

# **E-commerce Site for Slonina Inc.**

By

Frank Damen

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

May 2005

## **Acknowledgments**

I would like to thank my family for all the support and help that I received from them throughout the course of this project. The ability to bounce ideas and get meaningful responses from everyone in my family really helped during the brainstorming and creative process of this project. Specifically, I would like to thank my sister, Annamaria Damen, for her understanding and creative mind. She gave me invaluable guidance with the look and feel of the project. In addition, I would like to thank Dominic Ferreri and Tim McDade for all their support and flexibility with my work schedule, as well as helping me with the testing procedure. I also appreciate John Good's support, for when I hit a wall with a programming error, he was my best resource. His knowledge of debugging techniques, which he passed on to me, was critical to me while I was working on this project. Finally, I would like to thank Slonina Inc. for allowing me to do this project for them.

# Table of Contents

<b>Section</b>	<b>Page</b>
Acknowledgements	i
Table of Contents	ii
List of Figures	iv
Abstract	v
1. Statement of the Problem	1
2. Description of the Solution	1
2.1. User Profile	2
2.2. Design Protocols	4
2.2.1. Web Development/Multimedia	4
2.2.2. Programming	4
2.2.3. Database	5
3. Deliverables	5
4. Design and Development	6
4.1. Timeline	6
4.1.1. Senior Design I Accomplishments	6
4.1.2. Senior Design II Accomplishments	7
4.1.3. Senior Design III Accomplishments	8
4.2. Budget	9
4.2.1. Software Costs	9
4.2.2. Hardware Costs	10
5. Proof of Design	10
5.1. Web site Customer Pages Design	10
5.1.1. Web Site Entry Page	11
5.1.2. Web Site Main Pages	12
5.1.3. Web Site Product Page	13
5.2 Web site Admin Pages Design	14
5.2.1. Web Site Admin Screen Design	14
5.2.2. Web Site Administration Login	15
5.3 Database Design	16
5.3.1. Database Product Table Design	16
5.3.2. Database Customer/Order tables	18
5.3.3. Database Stored Procedures	19
5.3.3.1. Database Automated Order Entry	19
5.3.4. Database Logins	20
6. Testing Procedures	20

7. Conclusions and Recommendations	21
7.1. Conclusions	21
7.2. Recommendations	22
Appendix A	
A1. Html code modification with VB function	23
A2. Sample stored procedure	24
A3. Sample automated order entry stored procedure	25
Appendix B	26
References	27

## List of Figures

<b>Figure Number</b>	<b>Page</b>
Figure 1. Use Case Diagram for Users	3
Figure 2. Timeline	9
Figure 3. Budget	9
Figure 4. Web Site Entry Page	11
Figure 5. Basic Web Site Layout	12
Figure 6. Web Site Product Page	13
Figure 7. Web Site Admin Page	15
Figure 8. Product Database Diagram	17
Figure 9. Order/Customer Database Diagram	18

## **Abstract**

An excellent alternative for established companies to reach new customers is e-commerce; currently, it may be considered an excellent way of providing a test market for new companies and their products. The ability to quickly and cheaply implement an e-commerce site presents a real opportunity for new companies to provide actual data to support their proposed business plan. The major benefit is that it allows potential investors to clearly see a company's potential and accurately project future success. Slonina Inc., a new company, is currently designing and implementing an e-commerce site to test the market base for its products. This project covers the design and implementation of Slonina Inc.'s e-commerce site.

# **E-Commerce Web Site for Slonina Inc.**

## **1. Statement of Problem**

Slonina Inc. currently does not have a store to market its products. Although the owners have begun the process of creating a company, the necessary funding for opening a store location has become a challenge. Difficulties they encountered while searching for funding has led them to believe that they must first show that they have marketable goods before investors are willing to give them the start-up money required to open a store. To show that their products are marketable and will generate profits, Slonina Inc. is turning towards the Internet and e-commerce.

## **2. Description of the Solution**

The Web site is designed to handle all the details, including showcasing the items for sale, handling orders online, keeping track of the inventory, and collecting the money for the transactions. These tasks are be handled by different components of the e-commerce site.

The Web site itself displays the products and tries to persuade the potential clients to purchase the products from the site. The shopping cart handles the clients' information on which products they wish to purchase; then when the clients are ready, it will pass this information to a third party processor to handle collecting the money from the clients. The records of what items have been purchased must then be processed, storing the customer information, order information, and handling inventory.

Last, the order must be submitted to the company with the clients' information so that the order can be shipped out to the appropriate customers.

## **2.1 User Profiles**

Two main groups of users access the e-commerce site. The primary group is customers visiting the site. The secondary group is the site administrators.

Customers will begin by surfing the site, possibly researching the company and, hopefully, looking for items to buy. Once they have chosen to purchase an item, they will then proceed to submit their information, place an order, and receive confirmation of their order via email with the appropriate information.

The second group accessing the site is the administrators who are responsible for the content of the site. Their responsibilities include maintaining the database, removing and adding products as necessary, and making sure that all transactions between the different components of the site are functioning properly. They are also accountable for securing all data, company and customer, and ensuring that all valid orders are sent to the company to be filled.

The following Use Case Diagram will show the user roles:

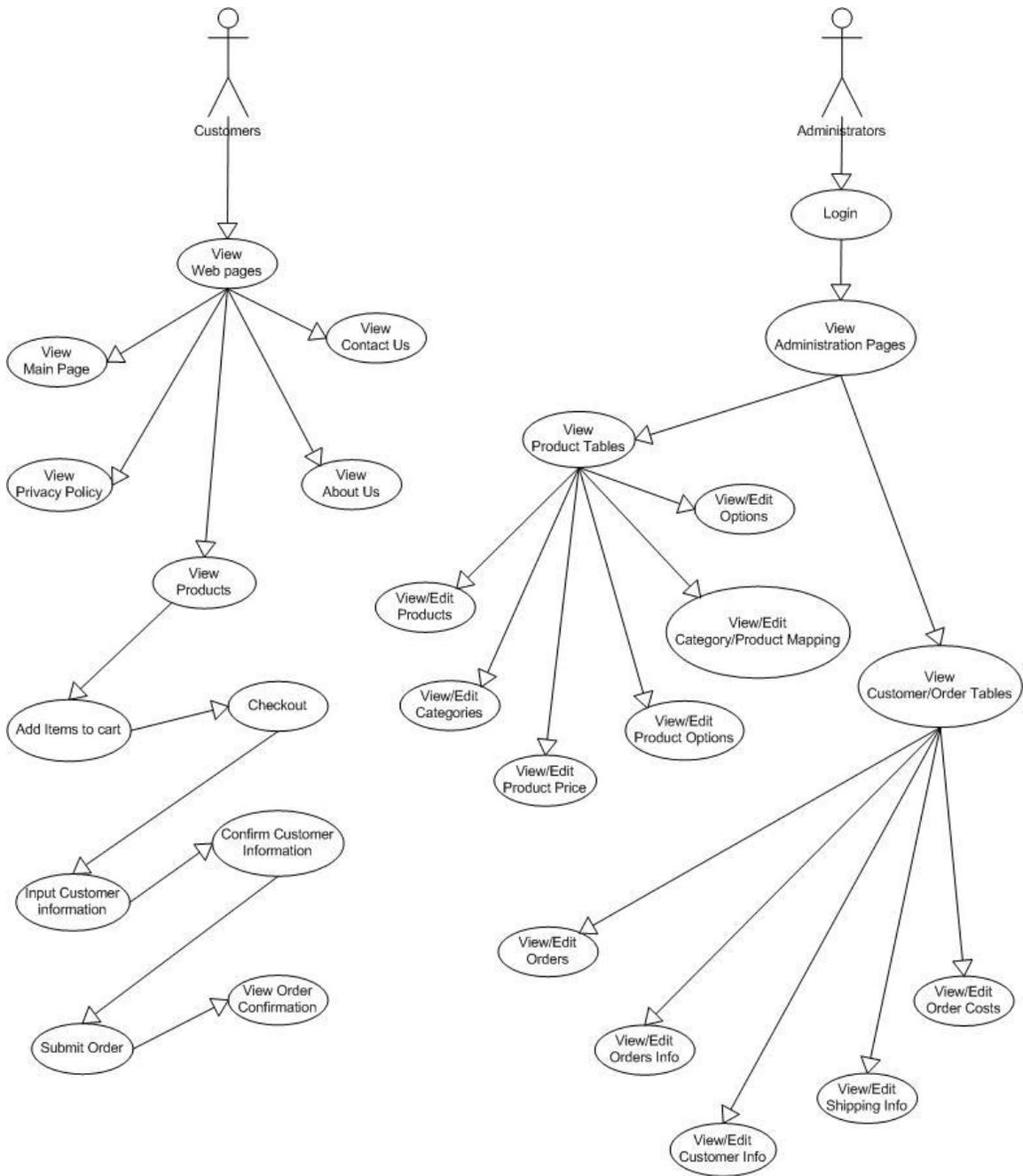


Figure #1: Use Case Diagram for Users

## **2.2 Design Protocols**

I worked with three of the four computing areas within the Information Engineering Technology program to complete this project. The main focus was on Web Development/Multimedia, followed by database and programming.

### **2.2.1 Web Development/Multimedia**

The creation of an e-commerce site requires having a Web site that is viewed by potential customers. This means care must be taken not only to include pertinent and useful information in the Web site design but also that its presentation be done in an easy to follow and comprehensible manner. The Web site must be user friendly. Since the Web site is also trying to sell products, it should keep the potential customers interested in the company and its products, hopefully, persuading them to purchase items. An e-commerce site also needs to make the customers feel that they are at a secure and legitimate Web site and that the company from which they are buying truly exists.

### **2.2.2 Programming**

Programming is required to complete an e-commerce site. Starting from the basic Web site creation, to incorporating an Internet payment gateway and database, along with a shopping cart, all require programming to get the job done. The .NET framework was used to handle this part of the project.

Programming languages used:

- .NET Framework
  - VB .net
  - Html
  - JavaScript

### **2.2.3 Database**

Since there is a need to track and record all the transaction that will occur, I incorporated an SQL database in my project. It was designed to handle product, customer, and order information. I wrote stored procedures used to access the information within the database and to make changes to it.

Programming languages used:

- T-SQL

### **3. Deliverables**

In order to provide a successful and clear project, certain items were considered critical for the e-commerce site. During the design phase of this project, the following deliverables were defined:

1. The Web site provides an online store for Slonina Inc. customers.
2. The Web site provides information on all aspects of Slonina Inc., including the following topics:
  - a. History/About Us
  - b. Mission Statement
  - c. Privacy Policy
  - d. Product Information
  - e. Contact Information
3. A general navigation bar on every page for easy navigation of the Web site.
4. The Web site uses a third-party processing service to handle the checkout process,

- i.e. confirm customer information and orders, receive customer payments, process orders, and send the customer information and orders to Slonina Inc.
5. The Web site has an administration section to handle maintaining all database information.
  6. The Web site's administration section is password protected for security.
  7. The Web site has a SQL database to store the following types of information:
    - a. Product information
    - b. Customer information
    - c. Order information
  8. The SQL database automates the following processes:
    - a. Receiving orders from the third party processor
    - b. Storing orders and customer information
    - c. Notifying Slonina Inc. when new orders have arrived
  9. The Web site interacts with the database through stored procedures for security reasons.

## **4. Design and Development**

### **4.1 Timeline**

Throughout the course of this project lessons were learned, mistakes were made and goals were reached. Below are the accomplishments of the Senior Design sequence.

#### **4.1.1 Senior Design I Accomplishments**

During Senior Design I the following tasks were accomplished:

- Analyzed existing e-commerce Web sites
- Performed research on e-commerce Web site design
  - Shopping carts
  - Third party processors
  - SSL
- Determined the site design and layout
- Began the development process
- Submitted the proposal and oral presentation

Research included determining the web hosting company and the third party processor choices. While looking for the best way to accomplish the goals and incorporating all the necessary features, I paid specific attention to how to design the site to be user friendly and show and convince the potential shoppers of the company's legitimacy.

#### **4.1.2 Senior Design II Accomplishments**

During Senior Design II the following tasks were accomplished:

- Designed base Web site layout
- Built database back-end
  - Designed and implement product tables
  - Wrote stored procedures to interact with database
- Designed Admin Section of Web site for the product tables
- Tested the existing portions of the Web site

- Submitted Design Freeze documentation and oral presentation

This phase of the project dealt with researching how to implement all the necessary features of the Web site, along with implementing enough of the Web site to be able to demonstrate functionality and error handling.

#### **4.1.3 Senior Design III Accomplishments:**

During Senior Design III, the following tasks were accomplished:

- Completed the Web site design and layout
- Completed admin section of Web site for all tables
- Built database back-end
  - Designed and implemented customer/order tables
  - Wrote stored procedures to interact with database
  - Automated the order entry process
- Tested the entire site for design and programming errors
- Presented the final project
- Completed documentation for the project

The primary work during this phase III was to complete the actual Web site and all the database functionality. The whole process consisted of heavy testing as each step of the process was completed. The following timeline shows the actual process from start to completion.

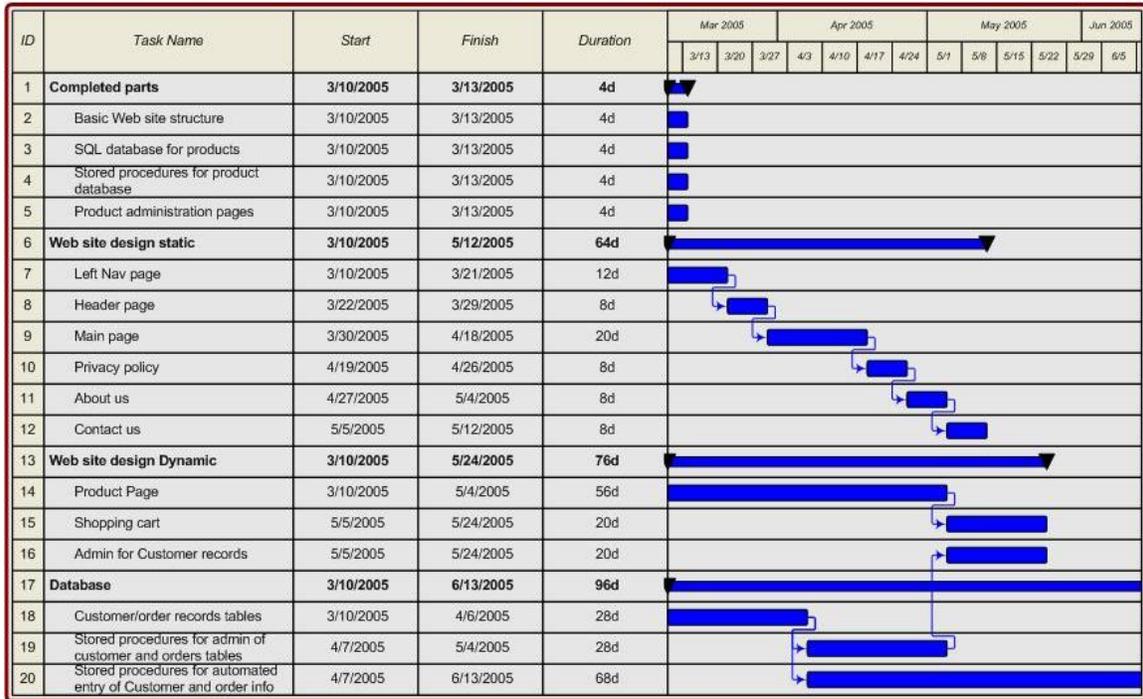


Figure #2: Timeline

## 4.2 Budget

### 4.21 Software Costs

Items	Cost
Visual Studio .NET Framework	\$549.00 One time purchase
2Checkout	\$49.00 Setup fee
HostOnce	\$83.40 Annually
SQL Server 2000 on HostOnce	\$80.00 Annually
<b>Total</b>	<b>\$761.40</b>

Figure #3: Budget

2Checkout also charges \$0.45 per sale and 5.5% of the sale amount. All expenses are the responsibility of Slonina Inc.

#### **4.2.2 Hardware Costs**

My personal computer and the computer labs at the University of Cincinnati were used to facilitate the creation of the e-commerce site. The site is hosted by HostOnce web hosting.

### **5. Proof of Design**

This section covers the actual design and implementation of the e-commerce Web site for Slonina Inc.

#### **5.1 Web Site Customer Pages Design**

The layout and design of the e-commerce Web sites was researched, specifically the best way to present the information while ensuring ease of navigation for the customers. It was determined the best way to do this was by designing the Web site with a header page that did not change along with a left navigation page that was also static. This page layout ensures that no matter where the customers are within the Web site, they will always know at which site they are and have access to all the available pages within the Web site. The use of frames makes achieving this page layout simple.

### 5.1.1 Web site Entry Page

The Web site entry page consists of one image linked to the main pages of the Web site. This is done to give a unique feel to the Web site, and it provides a page that is much more artistically designed as opposed to being merely functional.

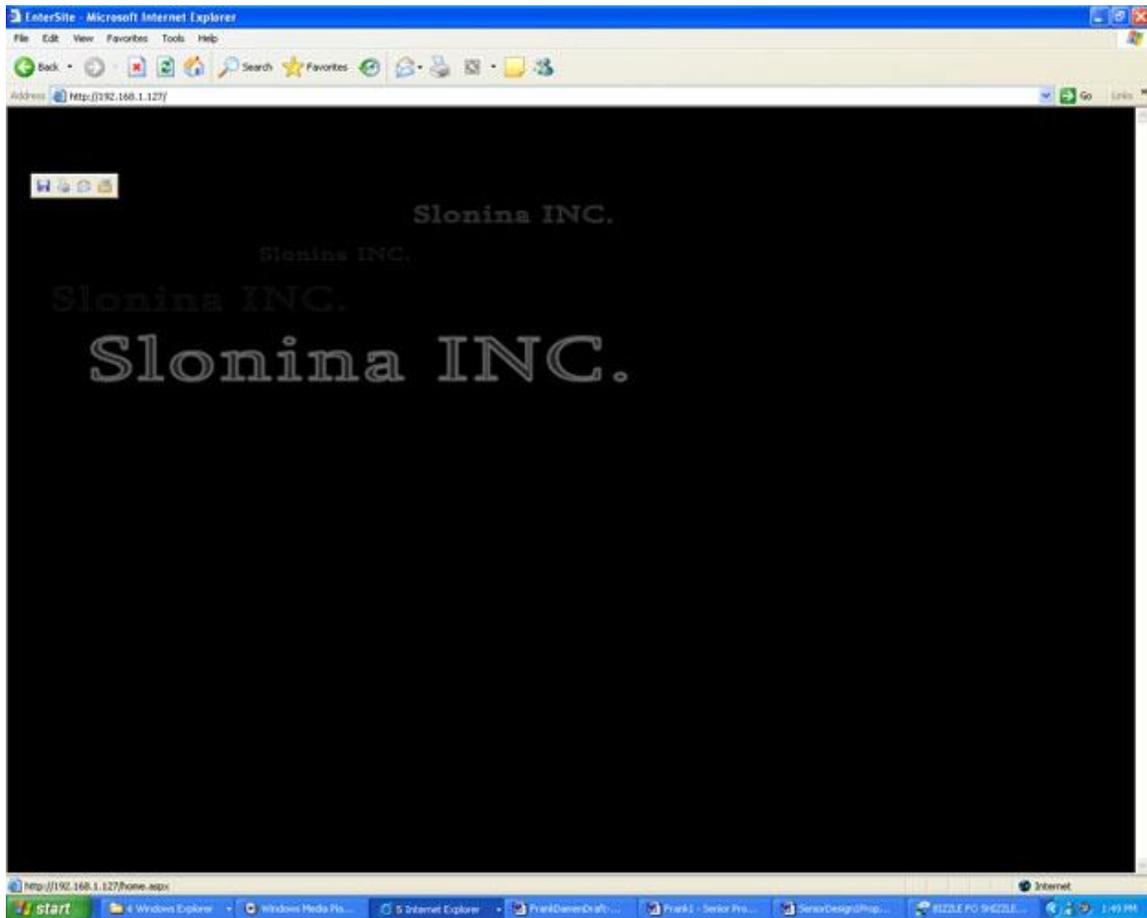


Figure #4: Web Site Entry Page

## 5.1.2 Web Site Main Pages

The Web site is designed using frames. It has a header frame and left navigation frame that will not change as the customer browses the site. This design is intended to ensure that the user will always know what site they are visiting, and they will always have the selection of web pages from which to choose. The following screen shot shows the basic layout for web site pages.



Figure #5: Basic Web Site Layout

### 5.1.3 Web site Products Page

The Web sites product page is written to use the third party processors features. These include the shopping cart and checkout process. The primary reason for this was to ensure compatibility and security while handling the checkout process. Since security is the critical issue when dealing with personal information and credit card information, it was determined be the safest way to deal with such critical information. The following is a screen shot of the products page:

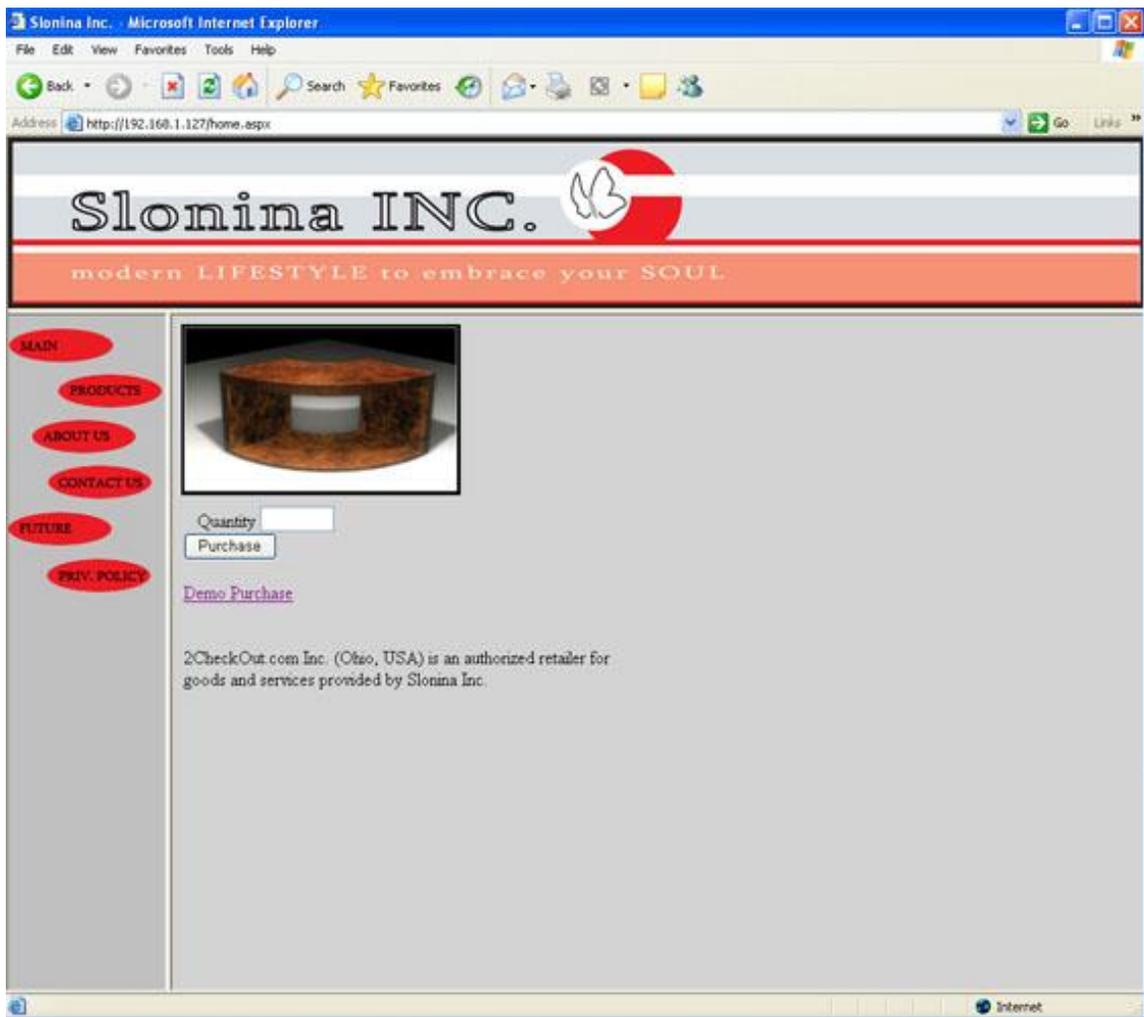


Figure #6: Web site product page

## **5.2 Web Admin Design**

The administration pages of the Web site have a similar design. They incorporate a header and left navigation page.

### **5.2.1 Web site Admin Screen Design**

The individual administration pages are designed using datagrids. The datagrids contain the following buttons:

- Edit
  - Update
  - Cancel
- Delete
- Add

The datagrids delete link has a JavaScript pop-up confirmation to ensure that records are not deleted accidentally.

The record information that is not user friendly, i.e. a number that does not mean anything to the administrator but uniquely identifies the records, is displayed in the datagrid more meaningfully. For example, instead of showing the CategoryID when adding a product to the database, the CategoryName is displayed. This information helps the site administrators when they are maintaining the database. This functionality was done by incorporating list boxes within the datagrids and writing functions to display more pertinent information while selecting the proper information when making changes to the database. See Appendix A1 for code sample. The following screen shot is an example of how the administration pages for the web site is laid out.

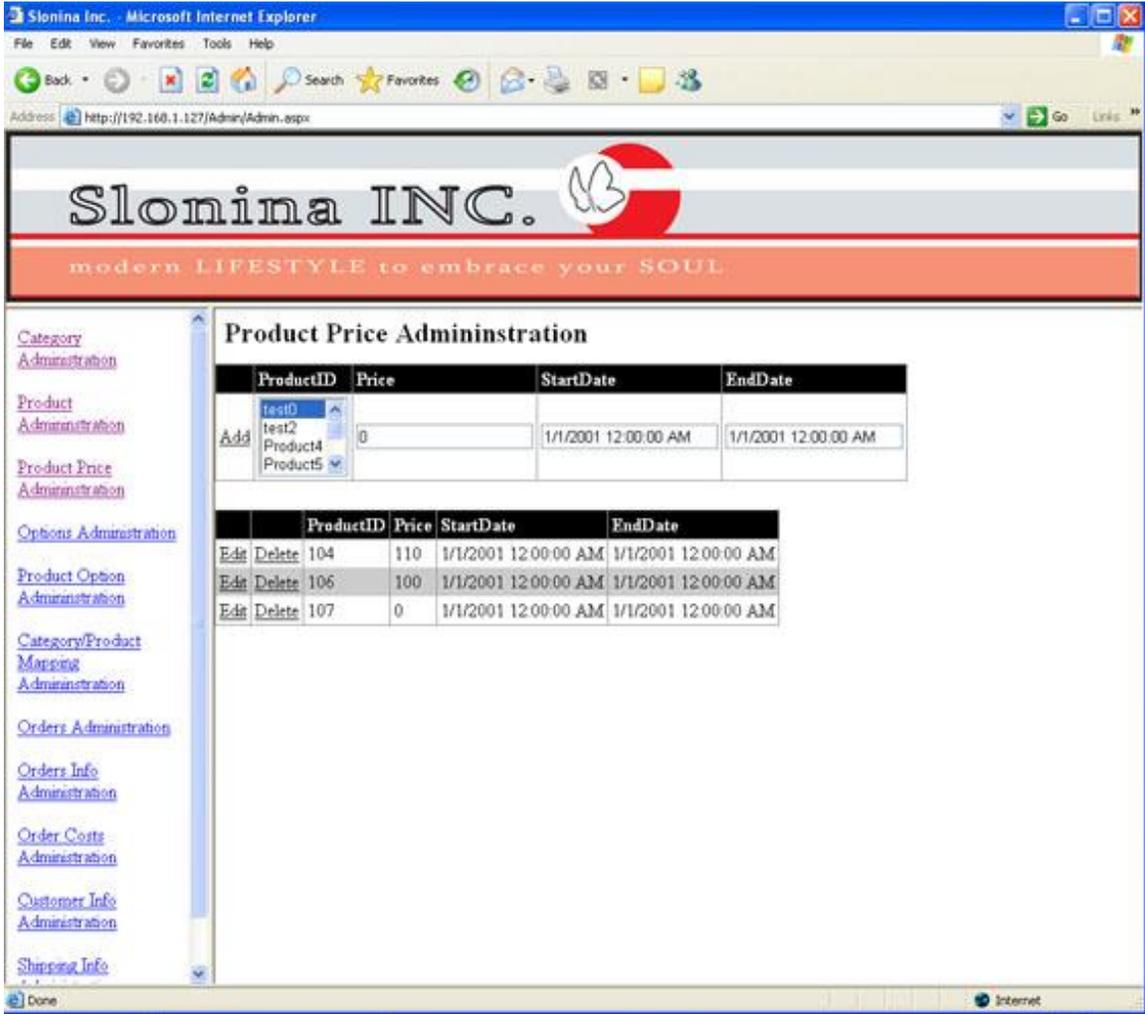


Figure #7: Web Site Admin Page

## 5.2.2 Web Site Administration Login

The entire collection of administration pages is protected using a login screen and password. This is accomplished using the built in functionality of Visual Studio .NET. By altering the web.config file to protect the Admin folder of Web site and entering user accounts, the entire folder becomes password protected. A login page is written to

confirm the user name and password against the web.config, and if authenticated, a cookie is generated granting access to the protected folders.

### **5.3 Database Design**

The Web Site uses an SQL Server 2000 database. It is designed to store the following information for the company:

- Product information
- Customer information/Order information

#### **5.3.1 Database Product Table Design**

The product tables are designed to house the product information along with any options that may be purchased as well. The database table is Normalized to the Third Normal Form. The product tables are laid in the following manner:

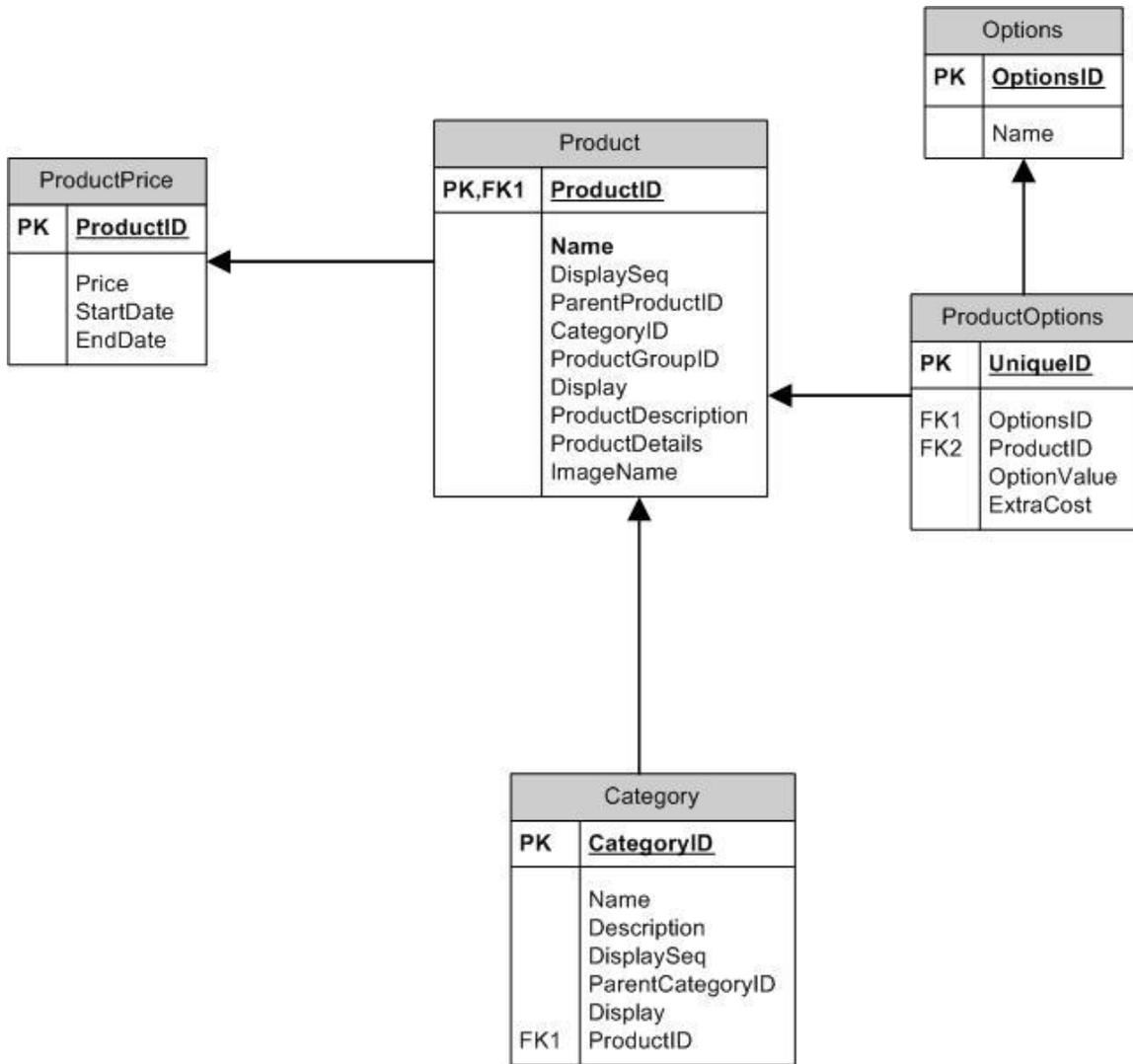


Figure #8: Product Database Diagram

### 5.3.2 Database Customer/Order Tables

The customer/order tables are designed to hold the customer information along with their order information. The database table is Normalized to the Third Normal Form. The customer/order tables are laid in the following manner:

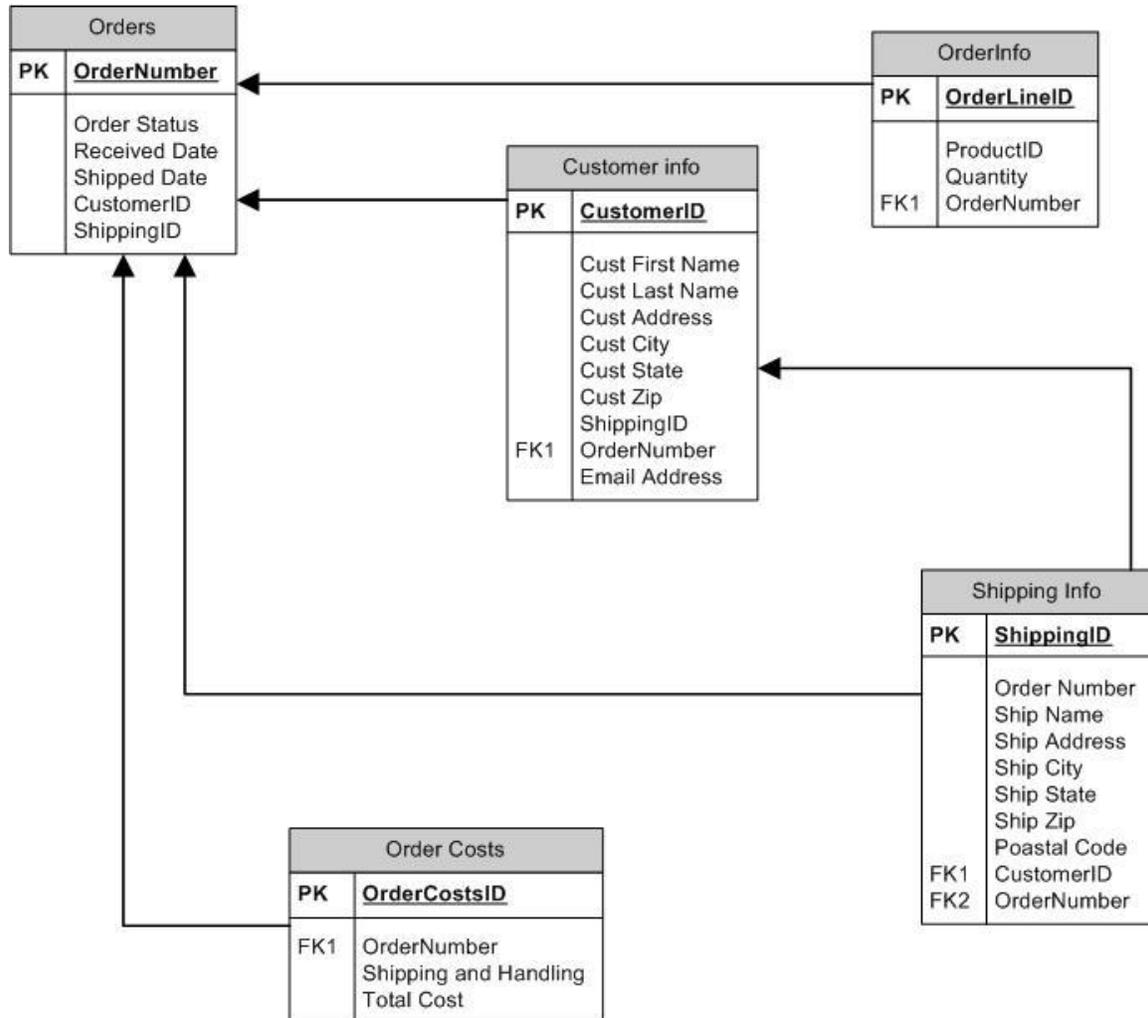


Figure #9: Order/Customer Database Diagram

### **5.3.3 Database Stored Procedures**

Stored procedures are precompiled collections of Transact-SQL statements stored under a name and processed as a unit. Stored procedures are used for all interactions between the Web site and the database. See Appendix A2 for code sample. This is used to create a layer of security between the Web site and the data tables.

#### **5.3.3.1 Database Automated Order Entry**

The T-SQL language is a very powerful tool when incorporating a database with a Web site. A great deal of processing can occur on the back-end that is an excellent tool for automating routine processes. The order entry process in this instance was automated through stored procedures. This task was accomplished by configuring the server's mail client to automatically check an email account. Then the built in xp\_ReadMail function was used to store the emails into a temporary email table. From here, a cursor stored procedure was used to read the emails looking for sales receipts. When an appropriate email was found, the body of the email was passed into another stored procedure to process the string into useful variables, and then the relative information was stored in the customer/order tables. See Appendix A3 for code sample.

By automating the order process, a couple of advantages were gained. First, time was saved by not having to manually input the data into the database, and second, the risk of mistakes being made during the entry processed was removed. Both are significant advantages and a great asset to Slonina Inc.

### **5.3.4 Database Logins**

The database login for the Web site is an account created uniquely for the site. The account is limited strictly to accessing the single database and only has the following permissions, Read/Execute for stored procedures. All other permissions have been disabled. The intention is that even in the unlikely event that the site does get hacked, the only access that can possibly be gained to the database is through the stored procedures.

## **6. Testing Procedure**

The testing of the Web site has been an iterative process. As each step of the project was completed, the following criteria was used to test the project sections:

- Functionality
- Error handling
- Data validation
- User error messages
- Administrator error messages

The testing of the Web site proceeded as follows:

1. I tested the Web site to determine potential errors.
2. Dominic Ferreri tested the Web site.
3. Novice users tested the Web site.

Throughout the testing process many changes were made and problems corrected. Many of the errors involved passing incorrect data to the database. This problem was handled in multiple ways. First, the use of list boxes helped to ensure that appropriate data was passed to the server. Since the values were being selected from the list of possible

answers, the user is unable to make an incorrect choice. Second, the data validation on both the client side and server side also ensured that only valid data was passed. By incorporating these two techniques server errors were reduced to a minimum.

The technique of using Try/Catch coding within the program also greatly reduced the number of errors that occurred. This ability allowed for a graceful way to catch and recover from programming mistakes, as well as provide a mechanism to pass more meaningful errors to the users as well as the administrators. See Appendix B for some issues and resolutions.

## **7. Conclusion and Recommendations**

The whole senior design process was an extremely valuable lesson in Web site development. Mistakes were made, but the lessons learned will be invaluable tools for the rest of my career.

### **7.1 Conclusion**

The development of an e-commerce Web site from scratch has taught me that the finished product is ultimately the goal, but that without the proper steps prior to developing the Web site, a useful and well designed site is extremely challenging if not unlikely. The research phase of the project is probably the most important step because this is where everything important occurs. From choosing the correct languages for development to the proper page layouts are all accomplished in this phase. If mistakes are made here, they can be extremely costly and time consuming to fix; sometimes can set the project back significantly or even break it. The testing process, being an iterative

process, also is a key approach when designing and implement an e-commerce Web site. This type of testing allows errors to be caught and resolved as they occur which greatly improves the overall product.

## **7.2 Recommendations**

For those who might choose to take on a similar endeavor, I would like to make a few recommendations. The most important thing I can stress is to make sure that you stay ahead of your timeline. When errors occur, and they will, time is the key element in being able to solve them. Use all the resources you have available to you. This may seem obvious, but friends, family, colleagues, and classmates are a priceless resource. Odds are when you come across something you cannot solve, someone else has had a similar problem. Although their solution might not work for you, it can and will take you in directions you have not thought of, hopefully leading to a solution.

On a more technical note, the use of frames in a Web site should be avoided whenever possible. Although they present good choice for proper layout and design, they are flawed in two extremely important ways. First, when returning to your Web site, it is not possible to specify a different page than the home page. Although this may not be important for some, when passing information to a third party processor and specifying a `return_url`, it can be critical in the actual experience the user receives at the respective Web site. Second, when users click the refresh button on their browsers, the home page will be loaded again instead of the current page they are viewing, taking away from the users' positive experience while visiting your Web site. Therefore, if you choose to use frames, you should be aware of these limitations, so start and plan accordingly.

# Appendix A

## Code Snippets

### Appendix A1

Here I edited the code for the datagrids to populate the list boxes with information from the database and wrote functions to show the correct information as well as select the correct information.

```
<asp:TemplateColumn HeaderText="OptionsID">
<ItemTemplate>
<asp:Label runat="server" Text='<#
ShowThis("Options", "Name",DataBinder.Eval(Container, "DataItem.OptionsID")) %>'
ID="Label2" NAME="Label2">
</asp:Label></ItemTemplate>

<EditItemTemplate>
<asp:ListBox ID=lstOptionsID runat="server" DataMember="Options"
DataSource="<# DataSet11 %>" DataTextField="Name" DataValueField="OptionsID"
SelectedIndex='<# SelectThis("Options", DataBinder.Eval( Container,
"DataItem.OptionsID" ) )%>'>
</asp:ListBox></EditItemTemplate></asp:TemplateColumn>
<asp:TemplateColumn HeaderText="ProductID">
```

### VB Function

```
Function SelectThis(ByVal TableName As String, ByVal This_ID As Object) As Integer
    Dim i As Integer

    Dim dtblThisTable As DataTable = DataSet11.Tables(TableName)
    If IsDBNull(This_ID) Then
        Return 0
    ElseIf This_ID = -1 Then
        Return 0
    Else
        For i = 0 To dtblThisTable.Rows.Count - 1
            If dtblThisTable.Rows(i)(TableName & "ID") = This_ID Then
                Return i
            End If
        Next
    End If
    Return 0
End Function
```

## Appendix A2

Here is a sample stored procedure used to access the database.

```
CREATE PROCEDURE dbo.spUpdateCustomerInfo
(
  @CustomerID int,
  @CustFirstName varchar(50) = "null",
  @CustLastName varchar(50) = "null",
  @CustAddress varchar(50) = "null",
  @CustCity varchar(50) = "null",
  @CustState varchar(30) = "null",
  @CustZip varchar(10) = "null",
  @ShippingID int = "null",
  @OrderNumber int = "null",
  @EmailAddress varchar(50) = "null"
)
AS
SET NOCOUNT OFF;
UPDATE CustomerInfo
SET CustFirstName = @CustFirstName,
    CustLastName = @CustLastName,
    CustAddress = @CustAddress,
    CustCity = @CustCity,
    CustState = @CustState,
    CustZip = @CustZip ,
    ShippingID = @ShippingID,
    OrderNumber = @OrderNumber,
    EmailAddress = @EmailAddress

WHERE CustomerID = @CustomerID
GO
```

## Appendix A3

The following code sample is a portion of the stored procedure used to automate the order entry process.

```
set @strVar = @curBody

--Parsing body of sales receipt email into variables
set @OrderNumber2CO = SUBSTRING(@strvar,CHARINDEX('2Checkout.com Order
Number:',@strvar)+27, 12)
Print @OrderNumber2CO
set @strQuantity = SUBSTRING(@strvar,CHARINDEX('Quantity:',@strvar)+10, 3)
set @strProductID = SUBSTRING(@strvar,CHARINDEX('Product ID:',@strvar)+11, 4)
set @strShipping = SUBSTRING(@strvar,CHARINDEX('Shipping: $',@strvar)+11, 8)
set @strTotal = SUBSTRING(@strvar,CHARINDEX('Total: $',@strvar)+8,8)
set @strCustomer = SUBSTRING(@strvar,CHARINDEX('The order was billed
to:',@strvar)+24,((CHARINDEX('This order should be shipped Ground to:',@strvar)+0)-
(CHARINDEX('The order was billed to:',@strvar)+24)))
set @strShipTo = SUBSTRING(@strvar,CHARINDEX('This order should be shipped
Ground to:',@strvar)+39,((CHARINDEX('Distributed By:',@strvar)+0)-
(CHARINDEX('This order should be shipped Ground to:',@strvar)+39)))

--Replacing carriage returns with a character
set @strCustomer = REPLACE(@strCustomer,@strCR,'_')

--Parsing the Customer information string into variables
set @strCustomer =
SUBSTRING(@strCustomer,CHARINDEX('_',@strCustomer)+2,LEN(@strCustomer)-
CHARINDEX('_',@strCustomer))

set @strCustName = SUBSTRING(@strCustomer,0,CHARINDEX('_',@strCustomer))
set @strCustomer =
SUBSTRING(@strCustomer,CHARINDEX('_',@strCustomer)+1,LEN(@strCustomer)-
CHARINDEX('_',@strCustomer))

set @strCustName = REPLACE(@strCustName,@strSpace,'_')

set @strCustFName =
SUBSTRING(@strCustName,0,CHARINDEX('_',@strCustName))
set @strCustName =
SUBSTRING(@strCustName,CHARINDEX('_',@strCustName)+1,LEN(@strCustName
)+1)

set @strCustLName = @strCustName
```

## Appendix B

### Issue and Resolutions

<b>Issue</b>	<b>Resolution</b>
Sql mail client not functioning	Outlook 2002 must be open on server for Sql client to work properly
Sql parameter errors	Used list boxes for critical fields
Delete confirmation	Used Javascript to write yes/no box
VB data type errors	Used CType function
Sql errors displaying info on web page	Try/Catch to display generic error page
Delay in automated order entry	Adjusted send/receive times and sql Job times
Stored proc. for order table not function	Renamed table to orders
Automated order entry inserting squares into database	Used ltrim and rtrim functions to clean up the variables
Parsing strings using the space character not working correctly	Used replace function to replace space character with more friendly character
Parsing strings with the carriage return character not working correctly	Used replace function to replace carriage return character with more friendly character

## References

1. Evans, Kirk Allen, Ashwin Kamanna and Joel Mueller. *XML and ASP .NET*. New York: New Riders Publishing, 2002.
2. Goodyear, Jonathan, Brian Peek and Brad Fox. *Debugging ASP .NET*. New York: New Riders Publishing, 2001.
3. Hurwitz, Dan and Jesse Liberty. *Programming ASP .NET*. Sebastopol: Oriely, 2002.
4. Kurniawan, Budi and Ted Neward. *VB .NET Core Classes in a NutShell*. Sebastopol: Oriely, 2002.
5. Lomax, Paul, Ron Petrusa and Steven Roman, Ph.D.. *VB .NET Language in a Nutshell*. Sebastopol: Oriely, 2001.
6. Lomax, Paul, Ron Petrusa and Steven Roman, Ph.D.. *VB .NET Language Pocket Reference*. Sebastopol: Oriely, 2002.
7. Mojica, Jose. *C# and VB .NET Conversion Pocket Reference*. Sebastopol: Oriely, 2002.
8. Smith, Steven A. *ASP .NET by Example*. Indianapolis: Que, 2002.
9. Worley, Scott. *Inside ASP .NET*. New York: New Riders Publishing, 2001