000D
    InventoryOption.BackColor = &H8000000D
    ReportsOption.BackColor = &H8000000D
    CatalogOption.BackColor = &H80&
    AboutOption.BackColor = &H8000000D
    ExitOption.BackColor = &H8000000D

    BillingOption.FontItalic = False
    InventoryOption.FontItalic = False
    ReportsOption.FontItalic = False
    CatalogOption.FontItalic = True
    AboutOption.FontItalic = False
    ExitOption.FontItalic = False
End Sub

Private Sub CatalogIcon_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BillingOption.BackColor = &H8000000D
    InventoryOption.BackColor = &H8000000D
    ReportsOption.BackColor = &H8000000D
    CatalogOption.BackColor = &H80&
    AboutOption.BackColor = &H8000000D
    ExitOption.BackColor = &H8000000D

    BillingOption.FontItalic = False
    InventoryOption.FontItalic = False
    ReportsOption.FontItalic = False
    CatalogOption.FontItalic = True
    AboutOption.FontItalic = False
    ExitOption.FontItalic = False
End Sub

Private Sub AboutOption_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BillingOption.BackColor = &H8000000D
    InventoryOption.BackColor = &H8000000D
    ReportsOption.BackColor = &H8000000D
    CatalogOption.BackColor = &H8000000D
    AboutOption.BackColor = &H80&
    ExitOption.BackColor = &H8000000D

    BillingOption.FontItalic = False

```

```
    InventoryOption.FontItalic = False
    ReportsOption.FontItalic = False
    CatalogOption.FontItalic = False
    AboutOption.FontItalic = True
    ExitOption.FontItalic = False
End Sub

Private Sub AboutIcon_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BillingOption.BackColor = &H8000000D
    InventoryOption.BackColor = &H8000000D
    ReportsOption.BackColor = &H8000000D
    CatalogOption.BackColor = &H8000000D
    AboutOption.BackColor = &H80&
    ExitOption.BackColor = &H8000000D

    BillingOption.FontItalic = False
    InventoryOption.FontItalic = False
    ReportsOption.FontItalic = False
    CatalogOption.FontItalic = False
    AboutOption.FontItalic = True
    ExitOption.FontItalic = False
End Sub

Private Sub ExitOption_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BillingOption.BackColor = &H8000000D
    InventoryOption.BackColor = &H8000000D
    ReportsOption.BackColor = &H8000000D
    CatalogOption.BackColor = &H8000000D
    AboutOption.BackColor = &H8000000D
    ExitOption.BackColor = &H80&

    BillingOption.FontItalic = False
    InventoryOption.FontItalic = False
    ReportsOption.FontItalic = False
    CatalogOption.FontItalic = False
    AboutOption.FontItalic = False
    ExitOption.FontItalic = True
End Sub

Private Sub ExitIcon_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BillingOption.BackColor = &H8000000D
    InventoryOption.BackColor = &H8000000D
    ReportsOption.BackColor = &H8000000D
    CatalogOption.BackColor = &H8000000D
    AboutOption.BackColor = &H8000000D
    ExitOption.BackColor = &H80&

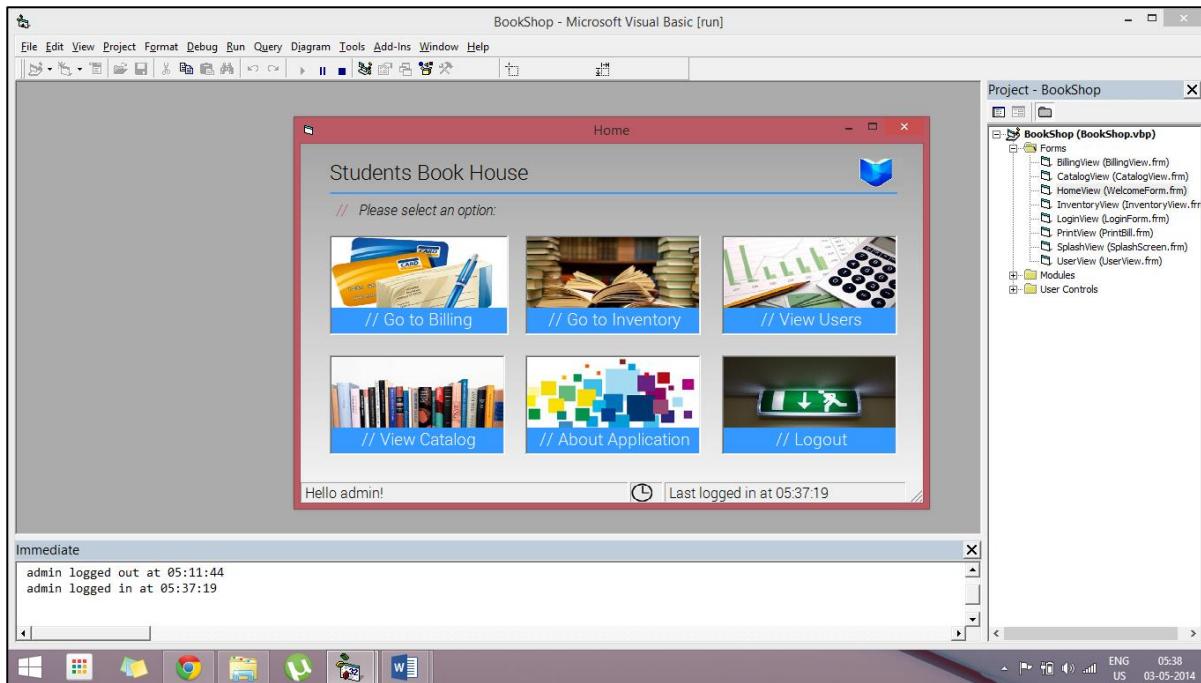
    BillingOption.FontItalic = False
```

```

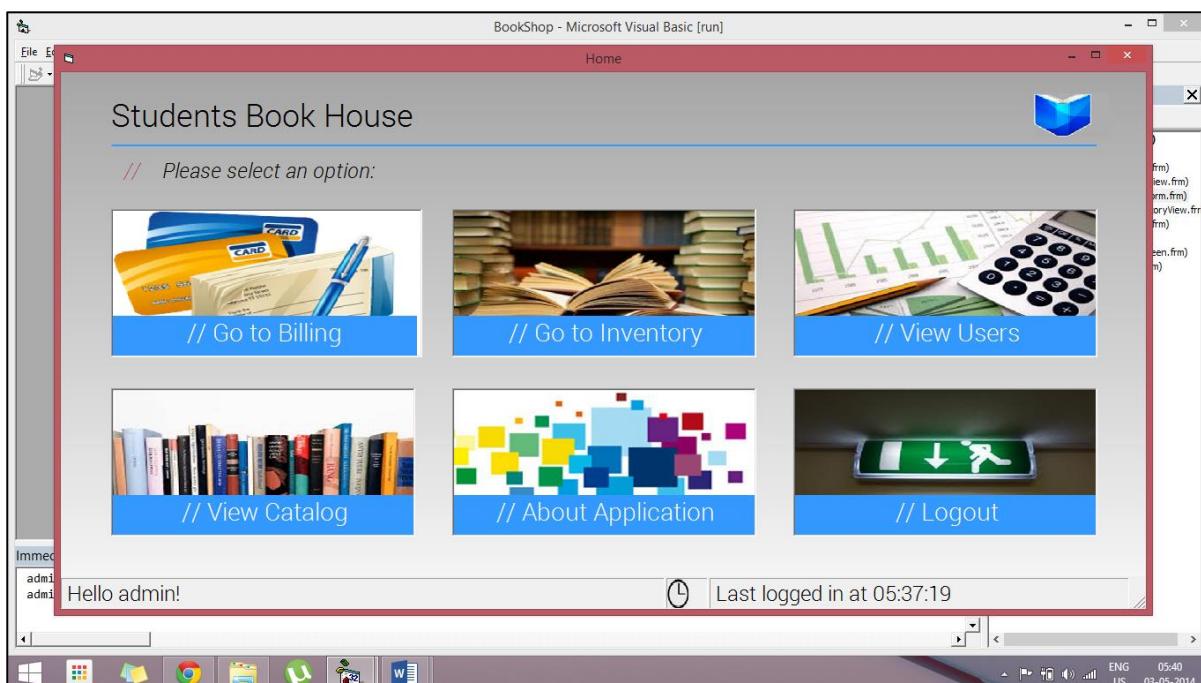
InventoryOption.UseFontItalic = False
ReportsOption.UseFontItalic = False
CatalogOption.UseFontItalic = False
AboutOption.UseFontItalic = False
ExitOption.UseFontItalic = True
End Sub

```

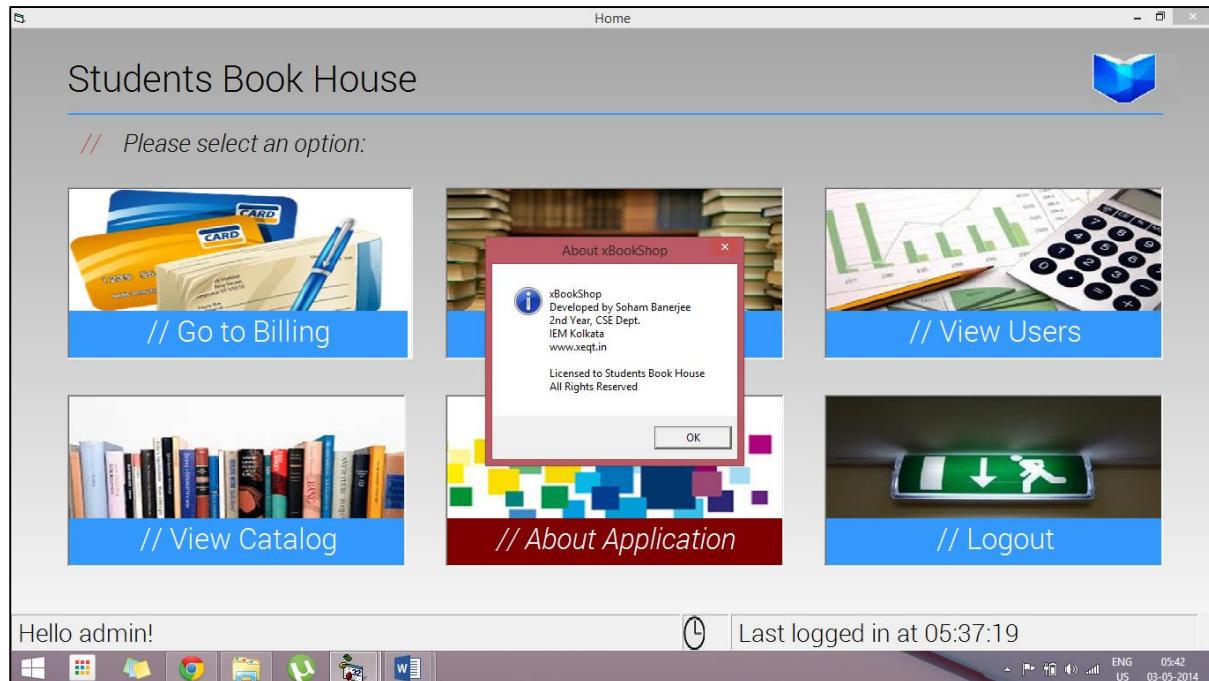
Screenshots



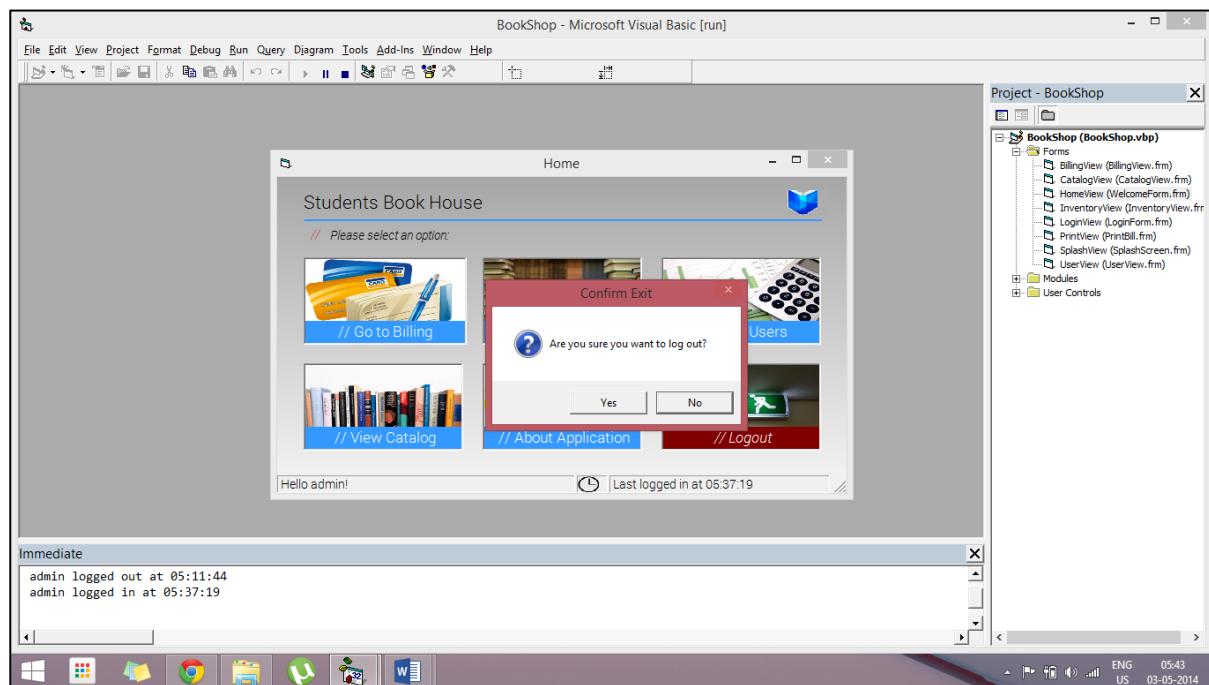
The Home Screen after logging in, showing the name of the currently logged in user, and the time of the last log-in.



Home Screen showing Active Resize in action. The app scales itself automatically on resize to use all available screen real estate.



The Home Screen on clicking the “About Application” option



The Home Screen on clicking the “Logout” option

Billing Screen (\Forms\BillingView.frm)

Source Code

```
Option Explicit

Dim Token As Long
Dim C As Long
Dim exitVal As Integer
Dim billCount As Integer

Private Sub BillList_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    If cmdProceed.Enabled Then cmdProceed.BackColor = &H8000000D
    If cmdBack.Enabled Then cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdProceed.FontItalic = False
    cmdDelete.FontItalic = False
    cmdBack.FontItalic = False
End Sub

Private Sub BookContainer_Click()
    If BookContainer.SelectedItem = Null Then
        Exit Sub
    Else
        BookList.Recordset.AbsolutePosition = BookContainer.SelectedItem
        txtAuthor.Caption = BookList.Recordset.Fields("AUTHORNAME")
        txtQty.Caption = BookList.Recordset.Fields("QTY")
        txtPrice.Caption = "Rs. " & BookList.Recordset.Fields("MRP")

        BookQty.Clear
        BookQty.Enabled = False
        cmdAdd.Enabled = False
        Dim q, i As Integer
        q = Val(BookList.Recordset.Fields("QTY"))
        If (q > 0) Then
            For i = 1 To q
                BookQty.AddItem i
            Next i
            BookQty.Enabled = True
            cmdAdd.Enabled = True
            txtQty.ForeColor = &H0&
        Else
            BookQty.AddItem 0
            txtQty.ForeColor = &HC0&
        End If
    End If
End Sub
```

```
Private Sub BookContainer_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    If cmdProceed.Enabled Then cmdProceed.BackColor = &H8000000D
    If cmdBack.Enabled Then cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdProceed.FontItalic = False
    cmdDelete.FontItalic = False
    cmdBack.FontItalic = False
End Sub

Private Sub cmdAdd_Click()
    If Val(BookQty.Text) >= 1 Then
        If Val(BookQty.Text) <= Val(BookList.Recordset.Fields("QTY")) Then
            Select Case MsgBox("Available in stock = " & BookList.Recordset.Fields("QTY") &
vbNewLine & "Remaining in stock = " & CStr(Int(Val(BookList.Recordset.Fields("QTY")))) -
(BookQty.ListIndex + 1)) & vbNewLine & "Add to cart?", vbQuestion + vbApplicationModal +
vbYesNo + vbDefaultButton1, "Confirm add")
                Case vbYes
                    BookList.Recordset.Fields("QTY") =
(Int(Val(BookList.Recordset.Fields("QTY")))) - (BookQty.ListIndex + 1))
                    BookList.Recordset.Update
                    txtQty.Caption = BookList.Recordset.Fields("QTY")
                    MessageBar.Caption = "" & BookList.Recordset.Fields("BOOKNAME") & ""
added to cart!"
                    billCount = billCount + 1
                    BillList.AddItem BookList.Recordset.Fields("BOOKNAME") & " - Rs. " &
BookList.Recordset.Fields("MRP") & " x " & (BookQty.ListIndex + 1)
                    BillList.ItemData(billCount - 1) = Val(BookList.Recordset.Fields("ID"))

                    cmdDelete.Enabled = True
                    cmdProceed.Enabled = True
                    cmdDelete.BackColor = &H8000000D
                    cmdProceed.BackColor = &H8000000D

                    BookQty.Clear
                    BookQty.Enabled = False
                    cmdAdd.Enabled = False
                    Dim q, i As Integer
                    q = Val(BookList.Recordset.Fields("QTY"))
                    If (q > 0) Then
                        For i = 1 To q
                            BookQty.AddItem i
                        Next i
                        BookQty.Enabled = True
                        cmdAdd.Enabled = True
                    Else
                        BookQty.AddItem 0
                        txtQty.ForeColor = &HC0&
                    End If
                End Case
            End If
        End If
    End Sub
```

```
Case vbNo
    MessageBar.Caption = "Cart not updated! Please review cart."
End Select
Else
    MsgBox "The quantity is more than available stock!", vbApplicationModal +
vbOKOnly + vbExclamation, "Please select valid quantity"
End If
Else
    MsgBox "Please select a valid quantity first!", vbApplicationModal + vbOKOnly +
vbExclamation, "Please select quantity"
End If
End Sub

Private Sub cmdAdd_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    cmdAdd.BackColor = &H80&
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    If cmdProceed.Enabled Then cmdProceed.BackColor = &H8000000D
    If cmdBack.Enabled Then cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = True
    cmdProceed.FontItalic = False
    cmdDelete.FontItalic = False
    cmdBack.FontItalic = False
End Sub

Private Sub cmdBack_Click()
    Unload Me
End Sub

Private Sub cmdBack_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    If cmdProceed.Enabled Then cmdProceed.BackColor = &H8000000D
    cmdBack.BackColor = &H80&
    cmdAdd.FontItalic = False
    cmdProceed.FontItalic = False
    cmdDelete.FontItalic = False
    cmdBack.FontItalic = True
End Sub

Private Sub cmdDelete_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    cmdDelete.BackColor = &H80&
    If cmdProceed.Enabled Then cmdProceed.BackColor = &H8000000D
    If cmdBack.Enabled Then cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdProceed.FontItalic = False
    cmdDelete.FontItalic = True
    cmdBack.FontItalic = False
End Sub
```

```
Private Sub cmdProceed_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdProceed.BackColor = &H80&
    If cmdBack.Enabled Then cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdProceed.FontItalic = True
    cmdDelete.FontItalic = False
    cmdBack.FontItalic = False
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    If cmdProceed.Enabled Then cmdProceed.BackColor = &H8000000D
    If cmdBack.Enabled Then cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdProceed.FontItalic = False
    cmdDelete.FontItalic = False
    cmdBack.FontItalic = False
End Sub

Private Sub cmdDelete_Click()
    If BillList.ListCount <= 0 Then
        MsgBox "Your cart is empty. Add items to cart first!", vbApplicationModal +
        vbOKOnly + vbExclamation, "No items to delete"
    Else
        If BillList.ListIndex < 0 Then
            MsgBox "Please select the item you want to delete!", vbApplicationModal +
            vbOKOnly + vbExclamation, "Select item to delete"
        Else
            Dim pos As Integer
            pos = InStrRev(BillList.Text, "x", , vbTextCompare)

            Select Case MsgBox("Are you sure you want to remove '" & Left$(BillList.Text,
            (InStrRev(BillList.Text, "-", , vbTextCompare) - 2)) & "' from your cart?", vbQuestion +
            vbApplicationModal + vbYesNo + vbDefaultButton1, "Confirm delete")
                Case vbYes
                    conn.Execute "UPDATE BookList SET QTY = QTY + " & Mid$(BillList.Text,
                    pos + 2) & " WHERE ID = " & BillList.ItemData(BillList.ListIndex) & ";"
                    BookList.RecordSource = "BookList"
                    BookList.Refresh

                    MessageBar.Caption = "" & Left$(BillList.Text,
                    (InStrRev(BillList.Text, "-", , vbTextCompare) - 2)) & " removed from cart."
                    BillList.RemoveItem BillList.ListIndex
                    billCount = billCount - 1
                    If BillList.ListCount <= 0 Then
                        cmdDelete.Enabled = False
                End Case
            End Select
        End If
    End If
End Sub
```

```
        cmdProceed.Enabled = False
        cmdDelete.BackColor = QBColor(8)
        cmdProceed.BackColor = QBColor(8)
    End If
Case vbNo
    MessageBar.Caption = "Item not removed! Please review cart."
End Select
End If
End If
End Sub

Private Sub cmdProceed_Click()
    Select Case MsgBox("Proceed to payment?", vbApplicationModal + vbYesNo + vbQuestion +
vbDefaultButton1, "Are you sure?")
        Case vbYes
            Dim i As Integer
            Dim amt As Double
            amt = 0
            For i = 0 To (BillList.ListCount - 1)
                PrintView.txtBill.Text = PrintView.txtBill.Text & (i + 1) & vbTab
                If Len(Left$(BillList.List(i), (InStrRev(BillList.List(i), "-", , vbTextCompare) - 2))) > 32 Then
                    PrintView.txtBill.Text = PrintView.txtBill.Text &
Left$(Left$(BillList.List(i), (InStrRev(BillList.List(i), "-", , vbTextCompare) - 2)), 32) & "..." & vbTab & vbTab
                Else
                    PrintView.txtBill.Text = PrintView.txtBill.Text &
Left$(BillList.List(i), (InStrRev(BillList.List(i), "-", , vbTextCompare) - 2)) &
String$(43 - Len(Left$(BillList.List(i), (InStrRev(BillList.List(i), "-", , vbTextCompare) - 2))), " ") & vbTab & vbTab
                End If
                PrintView.txtBill.Text = PrintView.txtBill.Text & Mid$(BillList.List(i),
InStrRev(BillList.List(i), "x", , vbTextCompare) + 2) & vbTab & Mid$(BillList.List(i),
InStrRev(BillList.List(i), ".", , vbTextCompare) + 2, (InStrRev(BillList.List(i), "x", ,
vbTextCompare) - 2) - (InStrRev(BillList.List(i), ".", , vbTextCompare) + 2) + 1) & vbTab &
Val(Mid$(BillList.List(i), InStrRev(BillList.List(i), "x", , vbTextCompare) + 2)) *
Val(Mid$(BillList.List(i), InStrRev(BillList.List(i), ".", , vbTextCompare) + 2,
(InStrRev(BillList.List(i), "x", , vbTextCompare) - 2) - (InStrRev(BillList.List(i), ".", ,
vbTextCompare) + 2) + 1)) & vbNewLine
                amt = amt + (Val(Mid$(BillList.List(i), InStrRev(BillList.List(i), "x", ,
vbTextCompare) + 2)) * Val(Mid$(BillList.List(i), InStrRev(BillList.List(i), ".", ,
vbTextCompare) + 2, (InStrRev(BillList.List(i), "x", , vbTextCompare) - 2) -
(InStrRev(BillList.List(i), ".", , vbTextCompare) + 2) + 1)))
            Next i
            PrintView.OptionLabel.Caption = "Bill for " & billCount & " item(s)"
            PrintView.txtAmt.Caption = "Rs. " & amt
            billCount = 0
            BillList.Clear
            BookQty.Enabled = False
            BookQty.Clear
            cmdAdd.Enabled = False
        End If
    End Select
End Sub
```

```
    cmdDelete.Enabled = False
    cmdProceed.Enabled = False
    cmdDelete.BackColor = QBColor(8)
    cmdProceed.BackColor = QBColor(8)
    Me.Hide
    PrintView.Show
Case vbNo
    Exit Sub
End Select
End Sub

Private Sub Form_Initialize()
    Token = InitGDIPlus
    C = Me.BackColor
    If C < 0 Then C = GetSysColor(C - &H80000000)

    BookList.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path
    & "\db\BookDB.mdb;Persist Security Info=False"
    BookList.CursorLocation = adUseClient
    BookList.CursorType = adOpenDynamic
    BookList.CommandType = adCmdTable
    BookList.RecordSource = "BookList"
    BookList.Refresh
End Sub

Private Sub Form_Load()
    ShopLogo.Picture = LoadPictureGDIPlus(App.Path & "\Images\logo.png", 100, 80, &HADADAD,
True)

    BookList.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path &
"\db\BookDB.mdb;Persist Security Info=False"
    BookList.CursorLocation = adUseClient
    BookList.CursorType = adOpenDynamic
    BookList.CommandType = adCmdTable
    BookList.RecordSource = "BookList"
    BookList.Refresh

    BookFrame.AutoReDraw = True
    BookFrame.ReDraw
    PriceFrame.AutoReDraw = True
    PriceFrame.ReDraw
    OptionsFrame.AutoReDraw = True
    OptionsFrame.ReDraw
    BillFrame.AutoReDraw = True
    BillFrame.ReDraw
    Me.Width = Me.Width + 10

    BookQty.Clear
    BookQty.AddItem 0
    BookQty.Enabled = False
    cmdAdd.Enabled = False
```

```

cmdDelete.Enabled = False
cmdProceed.Enabled = False
cmdDelete.BackColor = QBColor(8)
cmdProceed.BackColor = QBColor(8)
cmdAdd.BackColor = QBColor(8)
MessageBar.Caption = "Please add items to your cart."

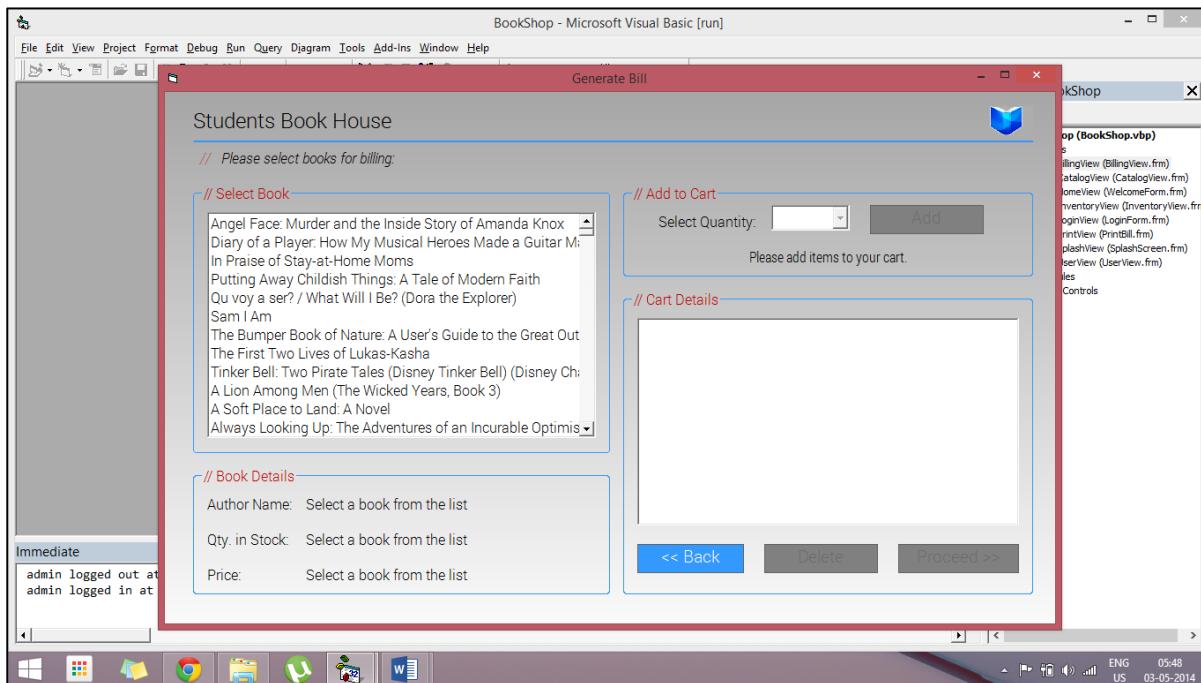
exitVal = vbNo
billCount = 0
End Sub

Private Sub Form_Resize()
    BookFrame.ReDraw
    PriceFrame.ReDraw
    OptionsFrame.ReDraw
    BillFrame.ReDraw
End Sub

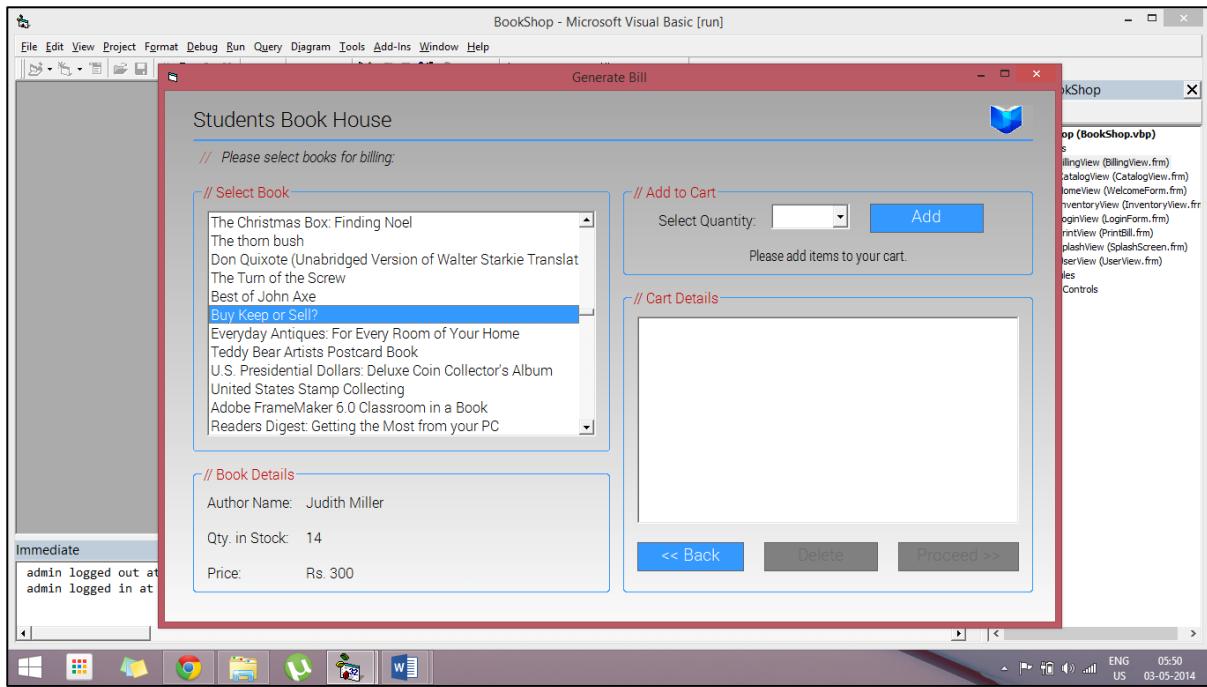
Private Sub Form_Unload(Cancel As Integer)
    exitVal = MsgBox("Go back to main screen?", vbYesNo + vbDefaultButton2 + vbQuestion + vbApplicationModal, "Confirm Exit")
    If exitVal = vbYes Then
        HomeView.Show
        FreeGDIPlus Token
        Unload Me
    Else
        Cancel = 1
    End If
End Sub

```

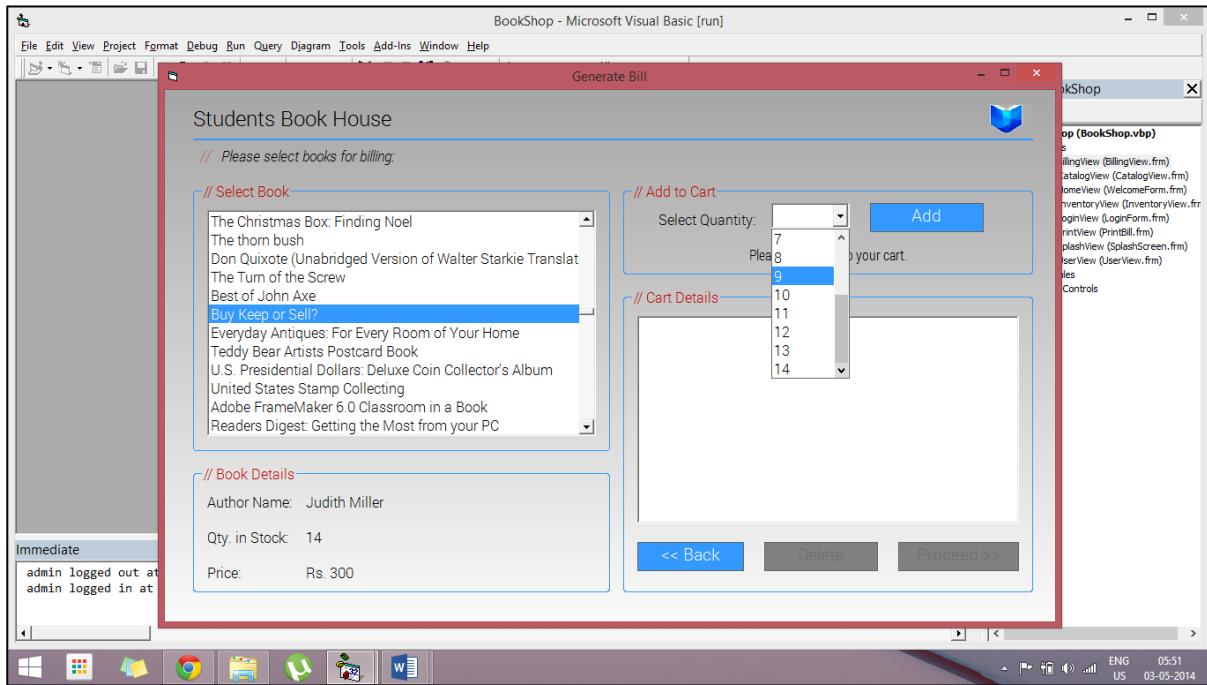
Screenshots



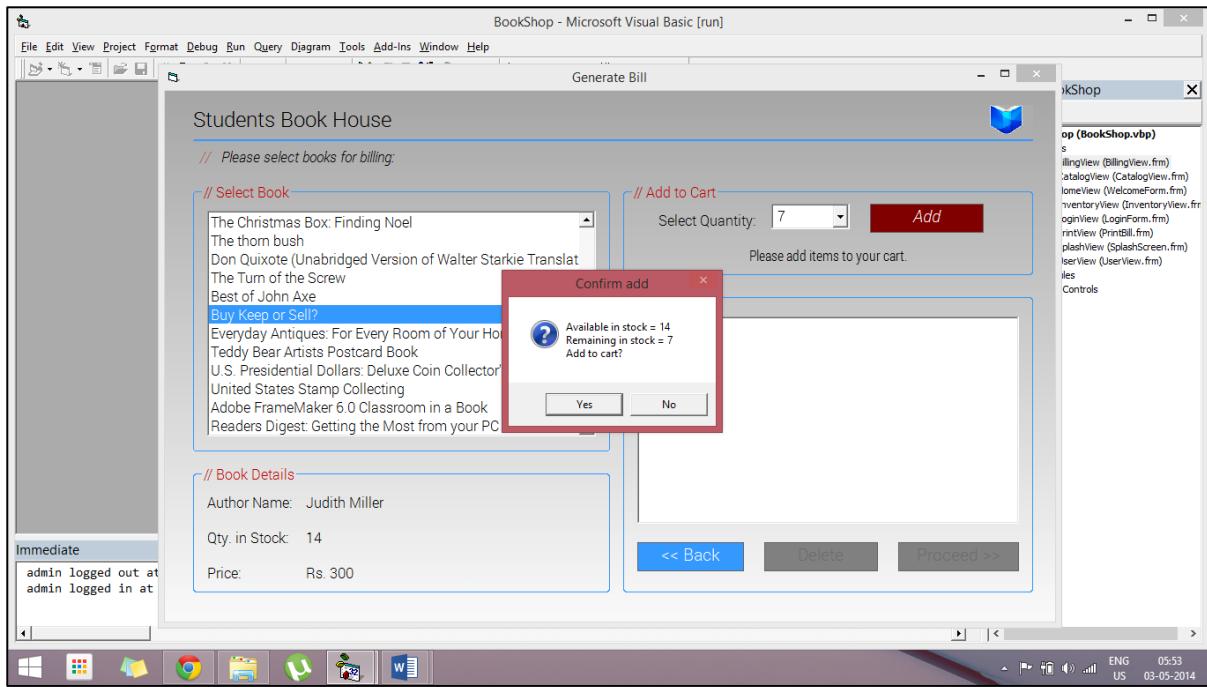
The initial view on opening the Billing Screen



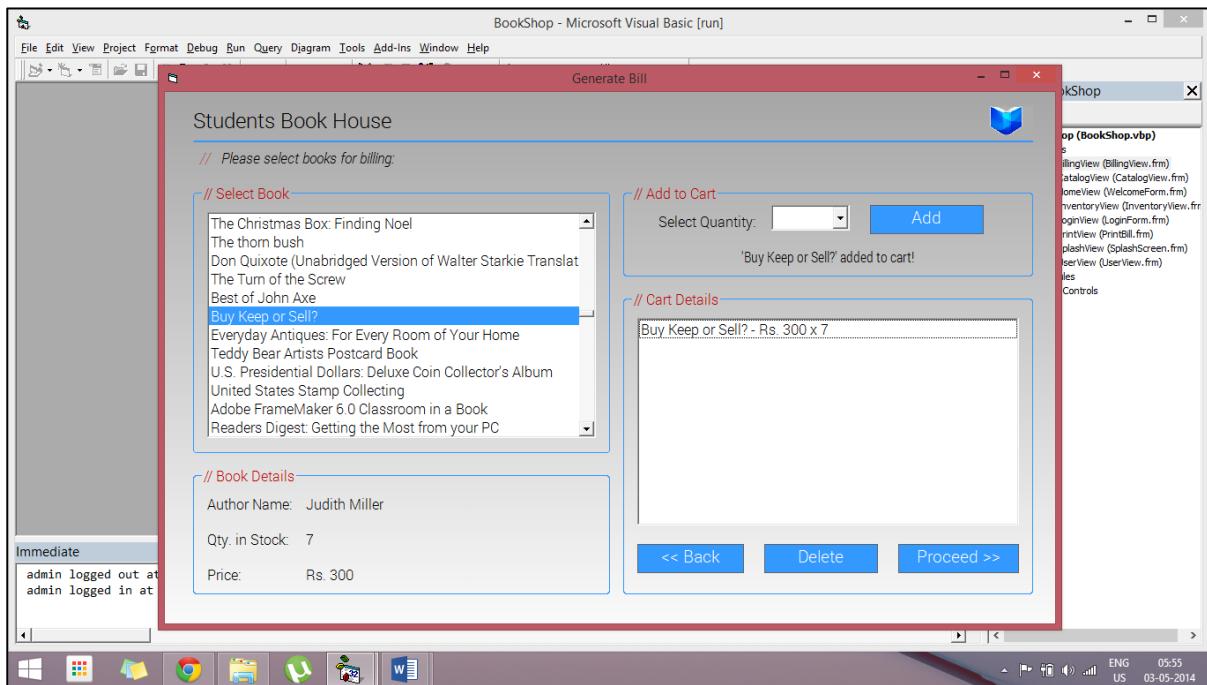
The Billing Screen after selecting a book from the list.



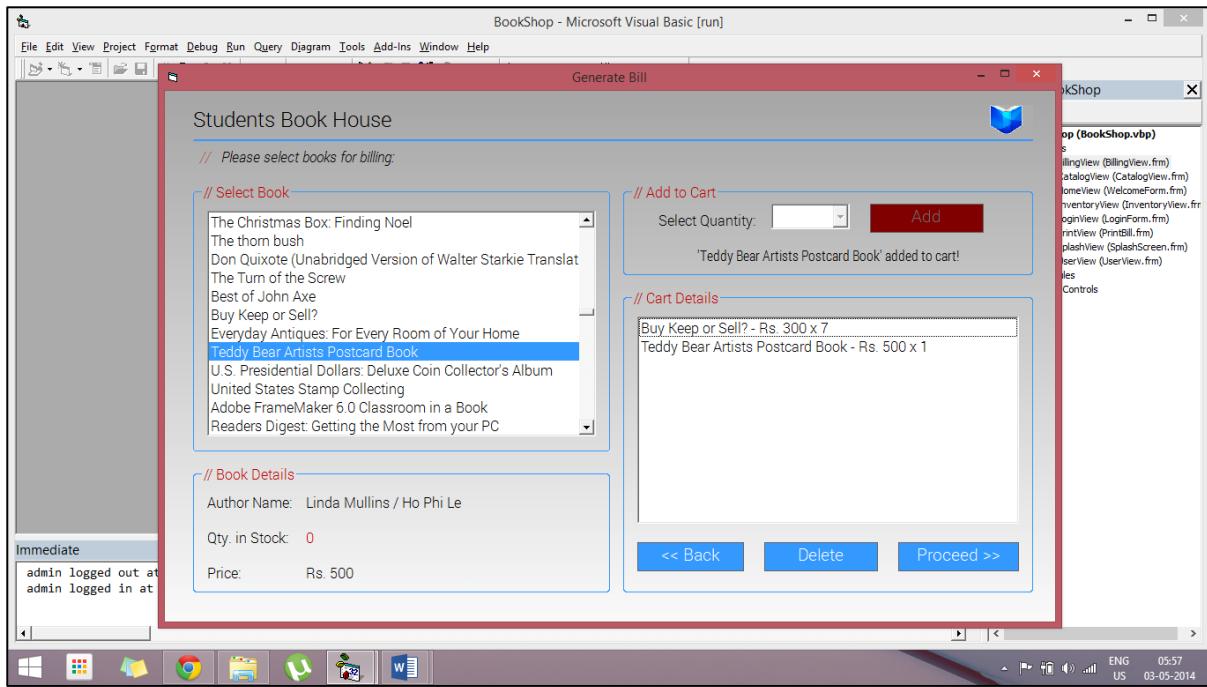
The Billing Screen showing quantity selection before adding to cart.



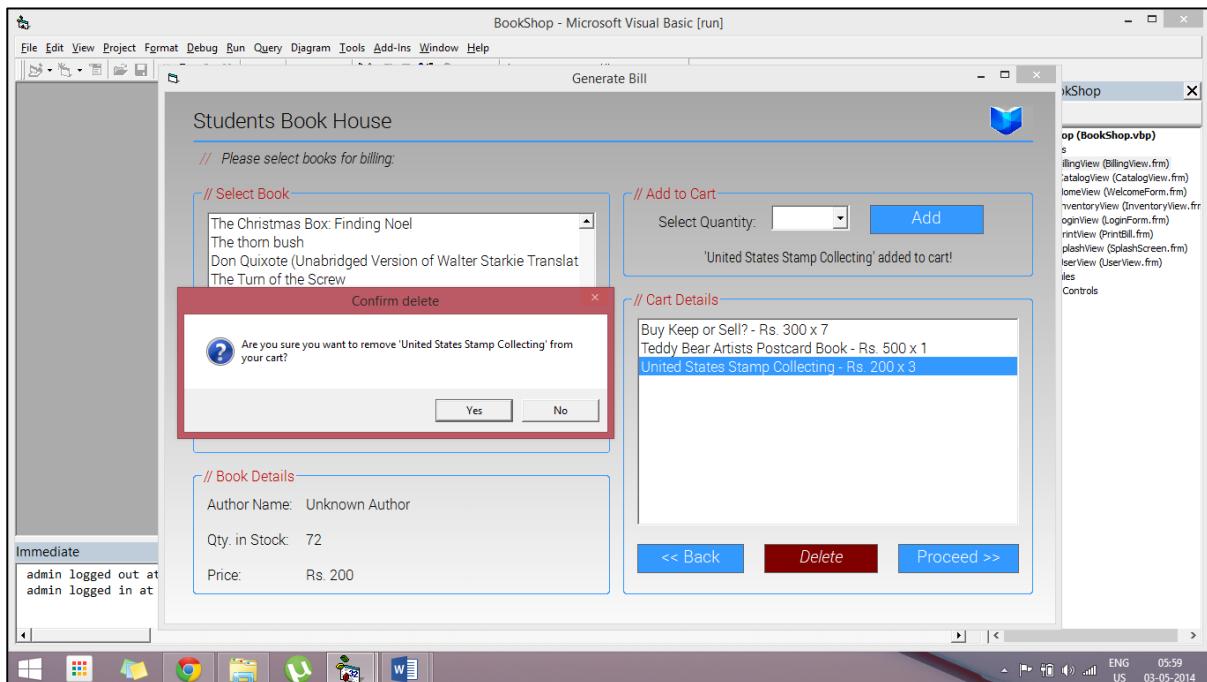
The Billing Screen showing the Add to Cart confirmation.



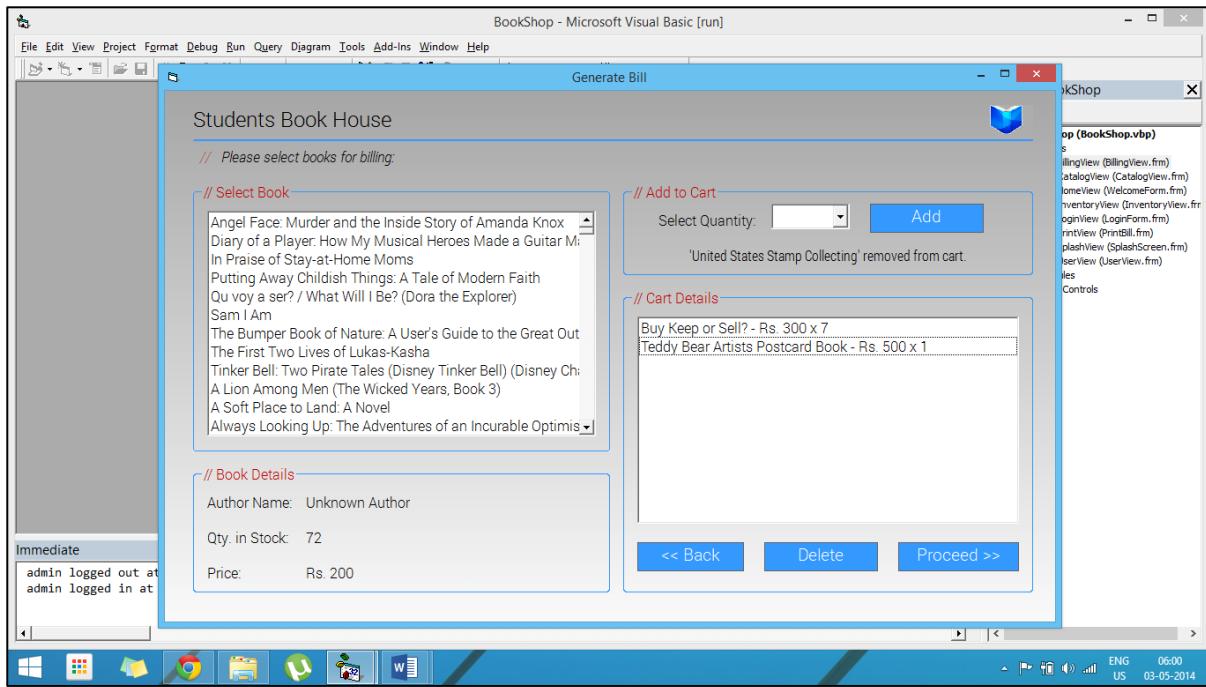
The Billing Screen after successfully adding to cart.



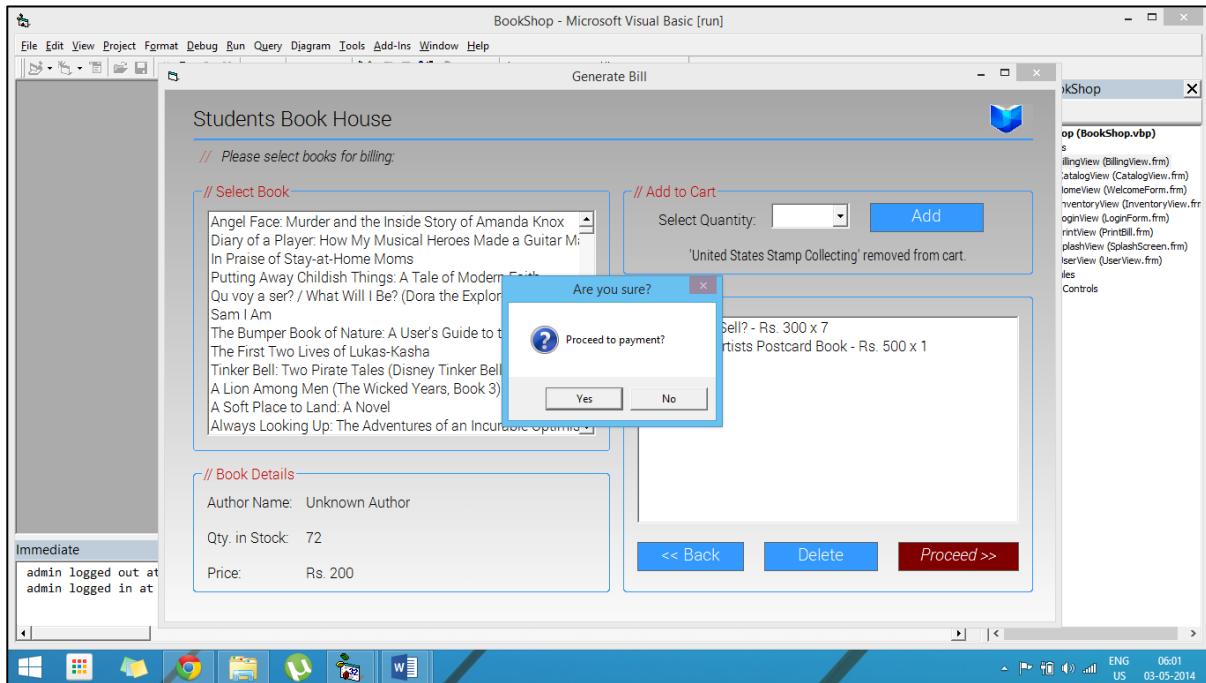
Billing Screen showing item unavailable in stock. The “Add” button becomes unavailable.



Billing Screen showing item removal from cart confirmation.



Billing Screen after item removal from cart.



Billing Screen showing proceed to payment confirmation.

Bill Print Screen (\Forms\PrintView.frm)

Source Code

```
Option Explicit

Dim Token As Long
Dim C As Long

Private Sub cmdPrint_Click()
    cmdPrint.Visible = False
    cmdProceed.Visible = False
    PrintForm
    cmdPrint.Visible = True
    cmdProceed.Visible = True
End Sub

Private Sub cmdProceed_Click()
    Select Case MsgBox("Go back to home screen?", vbApplicationModal + vbYesNo + vbQuestion + vbDefaultButton1, "Sure to proceed?")
        Case vbYes
            Unload Me
            HomeView.Show
        Case vbNo
            Exit Sub
    End Select
End Sub

Private Sub cmdPrint_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    cmdPrint.BackColor = &H80&
    cmdProceed.BackColor = &H8000000D
    cmdPrint.FontItalic = True
    cmdProceed.FontItalic = False
End Sub

Private Sub cmdProceed_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    cmdPrint.BackColor = &H8000000D
    cmdProceed.BackColor = &H80&
    cmdPrint.FontItalic = False
    cmdProceed.FontItalic = True
End Sub

Private Sub Form_Initialize()
    Token = InitGDIPlus
    C = Me.BackColor
    If C < 0 Then C = GetSysColor(C - &H80000000)
End Sub

Private Sub Form_Load()
```

```

ShopLogo.Picture = LoadPictureGDIPlus(App.Path & "\Images\logo.png", 100, 80, &HADADAD,
True)

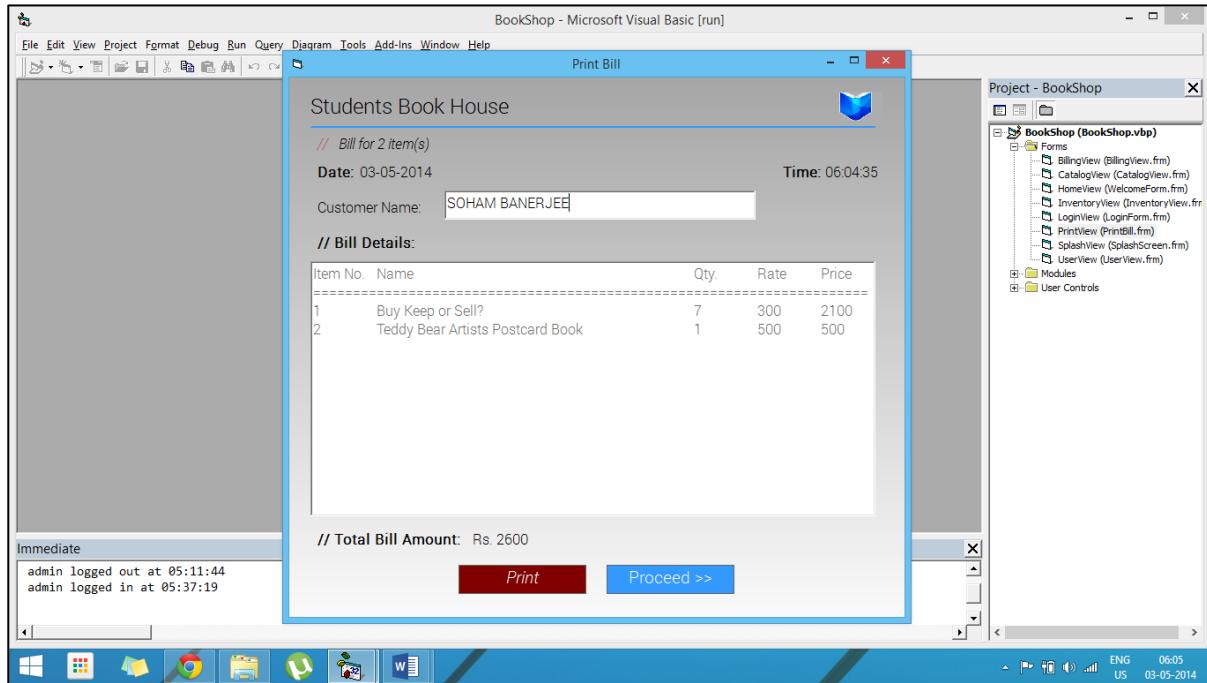
txtBill.Text = "Item No." & vbTab & "Name" & vbTab & vbTab & vbTab & vbTab &
"Qty." & vbTab & "Rate" & vbTab & "Price" & vbNewLine
txtBill.Text = txtBill.Text & String$(70, "=") & vbNewLine
txtDate.Caption = DateValue(Now)
txtTime.Caption = TimeValue(Now)
lblDate.FontBold = True
lblTime.FontBold = True
lblBill.FontBold = True
lblAmt.FontBold = True
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
cmdPrint.BackColor = &H8000000D
cmdProceed.BackColor = &H8000000D
cmdPrint.FontItalic = False
cmdProceed.FontItalic = False
End Sub

Private Sub Form_Unload(Cancel As Integer)
FreeGDIPlus Token
End Sub

```

Screenshots



The Bill Print Screen showing the generated invoice.

Inventory Screen (\Forms\InventoryView.frm)

Source Code

Option Explicit

```
Dim Token As Long
Dim C As Long
Dim LastI As Integer
Dim flagNew As Boolean

Private Sub BookResults_Click()
    BookList.Recordset.AbsolutePosition = BookResults.ListIndex + 1
    txtBookName.Text = BookList.Recordset.Fields("BOOKNAME")
    txtAuthorName.Text = BookList.Recordset.Fields("AUTHORNAME")
    txtISBN.Text = BookList.Recordset.Fields("ISBN")
    txtCategory.Text = BookList.Recordset.Fields("CATEGORY")
    txtQty.Text = BookList.Recordset.Fields("QTY")
    txtPrice.Text = BookList.Recordset.Fields("MRP")
    StatusView.Panels(1).Text = "Book " & (BookResults.ListIndex + 1) & " of " &
    BookResults.ListCount & " selected"
End Sub

Private Sub BookResults_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub

Private Sub cmdAdd_Click()
```

```
txtBookName.Text = ""
txtAuthorName.Text = ""
txtISBN.Text = ""
txtCategory.Text = ""
txtQty.Text = ""
txtPrice.Text = ""

txtBookName.Enabled = True
txtAuthorName.Enabled = True
txtISBN.Enabled = True
txtCategory.Enabled = True
txtQty.Enabled = True
txtPrice.Enabled = True

cmdAdd.Enabled = False
cmdAdd.BackColor = QBColor(8)
cmdEdit.Enabled = False
cmdEdit.BackColor = QBColor(8)
cmdDelete.Enabled = False
cmdDelete.BackColor = QBColor(8)
cmdSave.Enabled = True
cmdSave.BackColor = &H8000000D
cmdCancel.Enabled = True
cmdCancel.BackColor = &H8000000D

flagNew = True
BookList.Recordset.AddNew
txtBookName.SetFocus
End Sub

Private Sub cmdAdd_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If cmdAdd.Enabled Then cmdAdd.BackColor = &H80&
If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
cmdNext.BackColor = &H8000000D
cmdPrev.BackColor = &H8000000D
cmdFirst.BackColor = &H8000000D
cmdLast.BackColor = &H8000000D
cmdBack.BackColor = &H8000000D
cmdFilter.BackColor = &H8000000D
cmdReset.BackColor = &H8000000D
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdAdd.FontItalic = True
cmdEdit.FontItalic = False
cmdSave.FontItalic = False
cmdCancel.FontItalic = False
cmdNext.FontBold = False
cmdPrev.FontBold = False
cmdFirst.FontBold = False
cmdLast.FontBold = False
```

```
cmdBack.FontItalic = False
cmdFilter.FontItalic = False
cmdReset.FontItalic = False
cmdDelete.FontItalic = False
End Sub

Private Sub cmdBack_Click()
    Unload Me
End Sub

Private Sub cmdBack_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H80&
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = True
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub

Private Sub cmdCancel_Click()
    If flagNew Then
        BookList.Recordset.Cancel
        BookList.Recordset.CancelUpdate
        BookResults.ListIndex = 0
        flagNew = False
    End If

    cmdAdd.Enabled = True
    cmdAdd.BackColor = &H8000000D
    cmdEdit.Enabled = True
    cmdEdit.BackColor = &H8000000D
    cmdDelete.Enabled = True
```

```
cmdDelete.BackColor = &H8000000D
cmdSave.Enabled = False
cmdSave.BackColor = QBColor(8)
cmdCancel.Enabled = False
cmdCancel.BackColor = QBColor(8)

txtBookName.Enabled = False
txtAuthorName.Enabled = False
txtISBN.Enabled = False
txtCategory.Enabled = False
txtQty.Enabled = False
txtPrice.Enabled = False

End Sub

Private Sub cmdCancel_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H80&
        cmdNext.BackColor = &H8000000D
        cmdPrev.BackColor = &H8000000D
        cmdFirst.BackColor = &H8000000D
        cmdLast.BackColor = &H8000000D
        cmdBack.BackColor = &H8000000D
        cmdFilter.BackColor = &H8000000D
        cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = True
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub

Private Sub cmdDelete_Click()
    Dim index As Integer
    Select Case MsgBox("Are you sure you want to delete this book from the inventory?", vbApplicationModal + vbYesNo + vbQuestion + vbDefaultButton1, "Confirm delete")
        Case vbYes
            index = BookResults.ListIndex - 1
            BookList.Recordset.Delete
            BookList.Recordset.Update
            BookResults.RemoveItem BookResults.ListIndex
    End Select
End Sub
```

```
If index >= 0 Then
    BookResults.ListIndex = index
Else
    BookResults.ListIndex = 0
End If
End Select
StatusView.Panels(2).Text = BookResults.ListCount & " books in database"
End Sub

Private Sub cmdDelete_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H80&
        cmdAdd.FontItalic = False
        cmdEdit.FontItalic = False
        cmdSave.FontItalic = False
        cmdCancel.FontItalic = False
        cmdNext.FontBold = False
        cmdPrev.FontBold = False
        cmdFirst.FontBold = False
        cmdLast.FontBold = False
        cmdBack.FontItalic = False
        cmdFilter.FontItalic = False
        cmdReset.FontItalic = False
        cmdDelete.FontItalic = True
    End Sub

    Private Sub cmdEdit_Click()
        txtBookName.Enabled = True
        txtAuthorName.Enabled = True
        txtISBN.Enabled = True
        txtCategory.Enabled = True
        txtQty.Enabled = True
        txtPrice.Enabled = True

        cmdAdd.Enabled = False
        cmdAdd.BackColor = QBColor(8)
        cmdEdit.Enabled = False
        cmdEdit.BackColor = QBColor(8)
        cmdDelete.Enabled = False
        cmdDelete.BackColor = QBColor(8)
```

```
cmdSave.Enabled = True
cmdSave.BackColor = &H8000000D
cmdCancel.Enabled = True
cmdCancel.BackColor = &H8000000D

flagNew = False
txtBookName.SetFocus
End Sub

Private Sub cmdEdit_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H80&
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = True
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub

Private Sub cmdFilter_Click()
    If txtSearch.Text = "" Then
        MsgBox "Enter text to search for!", vbApplicationModal + vbOKOnly + vbInformation,
"Enter search term"
        txtSearch.SetFocus
        Exit Sub
    End If

    Dim i As Integer
    For i = LastI To BookResults.ListCount - 1
        If InStr(1, BookResults.List(i), txtSearch.Text, vbTextCompare) > 0 Then
            BookResults.ListIndex = i
            LastI = i
            Exit Sub
    End If
End Sub
```

```
End If
Next i
MsgBox "" & txtSearch.Text & " not found in the records!", vbApplicationModal +
vbOKOnly + vbInformation, "Not found"
End Sub

Private Sub cmdFilter_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H80&
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = True
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub

Private Sub cmdFirst_Click()
    BookResults.ListIndex = 0
End Sub

Private Sub cmdFirst_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H80&
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
```

```
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdAdd.FontItalic = False
cmdEdit.FontItalic = False
cmdSave.FontItalic = False
cmdCancel.FontItalic = False
cmdNext.FontBold = False
cmdPrev.FontBold = False
cmdFirst.FontBold = True
cmdLast.FontBold = False
cmdBack.FontItalic = False
cmdFilter.FontItalic = False
cmdReset.FontItalic = False
cmdDelete.FontItalic = False
End Sub

Private Sub cmdLast_Click()
    BookResults.ListIndex = BookResults.ListCount - 1
End Sub

Private Sub cmdLast_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H80&
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = True
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub

Private Sub cmdNext_Click()
    If BookResults.ListIndex = BookResults.ListCount - 1 Then
        MsgBox "No records after this!", vbApplicationModal + vbOKOnly + vbInformation,
    "End of file reached"

```

```
    Else
        BookResults.ListIndex = BookResults.ListIndex + 1
    End If
End Sub

Private Sub cmdNext_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H80&
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = True
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub

Private Sub cmdPrev_Click()
    If BookResults.ListIndex = 0 Then
        MsgBox "No records before this!", vbApplicationModal + vbOKOnly + vbInformation,
"Start of file reached"
    Else
        BookResults.ListIndex = BookResults.ListIndex - 1
    End If
End Sub

Private Sub cmdPrev_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H80&
    cmdFirst.BackColor = &H8000000D
```

```
cmdLast.BackColor = &H8000000D
cmdBack.BackColor = &H8000000D
cmdFilter.BackColor = &H8000000D
cmdReset.BackColor = &H8000000D
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdAdd.FontItalic = False
cmdEdit.FontItalic = False
cmdSave.FontItalic = False
cmdCancel.FontItalic = False
cmdNext.FontBold = False
cmdPrev.FontBold = True
cmdFirst.FontBold = False
cmdLast.FontBold = False
cmdBack.FontItalic = False
cmdFilter.FontItalic = False
cmdReset.FontItalic = False
cmdDelete.FontItalic = False
End Sub

Private Sub cmdReset_Click()
    LastI = 0
    BookResults.ListIndex = 0
    txtSearch.Text = ""
End Sub

Private Sub cmdReset_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H80&
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = True
    cmdDelete.FontItalic = False
```

End Sub

```
Private Sub cmdSave_Click()
    If txtBookName.Text = "" Then
        MsgBox "Book name cannot be blank!", vbApplicationModal + vbExclamation + vbOKOnly,
    "Error"
        txtBookName.SetFocus
        Exit Sub
    End If
    If txtAuthorName.Text = "" Then
        MsgBox "Author name cannot be blank!", vbApplicationModal + vbExclamation +
    vbOKOnly, "Error"
        txtAuthorName.SetFocus
        Exit Sub
    End If
    If txtISBN.Text = "" Then
        MsgBox "ISBN cannot be blank!", vbApplicationModal + vbExclamation + vbOKOnly,
    "Error"
        txtISBN.SetFocus
        Exit Sub
    End If
    If txtCategory.Text = "" Then
        MsgBox "Category cannot be blank!", vbApplicationModal + vbExclamation + vbOKOnly,
    "Error"
        txtCategory.SetFocus
        Exit Sub
    End If
    If txtQty.Text = "" Then
        MsgBox "Quantity cannot be blank!", vbApplicationModal + vbExclamation + vbOKOnly,
    "Error"
        txtQty.SetFocus
        Exit Sub
    End If
    If txtPrice.Text = "" Then
        MsgBox "Price cannot be blank!", vbApplicationModal + vbExclamation + vbOKOnly,
    "Error"
        txtPrice.SetFocus
        Exit Sub
    End If
    If Not IsNumeric(txtQty.Text) Then
        MsgBox "Quantity must be a number!", vbApplicationModal + vbExclamation + vbOKOnly,
    "Error"
        txtQty.Text = ""
        txtQty.SetFocus
        Exit Sub
    End If
    If Not IsNumeric(txtPrice.Text) Then
        MsgBox "Price must be a number!", vbApplicationModal + vbExclamation + vbOKOnly,
    "Error"
        txtPrice.Text = ""
        txtPrice.SetFocus
```

```
    Exit Sub
End If
BookList.Recordset.Fields("BOOKNAME") = txtBookName.Text
BookList.Recordset.Fields("AUTHORNAME") = txtAuthorName.Text
BookList.Recordset.Fields("ISBN") = txtISBN.Text
BookList.Recordset.Fields("CATEGORY") = txtCategory.Text
BookList.Recordset.Fields("QTY") = txtQty.Text
BookList.Recordset.Fields("MRP") = txtPrice.Text
BookList.Recordset.Update

If flagNew Then
    flagNew = False
    BookResults.AddItem txtBookName.Text
    BookResults.ListIndex = BookResults.ListCount - 1
End If

cmdAdd.Enabled = True
cmdAdd.BackColor = &H8000000D
cmdEdit.Enabled = True
cmdEdit.BackColor = &H8000000D
cmdDelete.Enabled = True
cmdDelete.BackColor = &H8000000D
cmdSave.Enabled = False
cmdSave.BackColor = QBColor(8)
cmdCancel.Enabled = False
cmdCancel.BackColor = QBColor(8)

txtBookName.Enabled = False
txtAuthorName.Enabled = False
txtISBN.Enabled = False
txtCategory.Enabled = False
txtQty.Enabled = False
txtPrice.Enabled = False
StatusView.Panels(2).Text = BookResults.ListCount & " books in database"
End Sub

Private Sub cmdSave_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H80&
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
```

```
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = True
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub

Private Sub Form_Activate()
    StatusView.Panels(2).Text = BookResults.ListCount & " books in database"
End Sub

Private Sub Form_Initialize()
    Token = InitGDIPlus
    C = Me.BackColor
    If C < 0 Then C = GetSysColor(C - &H80000000)

    BookList.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path
    & "\db\BookDB.mdb;Persist Security Info=False"
    BookList.CursorLocation = adUseClient
    BookList.CursorType = adOpenDynamic
    BookList.CommandType = adCmdTable
    BookList.RecordSource = "BookList"
    BookList.Refresh
End Sub

Private Sub Form_Load()
    ShopLogo.Picture = LoadPictureGDIPlus(App.Path & "\Images\logo.png", 100, 80, &HADADAD,
True)

    SearchFrame.AutoReDraw = True
    SearchFrame.ReDraw
    ListFrame.AutoReDraw = True
    DetailsFrame.ReDraw
    DetailsFrame.AutoReDraw = True
    DetailsFrame.ReDraw
    Me.Width = Me.Width + 20

    txtBookName.Enabled = False
    txtAuthorName.Enabled = False
    txtISBN.Enabled = False
    txtCategory.Enabled = False
    txtQty.Enabled = False
    txtPrice.Enabled = False

    cmdAdd.Enabled = True
```

```
cmdAdd.BackColor = &H8000000D
cmdEdit.Enabled = True
cmdEdit.BackColor = &H8000000D
cmdDelete.Enabled = True
cmdDelete.BackColor = &H8000000D
cmdSave.Enabled = False
cmdSave.BackColor = QBColor(8)
cmdCancel.Enabled = False
cmdCancel.BackColor = QBColor(8)

BookList.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path
& "\db\BookDB.mdb;Persist Security Info=False"
BookList.CursorLocation = adUseClient
BookList.CursorType = adOpenDynamic
BookList.CommandType = adCmdTable
BookList.RecordSource = "BookList"
BookList.Refresh

BookList.Recordset.MoveFirst
While Not BookList.Recordset.EOF
    BookResults.AddItem BookList.Recordset.Fields("BOOKNAME")
    BookList.Recordset.MoveNext
Wend

BookResults.ListIndex = 0
BookList.Recordset.AbsolutePosition = BookResults.ListIndex + 1
LastI = 0
flagNew = False
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
```

```
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub

Private Sub Form_Resize()
    SearchFrame.ReDraw
    ListFrame.ReDraw
    DetailsFrame.ReDraw
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Select Case MsgBox("Go back to home screen?", vbApplicationModal + vbYesNo + vbQuestion
+ vbDefaultButton1, "Sure to exit?")
        Case vbYes
            FreeGDIPlus Token
            HomeView.Show
            Unload Me
        Case vbNo
            Cancel = 1
    End Select
End Sub

Private Sub lblAuthorName_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub
```

```
Private Sub lblBookName_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub
```

```
Private Sub lblCategory_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
```

```
cmdBack.FontItalic = False
cmdFilter.FontItalic = False
cmdReset.FontItalic = False
cmdDelete.FontItalic = False
End Sub

Private Sub lblISBN_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub

Private Sub lblPrice_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
```

```
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub

Private Sub lblQty_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub

Private Sub lblSearch_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D

```

```
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdAdd.FontItalic = False
cmdEdit.FontItalic = False
cmdSave.FontItalic = False
cmdCancel.FontItalic = False
cmdNext.FontBold = False
cmdPrev.FontBold = False
cmdFirst.FontBold = False
cmdLast.FontBold = False
cmdBack.FontItalic = False
cmdFilter.FontItalic = False
cmdReset.FontItalic = False
cmdDelete.FontItalic = False
End Sub

Private Sub txtAuthorName_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
cmdNext.BackColor = &H8000000D
cmdPrev.BackColor = &H8000000D
cmdFirst.BackColor = &H8000000D
cmdLast.BackColor = &H8000000D
cmdBack.BackColor = &H8000000D
cmdFilter.BackColor = &H8000000D
cmdReset.BackColor = &H8000000D
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdAdd.FontItalic = False
cmdEdit.FontItalic = False
cmdSave.FontItalic = False
cmdCancel.FontItalic = False
cmdNext.FontBold = False
cmdPrev.FontBold = False
cmdFirst.FontBold = False
cmdLast.FontBold = False
cmdBack.FontItalic = False
cmdFilter.FontItalic = False
cmdReset.FontItalic = False
cmdDelete.FontItalic = False
End Sub

Private Sub txtBookName_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
cmdNext.BackColor = &H8000000D
cmdPrev.BackColor = &H8000000D
```

```
cmdFirst.BackColor = &H8000000D
cmdLast.BackColor = &H8000000D
cmdBack.BackColor = &H8000000D
cmdFilter.BackColor = &H8000000D
cmdReset.BackColor = &H8000000D
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdAdd.FontItalic = False
cmdEdit.FontItalic = False
cmdSave.FontItalic = False
cmdCancel.FontItalic = False
cmdNext.FontBold = False
cmdPrev.FontBold = False
cmdFirst.FontBold = False
cmdLast.FontBold = False
cmdBack.FontItalic = False
cmdFilter.FontItalic = False
cmdReset.FontItalic = False
cmdDelete.FontItalic = False
End Sub

Private Sub txtCategory_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
cmdNext.BackColor = &H8000000D
cmdPrev.BackColor = &H8000000D
cmdFirst.BackColor = &H8000000D
cmdLast.BackColor = &H8000000D
cmdBack.BackColor = &H8000000D
cmdFilter.BackColor = &H8000000D
cmdReset.BackColor = &H8000000D
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdAdd.FontItalic = False
cmdEdit.FontItalic = False
cmdSave.FontItalic = False
cmdCancel.FontItalic = False
cmdNext.FontBold = False
cmdPrev.FontBold = False
cmdFirst.FontBold = False
cmdLast.FontBold = False
cmdBack.FontItalic = False
cmdFilter.FontItalic = False
cmdReset.FontItalic = False
cmdDelete.FontItalic = False
End Sub

Private Sub txtISBN_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
```

```
If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
cmdNext.BackColor = &H8000000D
cmdPrev.BackColor = &H8000000D
cmdFirst.BackColor = &H8000000D
cmdLast.BackColor = &H8000000D
cmdBack.BackColor = &H8000000D
cmdFilter.BackColor = &H8000000D
cmdReset.BackColor = &H8000000D
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdAdd.FontItalic = False
cmdEdit.FontItalic = False
cmdSave.FontItalic = False
cmdCancel.FontItalic = False
cmdNext.FontBold = False
cmdPrev.FontBold = False
cmdFirst.FontBold = False
cmdLast.FontBold = False
cmdBack.FontItalic = False
cmdFilter.FontItalic = False
cmdReset.FontItalic = False
cmdDelete.FontItalic = False
End Sub
```

```
Private Sub txtPrice_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
cmdNext.BackColor = &H8000000D
cmdPrev.BackColor = &H8000000D
cmdFirst.BackColor = &H8000000D
cmdLast.BackColor = &H8000000D
cmdBack.BackColor = &H8000000D
cmdFilter.BackColor = &H8000000D
cmdReset.BackColor = &H8000000D
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdAdd.FontItalic = False
cmdEdit.FontItalic = False
cmdSave.FontItalic = False
cmdCancel.FontItalic = False
cmdNext.FontBold = False
cmdPrev.FontBold = False
cmdFirst.FontBold = False
cmdLast.FontBold = False
cmdBack.FontItalic = False
cmdFilter.FontItalic = False
cmdReset.FontItalic = False
cmdDelete.FontItalic = False
```

End Sub

```
Private Sub txtQty_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
    cmdBack.FontItalic = False
    cmdFilter.FontItalic = False
    cmdReset.FontItalic = False
    cmdDelete.FontItalic = False
End Sub
```

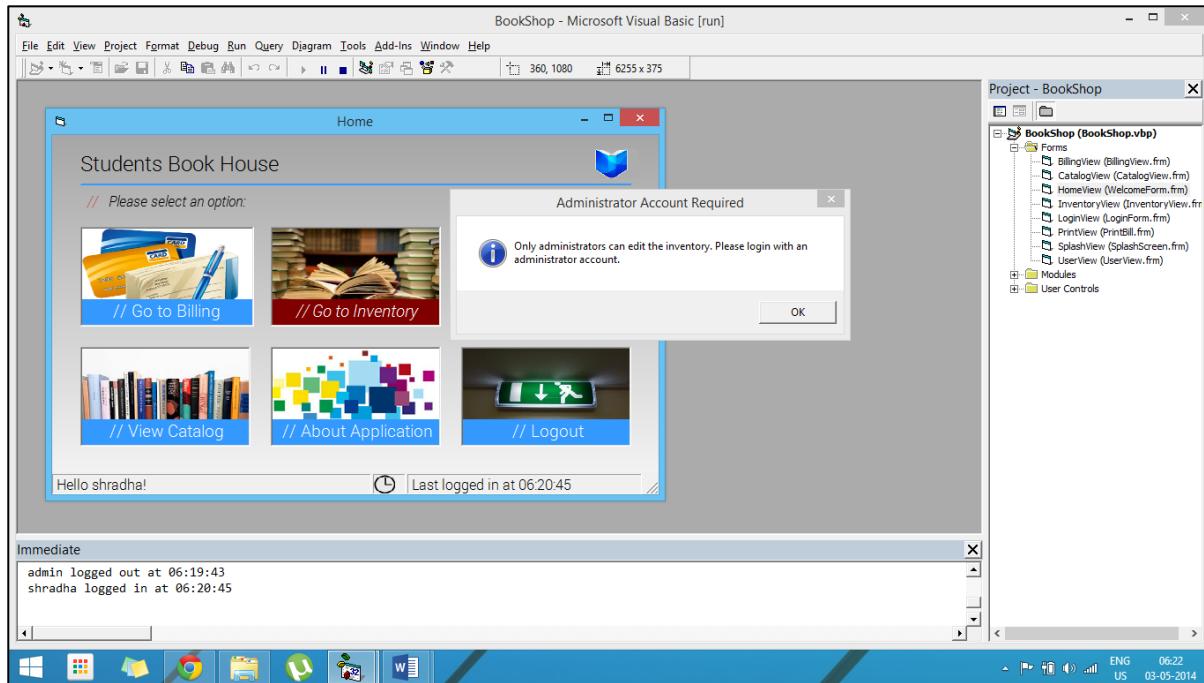
```
Private Sub txtSearch_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdNext.BackColor = &H8000000D
    cmdPrev.BackColor = &H8000000D
    cmdFirst.BackColor = &H8000000D
    cmdLast.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdFilter.BackColor = &H8000000D
    cmdReset.BackColor = &H8000000D
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdNext.FontBold = False
    cmdPrev.FontBold = False
    cmdFirst.FontBold = False
    cmdLast.FontBold = False
```

```

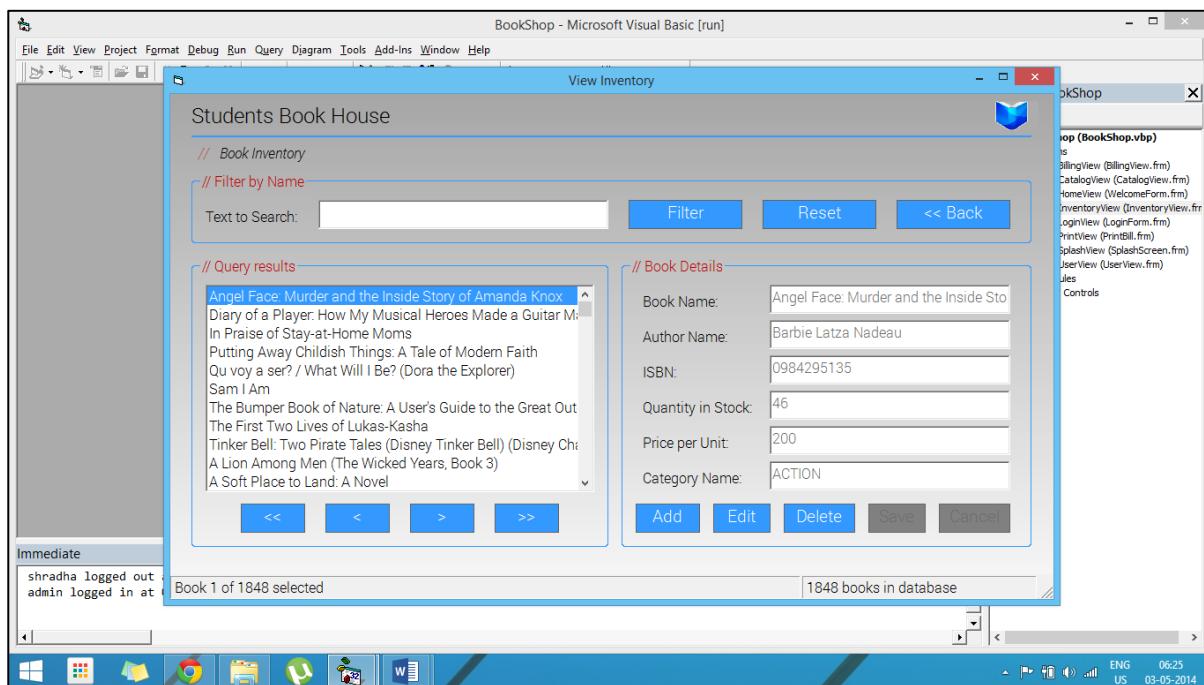
cmdBack.FontItalic = False
cmdFilter.FontItalic = False
cmdReset.FontItalic = False
cmdDelete.FontItalic = False
End Sub

```

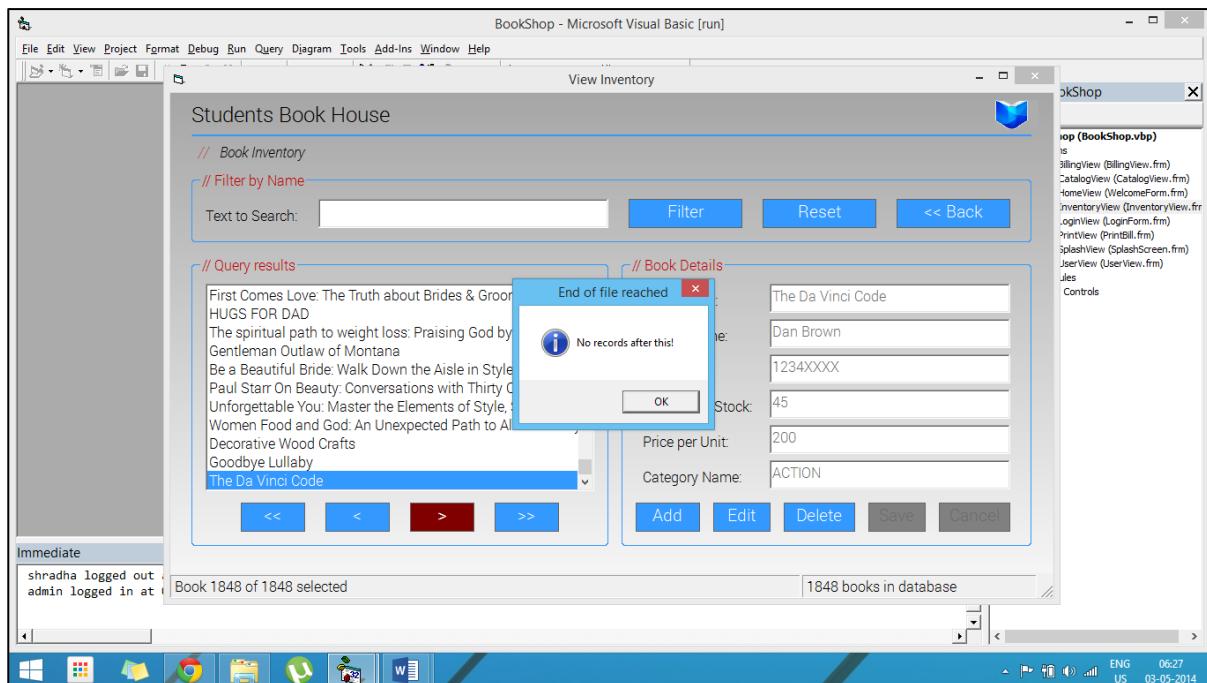
Screenshots



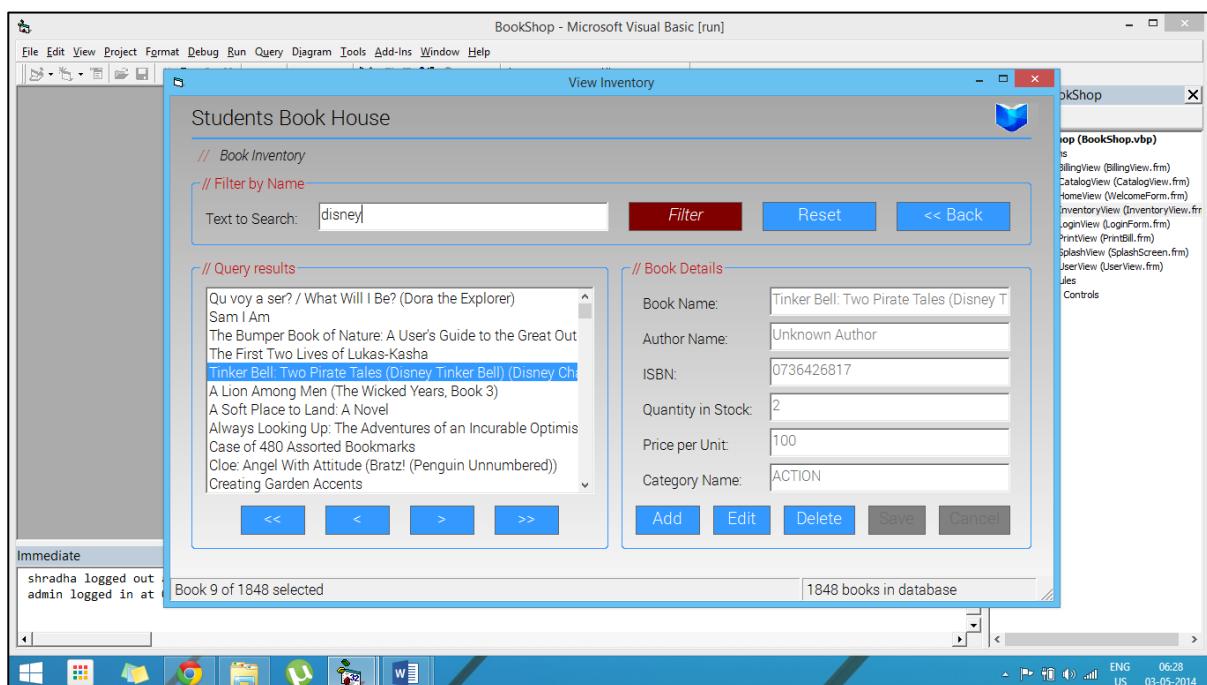
Inventory Screen can only be accessed from an Administrator account.



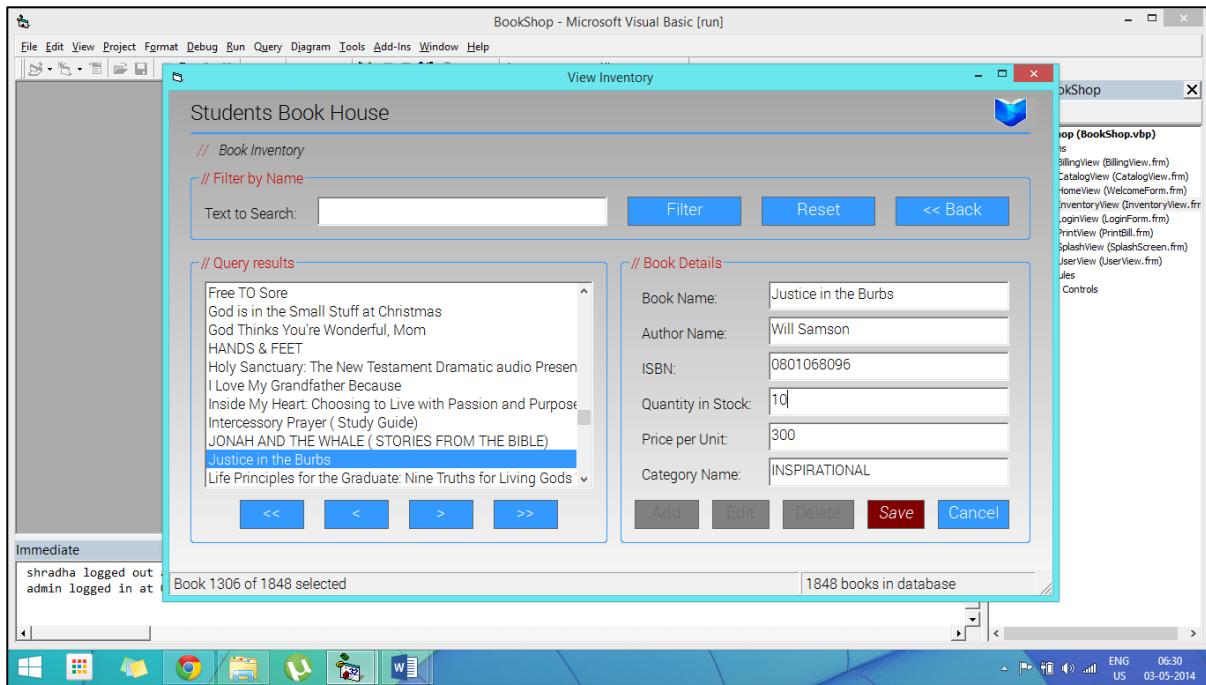
Initial view on opening the Inventory Screen.



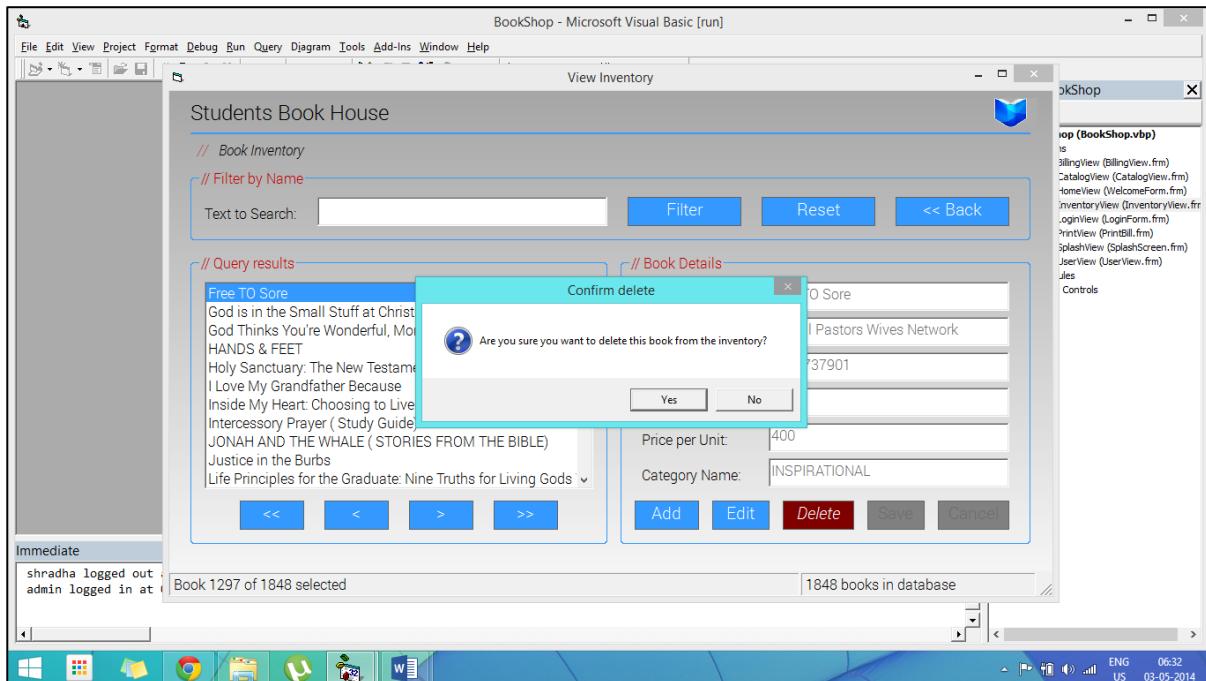
Traversing using the Previous and Next controls.



Search for a particular term in the book name.



Updating existing stocks of a book. Note that the “Add”, “Edit” and “Delete” buttons are disabled while modifying a record.



Book delete confirmation dialog

Catalog Screen (\Forms\CatalogView.frm)

Source Code

```
Option Explicit

Dim Token As Long
Dim C As Long

Private Sub cmdHome_Click()
    Unload Me
End Sub

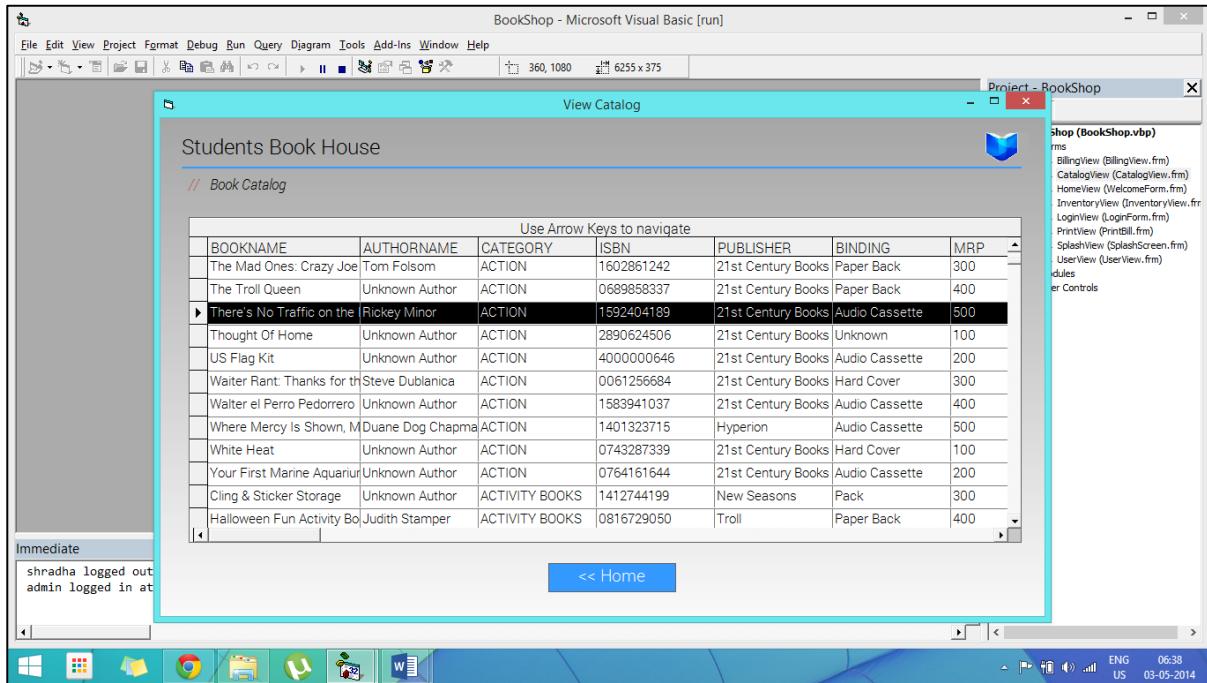
Private Sub Form_Initialize()
    Token = InitGDIPlus
    C = Me.BackColor
    If C < 0 Then C = GetSysColor(C - &H80000000)

    BookList.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path
    & "\db\BookDB.mdb;Persist Security Info=False"
    BookList.CursorLocation = adUseClient
    BookList.CursorType = adOpenDynamic
    BookList.CommandType = adCmdUnknown
    BookList.RecordSource = "Select BOOKNAME, AUTHORNAME, CATEGORY, ISBN, PUBLISHER,
    BINDING, MRP From BookList"
    BookList.Refresh
End Sub

Private Sub Form_Load()
    ShopLogo.Picture = LoadPictureGDIPlus(App.Path & "\Images\logo.png", 100, 80, &HADADAD,
    True)
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Select Case MsgBox("Go back to home screen?", vbApplicationModal + vbYesNo + vbQuestion
    + vbDefaultButton1, "Sure to exit?")
        Case vbYes
            FreeGDIPlus Token
            HomeView.Show
            Unload Me
        Case vbNo
            Cancel = 1
    End Select
End Sub
```

Screenshots



Catalog Screen displaying books and authors in a tabular form.

Users Screen (\Forms\UserView.frm)

Source Code

Option Explicit

```
Dim Token As Long
Dim C As Long
Dim flagNew As Boolean

Private Sub cmdDelete_Click()
    Dim index As Integer
    Select Case MsgBox("Are you sure you want to delete this user?", vbApplicationModal + vbYesNo + vbQuestion + vbDefaultButton1, "Confirm delete")
        Case vbYes
            index = BookResults.ListIndex - 1
            BookList.Recordset.Delete
            BookList.Recordset.Update
            BookResults.RemoveItem BookResults.ListIndex
            If index >= 0 Then
                BookResults.ListIndex = index
            Else
                BookResults.ListIndex = 0
            End If
        End Select
        StatusView.Panels(2).Text = BookResults.ListCount & " users in database"
    End Sub
```

```
Private Sub cmdDelete_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdBack.FontItalic = False
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H80&
        cmdDelete.FontItalic = True
End Sub

Private Sub BookResults_Click()
    BookList.Recordset.AbsolutePosition = BookResults.ListIndex + 1
    txtBookName.Text = BookList.Recordset.Fields("USERNAME")
    txtAuthorName.Text = BookList.Recordset.Fields("PASSWORD")
    txtISBN.Text = BookList.Recordset.Fields("USERTYPE")
    StatusView.Panels(1).Text = "User " & (BookResults.ListIndex + 1) & " of " &
    BookResults.ListCount & " selected"
End Sub

Private Sub BookResults_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdBack.FontItalic = False
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
        cmdDelete.FontItalic = False
End Sub

Private Sub cmdAdd_Click()
    txtBookName.Text = ""
    txtAuthorName.Text = ""
    txtISBN.Text = ""

    txtBookName.Enabled = True
    txtAuthorName.Enabled = True
    txtISBN.Enabled = True

    cmdAdd.Enabled = False

```

```
cmdAdd.BackColor = QBColor(8)
cmdEdit.Enabled = False
cmdEdit.BackColor = QBColor(8)
cmdDelete.Enabled = False
cmdDelete.BackColor = QBColor(8)
cmdSave.Enabled = True
cmdSave.BackColor = &H8000000D
cmdCancel.Enabled = True
cmdCancel.BackColor = &H8000000D

flagNew = True
BookList.Recordset.AddNew
txtBookName.SetFocus
End Sub

Private Sub cmdAdd_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If cmdAdd.Enabled Then cmdAdd.BackColor = &H80&
If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
cmdBack.BackColor = &H8000000D
cmdAdd.FontItalic = True
cmdEdit.FontItalic = False
cmdSave.FontItalic = False
cmdCancel.FontItalic = False
cmdBack.FontItalic = False
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdDelete.FontItalic = False
End Sub

Private Sub cmdBack_Click()
Unload Me
End Sub

Private Sub cmdBack_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
cmdBack.BackColor = &H80&
cmdAdd.FontItalic = False
cmdEdit.FontItalic = False
cmdSave.FontItalic = False
cmdCancel.FontItalic = False
cmdBack.FontItalic = True
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdDelete.FontItalic = False
End Sub

Private Sub cmdCancel_Click()
```

```
If flagNew Then
    BookList.Recordset.Cancel
    BookList.Recordset.CancelUpdate
    BookResults.ListIndex = 0
    flagNew = False
End If

cmdAdd.Enabled = True
cmdAdd.BackColor = &H8000000D
cmdEdit.Enabled = True
cmdEdit.BackColor = &H8000000D
cmdDelete.Enabled = True
cmdDelete.BackColor = &H8000000D
cmdSave.Enabled = False
cmdSave.BackColor = QBColor(8)
cmdCancel.Enabled = False
cmdCancel.BackColor = QBColor(8)

txtBookName.Enabled = False
txtAuthorName.Enabled = False
txtISBN.Enabled = False
End Sub

Private Sub cmdCancel_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H80&
        cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = True
    cmdBack.FontItalic = False
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdDelete.FontItalic = False
End Sub

Private Sub cmdEdit_Click()
    txtBookName.Enabled = True
    txtAuthorName.Enabled = True
    txtISBN.Enabled = True

    cmdAdd.Enabled = False
    cmdAdd.BackColor = QBColor(8)
    cmdEdit.Enabled = False
    cmdEdit.BackColor = QBColor(8)
    cmdDelete.Enabled = False
    cmdDelete.BackColor = QBColor(8)
    cmdSave.Enabled = True
```

```
cmdSave.BackColor = &H8000000D
cmdCancel.Enabled = True
cmdCancel.BackColor = &H8000000D

flagNew = False
txtBookName.SetFocus
End Sub

Private Sub cmdEdit_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H80&
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = True
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdBack.FontItalic = False
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdDelete.FontItalic = False
End Sub

Private Sub cmdSave_Click()
    If txtBookName.Text = "" Then
        MsgBox "Username cannot be blank!", vbApplicationModal + vbExclamation + vbOKOnly,
    "Error"
        txtBookName.SetFocus
        Exit Sub
    End If
    If txtAuthorName.Text = "" Then
        MsgBox "Password cannot be blank!", vbApplicationModal + vbExclamation + vbOKOnly,
    "Error"
        txtAuthorName.SetFocus
        Exit Sub
    End If
    txtISBN.Text = UCase(txtISBN.Text)
    If txtISBN.Text <> "ADMIN" Then txtISBN.Text = "USER"
    BookList.Recordset.Fields("USERNAME") = txtBookName.Text
    BookList.Recordset.Fields("PASSWORD") = txtAuthorName.Text
    BookList.Recordset.Fields("USERTYPE") = txtISBN.Text
    BookList.Recordset.Update

    If flagNew Then
        flagNew = False
        BookResults.AddItem txtBookName.Text
        BookResults.ListIndex = BookResults.ListCount - 1
    End If

    cmdAdd.Enabled = True
```

```
cmdAdd.BackColor = &H8000000D
cmdEdit.Enabled = True
cmdEdit.BackColor = &H8000000D
cmdDelete.Enabled = True
cmdDelete.BackColor = &H8000000D
cmdSave.Enabled = False
cmdSave.BackColor = QBColor(8)
cmdCancel.Enabled = False
cmdCancel.BackColor = QBColor(8)

txtBookName.Enabled = False
txtAuthorName.Enabled = False
txtISBN.Enabled = False
StatusView.Panels(2).Text = BookResults.ListCount & " users in database"
End Sub

Private Sub cmdSave_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H80&
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = True
    cmdCancel.FontItalic = False
    cmdBack.FontItalic = False
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdDelete.FontItalic = False
End Sub

Private Sub Form_Activate()
    StatusView.Panels(2).Text = BookResults.ListCount & " users in database"
End Sub

Private Sub Form_Initialize()
    Token = InitGDIPlus
    C = Me.BackColor
    If C < 0 Then C = GetSysColor(C - &H80000000)

    BookList.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path
    & "\db\BookDB.mdb;Persist Security Info=False"
    BookList.CursorLocation = adUseClient
    BookList.CursorType = adOpenDynamic
    BookList.CommandType = adCmdTable
    BookList.RecordSource = "Users"
    BookList.Refresh
End Sub

Private Sub Form_Load()
```

```
ShopLogo.Picture = LoadPictureGDIPlus(App.Path & "\Images\logo.png", 100, 80, &HADADAD,  
True)

SearchFrame.AutoReDraw = True
SearchFrame.ReDraw
ListFrame.AutoReDraw = True
DetailsFrame.ReDraw
DetailsFrame.AutoReDraw = True
DetailsFrame.ReDraw
Me.Width = Me.Width + 20

txtBookName.Enabled = False
txtAuthorName.Enabled = False
txtISBN.Enabled = False

cmdAdd.Enabled = True
cmdAdd.BackColor = &H8000000D
cmdEdit.Enabled = True
cmdEdit.BackColor = &H8000000D
cmdDelete.Enabled = True
cmdDelete.BackColor = &H8000000D
cmdSave.Enabled = False
cmdSave.BackColor = QBColor(8)
cmdCancel.Enabled = False
cmdCancel.BackColor = QBColor(8)

BookList.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path  
& "\db\BookDB.mdb;Persist Security Info=False"
BookList.CursorLocation = adUseClient
BookList.CursorType = adOpenDynamic
BookList.CommandType = adCmdTable
BookList.RecordSource = "Users"
BookList.Refresh

BookList.Recordset.MoveFirst
While Not BookList.Recordset.EOF
    BookResults.AddItem BookList.Recordset.Fields("USERNAME")
    BookList.Recordset.MoveNext
Wend

BookResults.ListIndex = 0
BookList.Recordset.AbsolutePosition = BookResults.ListIndex + 1
flagNew = False
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
```

```
cmdAdd.FontItalic = False
cmdEdit.FontItalic = False
cmdSave.FontItalic = False
cmdCancel.FontItalic = False
cmdBack.FontItalic = False
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdDelete.FontItalic = False
End Sub

Private Sub Form_Resize()
    SearchFrame.ReDraw
    ListFrame.ReDraw
    DetailsFrame.ReDraw
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Select Case MsgBox("Go back to home screen?", vbApplicationModal + vbYesNo + vbQuestion
+ vbDefaultButton1, "Sure to exit?")
        Case vbYes
            FreeGDIPlus Token
            HomeView.Show
            Unload Me
        Case vbNo
            Cancel = 1
    End Select
End Sub

Private Sub lblAuthorName_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdBack.FontItalic = False
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdDelete.FontItalic = False
End Sub

Private Sub lblBookName_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
```

```
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdBack.FontItalic = False
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdDelete.FontItalic = False
End Sub

Private Sub lblISBN_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdBack.FontItalic = False
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdDelete.FontItalic = False
End Sub

Private Sub txtAuthorName_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdBack.FontItalic = False
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdDelete.FontItalic = False
End Sub

Private Sub txtBookName_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
End Sub
```

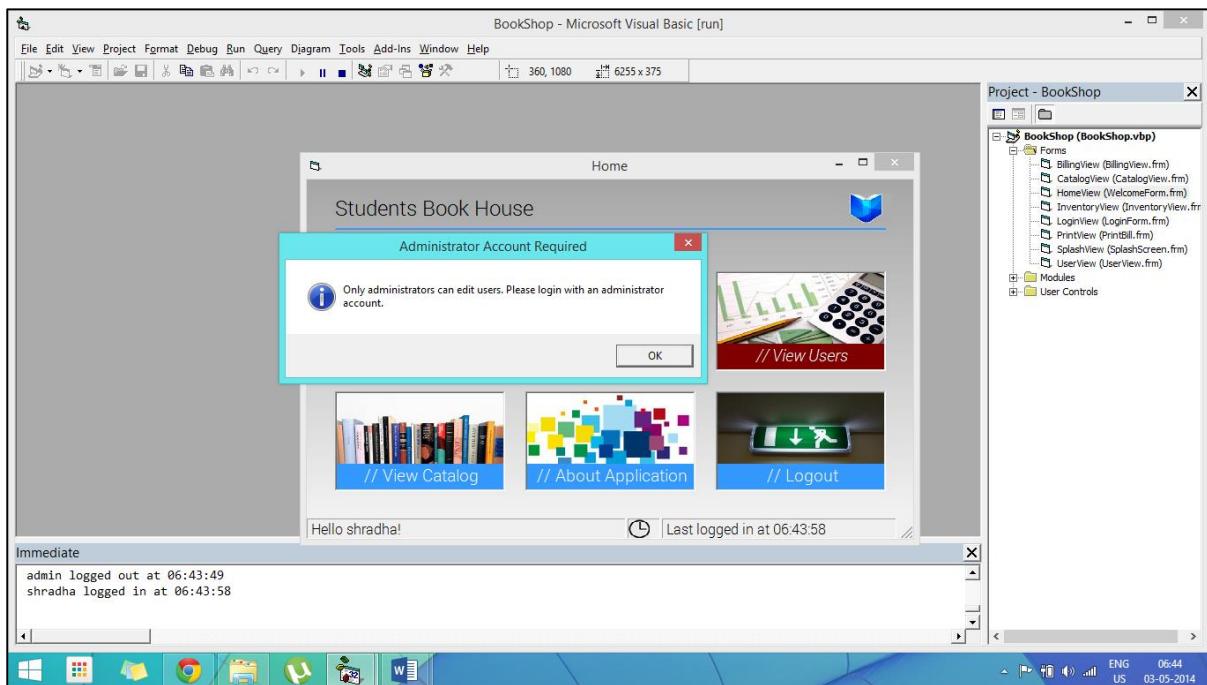
```

cmdBack.FontItalic = False
If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
cmdDelete.FontItalic = False
End Sub

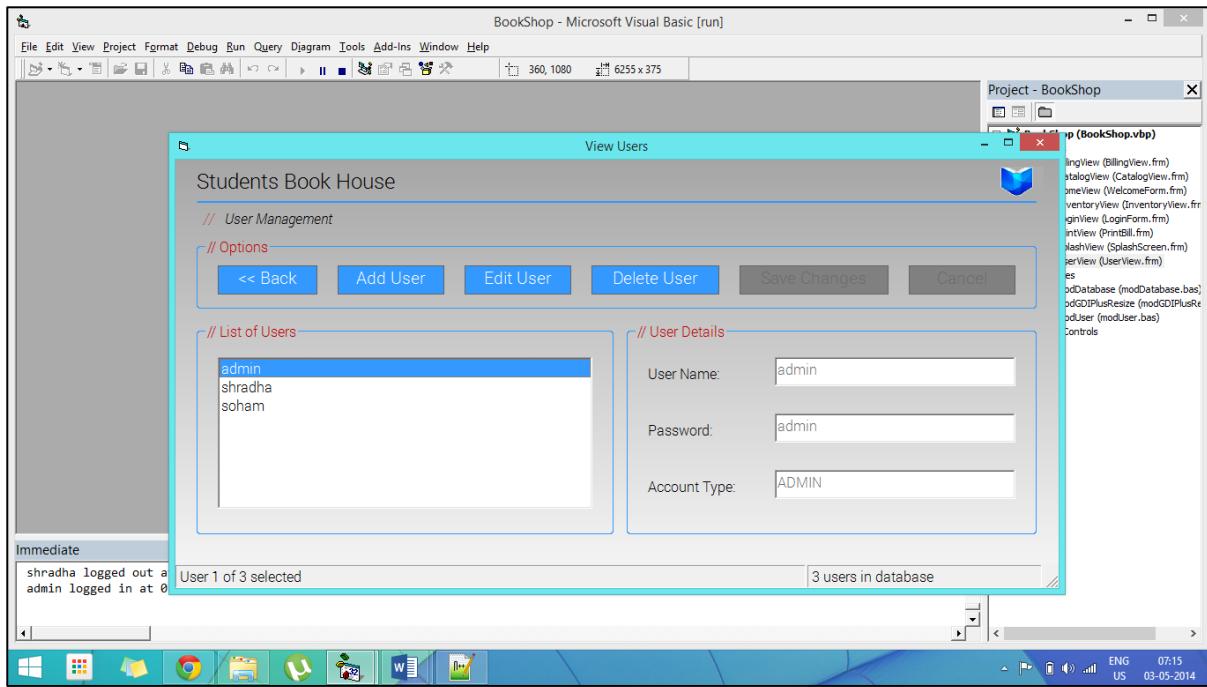
Private Sub txtISBN_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    If cmdAdd.Enabled Then cmdAdd.BackColor = &H8000000D
    If cmdEdit.Enabled Then cmdEdit.BackColor = &H8000000D
    If cmdSave.Enabled Then cmdSave.BackColor = &H8000000D
    If cmdCancel.Enabled Then cmdCancel.BackColor = &H8000000D
    cmdBack.BackColor = &H8000000D
    cmdAdd.FontItalic = False
    cmdEdit.FontItalic = False
    cmdSave.FontItalic = False
    cmdCancel.FontItalic = False
    cmdBack.FontItalic = False
    If cmdDelete.Enabled Then cmdDelete.BackColor = &H8000000D
    cmdDelete.FontItalic = False
End Sub

```

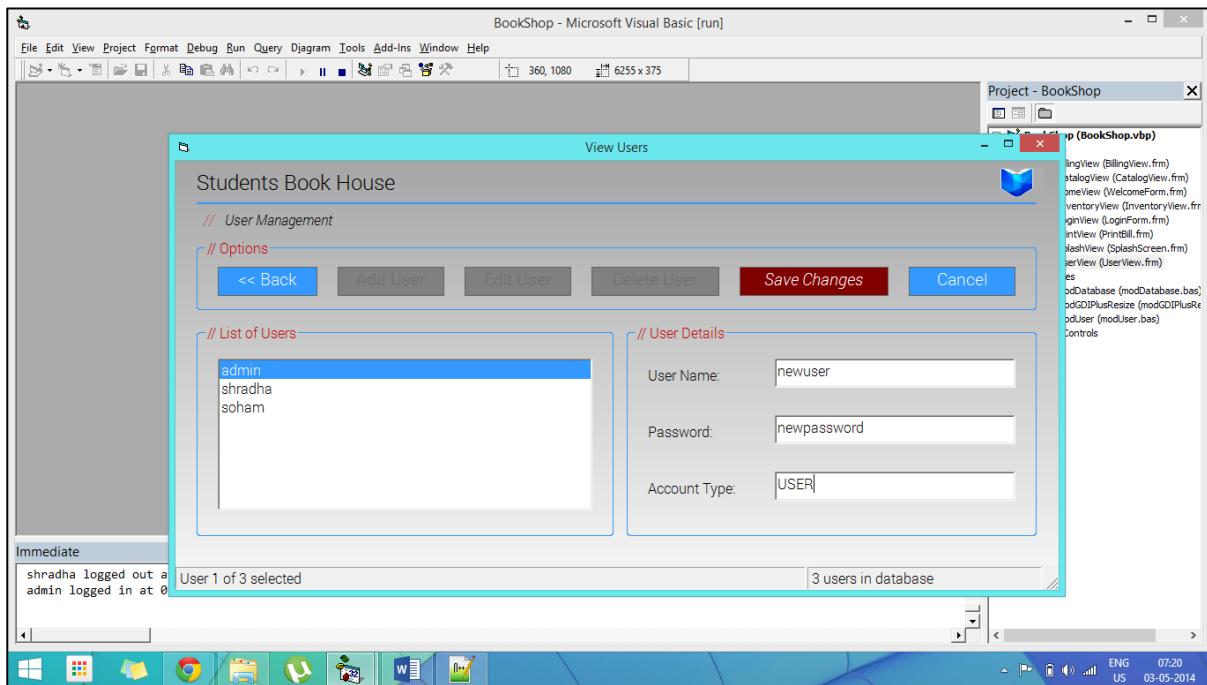
Screenshots



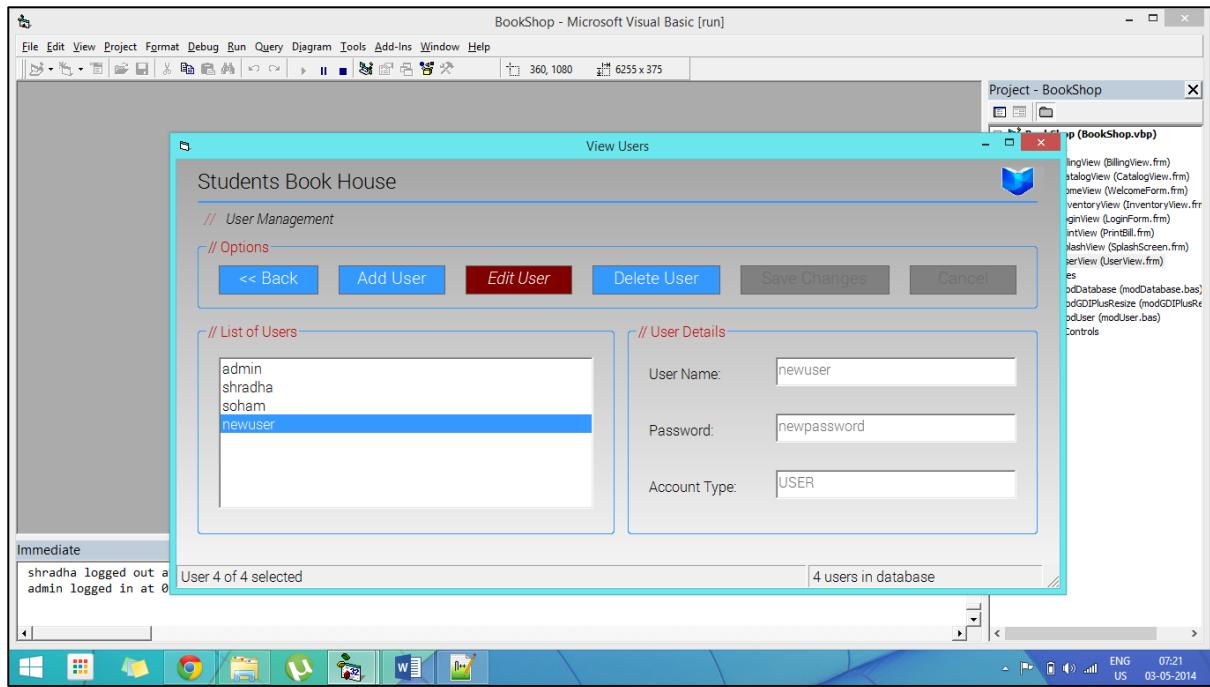
Access rights management being enforced when trying to access the Users Screen from a non-Administrator account.



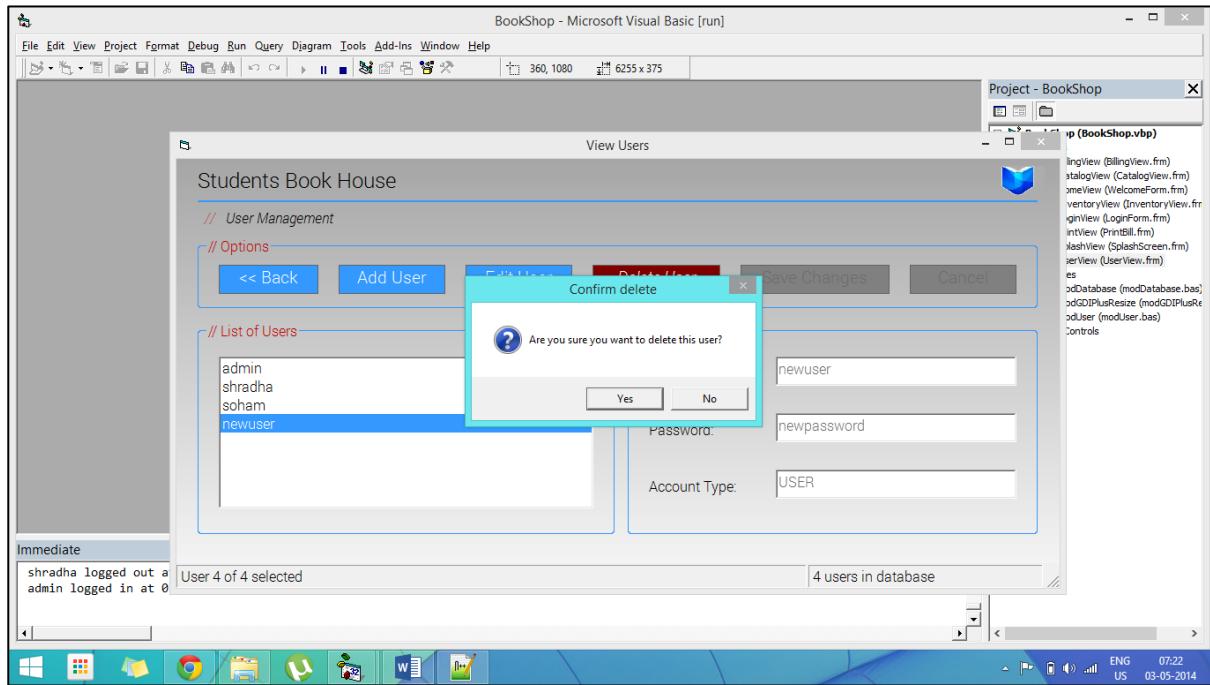
The initial view on opening the Users Screen



Add a new user to the database



Users Screen after adding new user



Delete user confirmation dialog

Modules

modGDIplusResize (\lib\modGDIplusResize.bas)

Description

It uses the Win32 API to offload graphics processing onto the Graphics Driver Interface library. It is needed for the Active Resizing of elements whenever a form is scaled. It also supports loading of PNGs and GIFs in a PictureBox with full alpha transparency layer. Together with the OpenFrameset control, this is used to create most of the transparencies and gradients that give the project its look and feel.

Source Code

Option Explicit

```
Public Declare Function GetSysColor Lib "user32" (ByVal nIndex As Long) As Long
```

```
Private Type GUID
```

```
    Data1 As Long
    Data2 As Integer
    Data3 As Integer
    Data4(7) As Byte
```

```
End Type
```

```
Private Type PICTDESC
```

```
    size As Long
    Type As Long
    hBmp As Long
    hPal As Long
    Reserved As Long
```

```
End Type
```

```
Private Type GdiplusStartupInput
```

```
    GdiplusVersion As Long
    DebugEventCallback As Long
    SuppressBackgroundThread As Long
    SuppressExternalCodecs As Long
```

```
End Type
```

```
Private Type PWMFRect16
```

```
    left As Integer
    top As Integer
    Right As Integer
    Bottom As Integer
```

```
End Type
```

```
Private Type wmfPlaceableFileHeader
```

```
    Key As Long
    hMF As Integer
    BoundingBox As PWMFRect16
    Inch As Integer
    Reserved As Long
    CheckSum As Integer
```

End Type

```
' GDI Functions
Private Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hDC As Long) As Long
Private Declare Function OleCreatePictureIndirect Lib "olepro32.dll" (PicDesc As PICTDESC,
RefIID As GUID, ByVal fPictureOwnsHandle As Long, IPic As IPicture) As Long
Private Declare Function CreateCompatibleBitmap Lib "gdi32" (ByVal hDC As Long, ByVal
nWidth As Long, ByVal nHeight As Long) As Long
Private Declare Function GetDeviceCaps Lib "gdi32" (ByVal hDC As Long, ByVal nIndex As
Long) As Long
Private Declare Function PatBlt Lib "gdi32" (ByVal hDC As Long, ByVal x As Long, ByVal y As
Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal dwRop As Long) As Long
Private Declare Function CreateBitmap Lib "gdi32" (ByVal nWidth As Long, ByVal nHeight As
Long, ByVal nPlanes As Long, ByVal nBitCount As Long, lpBits As Any) As Long
Private Declare Function SelectObject Lib "gdi32" (ByVal hDC As Long, ByVal hObject As
Long) As Long
Private Declare Function CreateSolidBrush Lib "gdi32" (ByVal crColor As Long) As Long
Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long
Private Declare Function DeleteDC Lib "gdi32" (ByVal hDC As Long) As Long

' GDI+ functions
Private Declare Function GdipLoadImageFromFile Lib "gdiplus.dll" (ByVal FileName As Long,
GpImage As Long) As Long
Private Declare Function GdipplusStartup Lib "gdiplus.dll" (Token As Long, gdipInput As
GdiplusStartupInput, GdiplusStartupOutput As Long) As Long
Private Declare Function GdipCreateFromHDC Lib "gdiplus.dll" (ByVal hDC As Long, GpGraphics
As Long) As Long
Private Declare Function GdipSetInterpolationMode Lib "gdiplus.dll" (ByVal Graphics As
Long, ByVal InterMode As Long) As Long
Private Declare Function GdipDrawImageRectI Lib "gdiplus.dll" (ByVal Graphics As Long,
ByVal Img As Long, ByVal x As Long, ByVal y As Long, ByVal Width As Long, ByVal Height As
Long) As Long
Private Declare Function GdipDeleteGraphics Lib "gdiplus.dll" (ByVal Graphics As Long) As
Long
Private Declare Function GdipDisposeImage Lib "gdiplus.dll" (ByVal Image As Long) As Long
Private Declare Function GdipCreateBitmapFromHBITMAP Lib "gdiplus.dll" (ByVal hBmp As Long,
ByVal hPal As Long, GpBitmap As Long) As Long
Private Declare Function GdipGetImageWidth Lib "gdiplus.dll" (ByVal Image As Long, Width As
Long) As Long
Private Declare Function GdipGetImageHeight Lib "gdiplus.dll" (ByVal Image As Long, Height
As Long) As Long
Private Declare Function GdipCreateMetafileFromWmf Lib "gdiplus.dll" (ByVal hWmf As Long,
ByVal deleteWmf As Long, WmfHeader As wmfPlaceableFileHeader, Metafile As Long) As Long
Private Declare Function GdipCreateMetafileFromEmf Lib "gdiplus.dll" (ByVal hEmf As Long,
ByVal deleteEmf As Long, Metafile As Long) As Long
Private Declare Function GdipCreateBitmapFromHICON Lib "gdiplus.dll" (ByVal hIcon As Long,
GpBitmap As Long) As Long
Private Declare Function GdipDrawImageRectRectI Lib "gdiplus.dll" (ByVal Graphics As Long,
ByVal GpImage As Long, ByVal dstx As Long, ByVal dsty As Long, ByVal dstwidth As Long,
ByVal dstheight As Long, ByVal srcx As Long, ByVal srcy As Long, ByVal srcwidth As Long,
```

```
ByVal srcheight As Long, ByVal srcUnit As Long, ByVal imageAttributes As Long, ByVal
callback As Long, ByVal callbackData As Long) As Long
Private Declare Sub GdipplusShutdown Lib "gdipplus.dll" (ByVal Token As Long)

' GDI and GDI+ constants
Private Const PLANES = 14           ' Number of planes
Private Const BITSPIXEL = 12         ' Number of bits per pixel
Private Const PATCOPY = &HF00021     ' (DWORD) dest = pattern
Private Const PICTYPE_BITMAP = 1      ' Bitmap type
Private Const InterpolationModeHighQualityBicubic = 7
Private Const GDIP_WMF_PLACEABLEKEY = &H9AC6CDD7
Private Const UnitPixel = 2

' Initialises GDI Plus
Public Function InitGDIPlus() As Long
    Dim Token As Long
    Dim gdipInit As GdipStartupInput

    gdipInit.GdipplusVersion = 1
    GdipplusStartup Token, gdipInit, ByVal 0&
    InitGDIPlus = Token
End Function

' Releases GDI Plus
Public Sub FreeGDIPlus(Token As Long)
    GdipplusShutdown Token
End Sub

' Loads the picture (optionally resized)
Public Function LoadPictureGDIPlus(PicFile As String, Optional Width As Long = -1, Optional
Height As Long = -1, Optional ByVal BackColor As Long = vbWhite, Optional RetainRatio As
Boolean = False) As IPicture
    Dim hDC As Long
    Dim hBitmap As Long
    Dim Img As Long

    ' Load the image
    If GdipLoadImageFromFile(StrPtr(PicFile), Img) <> 0 Then
        Err.Raise 999, "GDI+ Module", "Error loading picture " & PicFile
        Exit Function
    End If

    ' Calculate picture's width and height if not specified
    If Width = -1 Or Height = -1 Then
        GdipGetImageWidth Img, Width
        GdipGetImageHeight Img, Height
    End If

    ' Initialise the hDC
    InitDC hDC, hBitmap, BackColor, Width, Height
```

```
' Resize the picture
gdipResize Img, hDC, Width, Height, RetainRatio
GdipDisposeImage Img

' Get the bitmap back
GetBitmap hDC, hBitmap

' Create the picture
Set LoadPictureGDIPlus = CreatePicture(hBitmap)
End Function

' Initialises the hDC to draw
Private Sub InitDC(hDC As Long, hBitmap As Long, BackColor As Long, Width As Long, Height As Long)
    Dim hBrush As Long

    ' Create a memory DC and select a bitmap into it, fill it in with the backcolor
    hDC = CreateCompatibleDC(0&)
    hBitmap = CreateBitmap(Width, Height, GetDeviceCaps(hDC, PLANES), GetDeviceCaps(hDC, BITSPIXEL), 0&)
    hBitmap = SelectObject(hDC, hBitmap)
    hBrush = CreateSolidBrush(BackColor)
    hBrush = SelectObject(hDC, hBrush)
    PatBlt hDC, 0, 0, Width, Height, PATCOPY
    DeleteObject SelectObject(hDC, hBrush)
End Sub

' Resize the picture using GDI plus
Private Sub gdipResize(Img As Long, hDC As Long, Width As Long, Height As Long, Optional RetainRatio As Boolean = False)
    Dim Graphics As Long      ' Graphics Object Pointer
    Dim OrWidth As Long       ' Original Image Width
    Dim OrHeight As Long      ' Original Image Height
    Dim OrRatio As Double     ' Original Image Ratio
    Dim DesRatio As Double    ' Destination rect Ratio
    Dim DestX As Long          ' Destination image X
    Dim DestY As Long          ' Destination image Y
    Dim DestWidth As Long      ' Destination image Width
    Dim DestHeight As Long     ' Destination image Height

    GdipCreateFromHDC hDC, Graphics
    GdipSetInterpolationMode Graphics, InterpolationModeHighQualityBicubic

    If RetainRatio Then
        GdipGetImageWidth Img, OrWidth
        GdipGetImageHeight Img, OrHeight

        OrRatio = OrWidth / OrHeight
        DesRatio = Width / Height

        ' Calculate destination coordinates
    End If
End Sub
```

```
DestWidth = IIf(DesRatio < OrRatio, Width, Height * OrRatio)
DestHeight = IIf(DesRatio < OrRatio, Width / OrRatio, Height)
'   DestX = (Width - DestWidth) / 2
'   DestY = (Height - DestHeight) / 2

DestX = 0
DestY = 0

GdipDrawImageRectRectI Graphics, Img, DestX, DestY, DestWidth, DestHeight, 0, 0,
OrWidth, OrHeight, UnitPixel, 0, 0, 0
Else
    GdipDrawImageRectI Graphics, Img, 0, 0, Width, Height
End If
GdipDeleteGraphics Graphics
End Sub

' Replaces the old bitmap of the hDC, Returns the bitmap and Deletes the hDC
Private Sub GetBitmap(hDC As Long, hBitmap As Long)
    hBitmap = SelectObject(hDC, hBitmap)
    DeleteDC hDC
End Sub

' Creates a Picture Object from a handle to a bitmap
Private Function CreatePicture(hBitmap As Long) As IPicture
    Dim IID_IDispatch As GUID
    Dim Pic           As PICTDESC
    Dim IPic          As IPicture

    ' Fill in OLE IDispatch Interface ID
    IID_IDispatch.Data1 = &H20400
    IID_IDispatch.Data4(0) = &HC0
    IID_IDispatch.Data4(7) = &H46

    ' Fill Pic with necessary parts
    Pic.size = Len(Pic)           ' Length of structure
    Pic.Type = PICTYPE_BITMAP    ' Type of Picture (bitmap)
    Pic.hBmp = hBitmap            ' Handle to bitmap

    ' Create the picture
    OleCreatePictureIndirect Pic, IID_IDispatch, True, IPic
    Set CreatePicture = IPic
End Function

' Returns a resized version of the picture
Public Function Resize(Handle As Long, PicType As PictureTypeConstants, Width As Long,
Height As Long, Optional BackColor As Long = vbWhite, Optional RetainRatio As Boolean =
False) As IPicture
    Dim Img      As Long
    Dim hDC      As Long
    Dim hBitmap  As Long
    Dim WmfHeader As wmfPlaceableFileHeader
```

```
' Determine picture type
Select Case PicType
Case vbPicTypeBitmap
    GdipCreateBitmapFromHBITMAP Handle, ByVal 0&, Img
Case vbPicTypeMetafile
    FillInWmfHeader WmfHeader, Width, Height
    GdipCreateMetafileFromWmf Handle, False, WmfHeader, Img
Case vbPicTypeEMetafile
    GdipCreateMetafileFromEmf Handle, False, Img
Case vbPicTypeIcon
    ' Does not return a valid Image object
    GdipCreateBitmapFromHICON Handle, Img
End Select

' Continue with resizing only if we have a valid image object
If Img Then
    InitDC hDC, hBitmap, BackColor, Width, Height
    gdipResize Img, hDC, Width, Height, RetainRatio
    GdipDisposeImage Img
    GetBitmap hDC, hBitmap
    Set Resize = CreatePicture(hBitmap)
End If
End Function

' Fills in the wmfPlacable header
Private Sub FillInWmfHeader(WmfHeader As wmfPlaceableFileHeader, Width As Long, Height As Long)
    WmfHeader.BoundingBox.Right = Width
    WmfHeader.BoundingBox.Bottom = Height
    WmfHeader.Inch = 1440
    WmfHeader.Key = GDIP_WMF_PLACEABLEKEY
End Sub
```

modUser (\lib\modUser.bas)

Description

It is a multi-user handling library to enforce user profile based access rights management. It includes methods to automate login and logout for easy access.

Source Code

```
Public username As String, password As String, usertype As String
Public lastLogin As Date
Public loggedIn As Boolean

Public Sub initUser()
    username = ""
    password = ""
    usertype = "USER"
    loggedIn = False
End Sub

Public Sub usrLogin(usr As String, pwd As String, ut As String)
    If Not loggedIn Then
        username = usr
        password = pwd
        loggedIn = True
        usertype = ut
        lastLogin = Time
        Debug.Print username & " logged in at " & lastLogin
    Else
        Debug.Print username & " is already logged in! Please logout first."
    End If
End Sub

Public Sub usrLogout()
    If loggedIn Then
        Debug.Print username & " logged out at " & Time
        username = ""
        password = ""
        usertype = "USER"
        loggedIn = False
    Else
        Debug.Print "No user is logged in! Please login first."
    End If
End Sub
```

modDatabase (\lib\modDatabase.bas)

Description

It provides a wrapper for the Microsoft ActiveX Data Objects Database class. It provides methods to open and close a Database connection by specifying the path to the database. It returns a Recordset object which can be manipulated programmatically.

Source Code

```
Option Explicit

Public conn As ADODB.Connection
Public dbOpen As Boolean

Public Sub initDB()
StartConn:
    dbOpen = False
    Set conn = New ADODB.Connection
    conn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path &
"\db\BookDB.mdb;Persist Security Info=False"
    conn.CursorLocation = adUseClient
    conn.Open
    If Not conn.State = adStateOpen Then
        Select Case MsgBox("There was an error opening the database! Please exit and restart
the program. Alternately, you can try to connect again.", vbCritical + vbApplicationModal +
vbRetryCancel + vbDefaultButton1, "Database Error")
            Case vbRetry
                GoTo StartConn
            Case vbCancel
                End
            End Select
    Else
        dbOpen = True
    End If
End Sub

Public Sub closeDB()
    conn.Close
    dbOpen = False
End Sub
```

Conclusion

Roadmap

Further planned improvements to the software include:

- ✚ A better user authentication system, including password hashing and salting for better security.
- ✚ Offloading the billing section to a secondary database to keep track of individual sales.
- ✚ More thorough testing and bug fixes.

Discussion

The project has helped me develop a lot of understanding of the software test and release cycle apart from helping me have a better grasp over Visual Basic and designing fluid and responsive Graphical User Interfaces (GUIs). It has also helped to improve my knowledge of the Win32 API and the various methods that it provides.

References

- ✚ MSDN: Visual Basic 6.0 Technical Reference
<http://msdn.microsoft.com/en-us/library/hh127540.aspx>
- ✚ Programming Win32 Under the API
<http://books.google.co.in/books?isbn=1578200679>
- ✚ Windows API Index (Windows) – MSDN – Microsoft
[http://msdn.microsoft.com/en-us/library/ff818516\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ff818516(v=vs.85).aspx)