

Design and Implementation of a Secondhand E-bookshop

**Talla Ngala Yiga,
nta04001@student.mdh.se**

Akademin för innovation, design och teknik(IDT)

Mälardalens högskola

Supervisor: Radu Dobrin

3th February 2011

Västerås

Preface

I cannot believe that I am really writing the preface of my thesis to get a degree in computer science .It has been a rough road full of motivation to work with something that I love and passionate about .Before I decided to go back to the university I hesitated for a while because I thought of all the process I had to go through comprising of learning the Swedish language to meet the requirements to study in a Swedish university and taking a couple of math's courses to meet the requirement to be eligible be taken into the computer science department in any university.

All these will not have been possible if I did not have my beautiful and loving wife Cecilia Talla beside me all the time. She motivated and supported my decisions, stood beside me through this program and I will especially thank her for her critical analysis when I ask her opinion on the projects I was doing .Special thanks to my loving daughter Marilyn Talla who always came to me when I was working at home and reminded me of my role as a father and family man that I had to play with her when I am done with what I was doing.

I will like to thank my supervisor Radu Dobrin for the support and guidance as I wrote this thesis.

I will not forget to thank all those who tested the application related to this thesis and gave me their critical points of views on the application.

Abstract

Students have many course based literature text books that they have stopped reading and these books are laying on their book shelves unused. These books can be sold and the proceeds can be used to buy another book the student currently needs. These website will not only serve students but can also serve the entire population or anybody who wants to buy second hand books or wants to empty their book shelves and make financial gain from it. These books can just be simply be uploaded to the website and another student or person can have access to buy the book by just visiting the website. Students also have lesser incomes compared to the rest of the working population. Hence any reasonable resource for getting income is always welcomed.

All these will be achieved by creating a user friendly complete online shopping system for buying and selling secondhand books

Table of Contents

Preface.....	2
Abstract	3
Table of Contents	4
Introduction	7
Background	7
Purpose	7
Goals	7
Current Websites providing services on second hand books	8
Bokexpressen.se	8
Bokborsen.se	8
Kurslitt.se	8
Tradera.com	8
Bokfynd.nu	9
Blocket.se	9
Ebay.com	9
How the Visitor is perceived to use the website	9
Theoretical background	10
E-commerce:	10
The choice of framework for implementation	10
Web forms or MVC framework	11
The .Net Framework	11
Components of the .Net Framework	13
Important BCL classes that will be used in the implementation of this project	15
Models for design and implementation of software:	15
The waterfall model for software development	16
Requirements/Analyses	18
Specifications	18
Design	19
The site design	19
Architecture of the Application	20

The Shopping Module	20
The membership module	24
The search module	24
Admin facilities	25
Sales history	25
Feedback from the buyer	25
Implementation.....	26
The website design.....	26
The Master Page.....	27
Membership.....	27
Uploading books.....	28
Feedback from the buyer to the seller	28
The user's management page.	28
The shopping cart	28
The Checkout process.....	29
Searching.....	30
Site Management	31
Site Navigation	31
Configurations and error handling on pages	32
Testing and validation	33
Test cases	33
Browser check	33
Input source test.....	33
Functionality test	33
General view test.....	34
Results of test and recommendations.....	34
Results of browser check	34
Preventive measures taken to manage input data from various sources	34
Results of Input source test.....	35
Results of functionality test	35
Results of general view test	36
Conclusion and future work.....	37

Conclusion.....	37
Future work.....	37
References	39
Figure References.....	39

Table of Figures

Figure 1: The .Net Current Stack.....	12
Figure 2 : Overview of the Common Language Runtime Infrastructure	14
Figure 3: Communication between the BCL, CIL and Programming tools like Visual Studio.....	15
Figure 4: The waterfall Model for software development with feedback loops.....	17
Figure 5: Secondhand E-bookshop Mock-up.....	19
Figure 6: Two tier distributed architecture.....	20
Figure 7: Class diagrams to handle the shopping process.....	21
Figure 8: All database tables and their relations.....	22
Figure 9 : Payment handling via PayPal.....	23
Figure 10 : ImageHandling object to handle uploading of book images.....	24
Figure 11 : Final view of the website.....	26
Figure 12: Linking the profile object to the shopping cart	29

Introduction

Background

The internet plays a very important role in today's society. Nearly all windows based applications have a web based equivalent. For example Microsoft word has an online version of the same application call Office Web Apps and Adobe Photoshop has its own online version called Photoshop express .The list is very long and it shows one trend that is happening this century on software. The internet is becoming more and more important in software development especially as its makes it possible for millions all over the world to access and use that application directly. The internet has produced and enhanced the growth of some popular brands like Google, Yahoo, Twitter and Facebook.

Sometimes people want services directly without necessarily going to a shopping malls or stores. They will want to have the same service by just accessing the internet and making their purchases. This reduces transportation cost and time can be spent instead on other important activities and its environmental friendly .The is an increase in online usage and consequently an increase in online e – commerce. There is available technology to build such sites and provide secure and reliable services to customers.

In this case, the online secondhand book shop is just one of them trying to use the power of internet and technology to develop a website that that can make book recycling and sales easier. Students need many different types of literature in their studies. As they go through the educational process they move gradually to the next level and other students move to that level they were previously in .It can be described as a circular pattern and text books used in this cycle can easily be accessible to another student in that cycle using the internet as the main meeting place.

They are not only students that need books. Everybody need books or has a book or wants to buy a book. Hence a website of this nature that focuses on books can come a long way to make accessibility to secondhand books just a click away at a good and reasonable price

Purpose

The purpose of this project is to create a communication and user- friendly web portal for selling and buying secondhand books online.

Goals

The major goal for e-second book shop is that in future people, especially students should have an easy way to buy and sell secondhand books online. The users will be able to carry out all these in a secured and managed environment.

The secondhand-eBook shop shall be easy to navigate and user friendly and easy to find books in various categories.

It shall be easy to administer and handling of orders and sales processes shall be easy to handle by the administrator

Will the design, implementation and testing of this website achieves the following goals?

- Provide a way to buy cheaper books online
- Provide a better way to recycle text books.
- Users will be able to make money for themselves by selling secondhand-books.
- A User friendly secondhand e-bookshop

Current Websites providing services on second hand books

They are many websites out there on the web that provide services on second hand books .My goal is to provide a better, faster and user friendly website with services that other websites might be lacking or not well developed. The secondhand e-book website will be only focus on used books. Some available websites and their descriptions can be seen below.

Bokexpressen.se

- Sells secondhand books .It has no direct web shop and a payment service .The secondhand bookshop will provide this. The is a buyer to seller contact here .On the secondhand book shop there will be no direct contact between the seller and the buyer until a sale process has gone through. Hence assuring the buyer that if they pay for a book they are sure to get it else the Manager of the website can refund them directly.
- Has no online store
- Membership required to load books

Bokborsen.se

- Almost the same concept like the current project but sells music CDs, videos and films. Has a web based shop and membership .They have no single based focus compared to the secondhand bookshop that will focus on book sales and hence provide better information and details on books.
- Has an online store
- Registration required

Kurslitt.se

- Sells secondhand books. It falls under the category of bokexpressen.se. The analyses done there hold the same for kurslitt.se.
- No online store but membership required to load books to sale

Tradera.com

- Sales everything that is second hand including books .Has an e-web shop with membership. This is an auction site. They sale everything through auction .For students who want to buy books it will be preferable to go to the second hand book shop and buy the book you want and just pay for it than to be waiting for an auction to complete on an book and following the process the whole time. The time can be used to do something useful. Hence this current project provides a

faster, secured and guarantees service especially for students who want textbooks and have the money to buy them directly.

- Has an online store
- Membership is required

Bokfynd.nu

- More of a price comparing service for books. Hence there is nothing to compare with the current project apart from that it has something to do with getting books at an affordable price .It can help the student to get cheap books .This kind of service can in future be incorporated to the secondhand bookshop website.
- No online store and membership is required

Blocket.se

- Popular portal to buy second items including books .They sale everything. Their focus is not books. It just a contact portal for a buyer and the seller. The parties involved make their terms of sales and purchase
- No online store and membership is required except billing information

Ebay.com

- Sell secondhand items including books. It's a world renowned website auction portal .The comments on trader.com hold the same here.
- Has an online store and membership management.

How the Visitor is perceived to use the website

For the website to fulfill its functionality three users are required for a successful purchase process to take place.

1. John (user one) creates an account because he wants to buy a book.
2. Mary(user two) creates an account because she wants to sell a book
3. Mary uploads a books
4. The book appears at the websites home page directly based on its category.
5. John fills his profile so that he can easily buy books.
6. John has seen the book he wanted to buy. The book was uploaded by Mary (user two). He clicks on the book and goes through the checkout process. He buys the book.
7. Mary gets a mail that her book has been bought. She sends the book to John's address.
8. John tells the administrator of the website that he has gotten the book. The administrator credits Mary's account.
9. The Administrator (power user three) get a small percentage of the amount John paid
10. John can give a feedback about the book he bought from Mary.

Theoretical background

E-commerce:

Mathew Reynolds (2000) writes that e-commerce is the name given to any business process of selling goods, products and services over the World Wide Web .It gives an opportunity for the owner of a store to put out the company's catalog online. He further writes that any business can develop its own e-commerce strategy. The business model for the online store and offline store has to be different but to be successful a good business model for e-commerce has to put in place.

Mark Norris and Steve West (2001) write about different type of catalogs an e-store can have. From the low-end catalogues, high end catalogues and low end commercial catalogues .They elaborate on some of the problems building a low-end catalogues will have such as its impractical to build a shop that contains a large number of products because pages that describe this products have to be constructed entirely by hand . There is complexity in changes of products and prices, integrate a payment system, taxes, and stock control.

Donnelly (2001) writes that if a site expects the web user to find a product, buy it and provide his or her credit card details the site must be user goal oriented and concentrate on the tasks that the user would need to accomplish in order to achieve their goals. The major goal should be to the site more convenient than other the conventional way of doing shopping.

Bandyo (2002) writes that the user interface is a very important factor when creating e-commerce websites. He recommends that during development the user should be part of the design process. It should be a user centered design and it should be iterative making sure that feedback from users is used. He further writes that legal considerations for the site and logistics aspects for shipping have to be taken into consideration.

Marco (2006) concludes that implementing an e-commerce is a very big challenge for any site developer. He further writes that it requires good design and good knowledge local laws about licenses, tax collection and other administrative requirements. All these require a considerable amount of time. It's not only having a catalog but it also marketing your website and attracting users to come back and purchase items from the site .This is very important for the website to succeed. He even says that it is sometimes cheaper to buy a complete product than to implement one from scratch. Small e commerce websites can use another service for payment that is not integrated to the site but as the website gets bigger integrating a payment service become very important. It plays a part in the customer confidence when purchasing.

The choice of framework for implementation

This application is going to be implemented using the .Net framework. It has all the available tools ready to use for the implementation of such a website. It is robust, secure and there is a very large community online and books to get help from if difficulties are encountered during development. The tools are provided free by Microsoft to students wishing to develop applications in .Net. I am very comfortable

working with .Net since I have done a couple of courses that I had to work with .Net and use Visual Studio as the main integrated development environment (IDE).

Web forms or MVC framework

There are two ways to build web applications with .Net .There first way is by using web forms and the second way is using the MVC(Model View Controller) framework . I choose ASP.net form because I am comfortable with it and if I had to write this application in MVC it would have taken a longer period since I had to first learn the how to write web applications using this style and start implementation later. However I pay the price of not making unit testing of the application easy and the application uniform resource locator (URL) that are not so SEO (Search engine optimization) friendly for not using MVC but I win in terms of time and effectiveness to implement the application using ASP.net forms.

The Applications build with .Net are also portable and can function properly on any browser. If there is a need to adapt the application for a particular browser it can easily be done.

The major drawback in using .Net is that hosting services are more expensive due to license cost to be paid for its servers and not readily available compared to a website implemented in a PHP, CGI etc... based framework reasons being that only Microsoft products can host asp.net websites. Other server scripting languages can both be hosted by both Linux and Microsoft servers.

The .Net Framework

Microsoft released the .Net framework in 2002 with a promised to have a system that will solve the old problems it had with its software development tools and will be able to meet the needs of the next generation in software development. Solis (2008) writes that the .Net framework is much more consistent and is object oriented compared to the older MFC (Microsoft Foundation Classes) and COM (Component Object Model). He further writes that the system runs on a broad range of electronic devices such as PDAs, cell phones, computers and servers. It uses and understand industry standards communication protocol as such as HTTP,SOAP,WSDL and XML .It is secure and provides an execution environment that is save.

Hanspeter, Wolfgang, Dietrich, and Albrecht (2004) confirm this by writing that the .Net framework is a layer running on the windows operating system and add two main aspects.

1. A runtime environment that has an automated garbage collection system, proved security, versioning and interoperability between programs that are written in
2. Different languages.
3. Finally it offers an object oriented class Library for building applications.

Since then it has been evolving and today there is the 4.0 version available with more potential and services. Here is a view of its evolution and features added.



Figure 1: The .Net Current Stack

Figure 1 shows the growth and evolution of the .Net framework. The layers Common language Runtime, Base class library and the layer on top of it, Windows forms, Asp.net and ADO.net are the major and most important part of the stack.

- Win Forms and Asp.net can be described as user interfaces .Win Forms for desktop applications and ASP.net for dynamic web applications.
- ADO.net is used for database services and connections and provides stable and secure interfaces to access databases in .Net.
- The Base class library (BCL) is a large class library for .Net.
- The Common language runtime (CLR) is the *Just in time* compiler for .Net. Both BCL and CLR will be described more on the subsequent chapters.
- Web services are also provided by .Net. They are XML based data communication medium.

.NET Framework 3.0: It consists of three important components.

- Windows Presentation Foundation (WPF) .It's a user interface and API (application programming interface) that is based on XML and vector graphic.
- Windows Communication Foundation (WCF): A service that makes remodeling easier and allow programs to interoperate locally and remotely
- Windows Workflow Foundation (WWF): Used for building task automation and use of workflows in integrated transactions.
- Windows CardSpace: A software components for managing and securing digital

.NET Framework 3.5:

- The main feature of this version is the addition of Language Integrated Query (LINQ) and ADO.Net entity framework for database services and easy integration.

.NET Framework 4.0:

- Improved to better support parallel programming .This is to target distributed and multi core systems .To this is the availability of parallel Linq (PLINQ) and the Task parallel Library.

The main tool made for optimal .Net productivity is the different visual studio versions. It is the programming tool(IDE) for software development in .Net and the latest version is visual studio 10.They come in different versions .The free versions are the express versions. With visual studio and other development tools from Microsoft productivity ,debugging ,publishing, hosting and testing for software development in .Net becomes faster and easier.

Components of the .Net Framework

This framework is made up of three main components namely the Common language runtime (CLR), programming tools (Visual Studio) and a Base Library class (BCL).

The CLR

It manages all programs at runtime. It manages the memory, does garbage collection and code safety and its execution. It is similar to the Java *virtual machine* with its own instructions set. All programs written in .NET are translated to the Common Intermediate Language (CIL). Before they run they are compile just in time to the target code of the machine .The machine can then execute these instructions. The CIL guarantees for interoperability between languages and code portability. The *JIT* concept provides efficiency in code execution. Another problem comes when it concerns data types Microsoft.net solve this by creating a Common type System (CTS) that ensures and describes how classes, primitive types, and interfaces are represented. At the moment they are more than 20 languages that run on the CLR (Hanspeter, Wolfgang, Dietrich, and Albrecht[page 3]) .They all follow another important aspect and definition of the .net framework called the Common Language Specification (CLS).

An overview of the whole process can be seen below .It starts with a source code of any of the .Net supported languages like C#, VB.NET etc... .It is compiled into the Common Intermediate Language (CIL) specific to the .net platform and the Common Language Runtime (CLR) compiles the CIL to machine code just in time for execution on that current platform.

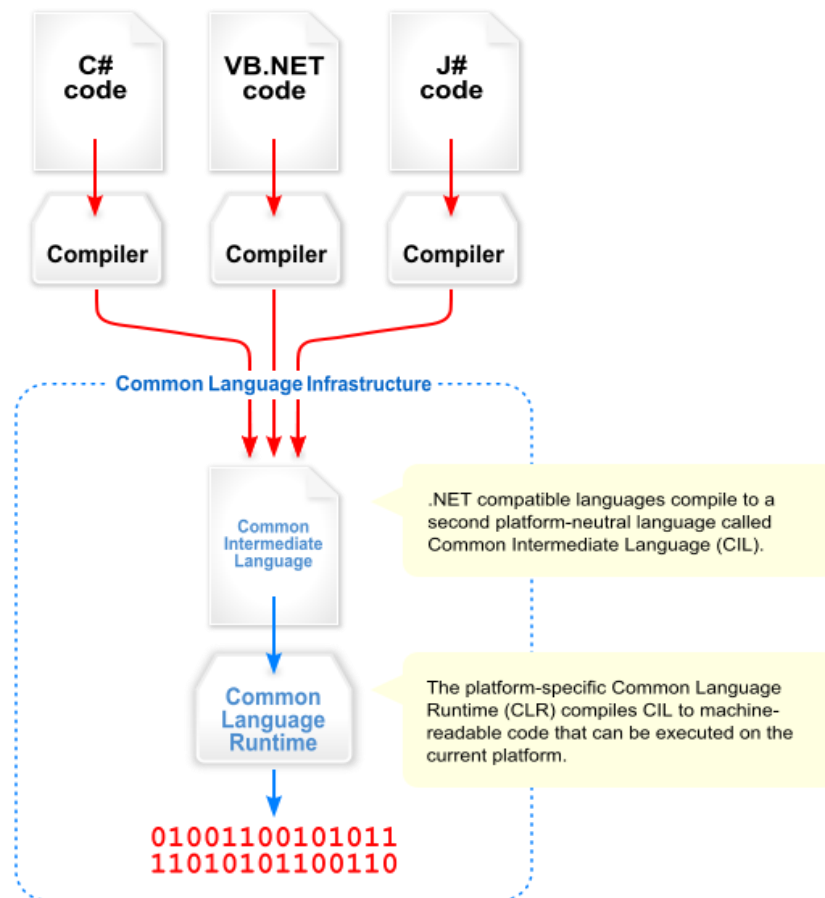


Figure 2 : Overview of the Common Language Runtime Infrastructure

The programming tools

.Net provides tools for software development in .Net. These tools help in making coding easier and debugging easier. The main tool for development (IDE) is the Visual Studio integrated development environment .Net provides compliant compiler in languages like(C#, C++, Python, VB, JScript , etc..).It provides debuggers for these languages as well. It provides tools for server side development like ASP.net and makes it easy to build web and windows based applications easily.

The Base Class Library

It is a very large class library in .Net available for use in application development. These are available for use in any program development in .Net .The base class Library is very big and some of the classes will be key in building and implementing this application .Here is an overview of how these three key aspects functions. Programming tools use classes provided by the BCL .The programming tools use the CLR to execute the program. The CLR in turn uses the BCL.

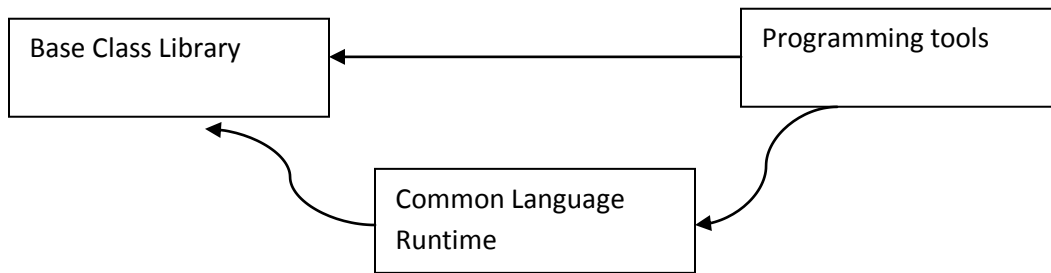


Figure 3: Communication between the BCL, CLR and Programming tools like Visual Studio

Important BCL classes that will be used in the implementation of this project

Two important namespaces that will be used to implement this application will be the System. Web and System.Data. System.Web allows for the creation of dynamic web pages under ASP.net and the latter is used accessing different types of databases and other data sources such as XML under ADO.net. These two are not only the classes that will be used but they are the back bone of the whole application. Data will stored in database using ADO.net (ActiveX Data Objects) classes and Dynamic pages will be created using server-side ASP.net. (Active Server Pages)

Models for design and implementation of software:

Considering the fact that this project involves design and implementation of a software system regardless that it is web-based, it will be important to mention and consider some models used in software development and deployment.

Somerville (2000.Pn:8) defines a software process as activities that lead to the production of a software product. And a software model is a simplified description of the software process and it is presented in a particular perspective. He further outlines different types of software process model. Some of the models are

- Workflow model representing sequence of activities
 - The data-flow model or activity model represents process as a set of activities.
 - The role/action model represents roles of people involve in the system.
- He further identifies some general models of software development. There are namely
- The Waterfall approach: It represents activities in requirements, specifications, design, implementation and testing. All these as separate processes
 - Evolutionary development: It involves a rapid development of the specifications and then refined later for the customer.
 - Formal transformation: This approach is based on setting and producing a mathematical specification for the system to meet.
 - System assembly from reusable components: This approach assumes that part of the system already exists. This model is focused on integration.

After reading though all these models. The Waterfall model fits the development of this website. It's had very clear iterations and considering the fact that I am working alone I can

focus on each part of the model during development and come back to it if need be .The project can easily de broken down into different parts based on this model.

Standish (2000) writes on the *Code and Fix Model* was used in early days of software development but because of its many flaws like now always matching the customer needs. This lead to the development of a waterfall model that clearly stated out what the system required and what the customer wanted and how the development and implementation will take place .All documentation was part of this process and waterfall model lead to easy legal binding if needed.

The waterfall model for software development

This is the model that the will be used develop the E-secondhand bookshop .However feedback loops will be allowed during the whole software development process .The model chosen for this project has to favors a single developer for a project. Considering the fact that I am the only one who is going to implement this project I find this model suitable for me to follow.

The waterfall model recommends software development to go sequentially and follow certain rules and stages. It requires that software development follows the following stages

1. analyses of the problem is made and requirements to be meet are proposed
2. The specifications of the project are put in place after the problem has been analyzed and the requirement to be meet are made clear.
3. A design for the software system is made
4. After the design is completed the process of implementation and debugging starts
5. The program is tested and integration of the system is made
6. The developed software is t in its final stage and operation and maintenance can then be made continuously

This process has been illustrated below on Fig 4 showing top-down development. The main stage description are in the boxes The model looks likes a waterfall .For this model to work efficiently feedback loops are added to create a coordination of the whole process development.

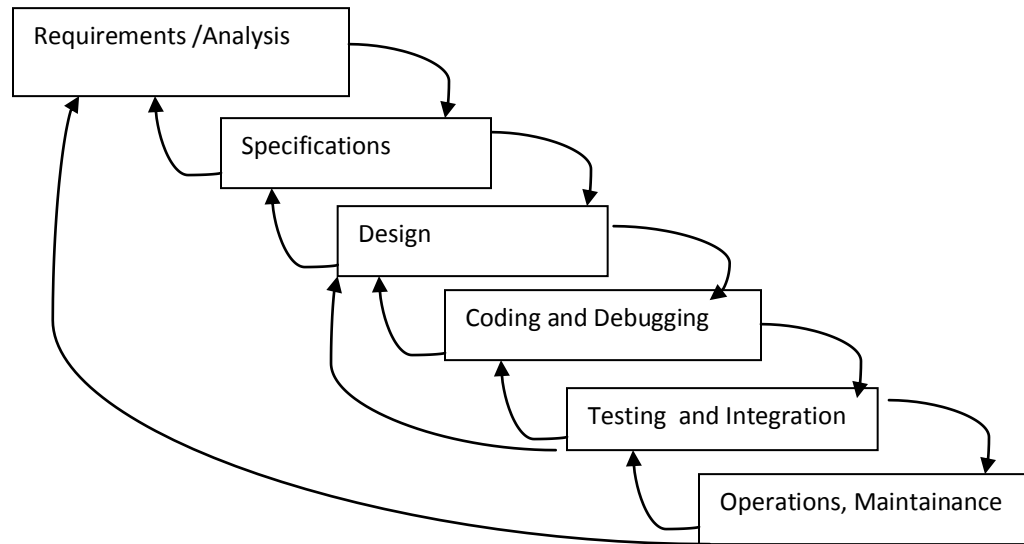


Figure 4: The waterfall Model for software development with feedback loops.

After adding feedback loops its clear how the process are inter linked and depend on each other. Each level can get and check back on the previous stage of development to sort out a problem. The waterfall model with feedback loops favors the software development in a more realistic way. It leads to a greater probability to meet the requirements of the customer.

On the other hand this model also has its flaws .Its time consuming because each stage has to be clearly defined before the next stage starts. There is also a waste of man power especially when testing is going on the application.

Requirements/Analyses

The website should be able to load and function properly on all major web browsers .Easy functionality for the user and the administrator. Easy user interface and fast data uploads and reloads into the website .The process should be secure especially when it comes in handling financial transactions money .A foreign provider can be used .This process has already been carried out and the customer and the developer can now write down the list of specifications and goals the website will achieve. The specifications should be able to fulfill the purpose and goals of this project.

Specifications

- Complete online shopping process should be developed
- Anybody should be able to browse books on sale
- Anonymous and registered user can buy books.
- The website should allows visitors to create ,manage and log into their accounts
- Members can sell books they uploaded. They can delete a books if there s no current buy process going on that book.
- Members can Create and update their profiles
- Comments on buyers and seller books after a sale process.
- History of transactions of a member
- Administrator has a simple order management system
- Capability of handling financial transactions(Use another Provider API e.g. PayPal)
- Mailing capabilities after selling or buying process
- Search for books on the website
- A visitor should rapidly find a book he or she is looking for.
- Latest books are shown on the main page
- Handle errors in a proper way.

Design

The site design

To begin developing a website it is very important to have a mockup of the site .This will be done using Paint or Adobe image maker or any available online graphic tool. The mock-up image gives an overall picture of the look of website portraying its functionality, key parts and key functions.

Figure of site mock up:

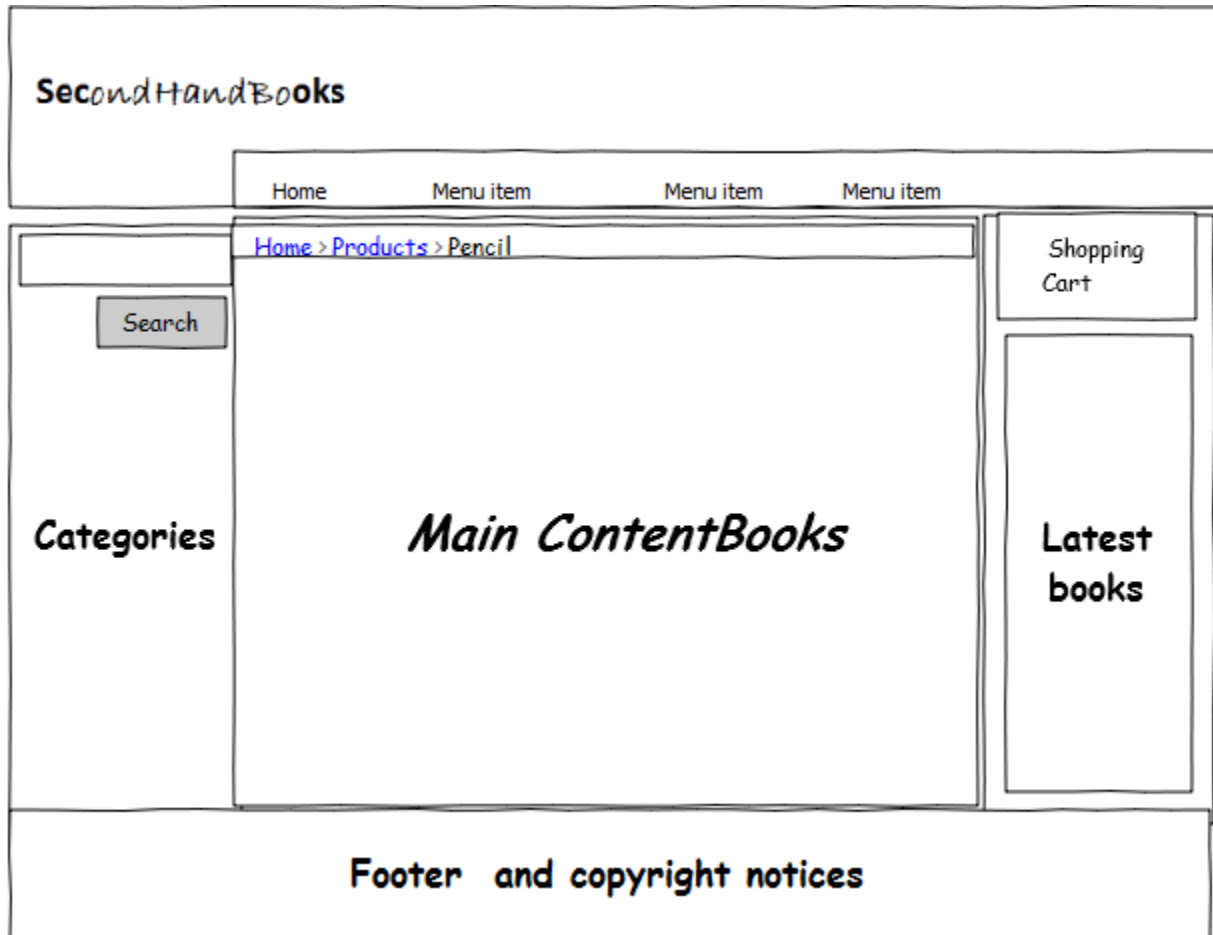


Figure 5: Secondhand E-bookshop Mock-up

After the mockup is accepted and considered optimal and user friendly the layout and design with HTML and CSS will be made based on this. Graphics will be added to add aesthetics on the website.

Architecture of the Application

The next phase of design will be planning the architecture of the application. There will be in some cases a separation of the user interface and data. The business layer will be totally independent and not embedded in the views of the website. Data will be stored in an SQL database and LINQ (language integrated query) to SQL classes will be used in some cases to query the database and code behind for server side scripting will be in C#. The other business objects will have their own layer. Views will all be from ASP.net view files. The architecture of the application is shown below in figure 7:

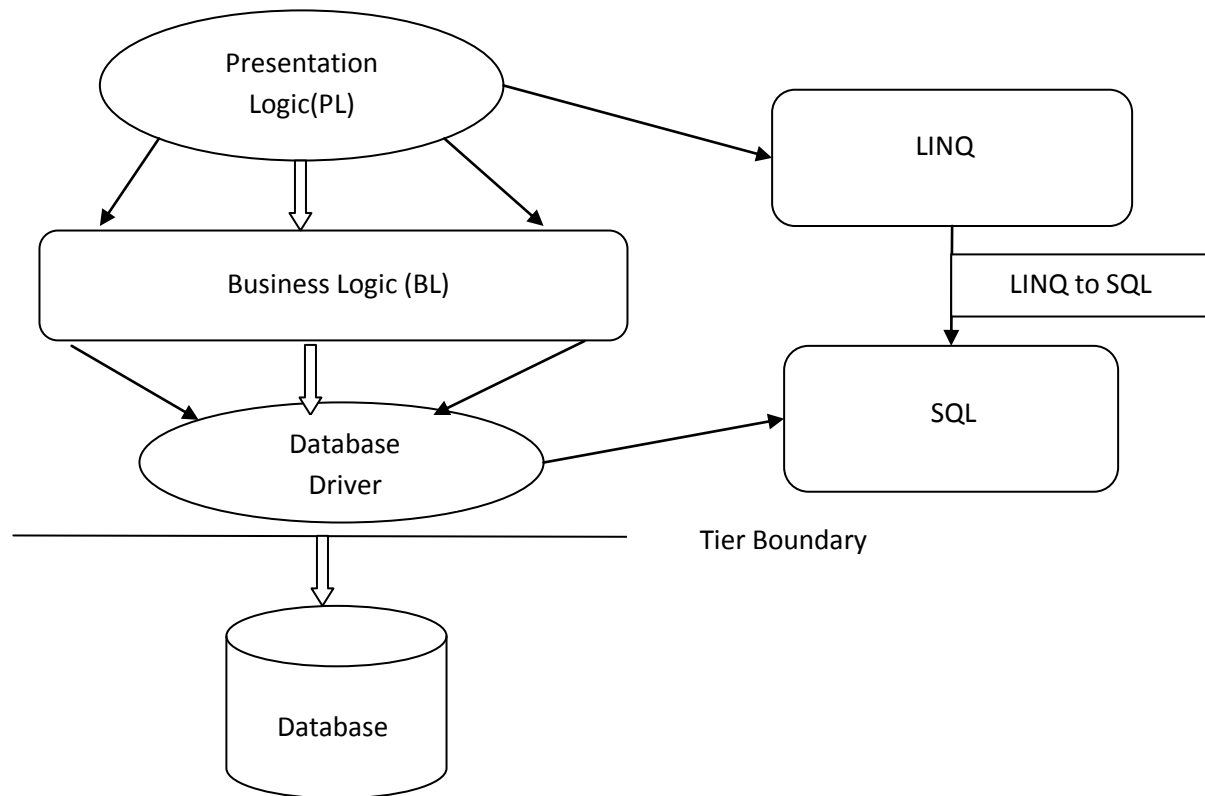


Figure 6: Two tier distributed architecture

The Shopping Module

Specifications

1. Shopping cart that is visible at all times and maintains its state
2. Delete, update from shopping cart
3. A check out process that is smooth
4. A page to show the items shopped made by the user showing shipping and location details
5. Payment service that people can trust to leave their credit card numbers.
6. You do not have to member to buy books. Hence anonymous user can buy books

Design of the shopping module

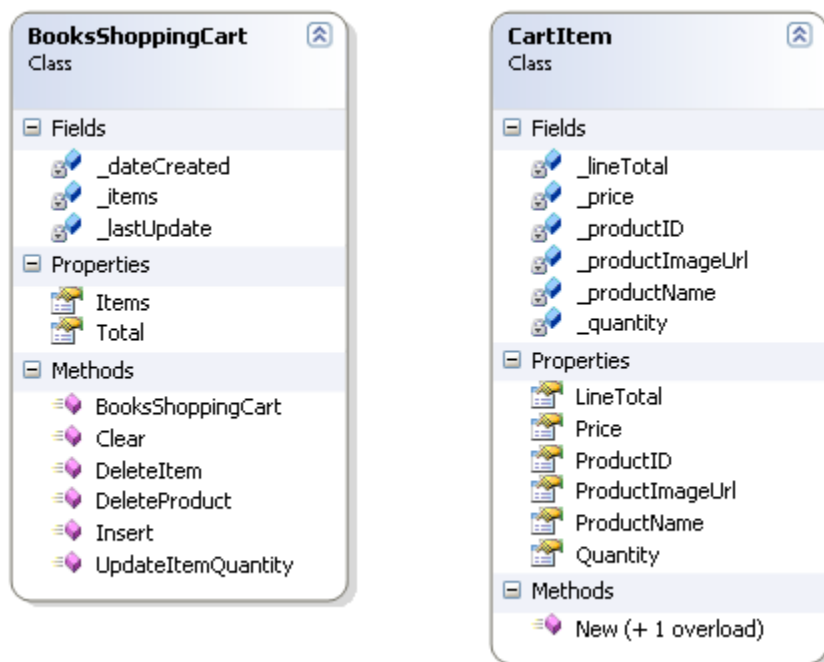


Figure 7: Class diagrams to handle the shopping process

There are two classes to handle the actual shopping process and the items to be bought. A book is represented by a `CartItem` class. It has all the properties a book object needs for information storage. The `BooksShoppingCart` class manages the books (`CartItem`) in the cart in this case the books added into the shopping cart. It has all the methods to handle shopping basket. This has been represented below

To store information about books and manage book orders I choose the MSSQL database. It is secured, easy to use and well integrated into the .Net framework. Using the ADO.NET and Linq utilities queries can easily be made on the database. Five tables will be created to handle this module. Each table will store relevant data that will be needed to fulfill the objectives of the website. These tables are the `Product`, `ProductCategory`, `Order`, `Orderlines` and `Comment_tb` tables. The contents and relationships between them can be seen below on figure 8

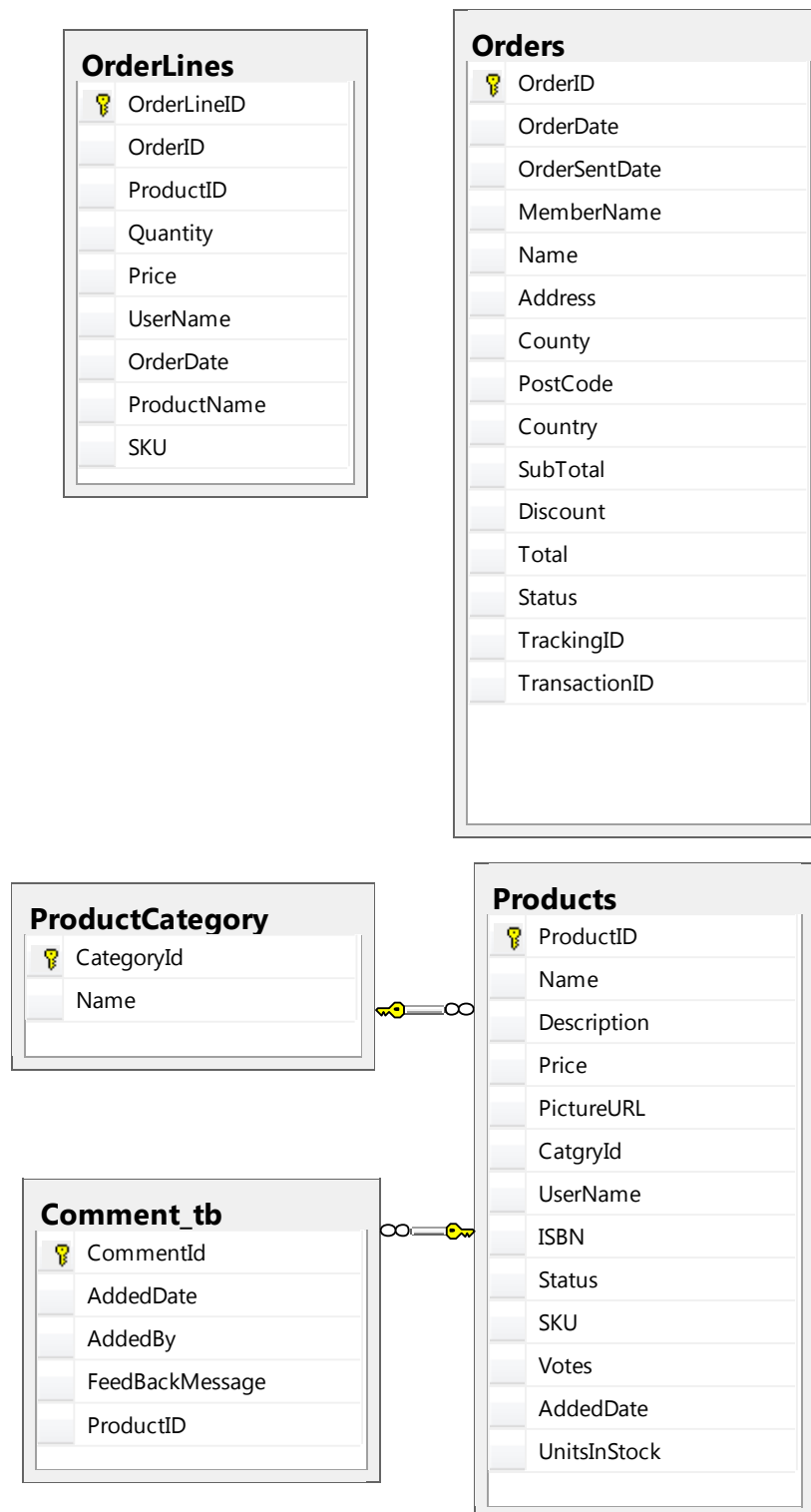


Figure 8: All database tables and their relations

- The Product table that stores all information about books. The product table has the CategoryId as the foreign key because each book should have one category in this case. It is a one to many relationship
- The Category table that holds the different categories of book.
- The Order table stores all orders and buyers address and finally
- The OrderLines table that represents the details of book that has been bought. It has a reference to the order table. It has a foreign key OrderId so that each book can be linked to a specific order that was made.
- The Comments table stores feedback comments from the buyer. It has the ProductID as a foreign key because comments can be written about a product. There is also a one-to-many relationship between the product table and the comment table.

Payment module

Considering the fact that the website is small for now and to make it easy for people to trust the website by giving out their credit card numbers and actually buy books on this website I will outsource financial handling to a trusted and well known online payment service. In the case of this project I choose PayPal. It is reliable, secured, found in more than 100 countries and it's easy to attach to your side if the right procedures are followed. It's going to be the payment gateway for this website. Here is how the payment gateway can be viewed.

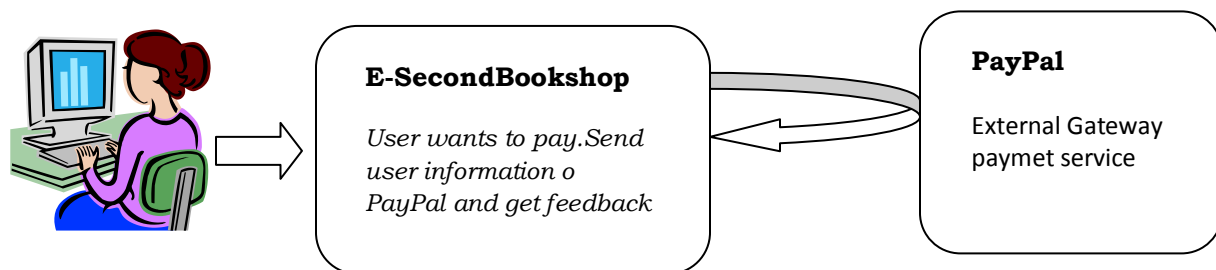


Figure 9 : Payment handling via PayPal

Here are the steps of how the financial handling will take place. The process begins after the buyer has successfully gone through the checkout process and the manager of the website has been notified that the website's account on PayPal has been credited. It's the final process of purchase process.

1. Buyer pays to website account
2. Buyer gets feedback about the process
3. Manager gets informed that account is credited
4. Manager informs the seller by mail to send items to the buyer.
5. The seller sends items to the buyer.
6. The buyer informs the Manager that he has the items.
7. The manager credits the user's account based on information from the seller.
8. The buyer can all give feedback comments that others can read.

The membership module

Specification

1. Log in and out
2. Users can become members of the website when they want.
3. User should have a profile
4. It should be secured
5. Users can change and create new pass words
6. Members can upload books and delete their own books they uploaded
7. Members know the state of their books uploaded

When building web applications with .Net beginning from ASP.net 2.0 Microsoft has provided very handy API(Application Programming Interface) for web development to ease work for development. One important module is the membership API in .Net .It handles membership and is secured and embedded in the framework. Membership includes security, logging in views, profile creation and many others that I will subsequently explain. This module will be used to handle membership in this web application. The profile object of a user will be directly linked to his or her shopping cart.

Design

Since I am using an API that is part of the Framework there is no design to made here. Some important controls (user-interface objects provided by Visual Studio to use for development) used in dev eloping this section will be the log in and registration controls .The .Net Profile object will also used. The profile object will be linked to the users shopping cart. Another option is to use the session variable. I will prefer to use the profile object because more aspects and objects can easily be linked to it and it solves a lot of problems by easing the creation of a user profile on the website.

Book Uploads

It will use the ASP.Net File Upload control. It is already ready for use and has just to be programmed against .For thumbnails of images the an image handling object was created to handle images of books so that they produce thumbnails that can be shown on the various ASP.net controls when need be .This class will take any uploaded image and make a thumbnail from it.

Figure 10 : ImageHandling object to handle uploading of book images

The search module

Search for books using their ISBN numbers or book title. Enter title or ISBN number in the Textbox and a feedback is sent back to the current user on the status of the search after the button search is clicked.

On click on the book results shows you a detail description of the book in another page. The shopping process can directly continue.

Admin facilities

Order management module

Orders are stored in a database. The Administrator of their website can delete and refuse access to any user. He sees all the orders of the website and is in charge of managing the website. He makes sure that the seller and buyer fulfill their part of service before crediting the sellers account.

The administrator will have all the rights to create, delete and update the databases.

The admin will have an interface to the product table and will have another interface to manage orders and see the status of each order.

Mailing

The admin manages the mailing part of the website. He will be able login into the membership table and manage members and also send mails to them updating them on different types of information. The Administrative module for ASP.net will be used in this case. .Net makes it easy to send emails. They have mail object in the System.net base classes to do this. This will be used in this application to send mails.

Sales history

Information of each user is stored after an order is made .So to get the history of the past orders. The order and product database is queried to get the details of the past order and sorted by the last order they made.

Feedback from the buyer

After a sale process the parties involved in the process can give feedback to each other .This feedback that will be public on the website. To design this module the following has to be taken into consideration. The buyer should have an interface to write feedback to the seller. The buyer can write a feedback only once. If a book has a feedback the message interface should be invisible.

Implementation

After the design was made and the problems arising from the design process were clarified and dealt with, it was time to start implementing the application. Implementing application of this scale requires lots of resources and explaining the whole implantation process will not be clarified in this paper. However major important aspects in the implementation will be described.

The website design.

The design was made directly from the accepted mockup (fig 5). The graphics are done in using an add on in Firefox web browser called Pencil Sketching and every other implementation like HTML and CSS tagging was done using Visual Studio 2008. There are two style sheets. One for the main website and one style sheet for the users private pages. The CSS style sheets are all linked to the master pages hence there is less or no presence of inline styling.

The ASP.net Master Page control is used to create constant view of the website key views presentations. Since every page will be attached to the Master Page at runtime the website has kept a constant look and feel. It serves like a template for the website design and views. After the design here is how the final page looked like.

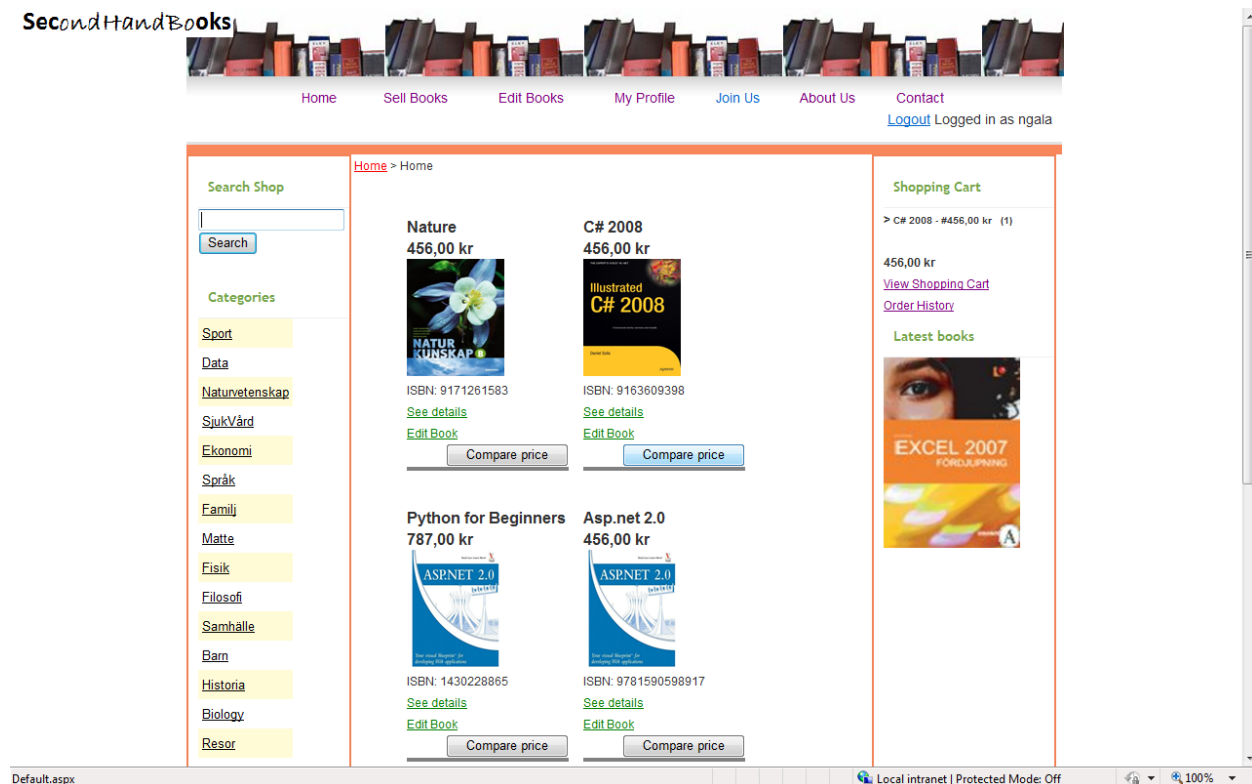


Figure 11 : Final view of the website

The Master Page

The Master page is made up of the header, menu, left column, right column, and footer. It's the main template of the website. At runtime it combines with another page that becomes like a child to it. Both pages merge together and the contents of the current page are shown in the contentPlaceholder control that was placed on the Master Page.

The middle column is reserved to show information for pages on the website.

In the page above the Default.aspx page (home page) .It contains a ListView control to view books . It has a template allow for me to add my own controls to it .Hence the book images are part of the template field and are made to thumbnails by the Imagehandling object that was earlier described. The ListView It gets all its information from the database. On the ListView control there are hyperlinks controls to view details of each book on click. The user is sent to another page showing the more information about that book.

There is the menu control under the header .It uses the sitemap provider control that reads an xml file that contains all the links to pages on the website .Since it's an XML file the contents of the menu can easily be organized in a tree format with certain links serving as the parent to other links.

Under the main menu there is a breadcrumb control so that the user can always find their way back to the page of origin and get information of the where they are at their time.

Shopping cart box is a user control. A user control inherits from the .Net Control object. It is an encapsulated self acting control like others and can just be placed anywhere on the website .It shows the current state of the user's shopping cart. It's on the master page since it will be always visible on any page on the website.

The left column contains a search button and textbox control to get input of book titles or ISBN to search for books in the database .The results are will be shown under the button on the same column. The categories are added to a DataList control. On click the selected category will be sent through query string to the database and the ListView on the Default.aspx page gets populated with books based on the selected category

The right column has a latest book label that shows newly uploaded books. It is a usercontrol that has a slide show implemented with jquery .It show book images according to uploaded dates.

There is a button on every book on the website with then name -Compare. When this button is clicked the it gets the current book's ISBN number and attach it as a query string to a chosen online bookstore of my choice. The user can see the cost of the same book if they wanted to buy a new one. The choice of the bookstore that I send the query is made based on its popularity and usage.

Membership

I described above that I was going to use the .Net Membership API that has been part of ASP.net from versions 2.0 to 4.0.It contains and has all the necessary utilities, conditions and variables needed to handle membership of a website. To add to membership to a site it just to add Login controls and CreateWizard control. The latter gives a readymade wizard that allows me to be able to register members on the site. Other controls I used that are linked to the membership API are the Login Status Control, Password recovery and creation controls and the LoginView control For all these to succeed Visual Studio adds an ASPNETDB database that will hold membership information for the site.

To manage members and give them roles I used the ASP web administrator management portal. It was possible to create my own administrator management portal but considering that time scale for this project I did not deem that necessary. As the site grows bigger in membership, that option can be later considered.

Uploading books

To upload books on the website you have to be a registered member. After the process you will be able to see a link on the menu that links you to a page to upload new books. On this page the DetailsView control is used as the main input interface for all relevant information needed to sell a book on the website. When a book is uploaded it is automatically viewable by everybody who visits the website. The buying process can start immediately on that book. I use the Ajax toolkit web editor in the insert template so that the user can adequately format the text written in the description section of the book.

On inserting I store the User's name into the database so that I can know the owner of the book. Below the DetailsView control is a calculator. It can be used by the user to calculate the percentage to be removed from the current book they want to upload on the website.

Feedback from the buyer to the seller

Since the buyer already has a history associated with his orders. I made the purchased items to a hyperlink. On clicking on that hyperlink the current user is sent to another page with the current item's identity as a query string to another page that has a textbox under that particular book to allow the user to write feedback comment. Every book is allowed to have just one feedback hence if a purchased book has a feedback attached to it the message textbox becomes invisible. All these checks happen during the page load event.

The user's management page.

As a registered member of the site you access to a page call myBooks.aspx that you can see the current state of your books, total you sold and other details relevant to the books you sold or bought. It's here that the seller can update, delete or sent a message regarding the books he or she uploaded.

The user has a profile page. It is made up of Textbox control, button control, Label control and the Password recovery Control. On this page the user's profile can be updated, deleted or created. The information is relevant especially when the user wishes to make a purchase or after a sale has been made. The profile page has detailed personal information about the user's address and preferences. On this page the user can also change their password. I also use an online editor gotten from the Ajax web tool kit. This is a full-fledged editor that has all the necessary functionality of an editor. The user can edit the description of books entered to the website.

The shopping cart

The shopping cart module is the most important module of this website. It must be fully functional and stable for the website to function as an e-commerce website. The shopping cart involves a check out process that its implementation will be described later.

They are two UserControls that simulate the shopping cart. The main shopping cart I represented by a GridView control that can be edited and the quantity of items added can be updated and deleted. It also shows the totals of the current items added. The miniature shopping User control that just gives a summary of the main shopping cart .Its is on the MasterPage because it has to be visible everytime to the user. This userControl has a Repeater control to show updates as the user adds items to the current shopping cart.

Each User is linked to one shopping cart Since ASP.net 2.0 when the membership module was made it made it easy to have a profile object that can be linked to one user. In this case I do not need to use a session object and cast it later to List object holding CartItems. The bookShopping Cart and CartItem object are both serialized. This gives out Meta data in the CLR that can be accessed and used by other objects. In this case the profile object takes advantage of this and one can therefore have a profile object linked to the shopping cart as

```
<add name="Cart" serializeAs="Binary"
type="BookShop.Commerce.BooksShoppingCart" allowAnonymous="true"/>
```

Figure 12: Linking the profile object to the shopping cart

This is done in the profile section in the webConfig file of the application. Since the Profile object is the sole identifier to this cart, any updates in the Shopping cart is also updated in the miniature shopping cart view that references the same Profile object properties. The profile object migration methods makes it possible for an anonymous user to buy books on this website .They are configurations to be made for this to be successful .This will be explained later.

The shopping Cart container is a dictionary object in .Net. It's a key –value container object and is perfect for such handling especially when products added to the shopping cart have to identified and referenced and updated .Since its part of the .Net collection library it returns an ICollection.

The Checkout process

This is where the user has finished shopping and wishes to check out and pay for the books they bought. T he bookShoppingCart.aspx page has a button to start checkout when the user wishes. On this page the user can see the books they bought and can make last updates and delete. When the check out button is clicked the process starts.

The check out process can be divided into 5 steps namely

1. Login if the user is not logged in.
2. Get the Delivery Address of the current user
3. Handle Payment
4. Finished
5. Order completed

This steps depend on each other and can be describes as a pipeline process. Hence the best option to use in this case when working with ASP.net is to use the Wizard control. The wizard control is a container control that each step can be added and given a specific reference and the steps the wizard will handle can follow the steps and the implementation requires. Each step in the process is added to the control and all the information and interface in that step can be seen.

When the Check out process start if the user is logged in there is a Checkbox control beside the Delivery address textbox controls to get the current user's address from the his profile .When its filled then the details of the items and books bought are shown on the next step and the user can proceed with payment .

This application uses PayPal as an external payment gateway. This is because PayPal provides a Sandbox API for developers. It actually simulates the complete PayPal system in terms of handling payment. Hence I created an account on the PayPal developer website so that can actually test my application payment functionality using the Sandbox website that can simulate paying to PayPal .It means that the when it comes to handling of payments the user is sent to PayPal website and details of the orders is sent to PayPal via a GET request and all parameters for this order can be passed as a query string .When the process has succeeded ,failed or completed then the user is sent back to the E-secondhand bookshop website. PayPal sends back important information of the payment status that is gotten by checking using the HttpRequest object on the PayPal website feedback. This checks if PayPal send back a VERIFIED string meaning that that payment was successful else payment failed.

Based on the outcome the user is send to a particular page with details of the feedback from the payment process .If it succeeded the user is shown the OrderSucced.aspx page and if the payment process failed the user is shown the OrderFailed.aspx page and can be asked to try again

During check out the user can abandon everything and the current order can be improperly processed .To solve such problems. The SQL connection object has a transaction method that is capable of rolling back the current order it the process of being saved. This makes it easy to know store successful orders and reduces unpredictability of the database storage process.

Searching

When the search button is clicked on the website, the current words in the Text Box control on the Master Page is first parsed and the words are turned to small letters and then and sent to the database using Linq to SQL as the main API for the implementation to actually search the database .The words are matched against the database of books and if there is a match or matches the result is shown just below the search button on a List View control that is visible after a query has been made and the button clicked. The results from the search are then added to as hyperlinks. The books identity is added as a Hyperlinks control query. The user can click on the hyperlinks of books and be taken to another page showing the details of that book .The visitor can then proceed by adding it to the shopping cart or doing something else.

Site Management

To make this section of the website easy to handle and secure all web pages related to the overall managing and maintenance of the website products are all found in a separate folder. This makes it easy to block and restrict access to these pages. When the manager logs in, he or she can have access to those pages.

The manager of the site has all the rights and access to all data on the website. The manager uses the Administrator unit of ASP.NET to monitor and configure the website. He can give and decide what role each user will have. The manager can also see the current status of those logged in.

The folder Admin on the website holds relevant pages that are only accessible to the administrator of the site. Hence these pages on the website can never be accessed if you have a different role.

One of the pages in this folder all the interfaces has CRUD (Create, update, delete) and even sorting facilities. The GridView control is very suitable for that. With the GridView it is easy to add all these facilities. The Manager sees all orders and their current status monitors that users respect and follow the ethics of the website.

The Manager of the site has also access to the categories of the website. The manager can update, add and delete categories.

Mail system

.NET has a mail object contained in the System.Net which is part of the .NET BCL. The mail object is used to send emails from the website to users. The Mail object provides methods to do the most important aspect in mail handling and asynchronous handling can even be added if the manager wants it to be done. This can be of relevance when the manager starts sending out newsletters in the future. Configuration was made on the webconfig file so that I could use SMTP (Simple Mail Transfer Protocol) when sending messages. Considering the fact that I use SSL (Secure Sockets Layer) when sending mails I had to implement the email interface to be able to understand that the mail has to go through the SSL channel. The mail of the website and login properties had to be defined in the webconfig file. After a member has finished registration I use the send email event in the membership module and give it certain attributes to make it possible for the newly registered member to receive an email confirming his or her registration. This same method is implemented in the password recovery phase.

Site Navigation

Menus on the website are seen based on who you are and what you can do. If you are an anonymous user you get the default view of the navigation menu. If you are logged in and an authenticated user the navigation menu changes to let you have access to pages to upload and manage your books. Finally the administrator of the site has access to all pages on the website.

This was done in the webconfig file and also by configuring the administrative unit of the website. To control and restrict menus that appear on the application, it was done in the webconfig file by adding some attributes to the sitemap provider of the site.

Configurations and error handling on pages

All configurations of this application have been placed in the webconfig file. It makes it easy to manage certain aspects of the website because there is only one page holding all these important information. On this page there are configurations on the profile, mail, membership, security, database connection and error handling aspects for this website in general.

When there is error on the webpage the user sees a friendly informative page saying what happened on the website. These pages are all found in the error folder on the website. This folder has two pages and in them are messages telling the user what's gone wrong on the website.

On the other hand the manager gets a mail about what went wrong on the website. The details about the problem and stack code are shown to the administrator of the website. When the manager gets this mail he can later fix the problem or inform the users about it in a newsletter if it warrants a longer period.

Testing and validation

For the application or website to be deployed it has to be tested .hence test cases will be written to test this application. They are many types of test to be carried out on a web application from performance, functionality, database loading time, response time, server time handling, user's actions and many others. I will not carry out all types of test for the application considering the time scale to present this project. Hence performance check related to upload time, memory usage will be part of a future test .I will focus the test cases on functionality, security and performance.

The latter test on the website will make sure that the website provides the right results and outcome. The test will help reduce unpredictability on the website .I will run test on various browsers making sure that the application produces the same result and is stable on the major popular browsers.

Finally the last test will be the checking of all input source such as query strings, web services and textboxes. This will help prevent cross-side scripting attacks and SQL injection

Test cases

Browser check

Mozilla, Firefox, Safari. I Explorer 8 Opera, Chrome

1. Load the website on all browsers
2. Registration on the website using all the above browsers and making sure a new member get registration feedback email

Input source test

1. Change the query string to a digit if it's a string or to string if it's a number
2. Inject JavaScript code on the feedback page
3. Fill forms with wrong data types for each data field on the website.
4. Test for SQL injection

Functionality test

Here is a list of some functionality that will be tested.

1. Load new books on the website rapidly to test on updates of books.
2. Is the information loaded on the website controlled and checked? For example putting the price of a book at 100 000kr and loading unethical images on the website
3. Add all the books to the shopping cart .Get the total before and total when checking out.
4. After paying at PayPal does the user get back the right feedback report?
5. Is it easy to move between pages on the website?
6. Adding, deleting and updating items the user uploaded
7. Clicking buttons haphazardly on the website and seeing what happens

8. Are the dates of events right on every feedback
9. Enter wrong ISBN numbers
10. Search for books
11. Do the categories reflect what type of books the user wants to buy?
12. Can the visitor easily find a book on the website?

General view test

This test will be carried out by four randomly selected people .They will try to use the website like it's in the real world application and give me feedback and comments. They will answer questions like

1. Can the user find what they want by browsing on the website?
2. Are the website links clear and explicit?
3. Does the user understand the goal of the website and what it does?
4. Is it easy to move between pages on the website?
5. What is lacking in the website?
6. What are their recommendations?
7. Can they find the book they want?

Their recommendations and views will be written in the test result section of this test.

Results of test and recommendations

Results of browser check

I tested the application on the following browsers, Internet Explorer, Mozilla Firefox, Google Chrome and Apple's Safari browser. During the test I encountered problems with Firefox relating to rendering of book images. I fixed the problem by going through the generated html source and later setting all links to images directly through the root of the application. This solved the problem .With the other browsers I did not encounter any major problems. However there were some delays in Safari and Chrome browser. I discovered that it was the same problem the made Firefox not to render images. After solving this problem I went on to test other functionalities like registering a new member to the website. All these tests passed. Hence I can conclude that the application is capable to function well in any these major browsers.

Preventive measures taken to manage input data from various sources

1. I mentioned earlier that if there is any error on the application the user get a friendly message and the admin of the website get an email detailing the problems on the stack .Since ASP.net is build with security from ground up injecting malicious scripts to the website will not work since the default pagevalidation property is set to true for an ASP.net page in the maching.config file. This prevents the page from receiving this kind of scripts .Hence the user will get an error and the manager will get the real problem and that user can be banned from the website.
2. Even if Asp.net takes care of some basic secure functionality I still encode input gotten from the user so that links entered in the feedback message will not seen as links but just as normal text. I

made some pages to inherit from the Basepage. This makes it easy to add functionality to all pages that inherit from this page. The pages can use the method defined in the Basepage. I validate all input from the user before saving it into the database .I use the

3. RequiredFieldValidator controls, the RangeFieldValidator , DatatypeValidator, and the RegularExpressionValidator controls for emails and ISBN for books to make sure that input sources are right and the types are right.
4. I use parameters when it comes to input into the database hence preventing SQL injections to go through .I also use LINQ to SQL which by itself naturally prevents such attacks because all the data is parameterized before sending it to the database.

I validate query string types before using them. This was easily done using regular expressions for example to check if it's a decimal or string before using the query to generate any data from the database.

Results of Input source test

1. Change the query string to a digit if it's a string or to string if it's a number
When the query string on the BookItemsPage.aspx that shows the details was changed to a digit that was not found in the database .It returned nothing but when the button add was clicked the application crashed .I solved this by first checking in the DataList control items collection count as greater than 0 if it was lesser than one the add to cart button was not enabled.
2. Inject JavaScript code on the feedback page
The user is given an error message
3. Fill forms with wrong data types for each data field on the website.
The user gets an error message from the allocated error pages on the website
4. Test for SQL injection
Same as above the user get an error message.

However with these error messages the website still functions and the user can still make their purchases or upload new books to the website.

Results of functionality test

The results are divided into three categories namely passed, failed and unstable respectively

When the test results are labeled as PASSED, it means any of the following happens.

- The user gets a feedback message to enter correct input
- The awaited feedback is right
- The shopping cart values were right.
- The application responds in the way it was expected.

When a result is labeled FAILED it means that the website failed to take care of that problem or did not handle it properly hence future checks into the current problem has to be made.

When the result is labeled UNSTABLE it means that it is unpredictable and the awaited results can be wrong or right. Hence work on this area has to continue.

The results

1. Load new books on the website rapidly to test on updates of books. ->PASSED
2. Is the information loaded on the website controlled and checked? For example putting the price of a book at 100 000kr and loading unethical images on the website -> UNSTABLE
3. Add all the books to the shopping cart .Get the total before and total when checking out. ->PASSED
4. After paying at PayPal does the user get back the right feedback report? ->PASSED
5. Is it easy to move between pages on the website? ->PASSED
6. Adding, deleting and updating items the user uploaded ->PASSED
7. Clicking buttons haphazardly on the website and seeing what happens ->PASSED
8. Are the dates of events right on every feedback ->PASSED
9. Enter wrong ISBN numbers -> PASSED
10. Search for books ->PASSED

Results of general view test

- Can the website handle cases for fraudulent cards on the payment gateway since it takes three days for transfers to get through .? Answer : The manager of the website will be responsible for that
- They could not save their current orders when the checkout process failed hence recommended that I create a way of saving orders.
- This user will like the website to be multicultural.
- Ability for users to make a bargain on books.
- Ability for users to disable their accounts.
- How will postal fee be taken care of?
- Movement within the website is easy and they could find their way around.
- A more detailed message about the owner of the website and their goals will build a user relationship and understanding because they know about the company.
- It will be nice if there was the ability for the users to reserve books on the website

Conclusion and future work

Conclusion

My goal was to create an application where people will upload books and be able to sell them online. The current application has fulfilled these goals. I followed the specifications strictly but enhanced some of the features when there was need for it to be done. With the goals achieved the basis of the application and this project has been achieved. Building this web application has been challenging and enriching because throughout the project I learnt a lot about .Net and understand what it takes to build an E-commerce store. There have been challenges especially when it came to the payment gateway and making sure that the application responses in a predictable.

Careful planning made my job easier because I had to carefully think about the type of architecture, the design, the database types to use and what type of business objects to create. When this was done I proceeded with implementation.

Choosing the .Net framework for this project made the project to be realizable in that amount of time because the framework handled most of the heavy burden in underlying connections, security, portability and functionality. I could focus on the creation of the application and hence in this amount of time could develop a complete online shopping cart and membership site.

As I came to the end of the project I realized that there are many enhancements that can be made on the application. Some of these ideas came from those who tested the application and some of them from me. I decided to follow the specification because there were realistic to achieve in this given amount of time. Any other enhancements to the application can be done in future development of the application.

Future work

Here is what can be added in the future on this website to increase its usability, user experience and portability of the website. There is a lot to be done hence this application can be considered as a starting point for something big to come. It will need more time and resources for all these to be done but it is still very realistic and possible to achieve.

- Add a screening mechanism to the website when uploading books. Control the prices, images and book names before loading it to the main page of the website
- Add more payment gateways
- Add multi categories for books
- Adding a better order management portal
- Addition of themes
- Possibility to save orders and confirming it later
- Reservation of books
- Add forums and videos
- Access the applications on a mobile phone and reserve books
- Add E-Book formats like PDF and others in the selling portal

- Add a preview page after adding books.
- Greater user experience
- Multilayer architecture with separation on concerns
- Possibility for the buyer to bargain

References

1. Daniel S , (2008),Illustrated C# 2008, Apress
2. Hanspeter M. ,Wolfgang B, Dietrich B., Albrecht W. (2003), .NET Application Development with C#, ADO.NET, ASP.NET and Web Services, Addison Wesley
3. Ian S. (2000), Software Engineering, Addison Wesley;
4. Matthew R. (2000) Beginning E - Commerce With Visual Basic, Asp, Sql Server 7.0 and MTS ,Wrox
5. Marco B. (2006) ASP.NET 2.0 Website Programming: Problem - Design – Solution, Wrox
6. Mark N. and Steve W.(2001) (eBuisness Essentials) Technology and network requirements for mobile and online markets , John Wiley and Sons
7. Thomas A. S., (1994), Data Structures, Algorithms and Software Principles in C , Addison Wesley
8. N.Bandyopadhyay(2002) E-commerce Context, Concepts and Consequences, McGraw
9. Vanessa D.(2001) , Designing Easy- to Use websites , Addison-Wesley
10. <http://msdn.microsoft.com/en-us/library/bb355989.aspx>.

Figure References

1. Fig 1 : <http://sv.wikipedia.org/wiki/Fil:DotNet.svg>
2. Fig 2: http://sv.wikipedia.org/wiki/Fil:Overview_of_the_Common_Language_Infrastructure.svg
3. Fig 3 :Data Structures, Algorithms and Software Principles in C (Page: 678)

Appendix

SecondHandBooks

Sell books and make money and buy books and save money

[Home](#) [Sell Books](#) [Edit Books](#) [My Profile](#) [Join Us](#) [About Us](#) [Contact](#)
[Logout](#) Logged in as ngala

Search Shop

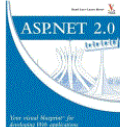
Categories

- Data
- Historia
- Ekonomi
- Matte
- Naturvetenskap
- Sport
- Biology
- Fisik
- Samhälle
- Resor
- Filosofi
- Familj
- Barn
- SjukVård
- Student
- Spel
- Tidning
- Roman

[Home](#) > Home

Asp.net 2.0


456,00 kr



ISBN: 4567-345-2345
[See details](#)

Men & Women


456,00 kr



ISBN: 3456-01-34-23
[See details](#)

C# 4.0


456,00 kr



ISBN: 5678978-7896
[See details](#)

Asp.net


345,00 kr



ISBN: 567-1234-456
[See details](#)

Kopplingar Mellan Oss


567,00 kr



ISBN: 456-0912-345-123
[See details](#)

Bok

567,00 kr



ISBN: 456-678
[See details](#)

Shopping Cart

Your cart is currently empty

[Order History](#)

Latest books

Done

Local intranet | Protected Mode: Off

100%

Log in view

[Home](#)
[Sell Books](#)
[Edit Books](#)
[My Profile](#)
[Join Us](#)
[About Us](#)
[Contact](#)
[Management](#)

[Manage Products](#)
[d in as Manager](#)
[See Orders](#)
[Manage Categories](#)

[Categories](#)
[Shopping Cart](#)

Your cart is currently empty

[Order History](#)
[Latest books](#)

[Home > Home](#)

Search Shop

Categories

- Data
- Historia
- Ekonomi
- Matte
- Naturvetenskap
- Sport
- Biology
- Fisik
- Samhälle
- Resor
- Filosofi
- Familj
- Barn
- SjukVård
- Student
- Spel
- Tidning
- Roman

Asp.net 2.0
456,00 kr

ISBN: 3456-01-34-23
[See details](#)

Lägg till

Men & Women
456,00 kr

ISBN: 3456-01-34-23
[See details](#)

Lägg till

C# 4.0
456,00 kr

ISBN: 5678978-7896
[See details](#)

Lägg till

Asp.net
345,00 kr

ISBN: 567-1234-456
[See details](#)

Lägg till

Kopplingar Mellan Oss
567,00 kr

ISBN: 456-0912-345-123
[See details](#)

Lägg till

Bok
567,00 kr

ISBN: 456-678
[See details](#)

Lägg till

http://localhost:50742/WebShop/Management/ManageCategories.aspx Local intranet | Protected Mode: Off 100%

Administrator view of the Website

Upload books view

Product	Price	Total	Quantity
C# 2008	456,00 kr	456,00 kr	1
Python for Beginners	787,00 kr	787,00 kr	1
Total: 1 243,00 kr		Update totals	

The Shopping cart view

SecondHandBooks

[Home](#) [Shopping Cart](#)

[Login](#)

[Delivery Address](#)

[Payment](#)

[Finish](#)

Order Complete

	Product	Price	Total	Quantity	
	Nature	456,00 kr	456,00 kr	<input type="text" value="1"/>	

	Asp.net 2.0	456,00 kr	456,00 kr	<input type="text" value="1"/>	
---	-------------	-----------	-----------	--------------------------------	---

Total: 912,00 kr

Shipping Method:

Please confirm you wish to have deducted from your credit card.



© Copyright 2010 Mangofly Design by: [ntasoftware](#) |

Local intranet | Protected Mode: Off

100%

CheckOut Process

ngalatalla@gmail.com

Order #87

Total with shipping: \$1,824.00 USD

PayPal is the safer, easier way to pay



PayPal securely processes payments for ngalatalla@gmail.com. You can finish paying in a few clicks.

Why use PayPal?

- Use your credit card online without exposing your card number to merchants.
- Speed through checkout. No need to enter your card number or address.

Sign up for a PayPal account and [continue checkout](#).

Cancel and return to [ngalatalla@gmail.com](#).

LOG IN TO PAYPAL

Email:

Password:

[Log In](#)

Forgot [email](#) or [password](#)?

PayPal. The safer, easier way to pay.

For more information, read our [User Agreement](#) and [Privacy Policy](#).

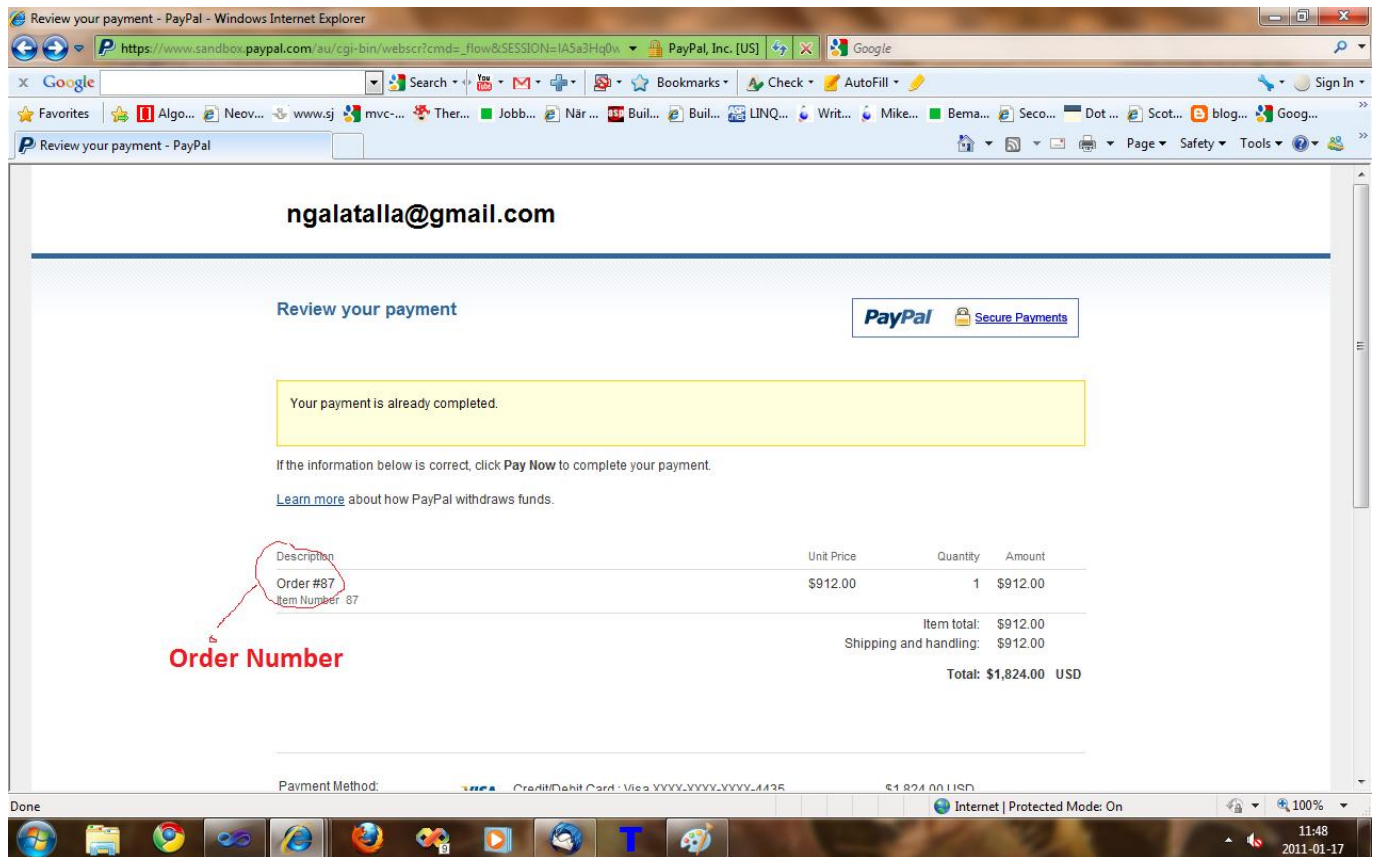


Test Site

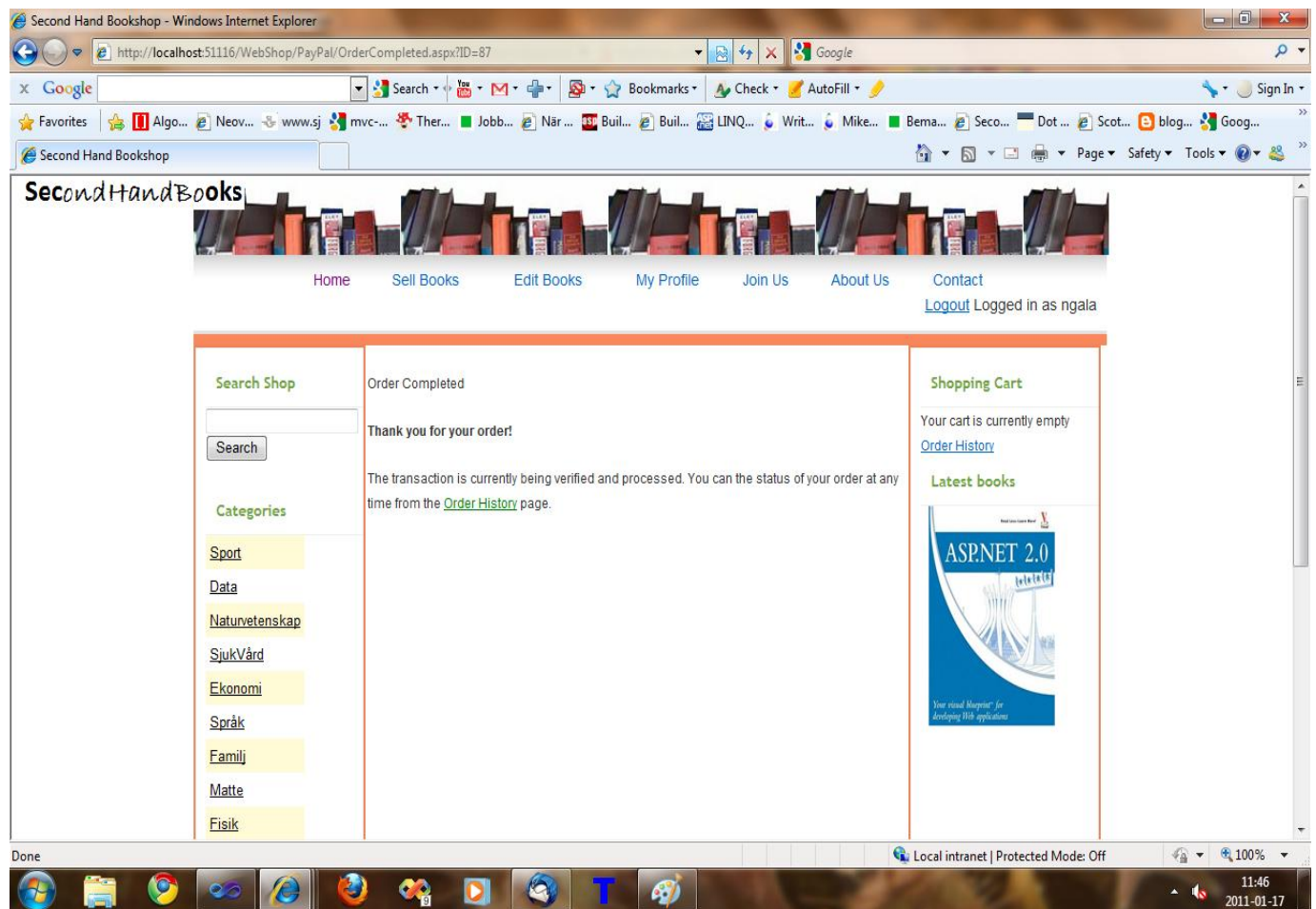
Internet | Protected Mode: On

100%

PayPal Login



Order in Paypal



Order completed confirmation