

**Online Book Store**  
**by**  
**Vamsi Krishna Mummaneni**

**A Report Submitted in partial fulfillment of the requirements of the degree of  
Master of Software Engineering**

**Kansas State University**  
**Manhattan**  
**Kansas**

**Committee Members**  
**Dr.Torben Amtoft (Major Professor).**  
**Dr. Dan Andresen.**  
**Dr. Mitchell L. Neilsen.**

## **Acknowledgements**

**I would like to thank my major professor Dr. Torben Amtoft who persuaded and continuously guided me during the whole course of my project. I would also like to thank Dr. Daniel Andresen and Dr. Mitchell L. Neilsen and for their assistance and insightful comments, and who willingly shared their expertise with me.**

**I sincerely acknowledge and thank my family members and friends who gave moral support for me from my childhood.**

## **Abstract**

**The main objective of the project is to create an online book store that allows users to search and purchase a book online based on title, author and subject. The selected books are displayed in a tabular format and the user can order their books online through credit card payment. Using this Website the user can purchase a book online instead of going out to a book store and wasting time.**

**There are many online book stores like Powell's, Amazon which were designed using Html. I want to develop a similar website using .NET, SQL Server.**

**Online Book store is an online web application where the customer can purchase books online. Through a web browser the customers can search for a book by its title or author, later can add to the shopping cart and finally purchase using credit card transaction. The user can login using his account details or new customers can set up an account very quickly. They should give the details of their name, contact number and shipping address. The user can also give feedback to a book by giving ratings on a score of five. The books are divided into many categories based on subject like Software, Database, English, Architecture etc.**

**The Online Book Store Website provides customers with online shopping through a web browser. A customer can, create, sign in to his account, place items into a shopping cart and purchase using his credit card details.**

**The Administrator will have additional functionalities when compared to the common user. He can add, delete and update the book details, book categories, member information and also confirm a placed order.**

**This application is developed using C#, ASP.NET programming language. The Master page, data sets, data grids, user controls are used to develop the Online Book store.**

## Table of Contents

### Section.1 Vision Document

1. Introduction	7
1.1 Purpose and Motivation	7
2. Project Overview	7
2.1 Background	7
3. Requirement Specifications	9
3.1 Main Requirements	9
3.2 Critical Use Case Requirements	9
3.3 Environment	11

### Section.2 Project Plan

1. Task Breakdown	15
1.1. Inception Phase	15
1.2. Elaboration Phase	15
1.3. Production Phase	16
2. Cost Estimate	16
2.1. COCOMO	16
3. Architecture Elaboration Plan	17
3.1. Vision Document 1.0 Revision	17
3.2. Project Plan 1.0 Revision	17
3.3. Formal Requirement Specification	18
3.4. Architectural Design	18
3.5. Test Plan	18
3.6. Formal Technical Inspection	18
3.7. Architecture Prototype	19

### Section 3.Architecture Design

1. Introduction	20
2. Architecture	21
3. Presentation Tier	23
4. Middle Tier	26
5. Data Tier	29
6. References	31

Section.4 Formal Requirement Specification 32

Section.5 Formal Technical Inspector Checklist

1. Introduction	37
2. Items to be Inspected	37
2.1. UML Diagrams	37
2.2. Formal Specification	38
3. Formal Technical Inspectors	38
4. Formal Inspection Checklist	39
5. Formal Technical Inspection Report	39

Section.6 Software Quality Assurance Plan

1. Purpose	40
2. Reference Documents	40
3. Management	40
3.1 Organization and Responsibilities	40
3.2 Tasks	40
4. Documentation	41
5. Standards, Practices, Conventions, and Metrics	42
6. Reviews and Audits	43
7. Test and Problem Reporting	44
8. Tools, Techniques, and Methodologies	44
9. Records collection, Maintenance, and Retention	44
10. Deliverables	44

Section.7. Test Plan

1. Test plan identifier	45
2. Introduction	45
3. Features to be tested	45
5. Approach	48
5.1 Unit testing	48
5.2 Load testing	49
5.3 System Testing	49
5.4 Performance Testing	49

6. Pass/fail criteria	49
7. Suspension criteria and resumption requirements	49
7.1 Suspension criteria	49
7.2 Resumption requirements	50
8. Test deliverables	50

## Section.8 Assessment Evaluation

1. Introduction	51
2. Manual Testing	51
3. Test Cases	52
3.1 User	52
3.2 Admin	53
4. Performance Testing	54
5. Observation	55
6. References	56

## Section.9 User Manual

59

## Section.10 References

76

# **Section.1 Vision Document**

## **1. Introduction**

### **1.1 Purpose and Motivation**

The main objective of the project is to create an online book store that allows users to search and purchase a book based on title, author and subject. The selected books are displayed in a tabular format and the user can order their books online through credit card payment. The Administrator will have additional functionalities when compared to the common user.

The motivation to create this project has many sources

- Interest to develop a good user friendly website with many online transactions using a database.
- To increase my knowledge horizon in technologies like .NET, SQL, CSS, HTML.
- To gain good experience in .NET before joining in a full time job.
- To gain expertise using Data Grid, Data Set, Data Table, Data Adapter and Data Readers.

## **2. Project Overview**

### **2.1 Background**

There are many online book stores like Powell's, Amazon which were designed using Html. I want to develop a similar website using .NET, SQL Server.

Online Book store is an online web application where the customer can purchase books online. Through a web browser the customers can search for a book by its title or author, later can add to the shopping cart and finally purchase using credit card transaction. The user can login using his account details or new customers can set up an account very quickly. They should give the details of their name, contact number and shipping address. The user can also give feedback to a book by giving ratings on a score of five. The books are divided into many categories based on subject Like Software, Database, English, Architecture etc.

This project has the following functionalities:

### **1) A Home page with product catalog**

This is the page where the user will be navigated after a successful login. It will display all the book categories and will have a search keyword option to search for the required book. It also includes some special sections like recommended titles, weekly special books.

### **2) Search**

A search by keyword option is provided to the user using a textbox .The keyword to be entered should be the book title.

### **3) Advanced Search**

Advanced search helps the user to search for a book based on Title, Author, Category and price range. All the books which match the particular search criteria and their total count will be displayed .From here the user can select a book and add to the shopping cart.

### **4) Book Description**

If the user would like to know details about a book he can click on the title from where he will be directed to a Book description page. It includes the notes on the book content and also a link to Amazon.com to get the book review.

### **5) User Voting**

The user can give rating to a book based on his interest. He can rate it by giving a score of five as Excellent, four for very good, three for good, two for regular and one for deficient. The final rating of a book will depend on all the individual user rating.

### **6) Shopping Cart**

The user can manage a shopping cart which will include all the books he selected. The user can edit, delete and update his shopping cart. A final shopping cart summary is displayed which includes all the items the user selected and the final total cost.

### **7) Managing user accounts**

Each user should have an account to access all the functionalities of website. User can login using login page and logout using the logout page. All the user sessions will be saved in the database.

### **8) Administration**

The Administrator will be provided with special functionalities like

- Add or delete a book category
- Add or delete a member
- Manage member orders.
- Add or delete a Credit Card type.



### 3. Requirement Specifications

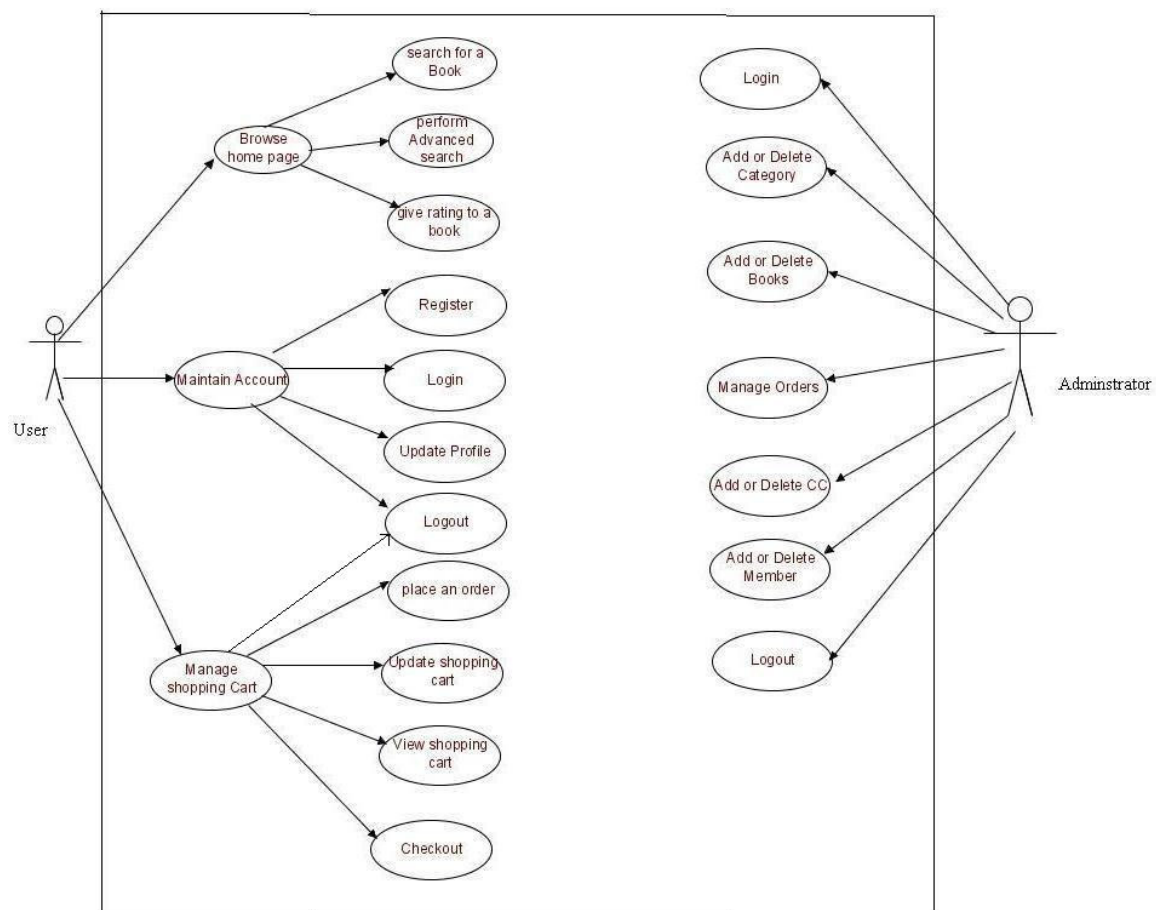
All the requirements are specified using OCL a software specification language in the second phase of my presentation.

#### 3.1 Main Requirements

The Main Requirements include Microsoft Visual Studio 2005 and ASP.NET to develop the web application, SQL Server 2005 to design the database and Mozilla as a main web browser to run the website.

#### 3.2 Critical Use Case Requirements

User and Administrator are the two actors included in the Online Book Store. Fig.2 shows the use case diagram for this website.



**Fig.2 Use Case Diagram**

## Use Cases:

### Browse Catalog

#### 1) Search for a Book

- **Purpose:** A user can search for a book of his choice by selecting category and title. Then a select query is used to retrieve data from the database and display the selected information.
- **Actor:** User
- **Input:** The user will select a category and enter title in a text box provided.
- **Output:** The system will display the books which matches the selected search criteria. A dataset is created as a result of select query. Later the dataset is binded to the data repeater to display the selected data.

#### 2) Perform Advanced Search

- **Purpose:** If the user wants to perform an advanced search he can search for a book of his choice by selecting category, title, author and price range. Then a select query is used to retrieve data from the database and display the selected information.
- **Actor:** User
- **Input:** The user will select a category and enter title, author, and price range in a text box provided.
- **Output:** The system will display the books which matches the selected search criteria. A dataset is created as a result of select query. Later the dataset is binded to the data repeater to display the selected data.

#### 3) Give rating to a book

- **Purpose:** If the user wants to give rating according to his opinion for a book he can select either Excellent, Very good, good, regular or deficient. The final rating of a book will depend on all the individual user rating.
- **Actor:** User
- **Input:** The user will select a rating based on his opinion.
- **Output:** The system will display the rating of a book and the total number of votes received. Below is the display for various rating.
  - \*\*\*\*\* Excellent
  - \*\*\*\* Very Good
  - \*\*\* Good
  - \*\* Regular
  - \* Deficient

## Maintain Account

### 1) Register

- **Purpose:** If the user doesn't have an account then he will be asked to register.
- **Actor:** User
- **Input:** The user will enter details in the registration form according to the required fields. The fields include
  1. Username
  2. Password
  3. confirm password
  4. first name
  5. last name
  6. email
  7. Address
  8. Phone
  9. CC details
- **Output:** After registration the user will be directed to the main home page.

### 2) Login

- **Purpose:** If the user wants to get access to all the functionalities of Online Book Store he should login using his username and password.
- **Actor:** User
- **Input:** The user will enter his username and password.
- **Output:** If it is a successful login the user will be directed to the main home page. Else if the user enters invalid information he will be asked to check the entered information.

### 3) Update Profile

- **Purpose:** If the user wants to change his personal account information then he can update his selected fields and the entire data will be updated in the data base through an update query.
- **Actor:** User
- **Input:** The user will update his account information.
- **Output:** The system will update the entered information in the database using an update query.

### 4) Logout

- **Purpose:** If the user wants to end his session and sign out of the website then he can use the logout option.
- **Actor:** User

- **Input:** The user will click the logout button.
- **Output:** The user's account session comes to an end and he should login again if he wants to enter into the website.

## Manage Shopping Cart

### 1) Place an order

- **Purpose:** If the user wants to purchase a book then he can place an order by selecting the add to shopping cart button and entering the quantity required under the book description.
- **Actor:** User
- **Input:** The user will enter the quantity required and click the add to shopping cart button.
- **Output:** The order will be added to the user's shopping cart.

### 2) Update Shopping Cart

- **Purpose:** If the user wants to change the quantity of a book or change a book then he can update his shopping cart.
- **Actor:** User
- **Input:** The user will click the details button in the shopping cart summary to edit and update his order details..
- **Output:** The updated order details are reflected in the shopping cart summary.

### 3) View Shopping Cart

- **Purpose:** If the user wants to view the items he added to the shopping cart then he can click the shopping cart link at the top of the page.
- **Actor:** User
- **Input:** The user will click the shopping cart link at the top of every page.
- **Output:** The user's shopping cart summary will be displayed in the form of a tabular format with all the books and their quantity. A total cost of all the items is also displayed at the bottom.

## Administrator

### 1) Login

- **Purpose:** If the Administrator wants to get access to all the functionalities of Online Book Store he should login using his username and password.
- **Actor:** Administrator
- **Input:** The Administrator will enter his username and password.

- **Output:** If it is a successful login the Administrator will be directed to his menu page. Else if the Administrator enters invalid information he will be asked to check the entered information.

## 2) Add or Delete Category

- **Purpose:** If the Administrator wants to add or delete a book category then he can insert or delete a book category using his administration rights and the category table will be updated in the database.
- **Actor:** Administrator
- **Input:** If the Administrator wants to add a book category the he should click the insert link button in the category page else he can delete a particular selected book category.
- **Output:** The updated categories list will be displayed in the main home page.

## 3) Add or Delete Book

- **Purpose:** If the Administrator wants to add or delete a book then he can insert or delete a book using his administration rights and the book table will be updated in the database.
- **Actor:** Administrator
- **Input:** If the Administrator wants to add a book the he should click the insert link button in the book page and fill the following fields related to the book.
  1. Title
  2. Author
  3. Price
  4. Category
  5. Notes
  6. Product url

If he wants to delete a book he can click the delete button to remove it from the database.

- **Output:** The updated books list will be displayed in the main home page under their particular category.

## 4) Manage Orders

- **Purpose:** If the Administrator wants to add or delete an order then he can insert or delete an order using his administration rights.
- **Actor:** Administrator
- **Input:** If the Administrator wants to add an order the he should click the insert link button in the orders page else he can delete a particular selected order
- **Output:** The updated orders list will be processed to the users.

### 5) Add or Delete CC (Credit Card)

- **Purpose:** If the Administrator wants to add or delete a CC type then he can insert or delete a CC type using his administration rights and the CC table will be updated in the database.
- **Actor:** Administrator
- **Input:** If the Administrator wants to add a CC type the he should click the insert link button in the CC page else he can delete a particular selected CC type..
- **Output:** The updated CC list will be displayed in registration page where the user will select it for his future transactions.

### 6) Add or Delete Member

- **Purpose:** If the Administrator wants to add or delete a book category then he can insert or delete a book category using his administration rights and the category table will be updated in the database.
- **Actor:** Administrator
- **Input:** If the Administrator wants to add a book category the he should click the insert link button in the category page else he can delete a particular selected book category.
- **Output:** The updated categories list will be displayed in the main home page.

### 7) Logout

- **Purpose:** If the Administrator wants to end his session and sign out of the website then he can use the logout option.
- **Actor:** Administrator
- **Input:** The Administrator will click the logout button.
- **Output:** The Administrator's account session comes to an end and he should login again if he wants to enter into the website.

### 3.3 Environment

- The Online Book store will be developed in Visual Studio 2005 environment.
- C# will be used as the programming language.

## **Section.2 Project Plan**

### **1. Task Breakdown**

#### **1.1. Inception Phase**

The inception phase will define all the project's requirements. This phase will include the production of a Vision Document, a Project Plan, a Software Quality Assurance Plan, and a Demonstration.

Vision Document will include the project's requirements and overview. It includes overview of the project, its purpose, goals, risks, constraints, and direction. It gives a listing of the main requirements and their respective Use case models to illustrate the functionality. Project Plan will detail the phases, iterations, and milestones that will comprise the project. It will include a timeline for the project and a cost estimate for completing this project. It includes the Architecture Elaboration plan will define the activities and actions that must be accomplished before the Architecture Presentation.

Software Quality Assurance Plan describes the required documentation, standards and conventions test tracking and problem reporting, and tools used during the project. The plan will also identify the set of quality metrics used to assess product reliability.

Demonstration of at least one executable prototype is required. Projects with a graphical user interface will include an executable prototype of the user interface.

This phase will be complete once the supervisory committee has approved all the above work.

#### **1.2. Elaboration Phase**

The elaboration phase defines the project's architecture. This phase will include the production of revisions to the Project Plan and the Vision Document, an Architecture Design Plan, a formal specification, Test Plan, Formal Technical Inspection and Architecture Prototype.

Revision of Vision Document will be an updated version to provide a complete representation of all requirements. These requirements will be ranked according to importance, and a set of critical requirements identified. Appropriate changes that were suggested by the committee at the end of phase one will also be updated in the updated version of Vision Document. Revision of Project Plan will include updated timeline and cost estimate for the project. It also includes The Implementation plan which will define the activities and actions that must be accomplished during implementation. Formal Specification will include OCL. Architecture Design Plan will include the complete architectural design documentation using appropriate diagrams such as class diagrams, sequence/collaboration diagrams, etc. It also includes documentation of reuse of commercial or pre-existing components. Test Plan will address the required tests to show that the product satisfies the requirements. The test plan will identify a set of test cases,

the types of tests that will be used for these test cases, the data that will be used for each case, and the requirement traces for each test case. Formal Technical Inspection will include inspections by two MSE students. One of the designs, formal requirement or executable prototype is subjected to inspection. Architecture Prototype will address all critical requirements identified in the vision document.

This phase will be complete once the supervisory committee has approved all the above work.

### **1.3. Production Phase**

The production phase defines the project implementation and testing. This phase includes the user manual, component design, assessment evaluation, project evaluation, references, and formal technical inspection.

User Manual includes an overview and explanations of common usage, user commands, error messages, and data formats. Component Design The internal design of each component will be documented using sequence/collaboration diagrams and state chart/activity diagrams. Source Code which corresponds to architecture and component design will be submitted. Assessment Evaluation will include a document detailing the testing done on the project. And Project Evaluation includes evaluation of the project ideas and quality. References and Formal Technical Inspection Letters will also be documented.

This phase will be complete once the supervisory committee has approved all the above work.

## **2. Cost Estimate**

The Cost Estimate is done using the COCOMO model.

### **2.1. COCOMO**

Project effort and time will be estimated using the COCOMO estimation model (Barry Boehm).

The Online Book store has an average complexity and fair flexibility. Therefore, it is classified as an organic mode project under the COCOMO model. The following formula is the COCOMO model for cost estimation for organic mode projects:

$$\text{Effort} = 3.2 * \text{EAF} * (\text{Size}) ^ 1.05$$

$$\text{Time} = 2.5 * (\text{Effort}) ^ 0.38$$

Where            Effort = number of staff months (PM)

                  EAF = effort adjustment factor

                  Size = number of lines of code for completed product. It is measured in KLOC (thousands of lines of codes)

                  Time = total number of months.



The Effort Adjustment Factor is the product of the 15 adjustment parameters. Each adjustment parameter is categorized as very low, low, nominal, high, or very high. All the adjustment parameters are listed below:

- RELY Required reliability 0.75 – 1.40
- DATA Database size 0.94 – 1.16
- CPLX Product complexity 0.70 – 1.65
- TIME Execution time constraint 1.00 – 1.66
- STOR Main storage constraint 1.00 – 1.56
- VIRT Virtual machine volatility 0.87 – 1.30
- TURN Computer turnaround time 0.87 – 1.15
- ACAP Analyst capability 1.46 – 0.71
- AEXP Applications experience 1.29 – 0.82
- PCAP Programmer capability 1.42 – 0.70
- VEXP Virtual machine experience 1.21 – 0.90
- LEXP Language experience 1.14 – 0.95
- MODP Use of modern practices 1.24 – 0.82
- TOOL Use of software tools 1.24 – 0.83
- SCED Required development schedule 1.23 – 1.10

Adjustment factors for the Online Book Store are listed below:

- RELY 1.00 Nominal
- DATA 1.00 Nominal
- CPLX 0.85 Low
- TIME 1.00 Nominal
- STOR 1.00 Nominal
- VIRT 0.87 Low
- TURN 0.87 Low
- ACAP 1.00 Nominal
- AEXP 1.13 Low
- PCAP 1.00 Nominal
- VEXP 1.00 Nominal
- LEXP 1.00 Nominal
- MODP 0.91 High
- TOOL 0.91 High
- SCED 1.00 Nominal

The EAF value evaluated to 0.60. I have estimated the size to be around 3.00.

From the calculation I got

EFFORT = 6.08

TIME = 4.96

### **3. Architecture Elaboration Plan**

All the activities to be done before the Architecture Presentation are defined here.

#### **3.1. Vision Document 1.0 Revision**

Vision Document 1.0 will be revised and updated to Vision Document 2.0. The revisions will contain a complete representation of project requirements and these requirements will be ranked according to importance, and a set of “critical” requirements identified. It also contain changes suggested by the committee members following presentation one. This revision will be approved by the Major Professor.

#### **3.2. Project Plan 1.0 Revision**

Project Plan 1.0 will be revised and updated to Project Plan 2.0. The revisions will contain Implementation plan. The Implementation plan will define the activities and actions that must be accomplished during implementation. It also contain changes suggested by the committee members following presentation one. This revision will be approved by the major professor.

#### **3.3. Formal Requirement Specification**

One part of the project will be formally specified using OCL. The specification will represent the formal requirements of the project, described in the Vision Document. This will be approved by the major professor.

#### **3.4. Architectural Design**

Architectural Design will be documented using UML diagrams such as class, sequence, and state chart diagrams. All architectural components will be documented at interface level. This will be approved by the major professor.

#### **3.5. Test Plan**

Test Plan will be developed. The document will follow the requirement listings found in Vision Document 2.0. It will also identify a set of test cases, the types of tests that will be used for these test cases, the data that will be used for each case, and the requirement traces for each test case. This will be approved by the major professor.

#### **3.6. Formal Technical Inspection**

Two MSE students will act as technical inspectors. The technical inspection will assess the project architecture. It will also include a formal checklist to be used by the inspectors. This will be approved by the major professor.

### **3.7. Architecture Prototype**

An executable prototype will be built including all critical requirements described in the vision document.

## Section.3 Architecture Design

### 1. Introduction

The purpose of this document is to provide an architectural design for the Online Book Store. The design will show the presentation tier, the middle tier consisting of classes, sequence diagrams, and the data tier consisting of the database design diagram.

### 2. Architecture

Three-tier (layer) is a client-server architecture in which the user interface, business process (business rules) and data storage and data access are developed and maintained as independent modules or most often on separate platforms.

The Architecture of Online Book Store is based on three-tier architecture. The three logical tiers are

- Presentation tier - ASP.NET Web forms, Master Pages, Images.
- Middle tier – C# classes.
- Data tier- Database

Fig.1 below shows the model of 3-tier architecture.

The main reason for considering three-tier architecture for the Online Book store is as follows:

#### ***Flexibility:***

- Management of data is independent from the physical storage support,
- Maintenance of the business logic is easier,
- Migration to new graphical environments is faster.
- If there is a minor change in the business logic, we don't have to install the entire system in individual user's PCs.

#### ***Reusability:***

- Reusability of business logic is greater for the presentation layer. As this component is developed and tested, we can use it in any other project and would be helpful for future use.

## Security:

- More secured architecture since the client cannot access the database directly.

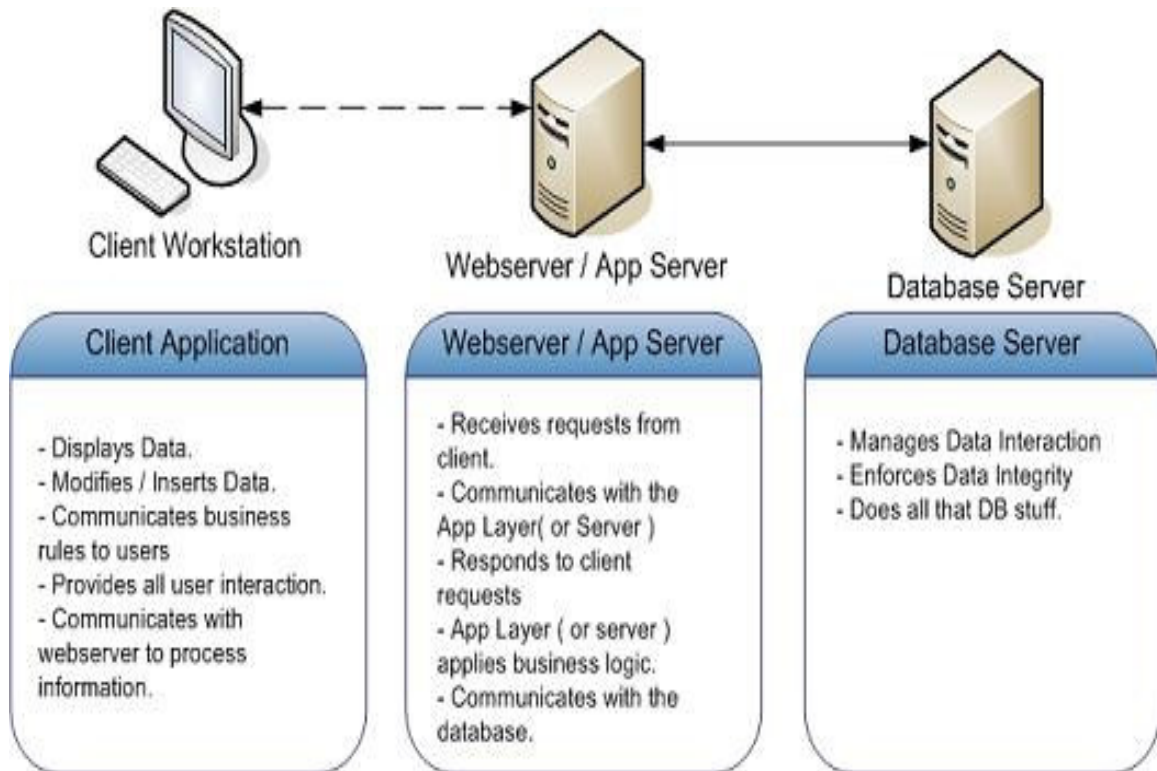


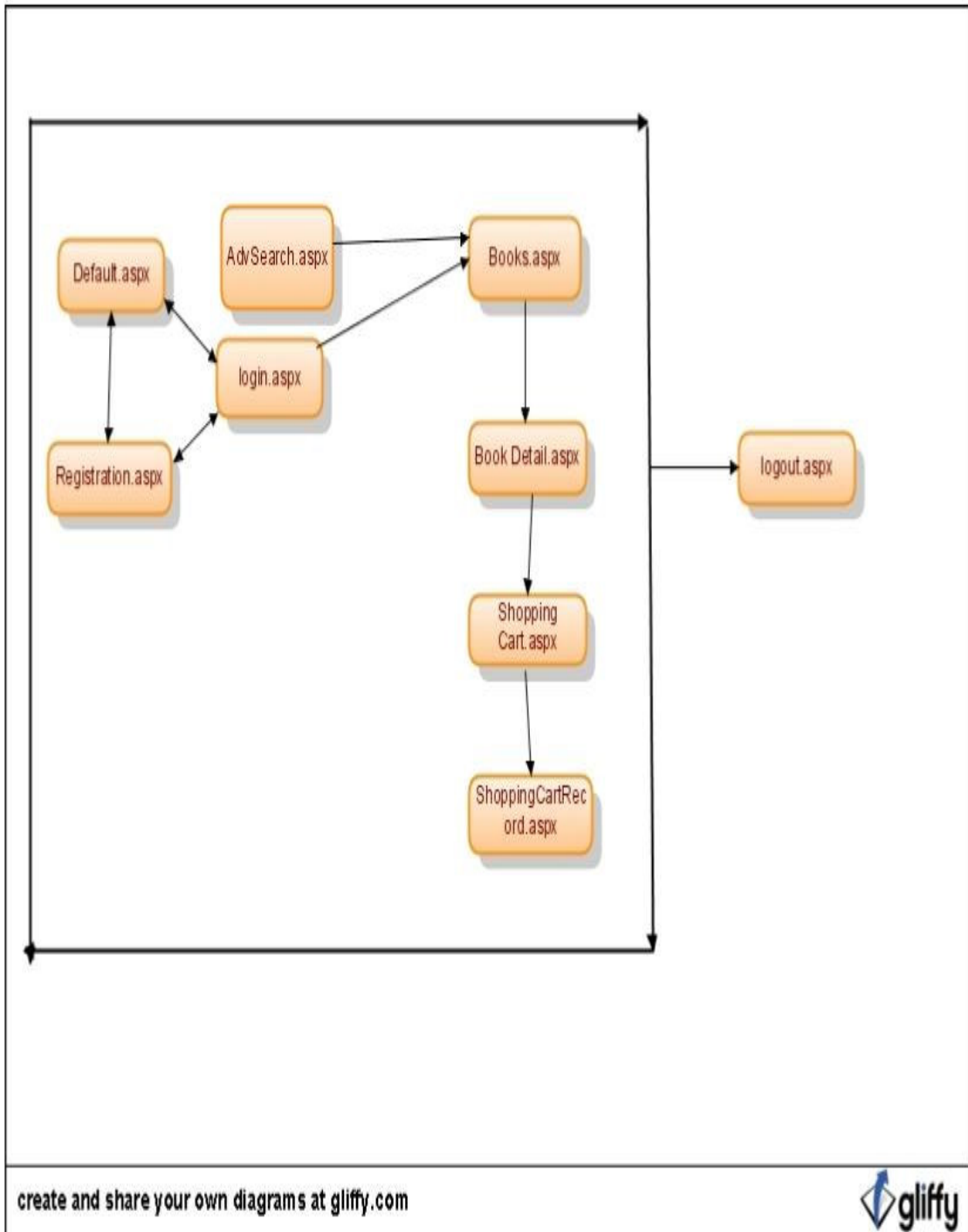
Fig.1 3-tier Architecture

### 3. Presentation tier

The presentation tier for the Online Book Store is ASP.NET Web Forms with Master Pages. The presentation tier takes care of the user interaction and the visual part of the website. The various Web forms used by clients are listed below.

<b>ASP.NET Web Forms</b>	<b>Definition</b>
Default.aspx	This web page is used to browse the entire website and also displays different categories, offers available
Registration.aspx	This web page is used by a new user to register his personal information.
Login.aspx	This web page is used for a user to login and enter the website.
Logout.aspx	This web page is used for a user to logout from the website
Books.aspx	This web page is used to search for books and displays books which matched the search criteria..
Advsearch.aspx	This web page is used to display an advanced search to find books.
BookDetail.aspx	This web page is used for displaying all the details of a book.
ShoppingCart.aspx	This web page is used to display the shopping cart and user information details.
ShoppingCartRecord.aspx	This web page is used to edit shopping cart.

The diagram below captures the page flow for user in the Online Book Store System.



**Fig.2 User-Page Flow**

The various Web forms used by Administrator are listed below.

<b>ASP.NET Web Forms</b>	<b>Definition</b>
Login.aspx	This web page is used for an Administrator to login and enter the website.
AdminMenu.aspx	The webpage displays the menu of the Administrator
MembersGrid.aspx	The web page displays the members list
MembersInfo.aspx	The web page displays the members personal information
MembersRecord.aspx	The web page is used to edit and update members information
OrdersGrid.aspx	The web page displays the orders list.
OrdersRecord.aspx	The web page is used to edit and update orders information
AdminBooks.aspx	The web page displays the books list.
BookMaint.aspx	The web page is used to edit and update book information
CategoriesGrid.aspx	The web page displays the members list
CategoriesRecord.aspx	The web page is used to edit and update Categories information
CardtypesGrid.aspx	The web page displays the members list
CardtypesRecord.aspx	The web page is used to edit and update Credit Card information
Logout.aspx	This web page is used for an Administrator to logout from the website

The diagram below captures the page flow for Administrator in the Online Book Store System



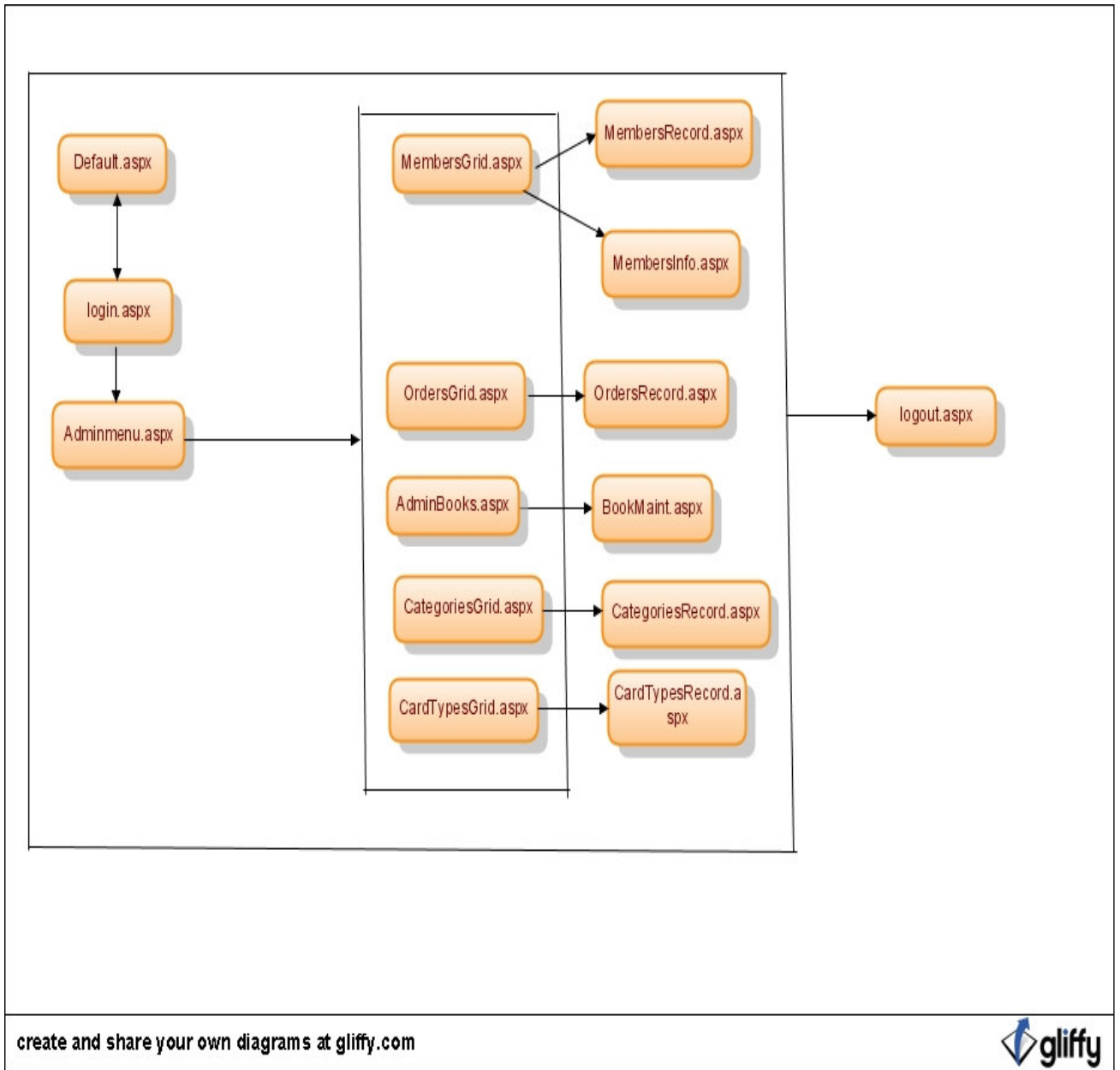


Fig.3 Administrator-Page Flow

#### 4. Middle Tier

The Middle Tier or Business Logic layer consists of eleven classes User, Customer, Adminstrator, SessionManager, Category, BookSet, Book, BooksOrder, Search, Advanced Search and Shopping Cart.

#### Class Diagram

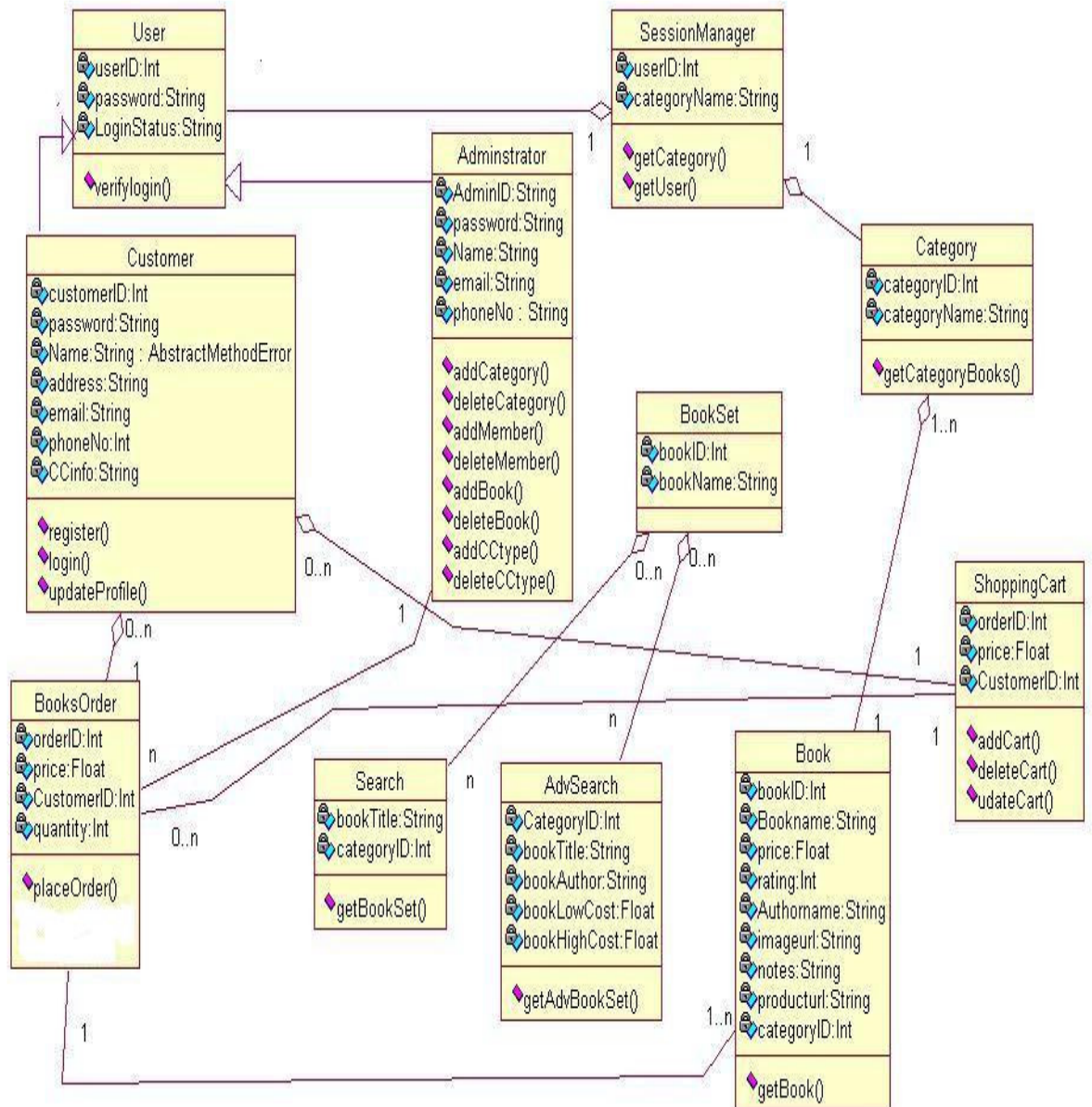


Fig.4 Class Diagram

## Sequence Diagrams

### 1) User Login

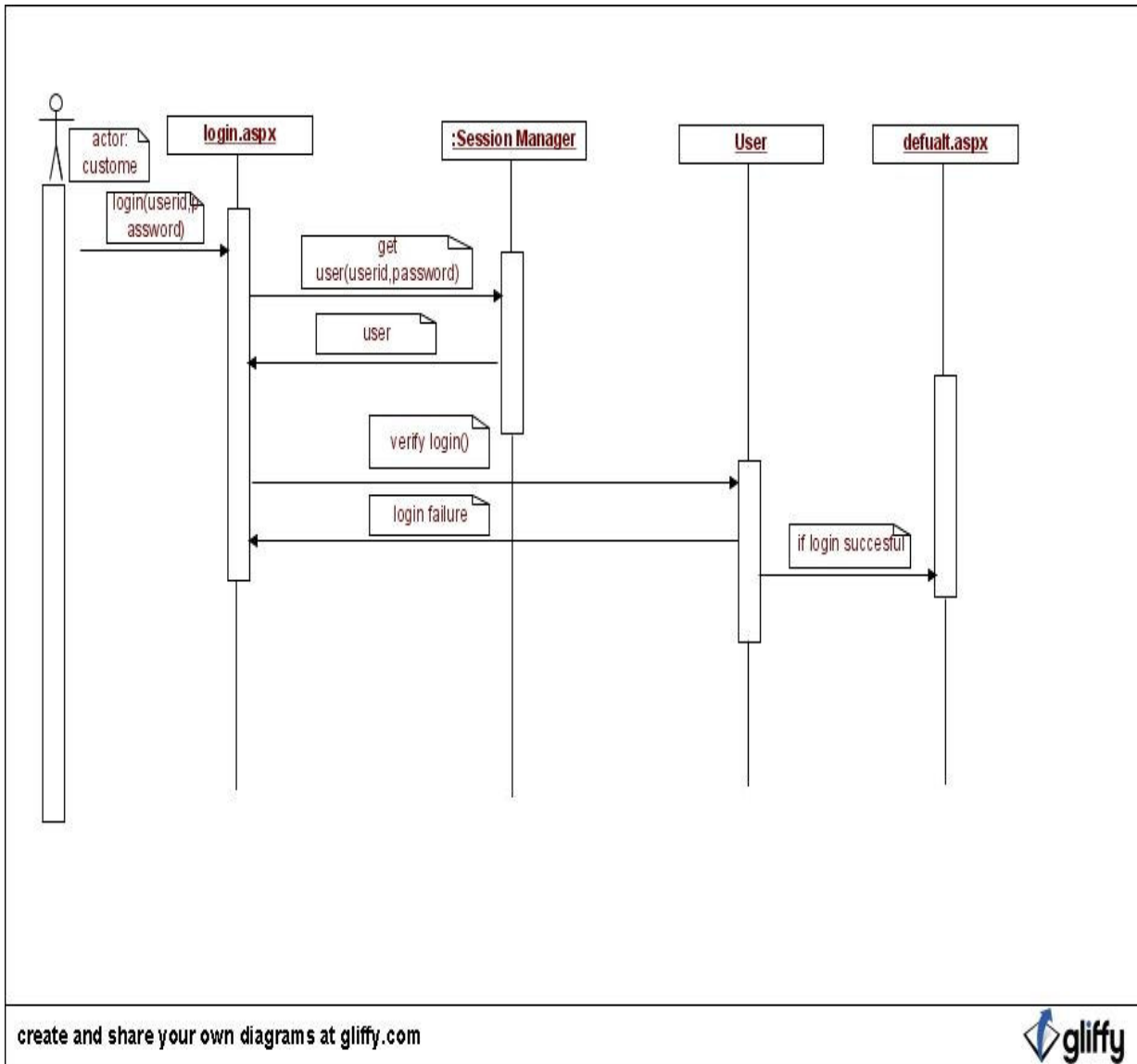


Fig.5 User-Login Sequence Diagram

## 2) Book Search

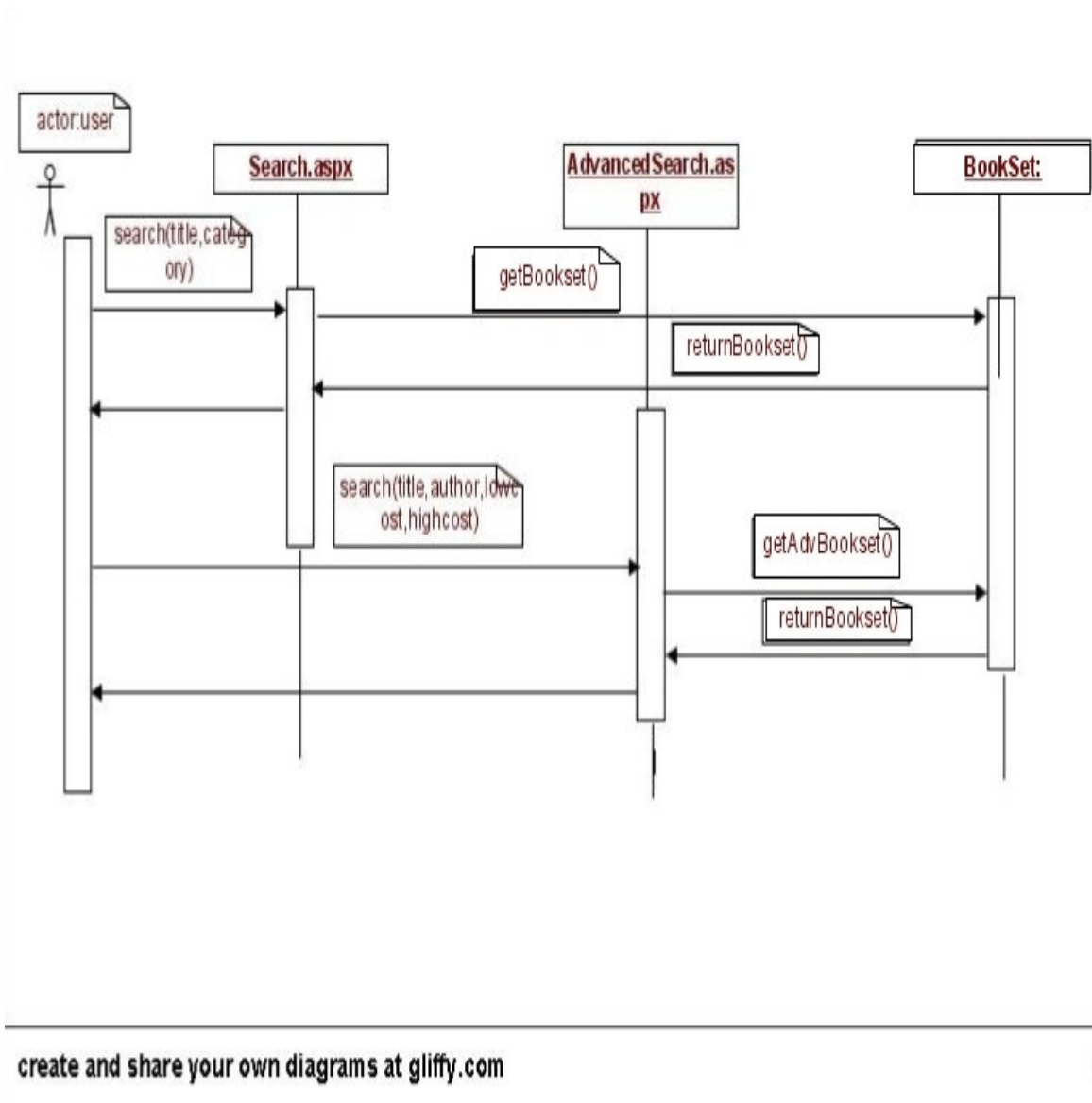


Fig.6 User-Book Search Sequence Diagram

### 3) Add to shopping cart

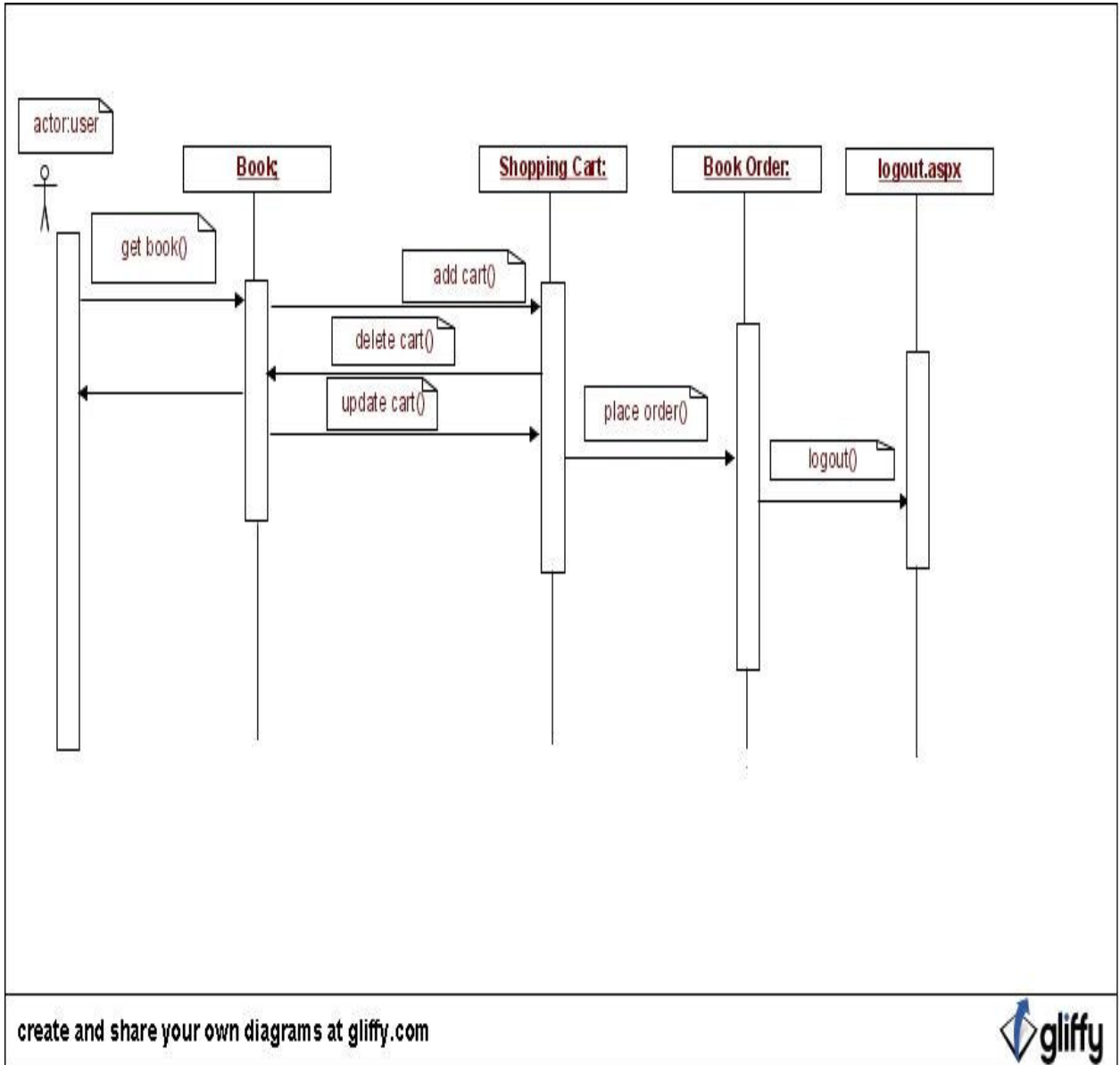


Fig.8 User-Add to Cart Sequence Diagram

#### 4) Administrator

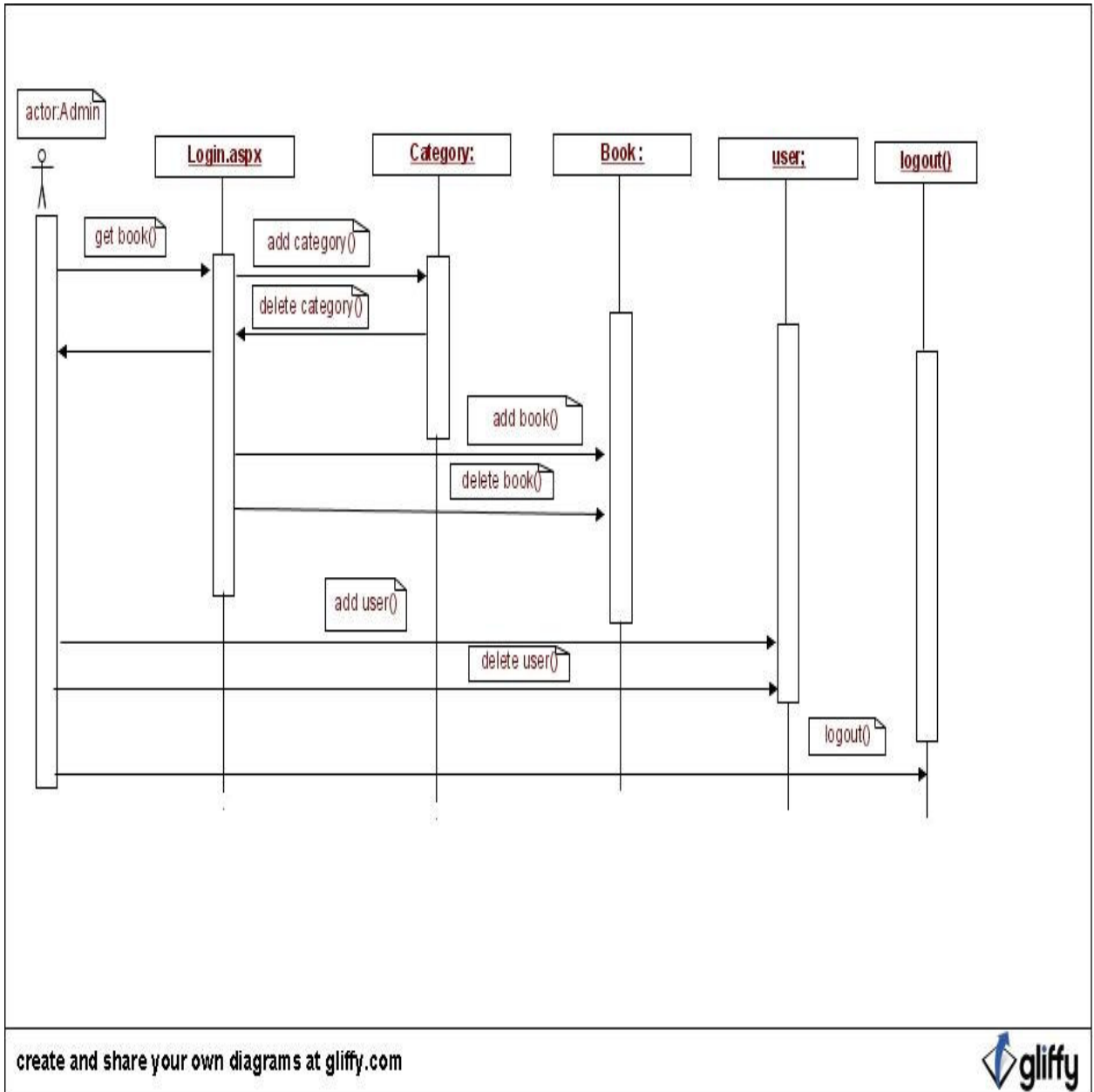


Fig.9 Administrator- Sequence Diagram

#### 4. Data Tier

The system database has five tables Categories, Items, Orders, Members and Card types. The system database design is shown below.

Table Name	Definition
Categories	Contains the Book Categories Information
Items	Contains the Book Information.
Members	Contains the Members Information
Orders	Contains the Book Orders Information
Card Types	Contains the Credit Card Information

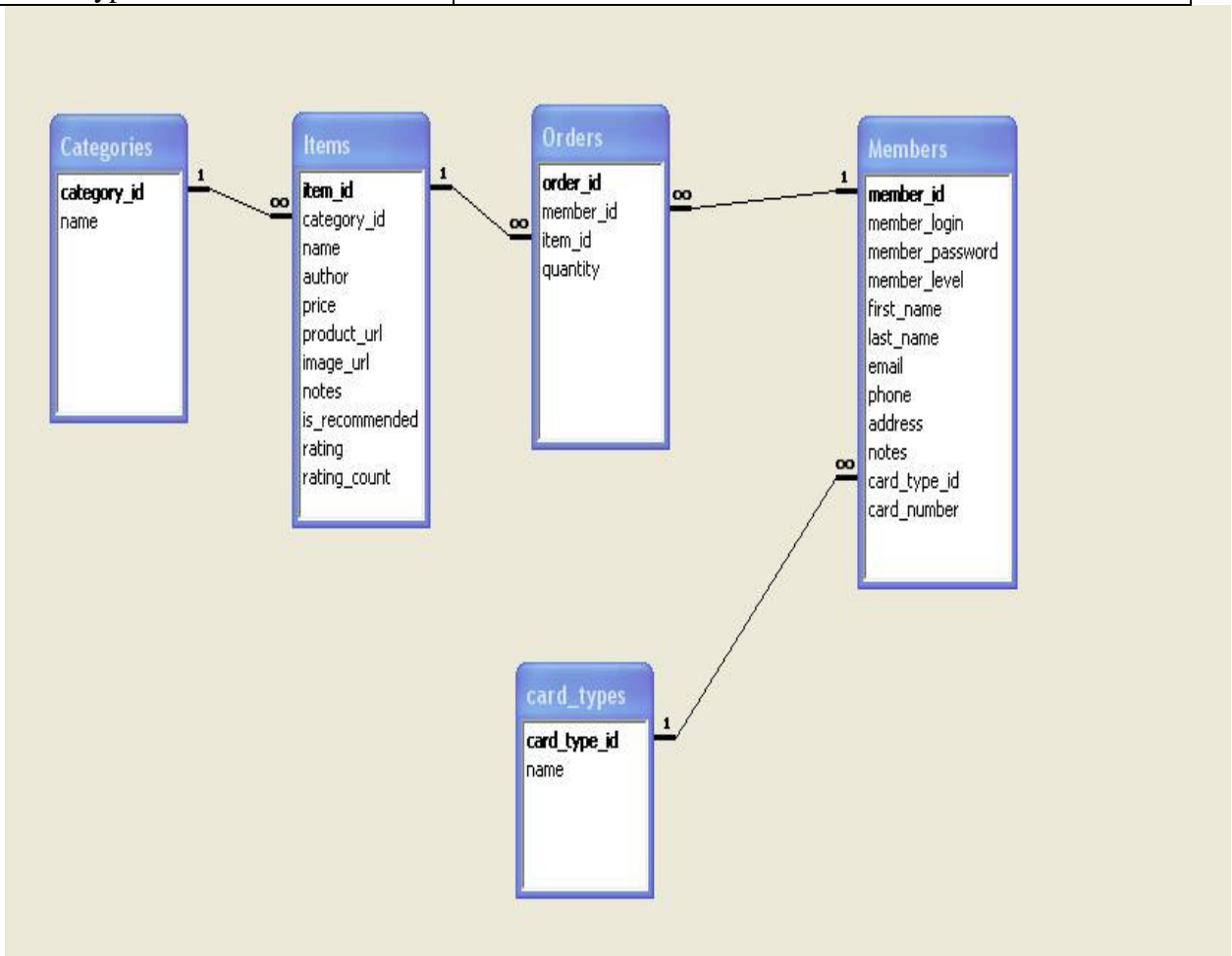


Fig.10 System Database Design

## Section.4 Formal Requirement Specification

model OnlineBookstore

-- Classes

```
class User
attributes
UserID: string
password: string
LoginStatus: string
```

```
operations
Verifylogin (): Boolean
end
```

```
class Administrator < User
attributes
AdminID: string
password: string
Name: string
email: string
phoneNo: integer
```

```
operations
addCategory(): Boolean
deleteCategory: Boolean
addMember(): Boolean
deleteMember(): Boolean
addBook(): Boolean
deleteBook(): Boolean
addCCtype(): Boolean
deleteCCtype(): Boolean
```

end

```
class Customer < User
attributes
customerID: string
password: string
Name: string
address: string
email: string
phoneno: integer
CCInfo: string
```

```
operations
register(): Boolean
login(): Boolean
--updateprofile is used to update user information
updateProfile(customerID: string, Name: string, address: string, email: string,
phoneno: integer, CCInfo: string): Boolean
```



```

--Customer already has an customerID
pre: Customer.allInstances.customerID->includes (customerID)
post: Customer.allInstances.customerID = user.allInstances.customerID@pre
--Customer name has been created
post: Customer.allInstances.Name =user.allInstances->select(C:Customer | C.customerID
<>customerID).Name@pre->includes (Name)
--Customer address has been created
post: Customer.allInstances.email= Customer.allInstances->select(C:Customer
| C.customerID<>customerID).email@pre->includes(email)
post: Customer.allInstances.address =
Customer.allInstances->select(C:Customer |
C.customerID<>customerID).address@pre->includes(address)
--Customer Phoneno has been created
post: Customer.allInstances.phoneno =
Customer.allInstances->select(C:Customer |
C.customerID<>customerID).phoneno@pre->includes(phoneno)
post: Customer.allInstances.CCInfo =
Customer.allInstances->select(C:Customer |
C.customerID<>customerID). CCInfo @pre->includes(CCInfo)

```

end

```
class Category
```

```
attributes
```

```
categoryID:integer
```

```
categoryName:string
```

```
operations
```

```
getCategoryBooks(bookID:int):Set(Book) =
```

```
Book.allInstances->select(b:Book| b. bookID = bookID)
```

```
end
```

```
class Book
```

```
attributes
```

```
bookID:integer
```

```
categoryID:integer
```

```
bookName:string
```

```
authorName:string
```

```
notes:string
```

```
price:float
```

```
imageurl:string
```

```
producturl:string
```

```
rating:int
```

```
operations
```

```
getBook():Boolean
```

```
end
```

```
class ShoppingCart
```

```
attributes
```

```
orderID:integer
```

```
customerID:integer
```

```
price:float
```

```
operations
addCart():Boolean
deleteCart():Boolean
updateCart():Boolean
```

```
end
```

```
class BookOrder
attributes
orderID:integer
customerID:integer
price:float
quantity:integer
operations
placeOrder(BO:BookOrder):Boolean
--User authentication is verified
pre: BO.User. Verifylogin (BO.User. UserID, BO.User.password)=true
pre: BookOrder.allInstances->excludes(BO)
--Check whether customerid and customerid in Order is same
post: Customer.allInstances -> forAll(C:Customer | C.customerID=BO.customerID
implies BO.orderID = C.orderID)
post: BookOrder.allInstances.orderID =
BookOrder.allInstances.orderID@pre->includes(BO.orderID)
end
```

```
class Search
attributes
bookTitle:String
categoryID:integer
operations
getBookset():Boolean
end
```

```
class AdvSearch
attributes
bookTitle:String
categoryID:integer
bookAuthor:String
bookLowCost:float
bookHighCost:float
operations
getBooksetbyAdv():Boolean
end
```

```
class BookSet
attributes
bookID:Int
bookName:String
end
```

```
-- Assosiations
```

```
--Each book should belong to only one Category
```

```
association bookCategory between
Category[1] role subCategory
Book[1..*] role allBook
end
```

```
--Each BooksOrder should contain atleast one Book
association BooksOrderHasBook between
BooksOrder[1] role theBookOrder
Book[1..*] role theBook
end
```

```
--Each BooksOrder should belong to exactly one Customer
association CustomerHasOrder between
Customer[1] role belongstocustomer
BooksOrder[0..*] role thecustomerbook
end
```

```
--Each Shopping cart should belong to only one Customer
association CustomerrelatedtoShoppingCart between
Customer[1] role thecustomer
ShoppingCart[0..*] role thecart
end
```

```
--Each shoppingcart should have atleast one BooksOrder
association ShoppingCartHasOrder between
ShoppingCart[1] role thebookCart
BooksOrder [1..*] role theorder
end
```

```
--Each search should result some bookset
association searchhassomebookset between
Search[1] role thesearch
BookSet[0..*] role thesearchset
end
```

```
--Each Advsearch should result some bookset
association Advsearchhassomebookset between
AdvSearch[1] role theAdvsearch
BookSet[0..*] role theAdvsearchset
end
```

```
--Constraints
```

```
-- Each user should have different userID
context User
inv distinctuserID:
User.allInstances -> forAll(user1, user2 |user1 <> user2 implies user1.userID <>
user2.userID)
```

```

-- Each Book should belong to exactly one Category
context Category
inv BookHasoneCategory:
Category.allInstances -> forAll(C1,C2 |C1<>C2 && C1. getCategoryBooks ->
includes(book) implies C2. getCategoryBooks -> excludes(book))

--Each Book should have a different bookID
context Book
inv DistinctBookID:
Book.allInstances -> forAll(B1, B2 |B1 <> B2 implies B1.bookID<>
B2.bookID)

--The OrderID for each Order must be different
context BookOrder
inv DistinctOrderID:
BookOrder.allInstances -> forAll(BO1, BO2 |BO1 <> BO2 implies BO1.orderID<>
BO2.orderID)

--Each BookOrder should have some books
context BookOrder
inv BookOrderHasbooks
self.contains -> notEmpty()

-- Each BookOrder belongs to exactly one customer
context BookOrder
inv OrdertoOneCustomer
Order.allInstances -> forAll (BO1, BO2 | BO1.orderID <> BO2.orderID
implies BO1.customerID <> BO2.customerID)

-- Quantity should always be a positive value
context BookOrder
inv BookOrder Positive
self.quantity > 0

--Each Shopping cart belongs to only one customer
context ShoppingCart
inv CarthasOneCustomer
ShoppingCart.allInstances -> forAll (SC1, SC2 | SC1.orderID <> SC2.orderID
implies SC1.customerID <> SC2.customerID)

--In search Low price should be less than High Price
context AdvSearch
inv PriceCompare
self. bookLowCost < self. bookHighCost

-- Price should always be a positive value
context Book
inv BookPricePositive:
self.price > 0

```

## Section.5 Formal Technical Inspectors Checklist

### 1. Introduction

The document specifies checklist to be inspected by the technical inspectors of Online Book Store. The main purpose of the formal technical inspection process is to ensure quality of the software design. Two MSE students will perform the inspection and provide the formal report on the result of their inspection.

### 2. Items to be inspected

#### 2.1. UML Diagrams

- Class diagrams
- Sequence diagrams

#### 2.2. Formal Specification

- USE Model

### 3. Formal Technical Inspectors

- Phaninder Surapaneni
- Snehal Monterio

### 4. Formal Inspection Checklist

Item to be Inspected	Pass/Fail/Partial	Comments
All the symbols used in the class diagrams are according to the UML standards.		
All the classes in the class diagram are clear as to what they represent in the architecture design document.		
The symbols used in the sequence diagram correspond to UML standards.		
Sequence diagram matches class diagram.		
All the classes in the USE model are represented in the class diagram.		
The multiplicities in the USE model have been depicted in the class diagram.		
All the requirements in the Software Requirements Specification have been covered in the		

Architecture Design Document.		
-------------------------------	--	--

## 5. Formal Technical Inspection Report

Dr.Amtoft,Dr. Andresen,Dr.Neilsen

I have completed the Formal Technical Inspection for Vamsi Mummaneni's Online Book Store project. All the items in the checklist passed the test criteria.

Item to be Inspected	Pass/Fail/Partial	Comments
All the symbols used in the class diagrams are according to the UML standards.	Pass	
All the classes in the class diagram are clear as to what they represent in the architecture design document.	Pass	
The symbols used in the sequence diagram correspond to UML standards.	Pass	
Sequence diagram matches class diagram.	Pass	
All the classes in the USE model are represented in the class diagram.	Pass	
The multiplicities in the USE model have been depicted in the class diagram.	Pass	
All the requirements in the Software Requirements Specification have been covered in the Architecture Design Document.	Pass	

Thanking you,  
Phaninder Surapaneni

Dr.Amtoft,Dr. Andresen,Dr.Neilsen

I have completed the Formal Technical Inspection for Vamsi Mummaneni's Online Book Store project. All the items in the checklist passed the test criteria.

<b>Item to be Inspected</b>	<b>Pass/Fail/Partial</b>	<b>Comments</b>
All the symbols used in the class diagrams are according to the UML standards.	Pass	
All the classes in the class diagram are clear as to what they represent in the architecture design document.	Pass	
The symbols used in the sequence diagram correspond to UML standards.	Pass	
Sequence diagram matches class diagram.	Pass	
All the classes in the USE model are represented in the class diagram.	Pass	
The multiplicities in the USE model have been depicted in the class diagram.	Pass	
All the requirements in the Software Requirements Specification have been covered in the Architecture Design Document.	Pass	

Thanking you,  
Snehal Monterio

## **Section.6 Software Quality Assurance Plan**

### **1. Purpose**

The purpose of Software Quality Assurance Plan is to define all the techniques, procedures, and methodologies that will be used in the project to assure timely delivery of the software that meets specified requirements within project resources. Software Quality Assurance involves reviewing and auditing the software products and activities to verify that they comply with the applicable procedures and standards and providing the software project and other appropriate managers with the results of these reviews and audits.

### **2. Reference Documents**

- Project Plan 1.0
- Vision Document 1.0
- IEEE standard for Software Quality Assurance plans, IEEE STD 730-1998.

### **3. Management**

#### **3.1 Organization and Responsibilities**

The organization consists of supervisory committee, major professor, developer and formal technical inspectors.

#### **Supervisory Committee**

- Dr. Torben Amtoft
- Dr. Daniel Andresen
- Dr. Mitchell Neilsen



The committee will be responsible for attending the presentations and reviews at the end of each phase. After each presentation, the committee will provide feedback and suggestions regarding the project.

### **Major Professor**

- Dr. Torben Amtoft

The major professor will be responsible for supervising and evaluating the work and progress done by the developer on a weekly basis.

### **Developer**

- Vamsi Mummaneni

The developer should submit all the deliverables and complete the project functionalities on time. He should keep updating his weekly progress to the Major Professor.

### **Formal Technical Inspectors**

The formal technical inspectors will be responsible for a technical review of the architecture design artifacts and the formal requirements specifications and will also be required to submit a formal report based on their findings.

#### Formal Technical Inspection Checklist

<b>Item to be Inspected</b>	<b>Pass/Fail/Partial</b>	<b>Comments</b>
All the symbols used in the class diagrams are according to the UML standards.		
All the classes in the class diagram are clear as to what they represent in the architecture design document.		
The symbols used in the sequence diagram correspond to UML standards.		
Sequence diagram matches class diagram.		
All the classes in the USE model are represented in the class diagram.		
The multiplicities in the USE model have been		

depicted in the class diagram.		
All the requirements in the Software Requirements Specification have been covered in the Architecture Design Document.		

### 3. 2 Tasks

All tasks to be performed are mentioned in the Project Plan1.0.After the first phase if any changes are required the Major Professor will discuss with the developer.

### 4. Documentation

The documentation will consist of all the deliverables. They are vision document, project plan, software quality assurance plan, formal requirements specification, architecture design, test plan, formal technical inspection, prototype, user manual, component design, source code, assessment evaluation, project evaluation, references, and formal technical inspection letters. The committee members will review all documentation for final approval.

All documentation will be posted on the developer’s website at <http://people.cis.ksu.edu/~vamsim/mseproject.html>

### 5. Standards, Practices, Conventions, and Metrics

- **Documentation Standards**

The IEEE standards are used as reference for all the documents of Online Book Store project. IEEE Standard for Software Quality Assurance Planning is used for SQA 1.0.

- **Coding Standards**

The project coding standards follow C# .As the project is developed using the C# language.

- **Testing Standards**

The various types of testing performed would be mentioned in the Test Plan document at the end of second phase.

## **6. Reviews and Audits**

All the deliverables produced in each phase of project development are reviewed by the major professor and supervising committee. These are evaluated by the committee, at the end of each phase and provide comments on the software prototype as well as suggestions for any changes or addition to the requirements specification. The two formal technical inspectors will assess the architecture design artifacts and submit a formal report based on their findings.

## **7. Test and Problem Reporting**

All the testing procedures used for the project would be mentioned in the test plan document at the end of second phase. The results would be reviewed and all the unresolved problems will be reported to the committee members.

## **8. Tools, Techniques and Methodologies**

The following tools, techniques and methods would be used for the project for the specified purpose:

- **Coding:** - C#, CSS, HTML, ASP.Net, Microsoft VisualStudio.Net 2005 IDE, Microsoft SQL Server 2005
- **Testing:-**Junit
- **Documentation:-**MS Word, Rational Rose, MS Project, OCL.

## **9. Records collection, Maintenance, and Retention**

Three sets of design documentation would be produced and distributed to the University Library, Major Professor and developer. The source code, documentation and web pages are submitted to the Major Professor in the form of a CD.

## **10. Deliverables**

The deliverables for all the three phases are listed below:

### **Phase I**

- Vision Document 1.0
- Project Plan 1.0

- Software Quality Assurance Plan

## **Phase II**

- Action Items –identified during phase I
- Vision Document 2.0
- Project Plan 2.0
- Formal Requirements Specification
- Architecture Design
- Test Plan
- Formal Technical Inspection
- Executable Architecture Prototype

## **Phase III**

- Action Items - identified during phase I
- User Manual
- Component Design
- Source Code
- Assessment Evaluation
- Project Evaluation
- References
- Formal Technical Inspection

## **Section.7 Test Plan**

### **1. Test plan identifier**

CIS 895-MSE Project Test plan Online Book Store V1.0

### **2. Introduction**

The goal of this document is to develop a test plan for the Online Book Store design system. This document defines all the procedures and activities required to prepare for testing of the functionalities of the system which are specified in Vision document. The objectives of the test plan are to define the activities to perform testing, define the test deliverables documents and to identify the various risks and contingencies involved in testing.

### **3. Features to be tested**

The following list describes the features to be tested:

#### **USER:**

- Registration
- Login
- Add To Cart
- Edit Cart

#### **ADMIN:**

- Create and Delete book from Category
- Create and Delete a Category
- Manage Orders
- Manage Members

## 4. Test Cases

### 4.1 USER

#### Registration

ID	TEST CASE	USER INPUT	PASS CRITERIA
U_REG_1	User Registration	User selects already existing user name	Display message to choose different user name
U_REG_2	User Registration	User enters different password in password confirm field	Display message that Password and Confirm Password fields don't match
U_REG_3	User Registration	User forgets to enter a particular required fields	Display message The value in field is required
U_REG_4	User Registration	User enters all the details successfully	User account created

#### Login

ID	TEST CASE	USER INPUT	PASS CRITERIA
U_LOG_1	User Login	User enters a wrong username	Display message Login or Password is incorrect.
U_LOG_2	User Login	User enters a wrong password	Display message Login or Password is incorrect.
U_LOG_3	User Login	User enters correct username and password	User logs in successfully

#### Add to Cart

ID	TEST CASE	USER INPUT	PASS CRITERIA
U_AC_1	Add to Cart	User selects a book and clicks add to cart button	Book is added to the shopping cart
U_AC_2	Add to Cart	Guest selects a book and clicks add to cart button	User should create an account.

## Edit Cart

ID	TEST CASE	USER INPUT	PASS CRITERIA
U_EC_1	Edit Cart	User changes the Quantity	Quantity and total cost of Cart should be updated
U_EC_2	Edit Cart	User deletes a book from shopping Cart	Books and total cost of Cart should be updated
U_EC_3	Edit Cart	User selects a new book to shopping Cart	Books and total cost of Cart should be updated

## 4.2 ADMIN

### Create and Delete a Book from Category

ID	TEST CASE	ADMIN INPUT	PASS CRITERIA
AD_CDB_1	Create and Delete a Book from Category	Admin adds a new book to category	Book should be updated in Categories list
AD_CDB_2	Create and Delete a Book from Category	Admin deletes a book from category	Book should be deleted in Categories list

### Create and Delete a Category

ID	TEST CASE	ADMIN INPUT	PASS CRITERIA
AD_CDC_1	Create and Delete a Category	Admin adds a new category	Category should be updated to system
AD_CDC_1	Create and Delete a Category	Admin deletes a category	Category should be deleted from system

## Manage Orders

ID	TEST CASE	ADMIN INPUT	PASS CRITERIA
AD_MO_1	Manage Orders	Admin accepts an order	Order is processed
AD_MO_2	Manage Orders	Admin deletes an order	Order is not processed

## Manage Members

ID	TEST CASE	ADMIN INPUT	PASS CRITERIA
AD_MM_1	Manage Members	Admin accepts Members	Member is accepted
AD_MM_2	Manage Members	Admin deletes Members	Member is not accepted

## 5. Approach

This section describes the overall approach of the testing which ensures that each feature and the combination of the features are adequately tested. The major tasks that are used are

### 5.1 Unit testing

Unit testing is a method of testing that verifies the individual units of source code are working properly. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. The NUnit a testing tool for C# will be used for unit testing.



## **5.2 Load testing**

Load testing is the process of creating demand on a system or device and measuring its response. It generally refers to the practice of modeling the expected usage of a software program by simulating multiple users accessing the program concurrently. As such, this testing is most relevant for multi-user systems; often one built using a client/server model, such as web servers

## **5.3 System Testing**

Once the entire system has been built then it has to be tested against the Software Requirement Specification and System Specification to check if it delivers the features required. System testing can involve a number of specialist types of test to see if all the functional and non-functional requirements have been met.

## **5.4 Performance Testing**

The system should meet the performance requirements as mentioned in the Vision document. The performance will be evaluated based on the response time of the GUI and the database commands. Using JMETER tool performance testing will be done.

## **5.5 Manual Testing**

Manual Testing will be done to ensure the correctness of various parts of the code using test cases generated by the tester.

## **6. Pass/fail criteria**

The system should satisfy all the functional requirements, in the Vision document. Each feature to be tested will be evaluated against its requirement as stated in the Vision Document. The pass or fail of a test depends on whether the system meets with all the particular post conditions.

Test cases executed on the Online Book Store will pass if they meet the specific requirements as mentioned in the Vision Document.

## **7. Suspension criteria and resumption requirements**

### **7.1 Suspension criteria**

If the system contains one or more critical defects like the defects in the GUI editor which provides the editing features for one line diagrams and database locking, unlocking and sharing features which provides the environment for multiple users to work in parallel, the entire system should be suspended.

The testing may also be suspended if the hardware and software components required are not available on time.

The failed test cases should be recorded along with the description for failure.

## **7.2 Resumption requirements**

When a new version of the system is transmitted to the test group after a suspension of testing has occurred, all previous tests will be rerun to ensure program changes have not inadvertently affected other portions of the program.

## **8. Test deliverables**

The following documents are the available test deliverables:-

- Test plan
- Test case specifications
- Test input and output data
- Test procedure specifications
- Test logs

## Section.8 Assessment Evaluation

### 1. Introduction

The goal of this document is to present all the test results using the test cases defined in the Test Plan Document. I have performed Manual and Performance testing for my Online Book store project.

### 2. Manual Testing

Manual testing is done to test the correctness of all the functionalities by manually entering the data.

Test case functionalities for manual testing include:

#### USER:

- Registration
- Login
- Add To Cart
- Edit Cart

#### ADMIN:

- Create and Delete book from Category
- Create and Delete a Category
- Manage Orders
- Manage Members

### 3. Test Cases

#### 3.1 USER

##### Registration

ID	TEST CASE	USER INPUT	RESULT	PASS CRITERIA
U_REG_1	User Registration	User selects already existing user name	PASSED	Display message to choose different user name
U_REG_2	User Registration	User enters different password in password confirm field	PASSED	Display message that Password and

				Confirm Password fields don't match
U_REG_3	User Registration	User forgets to enter a particular required fields	PASSED	Display message The value in field is required
U_REG_4	User Registration	User enters all the details successfully	PASSED	User account created

### Login

ID	TEST CASE	USER INPUT	RESULT	PASS CRITERIA
U_LOG_1	User Login	User enters a wrong username	PASSED	Display message Login or Password is incorrect.
U_LOG_2	User Login	User enters a wrong password	PASSED	Display message Login or Password is incorrect.
U_LOG_3	User Login	User enters correct username and password	PASSED	User logs in successfully

### Add to Cart

ID	TEST CASE	USER INPUT	RESULT	PASS CRITERIA
U_AC_1	Add to Cart	User selects a book and clicks add to cart button	PASSED	Book is added to the shopping cart
U_AC_2	Add to Cart	Guest selects a book and clicks add to cart button	PASSED	User should create an account.

### Edit Cart

ID	TEST CASE	USER INPUT	RESULT	PASS CRITERIA
U_EC_1	Edit Cart	User changes the Quantity	PASSED	Quantity and total cost of Cart should be updated
U_EC_2	Edit Cart	User deletes a book from shopping Cart	PASSED	Books and total cost of Cart should be updated
U_EC_3	Edit Cart	User selects a new book to shopping Cart	PASSED	Books and total cost of Cart should be updated

## 3.2 ADMIN

### Create and Delete a Book from Category

ID	TEST CASE	ADMIN INPUT	RESULT	PASS CRITERIA
AD_CDB_1	Create and Delete a Book from Category	Admin adds a new book to category	PASSED	Book should be updated in Categories list
AD_CDB_2	Create and Delete a Book from Category	Admin deletes a book from category	PASSED	Book should be deleted in Categories list

### Create and Delete a Category

ID	TEST CASE	ADMIN INPUT	RESULTS	PASS CRITERIA
AD_CDC_1	Create and Delete a Category	Admin adds a new category	PASSED	Category should be updated to system
AD_CDC_1	Create and Delete a Category	Admin deletes a category	PASSED	Category should be deleted from system

### Manage Orders

ID	TEST CASE	ADMIN INPUT	RESULTS	PASS CRITERIA
AD_MO_1	Manage Orders	Admin accepts an order	PASSED	Order is processed
AD_MO_2	Manage Orders	Admin deletes an order	PASSED	Order is not processed

## Manage Members

ID	TEST CASE	ADMIN INPUT	RESULTS	PASS CRITERIA
AD_MM_1	Manage Members	Admin accepts Members	PASSED	Member is accepted
AD_MM_2	Manage Members	Admin deletes Members	PASSED	Member is not accepted

## 4. Performance Testing

The system should meet the performance requirements as mentioned in the Vision document. The performance will be evaluated based on the response time of the GUI and the database commands. Using JMETER tool performance testing will be done.

Apache JMETER is a 100% pure Java desktop application designed to load test functional behavior and measure performance. Originally designed to test web applications, it is now a general tool for testing client/server applications like database servers, FTP servers and others. It lets you test your applications under different work loads, and graphically represent the application's behavior.

I have tested the performance of three different pages of my Online Book Store Website as mentioned in the Test Plan document during Phase-2.

### Pages Tested

- Home Page
- Search Page
- Shopping Cart Page

The Performance testing has been done using a sample of 8 threads which are generated with a loop of 100.

## Home Page

<http://localhost:3062/vamsi-bookstore/Default.aspx>

The graph below in Fig.1 shows the test results of Homepage of Online Book Store Website.

## Performance Graph



Fig.1

## Search Page

<http://localhost:3062/vamsi-bookstore/Books.aspx?>

The graph results below in Fig.2 shows the test results of Search Page of Online Book Store Website.

### Performance Graph



Fig.2



## Shopping Cart Page

<http://localhost:3062/vamsi-bookstore/ShoppingCart.aspx?>

The graph results below in Fig.3 shows the test results of Shopping Cart Page of Online Book Store Website.

### Performance Graph

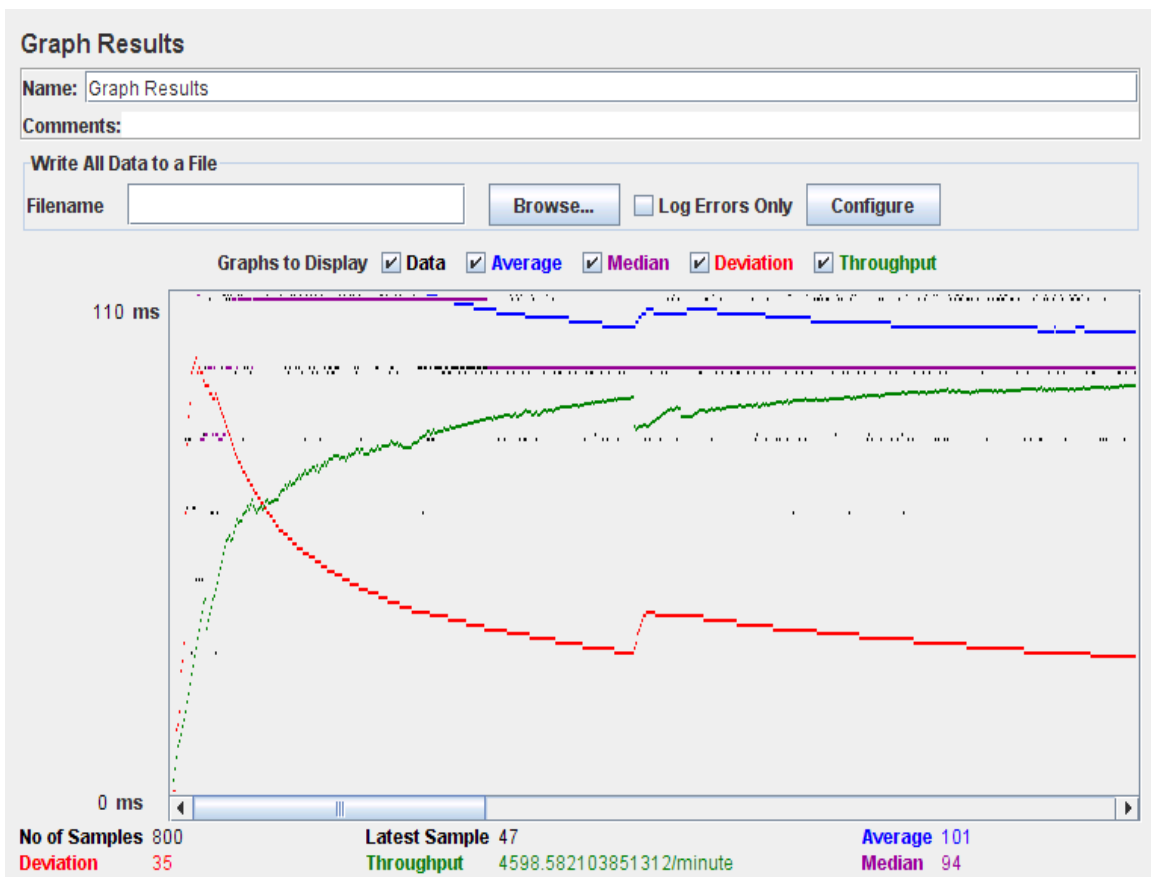


Fig.3

## 5. Observations

- Average response time is more for search page when compared to the Home page.
- Average response time is low for home page because it doesn't many database interactions.
- Average response time for Search page is more since it should wait for the results from the database.

## Section.9 User Manual

### User Manual

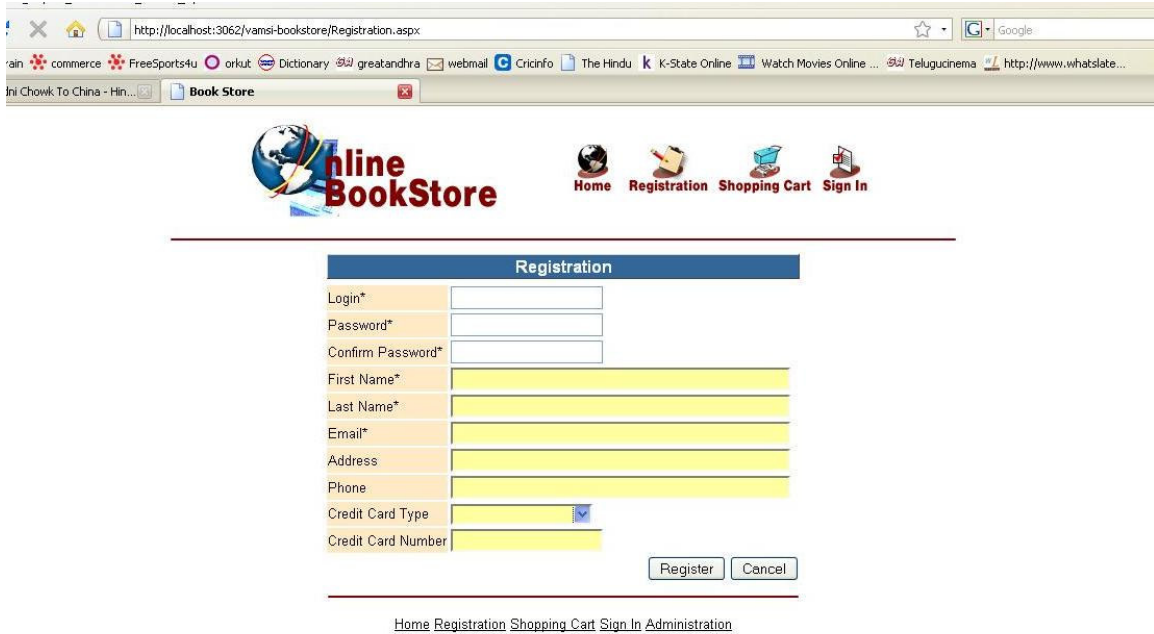
When we run the Online Book Store Website first home page is displayed. The home page will appear as below. The user and Admin will have different rights.



If you are a new user you can register using the register link or if you are already a user you can login to purchase book and pay using the checkout.

Any common user can use the search option in the home page to search for a book of his choice by entering a Title of the book. If he wants to search more precisely he can use the advanced search option on the home page.

The register page will appear as below



A user should enter all the required fields information. If he didn't fill all the fields he cannot create an account. After entering all the required data click the submit button to register. Now you can login to the Website. Login page is used to login to the website. Login page will appear as below.



After logging in to the website you will enter into a userinfo page which will show the account and shopping cart details.

The screenshot shows a web browser window with the URL `http://localhost:3062/vamsi-bookstore/ShoppingCart.aspx?`. The page features the 'Online BookStore' logo and navigation links: Home, Registration, Shopping Cart, and Sign In. Below the navigation is a 'User Information' section with the following details:

Login	vamsim
First Name	vamsi
Last Name	mummaneni
Address	1214 vattier st apt#4
Email	vamsim@ksu.edu
Phone	972-313-5729

Below the user information is an 'Items' table representing the shopping cart:

	Order #	Item	Price	Quantity	Total
<a href="#">Details</a>	19	Beginning ASP Databases	39.99	1	39.99
<b>Total</b>					39.99

At the bottom of the cart section are links for [Continue Shopping](#) and [Checkout](#). A footer navigation bar includes [Home](#), [Registration](#), [Shopping Cart](#), [Sign In](#), and [Administration](#).

If you want to update your account details click the login hyperlink and you will be directed to the update page.

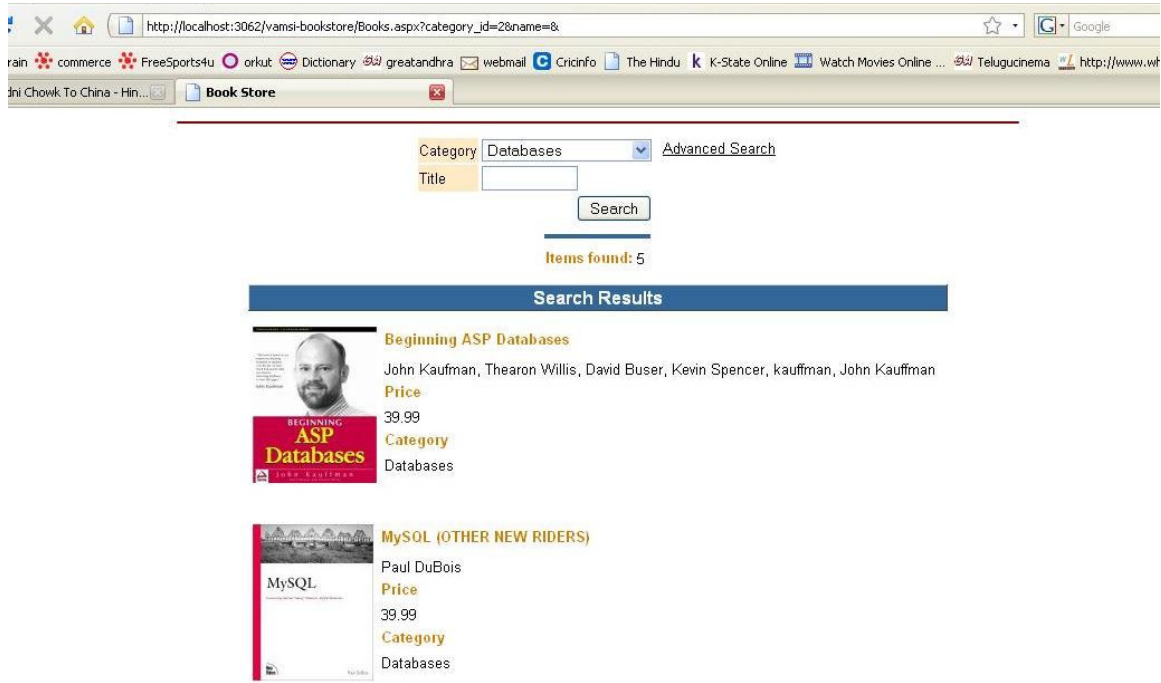
The screenshot shows a web browser window with the URL `http://localhost:3062/vamsi-bookstore/MyInfo.aspx?`. The page features the 'Online BookStore' logo and navigation links: Home, Registration, Shopping Cart, and Sign In. Below the navigation is a 'MyInfo' section with a form for updating user details:

Login	vamsim
Password*	••••••
First Name*	vamsi
Last Name*	mummaneni
Email*	vamsim@ksu.edu
Address	1214 vattier st apt#4
Phone	972-313-5729
Notes	<input type="text"/>
Credit Card Type	Mester
Credit Card Number	4412567845678

At the bottom of the form are [Update](#) and [Cancel](#) buttons. A footer navigation bar includes [Home](#), [Registration](#), [Shopping Cart](#), [Sign In](#), and [Administration](#).

In the update page user can change his account information and click update. The updated information will be saved in the database.

To search for a book the user can use the search and Advanced search options.



The user can search for a book based on price range, book title, author and Category. This option is provided in the advanced search page.

If the user is interested in any particular book he can click the book image to get the book details.

The book details page will appear as below.

History Bookmarks Tools Help

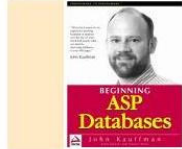
http://localhost:3062/vamsi-bookstore/BookDetail.aspx?item\_id=22&author=&category\_id=2&name=&pricemax=&pricemin=&

commerce FreeSports4u orkut Dictionary greatandhra webmail Cricinfo The Hindu K-State Online Watch Movies Online ... Teluginema http://www.whatslate...

Book Store

**Book Detail**

Title Beginning ASP Databases  
 Author John Kaufman, Thearon Willis, David Buser, Kevin Spencer, kauffman, John Kauffman  
 Category Databases  
 Price 39.99



Notes  
 As a tutorial, Beginning ASP Databases offers an entry point to one of the most crucial aspects of Microsoft-oriented Web development--database integration with Active Server Pages. In Beginning ASP Databases, a trio of authors covers the basics of working with databases from ASP--especially using ActiveX Database Objects (ADO). The book is quite substantive in content but is written in a somewhat light-hearted style that makes readers new to the technology comfortable. The authors begin with a discussion of the benefits of ASP and ADO, and then explain how to configure the technology on the various flavors of Windows. From there, they show how to access databases using ADO, demystify how it all works, and focus on several key areas, including debugging, SQL, and stored procedures.

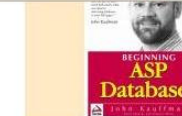
[Review this book on Amazon.com](#)

Quantity

http://localhost:3062/vamsi-bookstore/BookDetail.aspx?item\_id=22&author=&category\_id=2&name=&pricemax=&pricemin=&

commerce FreeSports4u orkut Dictionary greatandhra webmail Cricinfo The Hindu K-State Online Watch Movies Online ... Teluginema http://www.whatslate...

Book Store



Notes  
 As a tutorial, Beginning ASP Databases offers an entry point to one of the most crucial aspects of Microsoft-oriented Web development--database integration with Active Server Pages. In Beginning ASP Databases, a trio of authors covers the basics of working with databases from ASP--especially using ActiveX Database Objects (ADO). The book is quite substantive in content but is written in a somewhat light-hearted style that makes readers new to the technology comfortable. The authors begin with a discussion of the benefits of ASP and ADO, and then explain how to configure the technology on the various flavors of Windows. From there, they show how to access databases using ADO, demystify how it all works, and focus on several key areas, including debugging, SQL, and stored procedures.

[Review this book on Amazon.com](#)

Quantity

Add to Shopping Cart

**Rating**

Current Rating ★★☆☆☆

Total Votes 1

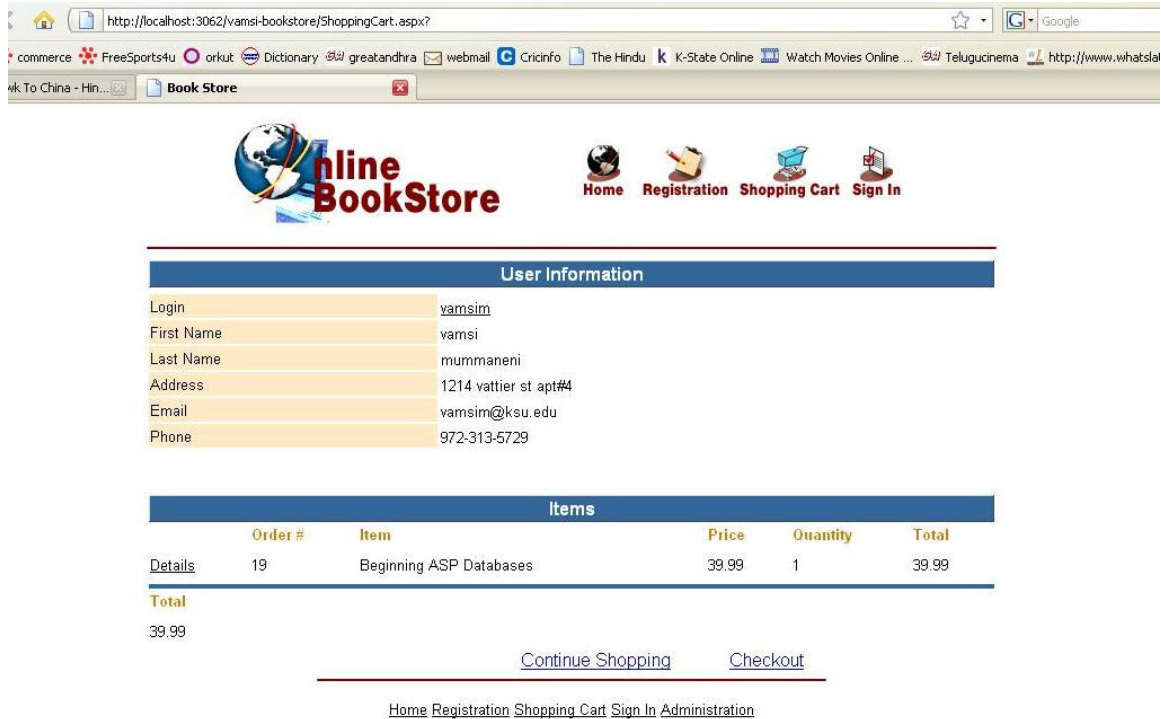
Your Rating

Vote

The user can add the book to a shopping cart by entering the quantity and clicking the add to shopping cart button. The quantity value should always be a positive value.

The user can also provide feedback to a book in the form of user rating by selecting the Your Rating dropdown list.

After adding to the shopping cart if the user wants to continue shopping he should click the continue shopping link else he can checkout using the checkout link.



The screenshot shows a web browser window with the URL <http://localhost:3062/vamsi-bookstore/ShoppingCart.aspx?>. The page features the "Online BookStore" logo and navigation links for Home, Registration, Shopping Cart, and Sign In. Below the navigation is a "User Information" section with the following details:

User Information	
Login	vamsim
First Name	vamsi
Last Name	mummaneni
Address	1214 vattier st apt#4
Email	vamsim@ksu.edu
Phone	972-313-5729

Below the user information is an "Items" table showing the contents of the shopping cart:

Items					
	Order #	Item	Price	Quantity	Total
<a href="#">Details</a>	19	Begining ASP Databases	39.99	1	39.99
<b>Total</b>					39.99

At the bottom of the cart page, there are two links: [Continue Shopping](#) and [Checkout](#). A footer navigation bar includes [Home](#), [Registration](#), [Shopping Cart](#), [Sign In](#), and [Administration](#).

The user can also edit his shopping cart details. If he wants to change the quantity or cancel his order he can use this page. When the user clicks the details link in the shopping cart page he will be redirected to the edit shopping cart page.





The above page shows the edit shopping cart page.

After proceeding to checkout the user should enter the billing details to process his order request.

For this he should enter his Credit Card details correctly.

The below page is the checkout page.

History Bookmarks Tools Help

http://localhost:3062/vamsi-bookstore/checkout.aspx

commerce FreeSports4u orkut Dictionary greatandhra webmail Cricinfo The Hindu K-State Online Watch Movies Online ... Telugucinema http://www.whatsla

on BollywoodHungam... Book Store

fox: to remember this password? Remember Never for This Site



CreditCard Type :

CreditCard Number :

CVV Code :

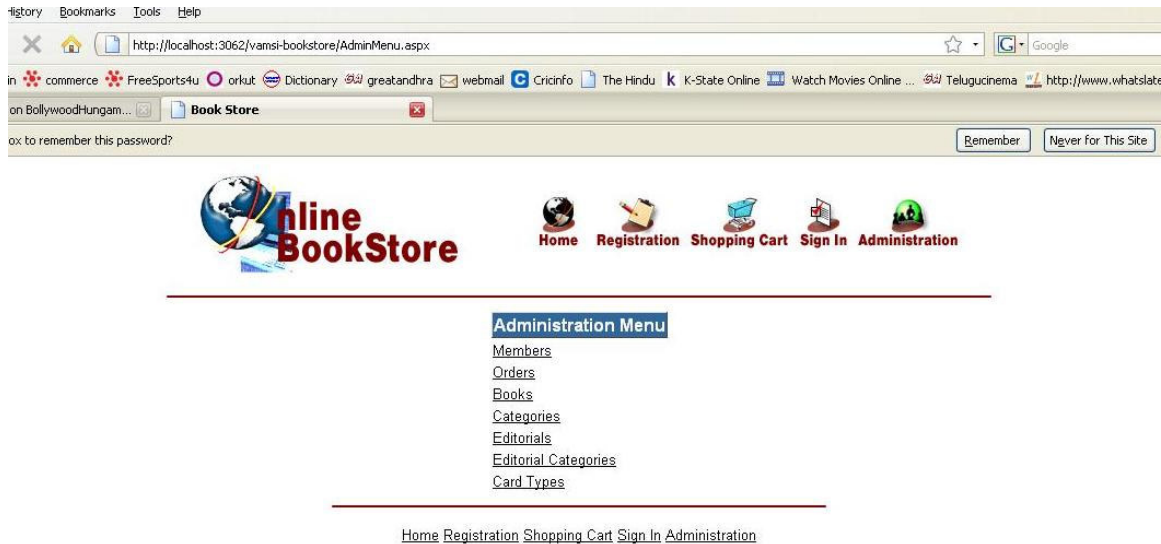
Expiry Date :

[Home](#) [Registration](#) [Shopping Cart](#) [Sign In](#) [Administration](#)

After entering the card details click the submit button to process the order.

# Administration Manual

The Admin should login to the Website to use his administrative options. After logging in he will be redirected to the admin menu page.



The Administrative options will be displayed in the menu. He can change the details of the members, books, orders, Categories. And also can process the order and add new books and categories to the books list.

When the Admin clicks the Members page he will be navigated to the page below which will show all the member details. The Admin can search for a member using the Search box option.

The screenshot shows a web browser window with the URL `http://localhost:3062/vamsi-bookstore/MembersGrid.aspx`. The page features a navigation menu with icons for Home, Registration, Shopping Cart, Sign In, and Administration. Below the menu is a search bar with the text "Enter full or partial login, first or last name" and a "Search" button. The main content area displays a table of members:

Members			
Login	First Name	Last Name	Level
<a href="#">admin</a>	Administrator	Account	Administrator
<a href="#">bhanu</a>	praveen	muwa	Member
<a href="#">guest</a>	Guest	Account	Member
<a href="#">vamsi</a>	vamsi	mummaneni	Member
<a href="#">vamsim</a>	vamsi	mummaneni	Member

At the bottom of the page, there is a navigation bar with links for Home, Registration, Shopping Cart, Sign In, and Administration.

The Admin will be directed to the books list page where he can update, delete and add book details. He can also search for a particular book using this page.

Browser: http://localhost:3062/vamsi-bookstore/AdminBooks.aspx

Category: All Recommended: All Search

Books				
Edit Title	Author	Price	Category	Recommended
<a href="#">Edit</a> Web Database Development : Step by Step	Jim Buyens	39.99	Databases	Yes
<a href="#">Edit</a> Programming Perl (3rd Edition)	Larry Wall, Tom Christiansen, Jon Orwant	39.96	Programming	No
<a href="#">Edit</a> Perl and CGI for the World Wide Web: Visual QuickStart Guide	Elizabeth Castro	15.19	Programming	No
<a href="#">Edit</a> MySQL (OTHER NEW RIDERS)	Paul DuBois	39.99	Databases	No
<a href="#">Edit</a> PHP and MySQL Web Development	Luke Welling, Laura Thomson	39.99	Programming	No
<a href="#">Edit</a> MySQL & PHP From Scratch	Wade Maxfield	23.99	Programming	Yes
<a href="#">Edit</a> MySQL and mSQL	Randy Jay Yarger, George Reese, Tim King	27.96	Databases	Yes
<a href="#">Edit</a> Flash 4 Magic (WITH CD-ROM)	David J. Emberton, J. Scott Hamlin, David Emberton	36	HTML & Web design	No
<a href="#">Edit</a> Web Application Development with PHP 4.0 (with CD-ROM)	Tobias Ratschiller, Till Gerken	36	Programming	No
<a href="#">Edit</a> Beginning ASP Databases	John Kaufman, Thearon Willis, David Buser, Kevin Spencer, kauffman, John Kauffman	39.99	Databases	Yes
<a href="#">Edit</a> Oracle8i Web Development	Bradley D. Brown, Brad Brown	41.99	Databases	No
<a href="#">Edit</a> Black Belt Web Programming Methods; Servers, Security, Databases and Sites		27.96	Programming	Yes
<a href="#">Edit</a> Web Development with Java Server	Duane K. Fields, Mark A. Kelly	35.96	Programming	Yes

The orders link will navigate the Admin to the orders page which will display all the orders placed by the user.

Browser: http://localhost:3062/vamsi-bookstore/OrdersGrid.aspx

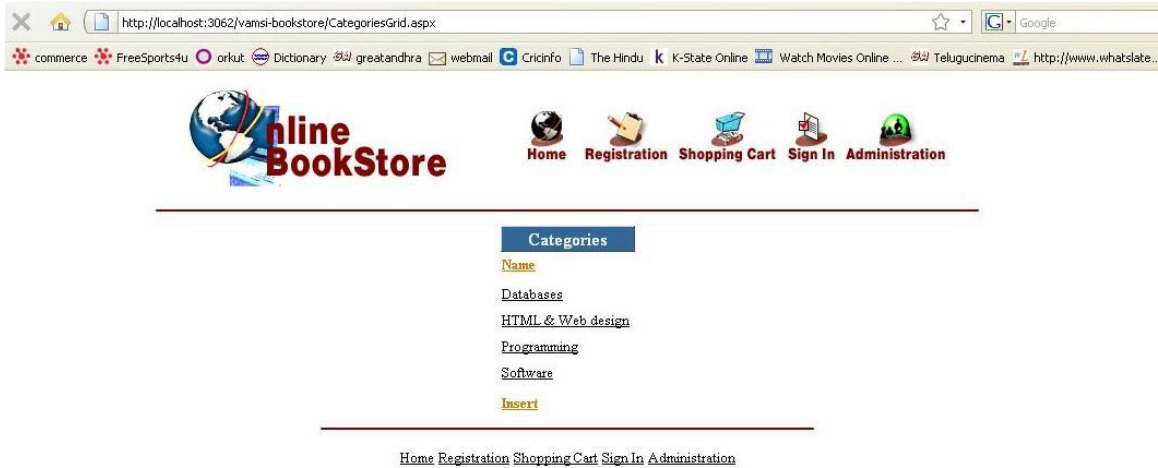
Item: All Member: All Search

Orders	
Edit Item	Member Quantity
<a href="#">Edit</a> Web Database Development : Step by Step	vamsi 2
<a href="#">Edit</a> Flash 4 Magic (WITH CD-ROM)	vamsi 1
<a href="#">Edit</a> ColdFusion Fast & Easy Web Development	bhanu 2
<a href="#">Edit</a> Beginning ASP Databases	bhanu 1
<a href="#">Edit</a> Black Belt Web Programming Methods; Servers, Security, Databases and Sites	bhanu 1
<a href="#">Edit</a> MySQL and mSQL	bhanu 1
<a href="#">Edit</a> Web Database Development : Step by Step	bhanu 2
<a href="#">Edit</a> C# - Programming with the Public Beta	guest 1
<a href="#">Edit</a> Beginning ASP Databases	vamsim 1

[Insert](#)

Home Registration Shopping Cart Sign In Administration

The Categories page will display the entire book Categories. The user can update, delete and add a category entry to the original list.



The Admin can logout at any moment by clicking the logout button.



## **Section.10 Project Evaluation**

### **1. Introduction**

This document evaluates the experience of the development of the Online Book Store project. A brief description of the tools, process, techniques employed as well as the mistakes made is presented so that lessons are documented and learned.

### **2. Problems faced**

The following are the problems faced during the design of Online Boos Store Website.

#### **2.1 C# language**

The main problem encountered during the process was my inexperience in the ASP.NET and C# Language and environment.

So I used the videos in asp.net website to get basic experience in C#.NET.I tried to solve some sample applications which helped me to design the website easily. It was difficult for me to format the home page of my website using the C#.NET code. So finally I choose to use the html div tags which made my work easier. I also didn't have good experience of using the J-Meter testing tool for performance testing. So I learnt about it using some tutorials which I found using google.com.

#### **2.2 Security**

I also had some security issues to be resolved.

- Only the user who will login can order for a book and checkout.
- And also the user should not be able to access the Administrative options.

It was easy for me to create the user security options for checkout but I faced problems with protecting Admintrative options from the user. But later I was able to figure it out using Google search engine and some videos about C# security.

### **3. Metrics**

#### **3.1 Lines of Code**

My initial estimate was 3000 LOC during Phase-1.This was calculated in the COCOMO model in Project Plan 1.0 document.

But now I found the total to be 3203 LOC after the coding part is completed. I used the LOC metrics tool to count the total number of Lines of Code.

The reason for increase in the size of the project may be due to my inexperience in programming in C# and using extra functions for functionality. And it was difficult for me to estimate the exact number of lines of code during the initial phase of my Project.

### 3.2 Time

The project took 27120 minutes, which is 452 hours, at 22 hours a week that is 19 weeks or 4.8 months. The initial estimate for my project using the COCOMO model analysis has come to 4.96 staff months.

Table below shows the time spent for each task at each phase. Design is the most important task that was performed at last two phases .Coding was performed in last two phases but mainly in the implementation phase.

<b>Phases</b>	<b>Start Date</b>	<b>End Date</b>
Inception	July 15 <sup>th</sup> 2008	Aug 18 <sup>th</sup> 2008
Implementation	Aug 20 <sup>th</sup> 2008	Nov 20 <sup>th</sup> 2008
Elaboration	Nov 21 <sup>st</sup> 2008	Dec 11 <sup>th</sup> 2008

Table.1 Project Phase Schedule

The Table below shows the break down of time spent in each phase for Research, Design, Coding, Testing and Documentation.

	<b>Inception(Hours)</b>	<b>Implementation(Hours)</b>	<b>Elaboration(Hours)</b>	<b>Total</b>
<b>Research</b>	20	25	5	50
<b>Design</b>	17	20	5	42
<b>Coding</b>	0	210	37	247
<b>Testing</b>	0	15	30	45
<b>Documentation</b>	20	28	20	68
<b>Total</b>	57	298	97	452

Table.2 Phase Breakdown



The Pie chart in Fig.1 shows the hour break down for each phase.

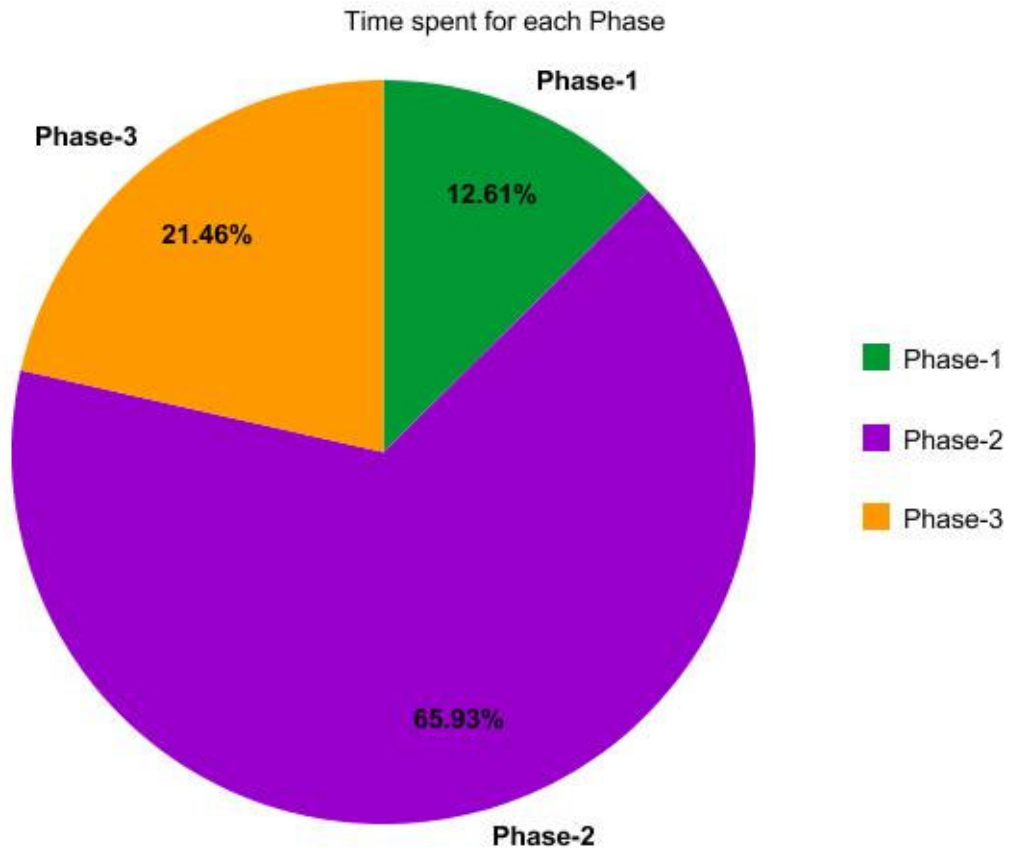


Fig.1 Project Phase Schedule

The pie chart in Fig.2 shows the Time allocated for various tasks during the Project Implementation.

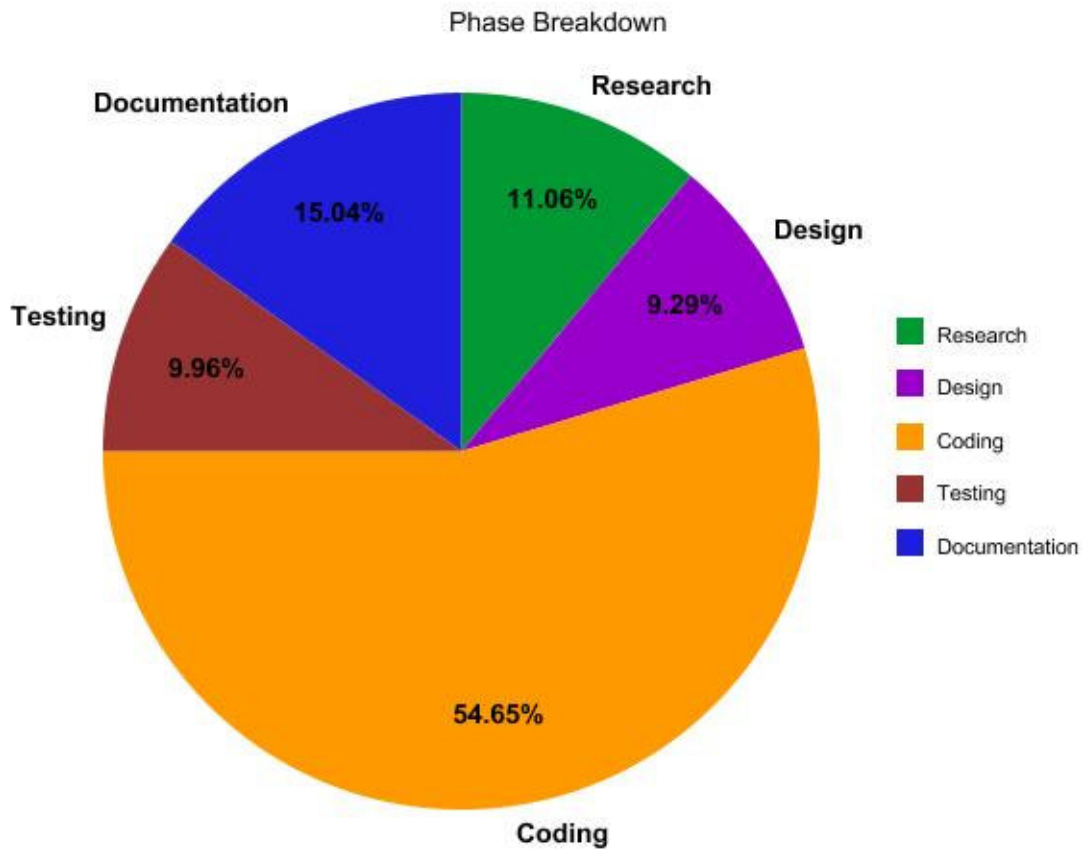


Fig.2 Task breakdown

#### 4. Lessons Learned

##### Programming

The Online Book Store Project helped me to improve my confidence level in C# Programming. Though I have made many mistakes during the initial phase I have learnt how to use user controls, master pages, data grid, data set and other data base functionalities.

##### Time Management

Since MSE Project is done as an individual I have learnt how to manage time during the Software Life Cycle Process. I have also learned how to face tense situations and meet the deadlines. This would add as a good experience for me for my future job prospective.

## **UML and Software Lifecycle**

As software student though I have good knowledge in UML and Software LIFE cycle I never had any good practical experience regarding them. Through this project I have learnt how to develop a project following the various stages in Software Life Cycle.

## **Documentation**

I always had a feeling that I am not good at documentation .But through this project and suggestions from my committee members I believe that I have improved my Documentation skills.

## Section 11. References

- IEEE Recommended Practice for Software Design Descriptions IEEE Std 1016-1998
- IEEE Recommended Practice for Software Requirements Specifications - IEEE Std 830-1998
- IEEE Standard for Software Test Documentation IEEE Std 829-1998
- IEEE Guide for Software Quality Assurance Planning - IEEE Std 730.1-1995
- <http://www.asp.net/learn/data-access/tutorial-16-vb.aspx>
- [www.gliffy.com](http://www.gliffy.com)
- [7http://msdn.microsoft.com/en-us/library/system.security.permissions.securitypermissionattribute\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/system.security.permissions.securitypermissionattribute(VS.71).aspx)
- <http://www.locmetrics.com/>
- <http://nces.ed.gov/nceskids/createAGraph/>
- [http://en.wikipedia.org/wiki/Load\\_testing](http://en.wikipedia.org/wiki/Load_testing)
- [http://en.wikipedia.org/wiki/Unit\\_test](http://en.wikipedia.org/wiki/Unit_test)
- <http://www.viveo-ooobject.com/savoirfaire/ecmfinance/concepts/usine/3tiers.php>
- <file:///C:/Documents%20and%20Settings/Owner/Desktop/jakarta-jmeter-2.3RC3/docs/usermanual/build-adv-web-test-plan.html>
- [http://en.wikipedia.org/wiki/Manual\\_testing](http://en.wikipedia.org/wiki/Manual_testing)