LZL Accelerated Equity Programs

By

Shawn O'Rourke

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

June 2004

LZL Accelerated Equity Programs

By

Shawn O'Rourke

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

Shawn O'Rourke	Date
Professor Annu Prabhakar, Faculty Advisor	Date
Professor James Sullivian, Department Head	Date

Acknowledgements

I would like to give special thanks to the University of Cincinnati, College of Applied Science Information Engineering Technology faculty and staff, especially Annu Prabhakar. Their patience and experience is a valuable asset to the University of Cincinnati. I would like to thank Larry Jedding, President of LZL Mortgage Programs. His patience and answers to specific questions were invaluable. Finally I would like to acknowledge the support and patience of my loving family during the time that their father or husband seemed to be missing.

Table of Contents

List of Figures

Abstract

The LZL Accelerated Mortgage Company has been offering an accelerated mortgage program for several years through their Web site I recently updated the LZL Web site using current technology. The LZL Accelerated Mortgage Web site is user friendly and automated as much as possible. The site has an explanation of the program, examples of the savings and a mortgage calculator for the potential customer. The site also allows the customers to view their account information. The site will enable the LZL staff to maintain all account information. The LZL staff has the ability to edit the customer information, process customer withdraw and deposit transactions. The Web site and database were designed using Microsoft SQL 2000 and Visual Studio .NET.

The LZL Accelerated Equity Program Web Site

1. Statement of Problem

LZL is an accelerated mortgage program company. The current business model consists of a Web site and several employees. LZL is an eBusiness. The current Web site is a block text style site that is not user friendly. The Web site allows for some navigation. and was created in Microsoft Word and is static. The only way to enroll in the program is to mail in an enrollment form. The colors are dull. Currently, each enrollment form is manually entered into a database and the account is maintained from there. The database was created in Microsoft Access. All transactions and entries are manually completed. The database is manually scanned weekly for transactions that require completion. The transactions are sent to Fifth Third bank through a secure Web site. The program is called Fifth Third Direct, (https://direct.53.com/logon53Direct.html). Fifth Third completes the transaction for a small fee. At the end of the month the database is manually scanned for all due payments on client accounts. A check is then printed and mailed. This will not serve the future of LZL. It is costly and relies heavily upon the input data and manual scanning to be correct. Entering the customer's data manually is time consuming. The process requires as much automation as possible.

2. Description of Solution

The LZL Accelerated Equity Program Web site is used for sales and administration. Using HTML and ASP.NET Web interface with a relational Microsoft SQL 2000 database, the Web site provides the LZL staff and customers the ability to monitor the accounts 24-hours a day with an Internet connection.

I have chosen to use ASP.NET for the following reasons:

- ASP.NET has the ability to run code on the server.
- The web browser only sees HTML code. This allows most of the current browsers to access the web pages.
- An efficient database link to many different databases that comply with the Open Database Connection (ODBC) standards.

2.1 User Profiles

There are three intended users for the LZL Accelerated Equity Program Web site. The users are the potential customer, the customer and the LZL staff.

- The potential customer will need to browse information about the program. These pages will contain detailed information about the LZL Accelerated Equity Program. They have moderate knowledge of the Internet. They are able to see what experts say about accelerated mortgage programs, view an example, find out how the program works, contact the LZL staff and have the ability to enroll.
- The customer has access to all the information that can be accessed by the potential customer. The notable exception is the customer has the ability to login to the database and view their account information and history.

2.2 Project Design

The project design has incorporated two areas of the Information Engineering Technology program to complete the project. The primary focuses are database management, programming and security for the data. Visual Basic.NET is the language used to communicate with the database. ASP.NET is used for this interface; they represent the best obtainable options. The database has been created using Microsoft SQL 2000. The database contains several different relationships and several different stored procedures.

3. Web Design

This site has been designed to increase sales and to automate as much of the process as possible. I have created a seamless Web site with a new design. Some of the key pages include:

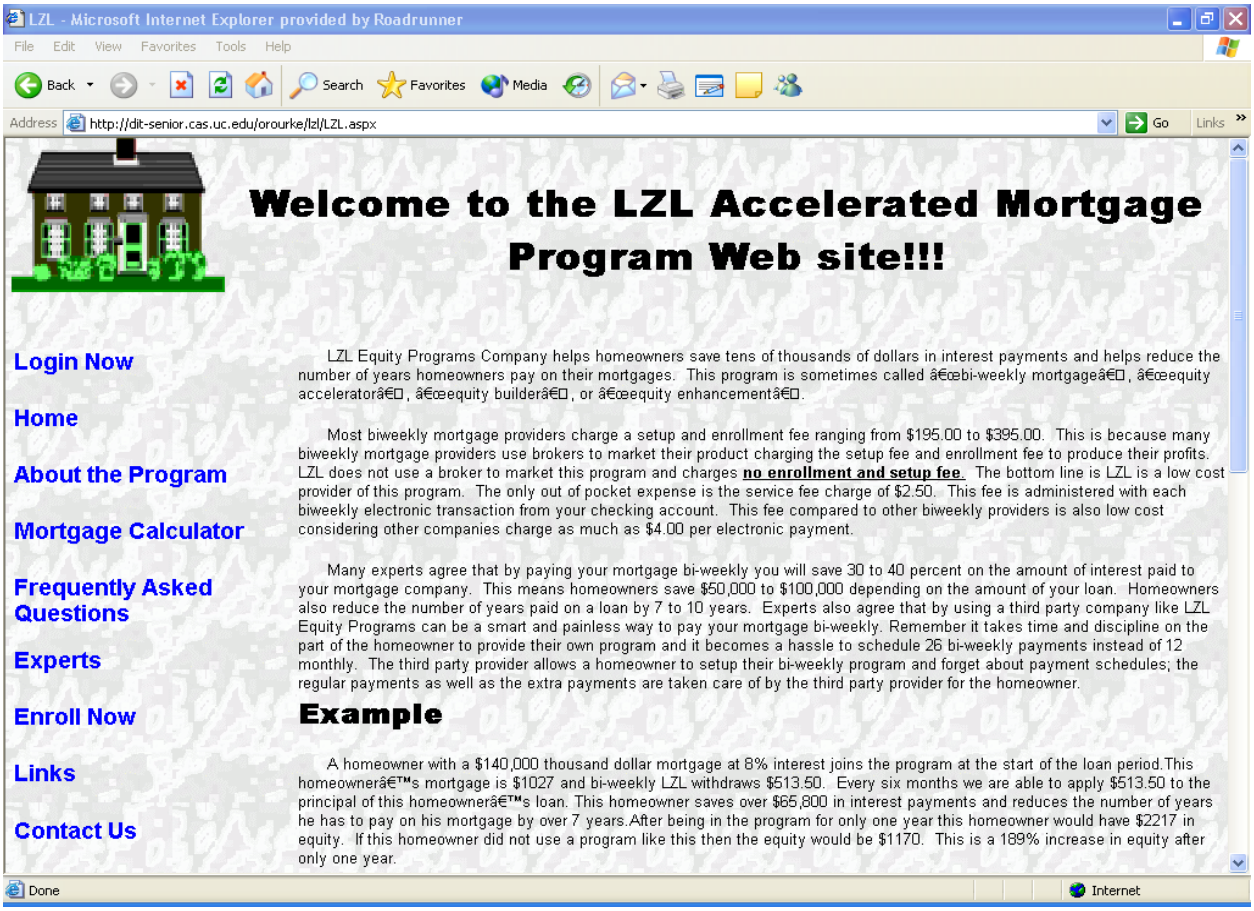

Figure 1. LZL Home Page

- The “Home” screen contains a link to all pages and an example of the savings along with a brief description of the program. I have provided a screen shot of the “Home” page above (Figure 1).

- The second screen contains all the pertinent details about the program. The screen is called “About the Program”. It contains links to all the other pages.

- The next screen contains links to all the other pages and is called “Experts”. This page has information about what some of the experts say about the accelerated mortgage programs.

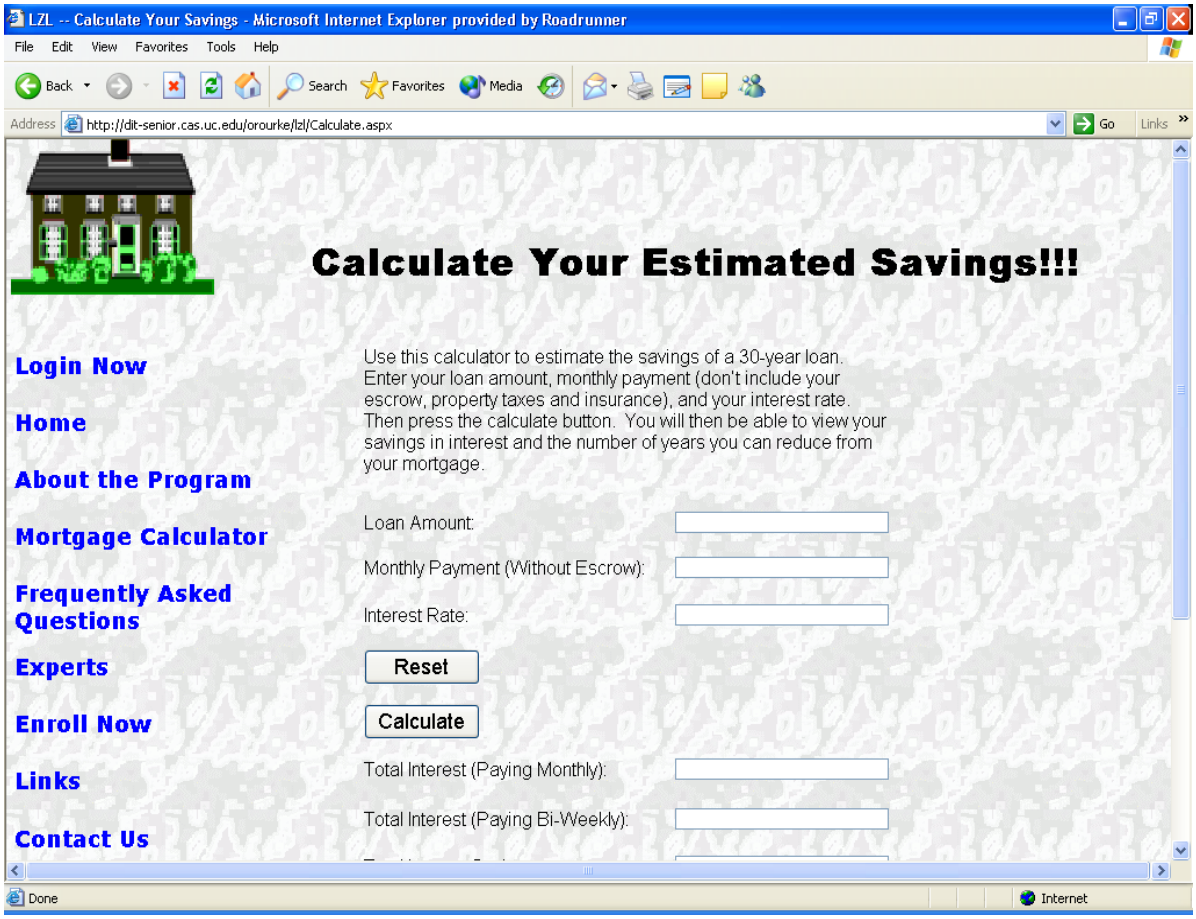

Figure 2. LZL Calculation Page

- The ‘Calculate’ page contains the potential savings based on the customer’s provided information for a 30-year loan. I have provided a screen shot above (Figure 2.) The provided information from the customer is:
    - Percentage Rate
    - Mortgage Balance
    - Monthly Payment

  The results include:

    - Total interest for Monthly and Bi-Weekly payments

- Total Savings
- Number of Payments
- Number of Years Reduced

- The site has a “Login” link to a login page. This page allows the customer to access his or her own data securely. It also permits the LZL staff to login and modify the database for their required needs. It also allows new users to be created. I have provided a screen shot below (Figure 3).

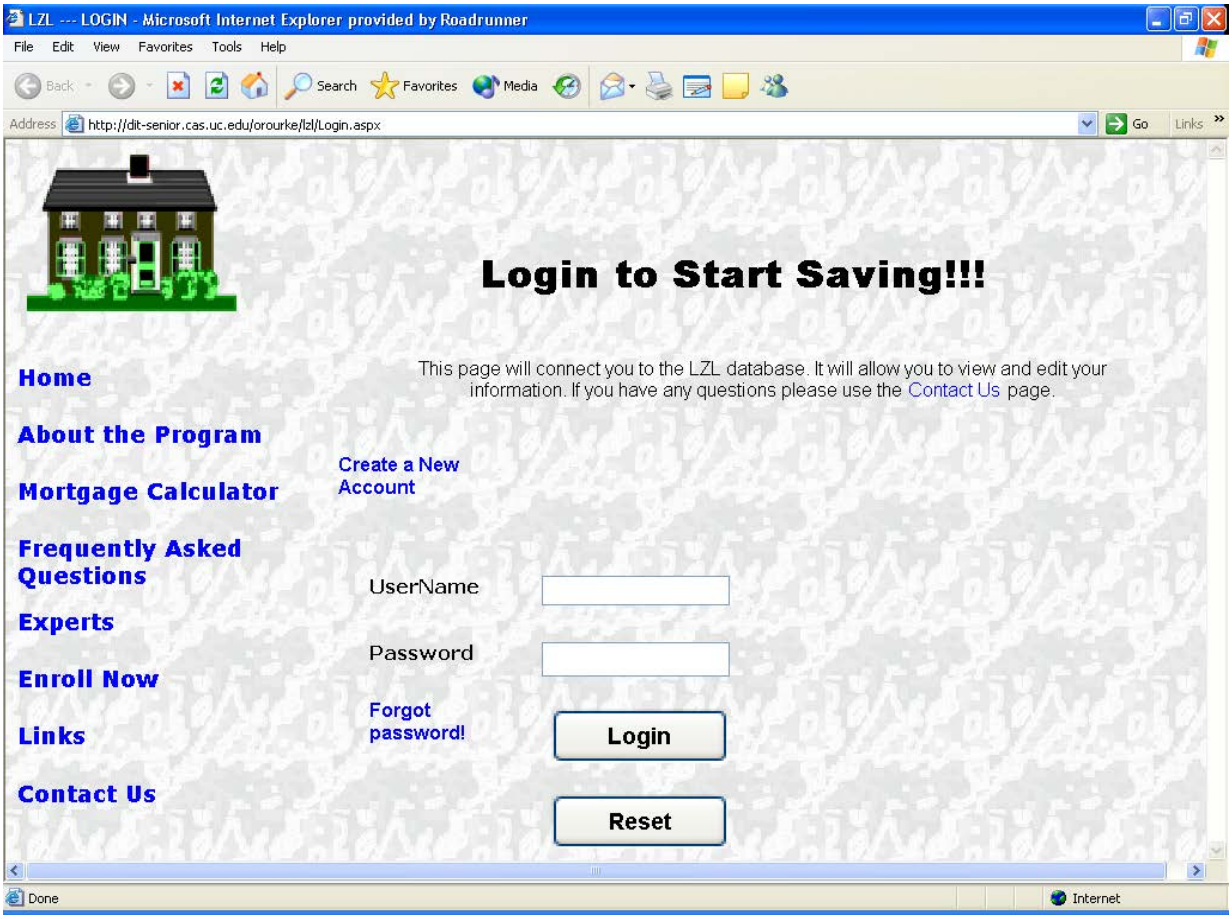

Figure 3. LZL -- Login Page

- The “Frequently Asked Questions” page includes links to all the other pages. The page lists commonly asked questions about the program and answers for the potential customer.

- The “Links” page includes links to several other sites that may be of use for the customer. This page also has links to all the other pages

.

- The “Contact Us” page contains information for the customer and potential customer to contact the company. This page also has links to all the other pages.

- The ‘Enroll Now’ page provides information on how to enroll in the program and the required information for the customer to provide. This page provides several options to enroll, such as, mailing an enrollment form, by phone, or by the Web site. The mailing form is in Text format for the customer to mail to LZL. This page contains links to all the other pages.

4. Web Security

Due to the sensitive nature of the information being transmitted, Verisign™ will secure the information. Verisign supports the familiar padlock icon and the “HTTPS:\\” prefix that indicates a secured Web Site. The Web site displays the secure site seal. Verisign™ Secure Socket Layer (SSL) allows the customer to verify the Web site authenticity. Verisign™ verifies that the company has a license to do business and must pass the verification process. With SSL installed on the server, the Web site can be protected by 128-bit SSL encryption system. 128-bit SSL encryption security is the maximum available. By displaying the Verisign™ identification icon, the customer has the reassurances of using a secure Web site. I have also added a Turing test to verify a script is not behind the login entries.

5. Database Design

Figure 4. shows a relationship diagram with all the tables below.

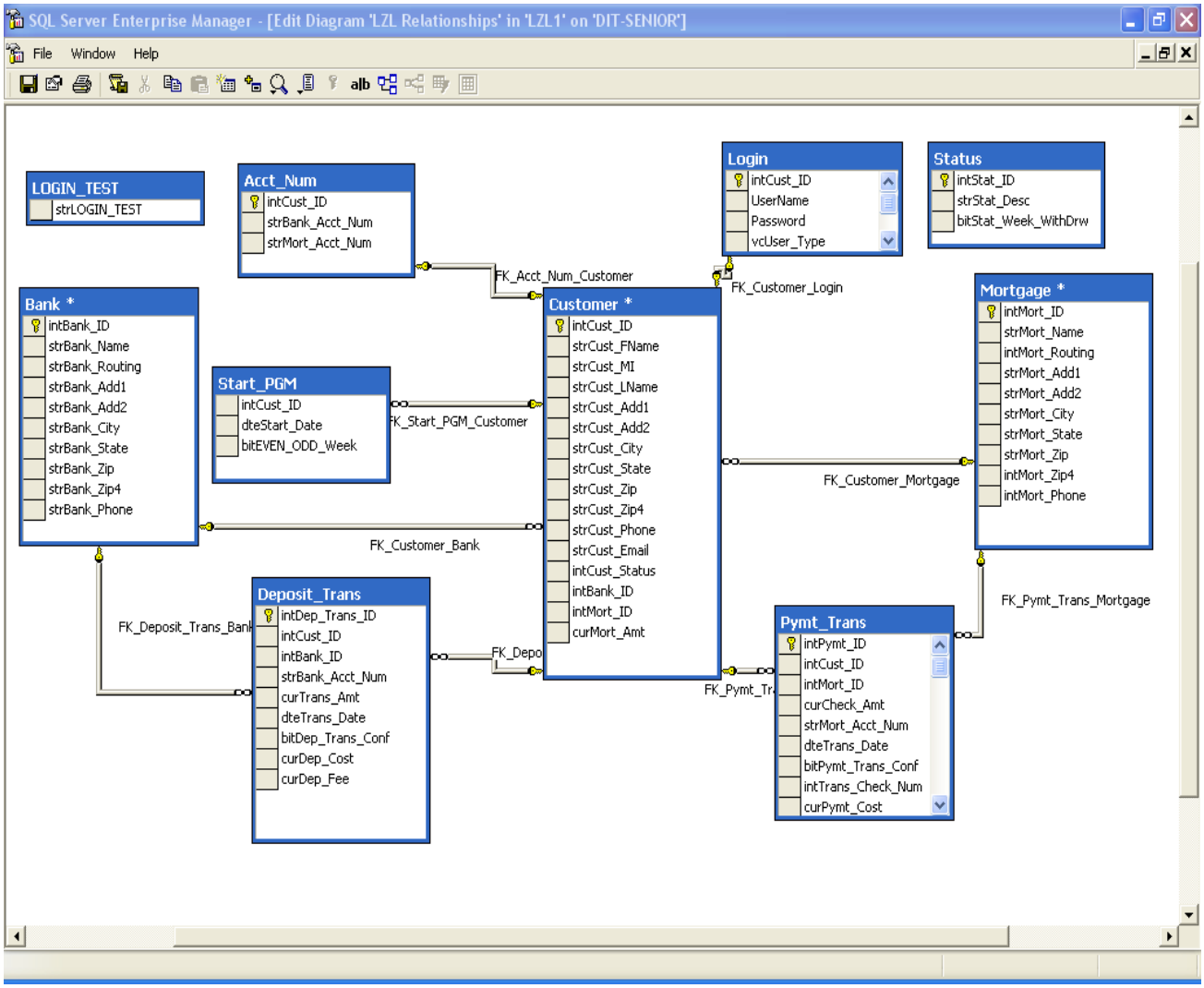

Figure 4. SQL Database Relationships

6. Deliverables

- A Web based accelerated equity program that manages customer accounts and transactions.

- The user interface is created using ASP. ASP is used to link to the database.

- The database is used to store customer information and stores procedures that allow the LZL staff to manage the accounts and the customer to view their own account information.

- Users of the accelerated equity program have a secure login authenticated by the database.

- Customers of the LZL Accelerated Equity Program have the ability to:

    - Create a new account with a login

    - View their account information

- The LZL staff have the ability to use the LZL Accelerated Equity Program to:

    - Edit all customer information

    - Process withdraw transactions

    - Process payment transactions

7. Testing

The primary testing was conducted on several different computers. The testing is ongoing and continuous. I have conducted testing in the 4th floor computer lab at the College of Applied Science, at Siemens MTBU, my employer, and at home. I have included testing by several available people. These people are Larry Jedding (LZL President), my spouse and daughter, and several coworkers, (Lonnie Carter, Chris Schanding and Terry Stang).

8. Conclusions and Recommendations

This project was created in response to LZL's need of a more efficient business process. The Web site is designed for existing and potential customers. To design the project I used Microsoft SQL 2000, Microsoft Visual Studio .NET and Microsoft Office XP. I created a visually soothing and functional site with the use of Verisign™ (SSL) and a Turing test designed for security. There is an abundance of information for the potential user and a functional calculator that can estimate the customer's savings. The budget of $10,875 is a real world estimate for the completion of the project without labor costs. Testing was performed to verify consumer usability and functionality.

While working on this project, I have encountered several hurdles that needed to be addressed. A solid understanding of .NET was required. As the project processed, the project became easier to complete. I began with the static pages and worked my way into the dynamic pages as the project progressed. The database was 90% complete early in the project. I made modifications to the database as required. I have scheduled several jobs in the database to be completed on a regular basis. Several of these jobs output files for the LZL staff to transmit to 5/3 Direct. The challenging portions of the programming were the validation of data and login code. Beginning users of Visual Studio may want to take this into account. Since I am a night student, and I have taken the programming classes years ago with what now is obsolete software. I would recommend that senior project and the design begin much earlier in the curriculum. I found that too much time has passed for many of the classes to be of much value.

Appendix A.

Timeline

Figure 5. LZL Development Timeline

Appendix B.

Code Snippets

In order to login to the database and view data the customer is required to enter a username and password. These items are entered into a textbox and password box. These are then compared to values in the ‘Login’ values in the dataset. I then select which user is valid and create session variables that are carried into the ‘Customer Data’ page. If the username or password are invalid, lbl1 is displayed notifying the user that they have entered invalid login information.

```
        Dim dtLogin As DataTable
        Dim drUname() As DataRow
        Dim strUtype As String
        Dim dtLOGIN_TEST As DataTable
        Dim drLOGIN_TEST As DataRow
        Dim key As String

        'get username and password for session
        Session.Contents("username") = txtUName.Text
        Session.Contents("password") = pwPWord.Value

        'Validate(username And password)
        SqlDataAdapter1.SelectCommand.Parameters("@param3").Value =
            txtUName.Text
        SqlDataAdapter1.SelectCommand.Parameters("@param4").Value =
            pwPWord.Value
        DataSetLogin1.Clear()
        SqlDataAdapter1.Fill(DataSetLogin1)
        If Not IsPostBack Then
            DataGrid1.DataBind()
        End If

        'tie username to table row and get customer_id and user type
for session
        dtlogin = Me.DataSetLogin1.Tables("login")
        drUname = dtLogin.Select("username = '" +
        Session.Contents("username") + "'")

        'catch bad login username or password
        Try
            Session.Contents("customer_id") =
                  drUname(0).Item("intcust_id")
```

```
        Session.Contents("usertype") =
              drUname(0).Item("vcUser_Type")

    Catch
        If i < 4 Then
            lbl1.Visible = True
            i += 1
            Exit Sub
        Else
            LOGIN_TEST(i)
        End If
    End Try

    'Invalid username or password display lbl1
    If Session.Contents("customer_id") = Nothing Then
        lbl1.Visible = True
        Exit Sub
    Else
        Server.Transfer("Customer_Data.aspx")
    End If
```

The code below is from the 'Customer Data' page load procedure. First, I have set some of the properties for the datagrid. I then have selected the user type from the session variable and display the appropriate items. I then have the customer data displayed based on the session variable customer_id and loaded into the datagrid and the textboxes.

```
    Dim intCust_ID As String
    Dim dtCustomerInfo As DataTable
    Dim drCustomer() As DataRow
    Dim i As Integer

    'Settings for paging with datagrid
    With DataGrid1
        ' Enable paging.
        .AllowPaging = True
        ' Display 5 page numbers at a time.
        .PagerStyle.Mode = PagerMode.NumericPages
        .PagerStyle.PageButtonCount = 10
        .PageSize = 10
    End With

    'display only the items relevant to the user type
    i = 1
    Select Case Session.Contents("usertype")
        Case "Staff"
            'For i = 1 To 16
            txt1.Visible = False
            txt2.Visible = False
```

```
            txt3.Visible = False
            txt4.Visible = False
            txt5.Visible = False
            txt6.Visible = False
            txt7.Visible = False
            txt8.Visible = False
            txt9.Visible = False
            txt10.Visible = False
            txt11.Visible = False
            txt12.Visible = False
            txt13.Visible = False
            txt14.Visible = False
            txt15.Visible = False
            txt16.Visible = False
            lbl1.Visible = False
            lbl2.Visible = False
            lbl3.Visible = False
            lbl4.Visible = False
            lbl5.Visible = False
            lbl6.Visible = False
            lbl7.Visible = False
            lbl8.Visible = False
            lbl9.Visible = False
            lbl10.Visible = False
            lbl11.Visible = False
            lbl12.Visible = False
            lbl13.Visible = False
            lbl14.Visible = False
            lbl15.Visible = False
            lbl16.Visible = False
            'txt(i).visible = False
            'i = i + 1
            'Next

        Case "Customer"
            DataGrid1.Visible = False
            btnTrans.Visible = False
            btnBanks.Visible = False
            btnMortgage.Visible = False
            btnPymtTrans.Visible = False
    End Select

    'set customer id to get customer data
    txt22.Visible = False
    txt21.Visible = False
    txt21.Text = Session.Contents("customer_id")
    intCust_ID = txt21.Text

    'fill datagrid with customer data
    SqlDataAdapter1.Fill(DataSetAll_Customer_Data1)
    If Not IsPostBack Then
        DataGrid1.DataBind()
    End If

    'Load Customer Data into Text Boxes
    dtCustomerInfo =
        Me.DataSetAll_Customer_Data1.Tables("Customer")
```

```
'drCustomer = dtCustomerInfo.Select("strCust_FName = '" +
    Session.Contents("FirstName") + "'")
drCustomer = dtCustomerInfo.Select("intCust_ID = '" +
    intCust_ID + "'")

If drCustomer.Length > 0 Then
    Me.txt1.Text = drCustomer(0).Item("intCust_ID")
    Me.txt2.Text = drCustomer(0).Item("strCust_FName")
    Me.txt3.Text = drCustomer(0).Item("strCust_LName")
    Me.txt4.Text = drCustomer(0).Item("strCust_City")
    Me.txt5.Text = drCustomer(0).Item("strCust_Zip")
    Me.txt6.Text = drCustomer(0).Item("strCust_Phone")
    Me.txt7.Text = drCustomer(0).Item("curMort_Amt")
    Me.txt8.Text = drCustomer(0).Item("intBank_ID")
    Me.txt9.Text = drCustomer(0).Item("intCust_Status")
    Me.txt10.Text = drCustomer(0).Item("strCust_MI")
    Me.txt11.Text = drCustomer(0).Item("strCust_Add1")
    Me.txt12.Text = drCustomer(0).Item("strCust_Add2")
    Me.txt13.Text = drCustomer(0).Item("strCust_State")
    Me.txt14.Text = drCustomer(0).Item("strCust_Zip4")
    Me.txt15.Text = drCustomer(0).Item("strCust_Email")
    Me.txt16.Text = drCustomer(0).Item("intMort_ID")
Else
    Response.Write("No Rows Returned")
End If

'Get Customer ID
Session.Contents("Bank_ID") = txt8.Text
Session.Contents("Mort_ID") = txt16.Text
Session.Contents("customer_ID") = txt1.Text
```

Appendix C.

Budget and Requirements

- Software Requirements
  - Microsoft 2003 Server Enterprise Edition
  - Microsoft Office XP
  - Adobe Acrobat 6.0 Standard
  - SQL Server Enterprise and Developers Edition
  - Microsoft Visual Studio .NET Professional 2003
- Hardware Requirements
  - The Web Server at the College of Applied Science
  - Client computer with internet connection

	Item	Explanation	Actual Cost	My Cost
Software	Microsoft 2003 Server Enterprise Edition	Have	$3,999	$0
	Microsoft Office XP	Own	$499	$0
	Adobe Acrobat 6.0 Standard	Own	$299	$0
	SQL Server Enterprise and Developers Edition	Have	$4,999	$0
	Microsoft Visual Studio .NET Professional 2003	Have	$1,079	$0
Hardware				
	UC College of Applied Science Web Server	Have	$0	$0
	Verisign	Need	$1,300	$0
	Actual Totals		$10,875	
	My Totals			$0

Figure 6. Budget

Notes

Note: The banks and mortgage companies accounts and routing is transferred to the transacting company through a secure Web site (https://direct.53.com/logon53Direct.html), via an EFT file. Due to the cost restraints, the Web site project is not implemented at this time, therefore I will simulate this by sending a text files to the Senior Server at the University of Cincinnati College of Applied Science.

Note: To simulate the transactions, I have produced an output file and placed the file on the Senior Server at the University of Cincinnati College of Applied Science.

References

1. Adobe Acrobat 6.0 Standard, http://www.adobe.com/store/products/master.jhtml?id=catAcrobatStnd, October 29, 2003.

2. Google, http://www.google.com, October 18, 2003

3. Hewlett Packard, http://www.smb.compaq.com/, October 25, 2003.

4. LZL Equity Programs Company, http://home.fuse.net/lzl/, October 15, 2003.

5. Microsoft Product Catalog, http://www.microsoft.com/catalog/default.asp?subid=22, October 22, 2003.

6. Network Solutions, http://www.networksolutions.com, December 7, 2003.

7. Nextag Incorporated, http://www.nextag.com/, October 24, 2003.

8. North American Funding Corp, http://www.northamericanfunding.nycap.rr.com/, October 20, 2003.

9. Time Warner Road Runner Connection, http://www.twcol.com/products/roadrunner_cost.asp, October 23, 2003.