School of Engineering

Degree project 15 Higher education credits

BookZone web shop

Amir Alagic
September 2007
Author, Program and Year/Författare, program och år:
Amir Alagic, Datasystemutveckling 2004

Instructor/Handledare:
Fredrik Jönsson, Teknologie Doktor

Examination/Examen:
This graduation work is a part of the requirements for a Degree in Bachelor of Science with a major in Computer Science
Detta examensarbete ingår i examenskraven för Kandidatexamen i Datalogi

English Title:
BookZone web shop

Abstract:
My task was to create an e-commerce solution or web shop that is integrated with PayPal system. The intent of this paper is to show how this e-commerce web application was built and integrated with the PayPal system.
The underlying technology used in the implementation of the application includes .Net 2.0 framework (ASP.Net 2.0 and C# 2005 and Express edition of SQL Server 2005)

Language/Språk:
English/Engelska

Approved by/Godkänd av:

Prof. Daniel Einarsson Date/Datum
Examinator/Examiner
Summary

As we live in times where internet technologies have a profound impact on the ways we go about shopping, I have decided to implement an e-commerce solution in ASP.Net 2.0, C# 2005 and SQL Server 2005 Express edition for my degree project. My personal goal was to learn more about the underlying technologies, issues, and thus acquire a better insight in what exactly is needed in order to implement a web shop. More precisely, I have decided to create an online web shop where customers will be able to buy computer books and make secure online payments using the PayPal system.

My goal was to create a user friendly web shop where young computer geeks could find and buy books in an easy way. The application also includes inventory and order management system which gives a web administrator the ability to change prices, add categories, add and update products and track orders through a user friendly interface, hence without any need to go to the database and write SQL to add, update or delete products or orders.
CONTENTS

1. Introduction ......................................................................................1
   1.1. Background ...............................................................................1
   1.2. Motivation ................................................................................1
   1.3. Method ......................................................................................2

2. Technology ........................................................................................3
   2.1. .Net and web application development .....................................3
   2.2. Database ....................................................................................3

3. Requirements ....................................................................................4

4. Database design ................................................................................6
   4.1. Database diagram ......................................................................6
   4.2. Tables in database ......................................................................7
   4.3. Stored procedures in DB ...........................................................10
   4.4. Classes used to work with database ..........................................11

5. Web site development.......................................................................13
   5.1. Introduction ...............................................................................13
   5.2. Home Page ................................................................................13
   5.3. Category/Find book(s) page ......................................................15
   5.4. Book details page ......................................................................16
   5.5. Shopping basket ........................................................................17
   5.6. Checkout page ...........................................................................18
   5.7. Administration panel introduction ............................................19

6. PayPal ................................................................................................24
   6.1. Introduction to PayPal...............................................................24
   6.2. PayPal integration with BookZone ...........................................26

7. Conclusion .........................................................................................31

References ..............................................................................................32
1. Introduction

1.1. Background

I have spent a significant amount of time trying to come up with a topic for my undergraduate project that will enable me to further improve my knowledge and understanding not only of the technology, but the underlying problem domain as well. As I have always been fascinated with online shopping, in which I myself became a satisfied participant a few years ago, I have made a decision to base my work on the creation of a book web shop that enables users to make payments using the PayPal secure payment system.

I have initially planned to create a web shop that sells t-shirts, but since I have always had a great love towards books, I have decided to create my own e-commerce solution that would deal with online book sales. This way I was able to learn more about what exactly takes to create and run a web shop similar to Amazon which I have used so many times to order quite a few computer books over the past few years.

At my spare time at home I have implemented quite a few projects using Java technologies (SE and J2EE) which all prepared me to take four Java certification exams (Sun Certified Java Programmer, Sun Certified Web Component Developer – JSP/Servlets, Sun Certified Business Component Developer – EJB 2.1 and Sun Certified Enterprise Architect Part I). As I have gained quite a good understanding on Java Technologies, in order to gain also a better knowledge on .NET technologies, I have decided to implement this web shop in ASP.Net 2.0, C# 2005, SQL Server 2005 Express Edition and Visual Studio as IDE.

.NET framework is very popular and many web sites are implemented using this technology. Similar to Java technologies, it is not only used to implement a small, personal web sites, but rather it has been extensively accepted in the development community circles that deal with large scale web development for huge international companies.

This small web shop is ideal for people that are starting up their own online businesses and can be further tailored and improved to take upon a role of even more demanding online shops.

1.2. Motivation

The intent of my undergraduate project work is to create an online e-commerce solution using the Visual Studio 2005 IDE and .Net framework as primary technologies and, further on, to integrate this solution with the PayPal system. The application interface should be user friendly and have the ability to handle inventory, categories and orders through an intuitive interface, thus eliminating the need to use command line SQL in order to administer products and orders in the database.
1.3. Method

I have started this project with some basic knowledge of ASP.Net 2.0 and C#. Even though I have taken a couple of ASP.Net courses, many of the basic concepts I’ve learned in these classes had to be used in more advanced and complex scenarios. For example, it took some time to learn more about web controls in order to be able to tailor them to the needs of this e-commerce solution. To accomplish this, I have taken advantage of knowledge I have gained reading relevant literature and researching the internet.

Hence, prior to project implementation, I visited a few online shops like www.amazon.com, www.adlibris.se as well as some other smaller online book stores to better acquaint myself with the web shop functionality. When I felt that I “have gathered the requirements”, I created a database design based on these requirements and then proceeded to application development.

In addition, I have also bought a few books about .Net like: SQL Server 2005 programming (Wrox), ASP.Net 2.0 instant results (Wrox) – this one is explaining how to create few types of web applications including web shop, Professional ASP.Net 2.0 (Wrox) and Professional C# 2005 (Wrox).

I have also extensively used the CodeProject web site and some web forums for .NET developers to resolve various issues that came up.
2. Technology

This section sheds some additional light on why I have chosen to implement this project in .Net.

2.1. .Net and web application development

In order to learn another relevant technology, I wanted to implement this project in programming language other than Java. In addition, as this is a large scale project for me, I did not dare to implant it in a technology in which I did not have any previous experience. Hence, I’ve decided to go with C#, ASP.Net and Visual Studio 2005 as IDE.

C# is one of the most popular programming languages and Java programmers don’t need to spend too much time to switch from Java to C# but still it has enough differences that one has to roll up his sleeves in order to gain a deeper understanding. There are many job postings for C# programmers, and ASP.Net 2.0 and Visual Studio or Visual Web Developer 2005 Express Edition are powerful tools for rapid application development (RAD). SQL Server 2005 Express Edition is also a very good RDBMS and it is free like Visual Web Developer 2005 Express Edition and unlike some free open source RDBMS it has a very good GUI that makes it very easy to create databases.

I also could have implemented this web application using Java JSP, Servlets, and JSF. Even though NetBeans IDE now has Visual Web pack that is almost as good as Visual Studio, ASP.Net 2.0 has more components, which makes it a little bit easier to work with, and hence it requires less time to complete the project. Other languages like Ruby (with its Rails framework), Python (with its Django framework) with sizable and active developer communities were interesting candidates, but I believe that even though these are great languages and frameworks, currently there is much more demand for .Net. .Net has larger community and there are more books on .Net than on Ruby or Python.

2.2. Database

I used SQL Server 2005 Express Edition database. This is a free version of Microsoft SQL Server RDBMS and its targeted use is in embedded applications and smaller scale applications¹. It allows users to easily create databases, tables, columns, and add data into database form Visual Studio.

¹ For more details see http://en.wikipedia.org/wiki/SQL_Server_Express and http://www.microsoft.com/sql/editions/express/default.mspx
3. Requirements

In order to gather the project requirements, I explored the functionality of popular online book stores like www.amazon.com, www.bn.com, www.adlibris.se as well as a few smaller web shops.

Upon the completion of the initial exploration phase, I have decided to create a web shop solution that is easy to navigate, that includes easily accessible shopping basket, ability to order books, and proceed to checkout. Since the PayPal is a very popular way to pay online I decided to integrate this online secure payment system into the web shop. In addition, I have also created Administrator’s panel so that administrator can easily manage inventory and orders through intuitive GUI, thus eliminating the need of using SQL to manage the underlying DB whenever one needs to add or remove a product or track orders.

The Figure 3.1 below depicts what actions a customer can perform.

![Customer Use Case Diagram](image)

**Figure 3.1 – Customer Use Case Diagram**

Hence, as shown in the Figure 3.1, site user can search for a book if s/he enters a keyword and clicks on the “Search” button. If there is a book that contains the desired keyword in its name or description then the book will be displayed.

When a site user comes to the web shop s/he can see what book categories are available. Currently there are four categories or groups of books (Java, Ruby, PHP and .Net) and when site user clicks on a desired category all of the books from that category will be listed.
In order to get a detailed description of the book, a user needs to click on the book image or name. After doing so, the customer is redirected to a new page where s/he can see the book details. Once the book description is expended, the user has the ability to add the book to the shopping basket.

Above the buttons listing the categories is the shopping basket button which user can access by clicking “Go to basket” button. Here s/he can see the contents of the shopping basket, change book quantity, remove book(s) from shopping basket and, if done shopping, proceed to checkout.

Figure 1.2 depicts use case diagram for Administrator.

\[\text{Figure 3.2 – Administrator Use Case Diagram}\]

As it can be seen from the figure 3.2, the administrator has the ability to list all the books in the database as well as change the book info by clicking on update button, the options only available after logging into the web site using administrator sign in credentials. Administrator can also add a new book to database and s/he can add, edit or remove a book category from database. To keep track of online orders, the administrator has the ability to list orders for any chosen time period. In addition, the administrator can update the order status (in process, cancelled, and shipped).

With the use case diagrams defined, the next step is to create web page prototypes as well as the underlying system design.
4. Database design

Based on the gathered requirements, and better understanding of the system’s functional requirements, my next task was to design a database which would host relevant web shop data. As with the most of the software projects, no one phase is set in stone, but rather an evolving process, and hence, as I proceeded, I realized that certain changes needed to be made to the initial database design.

4.1. Database diagram

![Database Design Diagram]

Figure 4.1 – Database Design

As shown in the figure 4.1, there is a total of eight tables in the web shop database. The Book table is used to store information about books and it has many to many relationship with the Author table. For this reason a cross table BookXAuthor was created. The Book table also contains CategoryID column so it can be linked to Category table. The ShoppingCart table has info on book id (link to Book table), number of books that customer ordered and time when the order was created was created like globalID – GUID that is unique and has 36 characters.

Customer table contains relevant customer info and it references the Order table via CustomerID field. Further more, the Order table references the OrderStatus table via OrderStatusID. The OrderStatus table contains valid Order Status states (in process, cancelled or shipped.)
4.2. Tables in database

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>BookID</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>CoverImageURL</td>
<td>varchar(250)</td>
<td></td>
</tr>
<tr>
<td>Price</td>
<td>money</td>
<td></td>
</tr>
<tr>
<td>BookName</td>
<td>varchar(150)</td>
<td></td>
</tr>
<tr>
<td>CategoryID</td>
<td>smallint</td>
<td></td>
</tr>
<tr>
<td>OnSale</td>
<td>bit</td>
<td></td>
</tr>
<tr>
<td>Pages</td>
<td>smallint</td>
<td></td>
</tr>
<tr>
<td>Publisher</td>
<td>varchar(150)</td>
<td></td>
</tr>
<tr>
<td>ISBN</td>
<td>varchar(20)</td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td>varchar(50)</td>
<td></td>
</tr>
<tr>
<td>Details</td>
<td>varchar(MAX)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.2.1 – Book table

As depicted in Figure 4.2.1, the BookID field is the primary key field in Book table and all of the columns must be filled (no null value is allowed).

CoverImageURL represents the name of the image file (for example .jpg, .gif, .png) containing book cover. It can be up to 250 characters long.
Price field contains book’s price and is money data type.
BookName field contains book’s name.
CategoryID field is associated with CategorID column in the Category table.
OnSale field serves as indicator of whether a book is on sale.
The Pages, Publisher, ISBN, Language and Details fields are self explanatory.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthorID</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>AuthorName</td>
<td>nchar(60)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.2.2 – Author table

This simple table is used to store info on book author. AuthorID is primary key field and AuthorName field contains author’s name.
Figure 4.2.3 – BookXAuthor table

BookXAuthor table links Book and Author tables, thus making it possible to have many to many relationship (a book can be written by more than one author and author can write more than one book).

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>BookID</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>AuthorID</td>
<td>int</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.2.4 – Category table

The Category table is “referenced” from the Book table and between these two tables we have one to many relationship (one category can be associated with more than one book).

Category field contains category name – description. I have loaded this table with four book categories: Java, Ruby, PHP and .Net.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>CategoryID</td>
<td>smallint</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>varchar(100)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.2.5 – Customer table

All the columns in the Customer table – Figure 4.2.5 are self explanatory. All columns except the PhoneHome and PhoneMobile are mandatory and not null.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomerID</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>FirstName</td>
<td>nchar(30)</td>
<td></td>
</tr>
<tr>
<td>LastName</td>
<td>nchar(30)</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>nchar(30)</td>
<td></td>
</tr>
<tr>
<td>City</td>
<td>nchar(30)</td>
<td></td>
</tr>
<tr>
<td>ZIP</td>
<td>nchar(10)</td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td>nchar(40)</td>
<td></td>
</tr>
<tr>
<td>PhoneHome</td>
<td>nchar(20)</td>
<td></td>
</tr>
<tr>
<td>PhoneMobile</td>
<td>nchar(20)</td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>nchar(50)</td>
<td></td>
</tr>
</tbody>
</table>
The Customer table could also be limited to CustomerID, FirstName and LastName columns and additional column CustomerDetailsID. Then a separate table could hold CustomerDetails info. However I do not think that this is necessary as SQL query would be little more complex and the query execution time would be longer - “Normalize until it hurts and de-normalize until it works.”

**Figure 4.2.6 – Order table**

This table is used to store info about orders. OrderID is primary key and similarly to other table keys it is automatically generated using auto increment feature. OrderStatusID column is used to reference to OrderStatusID in OrderStatus table. CustomerID field is used to reference Customer table, and thus obtain relevant customer info. OrderDate is date when order was created. GlobalID column (GUID) stores a unique 36 characters long number. The same number can also be found in ShoppingCart table where besides GlobalID we also have BookID and BookQuantity columns. Using these two columns we can get customer info as well as the info on books that are in the shopping basket. ShipmentDate is set to null when the order is made. Once the order is shipped, this field is filled.

**Figure 4.2.7 – OrderStatus table**

This table holds order status info, i.e. “cancelled”, “shipped” or “in process” – to be shipped. Administrator’s panel allows the web shop administrator to alter the status.
This table has column `TimeCreated` with info on when the shopping cart was initially created. BookID field is primary key which enables us to reference Book table in order to get more detailed info on the ordered book. BookQuantity column is holding info on the number of books that a customer ordered. To calculate price, this number is multiplied with price from the Book table. GlobalID is a unique identifier which links the Shopping cart table to Order table. While this design is fairly straightforward and easy to implement, one of its drawbacks is that for each different book in shopping cart there is a new row in the Shopping cart table. All the different rows from the Shopping cart table that belong to the same shopping basket share the same GlobalID, the field also used to link the Shopping cart table to the Order table.

### 4.3. Stored procedures in DB

The most of the database queries in this project are done using the stored procedures. It seems that there is no consensus among the developers’ community when it comes to using stored procedures. Some argue that stored procedures are safer and faster as they are precompiled, while others argue that there is no proof that they are faster and that they tie implementation to a specific db vendor. However, if one is not careful enough to use standard SQL, the application can be tied to specific db vendor without using the stored procedures.

As I like the idea of separating SQL from code and I don’t intend to move to another DB vendor, I decided to go with the stored procedures. Hence, for this project, I have created 29 stored procedures.

Here is an example of one fairly straightforward stored procedure:

```sql
ALTER PROCEDURE InsertAuthor(
    @BookID int,
    @AuthorName nchar(60)
)
AS

DECLARE @AuthorID int

    INSERT INTO Author (AuthorName) VALUES (@AuthorName)

    SELECT @AuthorID = @@Identity

    INSERT INTO BookXAuthor (BookID, AuthorID) VALUES(@BookID, @AuthorID)

RETURN
```
As the name of the stored procedure says InsertAuthor, the procedure is used to insert an author into database. As mentioned earlier, here we have many to many relationship between book and author.

In this stored procedure we have two input parameters or arguments and first is @BookID that is integer type and second is @AuthorName that is nchar type. There is declared @AuthorID, integer type.

After this we have INSERT statement which inserts AuthorName into Author table and AuthorID – primary key is stored in @AuthorID variable. Since we have many to many relationship we need to insert AuthorID and BookID into BookXAuthor table and we do this with the second INSERT statement that inserts BookID and AuthorID into BookXAuthor table.

Hence, a stored procedure is called from a class that sets the procedure parameters and adds them to DBCommand object upon which the stored procedure is called to make select, insert, update or delete on database.

4.4. Classes used to work with database

Stored procedures in this web application are used in order to manipulate db data using SELECT, INSERT, DELETE or UPDATE SQL commands.

SELECT is query operation that when executed will return results from database and in this project we are interested in DataTable object that will be used on GridView, DataList components.

INSERT is not query operation like DELETE and UPDATE and when these operations are executed we don’t get any other result but integer number that is telling us how many rows were affected with that operation.

It is also possible to create one method that will execute non query operations and it will work for all INSERT, DELETE and UPDATE statements. It is not necessary to have method for each of them. Therefore, I have created a class GeneralDBManager.cs that has method to execute query operation (SELECT) , non query operations (INSERT, UPDATE and DELETE) and one method to execute scalar.

We could have a separate method for each stored procedure that will on DBCCommand object open connection and execute query, non query or scalar and then close connection. But even though this is only a few lines of code it would violate DRY (Don’t Repeat Yourself) principle. We have 29 stored procedures and now all of them are creating GeneralDBManager object and then if SELECT is to be executed SelectFromDB() method will be called for all query actions.
public DataTable SelectFromDB(DbCommand sqlCommand)
{
    DataTable dataTable = null;
    try
    {
        //open connection
        sqlCommand.Connection.Open();
        //create DbDataReader object and execute sqlCommand object
        DbDataReader dbReader = sqlCommand.ExecuteReader();
        dataTable = new DataTable();
        //load data from dbReader object into dataTable
        dataTable.Load(dbReader);
        //close dbReader object
        dbReader.Close();
    }
    catch (Exception e)
    {
        //send exception object as in parameter to GetError static method
        //and send mail to admin if bool in Web.config is set to true
        MailUtil.GetError(e);
        throw e;
    }
    //to make sure that connection is closed Close() will be called in
    //finally block
    finally
    {
        //close connection to DB
        sqlCommand.Connection.Close();
    }
    //return data table object with data
    return dataTable;
}

In code snippet above you can see how one of methods in GeneralDBManager looks like.
DbCommand object is sent as parameter and then a connection is opened on it. After that we
call ExecuteReader() method on DbCommand object that returns DbDataReader object and
on DataTable object we call its Load() method with DbDataReader as input parameter that
will populate DataTable with data from DbDataReader. If exception is thrown then static
getError method is called from MailUtil class and exception is rethrown from catch block. In
finally block we close connection to database because finally block will execute even if
exception is thrown and we don’t want to have open and unused connections to database.
5. Web site development

5.1. Introduction

In this phase I have decided to divide the application into two parts. The first part is the actual web shop, used and accessed by customer, and the second part is administration panel used by web site’s administrator.

ASP.Net 2.0 allows us to create a reusable template or master page. Hence, I created one master page for web shop and another for administrator panel.

In ASP.Net 1.x programmer reuse was in fact copy paste of code. However, once the master page is created it becomes a template for all pages that use it. Thus, all these pages will have the same look and feel.

I have also decided to create a few user controls. These controls can be simply added to web application where it is necessary.

Admin folder contains all web forms that are used for site administration. In App_Code folder we have C# classes that are used to work with database and web.config file.

Images used in web forms are stored in “Images” folder while images of books are stored in “BookImages” folder. The book images are accessed via file system, rather than stored in the DB as BLOB, as this improves the application performance.

Web user controls are in “UserControls” folder and other “general” files are in web application root folder.

5.2. Home Page

When customer comes to my online store, s/he will come to index.aspx page. On this page, as depicted on the next page, you can see a welcome message that tells site user who created this web application as well as the technologies used for development -ASP.Net 2.0 was used like C#, Cassini server... This server comes with Visual Web Developer Express Edition. Programmers that have Windows XP Home Edition can’t run IIS.
By clicking the login button in the upper left corner and inputting the correct credentials administrator can log into the web site in order to manage orders and products.

Underneath the login button is the Shopping Basket where site customers can view the contents of the basket and the price of the order. By clicking on “Go to Basket” link, the client can make changes to the contents of the Shopping Basket and proceed to checkout.

Below the Shopping Basket is a menu implemented as web user control that displays categories from the database. In order to display books belonging to a category user can click on category name and another page will be shown with book list belonging to the chosen category.

Customer can also search books by keyword by entering the keyword text in the box next to the “Find Books” button.
5.3. Category/Find book(s) page

The picture below displays page listing books by category.

![Picture of book page]

Figure 5-2. Show books in category page

There is a maximum of four books listed on a page, and if there is more than four books in a chosen book category, user should use the “next page” link to display the remaining books. A book image is displayed for each book, and by clicking on the image or a book title listed below the image, a page showing detailed book information is displayed. In addition, on this page user can add the book to the shopping basket.

The number of books displayed per page can be changed in web.config since this value is used to determine number of books on page. In web.config we can also manage the number of characters shown in details text.

If user enters keyword into text box and then clicks on “Find Books” button s/he will get almost the same page as when category is clicked. But now page will show books that have in title or description keyword that customer is searching for.
5.4. Book details page

Below you can see what customer sees when s/he clicks on book image or book name.

![Book details page](image)

**Figure 5-3. Show book details**

Now customer can see more information about book. S/he can see author(s), ISBN, Language and detailed book description.

There is Add To Basket button that user can click if s/he wants to add this book to shopping basket. If customer clicks another time then one more book is added to shopping basket and price is increased by books price.

When user clicks on “Back to Shopping” button then s/he will be returned back to last page and in our case user will be redirected back to page where all books in Java category were shown. In MasterPage.master.cs code is implemented that makes that possible otherwise last page would be also the book detail page because when user click on “Add to Basket” button page is reloaded and last page will be in fact the same book details page. When user visits ~/index.aspx","~/Login.aspx","~/ShowBooks.aspx" or "~/ShowBooksFind.aspx" one of these is stored in Session object and when user clicks on “Back to Shopping” button then it is reading from Session object page name where user should go.
5.5. Shopping basket

If customer wants to buy a book and s/he clicks on “Go to Basket” link in Shopping basket then s/he will be redirected to page BookzoneBasket.aspx. Below you can see how this page looks.

![BookZone Shopping Basket](image)

**Figure 5-4. BookZone Shopping Basket**

As you can see it looks similar to a shopping cart that you had a chance to see when ordering online. Customer can change number of books s/he wants to buy and clicking the “Save Changes” button will ensure that the changes made are saved. It is also possible to delete a book from a shopping cart if customer no longer wants to buy it.

Should user want to buy more books after changes are made then the “Back to Shopping” button should be used. To go back to the home page (index.aspx) customer should click on top banner which is image button control. By doing so, the customer will be redirected back to index.aspx.
5.6. Checkout page

When customer wants to complete the order then s/he should click on “Checkout” button in order to be redirected to Checkout.aspx.

![Checkout page](image)

**Figure 5-5. Here is Checkout.aspx page where customer should enter contact information**

In this web form user should enter contact details like first and last name, address, email… Some of these fields are required and if user doesn’t enter first or last name, address, city, zip, country or email s/he will be asked to enter this information if s/he wants to checkout. If s/he doesn’t want that then user can click on “Home” and will be redirected to index.aspx or can click on any category s/he wants to see.

As I mentioned earlier, it is not required to fill out the phone or mobile text boxes. However, the customer must enter delivery and email address. The email address is used to communicate with the customer, and to send shipment notifications. The email address format is also code enforced. If customer writes some text or numbers that don’t follow email pattern then customer will be notified when s/he clicks on “Checkout with PayPal” button. When customer clicks on “Checkout with PayPal” button s/he is redirected to PayPal system. I will discuss the PayPal payment system later on.
5.7. Administration panel introduction

Administration panel is part of application that is accessible only by administrator(s) of BookZone web site. There as you’ll see administrators can change info about book(s), add new books, delete existing, manage orders etc…

*Administration.aspx*

![Administration.aspx](image)

*Figure 5-6. Administration.aspx is page where administrator see books*

When administrator is on this page s/he can see books that are in database. On this page administrator can see some information about book(s) but can’t change them. If administrator wants to change certain information like book price or book description (not visible here) then administrator must click on update book link and new page will be opened where administrator can change book info.
**UpdateBook.aspx**

Below you can see look of UpdateBook.aspx where administrator can change book info. It is possible to change book name, price, ISBN, book description, author name… Administrator needs to click on “Edit” button in the row s/he wants to make changes. Of course bookID, authorID columns should not be changed and are read only.

Also on UpdateBook.aspx there is “Delete Book” button and when administrator clicks on that button that book is removed from the database.

---

**Figure 5-6. UpdateBook.aspx is page where administrator can change book properties**

There are three GridView components on this page. The first one displays Authors’ names, bookID and authorID. Below is another GridView component that contains columns with price, book name, on sale check box, publisher name, ISBN and language. Finally, there is GridView component that has column called “Details” that has detailed information about the book.
*AdminCategory.aspx*

If administrator wants to change, remove or add category then s/he has to click on “Category” button and new page will be opened where administrator will see all categories in GridView component and will be able to edit, remove category.

Right below GridView component there is text box component and button “Add Category”. When user clicks on that button it takes category name from text box and sends it to stored procedure that is creating new category in database.

*Figure 5-7. AdminCategory.aspx is page where administrator can manage categories*
ShowOrdersByDate.aspx

Figure 5-7. ShowOrdersByDate.aspx is page where administrator can choose for what time interval s/he wants to see orders

When administrator clicks on “Orders” button then ShowOrdersByDate.aspx web form – page is opened in a browser. On this page administrator can choose date range for which the orders made should be displayed. It is easy to choose date because I used two calendar controls and on first one administrator chooses start date and on another administrator chooses end date. Also when administrator chooses start date the end date will be set one day after start date. If there are no orders for the chosen interval then administrator will be notified with message that will be shown in label control.

But if there are orders then administrator will see them on the same page in GridView control that will be shown in lower part of web page.

Here the administrator can click on See/Edit link in GridView component if s/he wants to change order status.
On this page administrator can change order status. Order can be cancelled, in process and shipped. Administrator can see customer info (name and last name, address, email…) and can see order date, order status, order id and shipment date if order is shipped. If order is not shipped then there is no value in shipment date column. When administrator checks shipped column and confirms this change, the shipment date is inserted into DB.

And administrator can access this part of application when s/he is logged in and in login control administrator will see link to administration panel.

After logging administrator will se that s/he is logged in and to visit administration panel administrator has to click on “Admin Panel” link. When administrator wants to log out s/he should simply click on “Log out” button.
6. PayPal

6.1. Introduction to PayPal

PayPal payment system was launched in 1998 and since then it is growing very fast in terms of both customer base and revenues. In 2002 eBay bought PayPal when its own payment system BillPoint was not well received among customers.

“Over 100,000 people sign up for PayPal’s service every day, more than $1,000 goes through the PayPal financial engine every second... PayPal is now available in about 190 countries and 17 currencies, with even broader expansion planned for the future. PayPal is the world’s fastest growing global currency exchange, and it is clear that PayPal is creating the new standard in online payments.”

Diagram below is showing growth of PayPal user base.

![Total PayPal account growth](image)

**Figure 6-1. Growth of PayPal accounts**

Since PayPal is very popular and secure system I decided to use PayPal as a payment system for my web site. There are many ways to integrate web site with PayPal and PayPal is even offering its own shopping cart but as you can see I implemented my own shopping cart instead.

---

2 Pro PayPal E-Commerce by Damon Williams, page 1
3 IBID, page 13
According to Damon Williams by using PayPal as payment option online shops see an average increase of sale by 14%. It is easy to understand because there are more than 130 million PayPal accounts. And PayPal is popular because it is secure. Buyers don’t give their credit card information to online shop and therefore PayPal customers feel safe and secure especially when they are buying things on small web shops that are not so popular.

There are other similar systems like PayPal but since PayPal is the most popular and has broad user base and given the fact that many books are published about PayPal I decided to use this online payment solution.

It is possible to create few types of accounts on PayPal system.

**Personal account**

This type of account should be used by people who are not selling or if they are selling they aren’t selling a lot of things. Main purpose of this account should be to buy online stuff and to exchange money with friends and relatives. One advantage of this account type is that to receive up to 500 dollars, monthly, there is no transaction fee. But if you receive more than 500 $ monthly then you’ll have to upgrade your account.

**Premier account**

This account is useful if you want to sell things online or if you want to receive money because with this type of account other PayPal customers can send you money from their credit cards and with “Personal account” that is not possible. Also users that have Premier account PayPal Shopping cart to sell products on their web site.

**Business account**

*Business account* has all benefits of *Premier account* but it is also offering more services. Finally, you must have a Business account in order to apply for PayPal’s Website Payments Pro payment solution.

**Sandbox account**

This type of account is used for testing. It is simulating real PayPal account but no money is transferred from account to account.

<table>
<thead>
<tr>
<th>Account Benefits</th>
<th>Personal</th>
<th>Premier</th>
<th>Business</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send money</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>24-hour fraud surveillance</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

4 Pro PayPal E-Commerce, page 15
5 IBID
It is free to send money but if one wants to get money then there is a fee that you should pay except in case of Personal account and up to 500$ monthly.

**Monthly Revenue Transaction Fee**

- $0–$3,000 2.9% + $0.30
- $3,001–$10,000 2.5% + $0.30
- $10,001–$100,000 2.2% + $0.30
- > $100,000 1.9% + $0.30

To get information about PayPal integration one can go to [www.paypal.com](http://www.paypal.com) and there it is possible to find PDF files that explain how to integrate PayPal with web application. It is also possible to buy book(s) about PayPal and one of them is “Pro PayPal E-Commerce” by Damon Williams, published by Apress. But before I got that book I downloaded “PayPal Website Payments Standard Integration Guide”, last update May 2007 where it is explained how to integrate web shop with PayPal system.

### 6.2. PayPal integration with BookZone

It is possible to integrate web shop and PayPal system in many different ways. Pro PayPal E-Commerce book explains this on almost 300 pages. I have also found on PayPal’s web site two PDF files one is “Website Payment Standard Integration Guide” and has 140 pages and other is “Website Payments Pro Integration Guide” on 58 pages. Since going in depth is beyond the scope of this paper, I will focus on integration of the PayPal with my web shop.

There are many ways to make payments using the PayPal system. PayPal system can be integrated even in web sites that aren’t selling anything. For example if one creates web site with free cooking recipes it is possible to display “Donate” button and visitor that clicks on...
that button will be redirected to PayPal system to donate money for web hosting. The “Website Payment Standard Integration Guide” shows different ways to integrate PayPal’s Shopping Cart into web shop. Hence, with very little HTML code it is possible to make checkout button for web site where in HTML web developer should have values for total amount – price, email address of web shop that is using as PayPal account ID, order name. However this is not secure and to make payments as secure as possible this code should be encrypted… While I was searching for information on Papal web site I found that there is PayPal SDK (Software Development Kit) for ASP.Net and I downloaded web controls “PayPal_ASP.NET_Web_Controls_1.0.5.msi”.

I didn’t find any info on how to work with controls in PDF documents I have previously downloaded (“Website Payment Standard Integration Guide” and “Website Payments Pro Integration Guide”). However, this was not a problem since it is pretty straightforward to use installed web controls.

As you can see from diagram above customer will add book to basket (one or more). When customer wants to buy book(s) he will click on “Checkout” button and will be redirected to page where customer must enter address info that will be stored into BookZones database. After that customer will be redirected to PayPal system where customer must log in and after that customer can confirm order.

*Figure 6-3*
When customer confirms order then PayPal system will send e-mail to BookZone's administrator with order number and price of order and customer details. Administrator will be able to see that order in admin panel and with all that information administrator will be able to send order to customer.

As you can see below PayPal button that I use to integrate BookZone with PayPal system has properties that will be set with parameters that will tell PayPal system what is BookZone's administrator e-mail address, what is order number, what is order price.

After installation in Visual Studio 2005 I got new controls in “Toolbox”.

**Figure 6-4. PayPal system (to get page like above session must be active also as you can see on picture above Total is 26$ and number 16 is order number and are not in connection to previous picture)**
As I made my own shopping cart I was not interested in "AddToCartButton", "CheckoutCartButton" and other buttons pertaining to shopping cart. As I have implemented my own shopping cart, I only needed to use “BuyNowButton”.

Let me first explain why I didn’t use HTML code to make payment as it is explained in “Website Payment Standard Integration Guide”. Here is code that is written there:

```
<form action="https://www.paypal.com/cgi-bin/webscr" method="post">
<input type="hidden" name="cmd" value="_cart">
<input type="hidden" name="upload" value="1">
<input type="hidden" name="business" value="seller@designerfotos.com">
<input type="hidden" name="item_name_1" value="Aggregated items">
<input type="hidden" name="amount_1" value="3.00">
<input type="submit" value="PayPal">
</form>8"
```

It is very easy to have this code in ASP.Net. It is also possible in C# class that is associated with web form to put code that will create URL like URL that will be created from this HTML and then use Response.Redirect(string url); but information is sent with get method and hackers could use this to set price to 1$ or even less for order that should cost 100$ or even more.

Like any other control I placed it on web form and then for Click event I created method in which I put code.

Here is how URL looks like when I use PayPal control:

```
https://www.paypal.com/row/cgi-bin/webscr?cmd=_flow&SESSION=JinoYHnD7c1AOvU MSyxWeKUwCceg6RwbW038RjJgJg
```

---

8 Website Payment Standard Integration Guide, page 80
The name of web control is BuyNowButton and in code below you can see how I implemented this control.

```csharp
BuyNowButton.Amount = Convert.ToDouble(totalAmount.Trim(new char[] {'$'}));
BuyNowButton.BusinessEmail = "aalagic@yahoo.com";
BuyNowButton.ItemName = orderID.ToString();
```

From previous page in Session object I stored object “OrderType” that has two properties: GUID and TotalAmount. First is used for identification and to create order in database while the second is used to store total price of order and to set Amount property.

TotalAmount is price and therefore it has $ sign as first character that must be removed and it is removed with help of Trim() method and then string is converted to double and Amount property is set to that value.

BusinessEmail property is set to my email address that I used to register PayPal account.

ItemName property is used to give name to item – order and I decided to have customer orderID as item – order name.

In BuyNowButton properties in Visual Studio I set CurrencyCode property to USD. It is possible to set it to EUR, CAD, JPY and GBP.

It is also possible to set other properties from code as well as from “Properties” in Visual Studio. It is possible to set return URL where customer will be redirected after purchase so that customer goes back to our web shop and it is important to get customer back where s/he started her/his shopping. It is also possible to set cancel URL where customer will be redirected if customer cancelled order.

When customer completes order e-mail will be sent to web shop’s email and administrator will be notified.
7. Conclusion

After more than three months of work I have created a fully functional E-Commerce solution which can be used to sell books and make payments via PayPal system.

I have greatly broadened my knowledge of ASP.Net 2.0, C#, ADO.Net, SQL Server and PayPal which was my main motivation for completing this project.

It would be nice to add new features to this web application in order to make it more user friendly. Amazon is always changing features with small improvements and if this is the case with Amazon that is developed and maintained by many users for many years it is obvious that BookZone web shop can also be improved. In addition, the underlying design can be improved, product recommendation added, newsletter, shopping with different credit cards, on sale products, improve user experience with more responsive UI that will use AJAX…

As this is my first serious project in ASP.Net, which I completed alone I have to say that I did learn a lot and that I am satisfied with the results and hope that you’ll like my web shop too.
References

Books:

This book is explaining how to work with ASP.Net and is good first book
JSP and PHP developers.

Interesting book with 12 ASP.Net projects in VB.Net

This book is explaining how to work with C#/VB.Net, ADO.Net 2.0 and databases.

Good as first C# book for Java programmers.

How to start with SQL Server.

This book is explaining what is PayPal and how to integrate E-Commerce web
applications with it.

It is explaining how to integrate web shop with PayPal system.

Advanced tutorial about PayPal integration into E-Commerce solutions

WWW:

[9] [www.codeproject.com](http://www.codeproject.com) visited this site to see solutions to problems that I encountered

[10] [www.asp.net](http://www.asp.net) another good web site with useful informations about ASP.Net
programming