

CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

DESIGN AND IMPLEMENTATION OF
A WEB BASED
BOOK BUYING/SELLING SYSTEM FOR STUDENTS
USING THREE TIER ARCHITECTURE

A thesis submitted in partial fulfillment of the requirements

For the degree of Masters of Science

in Software Engineering

By

Arastu J Shah

May 2011

The thesis of Arastu J. Shah is approved:

| | |
|------------------------------|-------|
| <hr/> | <hr/> |
| Richard Covington, Ph.D. | Date |
| <hr/> | <hr/> |
| George Wang, Ph.D. | Date |
| <hr/> | <hr/> |
| Robert Lingard, Ph.D., Chair | Date |

ACKNOWLEDGEMENTS

I would like to thank Professor Robert Lingard for being my chair and giving me his valuable suggestions, guidance and support during the entire duration of my Masters studies including the Graduate Level project and Thesis.

I would also like to thank my parents who supported me in every way possible and paid my tuition throughout the course of study.

Finally, I would like to thank my thesis committee members, faculty and staff of Computer Science Department and Honors Coop CECS Department at California State University, Northridge for a wonderful experience during my studies at CSUN and Honors Coop. at Amgen Inc.

TABLE OF CONTENTS

| | |
|---|------|
| SIGNATURE PAGE..... | ii |
| ACKNOWLEDGEMENTS..... | iii |
| LIST OF FIGURES..... | vi |
| LIST OF TABLES..... | vii |
| ABSTRACT..... | viii |
| 1. Introduction..... | 01 |
| 1.1. Analyzing the Problem..... | 02 |
| 1.1.1. Problem Statement Table..... | 02 |
| 2. Requirements Baseline..... | 06 |
| 2.1. Brainstorming Ideas - Initial Features..... | 06 |
| 2.2. Features Prioritization..... | 09 |
| 3. Vision Document | 11 |
| 4. Supplementary Specifications..... | 23 |
| 5. Unified Modeling Language..... | 29 |
| 5.1. Use Case Diagram..... | 30 |
| 5.2. Sequence Diagram..... | 31 |
| 5.3. Activity Diagram..... | 33 |
| 5.4. Software System Architecture Diagram..... | 34 |
| 5.5. Software System Architecture Detailed Diagram..... | 35 |
| 6. Functional Requirement Specifications..... | 36 |
| 7. Database Artifacts and Table Design..... | 57 |
| 8. Software Architecture..... | 61 |
| 8.1. Some types of Application Architecture..... | 61 |
| 8.1.1. Single Tier..... | 61 |
| 8.1.2. Two Tier..... | 61 |
| 8.1.3. Three Tier..... | 62 |
| 8.2. Why use Three Tier Architecture..... | 63 |
| 8.3. What are the disadvantages of using a Three Tier Architecture..... | 64 |
| 8.4. Three Tier Architecture in the Studntstore Application..... | 65 |
| 8.5. What are some other architectures..... | 69 |
| 8.5.1. ASP.NET Model-View-Controller..... | 69 |
| 8.5.1.1. Advantages..... | 70 |
| 8.5.1.2. Disadvantages..... | 71 |
| 8.6. ASP.NET Web Forms vs. ASP.NET MVC..... | 71 |
| 8.7. Three Tier Architecture vs. ASP.NET MVC..... | 71 |
| 9. Test Driven Development Software Methodology..... | 72 |
| 9.1. Advantages of using TDD..... | 73 |
| 9.2. Disadvantages of using TDD..... | 75 |
| 9.3. Limitations of TDD..... | 76 |
| 9.4. Development of the Studntstore Application using TDD..... | 76 |
| 10. Testing Framework..... | 77 |
| 10.1. What is Unit testing..... | 77 |
| 10.2. Advantages of Unit tests..... | 77 |
| 10.3. Disadvantages of Unit tests..... | 77 |
| 10.4. Techniques of Unit testing..... | 77 |

| | |
|---|-----|
| 10.5. What is Integration testing..... | 78 |
| 10.6. Benefits of Unit testing over Integration testing..... | 78 |
| 10.7. Benefits of Integration testing over Unit testing..... | 78 |
| 10.8. Integration Testing tool alternatives available in the market | 78 |
| 10.9. Other testing tool alternatives available in the market | 79 |
| 10.10. Comparison of MSTest and NUnit testing frameworks..... | 79 |
| 10.10.1. Advantages of MSTest over NUnit..... | 79 |
| 10.10.2. Disadvantages of MSTest over NUnit..... | 79 |
| 10.11. How to create Unit tests using MSTest in Visual Studio 2008..... | 80 |
| 11. Security - Forms Based Authentication..... | 87 |
| 11.1. Understanding the Forms Authentication Workflow..... | 87 |
| 11.2. Why use a Membership provider..... | 91 |
| 11.3. Roles using SqlRoleProvider..... | 91 |
| 11.4. Definitions..... | 92 |
| 12. Master Page and CSS Stylesheets..... | 93 |
| 13. Challenges faced during the development lifecycle of the Studntstore application..... | 94 |
| 13.1. Three Tier Architecture implementation..... | 94 |
| 13.2. Database design and programming..... | 100 |
| 13.3. Registration and Login Functionality implementation..... | 100 |
| 13.4. Business Class Layer implementation..... | 100 |
| 13.5. Testing and TDD..... | 101 |
| 13.6. User Interface..... | 101 |
| 13.7. Validating User Input..... | 101 |
| 13.8. Report Writing..... | 101 |
| 14. Conclusion..... | 103 |
| 15. References | 104 |

LIST OF FIGURES

| | | |
|-----------------|---|----|
| Figure 1.1: | Studntstore System Fishbone Diagram of Root Cause Analysis..... | 04 |
| Figure 1.2: | System Boundaries..... | 05 |
| Figure 3.1: | Product Perspective diagram..... | 14 |
| Figure 4.1: | An example display of Studntstore system - My Books page..... | 27 |
| Figure 5.1: | Use Case - All Features..... | 30 |
| Figure 5.2.1: | Intent to Register Sequence diagram..... | 31 |
| Figure 5.2.2: | Secure Login Sequence diagram..... | 31 |
| Figure 5.2.3: | Add book Sequence diagram..... | 32 |
| Figure 5.3: | Contact Seller Activity diagram..... | 33 |
| Figure 5.4: | Software System Architecture diagram..... | 34 |
| Figure 5.5: | Software System Architecture Detailed diagram..... | 35 |
| Figure 8.4: | Software System Architecture diagram..... | 65 |
| Figure 8.4.1: | ObjectDataSource diagram from MSDN website..... | 66 |
| Figure 8.4.2: | Software System Architecture Detailed diagram..... | 67 |
| Figure 8.4.3: | Business Logic Layer..... | 68 |
| Figure 8.4.4: | Data Access Layer..... | 69 |
| Figure 8.4.5: | MVC Architecture for the web - Diagram used from Pro asp.net MVC by Steven Sanderson..... | 68 |
| Figure 9.1: | Test driven development used from Wikipedia..... | 72 |
| Figure 10.11.1: | Unit test creation – Open a BLL file..... | 80 |
| Figure 10.11.2: | Unit test creation – Select Create Unit Tests..... | 81 |
| Figure 10.11.3: | Unit test creation – Select a BLL class method..... | 82 |
| Figure 10.11.4: | Unit test creation – Add expected and actual values in Unit test..... | 83 |
| Figure 10.11.5: | Unit test creation – Select Run Tests..... | 84 |
| Figure 10.11.6: | Unit test creation – Test in progress..... | 85 |
| Figure 10.11.7: | Unit test creation – Test Passed..... | 86 |
| Figure 11.1: | The Forms Authentication Workflow..... | 89 |
| Figure 11.2: | Membership framework-used from MSDN Microsoft ASP.NET website..... | 90 |
| Figure 13.1: | Describes my question posted on the ASP.NET forum..... | 97 |
| Figure 13.2: | An experienced developer trying to help on ASP.NET forum..... | 97 |
| Figure 13.3: | Describes how I resolved my issue using ASP.NET forum..... | 98 |
| Figure 13.4: | Another Example of use of Forum..... | 99 |

LIST OF TABLES

| | |
|--|----|
| Table 1: Controlled and quasi-controlled empirical experiments on TDD..... | 74 |
| Table 2: Empirical Case studies on TDD..... | 75 |

ABSTRACT

DESIGN AND IMPLEMENTATION OF A WEB BASED BOOK BUYING/SELLING SYSTEM FOR STUDENTS USING THREE TIER ARCHITECTURE

By

Arastu Shah

Masters of Science in Software Engineering

This thesis describes in complete detail all phases of the Software Development Lifecycle of an application for students that will allow them to buy and sell books on campus at their university to other students. The goal was to provide a solution, to students, that is more secured, economical and time saving compared to the present alternative websites such as Amazon, craigslist and other such similar websites. Further, an analysis and comparison of the Three-Tier software architecture, used in this project, is made with other types of software architectures to justify its benefits and suitability for this application. Additionally, the thesis also provides an insight into the Test Driven Development software methodology, MSTest Microsoft Testing framework, and Forms Authentication.

1. Introduction

The Studntstore application is a web-based book buying/selling system that will allow university students to buy and sell their books to other students in the university and get a better resale value by saving on website commissions, shipping charges, and shipping insurance costs. They will be able to do so by meeting on campus and using my web application features for free. It has been developed using Visual Studio 2008 tool, C# programming language, ASP.NET Web-Forms 3.5 and SQL Server 2008 Database. The project consists of a complete detailed set of documentation of all the software development processes involved in building a web application using a three-tier architecture. Chapter 1 is about analyzing the problem and the root cause which inspired me to develop this application. Chapter 2 gives an insight of the brainstorming ideas and initial features that I came up with along with their prioritization. Chapter 3 is a Vision document for this application. Chapter 4 is the Supplementary Specifications document. In Chapter 5, I have used Unified Modeling Language to come up with diagrams to get a better understanding of the product requirements. Chapter 6 is about the Functional Requirements Specifications. Chapter 7 highlights the Database Artifacts and tables design of my application. Chapter 8 discusses the different types of architecture, their advantages and disadvantages, and the three tier architecture as the best choice for the Studntstore application. Chapter 9 explains about the Test Driven Software Methodology applied in this project. Chapter 10 presents the testing framework used in this project. Chapter 11 helps us understand the Forms Based Authentication and Authorization used for user account management and secured login. Chapter 12 is a brief overview of the use of master pages and CSS Stylesheets. Chapter 13 helps the reader understand the challenges faced during the development of this application. Finally, Chapter 14 is a concluding chapter of this thesis.

Language and Platform used

I have used the following in developing the Studntstore application

1. ASP.NET Web Forms 3.5
2. C# Programming language
3. Visual Studio 2008 Version 3.5 SP1
4. SQL Server 2008

ASP.NET Web Forms technology was developed by Microsoft in 2002. The initial version 1.0 was updated with better versions 2.0 in 2005 and 3.5 in 2008. It consists of a set of User Interface (UI) components with a stateful object oriented Graphical User Interface (GUI) programming model.

1.1. Analyzing the Problem

1.1.1. Problem Statement Table

| Element | Description |
|---------------------------|--|
| The Problem of... | The CSUN students do not have a secure website through which they can buy/sell books to other CSUN students directly by meeting with them on campus. Thus, having to use websites such as Amazon and craigslist makes them loose more money and time. |
| Affects... | University Students |
| And Results in... | <ul style="list-style-type: none">● Time spent in listing their books up on websites such as Amazon and craigslist and waiting for them to be sold.● Time spent and money lost in competing with other Amazon sellers of the same book by having to lower their book value in order to sell it quickly.● Money lost in commission charged by Amazon for selling a book.● Time spent in going to the post office to ship their books to the seller.● Money spent in shipping costs associated with a book.● Money spent in insurance costs associated with shipping.● Time spent in refunding money for books that are lost in shipping.● Time spent in insurance claims for lost books.● Money lost in insurance claims for failure to produce a receipt of lost book to post office.● Time spent sometimes dealing with scams on websites such as craigslist where the seller identity is not known. |
| Benefits of a solution... | <ul style="list-style-type: none">● New online system will reduce the time spent in selling a book as all the users will be close to the CSUN campus and will have access to website.● New online system will not charge any commission for a transaction. |

| | |
|---------------------------|---|
| | <ul style="list-style-type: none"> ● New online system will have a messaging system using which members can decide on a place to meet on campus to complete a transaction. ● New online system will save time of students as they don't have to go to the post office anymore. ● New online system will eliminate the shipping costs as members can meet in person. ● New online system will also eliminate the insurance costs associated with shipping. ● New online system will eliminate the loss of time involved in dealing with books lost during shipping. ● New online system will allow only those users that have an .edu account to become a website member to use the features of the website. Thus, greatly reducing the possibility of scams. ● New online system will thus provide a better resale value to a student. ● New online system will allow students to interact with each other and get advice about a course from other students who have taken it in the previous semesters. This could happen when they meet on campus to complete a transaction. |
| Also Affects... | Arastu Shah |
| And Results in... | <ul style="list-style-type: none"> ● Absence of such a system not generating any revenue on advertising of different products to a potential market of university students. |
| Benefits of a solution... | <ul style="list-style-type: none"> ● New online system will help him get advertisers to advertise to students about any products. ● New system will allow future expansion of the website to allow students to sell other items through the website. ● New system will allow him to advertise any product to the students |

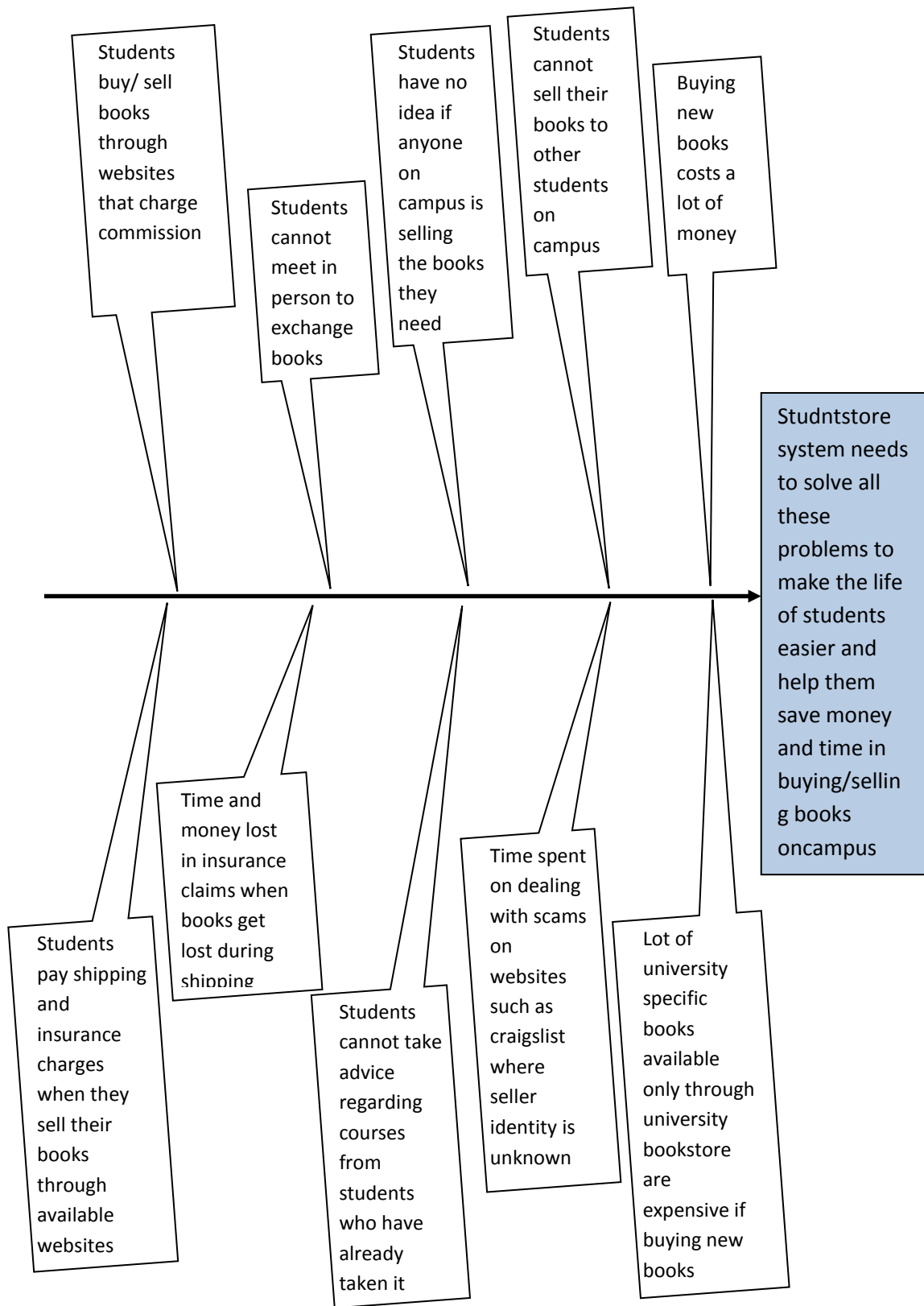


Figure 1.1: Studntstore System Fishbone Diagram of Root Cause Analysis

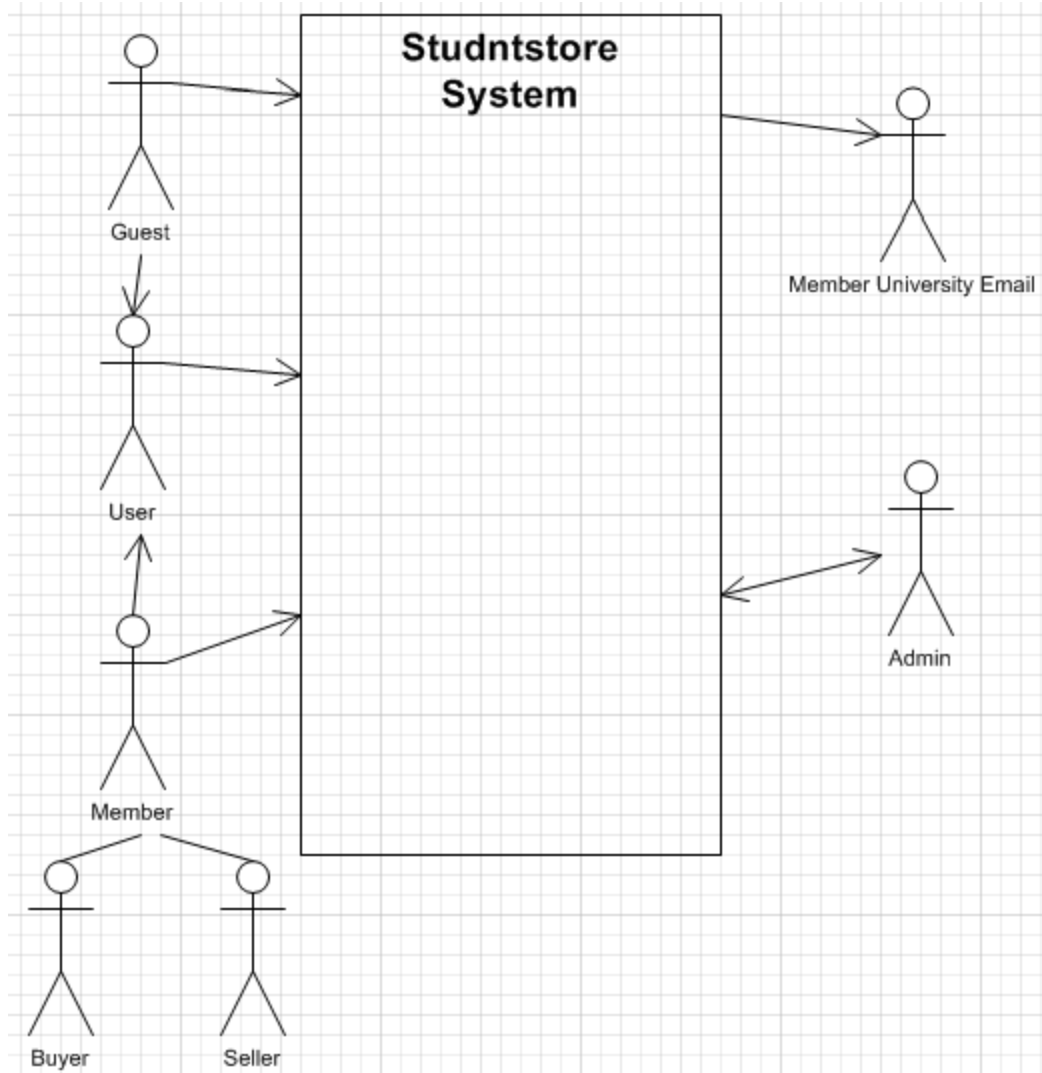


Figure 1.2: System Boundaries

2. Requirements Baseline

2.1. Brainstorming Ideas – Initial Features

While studying at CSUN, I realized that every semester, I was buying expensive books from websites such as Amazon. At the end of every semester, I had to sell those books through Amazon at a lower price because of competition from other sellers. I lost a lot of money also because of Amazon commission, shipping and insurance costs. My classmates at CSUN have experienced similar issues. Based on my own experience and conversations with my classmates, I thought of developing a web application, with the following features, that will solve this issue and help students save money.

| ID | Features |
|----|--|
| 1. | Link to Home <ul style="list-style-type: none">• A link to a page that explains the purpose of the website |
| 2. | Link to About us <ul style="list-style-type: none">• A link to a webpage explaining who works behind the scenes of this website |
| 3. | Link to Contact us and Feedback <ul style="list-style-type: none">• A link to help visitors and members to contact the site owner/admin for various issues |
| 4. | Link to view Listed Books for a visitor and search for a specific book <ul style="list-style-type: none">• A link to a webpage that shows the books listed on the website but doesn't allow the visitor to contact the seller |
| 5. | Link to Register <ul style="list-style-type: none">• A link for a visitor to register in order to be able to view and list books to buy and sell |
| 6. | Live help chat <ul style="list-style-type: none">• A feature to help the visitors and members in real time |
| 7. | Link to map of CSUN on Google maps <ul style="list-style-type: none">• A link to a map that shows the campus of CSUN on Google maps |
| 8. | Areas on leftmost side and rightmost side and bottom for advertisements <ul style="list-style-type: none">• Areas on website where people can advertise about something for a minimal fee |
| 9. | Link for Administrator login <ul style="list-style-type: none">• A link for admin to have full control over all pages on website |

| | |
|------------|---|
| 10. | Secure login <ul style="list-style-type: none"> • A secure forms-based authentication using username and password created on registration |
| 11. | Link to My Books <ul style="list-style-type: none"> • A link for a member to view all his listed books |
| 12. | Link to Listed Books <ul style="list-style-type: none"> • A link for members to view all member listed books |
| 13. | Search feature for listed books using categories of author, publisher and book name <ul style="list-style-type: none"> • A search box for a member to search a specific book among all listed books |
| 14. | Link to contact seller of a particular listed book <ul style="list-style-type: none"> • A link to direct to a page which allows a buyer to contact the seller of a particular listed book |
| 15. | Link to add books <ul style="list-style-type: none"> • A link for a member to add new books he wishes to sell |
| 16. | Link to delete my book <ul style="list-style-type: none"> • A link to delete a listed book |
| 17. | Link to a password recovery page <ul style="list-style-type: none"> • A Link to recover password or reset it incase member forgets his/her password |
| 18. | FAQ <ul style="list-style-type: none"> • A list of frequently asked questions about website or related to website features and other topics |

Features of Studntstore application, sorted by priority

The prioritization is based on the importance of each feature in meeting user needs and making this website stand apart from existing alternative websites such as Amazon, Craigslist etc. For example, since this website is for exclusive use by university students, registration with an .edu account is crucial.

| ID | Features |
|----|--|
| 5 | Link to Register |
| 10 | Secure Login |
| 17 | Link for password recovery page |
| 9 | Link for administrator login |
| 15 | Link to add books |
| 16 | Link to delete my book |
| 11 | Link to My Books |
| 12 | Link to Listed Books |
| 13 | Search feature for listed books using categories of author, publisher, and book name |
| 14 | Link to contact seller of a particular listed book |
| 1 | Link to Home |
| 2 | Link to About us |
| 3 | Link to Contact us and Feedback |
| 4 | Link to view Listed Books for visitor and search for a specific book |
| 8 | Areas on leftmost side and rightmost side and bottom for advertisements |
| 7 | Link to map of CSUN on Google maps |
| 6 | Live help chat |
| 18 | FAQ |

2.2. Features Prioritization

The ranking below has been done by Arastu Shah. It is based on his own experience, discussions with university students and their inputs.

Critical (Essential feature required to meet user needs)

Important (Feature important for effectiveness and efficiency of system. Lack of inclusion may affect user satisfaction but the release will not be delayed.)

Useful (This may be helpful but no significant revenue loss or user satisfaction impact if it is not included)

| ID | Feature Name | Critical | Important | Useful | Holds |
|----|--|----------|-----------|--------|-------|
| 1 | Link to Register | Y | | | Y |
| 2 | Secure Login | Y | | | Y |
| 3 | Link for password recovery page | Y | | | Y |
| 4 | Link for administrator login | Y | | | Y |
| 5 | Link to add books | Y | | | Y |
| 6 | Link to delete my book | Y | | | Y |
| 7 | Link to My Books | Y | | | Y |
| 8 | Link to Listed Books | Y | | | Y |
| 9 | Search feature for listed books using categories of author, publisher, and book name | y | | | Y |
| 10 | Link to contact seller of a particular listed book | Y | | | Y |
| 11 | Link to Home | | Y | | Y |

| | | | | | |
|----|---|--|---|---|---|
| 12 | Link to About us | | Y | | Y |
| 13 | Link to Contact us and Feedback | | Y | | Y |
| 14 | Link to view Listed Books for visitor and search for a specific book | | | Y | Y |
| 15 | Areas on leftmost side and rightmost side and bottom for advertisements | | | Y | Y |
| 16 | Link to map of CSUN on Google maps | | | Y | N |
| 17 | Live help chat | | | Y | N |
| 18 | FAQ | | | Y | N |

3. Vision Document

INDEX

| | |
|---|-----------|
| 1. Introduction..... | 12 |
| 1.1 Purpose of the Vision Document | |
| 1.2 Product Overview | |
| 1.3 References | |
| 2. User Description..... | 13 |
| 2.1 User demographics | |
| 2.2 User profiles | |
| 2.3 User environment | |
| 2.4 Key user needs | |
| 2.5 Alternative and competition | |
| 3. Product Overview..... | 14 |
| 3.1 Product Perspective | |
| 3.2 Product Position Statement | |
| 3.3 Summary of Capabilities | |
| 3.4 Assumptions and Dependencies | |
| 4. Feature Attributes..... | 15 |
| 5. Features..... | 16 |
| 6. Use Case Examples..... | 19 |
| 7. Other Product Requirements..... | 21 |
| 7.1 Applicable Standards | |
| 7.2 System Requirements | |
| 7.3 Licensing, Security, and Installation | |
| 7.4 Performance Requirements | |
| 8. Documentation Requirements..... | 21 |
| 8.1 User Manual | |
| 8.2 Online Help | |
| 8.3 Installation guides, Configuration and Read me files | |
| 8.4 Labeling and Packaging | |
| 9. Glossary..... | 22 |

1. Introduction

1.1. Purpose of the Vision Document

To collect, analyze and define high-level user needs, and features of Studntstore web application that will provide service of buying and selling of books for students using an intuitive and dynamic C# Web Forms user interface.

The User Description outlines the perspective of the users of my system through the Market/User Demographics, profiles, and environment and lists the key user needs.

The Product Overview makes us aware about the perspective of the product through a block diagram and a summary of the major benefits and features that the system will provide. It also gives us information about the cost and pricing associated with the project.

The Feature Attributes describe the attributes that would be used to evaluate, track, prioritize and manage the features.

The Product Features lists the most important features of the product.

The Exemplary Use Cases gives the user, an understanding of how the system is intended to be used.

The Other Product Requirements lists the Licensing, Security and Installation related issues, the Performance and the System requirements.

The Documentation Requirements section describes the documentations that will need to be developed to support successful application deployment.

The Glossary at the end of the document lists the important terminology that has been used in this document.

1.2. Product Overview

The purpose of the Studntstore web application is to help the students get better resale value from selling their books on campus to other students when compared to selling them on Amazon or other book stores and also to allow students to buy used books at a much lower price by cutting down on costs such as bookstore commissions, shipping, shipping insurance etc. The system will also provide opportunities to interact with the campus community and seek advice on courses related to the books.

The Studntstore application will provide a webpage through which a member can contact a seller by email and meet at an agreed upon location to buy/sell books.

Some of the most important characteristics of this application include no website commissions, no shipping charges and only students who register on website with their .edu account will be able to make use of the features of this application, links to all member listed books, link to upload new books, ability to search within listed books using different categories, link to contact website administrator regarding suggestions or complaints about any issue.

1.3. References

- 1.3.1. website www.Amazon.com
- 1.3.2. website www.craigslist.org
- 1.3.3. website www.half.com
- 1.3.4. website www.abebooks.com

2. User Description

2.1. User demographics

Students buy books at the beginning of every semester for their enrolled courses. Then, they usually sell those books at the end of the semester. They have a need of a web application which can help them get maximum resale value of their books and lower the cost of purchase of new books by avoiding website commissions, shipping costs etc. Many students sell their books on Amazon which charges commission on every transaction. Also, once their book has been bought, they are required to ship it to the buyer including the costs of insurance while shipping. Selling on a website such as Craigslist is not very safe as there is no way to determine the identity of a person using that website. Many students sell their used books to a university bookstore which gets them very little money in return. There is no web application, available in the market for the students, which solves these problems. My Studntstore application will help solve all these problems of a student.

2.2. User profiles

The users are students enrolled in a university and have access to their university email account. Their computer related skills and the knowledge, browsing the internet and using the web application might vary from beginner to advanced levels.

Students can be divided into three categories;

1. Guest- A person who visits the website but is not a member yet.
2. Member- A person with a username and password created during registration on the website and has access to a university email account.
Member can be subdivided into
 - a. Buyer – A member who wishes to buy a particular listed book.
 - b. Seller – A member who has listed a book in order to sell it.
3. User – A person who is either a guest or a member of the website.

2.3. User environment

This system is a web application. Thus, the user can access this system using;

1. Personal operating systems such as Window XP, Vista, 7 or Mac OS.
2. General web browsers such as Google Chrome, MS Explorer, Mozilla Firefox, or Mac Safari.

2.4. Key user needs

This system must have following five features

1. Only students with an .edu account who register with the application shall be able to use the features.

2. The application shall provide personalized information such as a link to edit his/her profile information, his/her listed books and all members listed books.
3. There shall be no commission charged by the application on a transaction.
4. There shall be no shipping costs involved in a transaction.
5. Students shall be able communicate with each other through the application in order to meet on campus or elsewhere in person to complete a transaction.

2.5 Alternative and competition

Students are able to buy/sell books using websites such as Amazon.com, craigslist.org, abebooks.com, buy.com, half.com and other similar websites or the university book store, and some buy/sell to their friends.

3. Product Overview

3.1. Product Perspective

The Studntstore system is an independent and self-contained system. However, the system does require sending an email to the user's university email account in order to confirm the legitimacy of his account.

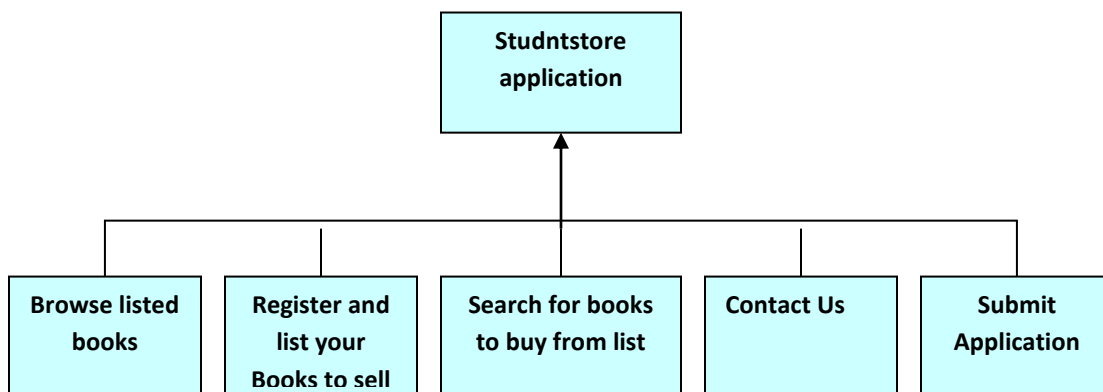


Figure 3.1: Product Perspective diagram

3.2. Product Position Statement

For students **who** are prospective book buyers or sellers and have a university .edu account, **the** Studntstore application is a web-based application **that** will help them accomplish this task in a more economical, time-saving, secured and organized manner. **Unlike** the present websites such as Amazon that charge commission for every product sold through their website, and websites in which you need to ship your book, websites such as craigslist where there are unknown users and scams, **my application** will allow only students with an .edu account to register and buy/sell books on campus without any commission or shipping costs.

3.3. Summary of Capabilities

The Studntstore application will have the following features:

- 3.3.1. Shall allow buying/selling books with no website commission, or shipping costs and save time by selling on campus in-person.
- 3.3.2. Shall allow buying/selling books among members with an .edu account only.

3.4. Assumptions and Dependencies

This application assumes current business practices and databases.

4. Feature Attributes

| | |
|-------------|--|
| Priority | Critical (Essential feature required to meet user needs) Important (Feature important for effectiveness and efficiency of system. Lack of inclusion may affect user satisfaction but the release will not be delayed.) Useful (This may be helpful but no significant revenue loss or user satisfaction impact if it is not included) |
| Risk | High (High probability that system will experience undesirable events such as schedule delays, cost overruns) Medium (The level of risk is moderate) Low (The level of risk is least with this feature) |
| Effort | High (This level of effort requires most time and resources than others) Medium (This level requires average time and resources) Low (This level requires the least time than others) |
| Feature | This describes the name of the feature that fulfills a user need. |
| Description | This describes in one or two sentences the system capability provided by feature to deliver benefit to the user. |

5. Features

| No. | Feature | Description | Priority | Risk | Effort |
|-----|---------------------------------|---|----------|------|--------|
| 5.1 | Link to Register | This allows a guest to become a member so that he can buy/sell his books through the application. | C | M | H |
| 5.2 | Secure Login | This feature allows members to login to application by authenticating their username and password with forms based authentication | C | H | H |
| 5.3 | Link for password recovery page | This will allow a member to recover his password by submitting his email so that the system sends the password to his university email address. | C | H | H |
| 5.4 | Link for administrator login | This allows an admin to login to the application in order to manage and administer all pages. | C | H | H |
| 5.5 | Link to add books | This capability allows a member to add books to his inventory for sale. | C | M | H |
| 5.6 | Link to delete my book | The feature will allow a member to delete a particular book and its information from his inventory. | C | M | H |
| 5.7 | Link to My Books | The member can view all his listed books by clicking on this link on his mainpage. | C | L | L |
| 5.8 | Link to Listed Books | This feature will allow a member to view all member listed books. | C | M | M |

| | | | | | |
|-------------|--|---|----------|----------|----------|
| 5.9 | Search feature for listed books using categories of author, publisher, and book name | This function will allow a member to search for a specific book using search categories such as author, publisher and book name. | C | M | H |
| 5.10 | Link to contact seller of a particular listed book | This feature will allow a member to contact the seller regarding inquiries related to a specific book listed by seller. | C | M | M |
| 5.11 | Link to Home | The user can read the contents of this page to find out about the purpose of this application and how he can benefit from using it. | I | L | L |
| 5.12 | Link to About us | This feature explains who build the application and the page will have a brief description related to the same. | I | L | L |
| 5.13 | Link to Contact us and Feedback | The user can contact the admin about any complains, comments or compliments and also about any other inquiries related to the application. | I | M | M |
| 5.14 | Link to view Listed Books for visitor and search for a specific book | A guest can use this feature to have an idea about the kind of books listed on the website. The guest can also search for a specific book but he cannot contact the seller before he registers. | U | M | M |
| 5.15 | Areas on leftmost side and rightmost side and bottom for advertisements | This feature will allow users, to advertise their products on this website. They can attract them to visit their websites or buy their products. | U | L | L |

| | | | | | |
|-------------|------------------------------------|--|----------|----------|----------|
| 5.16 | Link to map of CSUN on Google maps | This feature will allow a member to browse through the CSUN map embedded in Google maps and decide on a place where he can meet up with the seller in order to complete the transaction. | U | L | L |
| 5.17 | Live help chat | This will allow a user to get help in real time from a Chat Associate of the application during business hours. | U | L | L |
| 5.18 | FAQ | This feature will provide answers to commonly asked questions by users related to website. | U | L | L |

6. Use Case Examples

a. Add Books

| | |
|----------------------------|--|
| Use-case Name | Add books |
| Actors | System, Member |
| Brief Description | This use-case describes how a Member adds a book to his inventory in order to sell and so it also gets added to the general list of books for sale. |
| Flow of events | <p>Basic flow begins when a Member is logged on to his account in the system and clicks on Add books link</p> <ol style="list-style-type: none">1. The Member clicks on Add Books link on his Mainpage.2. The system shows a webpage with a form asking for book details.3. The Member fills in the form with book information and clicks Submit.4. System stores the book information in the database and lists the books in My Books list and All User Listed Books list. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">3. a. The Member fills in book information and clicks Cancel.b. Go to Step 2. |
| Pre-conditions | <ol style="list-style-type: none">1. The Member has access to internet.2. The Member is logged in the system. |
| Post-conditions | Book information is stored in database and the book gets listed. |

b. Intent to register

| | |
|----------------------------|---|
| Use-case Name | Intent to Register |
| Actors | System, Guest |
| Brief Description | The Guest intends to register in order to buy or sell his book |
| Flow of events | <p>Basic flow begins when a Guest is browsing one of website's general pages and he doesn't have a membership account yet.</p> <ol style="list-style-type: none">1. The Guest clicks the Register link or tries to access a Member's only page.2. The System directs the Guest to a webpage with a form to register. The Guest enters the information requested in form.3. The Guest clicks the submit button.4. The system shows a message that a new account has been created and asks the Guest to check his university email box for a mail which will contain a password to be used to login to this website. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">3. a. The Guest doesn't enter correct information. b. The System shows validation error. Go back to step 2. <p>Alternative flow 2.</p> <ol style="list-style-type: none">4. a. The Guest hits the cancel button. Go back to step 2. |
| Pre-conditions | <ol style="list-style-type: none">1. The Guest has access to internet.2. The Guest is accessing one of the website's general pages. |
| Post-conditions | <ol style="list-style-type: none">1. An email with a password is sent to his email address to verify the account. The member can login only using this password.2. The SQL database has his information stored. |

7. Other Product Requirements

7.1. Applicable Standards

The Studntstore is a web application and will be implemented on an interactive web portal based on C# and SQL Server 2008. This web application will be governed by following policies -

7.1.1. Conditions of Use

This webpage consists of the terms and conditions applicable while using this website. It consists of sections such as copyright, copyright claims, trademarks, license and site access, electronic communications etc.

7.1.2. Privacy Notice

This webpage answers questions related to the use, storage and access of personal information and how it is shared.

7.2. System Requirements

The supported host operating systems include Windows XP, Vista, and 7, Mac OS and supported browsers are Windows Internet Explorer, Mozilla Firefox, Google Chrome and Netscape.

7.3. Licensing, Security, and Installation

The application is not affiliated with the university and will not store credit card information. It does not require any installation.

7.4. Performance Requirements

The user must have access to the internet in order to use the Studntstore Application. A DSL, Cable, optical fiber or broadband internet connection is required to achieve better performance.

8. Documentation Requirements

This is a browser based web application so it will not require any user manual, installation guides, configuration files or label and packaging content. However, online help in the form of a FAQ page and a Live Help feature will be provided in the later release of the application.

8.1. Online Help

This application will include a Feedback page and Frequently Asked Questions link (FAQs) that will allow the students to submit/ask a question about something they cannot find on the website or related to any issues such as getting book information uploaded on website.

The application will also have a live help chat feature incorporated in the next release of the application.

9. Glossary

| No. | Notion | Definition |
|-----|-------------|--|
| 1. | Studntstore | Name of my application |
| 2. | Application | Refers to Studntstore |
| 3. | System | Refers to Studntstore |
| 4. | Website | Refers to Studntstore |
| 5. | Guest | A person who visits the website but is not a member yet. |
| 6. | Member | A person with a username and password created during registration on website and has access to a university email account. |
| 7. | User | A person who is either a guest to the website or a member of the website. |
| 8. | Inventory | List of Books and their information |
| 9. | FAQ. | Frequently Asked Questions |

4. Supplementary Specifications

INDEX

| | |
|--|-----------|
| 1. Introduction..... | 24 |
| 2. Functionality..... | 25 |
| 3. Usability..... | 25 |
| 4. Reliability..... | 25 |
| 5. Performance..... | 25 |
| 6. Supportability | 25 |
| 7. Design Constraints..... | 25 |
| 8. User Documentation and Help System Requirements..... | 26 |
| 9. Purchased Components..... | 27 |
| 10. Interface..... | 27 |
| 11. Licensing and Security Requirements..... | 28 |
| 12. Legal, Copyright, and Other Notices..... | 28 |
| 13. Applicable Standards..... | 28 |
| 14. Internationalization and Localization..... | 28 |
| 15. Physical Deliverables..... | 28 |
| 16. Installation and Deployment..... | 28 |

1. Introduction

1INT1 Purpose of the Supplementary Specifications

This document details the known requirements for Studntstore system. It also collects and organizes the system requirements not covered by the use case model such as additional functional requirements, nonfunctional requirements, and design constraints.

Scope

The requirements in this document pertains only to the Studntstore system

1INT2 Definitions, Acronyms and Abbreviations

| Term | Definition |
|-------------|--|
| Studntstore | Name of my application |
| Application | Refer to Studntstore Term |
| System | Refer to Studntstore Term |
| Website | Refer to Studntstore Term |
| Guest | A person who visits the website but is not a member yet. |
| Member | A person with a username and password created during registration on website and has access to a university email account. |
| User | A person who is either a guest to the website or a member of the website. |
| Inventory | List of Books and their information |
| FAQ. | Frequently Asked Questions |

2. Functionality

- F1. The Studntstore system will allow its members to list their books for sale and buy books from other members. The system will not charge any commission for buying/selling of book.
- F2. The system will have a messaging system using which members can contact each other regarding more information about book and also decide on a place to meet on campus at university.

3. Usability

- U1. The Studntstore system will have a GUI interface
- U2. It will have error messages displayed throughout the website to help user submit the correct input information and perform the correct steps in accomplishing a task.
- U3. The members will be able to search for a particular book through the search box rather than browsing through each and every book that is listed. The search can be done using different categories such as book name, author, publisher etc.
- U4. The users can submit a form through the Contact Us page requesting information to make the website easy to use.

3. Reliability

- R1. The Studntstore system will be available to users 24 hours per day and 7 days a week. The maintenance downtime will be 5 hours at max. on a weekend at a planned time and all members will be informed regarding the same.
- R2. The input data will be checked for range errors. This will make sure that the system doesn't crash with out-of-range values and thus making it more reliable.

4. Performance

The following performance characteristics of the system are based on minimum requirements of a broadband Internet connection with 1 Mbps (Megabits per second) or above speed:

- P1. The average response time for a transaction will be 1 second.
- P2. The maximum response time for a transaction will be 5 seconds

6. Supportability

The requirements that will enhance the supportability or maintainability of the system are:

- S1. The webpages will be built using the interactive C# Web Forms.
- S2. The naming conventions of Microsoft C# language will be followed.

7. Design Constraints

Studntstore System has design constraints on the system being built in three aspects; Developmental Tools, Platform Requirements, Internet Browsers and Architectural constraints. These constraints are based on the skill level and

knowledge of Arastu Shah and also based on the time frame within which this project had to be completed.

DC1. Developmental Tools

- The system shall be developed using Visual Studio 2008 with C# language, ASP.NET Web Forms 3.5, JavaScript and MS SQL Server 2008.

DC2. Platform Requirements

- The client portion of the Studntstore System shall operate on any personal computer with general operating systems such as MS Window 7, XP, Vista, or Mac OS. The client portion doesn't require any additional HDD and main memory (RAM) space.
- The server portion of the Studntstore System shall operate on the hosting website (Hostmonster or Godaddy) server.

DC3. Internet Browsers

- The web-based interface for the Studntstore System shall run in general web browsers such as MS Explorer, Google chrome, Mozilla Firefox, or Mac Safari.

DC4. Architectural constraints

- A three tier architecture consisting of presentation layer, business logic layer and data access layer will be used.

8. User Documentation and Help System Requirements

UD1. Getting started guides

Not applicable as this web application requires only a general knowledge of web browsing in order to be used.

UD2. User Guides

Not applicable as this web application requires only a general knowledge of web browsing in order to be used.

Online Help

This will include a Live Chat feature and Frequently Asked Questions link (FAQs) that will allow the users to submit/ask a question to the admin about something they cannot find on the website or related to any other issue.

UD3. Administration guide

Not applicable as this web application requires only a general knowledge of web browsing in order to be used.

User glossary

Not applicable as this web application requires only a general knowledge of web browsing in order to be used.

Read Me files and release notes

Not Applicable as this is a web application that can be browsed on internet and it will not be delivered to user.

UD4. Labeling and packing requirements

Not Applicable as this is a web application and it will not be delivered to user.

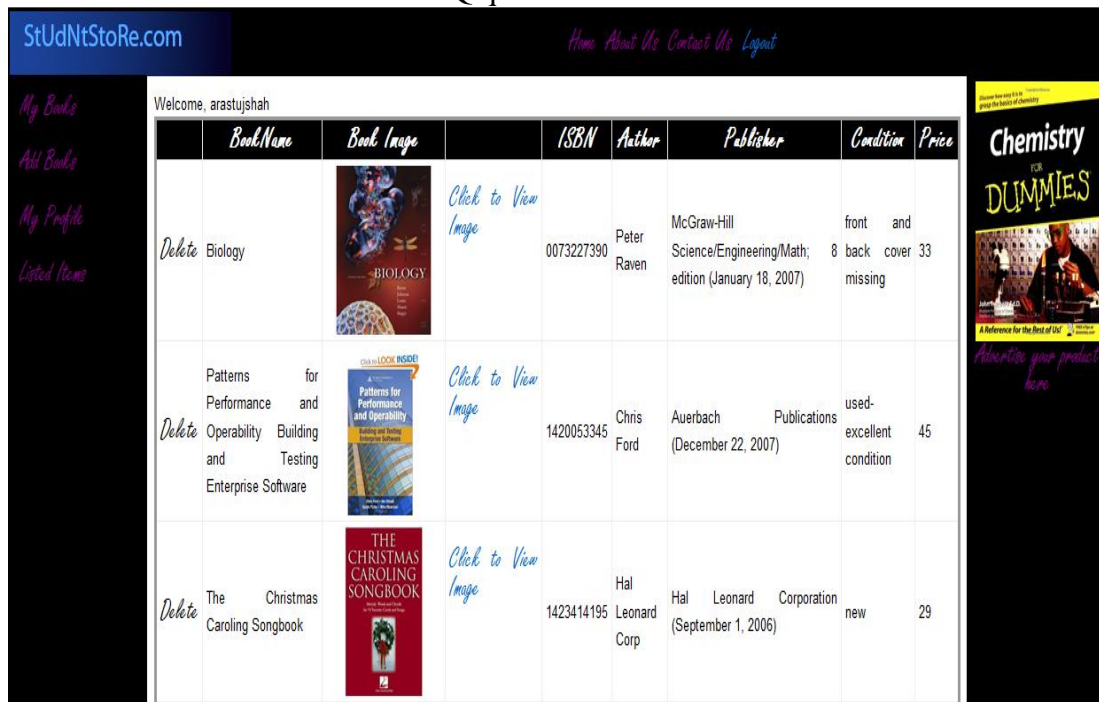
9. Purchased Components

This application does not require purchase of any components. It is web based and can be browsed for free using the internet.

10. Interface

IN1 User Interface

1. Register: Requests a guest to register before being able to use all the features of system.
2. My Account information - Storage for personal information (email addresses, phone numbers, etc.).
3. Add Books: A link for a member to add new books he wishes to sell.
4. My Books: A link for a member to view all his listed books.
5. Listed Items: A link for members to view all member listed books
6. Contact seller: A link which allows a member to contact the seller of a particular listed book.
7. Contact Us: A link to help visitors and members to contact the admin for any issue.
8. Ask: Links for FAQ questions.



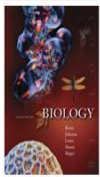

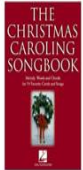
| | BookName | Book Image | | ISBN | Author | Publisher | Condition | Price |
|--------|---|---|-------------------------------------|------------|------------------|--|------------------------------|-------|
| Delete | Biology |  | Click to View Image | 0073227390 | Peter Raven | McGraw-Hill Science/Engineering/Math; edition (January 18, 2007) | front and back cover missing | 33 |
| Delete | Patterns for Performance and Operability Building and Testing Enterprise Software |  | Click to View Image | 1420053345 | Chris Ford | Auerbach Publications (December 22, 2007) | used-excellent condition | 45 |
| Delete | The Christmas Caroling Songbook |  | Click to View Image | 1423414195 | Hal Leonard Corp | Hal Leonard Corporation (September 1, 2006) | new | 29 |

Figure 4.1: An example display of Studntstore system - My Books page.

IN2 Hardware Interface

This application doesn't require any hardware interface to be supported.

IN3 Software Interface

1. HTTP protocol
2. Visual studio C# web interface

IN4 Communication Interface

This application doesn't require any communication interface to other systems or devices such as local area networks, local or remote devices.

11. Licensing and Security Requirements

There is no requirement of a license for using this system. The user needs to have a username and password created using registration to access all the features of the system. The authorization is done using Forms Based Authentication. All user information is kept confidential.

12. Legal, Copyright, and Other Notices

The system will have a trademark/logo and will not be affiliated with CSU Northridge or any other university.

13. Applicable Standards

The Studntstore is a web application and will be implemented on an interactive web portal based on C# language. This application will be governed by but not necessarily limited to the following policies:

AS1 Conditions Of Use

This webpage consists of the terms and conditions applicable while using it. It consists of sections such as copyright, copyright claims, trademarks, license and site access, electronic communications etc.

AS2 Privacy Notice

This webpage answers questions related to the use, storage and access of personal information and how it is shared.

14. Internationalization and Localization

The Studntstore application is for the students of CSUN, and currently, does not support any other universities. However, the system does not restrict students of other universities to use the website. Some features might not pertain to other universities. The components (information) of the system shall be accessible over the world for future expansion purposes.

15. Physical Deliverables

The system requires access to the internet. No CDs or manuals are required as system is to be used via internet using an internet browser.

16. Installation and Deployment

The Studntstore System does not require any installation on part of the user. The system will be hosted on a server of a hosting company. It will be available for user access via internet browser.

5. Unified Modeling Language

It is a standard modeling language used to model software applications, businesses and system architecture.

What is a model?

It is a representation of an object. For example, a model of a building represents an actual building which is to be constructed. It is used for purposes of understanding the requirements, and architecture, among other things.

What is modeling?

To construct a plan after a pattern. It is a process of creating a model based on the requirements to help better understand the system being developed.

Why do we do modeling?

It is used to understand a system before it has been developed. It helps in understanding complex problems and scenarios and helps in creating your design as close to the real system as possible before the implementation. Modeling helps develop a final product as close to what we had intended to develop. Thus, if you make any changes after the implementation, it won't break your system design and system itself.

Why use UML?

It is a common modeling language accepted worldwide and is understood by software designers, business analysts, developers, testers, managers etc. It helps understand the business needs, product features, and the architecture of a system.

What is a UML Diagram?

It is a graphical representation used to visualize certain features of a system.

Examples of UML diagrams

5.1. Use case Diagram

This diagram describes how the system responds to an action of a user or actor. It tells us who interacts with the system.

Use Case: All Features

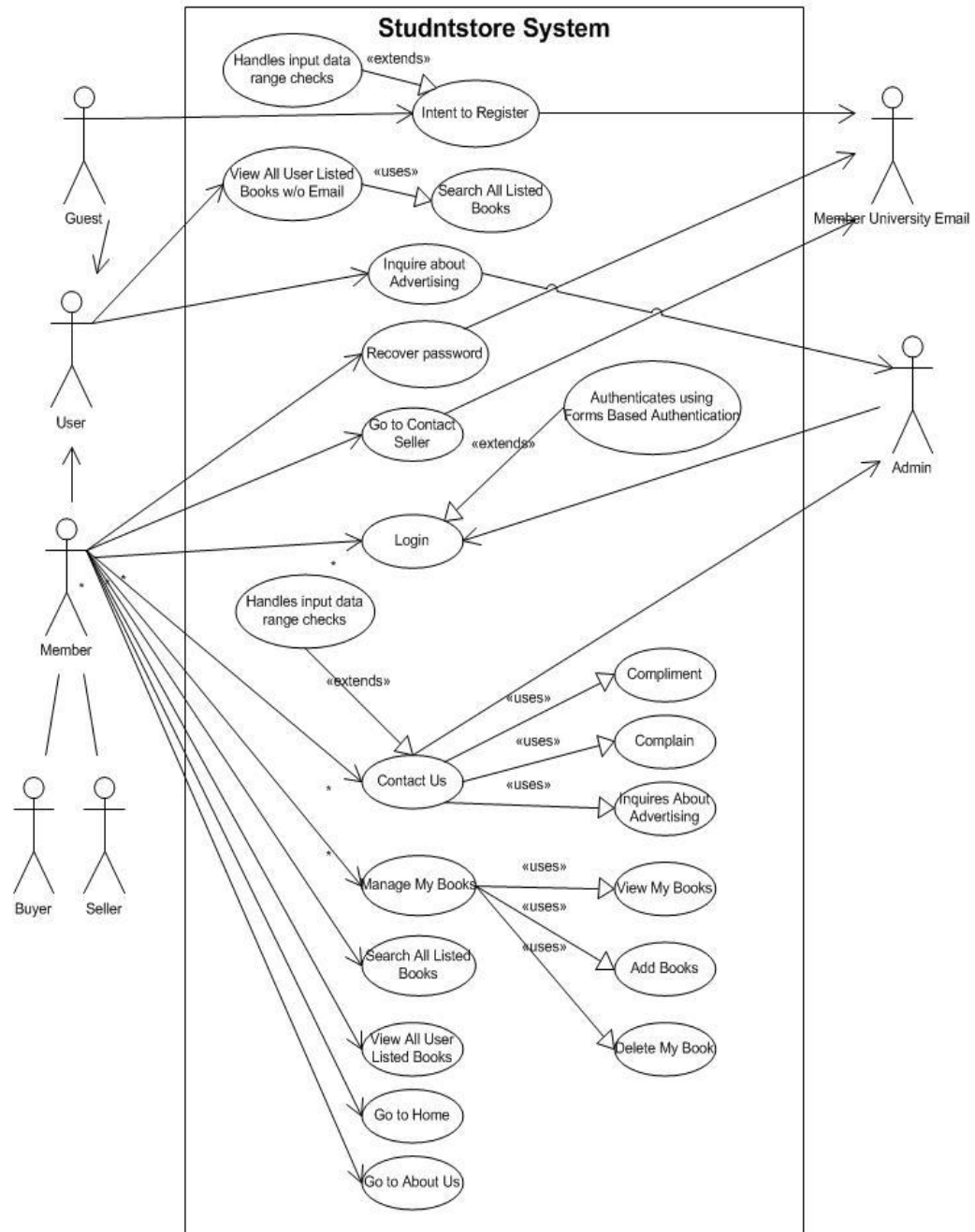


Figure 5.1: Use case - All Features

5.2. Sequence Diagram

It is a type of interaction diagram that describes how the different elements communicate with each other through messages during a process.

Intent to Register

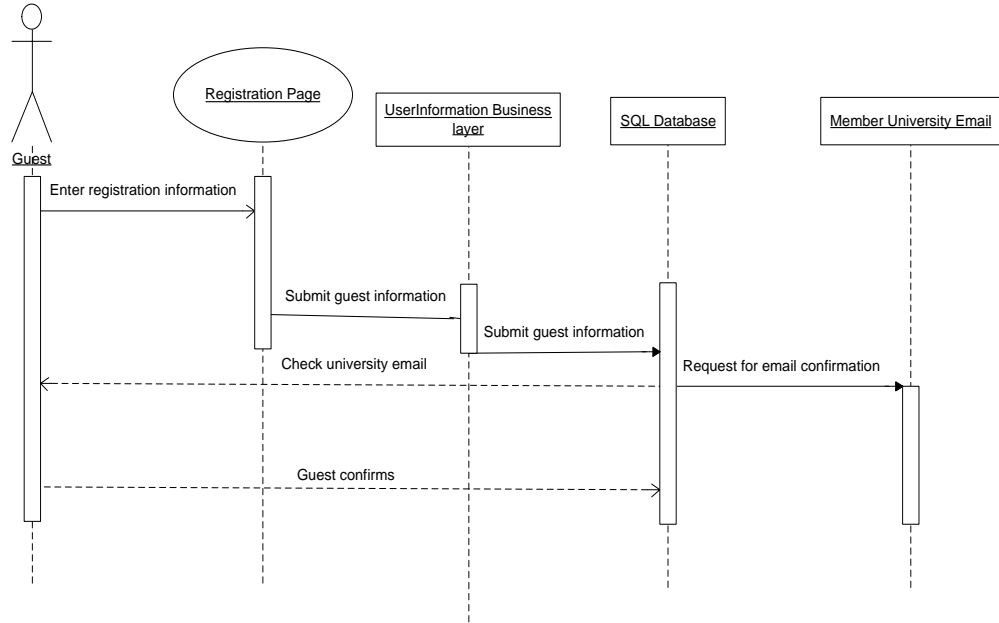


Figure 5.2.1:

Secure Login

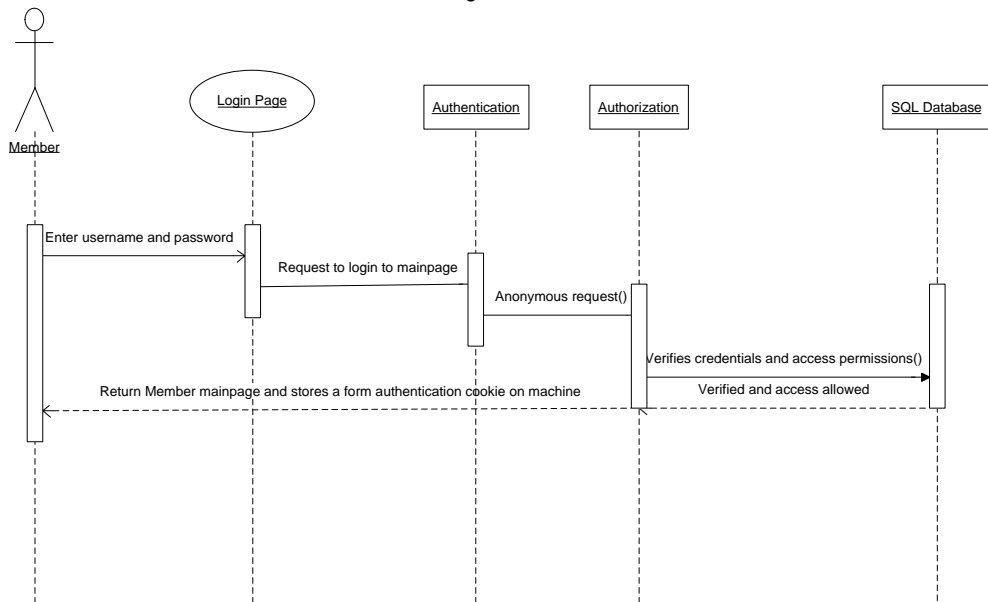


Figure 5.2.2:

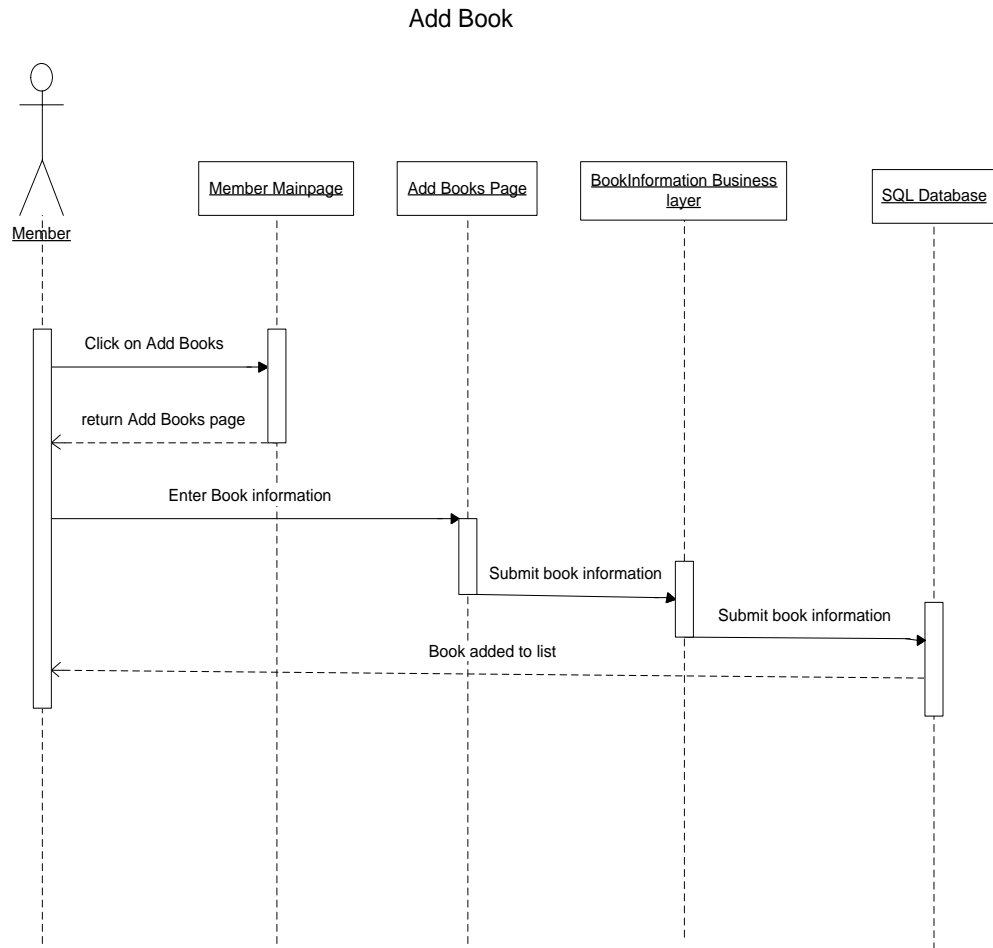


Figure 5.2.3:

5.3. Activity Diagram

This diagram describes the flow of activities during a specific task such as adding of books by a member in a bookstore system.

Activity Diagram – Contact seller

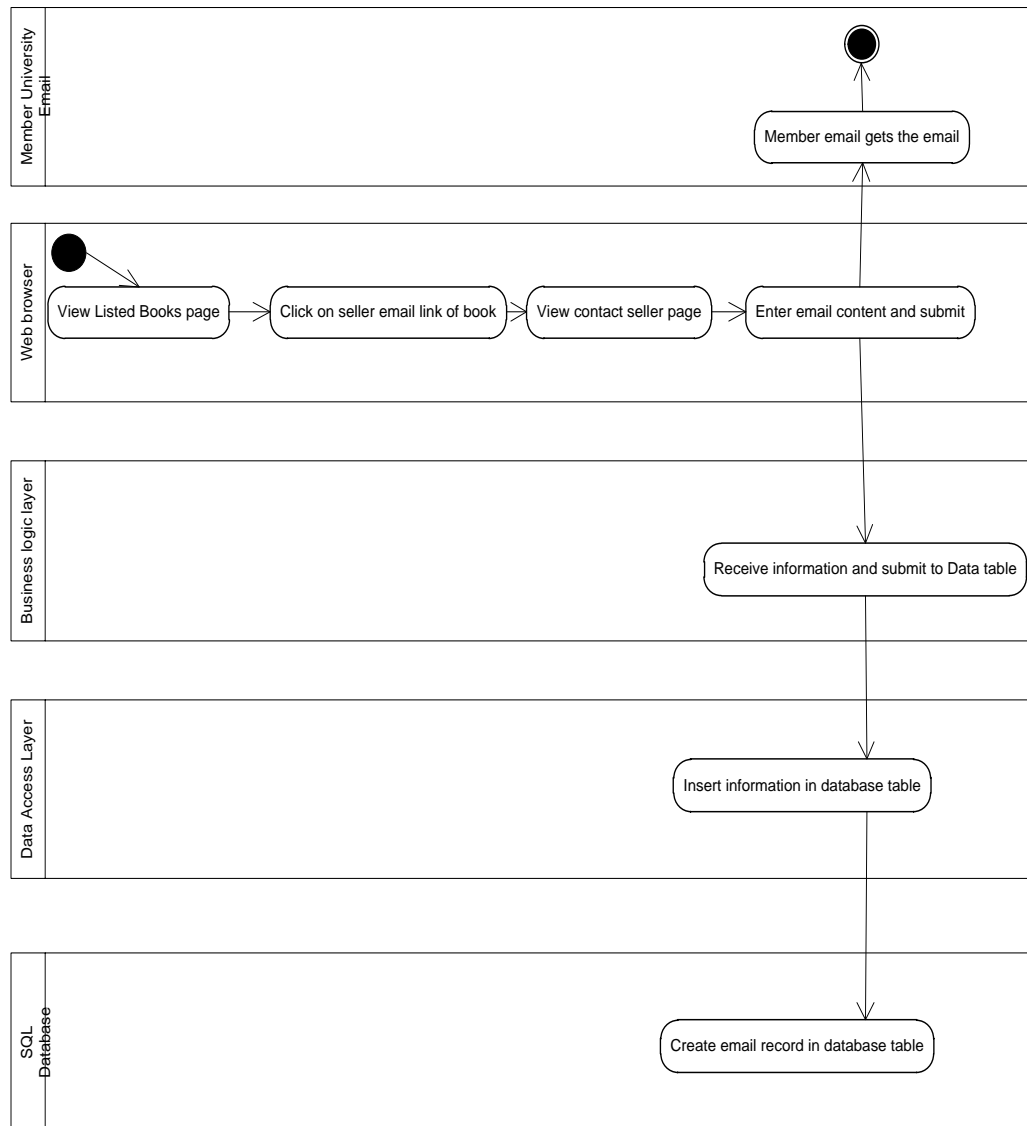


Figure 5.3: Contact Seller Activity Diagram

5.4. Software System Architecture Diagram

This diagram shows the different components or modules or layers of a system and how they interact with each other.

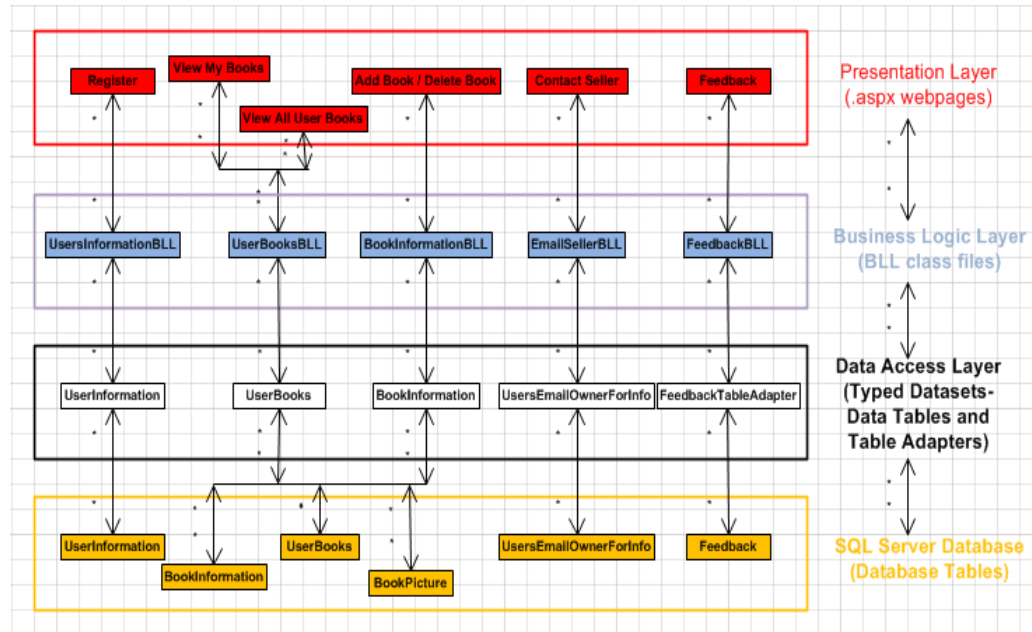


Figure 5.4: Software System Architecture Diagram

5.5. Software System Architecture Detailed Diagram

Software System Architecture Detailed Diagram

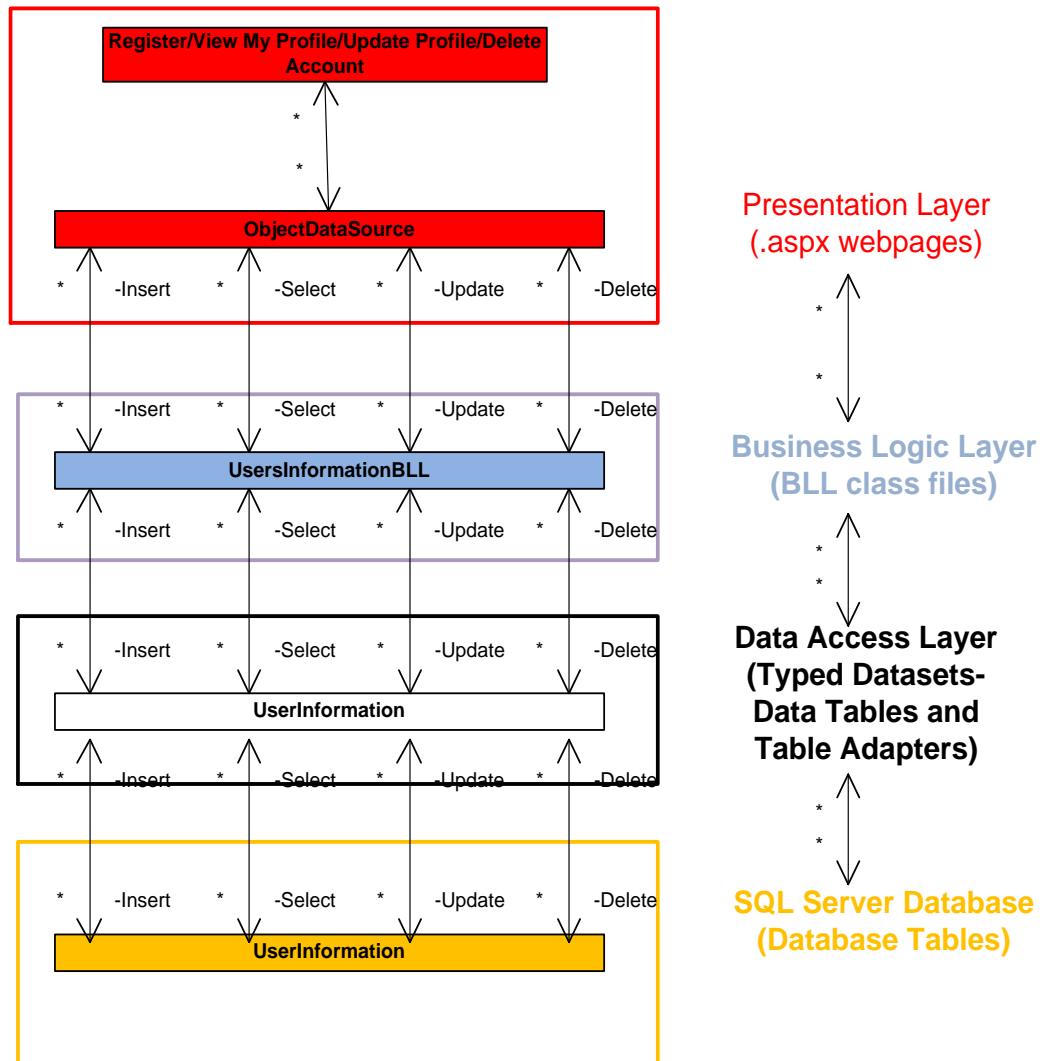


Figure 5.5:

6. Functional Requirement Specifications

Feature 1.

| | |
|----------------------------|---|
| Use-case Name | Intent to Register |
| Actors | System, Guest |
| Brief Description | The Guest intends to register in order to buy or sell his/her book |
| Flow of events | <p>Basic flow begins when a Guest is browsing one of website's general pages and he/she doesn't have a membership account yet.</p> <ol style="list-style-type: none">1. The Guest clicks the Register link or tries to access a Member's only page.2. The System directs the Guest to a webpage with a form to register.3. The Guest enters the information requested in form.4. The Guest clicks the submit button.5. The system shows a message that a new account has been created and asks the Guest to check for a mail in his university email box to verify the email. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">3. a. The Guest doesn't enter correct input.b. The System shows validation error. <p>Go back to step 2.</p> <p>Alternative flow 2.</p> <ol style="list-style-type: none">4. a. The Guest hits the cancel button. <p>Go back to step 2.</p> |
| Pre-conditions | <ol style="list-style-type: none">1. The Guest has access to internet.2. The Guest is accessing one of the website's general pages. |
| Post-conditions | <ol style="list-style-type: none">1. An email containing a password is sent to his |

| | |
|--|---|
| | <p>university email address that he requires to login to the system.</p> <p>2. The SQL database has his information stored.</p> |
|--|---|

Feature 2.

| | |
|----------------------------|---|
| Use-case Name | Secure Login |
| Actors | System, Member-Buyer or Seller |
| Brief Description | This use-case handles access to the Members only area of website. The Member logs on to the website using a username and password with a secure Forms based Authentication. If he isn't a member yet, he will be directed to the registration page. |
| Flow of events | <p>Basic flow begins when a Member is browsing the general webpage on website and wishes to login to his own account in order to buy/sell/manage his account.</p> <ol style="list-style-type: none">1. The Member is browsing one of website general webpages.2. The Member click on Login link or tries to access the Member's only link on website.3. The system will display textboxes to enter username and password, link for password recovery and register.4. The Member enters username and password and clicks Submit.5. The system verifies the information using Forms based Authentication.6. The information entered is authenticated and the system displays the Member Mainpage with the Member specific information. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">6. a. The credentials do not match.b. The system displays a message that username or password is not correct or not found.c. Goes to step 3 about three times before the account is locked for 24 hours. |

| | |
|-----------------|--|
| Pre-conditions | <ol style="list-style-type: none"> 1. The Member has access to the internet. 2. The Member must be browsing one of the website general pages and then be on login page in order to enter the Members only area of website. |
| Post-conditions | <ol style="list-style-type: none"> 1. The Member is now logged into the system. |

Feature 3.

| | |
|----------------------------|---|
| Use-case Name | Recover password |
| Actors | System, Member |
| Brief Description | This use-case is used to recover the password of a Member who tries to login to the system but fails. |
| Flow of events | <p>Basic flow begins when the Member fails to enter correct username and/or password and is shown login page with Forgot password link.</p> <ol style="list-style-type: none">1. The Member clicks on Forgot Password link.2. The Member is redirected to Password recovery page.3. System asks for the Member's email.4. The Member enters email and submits.5. System sends the password to email submitted.6. The Member checks his email for password. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">4. a. The Member enters a non-existing email. b. System shows an error message if email not found in database. |
| Pre-conditions | <ol style="list-style-type: none">1. The Member has access to internet. |
| Post-conditions | The Member password is reset and he/she can now login to the System. |

Feature 4.

| | |
|----------------------------|--|
| Use-case Name | Administrator logs in |
| Actors | System, Administrator(admin) |
| Brief Description | This use-case describes the steps to how an admin can login to the system to have access to all the webpages and be able to edit any information. |
| Flow of events | <p>Basic flow begins when an admin is on the login page where he must enter his admin username and password to get to Admin Mainpage.</p> <ol style="list-style-type: none">1. System shows login page asking for username and password.2. Admin enters admin username and password.3. System authenticates the admin.4. System logs the admin to the Admin Mainpage. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">2. a. System cannot authenticate the admin username and password. b. System shows an error message asking to enter correct username and password. c. Goes to step 1 for three times before the admin account gets locked for 24 hours. |
| Pre-conditions | <ol style="list-style-type: none">1. Admin has access to the internet.2. Admin must be on login page. |
| Post-conditions | Admin is logged on to the Admin Mainpage. |

Feature 5.

| | |
|----------------------------|--|
| Use-case Name | Add books |
| Actors | System, Member |
| Brief Description | This use-case describes how a Member adds a book to his/her inventory in order to sell and so it also gets added to the general list of books for sale. |
| Flow of events | <p>Basic flow begins when a Member is logged on to his account in the system and clicks on Add books link</p> <ol style="list-style-type: none">1. The Member clicks on Add Books link on his Mainpage.2. The system shows a webpage with a form asking for book details.3. The Member fills in the form with book information and clicks Submit.4. System stores the book information in the database and lists the books in My Books list and All User Listed Books list. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">3. a. The Member fills in book information and clicks Cancel.b. Go to Step 2. |
| Pre-conditions | <ol style="list-style-type: none">1. The Member has access to internet.2. The Member is logged in the system. |
| Post-conditions | Book information is stored in database and the book gets listed. |

Feature 6.

| | |
|----------------------------|---|
| Use-case Name | Delete my book |
| Actors | System, Member |
| Brief Description | This use-case explains how a Member can delete a listed book from his/her My Books list in his/her account. |
| Flow of events | <p>Basic flow begins when a Member is logged on his account mainpage in the system and wants to delete one of his listed books.</p> <ol style="list-style-type: none">1. The Member clicks on the My Books link.2. The system shows a list of his books for sale.3. The Member chooses a book he wishes to delete from the list and clicks the delete link next to that book.4. The system deletes the information related to that book from the database.5. The system shows a new list of My Books without that book. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">1. a. The system has no books in the list for sale. b. The system shows a message “No Books”. |
| Pre-conditions | <ol style="list-style-type: none">1. The Member has access to internet.2. The Member is logged in the system. |
| Post-conditions | The book selected by the Member is deleted from the database by the system. |

Feature 7.

| | |
|----------------------------|---|
| Use-case Name | View My Books |
| Actors | System, Member |
| Brief Description | This use-case is used by a Member to view his/her books listed for sale. |
| Flow of events | <p>Basic flow begins when a Member wishes to see a list of his books for sale.</p> <ol style="list-style-type: none">1. The Member clicks on View My Books link.2. The system shows a list of his books and its information entered by user. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">2. a. The system shows a blank page because the user hasn't added any books and shows a message "No Books". |
| Pre-conditions | <ol style="list-style-type: none">1. The Member has access to internet.2. The Member is logged in the system. |

Feature 8.

| | |
|----------------------------|---|
| Use-case Name | View all User Listed Books |
| Actors | System, Member |
| Brief Description | This use-case is used by a Member to view all the Members books listed for sale. |
| Flow of events | <p>Basic flow begins when a Member wishes to see a list of all Member listed books for sale.</p> <ol style="list-style-type: none">1. The Member clicks on View All User Listed Books link.2. The system shows a list of books and their information entered by all the Members. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">2. a. System doesn't show a list if none of the members have listed a book for sale. Instead, displays a message "No Books" |
| Pre-conditions | <ol style="list-style-type: none">1. The Member has access to internet.2. The Member is logged in the system. |

Feature 9.

| | |
|-------------------|---|
| Use-case Name | Search feature for listed books |
| Actors | System, Member |
| Brief Description | This use-case describes how a Member can search for a specific book from a list of all user listed books using search categories of author, publisher and bookname. |
| Flow of events | <p>Basic flow begins when a Member wishes to find a particular book from a list of all member listed books.</p> <ol style="list-style-type: none">1. The Member clicks on View all User Listed books.2. The system shows a list of all the Member listed books.3. The Member selects a category of search from Search dropdown menu.4. The Member enters the text related to that category and clicks Submit.5. The System searches for a match in database.6. The System returns a list of matching books related to search category. |
| Pre-conditions | <ol style="list-style-type: none">1. The Member has access to internet.2. The Member is logged in the system. |

Feature 10.

| | |
|----------------------------|--|
| Use-case Name | Contact seller |
| Actors | System, Member - buyer, seller |
| Brief Description | The use-case shows how a buyer interested in a book can contact the seller of that book through the system. |
| Flow of events | <p>Basic flow begins when a buyer intends to meet the seller or ask for more information related to the book.</p> <ol style="list-style-type: none">1. The buyer has a book, of his interest, listed in View All User books list.2. Buyer clicks on Email Seller link inside the row of that related book.3. The System redirects buyer to a webpage asking to enter the content in the body of letter that will be send to the seller.4. Buyer fills in form and clicks Submit.5. The System stores a copy of that email in the database and sends the email to the seller's email address. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">3. a. The buyer fills in the email body and then clicks Cancel.b. Go to step 3. |
| Pre-conditions | <ol style="list-style-type: none">1. The Member has access to internet.2. The Member is logged in the system. |
| Post-conditions | The system stores an email copy in the database. |

Feature 11.

| | |
|-------------------|--|
| Use-case Name | Go to Home |
| Actors | System, User – Guest or Member |
| Brief Description | This use-case explains how a user can access the Home page to read about the purpose of the system |
| Flow of events | <p>Basic flow begins when a user first enters the website name in browser or is on any other webpage and clicks on Home page link</p> <ol style="list-style-type: none">1. The user clicks on Home link.2. The system shows the home page to the user |
| Pre-conditions | <ol style="list-style-type: none">1. User has access to internet.2. User enters the website name on the browser or is already browsing a webpage of the website. |

Feature 12.

| | |
|-------------------|--|
| Use-case Name | Go to About us |
| Actors | System, User – Guest or Member |
| Brief Description | This use-case explains how a user can access the About us page to read about the owners who created this system. |
| Flow of events | <p>Basic flow begins when a user first enters the website name in browser or is on any other webpage and clicks on the About Us link.</p> <ol style="list-style-type: none">1. The user clicks on About Us link.2. The system shows the About Us page to the user |
| Pre-conditions | <ol style="list-style-type: none">1. User has access to internet.2. User enters the website name on the browser or is already browsing a webpage of the website. |

Feature 13.

| | |
|----------------------------|--|
| Use-case Name | Go to Contact us |
| Actors | System, User – Guest or Member |
| Brief Description | This use-case explains how a user can access the Contact Us page in order to complain, complement, and comment about the system or any other issue to the site administrator (admin). |
| Flow of events | <p>Basic flow begins when a user first enters the website name in browser or is on any other webpage and clicks on Contact Us link</p> <ol style="list-style-type: none">1. The user clicks on Contact Us link.2. The system shows the Contact page with a form to the user.3. The user fills out the form, clicks Submit.4. The System notifies the admin. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">3. a. User clicks the Cancel button. b. Go to step 2. |
| Pre-conditions | <ol style="list-style-type: none">1. User has access to internet.2. User has entered the website name in the browser and clicks Enter or is already browsing a webpage of the website. |
| Post-conditions | The system stores the Feedback form information in database. |

Feature 14.

| | |
|-------------------|--|
| Use-case Name | View Listed Books and search feature for any User |
| Actors | System, User – Guest or Member |
| Brief Description | This use-case describes how a user can see all Member listed books and search for a specific book from that list. |
| Flow of events | <p>Basic flow begins when a user wishes see all member listed books and find a particular book he wishes to buy from that list.</p> <ol style="list-style-type: none">1. User clicks on View all User Listed books.2. The system shows a list of all The Members listed books.3. User selects a category of search from Search dropdown menu.4. User enters the text related to that category and clicks Submit.5. System searches for a match in database.6. System returns a list of matching books related to search category. |
| Pre-conditions | <ol style="list-style-type: none">1. User has access to internet.2. User is browsing the View Listed Books general webpage of system |

Feature 15.

| | |
|----------------------------|--|
| Use-case Name | Inquire about Advertising |
| Actors | System, User- Member or Guest, Admin |
| Brief Description | This use-case describes how a user can contact the admin in order to get information related to putting up ads on the website. |
| Flow of events | <p>Basic flow begins when a user intends to advertise something on website and needs related information.</p> <ol style="list-style-type: none">1. User is browsing the website.2. User clicks on advertisement area which shows the message “Do you want to advertise your product here, then Click here”3. The system redirects the user to Contact Us page.4. The user fills the form and clicks Submit.5. The system stores the form information in database and notifies the Admin. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">4. a. The user clicks Cancel.b. Go to Step 3. |
| Pre-conditions | <ol style="list-style-type: none">1. User has access to internet.2. User is browsing the system webpages. |
| Post-conditions | The system stores the user inquiry in database. |

Feature 16.

| | |
|----------------------------|--|
| Use-case Name | Browse CSUN map on Google maps |
| Actors | System, Member- Buyer or Seller |
| Brief Description | This use-case explains how a member can use the Google map of CSUN in order to decide on a place to meet with seller. |
| Flow of events | <p>Basic flow begins when a member wishes to meet with another member to exchange books and money.</p> <ol style="list-style-type: none">1. The Member clicks on CSUN Maps link.2. The system shows a CSUN campus Map on Google maps. |
| Alternative flow of events | None |
| Pre-conditions | <ol style="list-style-type: none">1. The Member has access to internet.2. The Member is logged in the system. |

Feature 17.

| | |
|----------------------------|--|
| Use-case Name | Live help chat |
| Actors | System; User- Guest or Member, |
| Brief Description | This use-case describes how a user can use live help chat service in order to ask any questions related to website and its function. |
| Flow of events | <p>Basic flow begins when a user is browsing the website and wishes to get help in real time.</p> <ol style="list-style-type: none">1. User clicks on live chat link on website.2. The system connects the user to live chat.3. The user can now ask any questions in live chat and gets help. |
| Alternative flow of events | <p>Alternative flow 1.</p> <ol style="list-style-type: none">2. a. The system cannot connect user to live chat because it is not a business hour. b. The system shows a message asking user to try again during business hours. |
| Pre-conditions | <ol style="list-style-type: none">1. The user has access to internet2. The user is browsing the website. |


Feature 18.

| | |
|----------------------------|--|
| Use-case Name | Go to Frequently Asked Questions(FAQ) |
| Actors | System, User- Guest or Member |
| Brief Description | This use-case describes how a user can access a list of frequently asked questions. |
| Flow of events | <p>Basic flow begins when a user is browsing through the website looking for an answer to a specific question or just visits the page out of curiosity.</p> <ol style="list-style-type: none">1. The user clicks on FAQ link.2. The system shows the FAQ page to the user.3. The user can now read through the list of frequently asked questions. |
| Alternative flow of events | None |
| Pre-conditions | <ol style="list-style-type: none">1. The user has access to internet.2. The user is browsing the website. |

7. Database Artifacts and Table Design



I have used SQL Server 2008 to create a relational database for this application. The database stores all data of this application. The hosting servers of companies such as Godaddy provide SSL encryption. This will ensure secured exchange of information between the user and the system using the internet.

7.1. UserInformation Table

| | Column Name | Data Type | Allow Nulls |
|---|-----------------|--------------|-------------------------------------|
|  | UserID | int | <input type="checkbox"/> |
| | FirstName | nvarchar(50) | <input type="checkbox"/> |
| | LastName | nvarchar(50) | <input type="checkbox"/> |
| | Username | nvarchar(20) | <input type="checkbox"/> |
| | UserPwd | nvarchar(40) | <input type="checkbox"/> |
| | Gender | nvarchar(50) | <input type="checkbox"/> |
| | BirthDate | datetime | <input type="checkbox"/> |
| | Phone | nvarchar(25) | <input type="checkbox"/> |
| | Email | nvarchar(50) | <input type="checkbox"/> |
| | UniversityName | nvarchar(80) | <input type="checkbox"/> |
| | UniLocationAdd1 | nvarchar(60) | <input type="checkbox"/> |
| | UniLocationAdd2 | nvarchar(60) | <input checked="" type="checkbox"/> |
| | City | nvarchar(30) | <input type="checkbox"/> |
| | CState | nvarchar(50) | <input type="checkbox"/> |
| | Zip | nvarchar(15) | <input type="checkbox"/> |


This table stores all the information of a member. UserID is the primary key which increments automatically. Only one column UniLocationAdd2, which provides additional space for university address, has been allowed to be null.

7.2. UserBooks Table

| | Column Name | Data Type | Allow Nulls |
|---|-------------|---------------|-------------------------------------|
|  | UserID | int | <input type="checkbox"/> |
|  | BookID | int | <input type="checkbox"/> |
| | PictureID | int | <input checked="" type="checkbox"/> |
| | Condition | nvarchar(200) | <input type="checkbox"/> |
| | Price | int | <input type="checkbox"/> |


This table keeps a record of a list of books and their condition, picture and price. It has foreign keys UserID and BookID that together make up a unique key. PictureID is a foreign key of the BookPicture Table.

7.3. BookInformation Table

| | Column Name | Data Type | Allow Nulls |
|---|-------------|---------------|--------------------------|
|  | BookID | int | <input type="checkbox"/> |
| | BookName | nvarchar(200) | <input type="checkbox"/> |
| | ISBN | nvarchar(200) | <input type="checkbox"/> |
| | Author | nvarchar(200) | <input type="checkbox"/> |
| | Publisher | nvarchar(200) | <input type="checkbox"/> |


This table stores all the general information of a book. Each row can be accessed using the BookID as the primary key.

7.4. BookPicture Table

| | Column Name | Data Type | Allow Nulls |
|---|-------------|---------------|-------------------------------------|
|  | PictureID | int | <input type="checkbox"/> |
| | PicType | nchar(50) | <input checked="" type="checkbox"/> |
| | PicPath | nvarchar(500) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

This table stores the picture of a particular book listed by a particular seller. It gives a PictureID to each picture that is stored. It also keeps a record of the server location of the picture and the picture type.

7.5. UsersEmailOwnerForInfo Table

| | Column Name | Data Type | Allow Nulls |
|---|-------------|----------------|-------------------------------------|
|  | EmailID | int | <input type="checkbox"/> |
| | [From] | int | <input type="checkbox"/> |
| | [To] | int | <input type="checkbox"/> |
| | Date | datetime | <input type="checkbox"/> |
| | Subject | nvarchar(50) | <input type="checkbox"/> |
| | Location | nvarchar(100) | <input checked="" type="checkbox"/> |
| | Body | nvarchar(1000) | <input type="checkbox"/> |

Whenever a member fills in the form to contact the seller of a particular book, the information that is submitted to the database gets stored in this table. The EmailID is the primary key here.

7.6. Feedback Table

| | Column Name | Data Type | Allow Nulls |
|--|----------------|----------------|-------------------------------------|
| | FeedbackID | int | <input type="checkbox"/> |
| | FullName | nvarchar(50) | <input type="checkbox"/> |
| | Email | nvarchar(50) | <input type="checkbox"/> |
| | Phone | nvarchar(50) | <input checked="" type="checkbox"/> |
| | TypeOfFeedback | nvarchar(50) | <input type="checkbox"/> |
| | Feedback | nvarchar(1000) | <input type="checkbox"/> |

This table stores the feedback submitted by the users.

8. Software Architecture

8.1. Some types of Application Architecture

System architecture or application architecture is an integral part of an application design. It determines the number of pieces of an application, the way they interact with each other and the function performed by each piece. Three-Tiered architecture is better suited for the Studntstore application. I have compared Three-Tiered architecture with other types of tiered architectures and ASP.NET MVC architecture to emphasize the benefits of Three-Tiered architecture.

8.1.1. Single Tier or Monolithic architecture

In this tier, the user interface, business logic and data access code are all part of a single layer. There is no modularity. An example is a word processor.

Disadvantages

1. Inflexible - A minor change can lead to major unexpected results and problems arising out of that. These new problems are extremely hard to understand and fix.
2. High costs of maintenance – It requires a large number of programmers to test the functionality of the application to make sure everything works fine as before after even a very small change in the application.
3. The application or the code cannot be reused while building other applications.

8.1.2. Two Tier architecture

In this architecture, the business rules are a part of either the user interface layer or the database layer.

Advantages

1. You can find all your code, at one place, in the forms. Thus, you do not have to look at the code in different layers.
2. Development and coding becomes easier and faster as you do not have to create separate layers and make them interact with each other. You can use ADO.NET in your forms to directly access the database. You can complete coding an application within a short time compared to time required for implementing a three-tier architecture.

Disadvantages

1. It is very difficult to change the business rules of the application in all the clients. If one rule is changed in one client, it has to be updated in all clients which can be very tedious.
2. Business rules will occur repeatedly in different forms. If you change the name of a table or a field, it has to happen in all the forms. Thus, there is a repetition of business rules.
3. It is very difficult to manage a version control and distribute new versions of software as the application logic resides on client.
4. If you change the source of the data you used to fill a dataset, changing the code that helps to load that data will be extremely tough. For example, if you are now using a .csv file instead of a previously used SQL database, then access methods will change. You have to change that manually in all user interfaces where you have used this dataset.
5. A user might need a separate password for accessing each SQL server thus making the system security complicated. This results in possible corruption of database files if the user has access to the database using the same password.
6. A lot of data table field names, and values, and other field names and values are hard coded in the source code. So, if you are changing the values, you have to do that manually.
7. The application or the code cannot be reused while building other applications.

8.1.3. Three Tier architecture

It is a type of client server architecture with modularity. It consists of three or more layers that interact with each other through interfaces. Any modification in the internal working of one layer has no effect on the other layers. It is the most widely accepted and implemented architecture for a business application or software today. It consists of following layers –

1. Presentation Layer

The main purpose of this layer is to interact with the user. It consists of User Interface that uses .aspx form pages. The forms contain fields and user controls used to obtain input from the user and display output to the user through a browser such as internet explorer or Google chrome. The pages have a consistent layout through the use of master pages. The data access logic has been separated from the presentation layer. The presentation layer does not have any database connection strings, SQL queries or any other database related code. The presentation layer cannot interact with the database

directly. It calls the methods in the class files of business layer using `SqlDataSource` or `ObjectDataSource`.

2. Business Logic Layer

The business layer acts as the middle tier between the presentation layer and the data access layer. It contains all the custom business logic of the application such as business rules for setting access rights and permissions for each user depending on his role in the application. If any calculations need to be performed, then it is better to implement it in a method in the business layer class so that it can be reused anywhere else. This layer calls the methods of the data access layer.

3. Data Access Layer(DAL)

The data access layer consists of all the code that is used to access data stores such as a database. The primary .NET technology used here is ADO.NET. This layer consists of typed datasets that uses Data tables and Table adapters. It invokes the different SQL queries of insert, select, update and delete and makes changes in the database. No other layer can interact with the database. Thus, Data Access Layer helps access data from the database. With the use of a DAL, you can change the schema of a database without any dependencies on the database code.

8.2. Why use Three Tier architecture?

1. It doesn't depend on any implementation of the User Interface.

- 2. Scalability and Maintainability**

Each layer can be placed in different servers which can interact with each other. The layers can be implemented using a different programming language, platform or technology or hardware specifications.

If there are any changes to be made or bugs and errors in one layer, they can be isolated and fixed with less effort and without affecting the other layers in the application. Thus, it is easy to maintain such an application.

- 3. Flexibility**

Since you have different layers, you can host them on different physical servers. Thus, it makes the application more flexible.

4. The database can be easily replaced with a different type of database.
5. There is low coupling and high cohesion between the different layers of a three tier architecture. Also, you can replace an implementation of an interface of a layer with a new one with ease. This makes it easier to maintain and enhance the solution.

6. Each layer can be reused in another application or solution which might have need for a layer with the similar functions.
7. Each layer has an interface with a specific input and output decided during design. Layers are independent and interact with each other through the interfaces. So, the layers can be developed in parallel by independent teams anywhere across the globe.
8. The performance and fault tolerance of the application also increases.
9. Each layer can be updated, maintained and deployed at different time schedules.
10. Because of layer structure, it is possible to expose only some layers to external world and place other layers behind a firewall. Thus, making a secured and accessible application.
11. The testing of the application is easier because the layers divide the application into components, each with specific inputs and outputs. Each layer can be tested individually for its functionality without having to depend on the completion of the other layers. If one layer takes a long time in execution, it can be replaced with a different implementation method such that the new layer takes lesser time. Thus, the test cases can be quickly executed. Thus, it is easy to test such a system whose layer interfaces are well-defined.
12. Using typed datasets helps look up column names using Visual Studio's IntelliSense. So, you do not have to remember them.
13. A minor change in Data Access Layer doesn't require redistribution of the application to the clients.
14. A specific query can be reused by different forms by calling methods in the lower layers to get the results of the query. Similarly, a specific business rule might be used for accessing different forms. So, there is less repetition of code.

With all these advantages, we can conclude that a three tier application can be easily migrated to newer platforms and development environments.

8.3. What are the disadvantages of using a Three Tier architecture?

1. Using layers in simple small applications makes them more complicated.
2. It takes a longer time to design and develop an application with a three tier compared to one with a single layered architecture.
3. Traceability of how the data flows between the tiers becomes a difficult task with larger complex applications.
4. It takes longer for a new developer to understand and help in maintenance of an already built application because of the complexity and interdependency of the layers in achieving the functionality of the application as a whole.
5. The performance of an application with layer methods calling each other is adversely affected if not tested and kept in check during the implementation.

8.4. Three tier architecture in the Studntstore Application

The Studntstore application uses a three-tier architecture. Let's go into the depth of each layer and how they work.

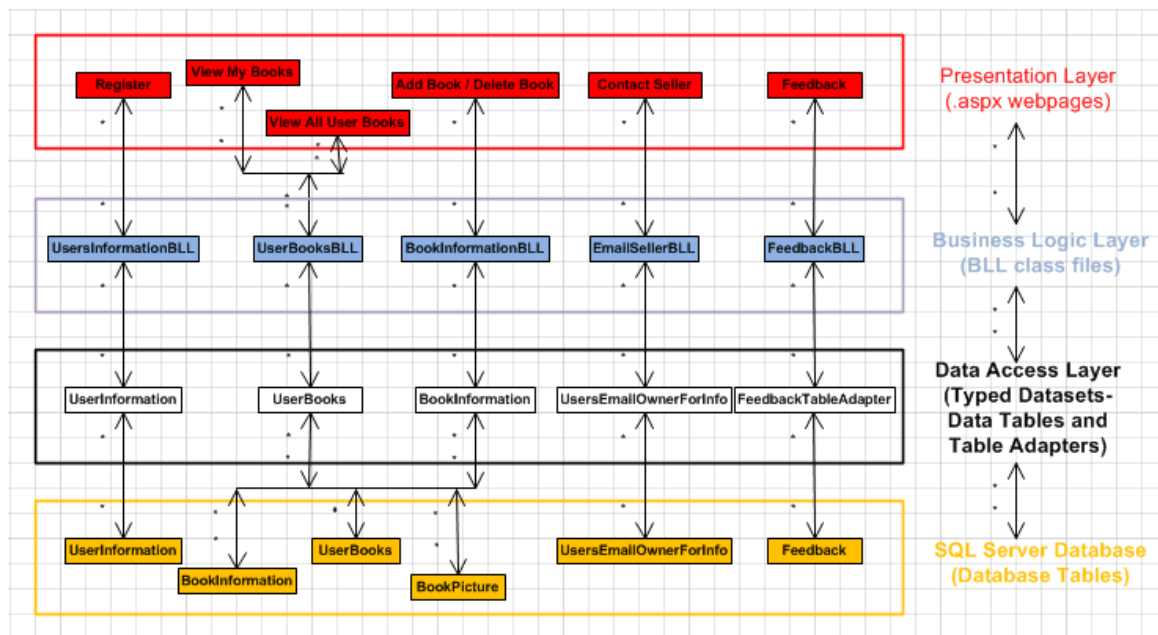


Figure 8.4: Software System Architecture diagram

1. Presentation Layer

This layer consists of ASP.NET pages to display data to the user and obtain input from them. The pages make use of ASP.NET web controls such as textbox, dropdown, GridView, DetailsView, Formview, RadioButtonList etc. I have used GridView and Formview to display the list of member books to the members. We have to use data source controls to bind data to the web controls. Some of the built in ASP.NET data source controls are SqlDataSource, AccessDataSource, ObjectDataSource, XmlDataSource, and SiteMapDataSource. You can also build your own custom data source controls.

I have used ObjectDataSource in the application. First, we map the Select, Insert, Update and Delete methods of the underlying object with that of the ObjectDataSource. Next, we bind the ObjectDataSource to any web control. The web control needs to support the Select, Insert, Update and Delete method in order to make calls to the corresponding methods of the ObjectDataSource which in turn calls the respective methods of the underlying object.

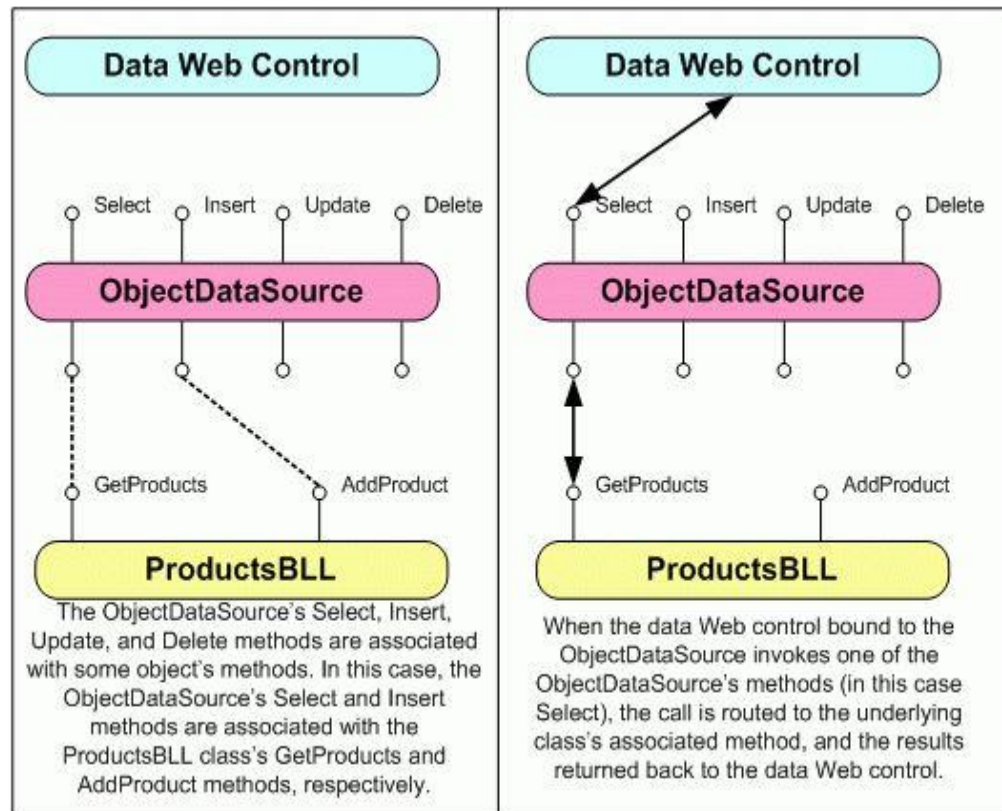


Figure 8.4.1: ObjectDataSource diagram from MSDN website

I will briefly describe the web controls **GridView**, **DetailsView** and **Formview** that I have used in my webpages.

GridView is used to display the data returned by ObjectDataSource as an HTML<table> with a number of rows, each row displaying one record out of a number of records in the table.

DetailsView is used in place of GridView when you wish to show just one record at a time on a page. It is a GridView with just a single record that is rotated 90 degrees. It is rendered as an HTML <table> with two columns and one row for each column or property bound to the control.

Both GridView and DetailsView are rigid in the way they display the data. Formview is a flexible web control which uses templates that combine web controls, static html and databinding syntax. Formview shows only one record at a time. We can enable paging in all these control to see multiple records one by one on each page.

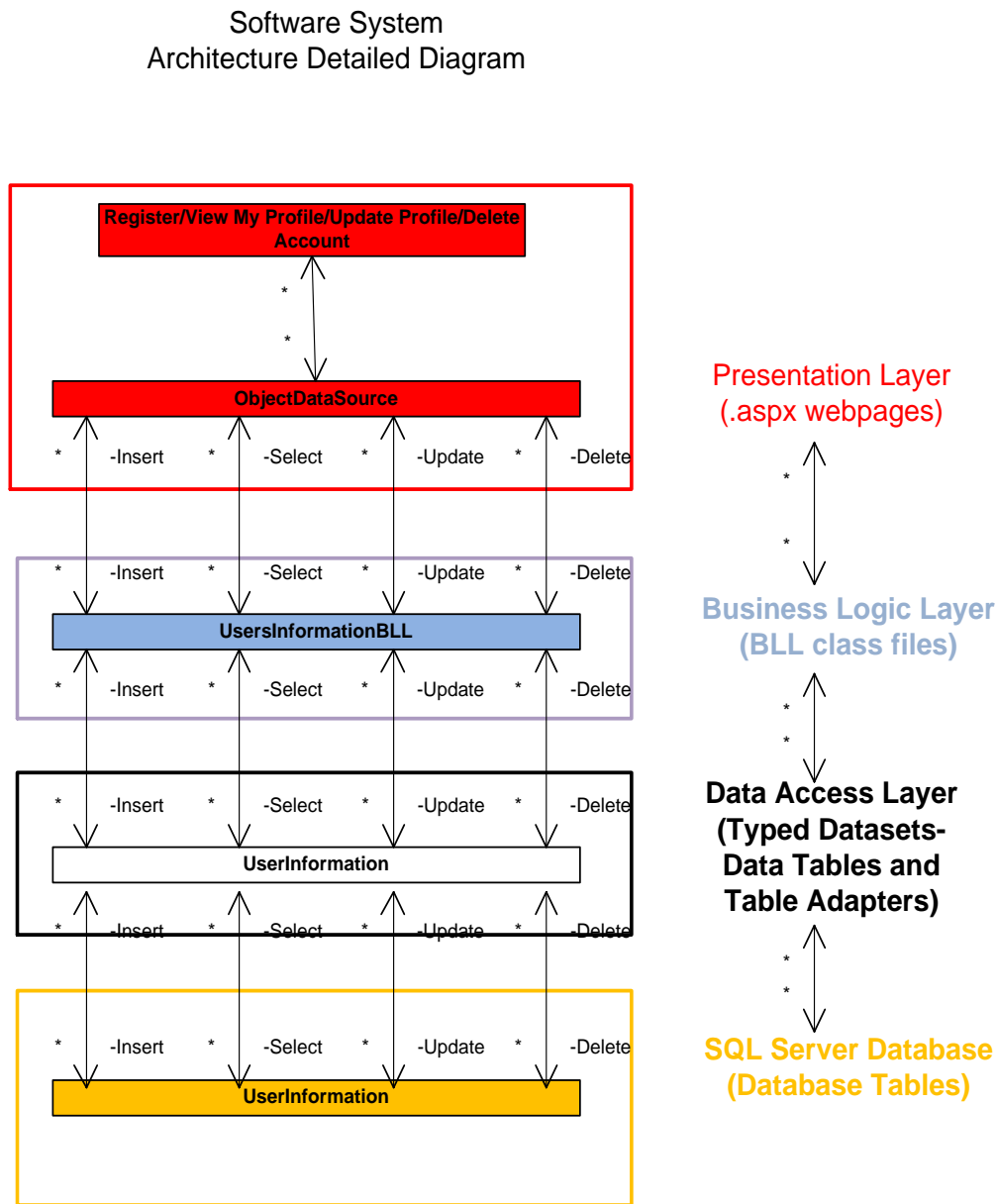


Figure 8.4.2: Software System Architecture Detailed diagram

2. Business Logic Layer(BLL)

This layer contains all the business access logic of the application. For example, if we want to allow users with specific roles to access only certain webpages or see certain data, we specify the rules for that requirement in this layer. Another case where we wish to highlight a certain column in the table if the value goes below a threshold value, we specify that here as well. With this approach, we do not have to depend on a specific User Interface. Thus, we can implement the UI using a new technology. Also, if we wish to change the database to Oracle or MS Access, we can do so without having to depend on just one type such as MS SQL. We write different Select, Insert, Update, and Delete methods in the business class and map it to the ObjectDataSource. The methods in the business class make calls to the DAL methods.

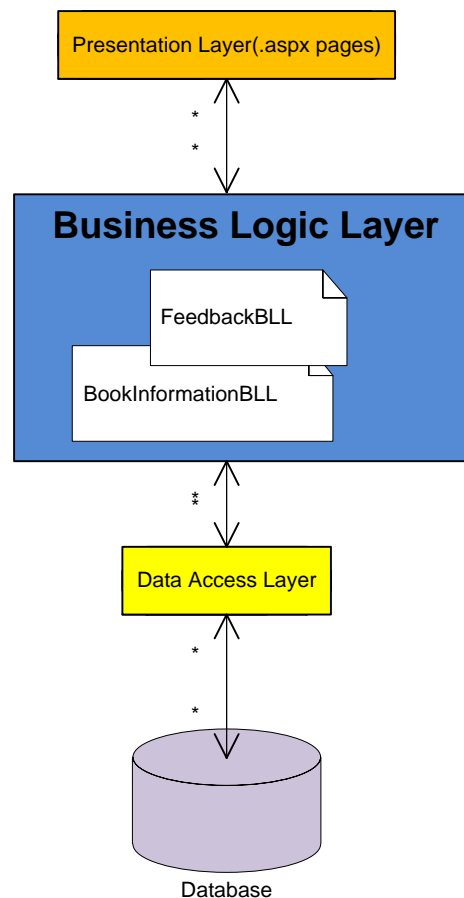


Figure 8.4.3: Business Logic Layer

3. Data Access Layer(DAL)

DAL keeps the data access logic separate from the Presentation layer. It contains all the database specific code such as database connection strings, INSERT, UPDATE, SELECT and DELETE command statements.

The DAL consists of **typed Datasets**. Typed Datasets are strongly typed objects (collection of data) whose schema is already known before runtime as they are rigidly defined at compile time. They are classes generated by Visual Studio based on a database schema and consist of classes such as **DataTable** and **TableAdapters**. TableAdapters are classes that have methods which populate the DataTables and communicate the changes made within the DataTables back to the database. Each DataTable consists of strongly typed collection of DataRow Instances. We can access the column of a DataTable using a command such as `DataTable.Rows[index].columnName`.

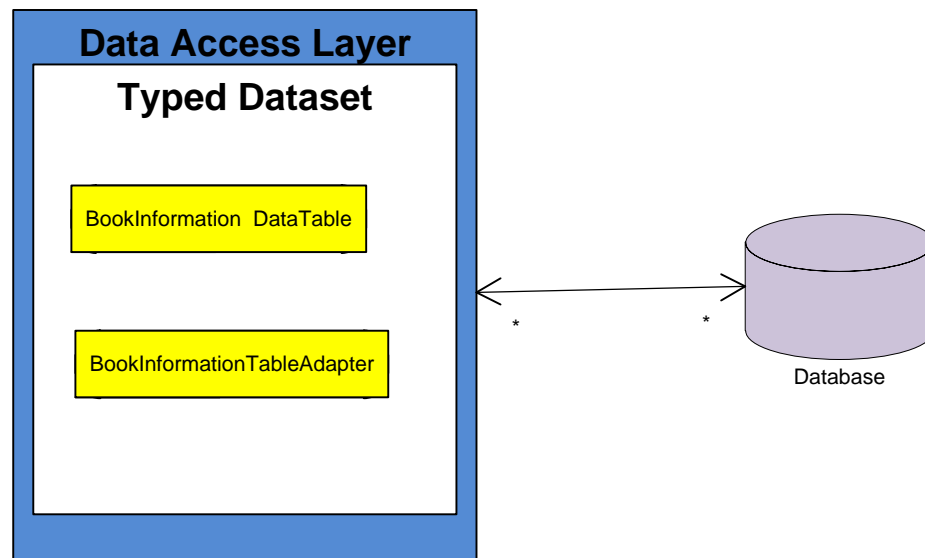


Figure 8.4.4: Data Access Layer

8.5. What are some other architectures?

8.5.1. ASP.NET Model-View-Controller(MVC) Architecture

ASP.NET MVC is an open source web development framework, built on core ASP.NET platform, provided by Microsoft. It is a complete alternative to ASP.NET C# Web Forms and three tier architecture. MVC

pattern was first introduced in 1978 but its regained popularity recently for web development. MVC framework is built as a series of independent components.

In this architecture, the controller class and domain model class work together to handle the http request. The domain model has the business logic such as business objects and rules. The http request first goes to an action method in a controller class. The action method contains the application logic and processes the data input by the user. The Controller class also controls the flow in the process with its different action methods and at the end produces the result to the user by sending a presentation model, with model objects, to the selected view. Views are ASP.NET Web Forms pages (.aspx templates with no code- behind class files and ViewState/postback). The View class converts the presentation model into HTML code to display the output to the user. The View contains only the presentation logic to keep it simple. You can write unit tests for model, view and controller classes as they are not strongly coupled with each other.

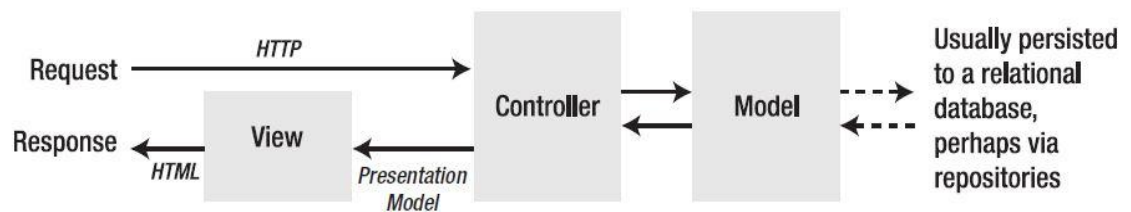


Figure 8.4.5: MVC Architecture for the web - Diagram used from Pro asp.net MVC by Steven Sanderson

8.5.1.1. Advantages of ASP.NET MVC

1. It provides a tight control over http and html. MVC pages do not contain any ViewState data, so the size of pages is smaller than ASP.NET Web Forms.
2. It supports unit testing because of its component-oriented design. ASP.NET MVC applications can be tested using UI Automation testing tools.
3. Powerful routing system – It helps the use of clean urls in code using System.Web.Routing facility.
4. You can use the ready-made asp.net platform features such as Master pages, Forms Authentication, roles, memberships, profiles, also certain Web Forms built in server-controls and thus, develop an application in a short time.

8.5.1.2. Disadvantages of ASP.NET MVC

1. MVC architecture makes a small application too complex and costly to implement. It is better suited for medium to large applications.
2. It takes a longer time to design and develop it as it requires a very thorough understanding of the architecture. Thus, it results in a waste of time for a less experienced developer.
3. ASP.NET Web Forms have an easier event model compared to MVC.

8.6. ASP.NET Web Forms vs. ASP.NET MVC

Web Forms are better suited for small and medium applications that deal with getting database tables from the database and binding them to the GridView control, don't care about the bandwidth issues related to ViewState, do not require regular maintenance and don't have files that need to be downloaded by the user. Web Forms have drag and drop development strengths as well.

MVC is better suited for building large applications that require regular maintenance, stability, search engine optimization, faster download speeds and automated testing. MVC applications have a longer lifespan in that case.

8.7. Three tier architecture vs. ASP.NET MVC architecture

In applications using ASP.NET Web Forms with a three-tier architecture; as the application grows, and it often does, the application logic and behaviors written in the event handlers of the UI make the business logic layer and UI tightly coupled. This makes the automated unit testing of User Interface very difficult.

9. Test Driven Development Software Methodology

It is a software development practice in which a developer writes unit tests before writing code for a module. It is carried out in a number of iterations. A developer writes tests first. Then he implements the functionality by writing the code. The code should pass the tests. If it doesn't, then the developer refactors the code or makes changes in the test case. This goes on till the code passes all unit tests. This helps a developer focus more on the behavior and interface of software and each of its modules.

In other words, the developer first defines the classes and the class interfaces for the application. After the interfaces are defined, he writes tests for each class to verify the behavior of class methods. The next step involves writing the code for each class method and testing it in iterations with refactoring as needed.

The unit tests should be simple, easy to understand and test a small piece of code. The implemented code should pass the unit test. Relevant refactoring in implementation or test case is done during iteration. All tests should pass before you add a new piece of code to an existing code.

TDD is used with a version control system so that if the code fails the unit tests unexpectedly, it's always best to revert back to the previous version, that had passed all tests, rather than debugging and trying to find the bugs.

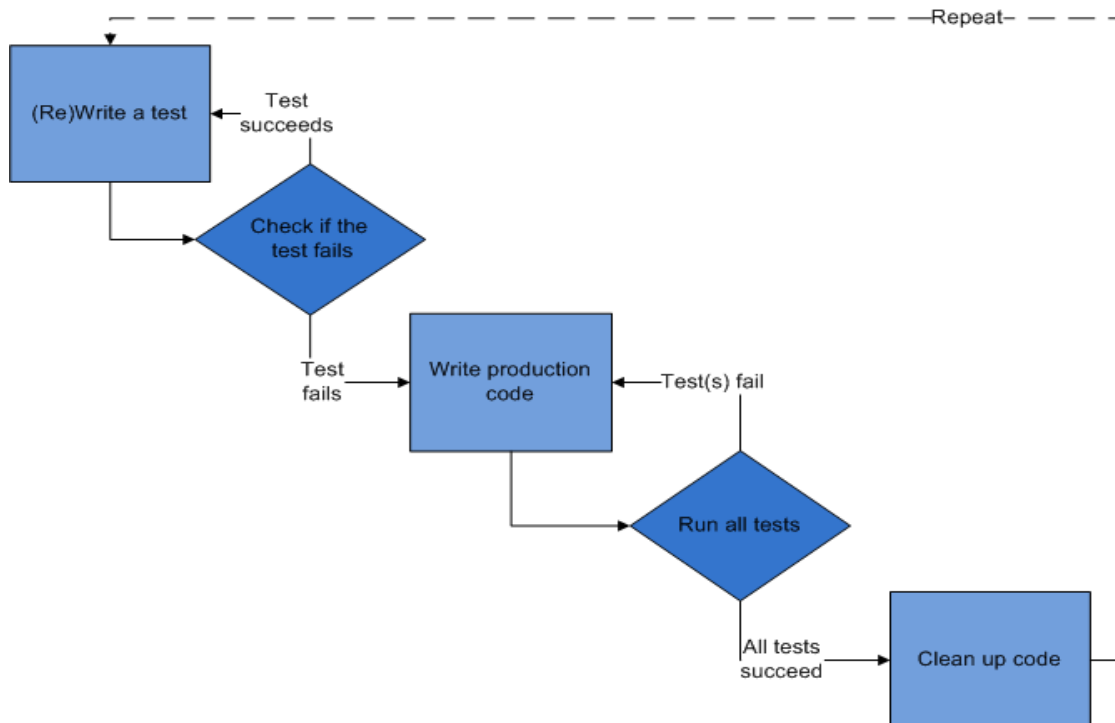


Figure 9.1: Test driven development used from Wikipedia

9.1 Advantages of using TDD

1. TDD helps a developer in better understanding of the code and its logic.
2. Using TDD, the developer becomes more responsible for his piece of code.
3. TDD helps save time in implementation due to better understanding of the business needs. Although, it is true that the amount of code increases.
4. TDD increases productivity.
5. The bugs and errors are found and fixed early during development. A bug found later in development lifecycle is much more costly compared to the same bug found early in the cycle. This helps in reducing the bugs per lines of code in final product. TDD results in 40% fewer defects in code compared to code produced that uses an ad-hoc testing approach. This also helps in saving the costs involved in maintenance.
6. Since, you write tests for every piece of code, whenever the system breaks, you know whether a change in the code is responsible for the error.
7. Unit tests form an essential component of design documentation.
8. TDD produces better quality of code as the developer is forced to keep testing his code and refactor it although the time taken to write the code increases by a small percentage. TDD gives rapid feedback to a developer about the workings and quality of his code. This also helps in improving the quality of code during development.
9. TDD increases the confidence of a developer in the behavior of the system, and ensuring that his code is working as intended and his ability to make changes in code without injecting new bugs increases.
10. It helps create a simple design and also achieve continuous integration.
11. It increases code coverage as there is a unit test for each piece of code. 100 % code coverage is expected although hard to achieve in reality. Thus, TDD improves software quality.

Table 1: Controlled and quasi-controlled empirical experiments on TDD

| Investigator | A/I | Subjects | Software Quality | Developer productivity |
|----------------------------------|------------|-----------------|--|---|
| Janzen and Saiedian (2008) | I | Teams of 1-3 | TDD had better coverage and smaller modules | N/A |
| Janzen and Saiedian (2008) | A | 1-2 teams of 3 | TDD had better coverage, smaller methods, modules, less complexity | N/A |
| Madeyski and Szala(2007) | A | 1 | N/A | TDD has 87-177% better productivity initially |
| Siniaalto and Abrahamsson (2007) | A | 13 | TDD improved coverage | N/A |
| Gupta and Jalote (2007) | A | 22 | Inconclusive | Improved overall productivity |
| George and Williams (2004) | I | 24 | TDD improved test coverage, possibly reduced cohesion | N/A |
| Geras et. Al. (2004) | I | 14 | TDD had better quality | No impact |
| Kaufmann and | A | 8 | N/A | 50% improvement |
| Erdogmus et al. (2005) | A | 35 | No change | Improved productivity |
| Muller and Hagner (2002) | A | 19 | Less reliable, but better reuse | No change |
| Pancur et al. (2003) | A | 38 | No change | No change |

Table 2: Empirical case studies on TDD

| Investigator | A/I | Subjects | Software quality | Developer productivity |
|--------------------------------|------------|-----------------|---|--------------------------------|
| Janzen and Saeidian (2008) | I | Team of 3 | TDD had better coverage and smaller methods and modules | N/A |
| Sanchez et. Al. (2007) | I | 9-17 | 30% reduction in defect density | Increased effort 19% |
| Damm and Lundberg (2006) | I | 100 | 5-30% reduction in fault slip-through, 55% reduction in fault costs | Project cost increased by 5-6% |
| Maximilien and Williams (2003) | I | 9 | 50% reduction in defect density | Minimal impact |
| Williams et al. (2003) | I | 9 | 40% reduction in defect density | No change |
| Bhat and Nagappan (2006) | A | 11 | 2-4 times reduction in defect density | 35% and 15% more time |
| Edwards (2004) | A | 59 | 54% fewer defects | N/A |

9.2 Disadvantages of using TDD

1. Since, the developers have to write more code in terms of unit tests, the management might consider it to be a waste of time. So, it might be difficult to justify its purpose.
2. If the tests are not written well, they can increase the maintenance costs of the system. For example, use of hard-coded error strings that are prone to failure.
3. Unit tests are typically written by a developer. So, if a developer doesn't feel the need to test certain input parameters which might result in error, then it might not be implemented either in unit tests or the code. So, both code and tests will be wrong in this case.

9.3 Limitations of TDD

1. You cannot apply TDD to your database schema.
2. You cannot test GUIs automatically using unit tests.
3. You cannot apply TDD to distributed objects automatically.

9.4 Development of the Studntstore Application using TDD

TDD is a widely followed practice. I started with the reading of IEEE papers to understand TDD in detail. I learned how to follow the practice throughout the development lifecycle. There are a number of available testing tools such as NUnit, xUnit etc. I could have used but I chose Microsoft tool **MSTest**. It is an inbuilt unit testing tool that is easy to use. TDD was very helpful during the development of my application. The benefits I mentioned above increased my understanding of the code. Initially, it was challenging to get used to the practice but my first implementation of the **Registration** Requirement made it a lot simpler later on.

Using TDD, I divided the system into smaller units that can be tested and then implemented. These smaller units are loosely coupled and have cleaner interfaces. TDD is not just a testing technique. It is an analysis technique or design technique.

It helped me create a simpler design because I coded for a module only as much as was required for each of its tests and thus eliminated duplication of code.

It helped me in continuous integration. Every time I found a tested module to be working fine, I integrated it with my main project file and the application with that much functionality was ready to go into production.

10 Testing Framework

I have used MSTest to test my methods and classes in the application. The topics that follow describe the concepts of Unit Testing, Integration Testing and finally, MSTest.

10.1 What is Unit testing?

It is a method by which the source code is divided into smaller testable units and each unit is tested by unit tests usually written and run by software developers.

10.2 Advantages of Unit tests

1. Help find bugs and errors early in software development lifecycle and fix them during development.
2. By writing tests, developers have a much better understanding of the code and become more responsible for their code.
3. Serve as documentation for understanding the code.
4. Increase the speed of software development.
5. Improves code refactoring process and reduces the time taken to do so.
6. If there is a unit test for every piece of code, and a change is made in the code, then developers can run the tests to check if that piece is working fine.
7. It helps in Integration testing. By testing each part of code first and then testing the system as a whole, it reduces the time taken in integration testing.
8. Unit tests act as a design document in Test-Driven Development. They are written first to make sure that the implementation can be traced back to the design. The interfaces are recognized first for each module before developing the modules. The module that is coded is required to pass its unit test.

10.3 Disadvantages of Unit tests

1. It cannot test non-functional test areas such as performance.
2. It cannot test errors caused by interaction of two modules. Thus, it is not useful to catch integration errors.

10.4 Techniques of Unit testing

1. Manual
2. Automated

10.5 What is Integration testing?

It is a method in which the different components of a system that are working together are tested as a whole to make sure their behaviors are what they are supposed to be.

In a web application, integration testing is done by simulating the action of a user that clicks links on webpages and submitting the forms and producing the output. This is often called UI automation.

10.6 Benefits of Unit testing over Integration testing

1. Integration tests take longer time and run more slowly because they deal with rendering HTML pages, database queries and HTTP requests etc.
2. Unit tests are used to test small individual units.
3. Unit tests are better suited for testing C# methods.
4. Unit tests require less maintenance compared to integration tests.
5. Integration testing makes use of a real database so you have to restore database after every test or reset it.

10.7 Benefits of Integration testing over Unit testing

1. Integration tests help in testing the behavior of some or all the components and their interactions in a system as a whole.
2. Integration tests are better suited for User Interfaces as they help specify the behavior of UI as a series of interactions in a sequence.
3. Integration testing helps understand how a user can perform a certain system operation with a specific sequence of actions. Thus, acting as a good design or document.
4. Integration testing helps describe, test and understand JavaScript and browser behavior.
5. Integration testing helps make sure that all the components of a deployed web application are working correctly.

10.8 Integration testing tool alternatives available in the market

1. *Selenium RC* (<http://seleniumhq.org/>)
2. *WatiN* (<http://watin.sourceforge.net/>)

10.9 Other testing tool alternatives available in the market

1. Top-quality Unit Testing frameworks (NUnit, xUnit),
2. Mocking frameworks (Moq, Rhino Mocks),
3. Inversion-of-control containers (Ninject, AutoFac),
4. Continuous integration servers (Cruise Control, TeamCity),
5. Object-relational mappers (NHibernate, Subsonic),

10.10 Comparison of MSTest AND NUnit testing frameworks

10.10.1 Advantages of MSTest over NUnit

1. Unlike NUnit, MSTest has TestContext that passes information about the test run. This can be useful in pulling information from datasources on disk to the unit tests.
2. MSTest is a built-in tool of Visual Studio 2008. It is easy to generate and run unit tests using MSTest. Unit tests can also be automatically generated for a piece of code.

10.10.2 Disadvantages of MSTest over NUnit

1. It doesn't support inheritance.
2. Tests are executed in parallel. So, if test R is dependent on output of test D for its success, then it will fail because both the tests start running at the same time.

10.11 How to create Unit tests using MSTest in Visual Studio 2008

In this section, I will discuss the steps in creating a Unit test in Visual Studio 2008 using Microsoft's built-in testing tool called MSTest. In order to test the Business Class Layer classes in the project, we need to have them in a separate class library so that the dlls can be unit tested.

10.11.1 First, open a Business Logic Layer Class file such as BookInformationBLL.cs.

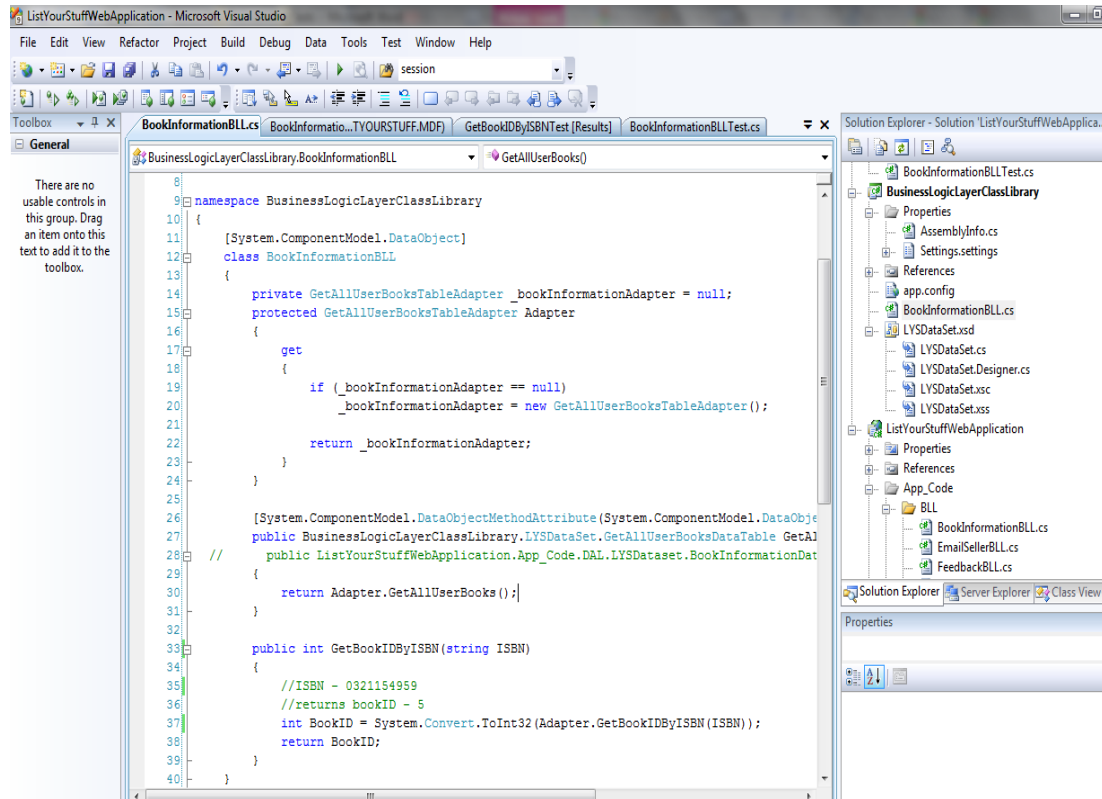


Figure 10.11.1: Unit test creation- Open a BLL file

10.11.2 Next, right click anywhere inside the file and a window pops up.
Select **Create Unit Tests...**

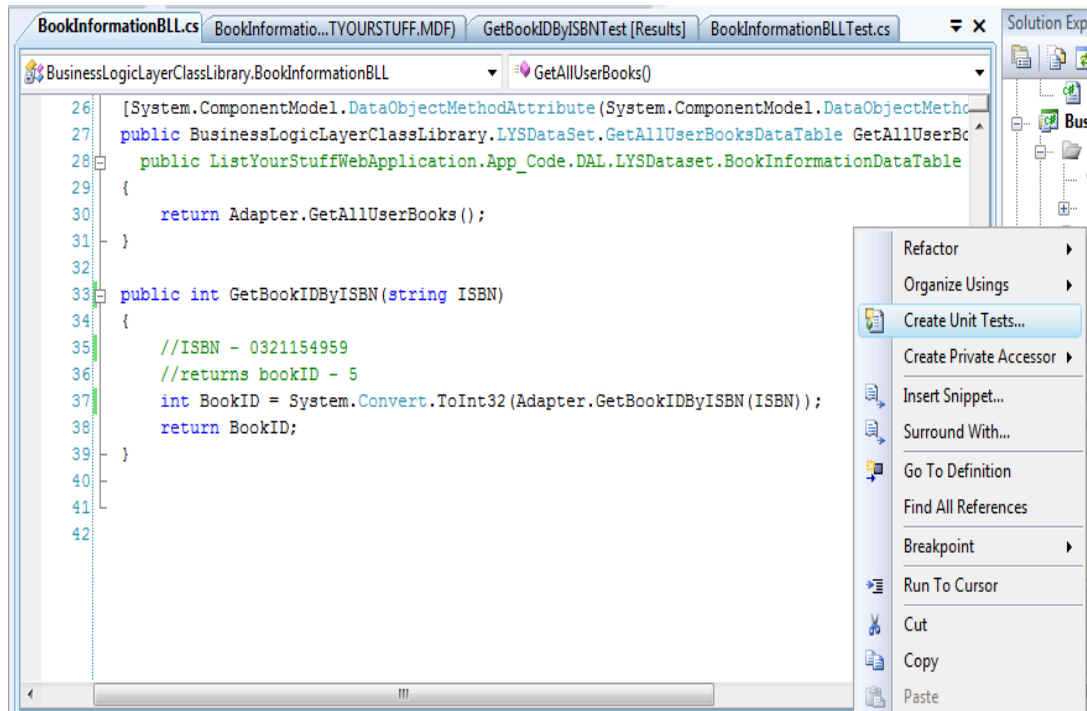


Figure 10.11.2: Unit test creation- Select Create Unit Tests

10.11.3 Next, select the method, such as `GetBookIDByISBN(System.String)` in order for MSTest to automatically generate Unit test code for you, and click **OK**.

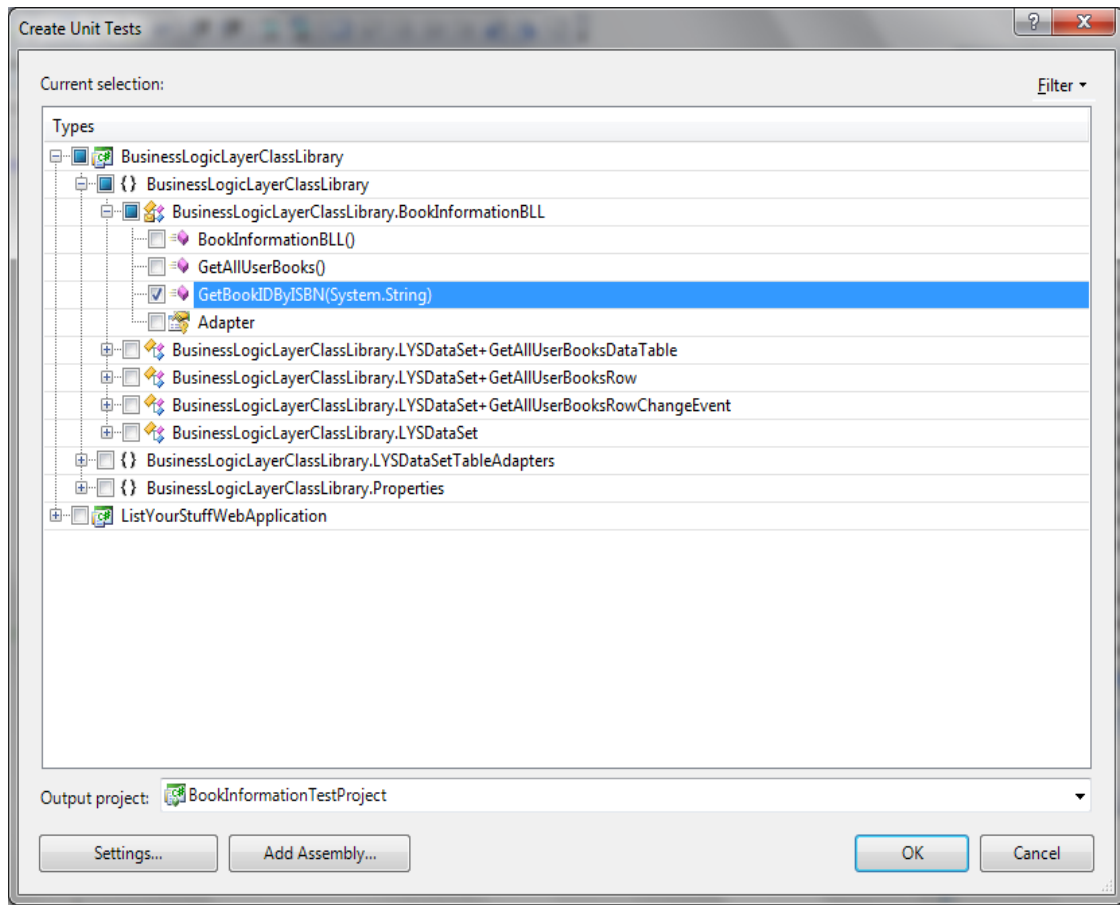
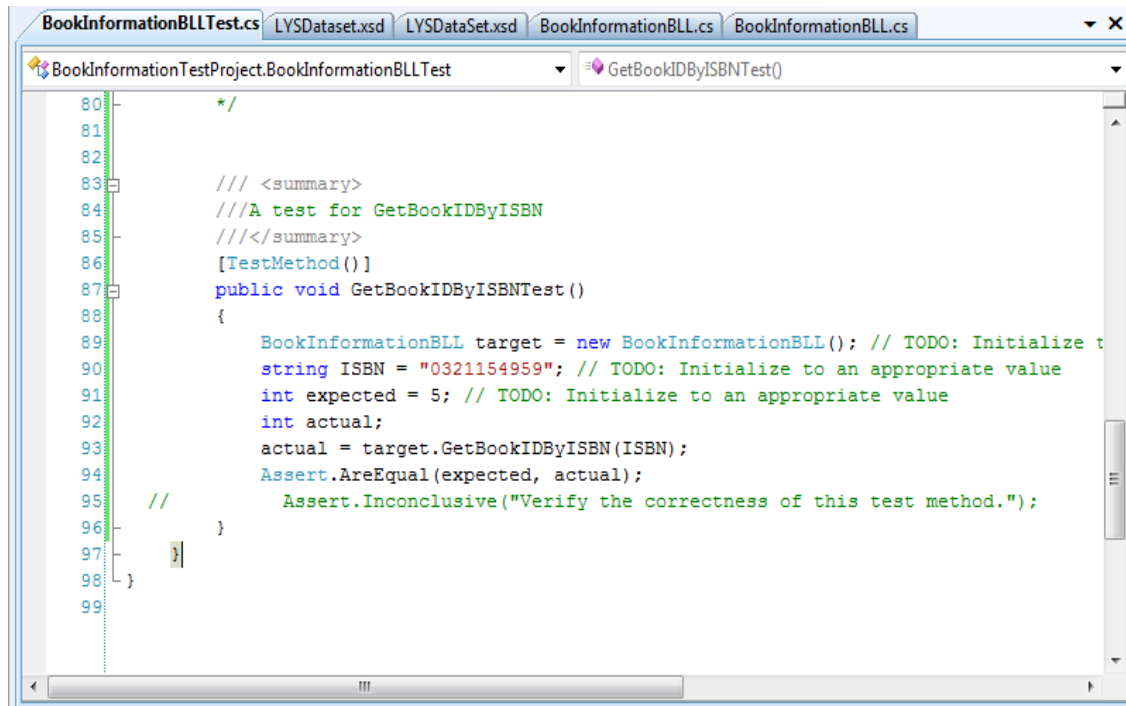


Figure 10.11.3: Unit test creation- Select a BLL class method

10.11.4 As below, the tool automatically creates most of the Unit test code. All that I have changed here is ISBN value to 0321154959 and expected value to 5.



```
80  */
81
82
83  /// <summary>
84  /// A test for GetBookIDByISBN
85  /// </summary>
86  [TestMethod()]
87  public void GetBookIDByISBNTTest()
88  {
89      BookInformationBLL target = new BookInformationBLL(); // TODO: Initialize t
90      string ISBN = "0321154959"; // TODO: Initialize to an appropriate value
91      int expected = 5; // TODO: Initialize to an appropriate value
92      int actual;
93      actual = target.GetBookIDByISBN(ISBN);
94      Assert.AreEqual(expected, actual);
95      // Assert.Inconclusive("Verify the correctness of this test method.");
96  }
97
98
99
```

Figure 10.11.4: Unit test creation - Add expected and actual values in Unit test

10.11.5 Next, right click anywhere inside the test file, and click **Run Tests**.

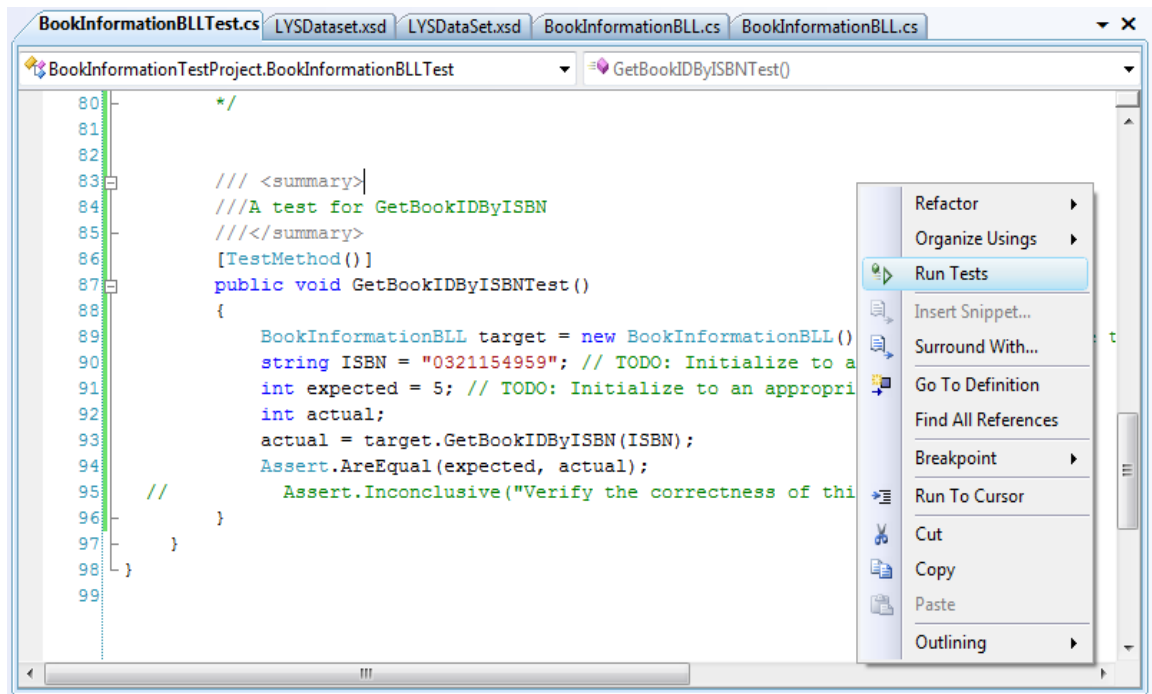


Figure 10.11.5: Unit test creation – Select Run Tests

10.11.6 The test starts as shown below to compare the expected value and actual value.

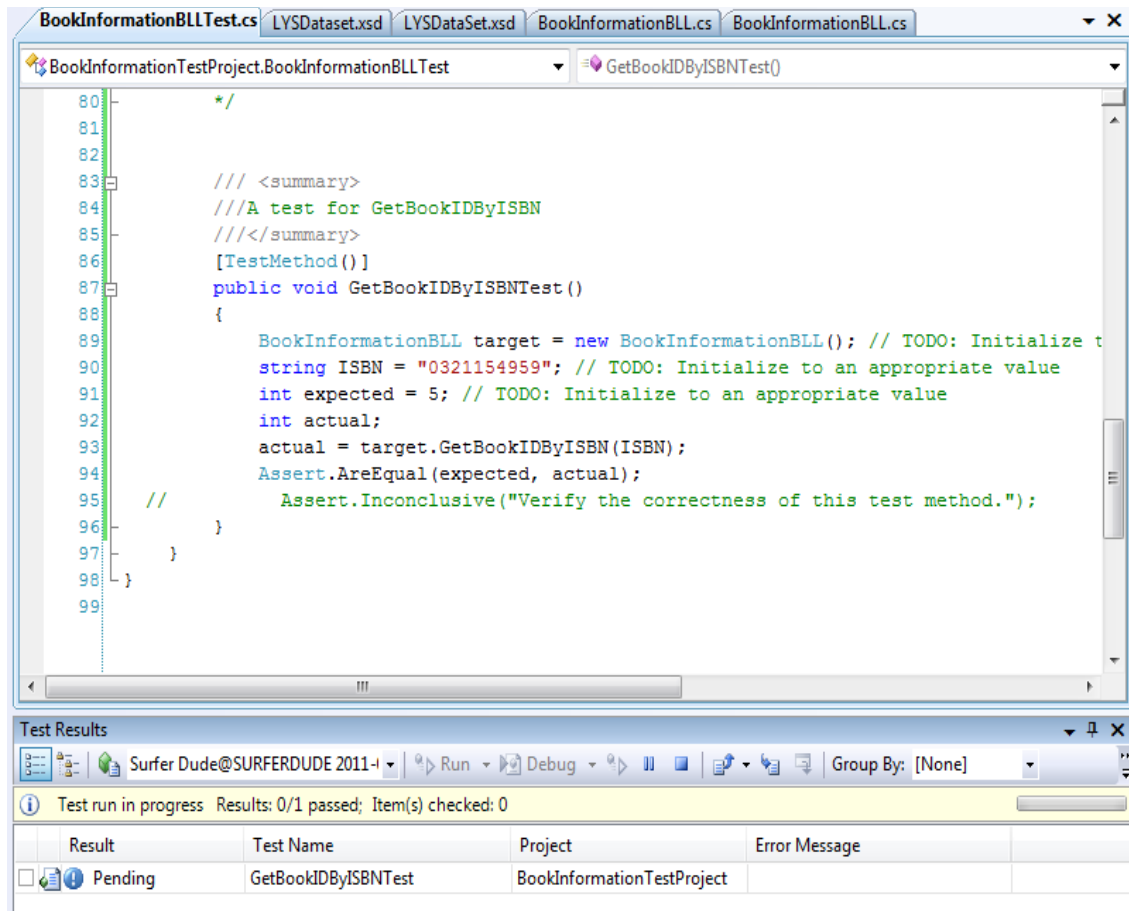


Figure 10.11.6: Unit test creation – Test in progress

10.11.7 The test passes as the expected and actual values match.

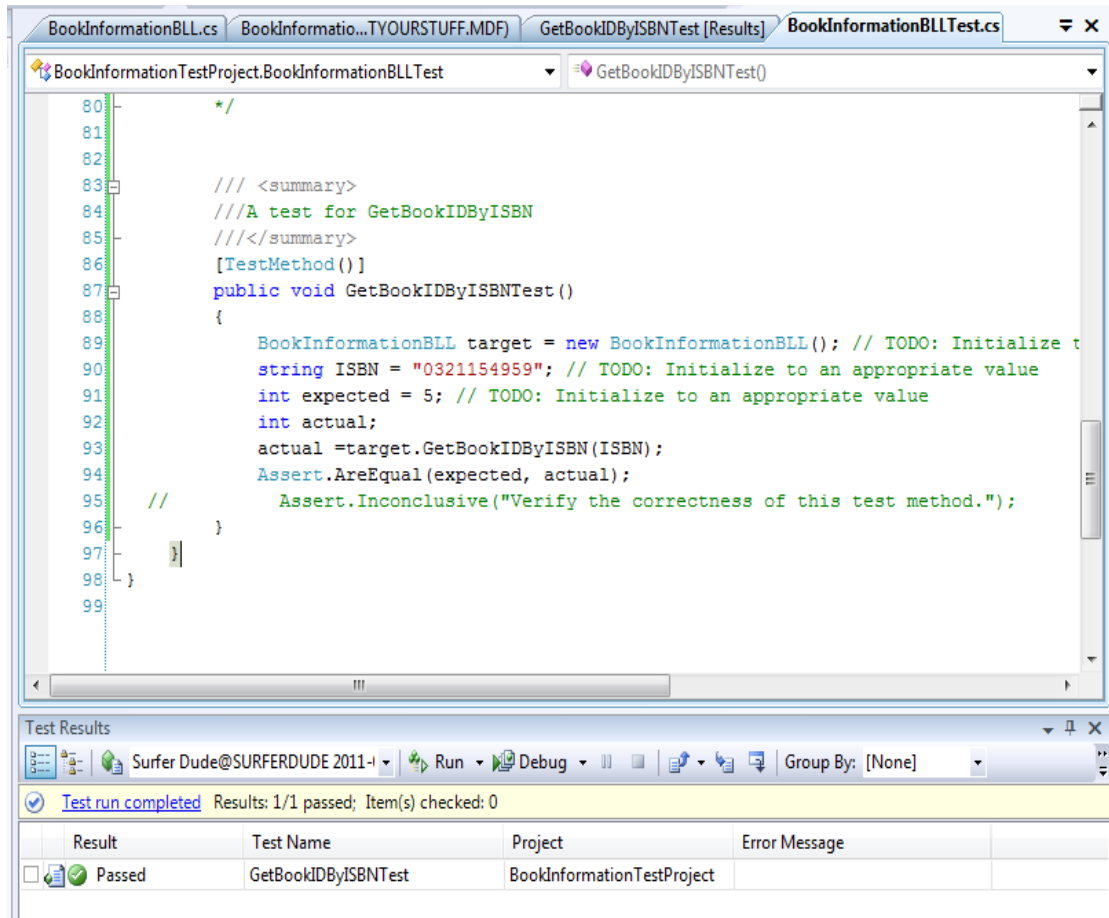


Figure 10.11.7: Unit test creation – Test Passed

11 Security - Forms Based Authentication

The Studntstore application uses Forms Authentication provided by Microsoft to manage the user accounts, roles and user access to its resources.

Forms Authentication is a commonly used authentication model in web applications. In this model, a user provides his credentials usually on a login form page and submits it to the system to get authenticated for access to the application's resources.

If a user tries to view an unauthorized page, he is directed to the login page. First, a user enters his credentials through a web form such as login page in a browser. These credentials are matched against the values in the user account stored in the database. With successful verification and match, a form authentication ticket is created. The ticket is stored in a cookie created on the user's computer. The cookie has a limited life. So within a specific time, if the user visits numerous pages requiring authentication, the forms authentication ticket is submitted with every HTTP request to the web server to keep the user logged in to the system. When the user is inactive for some time, the cookie expires. In this case, the user becomes anonymous and thus, has to provide his credentials again to get identified by system.

The form authentication merely identifies the user as a member or reports them as anonymous. It is the authorization system that decides whether a user has required permissions to access a certain resource. If the user doesn't have permissions to access a page, then the authorization system reports that the user is denied and this causes the forms authentication to redirect the user to the login page.

The forms authentication ticket is encrypted and digitally signed to prevent tampering.

11.1 Understanding the Forms Authentication Workflow

When a user requests a specific page such as AddBooks.aspx, the FormsAuthenticationModule looks for an authentication ticket in the request. If it can't find it, it presents the user as anonymous to authorization module. If the user is not authorized to access that resource, then the authorization module sends the HTTP 401 Unauthorized status response with the termination of request. This status is read by FormsAuthenticationModule and it changes the status to redirect the user to the login page through HTTP 301 Redirect status. If the user is authorized, then the authorization module redirects the user to the requested page.

In the second request, we see that the anonymous user has clicked on login link on website and requested for that page. Since the login page can be accessed by anyone, the authorization module returns the login page to the user.

In the third request, the user submits his username and password on login page. The FormsAuthenticationModule sends the information to the authorization module presenting the request as an anonymous request. The authorization module processes and validates the credentials and grants access. It also generates a form authentication ticket.

In the fourth request, when the user requests any other page, the ticket is submitted with HTTP request. Thus the FormsAuthenticationModule sends the authenticated request to the authorization module and depending on the access permissions; the authorization module either grants or denies access.

The FormsAuthenticationModule attempts to authenticate the user prior to the UrlAuthorizationModule (and FileAuthorizationModule) executing. If the user making the request is not authorized to access the requested resource, the authorization module terminates the request and returns an “HTTP 401 Unauthorized status”. In Windows authentication scenarios, the “HTTP 401 status” is returned to the browser. This status code causes the browser to prompt the user for their credentials via a modal dialog box. With forms authentication, however, the “HTTP 401 Unauthorized status” is never sent to the browser because the FormsAuthenticationModule detects this status and modifies it to redirect the user to the login page instead (via an “HTTP 302 Redirect status”).

The Forms Authentication model’s responsibility is to determine if the user’s credentials are valid and, if so, to create a forms authentication ticket and redirect the user back to the page they were attempting to visit. The authentication ticket is included in subsequent requests to the pages on the website, which the FormsAuthenticationModule uses to identify the user.

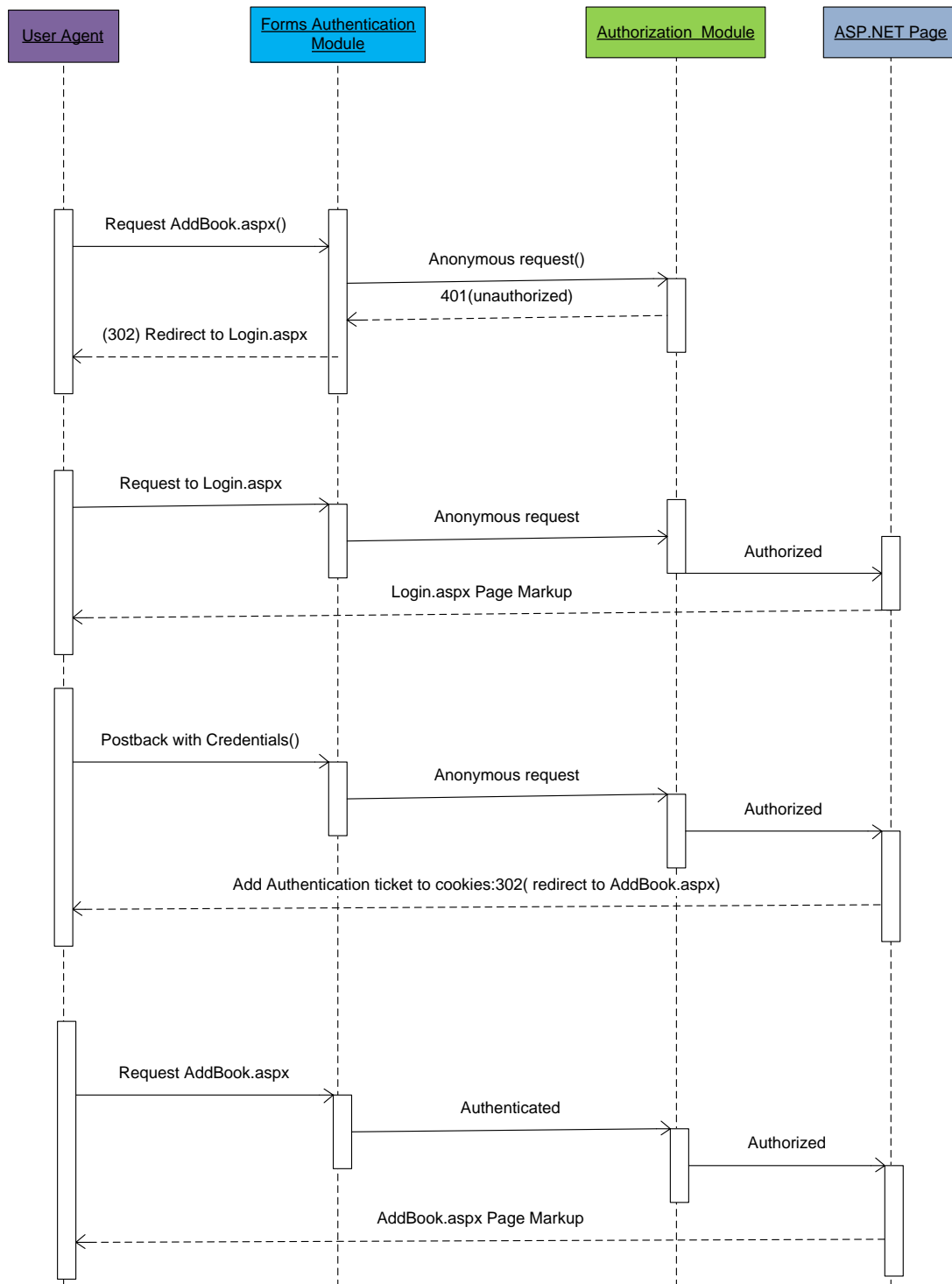


Figure 11.1: The Forms Authentication Workflow

ASP.NET provides a Membership Framework and built-in Login Web controls that help perform important user account-related tasks such as creating login page, storing account information etc. A Membership Framework is a collection of classes in the System.Web.Security namespace.

The diagram below shows how the Membership Framework works.

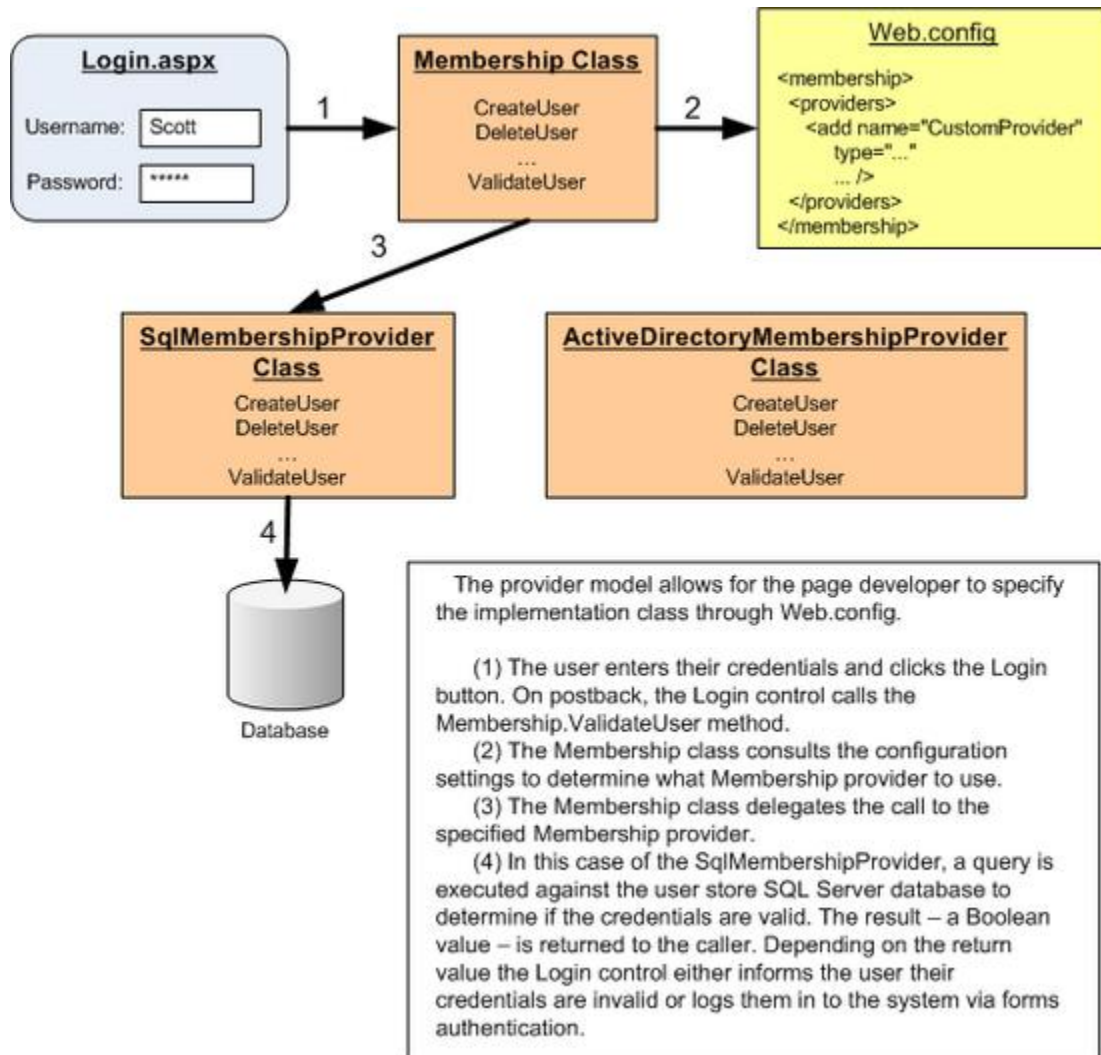


Figure 11.2: Membership framework- used from MSDN Microsoft asp.net website

An example of how it works

1. The user enters the username and password in the login page and submits.
2. On HTTP post, the login control invokes the `Membership.ValidateUser` method.
3. The `Membership` class checks in the configuration file `web.config` to decide on the `Membership` provider to be used.
4. According to the provider specified in the config file, the `Membership` class delegates the call, in our case, to the `SqlMembershipProvider`.
5. The provider runs a query against the ASP.NET database (which stores all user accounts) to verify the credentials of the user.
6. A Boolean value is returned to the caller depending on which the login control logs the user into the system or lets them know that the credentials are invalid and requests them to enter them again.

11.2 Why use a Membership provider?

As we observed, Microsoft has separated the `Membership` class and `Membership` provider. The membership provider deals with querying the database. We can easily replace it with our own custom membership provider which might be based on a totally different programming language and technology.

11.3 Roles using `SqlRoleProvider`

`Membership` Framework makes use of `SqlRoleProvider` to give roles to users based on their level of access permissions in the system. Some examples of roles might be Administrator, guest, member etc. Based on a specific role of a group, we can give specific access permissions to a set of users in that group.

11.4 Definitions

| No. | Words | Definition |
|-----|--------------------|---|
| 1. | Authentication | process of determining the user's identity |
| 2. | Authorization | Process by the system decides whether a user has the authority to access a certain resource or page |
| 3. | User account | An object that contains all the information of a specific user |
| 4. | Member | A user who has previously registered on the website |
| 5. | Credentials | Username and password of a user |
| 6. | Anonymous | A user who tries to access a member's only page but has not registered on the website and so doesn't have a username and password |
| 7. | Role | A label given to each user which helps determine the authorization rules and page- level functionality. |
| 8. | Authenticated user | A user who has been successfully identified by the system |
| 9. | Authorized user | A user whose role gives his access permission to a resource |
| 10. | Browser | A User Interface application such as Google chrome, Mozilla Firefox or Internet Explorer used to show output to user and get input from user. |
| 11. | Cookie | A small text file stored on the user's computer. It is transmitted with each HTTP request to the website that generated it. |

12 Master page and CSS Stylesheets

Visual Studio 2008 and ASP.NET C# allow the creation and use of a Master page to be used for all webpages in an application. A master page is a web page which has content common to all the web pages of an application. We centralize the common functionality which makes updating them easier. We create a Master page so that we do not need to add the same components again and again to each web page. A Master page gives a consistent look and feel and standard behavior to all webpages in a web application. It also makes the User Interface designing easier. We create a common page called the Master page and then we can create individual content pages that contain specific information applicable to that web page. This individual content merges with the content and common layout of the master page to give a consistent layout for the entire web application.

The Master page contains most of the top-level HTML elements of a webpage such as html, head and form. For example, you can add a company logo, standard top header and bottom header in a master page to make them appear in all web pages.

The ContentPlaceHolder control in a Master page defines a region which will contain the individual content generated by the other non-Master web pages.

We can create multiple Master pages for different layouts for different parts of a web application.

Cascading Style Sheets (CSS)

CSS are used to control the style and way in which you want to display the HTML elements. You save a lot of redundant code required to display styles of different controls. We can change the look, layout and appearance of a website by just making changes in the .css file.

13 Challenges faced during the development lifecycle of the Studntstore application

This section gives a brief description of some of the challenges I faced while working during the different phases in the project including database design, coding, implementing three-tier architecture, writing report and testing my website.

13.1 Three Tier Architecture implementation

I started with understanding the basics of the three tier architecture. I read through articles on the Microsoft's MSDN website which has very good tutorials that take you step by step in developing a sample application with a three tier approach. It explains the details of each layer including the components involved in each layer.

During the programming phase of development, I ran into a lot of errors. The ASP.NET forum has experienced developers that read your programming related errors, your logic and help you in your development. Whenever, I got stuck at any point while programming, which could be related to coming up with some logic for some functionality, or using a library method for some logic, or troubleshooting an error, I posted my errors or queries on the ASP.NET forum. The experienced software developers in the forum would take a look at it and then guide me to the correct source. For each error, I had to understand the root cause, research on the web and try out the various solutions related to those errors. Sometimes I found the correct solution and sometimes I had to post my errors on ASP.NET website forum.

One time, I was getting an ObjectDataSource Control error. I researched a lot but could not find a correct resolution for my issue. I posted the error on the forum. I got replies from software developers within 2 days. Meanwhile I was researching on the web and trying available solutions. Finally, the one suggested on the forum worked out for me. Troubleshooting and searching for a solution to an error helped me in increasing my understanding of the code and its workings and how the different modules interact with each other. While trying to fix a problem, I came across many areas of code that I wanted to refactor after I gained better understanding through research.

Example of ObjectDataSource Error

ObjectDataSource 'ObjectDataSource1' could not find a non-generic method 'AddUser' that has parameters: FirstName, LastName, Username, UserPwd, Gender, BirthDate, Phone, Email, UniversityName, UniLocationAdd1, UniLocationAdd2, City, state, Zip, CState.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.InvalidOperationException: ObjectDataSource 'ObjectDataSource1' could not find a non-generic method 'AddUser' that has parameters: FirstName, LastName, Username, UserPwd, Gender, BirthDate, Phone, Email, UniversityName, UniLocationAdd1, UniLocationAdd2, City, state, Zip, CState.

Source Error:

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Stack Trace:

[InvalidOperationException: ObjectDataSource 'ObjectDataSource1' could not find a non-generic method 'AddUser' that has parameters: FirstName, LastName, Username, UserPwd, Gender, BirthDate, Phone, Email, UniversityName, UniLocationAdd1, UniLocationAdd2, City, state, Zip, CState.]

System.Web.UI.WebControls.ObjectDataSourceView.GetResolvedMethodData(Type type, String methodName, IDictionary allParameters, DataSourceOperation operation) +1119426

System.Web.UI.WebControls.ObjectDataSourceView.ExecuteInsert(IDictionary values) +440

System.Web.UI.DataSourceView.Insert(IDictionary values, DataSourceViewOperationCallback callback) +86

System.Web.UI.WebControls.FormView.HandleInsert(String commandArg,

Boolean causesValidation) +272

System.Web.UI.WebControls.FormView.HandleEvent(EventArgs e, Boolean causesValidation, String validationGroup) +676

System.Web.UI.WebControls.FormView.OnBubbleEvent(Object source, EventArgs e) +95

System.Web.UI.Control.RaiseBubbleEvent(Object source, EventArgs args) +37

System.Web.UI.WebControls.FormViewRow.OnBubbleEvent(Object source, EventArgs e) +113

System.Web.UI.Control.RaiseBubbleEvent(Object source, EventArgs args) +37

System.Web.UI.WebControls.LinkButton.OnCommand(CommandEventArgs e) +118

System.Web.UI.WebControls.LinkButton.RaisePostBackEvent(String eventArgument) +135

System.Web.UI.WebControls.LinkButton.System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(String eventArgument) +10

System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl, String eventArgument) +13

System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData) +175

System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +1565

Version Information: Microsoft .NET Framework Version: 3.5; ASP.NET Version: 3.5

How the forum helped me resolve an issue

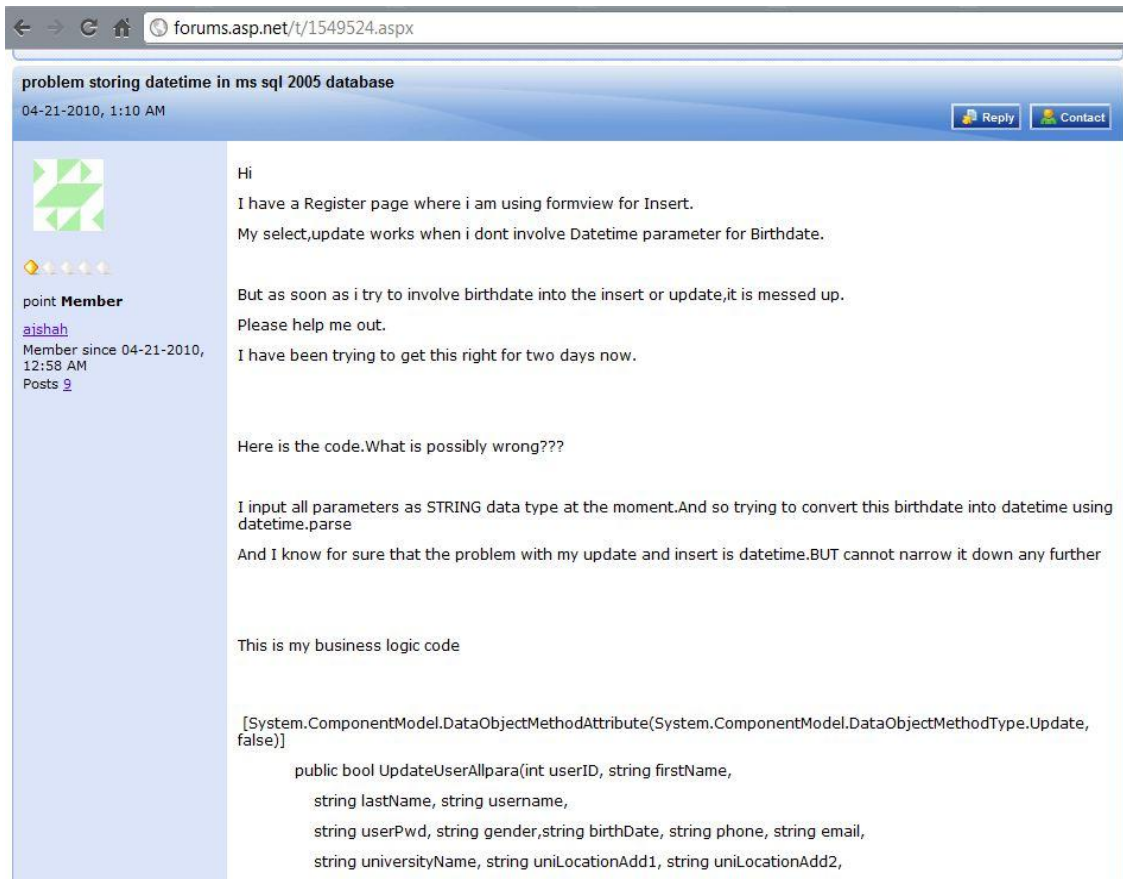


Figure 13.1: Describes my question posted on the ASP.NET forum

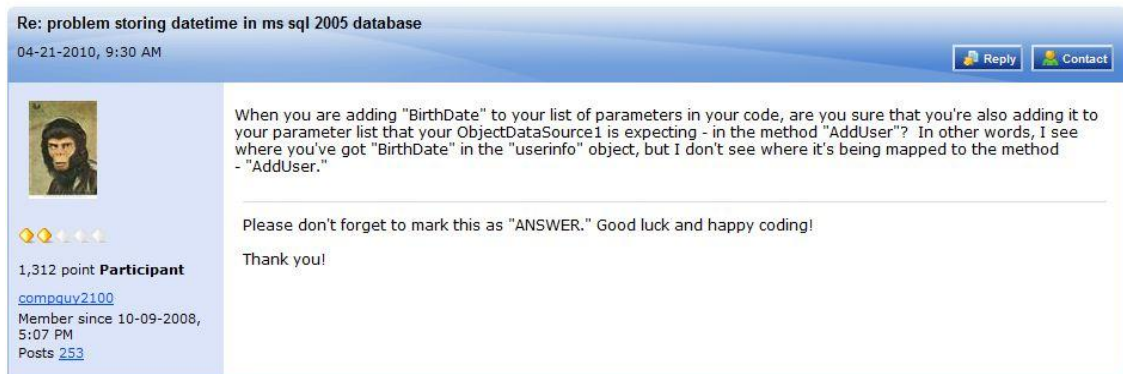


Figure 13.2: An experienced developer trying to help on ASP.NET forum

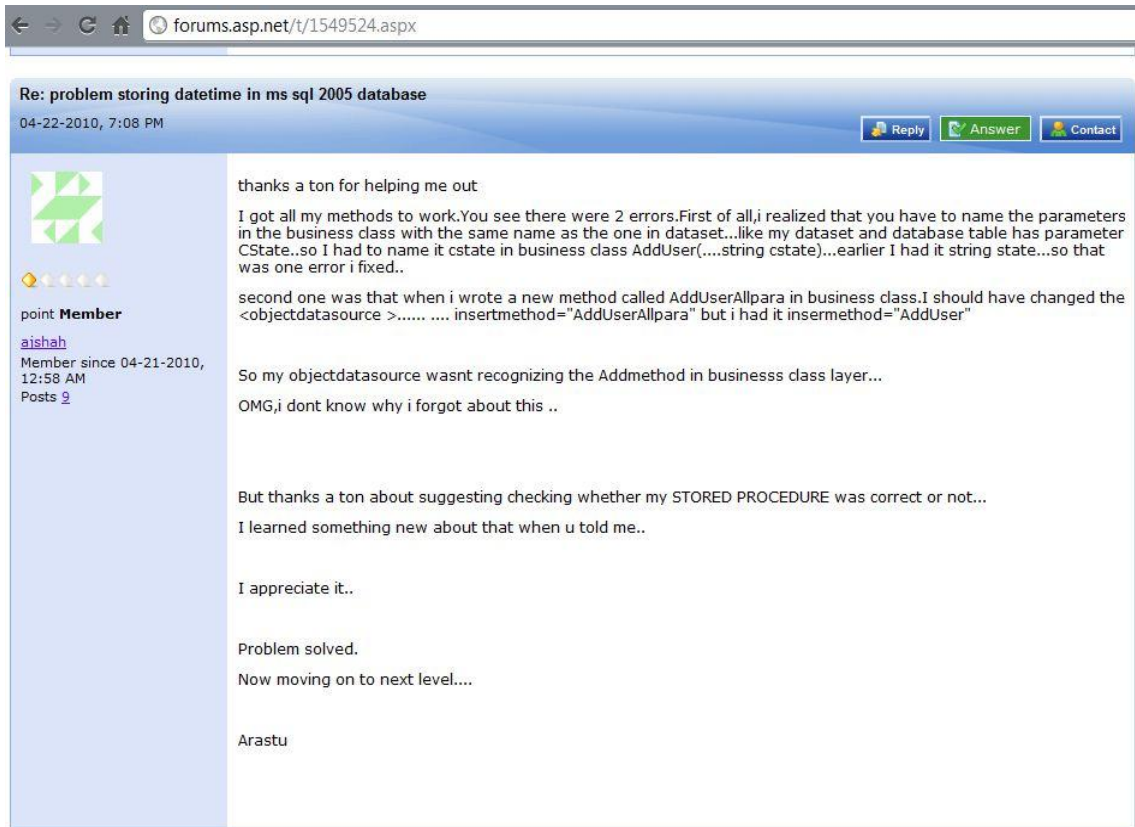


Figure 13.3: Describes how I resolved my issue using ASP.NET forum

After I had implemented a requirement for one function on the website, I got used to the three tier architecture development style. I first implemented the Register functionality on the website. Once complete, I could select, insert, update and delete users on the website. From there on, it got a little easier in implementing other functionalities of adding a book, deleting a book, viewing books.

[forums.asp.net/p/1596705/4059172.aspx#4059172](#)

[Home](#)
[Get Started](#)
[Downloads](#)
[Web Forms](#)
[MVC](#)
[AJAX](#)
[Community](#)
[Wiki](#)
[Forums](#)

[Home](#) > [ASP.NET Forums](#) > [Data Access](#) > [ADO.NET, Entity Framework, LINQ to SQL, NHibernate](#) > [asp.net insert into 3 related tables using three tier architecture and typed datasets](#)

[Previous](#)
[Next](#)

asp.net insert into 3 related tables using three tier architecture and typed datasets

Last post 09-03-2010 12:39 AM by [aishah](#). 3 replies.

Sort Posts: Oldest to newest

asp.net insert into 3 related tables using three tier architecture and typed datasets

08-29-2010, 4:25 PM
 [Reply](#)
[Contact](#)

point **Member**
[aishah](#)
 Member since 04-21-2010,
 12:58 AM
 Posts [2](#)

I am using ASP.NET (C# and sql server express 2005) for my web application project.
 I have three related tables.
 I want to do an insert using typed datasets,tableadapters and three tier architecture.

 I wish to know how to do simultaneous inserts in 3 tables.

 So i wish to insert a row in first table and get its primary key.
 Same with the second table.

 I will use the primary keys of those first two tables to do the insert in the third table.

 How can i accomplish this using formview control and what methods should i use??

 Please give me suggestions.

 thanks in advance for reading this. I really appreciate it.

[Re: asp.net insert into 3 related tables using three tier architecture and typed datasets](#)

09-01-2010, 2:06 AM
 [Reply](#)
[Contact](#)

56,554 point **All-Star**
[Wencui Qian - MSFT](#)
 Member since 01-07-2008,
 3:33 AM
 Posts [5,777](#)

Do you want to get the new inserted identity in the query? Usually, if you're executing a tsql query, you can add the query "; select @@IDENTITY;" in the end to get it. You can also put it in an output/return parameter in a stored procedure in order to get it.

 BTW, if you want to keep the three operations in a single logic unit, you may consider to do it in a transaction.

 Thanks.

Please mark the replies as answers if they help or unmark if not.
[Feedback to us](#)
[Microsoft One Code Framework](#)

[Re: asp.net insert into 3 related tables using three tier architecture and typed datasets](#)

09-03-2010, 12:39 AM
 [Reply](#)
[Answer](#)
[Contact](#)

point **Member**
[aishah](#)
 Member since 04-21-2010,
 12:58 AM
 Posts [2](#)

Hi

 I really appreciate your reply.. I found a solution..

I created a tableadapter with the main input fields which belong to 3 tables. The tableadapter has 3 stored procedures that insert into each of tables. The tableadapter also has stored procedures to return identity columns back with any one other column as parameter for the first two tables. I then use those parameters to insert in the final table. And I used the same business class file for all inserts in 3 related tables.

Your forum gives confidence in solving a problem. Knowing that someone is out there to help you out, the feels really good.

 Thanks everyone.

AJ

Figure 13.4: Another Example of use of Forum

With every implementation of a requirement, my confidence and speed in implementing the rest of requirements increased manifolds. I was much more comfortable in the later part of my development phase due to gaining more experience.

13.2 Database design and programming

At the beginning of project, I had to revise my database design and programming skills. I was trained in SQL 2000 database design and programming in 2006. I used those skills in a real world application development in 2007 at Cognizant. I had not done any SQL programming after 2008. So, beginning 2010, I used websites such as w3schools, and Wikipedia and Sam's Teach yourself SQL book to revise the basics of SQL. Then, I started with my database design. I came up with a few tables and their respective columns based on the entities. Once ready with the tables, I normalized them in order to remove the redundancy.

Since my database queries involve SQL Selects, Inserts, Updates and Deletes in 2-3 related tables at the same time, I had to read up about how to program correlated and non-correlated queries and how they could be used. I also had to make sure that the primary id of each table incremented automatically on an SQL insert. While defining the table column properties, I had to decide on which columns could have null values. While designing Typed Datasets and TableAdapters, I had to understand how to call each Dataset and TableAdapter class to execute an SQL query.

I have used a Master page template in my application. I had issues with relative urls, related to various webpages, used in the Master template. I researched on the web to find a solution which helped resolve folder url site.master conflicts using code such as `<%= ResolveClientUrl("~/images/mypic.gif") %>`.

13.3 Registration and Login Functionality implementation

For the login authentication and authorization, I have used Microsoft's Membership Framework and Forms Authentication. After a user logged-in, I had to write custom code to pull up a members email and username in order to implement some functionality that involved passing values to `contactseller.aspx` page before contacting a seller and displaying the username to the logged-in member. I had to read up on the Membership Classes of the Membership Framework in detail to have clear concepts.

13.4 Business Class Layer implementation

I faced a challenge of linking my business class methods to the ObjectDataSource control methods. I learned through troubleshooting that if I had coding errors in the

business class methods, the methods won't show up in ObjectDataSource linking wizard. I had to make sure that my coding syntax was correct.

13.5 Testing and TDD

While I was doing the development, I tested my website every now and then to make sure all links were working as intended and all functionality could be traced back to the requirements. I used MSTest unit testing provided by Microsoft.

13.6 User Interface

To improve the design of my User Interface, I learned how to use div tags, tableless layouts and CSS Stylesheets. CSS helped me manage the styles, fonts, text size etc. of the web page content very easily.

Another challenge in designing the User Interface was keeping the footer at the bottom of the webpage, arranging the different components of the User Interface and keeping them in their place. I made use of <div> tags and CSS design in order to arrange my content. I had to understand the different types of CSS positioning such as static, relative, absolute etc. in order to position the elements. Positioning can also be done using top, right, left and bottom properties. I had to make sure that my footer stayed at the bottom of the page when different web requests rendered contents of different amount and size. It was crucial to prevent overlapping of website sections on resizing the browser.

13.7 Validating User Input

In order to use client side input validation, I researched on how to use client side built in controls in Visual Studio 2008. For each input control, I had to decide on what I expected the user to type and what characters would be considered invalid. I used the regular expressions for the same from a website called regexlib.com. I searched for a specific regex for each input box, found one and then studied the syntax of that regex to make sure it catches all invalid input by the member.

I also used Microsoft MSDN website to get regular expressions for some input controls. For example, the regex used for the password textbox is provided by Microsoft on their MSDN website in order to request user to type in strong passwords for their account.

13.8 Report Writing

Some challenges faced while writing the report were learning how to create sequence diagrams and activity diagrams, difference between asynchronous calls and synchronous call in sequence diagrams and when each of them is to be used. I got the

advice of highly skilled professionals in my team at Amgen by requesting them to take a look at my use-cases. They checked it and suggested how I could improve on each use case. I set one on one meeting with some team members to learn out about their experience using three tier architecture, use case, sequence diagram, other UML diagrams etc. After I completed the report and polished it, I asked my team members, supervisors and managers at Amgen to read it and give feedback. I also asked them for help in testing my website.

14 Conclusion

This section concludes my thesis with a brief explanation of the goals and benefits achieved and features in the future releases of this application.

The goal to come up with an economical solution that would allow students to buy and sell their books to other students in their university has been achieved. The economic benefits of having an alternative to a similar solution offered by websites such as amazon and craigslist have been achieved. This application doesn't charge any fees for its service or any sale commissions and also doesn't require shipping of books. It reduces the occurrence of spams and bogus users with its exclusive use meant for university students with an .edu email account only. However, advertisers do have to pay a certain amount of fees for advertising their products to the users of this website.

A comparison of ASP.NET Web Forms and Three Tier Architecture has been made with alternative technologies. With this thesis, there is now a deeper understanding of the benefits of using Web Forms and Three Tier architecture. A complete cycle of software development has been studied along with a detailed set of documentation for each phase. The application has been thoroughly tested throughout the software development lifecycle to trace its features back to the requirements.

I have significantly improved my problem detection and solving skills while developing this application. I have learned how to take a bigger problem in hand, break it into smaller pieces and solving them one at a time. I have realized that solving any problem requires the help and advice of others and it is always beneficial to reach out to others. At the same time, I have learned to help others solve their problems by sharing my experiences. My programming skills have also been enhanced.

The criteria for success depend on a lot of factors. To understand the root cause of a problem and then coming up with a solution is very essential. I required a lot of determination, hard work and focus to develop this application. You also need to constantly add and improve the features of your application keeping in mind the change in time and technology.

I have started promoting my website at CSUN through different university sports clubs, fraternities, classmates and university newspaper. I have requested permission from CSUN to distribute flyers, related to my website, on campus to the students.

The future release of this application will allow students to sell any items such as cars, furniture, sports items etc. to other students. I also plan to have a live help feature to improve the user experience. Currently, the website will only support students of California State University, Northridge. But the website will be available for use to other university students. Supporting other universities might be a possibility in the future. At present, while adding a book, the system doesn't prepopulate book information based on the entered ISBN or author name but this will be implemented using Amazon web services in the second release.

15 References

1. David S. Janzen; “Software Architecture Improvement through Test-Driven Development”; ACM New York, NY, USA ©2005; ISBN:1-59593-193-7
2. Hongbing Kou, Philip M. Johnson; “Operational definition and automated inference of test-driven development with Zorro”; Automated Software Engineering, Volume 17, Issue 1, March 2010; Kluwer Academic Publishers Hingham, MA, USA; Pages 57 – 85
3. Jaime Spacco, William Pugh; “Helping Students Appreciate Test-Driven Development (TDD)”; ACM New York, NY, USA ©2006; ISBN:1-59593-491-X ;Pages 907-913
4. Gerardo Canfora, Aniello Cimitile, Felix Garcia; “Evaluating advantages of test driven development: a controlled experiment with professionals”; ACM New York, NY, USA ©2006; ISBN: 1-59593-218-6 Order Number: 594057; Pages 364 – 371
5. Robert A. Maksimchuk, Eric J Naiburg; UML for Mere Mortals; Addison-Wesley Professional; 1 edition (November 5, 2004); ISBN 0321246241; Pages 288
6. Pro ASP.NET MVC 2 Framework – Second Edition by Steven Sanderson
7. The Secure Three-Tier Architecture Pattern by E.B Fernandez ; M Fonoage ; M VanHilst ; M Marta
8. Kent Beck; Test Driven Development: By Example; Addison Wesley; ISBN 0-321-14653-0, Pages 240
9. Thirumalesh Bhat; Evaluating the Efficacy of Test-Driven Development: Industrial Case Studies; ACM New York, NY, USA ©2006; ISBN: 1-59593-218-6 Order Number: 594057; Pages 356 - 263
10. Jeff McWherter, Ben Hall; Testing ASP.NET Web applications; Wrox (October 26, 2009); ISBN 0470496649; Pages 432
11. James W. Newkirk, Alexei A. Vorontsov; Test-Driven Development in Microsoft .NET; Microsoft Press; 1 edition (March 17, 2004); ISBN 0735619484; Pages 304
12. <http://msdn.microsoft.com/en-us/library/aa581776.aspx>
(How to create a Data Access Layer, Date – 05/05/2010)
13. [http://en.wikipedia.org/wiki/Three-tier_\(computing\)#Three-tier_architecture](http://en.wikipedia.org/wiki/Three-tier_(computing)#Three-tier_architecture)
(Concepts of Multitier Architecture, Date – 05/06/2010)
14. <http://msdn.microsoft.com/en-us/library/ff650258.aspx>
(Principles of Layered Application, Date – 05/07/2010)
15. <http://msdn.microsoft.com/en-us/library/ff648105.aspx>
(Concepts of Three-Layered services application, Date – 03/06/2010)
16. <http://msdn.microsoft.com/en-us/library/aa480455.aspx>
(Three-Tier Application Model, Date – 02/04/2010)
17. http://www.unm.edu/~network/presentations/course/appendix/appendix_k/sld051.htm
(Disadvantages of Two-Tier Architecture, Date – 02/04/2010)
18. <http://msdn.microsoft.com/en-us/library/ms973829.aspx>
(Types of Architecture for .NET Application, Date – 03/04/2010)

19. <http://www.destinationcrm.com/Articles/Web-Exclusives/Viewpoints/The-Flexibility-Revolution-in-Customer-Information-Architecture-45157.aspx>
(Three Tier architecture, Date – 06/21/2010)
20. <http://dotnetslackers.com/articles/net/IntroductionTo3TierArchitecture.aspx>
(Introduction to Three-Tier Architecture, Date – 07/23/2010)
21. <http://www2.bc.edu/~gallaugh/research/ism95/cccsa.html>
(Comparison of Two-Tier and Three-Tier Architecture, Date – 01/01/2010)
22. <http://blogs.msdn.com/b/nnaderi/archive/2007/02/01/mstest-vs-nunit-frameworks.aspx>
(Comparison of MSTest and NUnit frameworks, Date – 09/09/2010)
23. <http://xunit.codeplex.com/wikipage?title=Comparisons&ProjectName=xunit>
(Comparison of different testing frameworks, Date – 10/10/2010)
24. http://en.wikipedia.org/wiki/Test-driven_development
(Concepts of TDD, Date – 10/12/2010)
25. http://en.wikipedia.org/wiki/Unit_testing
(Concepts of Unit testing, Date – 12/5/2010)
26. <http://www.asp.net/security/videos/using-basic-forms-authentication-in-aspnet>
(Basic forms authentication in aspnet, Date – 12/20/2010)
27. <http://www.asp.net/security/tutorials/an-overview-of-forms-authentication-cs>
(Overview of forms authentication, Date – 09/10/2010)
28. <http://www.asp.net/security/tutorials/security-basics-and-asp-net-support-cs>
(Forms Authentication, Date – 01/02/2011)
29. <http://webdesign.about.com/od/css/a/aa102102a.htm>
(Div and tableless layout concepts, Date – 02/12/2011)
30. <http://www.regexlib.com/Search.aspx?k=integers>
(Regex to get positive integer numbers between 1 and 999, Date – 01/22/2011)
31. <http://msdn.microsoft.com/en-us/library/ms972969.aspx>
(SQL injection attacks, Date – 03/31/2011)
32. <http://www.yourhtmlsource.com/examples/positioning2.html>
(Div tableless layouts, Date – 04/01/2011)
33. http://www.regexlib.com/REDetails.aspx?regexp_id=463
(Regex for ISBN, Date – 01/10/2011)
34. http://en.wikipedia.org/wiki/International_Standard_Book_Number
(Wikipedia article for ISBN, Date – 12/28/2011)
35. <http://msdn.microsoft.com/en-us/library/bb398932.aspx>
(Understanding CSS on MSDN, Date – 04/12/2010)
36. <http://www.beansoftware.com/asp.net-tutorials/css-asp.net.aspx>
(Understanding why use CSS and <p> tag, Date – 01/01/2010)
37. [http://msdn.microsoft.com/en-us/library/tw292whz\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/tw292whz(v=VS.90).aspx)
(Managing Users by Using Membership, Date – 02/24/2010)

38. <http://msdn.microsoft.com/en-us/library/ff650303.aspx>
(How To: Use Regular Expressions to Constrain Input in ASP.NET, Date – 02/02/2011)
39. <http://social.msdn.microsoft.com/Forums/en/regexp/thread/9b3cb75e-ff6f-4d44-880d-dcbfdb81fdad>
(Used password validation regular expression from here, Date – 01/19/2010)
40. <http://blogs.msdn.com/b/webdevelopertips/archive/2009/05/27/tip-64-did-you-know-how-to-convert-a-gridview-column-from-asp-boundfield-to-asp-templatefield-in-design-view.aspx>
(How to convert a GridView column from asp boundfield to asp template field in design view, Date – 04/05/2011)
41. <http://www.asp.net/data-access/tutorials/using-templatefields-in-the-gridview-control.cs>
(Using templatefields in GridView control, Date – 07/08/2010)
42. <http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.hyperlinkcolumn.datanavigateurlfield.aspx>
(Learned about HyperLinkColumn.DataNavigateUrlField property, Date – 08/19/2010)
43. <http://stackoverflow.com/questions/2771154/membership-get-user-email>
(Learned to retrieve current logged user's email and other information, Date – 04/09/2011)
44. <http://msdn.microsoft.com/en-us/library/system.web.sessionstate.aspx>
(Using session variables in application, Date – 01/28/2010)
45. <http://forums.asp.net/p/1056610/1505014.aspx>
(Resize the size and height of GridView and FormView on webpage, Date – 01/28/2011)
46. <http://social.msdn.microsoft.com/forums/en-US/sqldatabaseengine/thread/88def9dd-5bf7-4156-b416-6db1c6752d48/>
(Enable User Instances in SQL Server, Date – 03/03/2011)
47. <http://articles.sitepoint.com/article/amazon-web-services-asp-net>
(Amazon Web Services, Date – 04/14/2011)
48. <http://www.collectorz.com/book/>
(Amazon web services to retrieve book information using ISBN, Date – 03/28/2011)
49. <http://www.sitemeter.com/>
(Comprehensive real time website tracking and counter tools give you instant access to vital information and data about your sites audience. With our detailed reporting you'll have a clear picture of who is visiting your site, how they found you, where they came from, what interests them and much more, Date – 01/02/2010)
50. <http://www.bookoogole.com/bookoogole/>

- (Somewhat similar website done by a student earlier but not operational, Date – 04/22/2010)
51. <http://docs.amazonwebservices.com/AWSEcommerceService/4-0/ApiReference/ItemLookupOperation.html#SampleRequest>
(Book Information-Item lookup Amazon by ISBN, Date – 04/06/2010)
 52. <http://msdn.microsoft.com/en-us/library/ms187787.aspx>
(CONTAINS SQL Query example, Date – 06/16/2010)
 53. <http://www.sql-tutorial.net/SQL-IN.asp>
(Using IN clause to do search, Date – 02/08/2011)
 54. <http://www.goldcoastwebdesigns.com/change-hyperlink-color.shtml>
(How to change color of Hyperlink, Date – 07/07/2010)
 55. <http://blog.sqlauthority.com/2007/09/07/sql-server-correlated-and-noncorrelated-subquery-introduction-explanation-and-example/>
(Non-correlated vs. Correlated queries, Date – 08/08/2010)
 56. <http://www.pluralsight-training.net/community/blogs/fritz/archive/2006/02/06/18596.aspx>
(Helped resolve folder url site.master conflicts <%= ResolveClientUrl("~/images/mypic.gif") %>), Date – 01/04/2011)
 57. <http://www.asp.net/master-pages/tutorials/urls-in-master-pages-vb>
(Urls in Master page, Date – 07/09/2010)
 58. <http://www.4guysfromrolla.com/articles/120705-1.aspx>
(Information about various username details such as how to display logged in username, email etc., Date – 01/11/2010)
 59. <http://www.zparacha.com/validate-email-address-using-javascript-regular-expression/>
(Used it to obtain Regex to validate .edu accounts for Email input box, Date – 01/27/2011)
 60. http://en.wikipedia.org/wiki/Software_quality
(Referred it while writing Supplementary specifications, Date – 09/09/2010)
 61. http://en.wikipedia.org/wiki/Use_case
(Concepts of Use-case, Date – 11/10/2010)
 62. <http://www.cragssystem.co.uk/SFRWUC/index.htm>
(Understanding Use-cases, Date – 05/21/2010)
 63. http://www.codeproject.com/KB/architecture/OOP_Concepts_and_manymore.aspx
(OOPs Concepts, Date – 02/07/2011)
 64. http://en.wikipedia.org/wiki/Functional_requirement
(Functional Requirements Concepts, Date – 11/22/2010)
 65. <http://www.cs.sjsu.edu/~pearce/modules/labs/ooa/dmcd/visio.htm>
(Learned how to use Visio-create folder structures and Use-case)
 66. <http://www.agileea.com/Whitepapers/All3%20Business%20vs%20System%20Use%20Cases%20v1%209.pdf>

- (Difference between a System Use case and Business Use case, Date – 09/22/2010)
67. <http://msdn.microsoft.com/en-us/library/aa302171.aspx>
(Architecture and Application Modeling - How to use MS Visio tool, Date – 06/22/2010)
68. http://en.wikipedia.org/wiki/Multitier_architecture
(What is multi-tier architecture?, Date – 09/20/2010)
69. <http://www.uml-diagrams.org/sequence-diagrams.html#synchronous-call>
(Sequence Diagram explanation- synchronous, asynchronous calls, Date – 03/19/2011)
70. <http://stackoverflow.com/questions/2689576/is-n-tier-software-design-a-subset-of-soa>
(Difference between n tier and SOA, Date – 02/15/2011)
71. <http://office.microsoft.com/en-in/visio-help/create-a-uml-sequence-diagram-HP001208871.aspx>
(Explanation on each element of sequence diagram in MS Visio, Date – 11/11/2010)
72. <http://www.cs.sjsu.edu/~pearce/modules/lectures/uml/behavior/ActivityDiagrams.htm>
(Information about Activity Diagrams, Date – 05/05/2010)
73. <http://stackoverflow.com/questions/92869/nunit-vs-visual-studio-2008s-test-projects-for-unit-testing>
(NUnit vs. MSTest for Unit testing on Stackoverflow website, Date – 01/04/2011)
74. <http://forums.asp.net/p/1621799/4162200.aspx>
(How to do automated testing of ASP.NET Web Forms application, Date – 01/29/2011)
75. <http://www.asp.net/visual-studio-team-system/videos/introduction-to-unit-testing-with-team-system>
(Unit testing using team system or MSTest in Visual Studio 2008, Date – 01/31/2011)
76. [http://msdn.microsoft.com/en-us/library/ms182524\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/ms182524(v=VS.90).aspx)
(How to create and run a Unit test on MSDN, Date – 02/28/2010)
77. <http://msdn.microsoft.com/en-us/library/wtxbf3hh.aspx>
(ASP.NET Master Pages, Date – 03/31/2011)
78. http://www.w3schools.com/css/css_intro.asp
(Cascading Stylesheets, Date – 04/30/2010)
79. <http://msdn.microsoft.com/en-us/library/bb332381.aspx>
(Used ObjectDataSource Diagram from here, Date – 05/31/2010)