FURNITURE CO. CATALOG

18/SP-COP-2939-06337 Computer Programming Capstone

APRIL 16, 2018

SHAWN BROYLES

Table of Contents

Source Code and Documentation Package

Product Overview

The final project's end product will be a Visual Basic .NET program accompanied by an SQL database that will have data for accounts, payments, and products for a furniture store. The purpose of the program is to provide a way for customers to purchase items after they find out an item is physically not in stock at the business location. The program will be only available on a computer in the business so that the security risk is minimal. It will have forms for welcome, login, registration, catalog, checkout, user's account, catalog items, and payment. The catalog will have multiple search options for the user and buttons on every form that will provide a user-friendly experience.

Working Title	Furniture Co. Catalog
Purpose	To provide a user-friendly experience for purchasing items at a computer in a business when items are not physically in stock at the business location.
Platform	Windows
Intended User	An unsatisfied customer after he or she found out that an item is not in stock.
Source of the idea for the project	When I was trying to think of a project idea that would take up at least 1000 lines and use a database, I thought making a program for purchasing items from a catalog would be reasonable.
Development environment and tools that will be used	Visual Studio Community 2017 version 15.3.26730.8 Microsoft .NET Framework 4.7.02053 Visual Basic Power Packs Controls 12.0.2.21005.1 System.Data.SQLite.Core NuGet Package version 1.0.108 GitHub Extension for Visual Studio version 2.4.3.1737 Atomineer Pro Documentation Trial 9.42.3.2590
Development languages	Visual Basic .NET and SQL
Limitations or risks anticipated	The program should only be used at the business location since it will be using a database that contains sensitive information.
Schedule	February 14, 2018 – Update document with revisions, pseudocode, and more charts/diagrams. February 28, 2018 – Update document with more visual depictions and documentation for the methods and functions. March 14, 2018 – Create a skeleton program and build a test database. Build a unit test suite using NUnit. March 22, 2018 – Write and test source code. Consider using a version control system.
Estimated lines of code	4500
Documentation	XML documentation
User training	The user will be able to go to Help > About if help is needed.
Installation plan	The program will be made to run on a designated computer at Furniture Co.

Forms in the Visual Basic .NET Windows Application

Welcome – This form will tell customers that they're welcomed to use the application and it will have options for navigating to the registration form, navigating to the login form, and signing in as a guest. The purpose of signing in as a guest will be so that the user can browse the catalog before deciding if he/she wants to create an account.

Login – This form will have a sign in form where it accepts account id, username, or email if the first field and a password in the second field. There will be radio buttons for the user to indicate which login method he/she is attempting. After successfully signing in, the user will be sent to the catalog form.

Registration – This form will have textboxes for username, password, confirm password, first name, last name, email, phone number, and shipping address. It will also have a Submit button that will check for errors, such as an error connecting to the database or invalid data in the username field. The user will be required to have a username that starts with a letter and it must be between 4 and 16 characters. This will prevent any bugs appearing when attempting to sign in with an account id instead of a username or email. A new record will be created in the "ACCOUNT" table in the database. After successfully registering, the user will be sent to the catalog form.

Catalog – This form will have a search bar for searching for items, a shopping cart image that will take the user to the checkout form, radio buttons for filtering search results to a specific category, a button for the account form, and a button for signing out. Each item listed in the catalog will have buttons that will take the user to the item form that shows information for the item.

Checkout – This form will show a list of the items in the user's shopping cart. The user will be able to remove items if he/she chooses. After the user purchases items in the shopping cart with money from his/her account, the shopping cart will become empty and some records in the database will be changed. The "PROD_STOCK" will decrease for items purchased, the "ACC_MONEY" will decrease for the account that makes the purchase, and a new record will be made in the "PAYMENT" table for the payment.

Account – This form will show information about the account that is currently logged in after the user re-enters his or her password. It will show the account id, username, first name, last name, email, phone number, shipping address, amount of money on the account, and creation date. It will have buttons that will take the user to the catalog and checkout forms.

Item – This form will show information about an item in the catalog. It will include an item id, name, price, stock, shipping fee, category, and description for a specific item. The user will have the option to add the item with a specified quantity to the shopping cart.

Payment – This form will allow the user to purchase his or her selected items with the money that is on his or her account after he or she re-enters his or her password for confirmation. The user will see a line of text saying that there are currently no available options for putting money on his/her account.

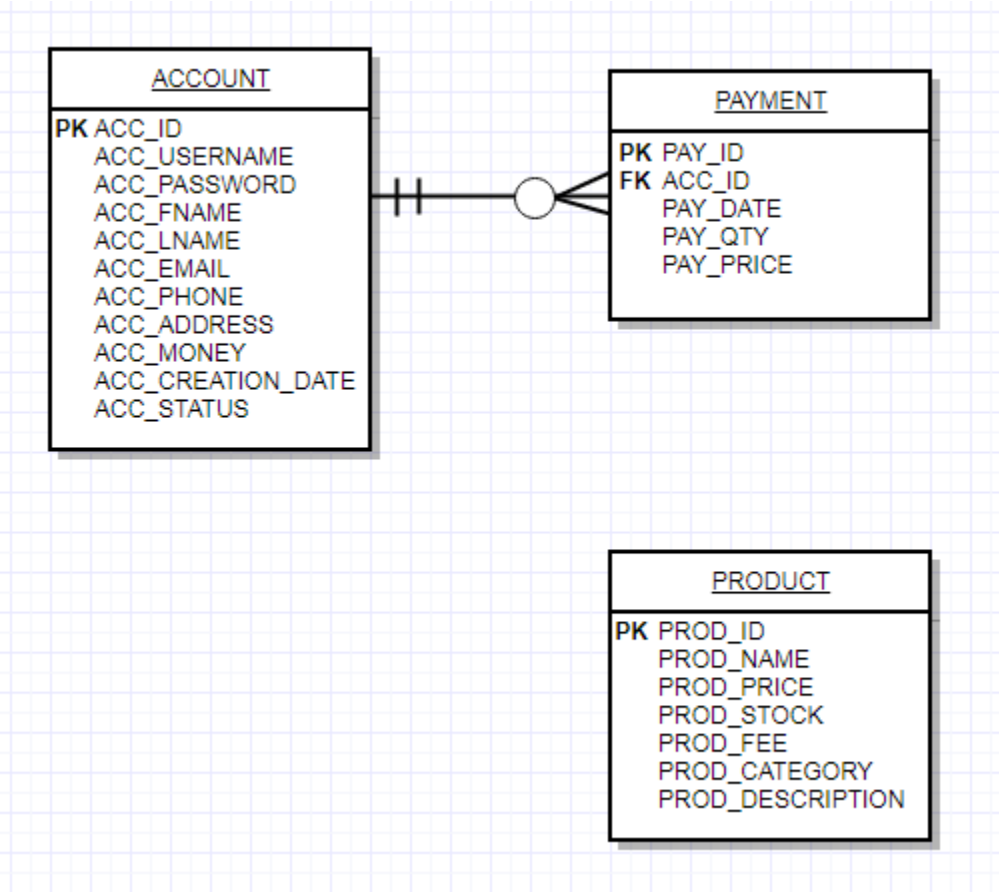
The program will also have modules named Main and SQL, and they will have public subroutines and functions that will be used by multiple forms in the application.

The Database

The “ACCOUNT” table will have information for each account registered in the database. The “PAYMENT” table will have information for items purchased. The “PRODUCT” table will have information for each item that is sold by the company. An account can have no payments or many payments. When someone makes a payment, the following things happen:

- The “PROD_STOCK” field for each item purchased will decrease
- The “ACC_MONEY” field for the account that made the purchase will decrease
- A new record will be made in the “PAYMENT” table
    - The “PAY_ID” of the payment is the unique payment number
    - The “ACC_ID” of the payment is from the account that made the payment
    - The “PAY_DATE” field is the date that the payment was made
    - The “PAY_QTY” field is the number of items purchased with the payment
    - The “PAY_PRICE” field is the total amount paid for the payment

Entity Relationship Diagram with crow’s foot notation:



(Diagram created with Gliffy)

The Database (Continued)

Decimal values (e.g. ACC_MONEY, PAY_PRICE, PROD_PRICE, and PROD_FEE) will be stored as TEXT because I don't know how well SQL databases are at interpreting decimal numbers such as 0.03 and I don't want there to be an issue where 0.029999999329 is truncated to something like 0.0299.

ACCOUNT Table Data Dictionary:

Column	Data Type	Description
ACC_ID	INTEGER PRIMARY KEY	The primary key of the table. The ID of an account.
ACC_USERNAME	TEXT UNIQUE	The username of an account.
ACC_PASSWORD	TEXT	The password of an account.
ACC_FNAME	TEXT	The first name of someone who owns an account.
ACC_LNAME	TEXT	The last name of someone who owns an account.
ACC_EMAIL	TEXT UNIQUE	The email address of someone who owns an account.
ACC_PHONE	TEXT	The phone number of someone who owns an account.
ACC_ADDRESS	TEXT	The shipping address of someone who owns an account.
ACC_MONEY	TEXT	The amount of money on an account.
ACC_CREATION_DATE	DATE	The date an account was created.
ACC_STATUS	TEXT	The status of an account. Examples: Good, Fine, Poor, Locked, Terminated.

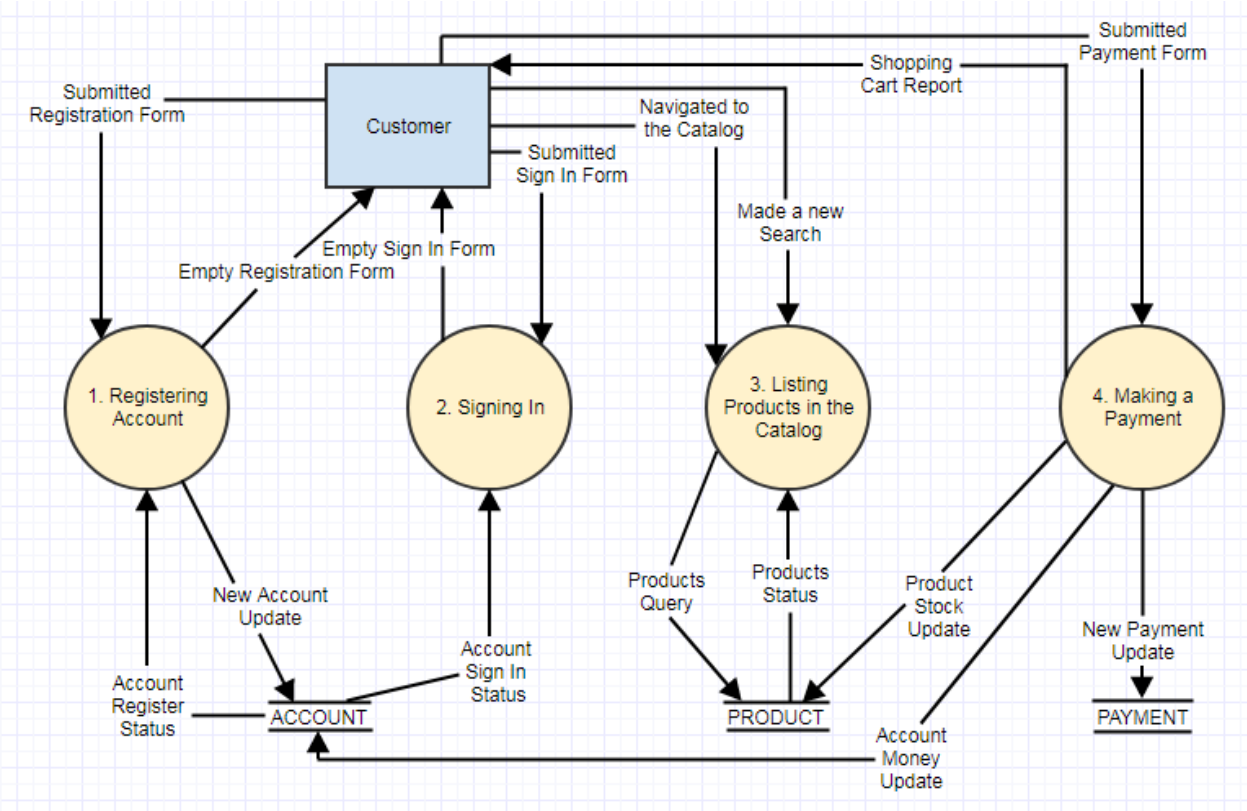
PAYMENT Table Data Dictionary:

Column	Data Type	Description
PAY_ID	INTEGER PRIMARY KEY	The primary key of the table. The ID of a payment.
ACC_ID	INTEGER	The foreign key of the table. The account ID that has made the payment.
PAY_DATE	DATE	The date the payment occurred.
PAY_QTY	INTEGER	The number of items being purchased.
PAY_PRICE	TEXT	The total price of the items being purchased.

PRODUCT Table Data Dictionary:

Column	Data Type	Description
PROD_ID	INTEGER PRIMARY KEY	The primary key of the table. The ID of a product.
PROD_NAME	TEXT	The name of a product.
PROD_PRICE	TEXT	The price of a product.
PROD_STOCK	INTEGER	The amount of stock for a product.
PROD_FEE	TEXT	The shipping fee for a product.
PROD_CATEGORY	TEXT	The category of a product.
PROD_DESCRIPTION	TEXT	The description of a product that the user will see in the catalog.

Top-Level Design - Level 0 Data Flow Diagram



(Diagram and key created with Gliffy)

User-Friendly Features

The program will have a MenuStrip on every form for Navigate, Edit, and Help. For Navigate, the user will have the options Welcome, Registration, Login, Catalog, Checkout, Account, Item, and Payment. For Edit, the user will have the options Reload Form, Sign Out, and Exit. For Help, the user will have the options About and Print. When the user clicks on a menu item to sign out, exit the application, or reload the form, there will be a pop-up to confirm the user's intention. The menu items in the MenuStrip will be there for the user's convenience. The user will also have the option to use buttons within the forms that perform the same functionalities as some of the menu items. The tab index for every button and text box on each form will be set appropriately. All forms in the program will have similar color schemes. Keyboard shortcuts (hotkeys) will be available for every menu item and button in the application.

Pseudocode for registering an account

```
Function register account
    If all required fields are filled out
        If the desired email is between 5 and 50 characters in length,
        contains only alphanumeric characters, a '@' character, and a
        '.' character, and the desired username starts with a letter,
        is between 4 and 16 characters in length, and contains only
        alphanumeric characters
            Attempt to open a database connection
            If the connection is open
                Search the database for a record that has a username
                or email that matches the data entered
                If there are no usernames and no emails that match
                    Create a new record in the database for the new
                    account
                    Clear all the fields on the registration form
                    Log the user into his/her new account
                    Navigate the user to the catalog
                EndIf
                Close the database connection
            Else
                Display a yes/no message box saying "Unable to connect
                to the database. Would you like to try again?"
            EndIf
        EndIf
    EndIf
    Return true if the account was created; else return false
EndFunction
```

Pseudocode for signing into an account

```
Function attempt sign in

    Declare Boolean ReturnBool

    Set ReturnBool = false

    If the argument is “Email”

        Search the database for a record that has an email that
        matches

        If there if an email that matches and the second field in
        the sign in form matches the password in the record

            Set ReturnBool = true

        EndIf

    ElseIf the argument is “Username”

        Search the database for a record that has a username that
        matches

        If there if a username that matches and the second field in
        the sign in form matches the password in the record

            Set ReturnBool = true

        EndIf

    ElseIf the argument is “Account_Id”

        Search the database for a record that has a username that
        matches

        If there if an account id that matches and the second field
        in the sign in form matches the password in the record

            Set ReturnBool = true

        EndIf

    EndIf

    Return ReturnBool

EndFunction
```

Pseudocode for signing into an account (Continued)

```
Function sign in

    Declare String ReturnString

    If the first field contains a '@' character

        Call on the attempt sign in function with the
        argument "Email"

        If the function called on returns false

            Set ReturnString = "Error: Email not found."

        Else

            Set ReturnString = "Success"

        EndIf

    ElseIf the first field starts with a letter and only contains
    alphanumeric characters

        Call on the attempt sign in function with the
        argument "Username"

        If the function called on returns false

            Set ReturnString = "Error: Username not found."

        Else

            Set ReturnString = "Success"

        EndIf

    ElseIf the first field is numeric

        Call on the attempt sign in function with the
        argument "Account_Id"

        If the function called on returns false

            Set ReturnString = "Error: Account Id not found."

        Else

            Set ReturnString = "Success"

        EndIf

    Else

        Set ReturnString = "Error: The value in the first field
        cannot be identified as an email, username, or account id."

    EndIf
```

Pseudocode for signing into an account (Continued)

```
    Return ReturnString
EndFunction


Function submit button clicked
    If the two fields on the sign in form are filled out
        Attempt to open a database connection
        If the connection is open
            Call on function sign in
            If the function called on doesn't return "Success"
                Display a popup that shows the text that was returned
                Display text at the top of the sign in form that shows
                the text that was returned
            Else
                Clear the two fields on the sign in form
                Log the user into his/her account
                Navigate the user to the catalog
            EndIf
            Close the database connection
        Else
            Display a yes/no message box saying "Unable to connect
            to the database. Would you like to try again?"
        EndIf
    EndIf
EndFunction
```

Use case table for registering an account

ID and Name:	UC-1 Registering Account		
Created By:	Shawn Broyles	Date Created:	2/16/2018
Primary Actor:	The customer	Secondary Actor(s):	The database
Description:	The customer fills out a registration form through the program to create an account. The program attempts to validate the fields and creates a record in the "ACCOUNT" table in the database for the new account.		
Trigger:	The customer presses the "Submit" button on the registration form after filling out information.		
Preconditions:	PRE-1. The customer's form has all required fields filled out. PRE-2. The database is working properly.		
Postconditions:	POST-1. The customer is signed in as the new account. POST-2. The customer is sent to the catalog form.		
Normal Flow:	**1.0 Registering Account** 1. The customer submits a registration form through the program. 2. The program checks if all the required fields contain valid data (see 1.0.E1). 3. A new record is made in the "ACCOUNT" table in the database for the new account (see 1.0.E2).		
Alternative Flows:	N/A		
Exceptions:	**1.0.E1 Invalid Fields** 1. The program displays a message telling the customer which fields don't contain valid data. 2. The program creates text at the top of the form that also tell the customer which fields don't contain valid data. 3. The program starts the normal flow over. **1.0.E2 Database is Not Available** 1. The program displays a yes/no message box that says: Unable to connect to the database. Would you like to try again? 2. The customer chooses if he/she wants to try to connect to the database again (3) or not (4a). 3. The program starts the normal flow over. 4a. The customer doesn't want to try again. 4b. The use case is terminated.		
Priority:	High		
Frequency of Use:	Once per new customer that wants to create an account		
Other Information:	An example of invalid data in a field would be trying to create an account with a username that already exists in the database. The customer will be unable to create a username that contains only numbers since the signing in form will allow for customers to use their account id instead of their username. The customer doesn't need to check his/her email to finish creating his/her account.		
Assumptions:	N/A		

Use case table for signing into an account

ID and Name:	UC-2 Signing In		
Created By:	Shawn Broyles	Date Created:	2/16/2018
Primary Actor:	The customer	Secondary Actor(s):	The database
Description:	The customer fills out a sign in form at the main menu. The program checks if the fields are valid by reading from the “ACCOUNT” table in the database.		
Trigger:	The customer presses the “Submit” button on the sign in form after filling out information.		
Preconditions:	PRE-1. The customer previously made an account. PRE-2. The customer’s form has the fields filled out. PRE-3. The database is working properly.		
Postconditions:	POST-1. The customer is signed in. POST-2. The customer is sent to the catalog form.		
Normal Flow:	**2.0 Signing In** 1. The customer submits a sign in form at the main menu. 2. The program reads from the “ACCOUNT” table in the database (see 2.0.E1). 3. The program checks if all the fields contain valid data (see 2.0.E2).		
Alternative Flows:	N/A		
Exceptions:	**2.0.E1 Database is Not Available** 1. The program displays a yes/no message box that says: Unable to connect to the database. Would you like to try again? 2. The customer chooses if he/she wants to try to connect to the database again (3) or not (4a). 3. The program starts the normal flow over. 4a. The customer doesn’t want to try again. 4b. The use case is terminated. **2.0.E2 Invalid Fields** 1. The program displays a message telling the customer that his/her username and password combination is invalid. 2. The program creates text at the top of the form that also tell the customer that his/her username and password combination is invalid. 3. The program starts the normal flow over.		
Priority:	High		
Frequency of Use:	Every time customers want to sign in before browsing the catalog through the program.		
Other Information:	The customer may also choose to use an account id and password combination or email and password combination for logging in instead of using a username and password combination.		
Assumptions:	N/A		

Specific Function Requirements

- The inputs will be buttons and textboxes. The buttons will be for navigating forms and changing information. The textboxes will be for signing in at the main menu form, registering an account on the registration form, and choosing a quantity for an item in the shopping cart. The maximum purchasable quantity of an item will be its amount in stock and the minimum quantity will be 1.
- The output will be text that the user will see in the application. For example, the user will see text for signing in, registering an account, items in the catalog, adding items to the shopping cart, and purchasing items.
- There will be an output format for items in the shopping cart. It will tell the user the product id, name, quantity, and total price for each item in his or her shopping cart.
- The hardware the user will use is a computer located at a business, a monitor, a keyboard, and a mouse. The software the user will use is a Windows operating system and the final project's end product.
- When the user tries to sign in at the main menu form, the program will check the "ACCOUNT" table in the database for matching data. When the user tries to register a new account, the program will check the textboxes for errors before attempting to create a new record in the "ACCOUNT" table. The user input is validated with regular expressions so that the program doesn't use bad data in SQL statements.
- All the tasks the user would want to perform are signing in, registering a new account, browsing the catalog, looking at their account info, changing their account info, adding items to their shopping cart, removing items from their shopping cart, changing the quantity of an item in their shopping cart, viewing items in their shopping cart, purchasing items listed in their shopping cart, and signing out.
- The data used in each task are:
    - Signing in – The "ACCOUNT" table in the database will be checked for a matching account id, username, or email for the first textbox in the form, and a matching password for the second textbox.
    - Registering a new account – The "ACCOUNT" table is checked for matching fields.
    - Browsing the catalog – The products will be read from the "PRODUCT" table and stored in memory for when the user searches for a product.
    - Looking at account info – The "ACCOUNT" table will be used to find information on the current user's account, so it can be displayed in the account form.
    - Adding or Removing items to or from the shopping cart – Variables will be used during the user's signed in session to keep track of items in the shopping cart.
    - Viewing items in the shopping cart – Variables will be used for keeping track of each item in the shopping cart and their quantities and the "PRODUCT" table will be used for retrieving information about each item.
- The data resulting from each task are:
    - Registering a new account – If there are no matching usernames or emails when trying to register a new account, a new record will be made in the "ACCOUNT" table.
    - Changing account info – The record for a user's account in the "ACCOUNT" table will be changed if the user decides to change information on the account form.

- Adding or Removing items to or from the shopping cart – Variables will have new or changed values if the user adds or removes an item to or from his or her shopping cart.
- Changing the quantity of an item in the shopping cart – A variable that has the quantity of an item in the shopping cart will have a new value.
- Purchasing items listed in the shopping cart – The amount of money on a user's account will decrease and the variables for the shopping cart items will be set back to their defaults.
- Signing out – A variable is used to keep track of the user that is signed in will be set back to the default.

Specific Nonfunctional (Quality) Requirements

- A line of text informing the user of how long it should take for an operation to happen will be present at the top or bottom of a form if it is necessary.
- The time it takes to process operations should be 4 seconds or less.
- The level of security is low. The passwords for accounts will be in plaintext in the "ACCOUNT" table in the database, but the program will only be available on a specific computer at the business location to lessen the risk.
- The program should provide a user-friendly experience. If there is a software failure, the user wouldn't be signed in anymore and the shopping cart would be empty. The vital information that needs to be protected from failure would be the data in the database.
- The minimum RAM recommended would be 32 MB, and the minimum free disk space recommended would be 16 MB to ensure the program will work as expected.
- The program will have a specific function for allowing users to sign into a guest account if they choose to do so.
- The program will be successful if a customer in at the business location is able to simply walk up to a computer running the program and figure out how to use it without frustration. The program will be a failure if the program lacks basic user-friendly features.

Requirements Quality

- Users should be able to understand the requirements since programs are made for the user's needs.
- There are no requirements that conflict with other requirements.
- Performance and reliability are preferred over security and availability.
- The requirements do not avoid specifying the design.
- The requirements are at a fairly consistent level of detail.
- The requirements are clear enough to be turned over to an independent group for construction and still be understood.
- Each item is relevant to the problem and its solution, and each item can be traced back to their origins in the problem environment.

- Each requirement is testable. It should be possible for independent testing to determine whether each requirement has been satisfied.
- I am unable to foresee any possible changes to the requirements.

Requirements Completeness

- An area that will have incompleteness will be the functionality for the payment options on the payment form because I am not familiar with creating or using code for secure transactions with payment processors. Another area that will have incompleteness will be the functionality for shipping items after the user (hypothetically) makes a purchase.
- The requirements are complete.
- I am comfortable with all the requirements.

Source Code

The following source code is for the 8 forms and 2 modules in the application. It does not include generated source code. To see the generated source code, you can go to the GitHub repository for the application and download the release titled “Release 1.0 With Documentation.”

Main Module

```
' Program:  Furniture Co. Catalog
' Version:  1.0
' Author:   Shawn Broyles
' Date:     4/9/2018
' Purpose:  This application provides a user-friendly experience for
'           purchasing items when they are not in stock at a business
'           location.

' The Main module contains public methods that are used by other forms
' in the application.

Module Main
    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   Values that represent the forms in the application. </summary>
    '''
    ''' <remarks>
    ''' The values are used for determining which form the user is currently at and also which
form
    ''' the user wants to navigate to.
    ''' </remarks>
    '''-----------------------------------------------------------------------------------------
-------

    Enum Forms
        ''' <summary>   An enum constant representing the Welcome form. </summary>
        WELCOME
        ''' <summary>   An enum constant representing the Registration form. </summary>
        REGISTRATION
        ''' <summary>   An enum constant representing the Login form. </summary>
        LOGIN
        ''' <summary>   An enum constant representing the Catalog form. </summary>
        CATALOG
        ''' <summary>   An enum constant representing the Checkout form. </summary>
        CHECKOUT
        ''' <summary>   An enum constant representing the Account form. </summary>
        ACCOUNT
        ''' <summary>   An enum constant representing the Item form. </summary>
        ITEM
        ''' <summary>   An enum constant representing the Payment form. </summary>
        PAYMENT
        ''' <summary>   An enum constant representing no form. </summary>
        NULL
    End Enum

    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   Values that represent product categories. </summary>
    '''
    ''' <remarks>   The values are used for filtering products listed in the Catalog form.
</remarks>
    '''-----------------------------------------------------------------------------------------
-------

    Enum ProductCategory
        ''' <summary>   An enum constant representing All categories. </summary>
        ALL
```

```
        ''' <summary>   An enum constant representing the Chair category. </summary>
        CHAIR
        ''' <summary>   An enum constant representing the Table category. </summary>
        TABLE
        ''' <summary>   An enum constant representing the Desk category. </summary>
        DESK
        ''' <summary>   An enum constant representing the Couch category. </summary>
        COUCH
        ''' <summary>   An enum constant representing the Carpet category. </summary>
        CARPET
    End Enum

    ''' <summary>   The global constant integer variable for zero. </summary>
    Public Const gcintZero As Integer = 0
    ''' <summary>   The global constant decimal variable for zero. </summary>
    Public Const gcdecZero As Decimal = 0.00D
    ''' <summary>   The global constant string variable for empty. </summary>
    Public Const gcstrEmpty As String = ""

    ''' <summary>   The global User for the currently logged in user. </summary>
    Public gusrCurrentUser As User = New User()
    ''' <summary>   The global List for the ShoppingCartItems. </summary>
    Public glstShoppingCart As New List(Of ShoppingCartItem)()
    ''' <summary>   The global List for all Items from the database. </summary>
    Public glstProducts As New List(Of Item)()
    ''' <summary>   The global List of Items from the search results on the Category form.
</summary>
    Public glstProductResults As New List(Of Item)()
    ''' <summary>   The global List of Items in the shopping cart. </summary>
    Public glstShoppingCartResults As New List(Of Item)()
    ''' <summary>   The global Item for the current Item displayed on the Item form. </summary>
    Public gitmCurrentItem As New Item()

    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   The PositionForm subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for positioning the form on the screen when it is
loaded or reset. </remarks>
    '''
    ''' <param name="frmCurrentForm">
    ''' The form that is currently displayed.
    ''' </param>
    '''-----------------------------------------------------------------------------------------
-------

    Public Sub PositionForm(ByVal frmCurrentForm As Form)
        ' Setting the form's size back to its default
        frmCurrentForm.Size = frmCurrentForm.RestoreBounds.Size

        ' Positioning the form
        Try
            Dim xCoordinate As Double
            Dim yCoordinate As Double
            xCoordinate = Screen.PrimaryScreen.WorkingArea.Width / 2 - frmCurrentForm.Size.Width
/ 2
            yCoordinate = Screen.PrimaryScreen.WorkingArea.Height / 2 -
frmCurrentForm.Size.Height / 1.5
            ' Making sure the entire title bar of the window is shown
            If (yCoordinate < gcintZero) Then
                yCoordinate = gcintZero
            End If
            frmCurrentForm.Location = New Point(Convert.ToInt32(xCoordinate),
Convert.ToInt32(yCoordinate))
        Catch ex As Exception
            ' Writing the error to the output
```

```
            Console.WriteLine("Position Form Error = " & ex.Message)
        End Try
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Navigate subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for navigating between the forms in the application.
</remarks>
    '''
    ''' <param name="intForm">
    ''' The form that the user wants to navigate to.
    ''' </param>
    ''' <param name="frmCurrentForm">
    ''' The currently displayed form.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub Navigate(ByVal intForm As Integer, ByVal frmCurrentForm As Form)
        Dim frmChosenForm As Form
        Dim blnSignedOutRequired As Boolean = False
        Dim blnSignedInRequired As Boolean = False
        Dim blnPasswordRequired As Boolean = False
        Dim blnItemRequired As Boolean = False
        Select Case intForm
            Case Forms.REGISTRATION
                frmChosenForm = frmRegistration
                blnSignedOutRequired = True
            Case Forms.LOGIN
                frmChosenForm = frmLogin
                blnSignedOutRequired = True
            Case Forms.CATALOG
                frmChosenForm = frmCatalog
                blnSignedInRequired = True
            Case Forms.CHECKOUT
                frmChosenForm = frmCheckout
                blnSignedInRequired = True
                blnPasswordRequired = True
            Case Forms.ACCOUNT
                frmChosenForm = frmAccount
                blnSignedInRequired = True
                blnPasswordRequired = True
            Case Forms.ITEM
                frmChosenForm = frmItem
                blnItemRequired = True
                blnSignedInRequired = True
            Case Forms.PAYMENT
                frmChosenForm = frmPayment
                blnSignedInRequired = True
                blnPasswordRequired = True
            Case Else
                frmChosenForm = frmWelcome
        End Select

        Dim strMessage As String = "Unable to navigate to the displayed form."
        Dim cstrTitle As String = "Error"

        ' Navigating
        If (frmCurrentForm.Equals(frmWelcome) And Not frmChosenForm.Equals(frmWelcome)) Then
            If NavigatePrerequisite(blnSignedOutRequired, blnSignedInRequired,
blnPasswordRequired, blnItemRequired) Then
                frmChosenForm.ShowDialog()
            End If
        ElseIf (Not frmCurrentForm.Equals(frmWelcome) And frmChosenForm.Equals(frmWelcome)) Then
```

```
            If NavigatePrerequisite(blnSignedOutRequired, blnSignedInRequired,
blnPasswordRequired, blnItemRequired) Then
                frmCurrentForm.Dispose()
            End If
        ElseIf (Not frmChosenForm.Equals(frmCurrentForm)) Then
            If NavigatePrerequisite(blnSignedOutRequired, blnSignedInRequired,
blnPasswordRequired, blnItemRequired) Then
                frmCurrentForm.Dispose()
                frmChosenForm.ShowDialog()
            End If
        Else
            MsgBox(strMessage, , cstrTitle)
        End If
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The NavigatePrerequisite function. </summary>
    ''' <remarks>
    ''' This function is used to check if the user is required to sign out, sign in, re-enter a
    ''' password, or choose an Item's details before navigating to a form.
    ''' </remarks>
    ''' <param name="blnSignedOutRequired">
    ''' True if the user has to be signed out.
    ''' </param>
    ''' <param name="blnSignedInRequired">
    ''' True if the user has to be signed in.
    ''' </param>
    ''' <param name="blnPasswordRequired">
    ''' True if the user has to re-enter a password.
    ''' </param>
    ''' <param name="blnItemRequired">
    ''' True if the user has to select an Item for its details to be displayed.
    ''' </param>
    ''' <returns>   True if the user meets the prerequisites, false if not. </returns>
    '''-------------------------------------------------------------------------------------------
-------

    Function NavigatePrerequisite(ByVal blnSignedOutRequired As Boolean, ByVal
blnSignedInRequired As Boolean, ByVal blnPasswordRequired As Boolean, ByVal blnItemRequired As
Boolean) As Boolean
        Dim blnNavigate As Boolean = True
        Dim strMessage As String = ""
        Dim cstrTitle As String = "Error"
        If (blnItemRequired AndAlso gitmCurrentItem.ID.Equals(gcintZero)) Then
            AskForItem()
        End If
        If (Not blnItemRequired Or gitmCurrentItem.ID.Equals(gcintZero)) Then
            If (blnSignedInRequired And gusrCurrentUser.SignedIn.Equals(False)) Then
                blnNavigate = False
                strMessage = "You must be signed in to navigate to this form."
                MsgBox(strMessage, , cstrTitle)
            ElseIf (blnSignedOutRequired And gusrCurrentUser.SignedIn.Equals(True)) Then
                blnNavigate = False
                strMessage = "You must be signed out to navigate to this form."
                MsgBox(strMessage, , cstrTitle)
            ElseIf (blnPasswordRequired AndAlso ConfirmPasswordPopup().Equals(False)) Then
                blnNavigate = False
                strMessage = "Invalid Password."
                MsgBox(strMessage, , cstrTitle)
            End If
        End If

        Return blnNavigate

    End Function
```

```
    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The ReloadForm subroutine. </summary>
    '''
    ''' <remarks>   This subroutine reloads the currently displayed form. </remarks>
    '''
    ''' <param name="frmCurrentForm">
    ''' The currently displayed form.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub ReloadForm(ByVal frmCurrentForm As Form)
        PositionForm(frmCurrentForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SignOut subroutine. </summary>
    '''
    ''' <remarks>   This subroutine signs the user out of his/her account. </remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub SignOut()
        If (gusrCurrentUser.SignedIn) Then
            Dim strUserSigningOut As String = gusrCurrentUser.Username
            gusrCurrentUser.SignOut()
            ClearShoppingCart()
            gitmCurrentItem = New Item()
            MsgBox("You are now signed out.", , "Sign Out Success")
        Else
            MsgBox("You are not signed in.", , "Sign Out Error")
        End If
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The CreateShoppingCart subroutine. </summary>
    '''
    ''' <remarks>   This subroutine adds products to the shopping cart with zero quantity.
</remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub CreateShoppingCart()
        Dim sciItem As ShoppingCartItem
        glstProducts.ForEach(Sub(itmItem)
                                 sciItem = New ShoppingCartItem(itmItem, gcintZero)
                                 glstShoppingCart.Add(sciItem)
                             End Sub)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The ClearShoppingCart subroutine. </summary>
    '''
    ''' <remarks>   This subroutine empties the shopping cart after the user confirms his/her
intention. </remarks>
    '''
    ''' <param name="blnConfirmIntention">
    ''' (Optional) True if the user has to confirm his/her intention to clear the shopping cart.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------
```

```
    Public Sub ClearShoppingCart(Optional ByVal blnConfirmIntention As Boolean = False)
        If (glstShoppingCart.Count > gcintZero) Then
            If (blnConfirmIntention AndAlso MessageBox.Show("Are you sure you want to empty the
shopping cart?", "Empty Shopping Cart?", MessageBoxButtons.YesNo).Equals(DialogResult.Yes)) Then
                glstShoppingCart.ForEach(Sub(sciItem)
                                             sciItem.Quantity = gcintZero
                                         End Sub)
            Else
                glstShoppingCart.ForEach(Sub(sciItem)
                                             sciItem.Quantity = gcintZero
                                         End Sub)
            End If
        End If
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SignInAsGuest subroutine. </summary>
    '''
    ''' <remarks>   This subroutine signs the user into the guest account. </remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub SignInAsGuest()
        ' Hard-coded guest sign-in (no SQL validation)
        gusrCurrentUser.SignIn(SQLGetRecordID(DatabaseTables.ACCOUNT, "ACC_USERNAME", "Guest"))
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The AskForItem function. </summary>
    '''
    ''' <remarks>   This function asks the user to enter the name or ID of a product. </remarks>
    '''
    ''' <returns>   True if the item is valid, false otherwise. </returns>
    '''-------------------------------------------------------------------------------------------
-------

    Function AskForItem() As Boolean
        Dim blnValidItem As Boolean = False
        Dim strProduct As String
        strProduct = InputBox("Enter the name or ID of a product.", "Unknown Item")
        If (RegexValidateUserData(strProduct, RegexValidate.ID)) Then
            GetProduct(Convert.ToInt32(strProduct))
        Else
            GetProduct(strProduct)
        End If
        If (gitmCurrentItem.ID.Equals(gcintZero)) Then
            MsgBox("Item not found.", , "Error")
        Else
            blnValidItem = True
        End If
        Return blnValidItem
    End Function

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The ConfirmPasswordPopup function. </summary>
    '''
    ''' <remarks>   This function asks the user to re-enter his/her password. </remarks>
    '''
    ''' <returns>   True if the entered password is correct, false otherwise. </returns>
    '''-------------------------------------------------------------------------------------------
-------
```

```
    Function ConfirmPasswordPopup() As Boolean
        Dim blnConfirmed As Boolean = False
        Dim strPassword As String = InputBox("Re-Enter your password to confirm your identity.",
"Re-Enter Password")
        ' SQL validation is not required here because User.Password is in memory
        blnConfirmed = strPassword.Equals(gusrCurrentUser.Password)
        Return blnConfirmed
    End Function

    '''------------------------------------------------------------------------------------------
-------
    ''' <summary>   The ExitApplication subroutine. </summary>
    '''
    ''' <remarks>   This subroutine terminates the application after the user confirms his/her
intention. </remarks>
    '''------------------------------------------------------------------------------------------
-------

    Public Sub ExitApplication()
        Const cstrMessage As String = "Are you sure you want to exit the application?"
        Const cstrTitle As String = "Exit?"
        Dim intExitApplicationInput As Integer = MessageBox.Show(cstrMessage, cstrTitle,
MessageBoxButtons.YesNo)
        If (intExitApplicationInput = DialogResult.Yes) Then
            Application.Exit()
        End If
    End Sub

    '''------------------------------------------------------------------------------------------
-------
    ''' <summary>   The AboutApplication subroutine. </summary>
    '''
    ''' <remarks>   This subroutine creates a popup that gives the user some information about
the program and the form that is currently displayed. </remarks>
    '''
    ''' <param name="intForm">
    ''' (Optional) The form that the user wants information about.
    ''' </param>
    '''------------------------------------------------------------------------------------------
-------

    Public Sub AboutApplication(Optional ByVal intForm As Integer = Forms.NULL)
        Dim strMessageEnd As String = ""
        Select Case intForm
            Case Forms.WELCOME
                strMessageEnd = "The Welcome form is used for letting customers know they're" &
vbCrLf &
                                "welcomed to use the application. Also, it gives options for" &
vbCrLf &
                                "signing in."
            Case Forms.REGISTRATION
                strMessageEnd = "The Registration form is used by customers to create accounts."
& vbCrLf &
                                "Fill out the form to create an account."
            Case Forms.LOGIN
                strMessageEnd = "The Login form is used for logging into existing accounts." &
vbCrLf &
                                "Fill out the form to login if you have an existing account."
            Case Forms.CATALOG
                strMessageEnd = "The Catalog form is used for browsing the catalog. Search for" &
vbCrLf &
                                "items in the catalog with the search bar. Use the buttons to" &
vbCrLf &
                                "filter results, go to the next page of results, or go to the" &
vbCrLf &
```

```
                                "previous page of results. Then, click on an item to navigate to"
& vbCrLf &
                                "a form that has more details for it."
            Case Forms.CHECKOUT
                strMessageEnd = "The Checkout form displays a list of items in the user's
shopping" & vbCrLf &
                                "cart. There are options for removing items and making
purchases."
            Case Forms.ACCOUNT
                strMessageEnd = "The Account form provides options for changing the user's" &
vbCrLf &
                                "account information."
            Case Forms.ITEM
                strMessageEnd = "The Item form displays information about a specific item in the"
& vbCrLf &
                                "catalog with the option to add the item with a specified
quantitiy" & vbCrLf &
                                "to the shopping cart."
            Case Forms.PAYMENT
                strMessageEnd = "Use the Payment form to exchange real money for account money."
        End Select
        Const cstrMessage As String = "Program Name: Furniture Co. Catalog" & vbCrLf &
                                      "Developed By: Shawn Broyles" & vbCrLf &
                                      "Purpose: This application provides a user-friendly
experience for" & vbCrLf &
                                      "purchasing items when they are not in stock at a business
location." & vbCrLf & vbCrLf &
                                      "Use the Navigate item in the MenuStrip to navigate between
the forms" & vbCrLf &
                                      "(It's located to the left of Help)."
        Const cstrTitle As String = "About"
        If String.IsNullOrWhiteSpace(strMessageEnd) Then
            MsgBox(cstrMessage, , cstrTitle)
        Else
            MsgBox(cstrMessage & vbCrLf & vbCrLf & strMessageEnd, , cstrTitle)
        End If
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The PrintForm subroutine. </summary>
    '''
    ''' <remarks>   This subroutine creates a print preview of the current form. </remarks>
    '''
    ''' <param name="pfObject">
    ''' The PrintForm object that on the currently displayed form.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub PrintForm(ByRef pfObject As Object)
        ' When the PrintForm subroutine is invoked, a print preview appears
        pfObject.PrintAction = Printing.PrintAction.PrintToPreview
        pfObject.Print()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The LoadFormDefaults subroutine. </summary>
    '''
    ''' <remarks>   This subroutine changes the colors of the form and positions the form when
its loaded. </remarks>
    '''
    ''' <param name="frmCurrentForm">
    ''' The currently displayed form.
    ''' </param>
```

```
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub LoadFormDefaults(ByVal frmCurrentForm As Form)
        frmCurrentForm.ForeColor = My.Settings.ForeColor
        frmCurrentForm.BackColor = My.Settings.BackColor

        PositionForm(frmCurrentForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The AddToShoppingCart subroutine. </summary>
    '''
    ''' <remarks>   This subroutine adds a ShoppingCartItem to the shopping cart list. </remarks>
    '''
    ''' <param name="sciNewItem">
    ''' The new ShoppingCartItem.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub AddToShoppingCart(ByRef sciNewItem As ShoppingCartItem)
        glstShoppingCart.Add(sciNewItem)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The AddToProducts subroutine. </summary>
    '''
    ''' <remarks>   This subroutine adds a product to the products list. </remarks>
    '''
    ''' <param name="itmNewItem">
    ''' The new Item.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub AddToProducts(ByRef itmNewItem As Item)
        glstProducts.Add(itmNewItem)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The GetSelectedItem subroutine. </summary>
    '''
    ''' <remarks>   This subroutine changes the current item to the selected item in the results
list on the catalog form. </remarks>
    '''
    ''' <param name="lstListBox">
    ''' The ListBox object on the catalog form.
    ''' </param>
    ''' <param name="Results">
    ''' The search results List from searching on the catalog form.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub GetSelectedItem(ByRef lstListBox As ListBox, ByRef Results As List(Of Item))
        Try
            If (Not String.IsNullOrWhiteSpace(lstListBox.SelectedItem)) Then
                gitmCurrentItem = Results(lstListBox.SelectedIndex())
            End If
        Catch ex As Exception
            Console.WriteLine("Error occurred when attempting to get the current item from the
selected item.")
```

```
            Console.WriteLine(ex.Message)
        End Try
    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
    ''' <summary>   The GetProduct subroutine. </summary>
    '''
    ''' <remarks>   This subroutine changes the current item to an Item from the products List
based off of its ID. </remarks>
    '''
    ''' <param name="intID">
    ''' The ID of an Item.
    ''' </param>
    '''-----------------------------------------------------------------------------------------------
-------

    Public Sub GetProduct(ByVal intID As Integer)
        Try
            glstProducts.ForEach(Sub(itmItem)
                                     If (itmItem.ID.Equals(intID)) Then
                                         gitmCurrentItem = itmItem
                                     End If
                                 End Sub)
        Catch ex As Exception
            MsgBox("An unknown error has occurred when trying to get product information.", ,
"Error")
            Console.WriteLine("Failed at getting product information for ID: " &
intID.ToString())
            Console.WriteLine(ex.Message)
        End Try

    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
    ''' <summary>   The GetProduct subroutine. </summary>
    '''
    ''' <remarks>   This subroutine changes the current item to an Item from the products List
based off of its ID. </remarks>
    '''
    ''' <param name="strName">
    ''' The Name of an Item.
    ''' </param>
    '''-----------------------------------------------------------------------------------------------
-------

    Public Sub GetProduct(ByVal strName As String)
        Try
            glstProducts.ForEach(Sub(itmItem)
                                     If (itmItem.Name.Equals(strName)) Then
                                         gitmCurrentItem = itmItem
                                     End If
                                 End Sub)
        Catch ex As Exception
            MsgBox("An unknown error has occurred when trying to get product information.", ,
"Error")
            Console.WriteLine("Failed at getting product information for Name: " &
strName.ToString())
            Console.WriteLine(ex.Message)
        End Try

    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
```

```
    ''' <summary>   The GetProductCategory function. </summary>
    '''
    ''' <remarks>   This function gets the category of an item. </remarks>
    '''
    ''' <param name="itmItem">
    ''' The Item whose category we want to retrieve.
    ''' </param>
    '''
    ''' <returns>   The category of an item. </returns>
    '''-------------------------------------------------------------------------------------------
-------

    Function GetProductCategory(ByRef itmItem As Item) As Integer
        Dim intCategory As Integer
        If (itmItem.Category.Equals("Chair")) Then
            intCategory = ProductCategory.CHAIR
        ElseIf (itmItem.Category.Equals("Table")) Then
            intCategory = ProductCategory.TABLE
        ElseIf (itmItem.Category.Equals("Desk")) Then
            intCategory = ProductCategory.DESK
        ElseIf (itmItem.Category.Equals("Couch")) Then
            intCategory = ProductCategory.COUCH
        ElseIf (itmItem.Category.Equals("Carpet")) Then
            intCategory = ProductCategory.CARPET
        Else
            intCategory = ProductCategory.ALL
        End If
        Return intCategory
    End Function

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The CheckOutOfStock function. </summary>
    '''
    ''' <remarks>   This function determines if a product in the catalog is out of stock.
</remarks>
    '''
    ''' <param name="itmItem">
    ''' The Item in question.
    ''' </param>
    '''
    ''' <returns>   True if the Item is out of stock, false otherwise. </returns>
    '''-------------------------------------------------------------------------------------------
-------

    Function CheckOutOfStock(ByRef itmItem As Item) As Boolean
        Dim blnItemOutOfStock As Boolean
        blnItemOutOfStock = itmItem.Stock.Equals(gcintZero)
        Return blnItemOutOfStock
    End Function

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The UpdateProducts subroutine. </summary>
    '''
    ''' <remarks>   This subroutine reads from the database and updates values in memory.
</remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub UpdateProducts()
        Try
            glstProducts.ForEach(Sub(itmItem)
                                     itmItem.Update()
                                 End Sub)
        Catch ex As Exception
```

```
            Console.WriteLine(ex.Message)
            Console.WriteLine("Unable to update products list.")
        End Try
    End Sub

    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   The GetMatch function. </summary>
    '''
    ''' <remarks>   This function is used for searching for items in the database. </remarks>
    '''
    ''' <param name="itmItem">
    ''' An item in the catalog.
    ''' </param>
    ''' <param name="strSearchQuery">
    ''' The regular expression search query from the catalog form.
    ''' </param>
    '''
    ''' <returns>   True if a property of the Item matches the regular expression, false
otherwise. </returns>
    '''-----------------------------------------------------------------------------------------
-------

    Function GetMatch(ByRef itmItem As Item, ByVal strSearchQuery As String) As Boolean
        Dim blnMatched As Boolean = False
        If (System.Text.RegularExpressions.Regex.IsMatch(itmItem.ID.ToString(), strSearchQuery)
OrElse
            System.Text.RegularExpressions.Regex.IsMatch(itmItem.Name, strSearchQuery) OrElse
            System.Text.RegularExpressions.Regex.IsMatch(itmItem.Price.ToString(),
strSearchQuery) OrElse
            System.Text.RegularExpressions.Regex.IsMatch(itmItem.Price.ToString("C2"),
strSearchQuery) OrElse
            System.Text.RegularExpressions.Regex.IsMatch(itmItem.Stock.ToString(),
strSearchQuery) OrElse
            System.Text.RegularExpressions.Regex.IsMatch(itmItem.Fee.ToString(), strSearchQuery)
OrElse
            System.Text.RegularExpressions.Regex.IsMatch(itmItem.Fee.ToString("C2"),
strSearchQuery) OrElse
            System.Text.RegularExpressions.Regex.IsMatch(itmItem.Category, strSearchQuery) OrElse
            System.Text.RegularExpressions.Regex.IsMatch(itmItem.Description, strSearchQuery))
Then
            blnMatched = True
        End If
        Return blnMatched
    End Function

    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   The Item class. </summary>
    '''
    ''' <remarks>   This class is used for creating objects for products that the company is
selling. </remarks>
    '''-----------------------------------------------------------------------------------------
-------

    Public Class Item

        '''-------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the ID. </summary>
        '''
        ''' <value> The unique ID of an Item. </value>
        '''-------------------------------------------------------------------------------------
-----------

        Public Property ID As Integer
```

```
        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Name. </summary>
        '''
        ''' <value> The name of an Item. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property Name As String

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Price. </summary>
        '''
        ''' <value> The price of an item. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property Price As Decimal

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Stock. </summary>
        '''
        ''' <value> The stock of an item. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property Stock As Integer

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Fee. </summary>
        '''
        ''' <value> The fee of an item. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property Fee As Decimal

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Category. </summary>
        '''
        ''' <value> The category of an item. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property Category As String

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Description. </summary>
        '''
        ''' <value> The description of an item. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property Description As String

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   The default constructor for creating an Item. </summary>
        '''
```

```
        ''' <remarks>   This constructor sets the properties of an Item to zeros and empty
strings. </remarks>
        '''-------------------------------------------------------------------------------------
-----------

        Public Sub New()
            ID = gcintZero
            Name = gcstrEmpty
            Price = gcdecZero
            Stock = gcintZero
            Fee = gcdecZero
            Category = gcstrEmpty
            Description = gcstrEmpty
        End Sub

        '''-------------------------------------------------------------------------------------
-----------
        ''' <summary>   The parameterized constructor for creating an Item. </summary>
        '''
        ''' <remarks>   This constructor allows the program to create an Item with properties.
</remarks>
        '''
        ''' <param name="intID">
        ''' The unique ID of an Item.
        ''' </param>
        ''' <param name="strName">
        ''' The name of an Item.
        ''' </param>
        ''' <param name="decPrice">
        ''' The price of an Item.
        ''' </param>
        ''' <param name="intStock">
        ''' The stock of an Item.
        ''' </param>
        ''' <param name="decFee">
        ''' The fee of an Item.
        ''' </param>
        ''' <param name="strCategory">
        ''' The category of an Item.
        ''' </param>
        ''' <param name="strDescription">
        ''' The description of an Item.
        ''' </param>
        '''-------------------------------------------------------------------------------------
-----------

        Public Sub New(ByVal intID As Integer, ByVal strName As String, ByVal decPrice As
Decimal, ByVal intStock As Integer, ByVal decFee As Decimal, ByVal strCategory As String, ByVal
strDescription As String)
            ID = intID
            Name = strName
            Price = decPrice
            Stock = intStock
            Fee = decFee
            Category = strCategory
            Description = strDescription
        End Sub

        '''-------------------------------------------------------------------------------------
-----------
        ''' <summary>   The Update subroutine of an Item. </summary>
        '''
        ''' <remarks>   This subroutine updates an Item with information from the database.
</remarks>
        '''-------------------------------------------------------------------------------------
-----------
```

```
        Public Sub Update()
            Dim intRecordID As Integer = ID
            Name = SQLGetFieldInfo(DatabaseTables.PRODUCT, intRecordID, "PROD_NAME")
            Price = Convert.ToDecimal(SQLGetFieldInfo(DatabaseTables.PRODUCT, intRecordID,
"PROD_PRICE"))
            Stock = Convert.ToInt32(SQLGetFieldInfo(DatabaseTables.PRODUCT, intRecordID,
"PROD_STOCK"))
            Fee = Convert.ToDecimal(SQLGetFieldInfo(DatabaseTables.PRODUCT, intRecordID,
"PROD_FEE"))
            Category = SQLGetFieldInfo(DatabaseTables.PRODUCT, intRecordID, "PROD_CATEGORY")
            Description = SQLGetFieldInfo(DatabaseTables.PRODUCT, intRecordID,
"PROD_DESCRIPTION")
            Console.WriteLine("Product " & Name & " was updated.")
        End Sub

        '''-----------------------------------------------------------------------------------------
-----------
        ''' <summary>   The GetShoppingCartItem function of an Item. </summary>
        '''
        ''' <remarks>   This function gets the ShoppingCartItem equivalent of an Item. </remarks>
        '''
        ''' <returns>   The ShoppingCartItem equivalent of an Item. </returns>
        '''-----------------------------------------------------------------------------------------
-----------

        Function GetShoppingCartItem() As ShoppingCartItem
            Dim sciRelatedItem As New ShoppingCartItem()
            glstShoppingCart.ForEach(Sub(sciItem)
                                         If (sciItem.ID.Equals(ID)) Then
                                             sciRelatedItem = sciItem
                                         End If
                                     End Sub)
            Return sciRelatedItem
        End Function
    End Class

    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   The ShoppingCartItem class. </summary>
    '''
    ''' <remarks>   This class is used for creating objects for products in the shopping cart.
</remarks>
    '''-----------------------------------------------------------------------------------------
-------

    Public Class ShoppingCartItem

        '''-----------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the ID. </summary>
        '''
        ''' <value> The unique ID of a ShoppingCartItem. </value>
        '''-----------------------------------------------------------------------------------------
-----------

        Public Property ID As Integer

        '''-----------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Quantity. </summary>
        '''
        ''' <value> The quantity of an Item in the shopping cart. </value>
        '''-----------------------------------------------------------------------------------------
-----------
```

```
        Public Property Quantity As Integer

        '''-----------------------------------------------------------------------------------------
-----------
        ''' <summary>   The default constructor of ShoppingCartItem. </summary>
        '''
        ''' <remarks>   This constructor creates a new ShoppingCartItem with properties set to
zero. </remarks>
        '''-----------------------------------------------------------------------------------------
-----------

        Public Sub New()
            ID = gcintZero
            Quantity = gcintZero
        End Sub

        '''-----------------------------------------------------------------------------------------
-----------
        ''' <summary>   The parameterized constructor of ShoppingCartItem. </summary>
        '''
        ''' <remarks>   This constructor creates a new ShoppingCartItem with properties set from
arguments. </remarks>
        '''
        ''' <param name="itmItem">
        ''' The Item that represents a new ShoppingCartItem.
        ''' </param>
        ''' <param name="intQuantity">
        ''' The quantity of the Item that is being added to the shopping cart.
        ''' </param>
        '''-----------------------------------------------------------------------------------------
-----------

        Public Sub New(ByRef itmItem As Item, ByVal intQuantity As Integer)
            ID = itmItem.ID
            Quantity = intQuantity
        End Sub
    End Class

    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   The User class. </summary>
    '''
    ''' <remarks>   This class is used for creating a User object that represents the signed in
user. </remarks>
    '''-----------------------------------------------------------------------------------------
-------

    Public Class User

        '''-----------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the ID. </summary>
        '''
        ''' <value> The unique ID of a User. </value>
        '''-----------------------------------------------------------------------------------------
-----------

        Public Property ID As Integer

        '''-----------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Username. </summary>
        '''
        ''' <value> The username of a User. </value>
        '''-----------------------------------------------------------------------------------------
-----------
```

```
        Public Property Username As String

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Password. </summary>
        '''
        ''' <value> The password of a User. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property Password As String

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the First name. </summary>
        '''
        ''' <value> The first name of a User. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property FirstName As String

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Last name. </summary>
        '''
        ''' <value> The last name of a User. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property LastName As String

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Email. </summary>
        '''
        ''' <value> The email of a User. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property Email As String

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Phone. </summary>
        '''
        ''' <value> The phone of a User. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property Phone As String

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Address. </summary>
        '''
        ''' <value> The address of a User. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property Address As String

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Money. </summary>
```

```
        '''
        ''' <value> The amount of money on a User's account. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property Money As Decimal

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Creation date. </summary>
        '''
        ''' <value> The creation date of a User. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property CreationDate As String

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Status. </summary>
        '''
        ''' <value> The status of a User's account. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property Status As String

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   Gets or sets the Signed in property. </summary>
        '''
        ''' <value> The signed in property says if a User is signed in or not. </value>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Property SignedIn As Boolean

        ''' <summary>   The UserUpdated event of a User. </summary>
        Public Event UserUpdated()

        '''-------------------------------------------------------------------------------------------
-----------
        ''' <summary>   The default constructor of a User. </summary>
        '''
        ''' <remarks>   This constructor sets the properties of a User to zeros and empty
strings. </remarks>
        '''-------------------------------------------------------------------------------------------
-----------

        Public Sub New()
            ID = gcintZero
            Username = gcstrEmpty
            Password = gcstrEmpty
            FirstName = gcstrEmpty
            LastName = gcstrEmpty
            Email = gcstrEmpty
            Phone = gcstrEmpty
            Address = gcstrEmpty
            Money = gcdecZero
            CreationDate = gcstrEmpty
            Status = gcstrEmpty
            SignedIn = False
        End Sub

        '''-------------------------------------------------------------------------------------------
-----------
```

```
        ''' <summary>   The SignIn subroutine of a User. </summary>
        '''
        ''' <remarks>   This subroutine changes a User object's properties to that of a user from
the database. </remarks>
        '''
        ''' <param name="intRecordID">
        ''' The unique ID of a record in the ACCOUNT table in the database.
        ''' </param>
        '''-------------------------------------------------------------------------------------
-----------

        Public Sub SignIn(ByVal intRecordID As Integer)
            ID = intRecordID
            Username = SQLGetFieldInfo(DatabaseTables.ACCOUNT, intRecordID, "ACC_USERNAME")
            Password = SQLGetFieldInfo(DatabaseTables.ACCOUNT, intRecordID, "ACC_PASSWORD")
            FirstName = SQLGetFieldInfo(DatabaseTables.ACCOUNT, intRecordID, "ACC_FNAME")
            LastName = SQLGetFieldInfo(DatabaseTables.ACCOUNT, intRecordID, "ACC_LNAME")
            Email = SQLGetFieldInfo(DatabaseTables.ACCOUNT, intRecordID, "ACC_EMAIL")
            Phone = SQLGetFieldInfo(DatabaseTables.ACCOUNT, intRecordID, "ACC_PHONE")
            Address = SQLGetFieldInfo(DatabaseTables.ACCOUNT, intRecordID, "ACC_ADDRESS")
            Money = Convert.ToDecimal(SQLGetFieldInfo(DatabaseTables.ACCOUNT, intRecordID,
"ACC_MONEY"))
            CreationDate = SQLGetFieldInfo(DatabaseTables.ACCOUNT, intRecordID,
"ACC_CREATION_DATE")
            Status = SQLGetFieldInfo(DatabaseTables.ACCOUNT, intRecordID, "ACC_STATUS")
            SignedIn = True
            Console.WriteLine("Signed in = " & gusrCurrentUser.SignedIn & " (Signed in as " &
gusrCurrentUser.Username & ")")
            RaiseEvent UserUpdated()
        End Sub

        '''-------------------------------------------------------------------------------------
-----------
        ''' <summary>   The SignOut subroutine of a User. </summary>
        '''
        ''' <remarks>   This subroutine sets the properties of a User to zeros and empty strings.
</remarks>
        '''-------------------------------------------------------------------------------------
-----------

        Public Sub SignOut()
            Dim strUserSigningOut As String = gusrCurrentUser.Username
            ID = gcintZero
            Username = gcstrEmpty
            Password = gcstrEmpty
            FirstName = gcstrEmpty
            LastName = gcstrEmpty
            Email = gcstrEmpty
            Phone = gcstrEmpty
            Address = gcstrEmpty
            Money = gcdecZero
            CreationDate = gcstrEmpty
            Status = gcstrEmpty
            SignedIn = False
            Console.WriteLine("Signed in = " & gusrCurrentUser.SignedIn & " (Signed out of " &
strUserSigningOut & ")")
            RaiseEvent UserUpdated()
        End Sub

    End Class

End Module
```

SQL Module

```
' Developed by Shawn Broyles
' The SQL module contains all of the methods that directly
' access and manipulate data in the database.

Option Strict On

Imports System.Data.SQLite

Module SQL

    '''------------------------------------------------------------------------------------------
-------
    ''' <summary>   Values that represent the tabls in the database. </summary>
    '''
    ''' <remarks>   The values are used for determining which table the program wants to
read/write to/from when it accesses the database. </remarks>
    '''------------------------------------------------------------------------------------------
-------

    Enum DatabaseTables
        ''' <summary>   An enum constant representing the ACCOUNT table. </summary>
        ACCOUNT
        ''' <summary>   An enum constant representing the PAYMENT table. </summary>
        PAYMENT
        ''' <summary>   An enum constant representing the PRODUCT table. </summary>
        PRODUCT
    End Enum

    '''------------------------------------------------------------------------------------------
-------
    ''' <summary>   Values that represent the different RegEx validations. </summary>
    '''
    ''' <remarks>   The values are used for matching regular expressions to use input. </remarks>
    '''------------------------------------------------------------------------------------------
-------

    Enum RegexValidate
        ''' <summary>   An enum constant representing the Username RegEx. </summary>
        USERNAME
        ''' <summary>   An enum constant representing the Password RegEx. </summary>
        PASSWORD
        ''' <summary>   An enum constant representing the Email RegEx. </summary>
        EMAIL
        ''' <summary>   An enum constant representing other data with allowing empty RegEx.
</summary>
        OTHER_EMPTY
        ''' <summary>   An enum constant representing the ID RegEx. </summary>
        ID
    End Enum

    ''' <summary>   The constant string variable for the name of the database. </summary>
    Const _cstrDatabaseName As String = "Database.db"
    ''' <summary>   The constant string variable for the connection. </summary>
    Const _cstrConnection As String = "Data Source=" & _cstrDatabaseName
    ''' <summary>   The constant string for allowing case insensitivity. </summary>
    Const _cstrCaseInsensitive As String = " COLLATE NOCASE"

    '''------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SQLInitializeDatabase subroutine. </summary>
    '''
    ''' <remarks>   This subroutine creates the database and sample data if appropriate.
</remarks>
```

```
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub SQLInitializeDatabase()
        SQLCreateDatabaseIfNotExists()
        SQLCreateGuestIfNotExists()
        SQLCreateSampleProductsIfNotExists()
        SQLGetProducts()
        CreateShoppingCart()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SQLCreateDatabaseIfNotExists subroutine. </summary>
    '''
    ''' <remarks>   This subroutine creates the database if it doesn't exist. </remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub SQLCreateDatabaseIfNotExists()
        Dim strSQLCreateDatabase As String = "CREATE TABLE IF NOT EXISTS ACCOUNT (
        ACC_ID              INTEGER PRIMARY KEY,
        ACC_USERNAME        TEXT UNIQUE NOT NULL,
        ACC_PASSWORD        TEXT NOT NULL,
        ACC_FNAME           TEXT DEFAULT '',
        ACC_LNAME           TEXT DEFAULT '',
        ACC_EMAIL           TEXT UNIQUE DEFAULT '',
        ACC_PHONE           TEXT DEFAULT '',
        ACC_ADDRESS         TEXT DEFAULT '',
        ACC_MONEY           TEXT DEFAULT '0.00',
        ACC_CREATION_DATE   DATE DEFAULT (datetime('now','localtime')),
        ACC_STATUS          TEXT DEFAULT 'Good'
        );
        CREATE TABLE IF NOT EXISTS PAYMENT (
        PAY_ID              INTEGER PRIMARY KEY,
        ACC_ID              INTEGER,
        PAY_DATE            DATE DEFAULT CURRENT_TIMESTAMP,
        PAY_QTY             INTEGER,
        PAY_PRICE           TEXT DEFAULT '0.00',
        FOREIGN KEY (ACC_ID) REFERENCES ACCOUNT (ACC_ID)
        );
        CREATE TABLE IF NOT EXISTS PRODUCT (
        PROD_ID             INTEGER PRIMARY KEY,
        PROD_NAME           TEXT DEFAULT '' UNIQUE,
        PROD_PRICE          TEXT DEFAULT '0.00',
        PROD_STOCK          INTEGER DEFAULT 0,
        PROD_FEE            TEXT DEFAULT '0.00',
        PROD_CATEGORY       TEXT DEFAULT '',
        PROD_DESCRIPTION    TEXT DEFAULT ''
        );"

        ' Creating the connection
        Dim connection As String = "Data Source=" & _cstrDatabaseName
        Dim SQLConn As New SQLiteConnection(connection)
        Dim SQLcmd As New SQLiteCommand(SQLConn)
        SQLConn.Open()

        SQLcmd.Connection = SQLConn

        ' Executing the SQL statements to create the database
        SQLcmd.CommandText = strSQLCreateDatabase
        SQLcmd.ExecuteNonQuery()

        ' Close the connection
        SQLConn.Close()
```

```
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Exists function. </summary>
    '''
    ''' <remarks>   This function determines if a record with a unique ID exists. </remarks>
    '''
    ''' <param name="intRecordID">
    ''' Unique ID for the record.
    ''' </param>
    '''
    ''' <returns>   True if the record exists, false otherwise. </returns>
    '''-------------------------------------------------------------------------------------------
-------

    Function Exists(ByVal intRecordID As Integer) As Boolean
        Dim blnExists As Boolean
        Dim intErrorID As Integer = -1

        If (intRecordID.Equals(intErrorID)) Then
            blnExists = False
        Else
            blnExists = True
        End If

        Return blnExists

    End Function

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SQLCreateSampleProductsIfNotExists subroutine. </summary>
    '''
    ''' <remarks>   This subroutine creates sample products in the database if they don't already
exist. </remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub SQLCreateSampleProductsIfNotExists()
        ' Creating new records in the PRODUCT table for sample products if they don't already
exist
        Try
            Dim intCounter As Integer = gcintZero

            ' Hard-coded sample products
            CreateSampleProduct(intCounter, "Pine Chair", 72D, 1, 4.5D, "Chair", "Our pine chairs
are created from the finest pine wood.")
            CreateSampleProduct(intCounter, "Maple Chair", 80D, 7, 5D, "Chair", "The maple chair
is a household favorite!")
            CreateSampleProduct(intCounter, "Oak Chair", 83.33D, 22, 4.55D, "Chair", "Oak never
goes out of style!")
            CreateSampleProduct(intCounter, "Pine Table", 141.5D, 3, 11.2D, "Table", "This pine
table is extremely sturdy.")
            CreateSampleProduct(intCounter, "Maple Table", 148.73D, 0, 13.1D, "Table", "Tables
made out of maple look good.")
            CreateSampleProduct(intCounter, "Oak Table", 162D, 4, 17.05D, "Table", "Our oak
tables are always well-built!")
            CreateSampleProduct(intCounter, "Pine Desk", 138.6D, 2, 11.1D, "Desk", "The pine
desk.")
            CreateSampleProduct(intCounter, "Maple Desk", 145D, 5, 12.7D, "Desk", "Maple desks
are reliable.")
            CreateSampleProduct(intCounter, "Oak Desk", 160.9D, 11, 13D, "Desk", "This oak desk
is smooth.")
            CreateSampleProduct(intCounter, "Leather Couch", 341.5D, 1, 51.28D, "Couch", "This
leather couch is water-resistant!")
```

```
            CreateSampleProduct(intCounter, "Silk Couch", 343.73D, 1, 53.62D, "Couch", "The silk
couch is easy to clean.")
            CreateSampleProduct(intCounter, "Wool Couch", 362.99D, 9, 77.2D, "Couch", "Our wool
couches are comfortable.")
            CreateSampleProduct(intCounter, "Linen Carpet", 421.22D, 2, 54D, "Carpet", "Linen
carpets are very popular.")
            CreateSampleProduct(intCounter, "Silk Carpet", 440D, 5, 55.99D, "Carpet", "This silk
carpet can withstand some wear and tear.")
            CreateSampleProduct(intCounter, "Wool Carpet", 442.99D, 8, 78.37D, "Carpet", "Our
wool carpets are stain-resistant.")

            If (intCounter.Equals(gcintZero)) Then
                Console.WriteLine("Sample products already exist in the database.")
            Else
                Console.WriteLine("Created " & intCounter.ToString() & " sample products in the
database.")
            End If

        Catch ex As Exception
            Console.WriteLine(ex.Message)
            Console.WriteLine("Unknown error occurred when trying to create sample products.")
        End Try
    End Sub

    '''------------------------------------------------------------------------------------------
-------
    ''' <summary>   The CreateSampleProduct subroutine. </summary>
    '''
    ''' <remarks>   This subroutine creates a sample product in the database. </remarks>
    '''
    ''' <param name="intCounter">
    ''' The counter for keeping track of how many sample products are being added to the
database.
    ''' </param>
    ''' <param name="strName">
    ''' The name of a sample product.
    ''' </param>
    ''' <param name="decPrice">
    ''' The price of a sample product.
    ''' </param>
    ''' <param name="intStock">
    ''' The stock of a sample product.
    ''' </param>
    ''' <param name="decFee">
    ''' The fee of a sample product.
    ''' </param>
    ''' <param name="strCategory">
    ''' The category of a sample product.
    ''' </param>
    ''' <param name="strDescription">
    ''' The description of a sample product.
    ''' </param>
    '''------------------------------------------------------------------------------------------
-------

    Private Sub CreateSampleProduct(ByRef intCounter As Integer,
                                    ByVal strName As String,
                                    ByVal decPrice As Decimal,
                                    ByVal intStock As Integer,
                                    ByVal decFee As Decimal,
                                    ByVal strCategory As String,
                                    ByVal strDescription As String)
        Dim strField As String = "PROD_NAME"
        Dim strFieldValue As String = strName
        If (Not Exists(SQLGetRecordID(DatabaseTables.PRODUCT, strField, strFieldValue))) Then
            SQLCreateProduct(strName, decPrice, intStock, decFee, strCategory, strDescription)
```

```
            intCounter += 1
        End If

    End Sub

    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   The SQLCreateProduct subroutine. </summary>
    '''
    ''' <remarks>   This subroutine creates a product in the database. </remarks>
    '''
    ''' <param name="strName">
    ''' The name of a product.
    ''' </param>
    ''' <param name="decPrice">
    ''' The price of a product.
    ''' </param>
    ''' <param name="intStock">
    ''' The stock of a product.
    ''' </param>
    ''' <param name="decFee">
    ''' The fee of a product.
    ''' </param>
    ''' <param name="strCategory">
    ''' The category of a product.
    ''' </param>
    ''' <param name="strDescription">
    ''' The description of a product.
    ''' </param>
    '''-----------------------------------------------------------------------------------------
-------

    Public Sub SQLCreateProduct(ByVal strName As String,
                                ByVal decPrice As Decimal,
                                ByVal intStock As Integer,
                                ByVal decFee As Decimal,
                                ByVal strCategory As String,
                                ByVal strDescription As String)

        Dim strSQLCreateProductRecord As String = "INSERT INTO PRODUCT (PROD_NAME, PROD_PRICE,
PROD_STOCK, PROD_FEE, PROD_CATEGORY, PROD_DESCRIPTION)
        VALUES('" & strName & "', '" & decPrice & "', " & intStock & ", '" & decFee & "', '" &
strCategory & "', '" & strDescription & "');"

        ' Creating the connection
        Dim connection As String = "Data Source=" & _cstrDatabaseName
        Dim SQLConn As New SQLiteConnection(connection)
        Dim SQLcmd As New SQLiteCommand(SQLConn)
        SQLConn.Open()

        SQLcmd.Connection = SQLConn

        ' Executing the SQL statement to create a new record in the ACCOUNT table
        SQLcmd.CommandText = strSQLCreateProductRecord
        SQLcmd.ExecuteNonQuery()

        ' Close the connection
        SQLConn.Close()
    End Sub

    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   The SQLCreatePayment subroutine. </summary>
    '''
    ''' <remarks>   This subroutine creates a new record in the database for information about a
payment. </remarks>
```

```
    '''
    ''' <param name="intAccountID">
    ''' The account ID that made the payment.
    ''' </param>
    ''' <param name="intQuantity">
    ''' The quantity of items for the payment.
    ''' </param>
    ''' <param name="decPrice">
    ''' The price for the payment.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub SQLCreatePayment(ByVal intAccountID As Integer,
                                ByVal intQuantity As Integer,
                                ByVal decPrice As Decimal)

        Dim strSQLCreatePaymentRecord As String = "INSERT INTO PAYMENT (ACC_ID, PAY_QTY,
PAY_PRICE)
        VALUES(" & intAccountID & ", " & intQuantity & ", '" & decPrice & "');"

        ' Creating the connection
        Dim connection As String = "Data Source=" & _cstrDatabaseName
        Dim SQLConn As New SQLiteConnection(connection)
        Dim SQLcmd As New SQLiteCommand(SQLConn)
        SQLConn.Open()

        SQLcmd.Connection = SQLConn

        ' Executing the SQL statement to create a new record in the ACCOUNT table
        SQLcmd.CommandText = strSQLCreatePaymentRecord
        SQLcmd.ExecuteNonQuery()

        ' Close the connection
        SQLConn.Close()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SQLCreateGuestIfNotExists subroutine. </summary>
    '''
    ''' <remarks>   This subroutine creates a guest account if it doesn't already exist.
</remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub SQLCreateGuestIfNotExists()
        ' Creating a new record in the ACCOUNT table for Guest if it doesn't already exist
        Try
            Dim strField As String = "ACC_USERNAME"
            Dim strFieldValue As String = "Guest"
            Dim intGuestRecordID As Integer = SQLGetRecordID(DatabaseTables.ACCOUNT, strField,
strFieldValue)
            If (Not Exists(intGuestRecordID)) Then
                SQLCreateAccount("Guest", "password", "Guest_FName", "Guest_LName",
"Guest@example.com", "+1 (234) 567-8901", "123 North Main St.")
                Console.WriteLine("Guest account created.")
            Else
                Console.WriteLine("Guest account already exists.")
            End If
        Catch ex As Exception
            Console.WriteLine(ex.Message)
            Console.WriteLine("Unknown error occurred when trying to create the Guest account.")
        End Try

    End Sub
```

```
    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SQLCreateAccount subroutine. </summary>
    '''
    ''' <remarks>   This subroutine creates a new account record in the database. </remarks>
    '''
    ''' <param name="strUsername">
    ''' The username of a new account.
    ''' </param>
    ''' <param name="strPassword">
    ''' The password of a new account.
    ''' </param>
    ''' <param name="strFName">
    ''' The first name of a new account.
    ''' </param>
    ''' <param name="strLName">
    ''' The last name of a new account.
    ''' </param>
    ''' <param name="strEmail">
    ''' The email for a new account.
    ''' </param>
    ''' <param name="strPhone">
    ''' The phone for a new account.
    ''' </param>
    ''' <param name="strAddress">
    ''' The address for a new account.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Public Sub SQLCreateAccount(ByVal strUsername As String,
                                ByVal strPassword As String,
                                ByVal strFName As String,
                                ByVal strLName As String,
                                ByVal strEmail As String,
                                ByVal strPhone As String,
                                ByVal strAddress As String)

        Dim strSQLCreateAccountRecord As String = "INSERT INTO ACCOUNT (ACC_USERNAME,
ACC_PASSWORD, ACC_FNAME, ACC_LNAME, ACC_EMAIL, ACC_PHONE, ACC_ADDRESS)
        VALUES('" & strUsername & "', '" & strPassword & "', '" & strFName & "', '" & strLName &
"', '" & strEmail & "', '" & strPhone & "', '" & strAddress & "');"

        ' Creating the connection
        Dim connection As String = "Data Source=" & _cstrDatabaseName
        Dim SQLConn As New SQLiteConnection(connection)
        Dim SQLcmd As New SQLiteCommand(SQLConn)
        SQLConn.Open()

        SQLcmd.Connection = SQLConn

        ' Executing the SQL statement to create a new record in the ACCOUNT table
        SQLcmd.CommandText = strSQLCreateAccountRecord
        SQLcmd.ExecuteNonQuery()

        ' Close the connection
        SQLConn.Close()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The RegexValidateUserData subroutine. </summary>
    '''
    ''' <remarks>   This subroutine validates user input so that bad data isn't used in a SQL
statement. </remarks>
```

```
    '''
    ''' <param name="strData">
    ''' The user input.
    ''' </param>
    ''' <param name="intDataType">
    ''' The type of data the user inputted.
    ''' </param>
    '''
    ''' <returns>   True if it the regular expression matches the user input, false otherwise.
</returns>
    '''-------------------------------------------------------------------------------------------
-------

    Function RegexValidateUserData(ByVal strData As String, ByVal intDataType As Integer) As
Boolean
        Dim blnReturn As Boolean
        Dim strAllowedCharactersRegex As String
        Select Case intDataType
            Case RegexValidate.USERNAME
                strAllowedCharactersRegex = "^[A-Za-z][A-Za-z0-9]{3,}$"
            Case RegexValidate.PASSWORD
                strAllowedCharactersRegex = "^[A-Za-z0-9@.\-_\s]{6,}$"
            Case RegexValidate.EMAIL
                strAllowedCharactersRegex = "^[A-Za-z0-9.\-_]+@[A-Za-z0-9.\-_]+.[A-Za-z0-9.\-
_]+$"
            Case RegexValidate.OTHER_EMPTY
                strAllowedCharactersRegex = "^[A-Za-z0-9@.\-_\s\(\)]*$"
            Case RegexValidate.ID
                strAllowedCharactersRegex = "^[0-9]+$"
            Case Else
                strAllowedCharactersRegex = "^[A-Za-z0-9@.\-_\s\(\)]+$"
        End Select
        If (System.Text.RegularExpressions.Regex.IsMatch(strData, strAllowedCharactersRegex))
Then
            blnReturn = True
        Else
            blnReturn = False
        End If
        Return blnReturn
    End Function

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SQLGetFieldInfo subroutine. </summary>
    '''
    ''' <remarks>   This subroutine reads information for a field from the database. </remarks>
    '''
    ''' <param name="intTable">
    ''' An integer representing the table in the database that is being accessed.
    ''' </param>
    ''' <param name="intPrimaryKeyID">
    ''' The unique ID of a record.
    ''' </param>
    ''' <param name="strField">
    ''' The field in the database that is being read.
    ''' </param>
    ''' <param name="blnCaseSensitive">
    ''' (Optional) True for using case sensitivity in the SQL statement, false otherwise.
    ''' </param>
    '''
    ''' <returns>   The value of the field found. </returns>
    '''-------------------------------------------------------------------------------------------
-------

    Function SQLGetFieldInfo(ByVal intTable As Integer, ByVal intPrimaryKeyID As Integer, ByVal
strField As String, Optional ByVal blnCaseSensitive As Boolean = False) As String
```

```
        Const cstrError As String = "Error - Unable to get field info from the database."

        Dim strReturn As String = gcstrEmpty

        Dim strTable As String = gcstrEmpty
        Dim strPrimaryKey As String = gcstrEmpty
        Dim strCaseSensitivity As String = gcstrEmpty

        If (blnCaseSensitive) Then
            ' SQL queries are case sensitive by default
        Else
            strCaseSensitivity = _cstrCaseInsensitive
        End If

        Select Case intTable
            Case DatabaseTables.ACCOUNT
                strTable = "ACCOUNT"
                strPrimaryKey = "ACC_ID"
            Case DatabaseTables.PAYMENT
                strTable = "PAYMENT"
                strPrimaryKey = "PAY_ID"
            Case DatabaseTables.PRODUCT
                strTable = "PRODUCT"
                strPrimaryKey = "PROD_ID"
        End Select

        If (String.IsNullOrWhiteSpace(strTable) Or String.IsNullOrWhiteSpace(strPrimaryKey)) Then
            Console.WriteLine(cstrError)
            strReturn = cstrError
        Else

            Dim strFieldInfo As String = gcstrEmpty

            Dim SQLstr As String = "SELECT * FROM " & strTable & " WHERE " & strPrimaryKey & "="
& intPrimaryKeyID.ToString() & strCaseSensitivity & ";"

            Dim SQLConn As New SQLiteConnection(_cstrConnection)
            Dim SQLcmd As New SQLiteCommand(SQLConn)
            Dim SQLdr As SQLiteDataReader
            SQLConn.Open()

            SQLcmd.Connection = SQLConn
            SQLcmd.CommandText = SQLstr
            SQLdr = SQLcmd.ExecuteReader()

            ' Loop through all records returned
            While SQLdr.Read()
                Try
                    strFieldInfo = (SQLdr.GetValue(SQLdr.GetOrdinal(strField))).ToString()
                Catch ex As IndexOutOfRangeException
                    strFieldInfo = "Not found"
                    Console.WriteLine(ex.Message)
                    Console.WriteLine("Invalid column name (" & strField & ") in " & strTable &
".")
                Catch ex As InvalidCastException
                    strFieldInfo = "Unknown"
                    Console.WriteLine(ex.Message)
                    Console.WriteLine("Invalid cast when executing a SQLiteDataReader method.")
                End Try
            End While

            ' Close the connection
            SQLdr.Close()
            SQLConn.Close()
```

```
            strReturn = strFieldInfo

        End If

        Return strReturn

    End Function

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SQLGetRecordID subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for getting the record ID of a record from its
specific value in a field. </remarks>
    '''
    ''' <param name="intTable">
    ''' The table in the database that is being accessed.
    ''' </param>
    ''' <param name="strField">
    ''' The field in the record.
    ''' </param>
    ''' <param name="strFieldValue">
    ''' The specific value for the field.
    ''' </param>
    ''' <param name="blnCaseSensitive">
    ''' (Optional) True for using case sensitivity in the SQL statement, false otherwise.
    ''' </param>
    '''
    ''' <returns>   The record ID of a record that has a specific value in a field. </returns>
    '''-------------------------------------------------------------------------------------------
-------

    Function SQLGetRecordID(ByVal intTable As Integer, ByVal strField As String, ByVal
strFieldValue As String, Optional ByVal blnCaseSensitive As Boolean = False) As Integer

        Dim intErrorID As Integer = -1
        Dim intRecordID As Integer = intErrorID

        Dim strTable As String = gcstrEmpty
        Dim strPrimaryKey As String = gcstrEmpty
        Dim strCaseSensitivity As String = gcstrEmpty

        If (blnCaseSensitive) Then
            ' SQL queries are case sensitive by default
        Else
            strCaseSensitivity = _cstrCaseInsensitive
        End If

        Select Case intTable
            Case DatabaseTables.ACCOUNT
                strTable = "ACCOUNT"
                strPrimaryKey = "ACC_ID"
            Case DatabaseTables.PAYMENT
                strTable = "PAYMENT"
                strPrimaryKey = "PAY_ID"
            Case DatabaseTables.PRODUCT
                strTable = "PRODUCT"
                strPrimaryKey = "PROD_ID"
        End Select

        If (String.IsNullOrWhiteSpace(strTable)) Then
            Console.WriteLine("Error: Table not found from intTable")
        Else

            Dim SQLstr As String = "SELECT * FROM " & strTable & " WHERE " & strField & "='" &
strFieldValue & "'" & strCaseSensitivity & ";"
```

```
            Dim SQLConn As New SQLiteConnection(_cstrConnection)
            Dim SQLcmd As New SQLiteCommand(SQLConn)
            Dim SQLdr As SQLiteDataReader
            SQLConn.Open()

            SQLcmd.Connection = SQLConn
            SQLcmd.CommandText = SQLstr
            SQLdr = SQLcmd.ExecuteReader()

            ' Loop through all records returned
            While SQLdr.Read()
                Try
                    intRecordID =
Convert.ToInt32(SQLdr.GetValue(SQLdr.GetOrdinal(strPrimaryKey)))
                Catch ex As IndexOutOfRangeException
                    Console.WriteLine(ex.Message)
                    Console.WriteLine("Invalid column name (" & strField & ") in " & strTable &
". [2]")
                Catch ex As InvalidCastException
                    Console.WriteLine(ex.Message)
                    Console.WriteLine("Invalid cast when executing a SQLiteDataReader method.
[2]")
                End Try
            End While

            ' Close the connection
            SQLdr.Close()
            SQLConn.Close()

        End If

        Return intRecordID

    End Function

    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   The SQLSetFieldInfo subroutine. </summary>
    '''
    ''' <remarks>   This subroutine sets the value of a field in a record in the database.
</remarks>
    '''
    ''' <param name="intTable">
    ''' The table in the database being accessed.
    ''' </param>
    ''' <param name="intPrimaryKeyID">
    ''' The unique ID of a record.
    ''' </param>
    ''' <param name="strField">
    ''' The field whose value is being changed.
    ''' </param>
    ''' <param name="strFieldNewValue">
    ''' The new value for a field.
    ''' </param>
    ''' <param name="blnCaseSensitive">
    ''' (Optional) True for using case sensitivity in the SQL statement, false otherwise.
    ''' </param>
    '''-----------------------------------------------------------------------------------------
-------

    Public Sub SQLSetFieldInfo(ByVal intTable As Integer, ByVal intPrimaryKeyID As Integer, ByVal
strField As String, ByVal strFieldNewValue As String, Optional ByVal blnCaseSensitive As Boolean
= False)
        Dim intErrorID As Integer = -1
        Dim intRecordID As Integer = intErrorID
```

```
        Dim strTable As String = gcstrEmpty
        Dim strPrimaryKey As String = gcstrEmpty
        Dim strCaseSensitivity As String = gcstrEmpty

        If (blnCaseSensitive) Then
            ' SQL queries are case sensitive by default
        Else
            strCaseSensitivity = _cstrCaseInsensitive
        End If

        Select Case intTable
            Case DatabaseTables.ACCOUNT
                strTable = "ACCOUNT"
                strPrimaryKey = "ACC_ID"
            Case DatabaseTables.PAYMENT
                strTable = "PAYMENT"
                strPrimaryKey = "PAY_ID"
            Case DatabaseTables.PRODUCT
                strTable = "PRODUCT"
                strPrimaryKey = "PROD_ID"
        End Select

        If (String.IsNullOrWhiteSpace(strTable)) Then
            Console.WriteLine("Error: Table not found from intTable")
        Else
            Dim SQLstr As String = "UPDATE " & strTable & "
            SET " & strField & " = '" & strFieldNewValue & "'
            WHERE " & strPrimaryKey & " = " & intPrimaryKeyID & strCaseSensitivity & ";"

            Dim SQLConn As New SQLiteConnection(_cstrConnection)
            Dim SQLcmd As New SQLiteCommand(SQLConn)
            Dim SQLdr As SQLiteDataReader
            SQLConn.Open()

            SQLcmd.Connection = SQLConn
            SQLcmd.CommandText = SQLstr
            SQLdr = SQLcmd.ExecuteReader()

            ' Close the connection
            SQLdr.Close()
            SQLConn.Close()

        End If

    End Sub

    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   The SQLGetProducts subroutine. </summary>
    '''
    ''' <remarks>   This subroutine reads all of the record from the PRODUCT table and stores
values in memory. </remarks>
    '''-----------------------------------------------------------------------------------------
-------

    Public Sub SQLGetProducts()
        ' Reading products from the database, creating Items, and adding the Items
        ' to the glstProducts list
        Dim SQLstr As String = "SELECT * FROM PRODUCT;"

        Dim SQLConn As New SQLiteConnection(_cstrConnection)
        Dim SQLcmd As New SQLiteCommand(SQLConn)
        Dim SQLdr As SQLiteDataReader
        SQLConn.Open()
```

```
        SQLcmd.Connection = SQLConn
        SQLcmd.CommandText = SQLstr
        SQLdr = SQLcmd.ExecuteReader()

        ' Loop through all records returned
        While SQLdr.Read()
            Try
                Dim itmNewItem As Item = New Item() With {
                    .ID =
Convert.ToInt32(SQLdr.GetValue(SQLdr.GetOrdinal("PROD_ID")).ToString()),
                    .Name = SQLdr.GetValue(SQLdr.GetOrdinal("PROD_NAME")).ToString(),
                    .Price =
Convert.ToDecimal(SQLdr.GetValue(SQLdr.GetOrdinal("PROD_PRICE")).ToString()),
                    .Stock =
Convert.ToInt32(SQLdr.GetValue(SQLdr.GetOrdinal("PROD_STOCK")).ToString()),
                    .Fee =
Convert.ToDecimal(SQLdr.GetValue(SQLdr.GetOrdinal("PROD_FEE")).ToString()),
                    .Category = SQLdr.GetValue(SQLdr.GetOrdinal("PROD_CATEGORY")).ToString(),
                    .Description =
SQLdr.GetValue(SQLdr.GetOrdinal("PROD_DESCRIPTION")).ToString()
                }
                AddToProducts(itmNewItem)
            Catch ex As IndexOutOfRangeException
                Console.WriteLine(ex.Message)
                Console.WriteLine("Invalid column name.")
            Catch ex As InvalidCastException
                Console.WriteLine(ex.Message)
                Console.WriteLine("Invalid cast when executing a SQLiteDataReader method.")
            End Try
        End While

        ' Close the connection
        SQLdr.Close()
        SQLConn.Close()

    End Sub

End Module
```

Welcome Form

```
'''-----------------------------------------------------------------------------------------------
---
''' <summary>   The Welcome form. </summary>
'''
''' <remarks>   This form is the application's main form. It's used for letting customers know
they're welcome to use the application and it gives options for signing in. </remarks>
'''-----------------------------------------------------------------------------------------------
---

Public Class frmWelcome

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   Gets the required creation parameters when the control handle is created. The
ClassStyle property is overridden to disable the Close button. </summary>
    '''
    ''' <value>
    ''' A <see cref="T:System.Windows.Forms.CreateParams" /> that contains the required creation
    ''' parameters when the handle to the control is created.
    ''' </value>
    '''-------------------------------------------------------------------------------------------
-------

    Protected Overloads Overrides ReadOnly Property CreateParams() As CreateParams
```

```
        Get
            Dim myParams As CreateParams = MyBase.CreateParams
            Const cintStyleNoCloseButton As Integer = 512
            myParams.ClassStyle = cintStyleNoCloseButton
            Return myParams
        End Get
    End Property

    ''' <summary>   The constant integer for the current form. </summary>
    Const _cintForm As Integer = Forms.WELCOME

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Welcome ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Welcome form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuWelcome_Click(sender As Object, e As EventArgs) Handles mnuWelcome.Click
        Navigate(Forms.WELCOME, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Registration ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Registration form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuRegistration_Click(sender As Object, e As EventArgs) Handles
mnuRegistration.Click
        Navigate(Forms.REGISTRATION, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Login ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Login form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------
```

```
    Private Sub mnuLogin_Click(sender As Object, e As EventArgs) Handles mnuLogin.Click
        Navigate(Forms.LOGIN, Me)
    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Catalog ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Catalog form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-----------------------------------------------------------------------------------------------
-------

    Private Sub mnuCatalog_Click(sender As Object, e As EventArgs) Handles mnuCatalog.Click
        Navigate(Forms.CATALOG, Me)
    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Checkout ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Checkout form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-----------------------------------------------------------------------------------------------
-------

    Private Sub mnuCheckout_Click(sender As Object, e As EventArgs) Handles mnuCheckout.Click
        Navigate(Forms.CHECKOUT, Me)
    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Account ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Account form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-----------------------------------------------------------------------------------------------
-------

    Private Sub mnuAccount_Click(sender As Object, e As EventArgs) Handles mnuAccount.Click
        Navigate(Forms.ACCOUNT, Me)
    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Item ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Item form. </remarks>
```

```
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuItem_Click(sender As Object, e As EventArgs) Handles mnuItem.Click
        Navigate(Forms.ITEM, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Payment ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Payment form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPayment_Click(sender As Object, e As EventArgs) Handles mnuPayment.Click
        Navigate(Forms.PAYMENT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Reload ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for reloading the current form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuReload_Click(sender As Object, e As EventArgs) Handles mnuReload.Click
        ReloadForm(Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SignOut ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for signing out of an account. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------
```

```
    Private Sub mnuSignOut_Click(sender As Object, e As EventArgs) Handles mnuSignOut.Click
        SignOut()
    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Exit ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for terminating the application. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-----------------------------------------------------------------------------------------------
-------

    Private Sub mnuExit_Click(sender As Object, e As EventArgs) Handles mnuExit.Click
        ExitApplication()
    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
    ''' <summary>   The About ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a popup that tells the user some information
about the current form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-----------------------------------------------------------------------------------------------
-------

    Private Sub mnuAbout_Click(sender As Object, e As EventArgs) Handles mnuAbout.Click
        AboutApplication(_cintForm)
    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Print ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a print preview of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-----------------------------------------------------------------------------------------------
-------

    Private Sub mnuPrint_Click(sender As Object, e As EventArgs) Handles mnuPrint.Click
        PrintForm(pfPrintForm)
    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
```

```
    ''' <summary>   The frmWelcome_Load subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for loading the defaults of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub frmWelcome_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadFormDefaults(Me)
        SQLInitializeDatabase()
        AddHandler gusrCurrentUser.UserUpdated, AddressOf UpdateSignedIn
        Console.WriteLine("glstProducts has " & glstProducts.Count() & " items.")
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Register button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for navigating to the Registration form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnRegister_Click(sender As Object, e As EventArgs) Handles btnRegister.Click
        Navigate(Forms.REGISTRATION, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Sign In button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for navigating to the Login form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnSignIn_Click(sender As Object, e As EventArgs) Handles btnSignIn.Click
        Navigate(Forms.LOGIN, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Sign In As Guest button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for signing the user into the guest account and
navigating to the catalog form. </remarks>
    '''
```

```
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnSignInAsGuest_Click(sender As Object, e As EventArgs) Handles
btnSignInAsGuest.Click
        SignInAsGuest()
        Navigate(Forms.CATALOG, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Sign Out button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for signing out of an account. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnSignOut_Click(sender As Object, e As EventArgs) Handles btnSignOut.Click
        SignOut()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The UpdateSignedIn subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for changing objects' properties for when the user
signs in and out. </remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub UpdateSignedIn()
        Const cstrSignedInFormat As String = "Currently Signed In: {0}"
        Dim strSignedInUsername As String
        If (gusrCurrentUser.SignedIn) Then
            lblCurrentlySignedIn.ForeColor = Color.ForestGreen
            btnRegister.Enabled = False
            btnSignIn.Enabled = False
            btnSignInAsGuest.Enabled = False
            btnSignOut.Enabled = True
            strSignedInUsername = gusrCurrentUser.Username
        Else
            lblCurrentlySignedIn.ForeColor = Color.PaleVioletRed
            btnRegister.Enabled = True
            btnSignIn.Enabled = True
            btnSignInAsGuest.Enabled = True
            btnSignOut.Enabled = False
            strSignedInUsername = "N/A"
        End If
        lblCurrentlySignedIn.Text = String.Format(cstrSignedInFormat, strSignedInUsername)
    End Sub
End Class
```

Registration Form

```
'''-------------------------------------------------------------------------------------------
---
''' <summary>   The Registration form. </summary>
'''
''' <remarks>   This form is used by customers to create accounts. </remarks>
'''-------------------------------------------------------------------------------------------
---

Public Class frmRegistration
    ''' <summary>   The constant integer for the current form. </summary>
    Const _cintForm As Integer = Forms.REGISTRATION

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Welcome ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Welcome form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuWelcome_Click(sender As Object, e As EventArgs) Handles mnuWelcome.Click
        Navigate(Forms.WELCOME, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Registration ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Registration form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuRegistration_Click(sender As Object, e As EventArgs) Handles
mnuRegistration.Click
        Navigate(Forms.REGISTRATION, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Login ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Login form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
```

```
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuLogin_Click(sender As Object, e As EventArgs) Handles mnuLogin.Click
        Navigate(Forms.LOGIN, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Catalog ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Catalog form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCatalog_Click(sender As Object, e As EventArgs) Handles mnuCatalog.Click
        Navigate(Forms.CATALOG, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Checkout ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Checkout form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCheckout_Click(sender As Object, e As EventArgs) Handles mnuCheckout.Click
        Navigate(Forms.CHECKOUT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Account ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Account form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAccount_Click(sender As Object, e As EventArgs) Handles mnuAccount.Click
        Navigate(Forms.ACCOUNT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
```

```
    ''' <summary>   The Item ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Item form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuItem_Click(sender As Object, e As EventArgs) Handles mnuItem.Click
        Navigate(Forms.ITEM, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Payment ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Payment form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPayment_Click(sender As Object, e As EventArgs) Handles mnuPayment.Click
        Navigate(Forms.PAYMENT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Reload ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for reloading the current form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuReload_Click(sender As Object, e As EventArgs) Handles mnuReload.Click
        ReloadForm(Me)
        ClearForm()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SignOut ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for signing out of an account. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
```

```
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuSignOut_Click(sender As Object, e As EventArgs) Handles mnuSignOut.Click
        SignOut()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Exit ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for terminating the application. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuExit_Click(sender As Object, e As EventArgs) Handles mnuExit.Click
        ExitApplication()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The About ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a popup that tells the user some information
about the current form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAbout_Click(sender As Object, e As EventArgs) Handles mnuAbout.Click
        AboutApplication(_cintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Print ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a print preview of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPrint_Click(sender As Object, e As EventArgs) Handles mnuPrint.Click
        PrintForm(pfPrintForm)
```

```
    End Sub

    '''------------------------------------------------------------------------------------------
-------
    ''' <summary>   The frmRegistration_Load subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for loading the defaults of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''------------------------------------------------------------------------------------------
-------

    Private Sub frmRegistration_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadFormDefaults(Me)
        txtUsername.Focus()
    End Sub

    '''------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Submit button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for creating a new account in the database.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''------------------------------------------------------------------------------------------
-------

    Private Sub btnSubmit_Click(sender As Object, e As EventArgs) Handles btnSubmit.Click
        Const cstrSuccessTitle As String = "Success"
        Const cstrSuccessMessage As String = "Account successfully created!"
        Const cstrErrorTitle As String = "Error"
        Const cstrBaseErrorMessage As String = "Validation for the following fields failed:"
        Const cstrUnknownErrorMessage As String = "An unknown error has occurred when trying to
create an account."
        Dim strErrorMessage As String = cstrBaseErrorMessage

        Dim strUsername As String = txtUsername.Text
        Dim strPassword As String = txtPassword.Text
        Dim strConfirmPassword As String = txtConfirmPassword.Text
        Dim strFName As String = txtFirstName.Text
        Dim strLName As String = txtLastName.Text
        Dim strEmail As String = txtEmail.Text
        Dim strPhone As String = txtPhoneNumber.Text
        Dim strAddress As String = txtAddress.Text

        Dim blnUsernameValidated As Boolean = RegexValidateUserData(strUsername,
RegexValidate.USERNAME)
        Dim blnPasswordValidated As Boolean = RegexValidateUserData(strPassword,
RegexValidate.PASSWORD)
        Dim blnFNameValidated As Boolean = RegexValidateUserData(strFName,
RegexValidate.OTHER_EMPTY)
        Dim blnLNameValidated As Boolean = RegexValidateUserData(strLName,
RegexValidate.OTHER_EMPTY)
        Dim blnEmailValidated As Boolean = RegexValidateUserData(strEmail, RegexValidate.EMAIL)
```

```
        Dim blnPhoneValidated As Boolean = RegexValidateUserData(strPhone,
RegexValidate.OTHER_EMPTY)
        Dim blnAddressValidated As Boolean = RegexValidateUserData(strAddress,
RegexValidate.OTHER_EMPTY)

        If (Not blnUsernameValidated) Then
            strErrorMessage = strErrorMessage & vbCrLf & "Username must be 4-16 characters, start
with a letter, and only contain A-Z a-z 0-9"
        End If
        If (Not blnPasswordValidated) Then
            strErrorMessage = strErrorMessage & vbCrLf & "Password must be 6-20 characters and
only contain A-Z a-z 0-9 @ . - _ "
        End If
        If (Not blnFNameValidated) Then
            strErrorMessage = strErrorMessage & vbCrLf & "First Name is optional. Only contains
spaces and A-Z a-z 0-9 @ . - _ ( )"
        End If
        If (Not blnLNameValidated) Then
            strErrorMessage = strErrorMessage & vbCrLf & "Last Name is optional. Only contains
spaces and A-Z a-z 0-9 @ . - _ ( )"
        End If
        If (Not blnEmailValidated) Then
            strErrorMessage = strErrorMessage & vbCrLf & "Email must be 5-50 characters and
contain @ and . characters"
        End If
        If (Not blnPhoneValidated) Then
            strErrorMessage = strErrorMessage & vbCrLf & "Phone Number is optional. Only contains
spaces and A-Z a-z 0-9 @ . - _ ( )"
        End If
        If (Not blnAddressValidated) Then
            strErrorMessage = strErrorMessage & vbCrLf & "Address is optional. Onlys contain
spaces and A-Z a-z 0-9 @ . - _ ( )"
        End If
        If (Not strPassword.Equals(strConfirmPassword)) Then
            strErrorMessage = strErrorMessage & vbCrLf & "Password and Confirm Password fields do
not match"
        End If

        Const cstrEmailField As String = "ACC_EMAIL"
        Const cstrUsernameField As String = "ACC_USERNAME"
        Const cintMissingRecordID As Integer = -1
        Dim intExistingUsernameRecordID As Integer = SQLGetRecordID(DatabaseTables.ACCOUNT,
cstrUsernameField, strUsername)
        Dim intExistingEmailRecordID As Integer = SQLGetRecordID(DatabaseTables.ACCOUNT,
cstrEmailField, strEmail)

        If (Not intExistingUsernameRecordID.Equals(cintMissingRecordID)) Then
            strErrorMessage = strErrorMessage & vbCrLf & "Username already exists in the
database"
        End If
        If (Not intExistingEmailRecordID.Equals(cintMissingRecordID)) Then
            strErrorMessage = strErrorMessage & vbCrLf & "Email already exists in the database"
        End If

        If (strErrorMessage.Equals(cstrBaseErrorMessage)) Then
            Try
                SQLCreateAccount(strUsername, strPassword, strFName, strLName, strEmail,
strPhone, strAddress)
                gusrCurrentUser.SignIn(SQLGetRecordID(DatabaseTables.ACCOUNT, cstrUsernameField,
strUsername))
                MsgBox(cstrSuccessMessage, , cstrSuccessTitle)
                ClearForm()
                Navigate(Forms.CATALOG, Me)
            Catch ex As Exception
                Console.WriteLine(ex.Message)
                MsgBox(cstrUnknownErrorMessage, , cstrErrorTitle)
```

```
            End Try

        Else
            MsgBox(strErrorMessage, , cstrErrorTitle)
        End If
    End Sub

    '''------------------------------------------------------------------------------------------
-------
    ''' <summary>   The ClearForm subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for clearing the form. </remarks>
    '''------------------------------------------------------------------------------------------
-------

    Private Sub ClearForm()
        txtUsername.Clear()
        txtPassword.Clear()
        txtConfirmPassword.Clear()
        txtFirstName.Clear()
        txtLastName.Clear()
        txtEmail.Clear()
        txtPhoneNumber.Clear()
        txtAddress.Clear()
        txtUsername.Focus()
    End Sub

    '''------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Clear button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for clearing the form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''------------------------------------------------------------------------------------------
-------

    Private Sub btnClear_Click(sender As Object, e As EventArgs) Handles btnClear.Click
        ' Clearing textboxes and setting the focus to txtUsername
        ClearForm()
        txtUsername.Focus()
    End Sub
End Class
```

Login Form

```
'''------------------------------------------------------------------------------------------
---
''' <summary>   The Login form. </summary>
'''
''' <remarks>   This form is used for logging into existing accounts. </remarks>
'''------------------------------------------------------------------------------------------
---

Public Class frmLogin

    ''' <summary>   The constant integer for the current form. </summary>
    Const _cintForm As Integer = Forms.LOGIN
```

```
    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Welcome ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Welcome form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuWelcome_Click(sender As Object, e As EventArgs) Handles mnuWelcome.Click
        Navigate(Forms.WELCOME, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Registration ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Registration form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuRegistration_Click(sender As Object, e As EventArgs) Handles
mnuRegistration.Click
        Navigate(Forms.REGISTRATION, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Login ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Login form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuLogin_Click(sender As Object, e As EventArgs) Handles mnuLogin.Click
        Navigate(Forms.LOGIN, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Catalog ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Catalog form. </remarks>
    '''
    ''' <param name="sender">
```

```
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCatalog_Click(sender As Object, e As EventArgs) Handles mnuCatalog.Click
        Navigate(Forms.CATALOG, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Checkout ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Checkout form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCheckout_Click(sender As Object, e As EventArgs) Handles mnuCheckout.Click
        Navigate(Forms.CHECKOUT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Account ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Account form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAccount_Click(sender As Object, e As EventArgs) Handles mnuAccount.Click
        Navigate(Forms.ACCOUNT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Item ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Item form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuItem_Click(sender As Object, e As EventArgs) Handles mnuItem.Click
```

```
        Navigate(Forms.ITEM, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Payment ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Payment form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPayment_Click(sender As Object, e As EventArgs) Handles mnuPayment.Click
        Navigate(Forms.PAYMENT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Reload ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for reloading the current form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuReload_Click(sender As Object, e As EventArgs) Handles mnuReload.Click
        ClearForm()
        ReloadForm(Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SignOut ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for signing out of an account. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuSignOut_Click(sender As Object, e As EventArgs) Handles mnuSignOut.Click
        SignOut()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Exit ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for terminating the application. </remarks>
```

```
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuExit_Click(sender As Object, e As EventArgs) Handles mnuExit.Click
        ExitApplication()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The About ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a popup that tells the user some information
about the current form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAbout_Click(sender As Object, e As EventArgs) Handles mnuAbout.Click
        AboutApplication(_cintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Print ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a print preview of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPrint_Click(sender As Object, e As EventArgs) Handles mnuPrint.Click
        PrintForm(pfPrintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The frmLogin_Load subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for loading the defaults of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
```

```
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub frmLogin_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadFormDefaults(Me)
        ClearForm()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Submit button subroutine. </summary>
    '''
    ''' <remarks>   This button is used for submitting the form and logging into an account.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnSubmit_Click(sender As Object, e As EventArgs) Handles btnSubmit.Click
        Const cstrSuccessTitle As String = "Success"
        Const cstrSuccessMessage As String = "Signed in successfully!"
        Const cstrErrorTitle As String = "Error"
        Const cstrBaseErrorMessage As String = "Validation for the following fields failed:"
        Const cstrUnknownErrorMessage As String = "An unknown error has occurred when trying to
login."
        Dim strErrorMessage As String = cstrBaseErrorMessage

        Dim strUsername As String = txtUsername.Text
        Dim strPassword As String = txtPassword.Text

        Const cstrEmailField As String = "ACC_EMAIL"
        Const cstrAccountIDField As String = "ACC_ID"
        Const cstrUsernameField As String = "ACC_USERNAME"
        Const cstrPasswordField As String = "ACC_PASSWORD"
        Const cintMissingRecordID As Integer = -1

        Dim strLoginFieldType As String
        Dim strLoginField As String
        Dim intValidateType As Integer

        If (rbEmail.Checked) Then
            strLoginFieldType = "Email"
            strLoginField = cstrEmailField
            intValidateType = RegexValidate.EMAIL
        ElseIf (rbAccountID.Checked) Then
            strLoginFieldType = "Account ID"
            strLoginField = cstrAccountIDField
            intValidateType = RegexValidate.ID
        Else
            strLoginFieldType = "Username"
            strLoginField = cstrUsernameField
            intValidateType = RegexValidate.USERNAME
        End If

        Dim blnUsernameValidated As Boolean = RegexValidateUserData(strUsername, intValidateType)
        Dim blnPasswordValidated As Boolean = RegexValidateUserData(strPassword,
RegexValidate.PASSWORD)

        If (Not blnUsernameValidated) Then
```

```
            Select Case intValidateType
                Case RegexValidate.EMAIL
                    strErrorMessage = strErrorMessage & vbCrLf & strLoginFieldType & "s only
contain A-Z a-z 0-9 . - _ @"
                Case RegexValidate.ID
                    strErrorMessage = strErrorMessage & vbCrLf & strLoginFieldType & "s only
contain 0-9"
                Case Else
                    strErrorMessage = strErrorMessage & vbCrLf & strLoginFieldType & "s start
with a letter and only contain A-Z a-z 0-9"
            End Select
        End If
        If (Not blnPasswordValidated) Then
            strErrorMessage = strErrorMessage & vbCrLf & "Password field must only contain A-Z a-
z 0-9 @ . - _ "
        End If

        Dim blnSignedIn As Boolean = False
        Dim blnAttemptSignIn As Boolean = False
        Dim blnPasswordCaseSensitive As Boolean = True

        If (strErrorMessage.Equals(cstrBaseErrorMessage)) Then
            Try
                blnAttemptSignIn = True
                strErrorMessage = "Logging in failed for the following reason(s):"
                Dim intUsernameRecordID As Integer = SQLGetRecordID(DatabaseTables.ACCOUNT,
strLoginField, strUsername)
                If (Not intUsernameRecordID.Equals(cintMissingRecordID)) Then
                    If (SQLGetFieldInfo(DatabaseTables.ACCOUNT, intUsernameRecordID,
cstrPasswordField, blnPasswordCaseSensitive).Equals(strPassword)) Then
                        gusrCurrentUser.SignIn(intUsernameRecordID)
                        blnSignedIn = True
                        MsgBox(cstrSuccessMessage, , cstrSuccessTitle)
                        ClearForm()
                        Navigate(Forms.CATALOG, Me)
                    Else
                        strErrorMessage = strErrorMessage & vbCrLf & "Invalid Password"
                    End If
                Else
                    strErrorMessage = strErrorMessage & vbCrLf & strLoginFieldType & " doesn't
exist"
                End If
            Catch ex As Exception
                Console.WriteLine(ex.Message)
                MsgBox(cstrUnknownErrorMessage, , cstrErrorTitle)
            End Try

        Else
            MsgBox(strErrorMessage, , cstrErrorTitle)
        End If
        If (blnAttemptSignIn And Not blnSignedIn) Then
            ' Username/Password combo was incorrect
            MsgBox(strErrorMessage, , cstrErrorTitle)
        End If
    End Sub

    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   The ClearForm subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for clearing the text on the form. </remarks>
    '''-----------------------------------------------------------------------------------------
-------

    Private Sub ClearForm()
        txtUsername.Clear()
```

```
        txtPassword.Clear()
        rbUsername.Select()
        txtUsername.Focus()
    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Clear button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for clearing the text on the form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-----------------------------------------------------------------------------------------------
-------

    Private Sub btnClear_Click(sender As Object, e As EventArgs) Handles btnClear.Click
        ' Clearing textboxes and setting the focus to txtUsername
        ClearForm()
    End Sub
End Class
```

Catalog Form

```
'''-----------------------------------------------------------------------------------------------
---
''' <summary>   The Catalog form. </summary>
'''
''' <remarks>   This form is used for browsing the catalog. </remarks>
'''-----------------------------------------------------------------------------------------------
---

Public Class frmCatalog

    ''' <summary>   The constant integer for the current form. </summary>
    Const _cintForm As Integer = Forms.CATALOG

    '''-----------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Welcome ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Welcome form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-----------------------------------------------------------------------------------------------
-------

    Private Sub mnuWelcome_Click(sender As Object, e As EventArgs) Handles mnuWelcome.Click
        Navigate(Forms.WELCOME, Me)
    End Sub

    '''-----------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Registration ToolStripMenuItem on the MenuStrip. </summary>
    '''
```

```
    ''' <remarks>   This button is used for navigating the user to the Registration form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuRegistration_Click(sender As Object, e As EventArgs) Handles
mnuRegistration.Click
        Navigate(Forms.REGISTRATION, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Login ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Login form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuLogin_Click(sender As Object, e As EventArgs) Handles mnuLogin.Click
        Navigate(Forms.LOGIN, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Catalog ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Catalog form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCatalog_Click(sender As Object, e As EventArgs) Handles mnuCatalog.Click
        Navigate(Forms.CATALOG, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Checkout ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Checkout form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
```

```
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCheckout_Click(sender As Object, e As EventArgs) Handles mnuCheckout.Click
        Navigate(Forms.CHECKOUT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Account ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Account form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAccount_Click(sender As Object, e As EventArgs) Handles mnuAccount.Click
        Navigate(Forms.ACCOUNT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Item ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Item form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuItem_Click(sender As Object, e As EventArgs) Handles mnuItem.Click
        Navigate(Forms.ITEM, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Payment ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Payment form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPayment_Click(sender As Object, e As EventArgs) Handles mnuPayment.Click
        Navigate(Forms.PAYMENT, Me)
    End Sub
```

```
    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Reload ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for reloading the currently displayed form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuReload_Click(sender As Object, e As EventArgs) Handles mnuReload.Click
        ReloadForm(Me)
        ResetForm()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SignOut ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for signing the user out of his/her account. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuSignOut_Click(sender As Object, e As EventArgs) Handles mnuSignOut.Click
        SignOut()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Exit ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for terminating the application. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuExit_Click(sender As Object, e As EventArgs) Handles mnuExit.Click
        ExitApplication()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The About ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for telling the user information about the currently
displayed form. </remarks>
    '''
    ''' <param name="sender">
```

```
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAbout_Click(sender As Object, e As EventArgs) Handles mnuAbout.Click
        AboutApplication(_cintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Print ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a print preview popup of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPrint_Click(sender As Object, e As EventArgs) Handles mnuPrint.Click
        PrintForm(pfPrintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The frmCatalog_Load subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for loading the defaults of this form when it is
loaded. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub frmCatalog_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadFormDefaults(Me)
        ResetForm()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The ResetForm subroutine. </summary>
    '''
    ''' <remarks>   This subroutine resets all of the data on the form. </remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub ResetForm()
        txtSearchQuery.Clear()
        rbCategoryAll.Select()
        GetResults()
    End Sub
```

```
    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The GetResults subroutine. </summary>
    '''
    ''' <remarks>   This subroutines retrieves and displays information about items in the
catalog. </remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub GetResults()
        lstProducts.Items.Clear()
        glstProductResults.Clear()
        glstProducts.ForEach(Sub(itmItem)
                                 If (GetSelectedCategory().Equals(ProductCategory.ALL) Or
                                  GetSelectedCategory().Equals(GetProductCategory(itmItem))) Then
                                     If (cbShowItemsOutOfStock.Checked Or Not
CheckOutOfStock(itmItem)) Then
                                         If (GetMatch(itmItem, txtSearchQuery.Text)) Then
                                             lstProducts.Items.Add(itmItem.Name & " - " &
itmItem.Price.ToString("C2"))
                                             glstProductResults.Add(itmItem)
                                         End If
                                     End If
                                 End If
                             End Sub)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The GetSelectedCategory subroutine. </summary>
    '''
    ''' <remarks>   This subroutine get the category of an item. </remarks>
    '''
    ''' <returns>   The category of an item as an integer. </returns>
    '''-------------------------------------------------------------------------------------------
-------

    Function GetSelectedCategory() As Integer
        Dim intCategory As Integer
        If (rbCategoryChair.Checked) Then
            intCategory = ProductCategory.CHAIR
        ElseIf (rbCategoryTable.Checked) Then
            intCategory = ProductCategory.TABLE
        ElseIf (rbCategoryDesk.Checked) Then
            intCategory = ProductCategory.DESK
        ElseIf (rbCategoryCouch.Checked) Then
            intCategory = ProductCategory.COUCH
        ElseIf (rbCategoryCarpet.Checked) Then
            intCategory = ProductCategory.CARPET
        Else
            intCategory = ProductCategory.ALL
        End If
        Return intCategory
    End Function

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Search button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine retrieves and displays the results of a search. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
```

```
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnSearch_Click(sender As Object, e As EventArgs) Handles btnSearch.Click
        GetResults()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The More details button. </summary>
    '''
    ''' <remarks>   This button navigates the user to a form that displays details about the
selected item in the catalog. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnMoreDetails_Click(sender As Object, e As EventArgs) Handles
btnMoreDetails.Click
        GetSelectedItem(lstProducts, glstProductResults)
        Navigate(Forms.ITEM, Me)
    End Sub
End Class
```

Checkout Form

```
'''-------------------------------------------------------------------------------------------
---
''' <summary>   The Checkout form. </summary>
'''
''' <remarks>   This form allows a user to purchase items in the shopping cart. </remarks>
'''-------------------------------------------------------------------------------------------
---

Public Class frmCheckout

    ''' <summary>   The constant integer for the current form. </summary>
    Const _cintForm As Integer = Forms.CHECKOUT
    ''' <summary>   The full price for everything in the shopping cart. </summary>
    Dim _FullPrice As Decimal = gcdecZero
    ''' <summary>   The number of items in the shopping cart. </summary>
    Dim _FullItemCount As Integer = gcintZero

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Welcome ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Welcome form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------
```

```
    Private Sub mnuWelcome_Click(sender As Object, e As EventArgs) Handles mnuWelcome.Click
        Navigate(Forms.WELCOME, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Registration ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Registration form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuRegistration_Click(sender As Object, e As EventArgs) Handles
mnuRegistration.Click
        Navigate(Forms.REGISTRATION, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Login ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Login form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuLogin_Click(sender As Object, e As EventArgs) Handles mnuLogin.Click
        Navigate(Forms.LOGIN, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Catalog ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Catalog form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCatalog_Click(sender As Object, e As EventArgs) Handles mnuCatalog.Click
        Navigate(Forms.CATALOG, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
```

```
    ''' <summary>   The Checkout ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Checkout form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCheckout_Click(sender As Object, e As EventArgs) Handles mnuCheckout.Click
        Navigate(Forms.CHECKOUT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Account ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Account form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAccount_Click(sender As Object, e As EventArgs) Handles mnuAccount.Click
        Navigate(Forms.ACCOUNT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Item ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Item form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuItem_Click(sender As Object, e As EventArgs) Handles mnuItem.Click
        Navigate(Forms.ITEM, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Payment ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Payment form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
```

```
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPayment_Click(sender As Object, e As EventArgs) Handles mnuPayment.Click
        Navigate(Forms.PAYMENT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Reload ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for reloading the current form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuReload_Click(sender As Object, e As EventArgs) Handles mnuReload.Click
        ReloadForm(Me)
        ResetForm()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SignOut ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for signing out of an account. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuSignOut_Click(sender As Object, e As EventArgs) Handles mnuSignOut.Click
        SignOut()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Exit ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for terminating the application. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuExit_Click(sender As Object, e As EventArgs) Handles mnuExit.Click
        ExitApplication()
    End Sub
```

```
    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The About ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a popup that tells the user some information
about the current form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAbout_Click(sender As Object, e As EventArgs) Handles mnuAbout.Click
        AboutApplication(_cintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Print ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a print preview of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPrint_Click(sender As Object, e As EventArgs) Handles mnuPrint.Click
        PrintForm(pfPrintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The frmCheckout_Load subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for loading the defaults of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub frmCheckout_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadFormDefaults(Me)
        ResetForm()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The ResetForm subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for resetting the data on the form. </remarks>
```

```
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub ResetForm()
        GetShoppingCart()
        Dim strFullPrice As String = "Price of all shopping cart items combined: {0}"
        lblFullPriceOutput.Text = String.Format(strFullPrice, _FullPrice.ToString("C2"))
        If (gusrCurrentUser.Money >= _FullPrice) Then
            btnPaymentOptions.Enabled = False
        Else
            btnPaymentOptions.Enabled = True
        End If
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The GetShoppingCart subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for updating the shopping cart. </remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub GetShoppingCart()
        lstShoppingCart.Items.Clear()
        glstShoppingCartResults.Clear()
        _FullPrice = gcintZero
        Dim itmItem As Item
        Dim strName As String
        Dim intQuantity As Integer
        Dim decPrice As Decimal
        Dim decFee As Decimal
        Dim decTotalPrice As Decimal
        Dim strListItem As String
        glstShoppingCart.ForEach(Sub(sciItem)
                                     If (sciItem.Quantity > gcintZero) Then
                                         GetProduct(sciItem.ID)
                                         itmItem = gitmCurrentItem
                                         strName = itmItem.Name
                                         intQuantity = sciItem.Quantity
                                         decPrice = itmItem.Price
                                         decFee = itmItem.Fee
                                         decTotalPrice = intQuantity * (decPrice + decFee)
                                         _FullPrice += decTotalPrice
                                         _FullItemCount += sciItem.Quantity
                                         strListItem = strName & " = " &
decTotalPrice.ToString("C2") & " (" & intQuantity.ToString() & " * (" & decPrice.ToString("C2") &
" + " & decFee.ToString("C2") & "))"
                                         lstShoppingCart.Items.Add(strListItem)
                                         glstShoppingCartResults.Add(itmItem)
                                     End If
                                 End Sub)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Remove One button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for removing one of the selected item from the
shopping cart. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
```

```
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnRemoveOne_Click(sender As Object, e As EventArgs) Handles btnRemoveOne.Click
        If (MessageBox.Show("Are you sure you want to remove one of the selected item?", "Remove
One?", MessageBoxButtons.YesNo).Equals(DialogResult.Yes) And
            lstShoppingCart.SelectedItems.Count > gcintZero) Then
            GetSelectedItem(lstShoppingCart, glstShoppingCartResults)
            GetProduct(gitmCurrentItem.ID)
            If (Not gitmCurrentItem.GetShoppingCartItem().Quantity.Equals(gcintZero)) Then
                gitmCurrentItem.GetShoppingCartItem().Quantity -= 1
                ResetForm()
            End If
        End If
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Remove All button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for removing all of the selected item from the
shopping cart. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnRemoveAll_Click(sender As Object, e As EventArgs) Handles btnRemoveAll.Click
        If (lstShoppingCart.SelectedItems.Count > gcintZero AndAlso
        MessageBox.Show("Are you sure you want to remove all of the selected item?", "Remove
All?", MessageBoxButtons.YesNo).Equals(DialogResult.Yes)) Then
            GetSelectedItem(lstShoppingCart, glstShoppingCartResults)
            GetProduct(gitmCurrentItem.ID)
            gitmCurrentItem.GetShoppingCartItem().Quantity = gcintZero
            ResetForm()
        End If
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Make Purchase button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for making a purchase. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnMakePurchase_Click(sender As Object, e As EventArgs) Handles
btnMakePurchase.Click
        If (gusrCurrentUser.SignedIn.Equals(False) Or gusrCurrentUser.ID.Equals(gcintZero)) Then
            MsgBox("Unable to make purchase." & vbCrLf & "You are not signed in.", , "Error")
        ElseIf (_FullItemCount.Equals(gcintZero)) Then
            MsgBox("You have zero items in your shopping cart.", , "Error")
        ElseIf (gusrCurrentUser.Money < _FullPrice) Then
```

```
            MsgBox("Unable to make purchase." & vbCrLf & "You have " &
gusrCurrentUser.Money.ToString("C2") & " and the total is " & _FullPrice.ToString("C2") & ".", ,
"Error")
            btnPaymentOptions.Enabled = True
        ElseIf (MessageBox.Show("Are you sure you want to make this purchase? (Total: " &
_FullPrice.ToString("C2") & ")" & vbCrLf & "Your current balance is " &
gusrCurrentUser.Money.ToString("C2") & "." & vbCrLf & "Your balance after the transaction will be
" & (gusrCurrentUser.Money - _FullPrice).ToString("C2") & ".", "Make Purchase?",
MessageBoxButtons.YesNo).Equals(DialogResult.Yes)) Then
            Try
                ' Making the payment
                SQLCreatePayment(gusrCurrentUser.ID, _FullItemCount, _FullPrice)
                SQLSetFieldInfo(DatabaseTables.ACCOUNT, gusrCurrentUser.ID, "ACC_MONEY",
(gusrCurrentUser.Money - _FullPrice).ToString())
                Dim itmItem As Item
                Dim strName As String
                Dim intQuantity As Integer
                Dim decPrice As Decimal
                Dim decFee As Decimal
                glstShoppingCart.ForEach(Sub(sciItem)
                                             If (sciItem.Quantity > gcintZero) Then
                                                 GetProduct(sciItem.ID)
                                                 itmItem = gitmCurrentItem
                                                 strName = itmItem.Name
                                                 intQuantity = sciItem.Quantity
                                                 decPrice = itmItem.Price
                                                 decFee = itmItem.Fee
                                                 SQLSetFieldInfo(DatabaseTables.PRODUCT,
itmItem.ID, "PROD_STOCK", (itmItem.Stock - sciItem.Quantity).ToString())
                                                 itmItem.GetShoppingCartItem().Quantity =
gcintZero
                                                 itmItem.Update()
                                             End If
                                         End Sub)
                gusrCurrentUser.SignIn(gusrCurrentUser.ID)
                ResetForm()
                MsgBox("Purchase successfully made." & vbCrLf & "Your balance is now " &
gusrCurrentUser.Money.ToString("C2") & ".", , "Success")
            Catch ex As Exception
                MsgBox("Failed to create payment.", , "Error")
                Console.WriteLine("Unknown error occurred while trying to make a payment.")
                Console.WriteLine(ex.Message)
            End Try
        End If
    End Sub

    '''-----------------------------------------------------------------------------------------
-------
    ''' <summary>   The Payment Options button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for navigating the user to the Payment form so that
he/she can add money to his/her account. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-----------------------------------------------------------------------------------------
-------

    Private Sub btnPaymentOptions_Click(sender As Object, e As EventArgs) Handles
btnPaymentOptions.Click
        Navigate(Forms.PAYMENT, Me)
    End Sub
```

```
End Class
```

Account Form

```
'''-----------------------------------------------------------------------------------------------
---
''' <summary>   The Account form. </summary>
'''
''' <remarks>   This form displays the user's account information and provides options for
changing them. </remarks>
'''-----------------------------------------------------------------------------------------------
---

Public Class frmAccount

    ''' <summary>   The constant integer for the current form. </summary>
    Const _cintForm As Integer = Forms.ACCOUNT

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Welcome ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Welcome form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuWelcome_Click(sender As Object, e As EventArgs) Handles mnuWelcome.Click
        Navigate(Forms.WELCOME, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Registration ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Registration form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuRegistration_Click(sender As Object, e As EventArgs) Handles
mnuRegistration.Click
        Navigate(Forms.REGISTRATION, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Login ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Login form. </remarks>
    '''
    ''' <param name="sender">
```

```
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuLogin_Click(sender As Object, e As EventArgs) Handles mnuLogin.Click
        Navigate(Forms.LOGIN, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Catalog ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Catalog form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCatalog_Click(sender As Object, e As EventArgs) Handles mnuCatalog.Click
        Navigate(Forms.CATALOG, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Checkout ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Checkout form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCheckout_Click(sender As Object, e As EventArgs) Handles mnuCheckout.Click
        Navigate(Forms.CHECKOUT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Account ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Account form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAccount_Click(sender As Object, e As EventArgs) Handles mnuAccount.Click
```

```
        Navigate(Forms.ACCOUNT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Item ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Item form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuItem_Click(sender As Object, e As EventArgs) Handles mnuItem.Click
        Navigate(Forms.ITEM, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Payment ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Payment form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPayment_Click(sender As Object, e As EventArgs) Handles mnuPayment.Click
        Navigate(Forms.PAYMENT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Reload ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for reloading the currently displayed form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuReload_Click(sender As Object, e As EventArgs) Handles mnuReload.Click
        ReloadForm(Me)
        ResetForm()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SignOut ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for signing the user out of his/her account. </remarks>
```

```
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuSignOut_Click(sender As Object, e As EventArgs) Handles mnuSignOut.Click
        SignOut()
        ResetForm()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Exit ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for terminating the application. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuExit_Click(sender As Object, e As EventArgs) Handles mnuExit.Click
        ExitApplication()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The About ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for telling the user information about the currently
displayed form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAbout_Click(sender As Object, e As EventArgs) Handles mnuAbout.Click
        AboutApplication(_cintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Print ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a print preview popup of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
```

```
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPrint_Click(sender As Object, e As EventArgs) Handles mnuPrint.Click
        PrintForm(pfPrintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The frmAccount_Load subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for loading the defaults of this form when it is
loaded. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub frmAccount_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadFormDefaults(Me)
        ResetForm()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The ResetForm subroutine. </summary>
    '''
    ''' <remarks>   This subroutine resets all of the data on the form. </remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub ResetForm()
        lblAccountIDOutput.Text = gusrCurrentUser.ID.ToString()
        txtUsername.Text = gusrCurrentUser.Username
        txtFirstName.Text = gusrCurrentUser.FirstName
        txtLastName.Text = gusrCurrentUser.LastName
        txtEmail.Text = gusrCurrentUser.Email
        txtPhoneNumber.Text = gusrCurrentUser.Phone
        txtAddress.Text = gusrCurrentUser.Address
        lblMoneyOutput.Text = gusrCurrentUser.Money.ToString("C2")
        lblCreationDateOutput.Text = gusrCurrentUser.CreationDate
        lblAccountStatusOutput.Text = gusrCurrentUser.Status
        txtUsername.Enabled = False
        txtFirstName.Enabled = False
        txtLastName.Enabled = False
        txtEmail.Enabled = False
        txtPhoneNumber.Enabled = False
        txtAddress.Enabled = False
        btnChangeInfo.Enabled = True
        btnSave.Enabled = False
        btnReset.Enabled = False
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Change Info button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine allows the user to change his/her account information.
</remarks>
    '''
```

```
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnChangeInfo_Click(sender As Object, e As EventArgs) Handles btnChangeInfo.Click
        If (gusrCurrentUser.ID = gcintZero Or gusrCurrentUser.SignedIn = False) Then
            MsgBox("You are not signed in.", , "Error")
        ElseIf (gusrCurrentUser.Username = "Guest") Then
            MsgBox("You cannot change the info for Guest.", , "Error")
        Else
            txtUsername.Enabled = True
            txtFirstName.Enabled = True
            txtLastName.Enabled = True
            txtEmail.Enabled = True
            txtPhoneNumber.Enabled = True
            txtAddress.Enabled = True
            btnChangeInfo.Enabled = False
            btnSave.Enabled = True
            btnReset.Enabled = True
        End If
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Save button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine commits new data to the database for the user's account.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnSave_Click(sender As Object, e As EventArgs) Handles btnSave.Click
        If (gusrCurrentUser.ID = gcintZero Or gusrCurrentUser.SignedIn = False) Then
            MsgBox("You are not signed in.", , "Error")
        Else
            Const cstrSuccessTitle As String = "Success"
            Const cstrSuccessMessage As String = "Account info changed."
            Const cstrErrorTitle As String = "Error"
            Const cstrBaseErrorMessage As String = "Validation for the following fields failed:"
            Const cstrUnknownErrorMessage As String = "An unknown error has occurred when trying
to change account info."
            Dim strErrorMessage As String = cstrBaseErrorMessage

            Dim strUsername As String = txtUsername.Text
            Dim strFName As String = txtFirstName.Text
            Dim strLName As String = txtLastName.Text
            Dim strEmail As String = txtEmail.Text
            Dim strPhone As String = txtPhoneNumber.Text
            Dim strAddress As String = txtAddress.Text

            Dim blnUsernameValidated As Boolean = RegexValidateUserData(strUsername,
RegexValidate.USERNAME)
            Dim blnFNameValidated As Boolean = RegexValidateUserData(strFName,
RegexValidate.OTHER_EMPTY)
```

```
            Dim blnLNameValidated As Boolean = RegexValidateUserData(strLName,
RegexValidate.OTHER_EMPTY)
            Dim blnEmailValidated As Boolean = RegexValidateUserData(strEmail,
RegexValidate.EMAIL)
            Dim blnPhoneValidated As Boolean = RegexValidateUserData(strPhone,
RegexValidate.OTHER_EMPTY)
            Dim blnAddressValidated As Boolean = RegexValidateUserData(strAddress,
RegexValidate.OTHER_EMPTY)

            If (Not blnUsernameValidated) Then
                strErrorMessage = strErrorMessage & vbCrLf & "Username must be 4-16 characters,
start with a letter, and only contain A-Z a-z 0-9"
            End If
            If (Not blnFNameValidated) Then
                strErrorMessage = strErrorMessage & vbCrLf & "First Name is optional. Can only
contain spaces and A-Z a-z 0-9 @ . - _ ( )"
            End If
            If (Not blnLNameValidated) Then
                strErrorMessage = strErrorMessage & vbCrLf & "Last Name is optional. Can only
contain spaces and A-Z a-z 0-9 @ . - _ ( )"
            End If
            If (Not blnEmailValidated) Then
                strErrorMessage = strErrorMessage & vbCrLf & "Email must be 5-50 characters and
contain @ and . characters"
            End If
            If (Not blnPhoneValidated) Then
                strErrorMessage = strErrorMessage & vbCrLf & "Phone Number is optional. Can only
contain spaces and A-Z a-z 0-9 @ . - _ ( )"
            End If
            If (Not blnAddressValidated) Then
                strErrorMessage = strErrorMessage & vbCrLf & "Address is optional. Can only
contain spaces and A-Z a-z 0-9 @ . - _ ( )"
            End If

            Const cstrEmailField As String = "ACC_EMAIL"
            Const cstrUsernameField As String = "ACC_USERNAME"
            Const cintMissingRecordID As Integer = -1
            Dim intExistingUsernameRecordID As Integer = SQLGetRecordID(DatabaseTables.ACCOUNT,
cstrUsernameField, strUsername)
            Dim intExistingEmailRecordID As Integer = SQLGetRecordID(DatabaseTables.ACCOUNT,
cstrEmailField, strEmail)

            If (Not intExistingUsernameRecordID.Equals(cintMissingRecordID) And Not
intExistingUsernameRecordID.Equals(gusrCurrentUser.ID)) Then
                strErrorMessage = strErrorMessage & vbCrLf & "Username already exists in the
database"
            End If
            If (Not intExistingEmailRecordID.Equals(cintMissingRecordID) And Not
intExistingEmailRecordID.Equals(gusrCurrentUser.ID)) Then
                strErrorMessage = strErrorMessage & vbCrLf & "Email already exists in the
database"
            End If

            If (strErrorMessage.Equals(cstrBaseErrorMessage)) Then

                Try
                    Dim strField As String
                    If (Not gusrCurrentUser.Username.Equals(strUsername)) Then
                        strField = "ACC_USERNAME"
                        SQLSetFieldInfo(DatabaseTables.ACCOUNT, gusrCurrentUser.ID, strField,
strUsername)
                    End If
                    If (Not gusrCurrentUser.FirstName.Equals(strFName)) Then
                        strField = "ACC_FNAME"
                        SQLSetFieldInfo(DatabaseTables.ACCOUNT, gusrCurrentUser.ID, strField,
strFName)
```

```
                    End If
                    If (Not gusrCurrentUser.FirstName.Equals(strLName)) Then
                        strField = "ACC_LNAME"
                        SQLSetFieldInfo(DatabaseTables.ACCOUNT, gusrCurrentUser.ID, strField,
strLName)
                    End If
                    If (Not gusrCurrentUser.FirstName.Equals(strEmail)) Then
                        strField = "ACC_EMAIL"
                        SQLSetFieldInfo(DatabaseTables.ACCOUNT, gusrCurrentUser.ID, strField,
strEmail)
                    End If
                    If (Not gusrCurrentUser.FirstName.Equals(strPhone)) Then
                        strField = "ACC_PHONE"
                        SQLSetFieldInfo(DatabaseTables.ACCOUNT, gusrCurrentUser.ID, strField,
strPhone)
                    End If
                    If (Not gusrCurrentUser.FirstName.Equals(strAddress)) Then
                        strField = "ACC_ADDRESS"
                        SQLSetFieldInfo(DatabaseTables.ACCOUNT, gusrCurrentUser.ID, strField,
strAddress)
                    End If

                    gusrCurrentUser.SignIn(gusrCurrentUser.ID)
                    ResetForm()
                    MsgBox(cstrSuccessMessage, , cstrSuccessTitle)
                Catch ex As Exception
                    Console.WriteLine(ex.Message)
                    MsgBox(cstrUnknownErrorMessage, , cstrErrorTitle)
                End Try

            Else
                strErrorMessage = strErrorMessage & vbCrLf & vbCrLf & "No changes were made."
                MsgBox(strErrorMessage, , cstrErrorTitle)
            End If
        End If
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Reset button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine resets all of the data on the form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnReset_Click(sender As Object, e As EventArgs) Handles btnReset.Click
        ResetForm()
    End Sub
End Class
```

Item Form

```
'''-------------------------------------------------------------------------------------------
---
''' <summary>   The Item form. </summary>
'''
''' <remarks>   This form displays information about a specific product from the catalog and
gives the option for adding it to the shopping cart. </remarks>
'''-------------------------------------------------------------------------------------------
---

Public Class frmItem

    ''' <summary>   The constant integer for the current form. </summary>
    Const _cintForm As Integer = Forms.ITEM

    '''-------------------------------------------------------------------------------------
-------
    ''' <summary>   The Welcome ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Welcome form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------
-------

    Private Sub mnuWelcome_Click(sender As Object, e As EventArgs) Handles mnuWelcome.Click
        Navigate(Forms.WELCOME, Me)
    End Sub

    '''-------------------------------------------------------------------------------------
-------
    ''' <summary>   The Registration ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Registration form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------
-------

    Private Sub mnuRegistration_Click(sender As Object, e As EventArgs) Handles
mnuRegistration.Click
        Navigate(Forms.REGISTRATION, Me)
    End Sub

    '''-------------------------------------------------------------------------------------
-------
    ''' <summary>   The Login ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Login form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
```

```
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuLogin_Click(sender As Object, e As EventArgs) Handles mnuLogin.Click
        Navigate(Forms.LOGIN, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Catalog ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Catalog form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCatalog_Click(sender As Object, e As EventArgs) Handles mnuCatalog.Click
        Navigate(Forms.CATALOG, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Checkout ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Checkout form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCheckout_Click(sender As Object, e As EventArgs) Handles mnuCheckout.Click
        Navigate(Forms.CHECKOUT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Account ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Account form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAccount_Click(sender As Object, e As EventArgs) Handles mnuAccount.Click
        Navigate(Forms.ACCOUNT, Me)
    End Sub
```

```
    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Item ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Item form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuItem_Click(sender As Object, e As EventArgs) Handles mnuItem.Click
        Navigate(Forms.ITEM, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Payment ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Payment form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPayment_Click(sender As Object, e As EventArgs) Handles mnuPayment.Click
        Navigate(Forms.PAYMENT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Reload ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for reloading the currently displayed form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuReload_Click(sender As Object, e As EventArgs) Handles mnuReload.Click
        ReloadForm(Me)
        ResetForm()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SignOut ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for signing the user out of his/her account. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
```

```
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuSignOut_Click(sender As Object, e As EventArgs) Handles mnuSignOut.Click
        SignOut()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Exit ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for terminating the application. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuExit_Click(sender As Object, e As EventArgs) Handles mnuExit.Click
        ExitApplication()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The About ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for telling the user information about the currently
displayed form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAbout_Click(sender As Object, e As EventArgs) Handles mnuAbout.Click
        AboutApplication(_cintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Print ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a print preview popup of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------
```

```
    Private Sub mnuPrint_Click(sender As Object, e As EventArgs) Handles mnuPrint.Click
        PrintForm(pfPrintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The frmItem_Load subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for loading the defaults of this form when it is
loaded. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub frmItem_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadFormDefaults(Me)
        ResetForm()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The ResetForm subroutine. </summary>
    '''
    ''' <remarks>   This subroutine resets all of the data on the form. </remarks>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub ResetForm()
        updAmountInCart.Maximum = gitmCurrentItem.Stock
        updAmountInCart.Value = gitmCurrentItem.GetShoppingCartItem().Quantity
        lblProductIDOutput.Text = gitmCurrentItem.ID.ToString()
        lblNameOutput.Text = gitmCurrentItem.Name
        lblPriceOutput.Text = gitmCurrentItem.Price.ToString("C2")
        lblStockOutput.Text = gitmCurrentItem.Stock.ToString()
        lblFeeOutput.Text = gitmCurrentItem.Fee.ToString("C2")
        lblCategoryOutput.Text = gitmCurrentItem.Category
        lblDescriptionOutput.Text = gitmCurrentItem.Description
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Refresh button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine resets all of the data on the form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnRefresh_Click(sender As Object, e As EventArgs) Handles btnRefresh.Click
        ResetForm()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
```

```
    ''' <summary>   The AmountIntCart NumericUpDown subroutine. </summary>
    '''
    ''' <remarks>   This subroutine changes the quantity of an item in the shopping cart when the
value of the NumericUpDown object changes. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub updAmountInCart_ValueChanged(sender As Object, e As EventArgs) Handles
updAmountInCart.ValueChanged
        gitmCurrentItem.GetShoppingCartItem().Quantity = updAmountInCart.Value
    End Sub
End Class
```

Payment Form

```
'''-------------------------------------------------------------------------------------------
---
''' <summary>   The Payment form. </summary>
'''
''' <remarks>   This form is used for exchanging real money for account money. </remarks>
'''-------------------------------------------------------------------------------------------
---

Public Class frmPayment

    ''' <summary>   The constant integer for the current form. </summary>
    Const _cintForm As Integer = Forms.PAYMENT

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Welcome ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Welcome form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuWelcome_Click(sender As Object, e As EventArgs) Handles mnuWelcome.Click
        Navigate(Forms.WELCOME, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Registration ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Registration form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
```

```
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuRegistration_Click(sender As Object, e As EventArgs) Handles
mnuRegistration.Click
        Navigate(Forms.REGISTRATION, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Login ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Login form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuLogin_Click(sender As Object, e As EventArgs) Handles mnuLogin.Click
        Navigate(Forms.LOGIN, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Catalog ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Catalog form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCatalog_Click(sender As Object, e As EventArgs) Handles mnuCatalog.Click
        Navigate(Forms.CATALOG, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Checkout ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Checkout form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuCheckout_Click(sender As Object, e As EventArgs) Handles mnuCheckout.Click
        Navigate(Forms.CHECKOUT, Me)
    End Sub
```

```
    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Account ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Account form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAccount_Click(sender As Object, e As EventArgs) Handles mnuAccount.Click
        Navigate(Forms.ACCOUNT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Item ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Item form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuItem_Click(sender As Object, e As EventArgs) Handles mnuItem.Click
        Navigate(Forms.ITEM, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Payment ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for navigating the user to the Payment form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPayment_Click(sender As Object, e As EventArgs) Handles mnuPayment.Click
        Navigate(Forms.PAYMENT, Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Reload ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for reloading the current form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
```

```
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuReload_Click(sender As Object, e As EventArgs) Handles mnuReload.Click
        ReloadForm(Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The SignOut ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for signing out of an account. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuSignOut_Click(sender As Object, e As EventArgs) Handles mnuSignOut.Click
        SignOut()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Exit ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for terminating the application. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuExit_Click(sender As Object, e As EventArgs) Handles mnuExit.Click
        ExitApplication()
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The About ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a popup that tells the user some information
about the current form. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuAbout_Click(sender As Object, e As EventArgs) Handles mnuAbout.Click
```

```
        AboutApplication(_cintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Print ToolStripMenuItem on the MenuStrip. </summary>
    '''
    ''' <remarks>   This button is used for creating a print preview of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub mnuPrint_Click(sender As Object, e As EventArgs) Handles mnuPrint.Click
        PrintForm(pfPrintForm)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The frmPayment_Load subroutine. </summary>
    '''
    ''' <remarks>   This subroutine is used for loading the defaults of the current form.
</remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub frmPayment_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadFormDefaults(Me)
    End Sub

    '''-------------------------------------------------------------------------------------------
-------
    ''' <summary>   The Add Money button subroutine. </summary>
    '''
    ''' <remarks>   This subroutine would be used for exchanging real money for account money,
but that feature is not implemented. </remarks>
    '''
    ''' <param name="sender">
    ''' Source of the event.
    ''' </param>
    ''' <param name="e">
    ''' Event information.
    ''' </param>
    '''-------------------------------------------------------------------------------------------
-------

    Private Sub btnAddMoney_Click(sender As Object, e As EventArgs) Handles btnAddMoney.Click
        MsgBox("Feature not yet implemented yet." & vbCrLf & "Contact a database administrator
for more information.", , "Error")
    End Sub
End Class
```

Appendix

GitHub links for the project:

- Repository - https://github.com/ShawnBroyles/Furniture-Co-Catalog
- Wiki - https://github.com/ShawnBroyles/Furniture-Co-Catalog/wiki

Tools used for developing the application:

- Visual Studio Community 2017 version 15.3.26730.8 – Used as an Integrated Development Environment.
- Microsoft .NET Framework 4.7.02053 – Used for creating the Visual Basic .NET Windows Form Application within Visual Studio.
- Visual Basic Power Packs Controls 12.0.2.21005.1 – Used for the PrintForm control.
- System.Data.SQLite.Core NuGet Package version 1.0.108 – Used as a database management system.
- GitHub Extension for Visual Studio version 2.4.3.1737 – Used for making commits for the GitHub repository.
- Atomineer Pro Documentation Trial 9.42.3.2590 – Used for creating the XML documentation tags <summary>, <remarks>, <value>, <param>, and <returns> at the appropriate locations.

Links for examples of code viewed while developing the application:

- Removing the close button from the Welcome form - https://stackoverflow.com/questions/1743433/visually-remove-disable-close-button-from-title-bar-net
- Reading/writing to/from a SQLite database - https://stackoverflow.com/questions/19553165/vb-net-open-local-sqlite-table
- Example of a module - https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/concepts/linq/how-to-create-a-list-of-items
- Regular expression validation - https://stackoverflow.com/questions/27126257/vb-net-validation-checking-if-text-contains-only-letters
- Popup with Yes and No options - https://stackoverflow.com/questions/2256909/messagebox-with-yesnocancel-no-cancel-triggers-same-event
- Example of creating and using an event - https://docs.microsoft.com/en-us/dotnet/visual-basic/language-reference/statements/raiseevent-statement
- Adding a handler - https://stackoverflow.com/questions/8565671/vb-net-add-handler-to-object-of-unknown-type
- Adding items to an Array - https://stackoverflow.com/questions/18097756/fastest-way-to-add-an-item-to-an-array
- Iterating through a list - https://stackoverflow.com/questions/8897906/list-foreach-in-vb-net-perplexing-me
- Creating a date in SQLite - https://stackoverflow.com/questions/381371/sqlite-current-timestamp-is-in-gmt-not-the-timezone-of-the-machine