Project Name ***ONLINE BOOK STORE***

THE DESIGN AND IMPLEMENTATION OF AN E-COMMERCE SITE FOR ONLINE BOOK SALES

Project Report submitted to Mewar University, Chittorgarh
towards the partial fulfillment of the degree of
Master of Technology
in
Computer Science & Engineering

By
"Sanjay agal"

Faculty of Engineering & Technology
Mewar University Chittorgarh (Raj.)

Project Name *ONLINE BOOK STORE*

THE DESIGN AND IMPLEMENTATION OF AN E-COMMERCE SITE FOR ONLINE BOOK SALES

Project Report submitted to Mewar University, Chittorgarh
towards the partial fulfillment of the degree of
Master of Technology
in
Computer Science & Engineering

Submitted to

Mr B.L.Pal
Mtech Coordinator ,
Mewar University

Submitted by

Er Sanjay Agal
Roll No. MU09MCSB014

Guided By:
Mr. G.Balakrishna
Asst Professor Mewar University

Faculty of Engineering & Technology
Mewar University Chittorgarh (Raj.)

Faculty of Engineering & Technology
Mewar University Chittorgarh (Raj.)

STATEMENT OF ORIGINALITY

I hereby declare that the work which is being presented in the project entitled "**ONLINE BOOK STORE "** in partial fulfillment of the requirements for the award of the degree of Masters of Technology submitted in the Department of Computer Science & Engineering, Mewar University Chittorgarh (Raj.) is an authentic record of my own work carried out during the period from April, 2011 to September, 2011 under the supervision and guidance of Mr. G.Balakrishna", Asst Prof Mewar University .

The matter embodied in this project work has not been submitted for the award of any other degree.

Name of Guide
Mr. G.Balakrishna
Asst Prof Mewar University

Name of Student
Er Sanjay Agal

Date:

Place: Gangrar Chittor

Faculty of Engineering & Technology
Mewar University Chittorgarh (Raj.)

CERTIFICATE

It is certified that the Minor Project report entitled "**ONLINE BOOK STORE**" is the result of work carried out during the period of April, 2011 to September 2011, by"Er sanjay agal", Department of Computer Science & Engineering, Mewar University, Chittorgarh (Raj), Under my supervision. His work neither in part nor in whole has been submitted for any other degree.

Internal Examiner **External Examiner**

HOD,
Department of Computer Science & Engineering

Date:
Place: ***Gangrar Chittor***

Abstract

The business-to-consumer aspect of electronic commerce (e-commerce) is the most visible business use of the World Wide Web. The primary goal of an e-commerce site is to sell goods and services online.

This project deals with developing an e-commerce website for Online Book Sale. It provides the user with a catalog of different books available for purchase in the store. In order to facilitate online purchase a shopping cart is provided to the user. The system is implemented using a 3-tier approach, with a backend database, a middle tier of Sun J2EE 1.4 application server and JSP, and a web browser as the front end client.

In order to develop an e-commerce website, a number of Technologies must be studied and understood. These include multi-tiered architecture, server and client side scripting techniques, implementation technologies such as JSP, programming language (such as JAVA, JavaScript, and HTML), relational databases (such as MySQL, Access).

This is a project with the objective to develop a basic website where a consumer is provided with a shopping cart application and also to know about the technologies used to develop such an application.

This document will discuss each of the underlying technologies to create and implement an e-book store.

Acknowledgments

In completing this minor project I have been fortunate to have help, support and encouragement from many people. I would like to acknowledge them for their cooperation.

First, I would like to thank Er B L Pal, my project advisor, for guiding me through each and every step of the process with knowledge and support. Thank you for your advice, guidance and assistance.

I would also like to thank Dr K K Chhabra, Director Pacific College of Engineering, Udaipur who showed immense patience and understanding throughout the project and provided suggestions.

Finally, I would like to dedicate this project to my wife Dr Pooja Devija Agal, my baby Aarahan agal and my parents, for their love, encouragement and help throughout the project.

TABLE OF CONTENTS

1. INTRODUCTION ……1

2. LITERATURE REVIEW ……...1

3. IMPLEMENTATION TECHNOLOGIES ……3

3.1. INTRODUCTION TO JSP (J2EE 1.4) ……....3

3.2 JSP ARCHITECTURE ……..4

3.2.1. ACCESS MODEL ……6

3.3. MYSQL DATABASE ……...8

4. WEB PAGE PROGRAMMING OPTIONS ……10

5. WEB BASED APPLICATION DEVELOPMENT ……11

6. PROJECT DESIGN ……...13

6.1 DATA MODEL ……13

3.1.1 Database Design ……19

6.2. PROCESS MODEL ……19

6.2.1. Functional Decomposition Diagram ……21

6.2.2 Data Flow Diagram (DFD) ……21

6.3 USER INTERFACE DESIGN ……31

7. LIMITATIONS AND FUTURE DEVELOPMENT ……..44

8. CONCLUSION ……...45

9. BIBLIOGRAPHY ……...46

List of Figures

Figure 3.2.1	Servlet Container	……..5
Figure 3.2.2	“jsp servlet”	……..6
Figure 3.2.1.1	“access model”	……6
Figure 3.2.1.2	Mvc Design Pattern	……...7
Figure 4.1	Web Page Programming Option	……10
Figure6.1.1	Entity relationship diagram	……...13
Figure 6.2.1.1	Functional Decomposition Program	……20
Figure 6.2.2.1	A Context Level Diagram	……...21
Figure 6.2.2.2	A First Level Diagram	……22
Figure 6.2.2.3	A Second Level Diagram	……23
Figure 6.2.2.4	A Second Level Diagram	……24
Figure 6.2.2.5	Use Case Diagram	……...25
Figure 6.2.2.6	Customer - Browse Context DFD	……...26
Figure 6.2.2.7	Customer - Browse Detailed DFD	……..26
Figure 6.2.2.8	Customer – Shopping Cart Context DFD	……26
Figure 6.2.2.9	Customer - Shopping Cart Detailed DFD	……...27
Figure 6.2.2.10	Customer - Shopping Cart Detailed DFD	……...27
Figure 6.2.2.11	Customer - Shopping Cart Detailed DFD	……...27
Figure 6.2.2.12	Customer – Authentication – UserProfile DFD	……..28
Figure 6.2.2.13	Authenticated User-Purchase Context DFD	……29
Figure 6.2.2.14	Customer - Authentication - Purchase DFD	……29
Figure 6.3.1	Splash Screen	……..32
Figure 6.3.2	Menu	……33
Figure 6.3.3	Book Details	……34
Figure 6.3.4	Advanced Search	……35

Figure 6.3.5 User Information ……35
Figure 6.3.6 Registration ……36
Figure 6.3.7 Login ……...36
Figure 6.3.8 Book ……37
Figure 6.3.9 Members ……..37
Figure 6.3.10 Administration Menu ……..38
Figure 6.3.11 My Info ……38
Figure 6.3.12 Categories ……39
Figure 6.3.13 Add Member ……39
Figure 6.3.14 Books ……..40
Figure 6.3.15 Card Types ……..40
Figure 6.3.16 Add Categories ……41
Figure 6.3.17 Book Edit ……41
Figure 6.3.18 Editorial ……...42
Figure 6.3.19 Orders ……..42
Figure 6.3.20 Edit Editorials ……..43
Figure 6.3.21 Editorial Categories ……43

List of Tables

Table 6.1.1.1: card_types ……14
Table 6.1.1.2: categories ……14
Table 6.1.1.3: editorial_categories ……14
Table 6.1.1.4: editorials……...15
Table 6.1.1.5: items ……15
Table 6.1.1.6: members ……..16
Table 6.1.1.7: orders ……...17

1. Introduction

In the world of software development there lots of improvement in the area of Architectural design and principles. The philosophies and implementation details are changing as the people guiding the development of the application. In this fantastic and yet sometimes complex world of software development there are some tried and true architecture patterns and software development guidelines employed by most architects. Also your design must have an ability to turn towards innovation instead of lending itself to common practices. Web services are one such area where architects must lean on their creative side and hope that their solutions are still successful. In this report we will explain an exciting voyage down the road of Web services application. From requirements to use cases, to database design, to component frameworks, to user interfaces, we will cover each and every aspect of system design required to build an application with collaborative Web services. The reason why we selected online Bookstore web service is everybody walking down the street has some idea about bookstores. The objective of this project is to develop an e- book store where books can be bought from the comfort of home through the Internet.

An online book store is a virtual store on the Internet where customers can browse the catalog and select books of interest. The selected books may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as credit card number. An e- mail notification is sent to the customer as soon as the order is placed.

2. Literature Review

Electronic Commerce (e-commerce) applications support the interaction between different parties participating in a commerce transaction via the network, as well as the management of the data involved in the process [2].

The increasing importance of e-commerce is apparent in the study conducted by researchers at the GVU (Graphics, Visualization, and Usability) Center at the Georgia Institute of Technology. In their summary of the findings from the eighth survey, the researchers report that "e-commerce is taking off both in terms of the number of users

shopping as well as the total amount people are spending via Internet based transactions".

Over three quarters of the 10,000 respondents report having purchased items online. The most cited reason for using the web for personal shopping was convenience (65%), followed by availability of vendor information (60%), no pressure from sales person (55%) and saving time (53%).

Although the issue of security remains the primary reasons why more people do not purchase terms online, the GVA survey also indicates that faith in the security of e-commerce is increasing. As more people gain confidence in current encryption technologies, more and more users can be expected to frequently purchase items online [11].

A good e-commerce site should present the following factors to the customers for better usability [11]:

- Knowing when an item was saved or not saved in the shopping cart.
- Returning to different parts of the site after adding an item to the shopping cart.
- Easy scanning and selecting items in a list.
- Effective categorical organization of products.
- Simple navigation from home page to information and order links for specific products.
- Obvious shopping links or buttons.
- Minimal and effective security notifications or messages.
- Consistent layout of product information.

Another important factor in the design of an e-commerce site is feedback [4]. The interactive cycle between a user and a web site is not complete until the web site responds to a command entered by the user. According to Norman [5], "feedback--sending back to the user information about what action has actually been done, what result has been accomplished--is a well known concept in the science of control and information theory. Imagine trying to talk to someone when you cannot even hear

your own voice, or trying to draw a picture with a pencil that leaves no mark: there would be no feedback".

Web site feedback often consists of a change in the visual or verbal information presented to the user. Simple examples include highlighting a selection made by the user or filling a field on a form based on a user's selection from a pull down list. Another example is using the sound of a cash register to confirm that a product has been added to an electronic shopping cart.

Completed orders should be acknowledged quickly. This may be done with an acknowledgment or fulfillment page. The amount of time it takes to generate and download this page, however, is a source of irritation for many e-commerce users. Users are quick to attribute meaning to events. A blank page, or what a user perceives to be "a long time" to receive an acknowledgment, may be interpreted as "there must be something wrong with the order." If generating an acknowledgment may take longer than what may be reasonably expected by the user, then the design should include intermediate feedback to the user indicating the progress being made toward acknowledgment or fulfillment.

Finally, feedback should not distract the user. Actions and reactions made by the web site should be meaningful. Feedback should not draw the user's attention away from the important tasks of gathering information, selecting products, and placing orders.

3. Implementation Technologies

3.1 Introduction

While there are numerous technologies for building web applications that serve dynamic content, the one that has really caught the attention of the development community is JavaServer Pages (JSP). And not without ample reason either. JSP not only enjoys cross-platform and cross-Web-server support, but effectively melds the power of server-side Java technology with the WYSIWYG features of static HTML pages.

JSP pages typically comprise of:

- Static HTML/XML components.
- Special JSP tags

Optionally, snippets of code written in the Java programming language called "scriptlets." Consequently, you can create and maintain JSP pages by conventional HTML/XML tools. It is important to note that the JSP specification is a standard extension defined on top of the Servlet API. Thus, it leverages all of your experience with servlets. There are significant differences between JSP and servlet technology. Unlike servlets, which is a programmatic technology requiring significant developer expertise, JSP appeals to a much wider audience. It can be used not only by developers, but also by page designers, who can now play a more direct role in the development life cycle. Another advantage of JSP is the inherent separation of presentation from content facilitated by the technology, due its reliance upon reusable component technologies like the JavaBeans component architecture and Enterprise JavaBeans technology. This course provides you with an in-depth introduction to this versatile technology, and uses the Tomcat JSP 1.1 Reference Implementation from the Apache group for running the example programs.

3.2 JSP Architecture

The purpose of JSP is to provide a declarative, presentation-centric method of developing servlets. As noted before, the JSP specification itself is defined as a standard extension on top the Servlet API. Consequently, it should not be too surprisingly that under the covers, servlets and JSP pages have a lot in common.

Typically, JSP pages are subject to a translation phase and a request processing phase. The translation phase is carried out only once, unless the JSP page changes, in which case it is repeated. Assuming there were no syntax errors within the page, the result is a JSP page implementation class file that implements the Servlet interface, as shown below.

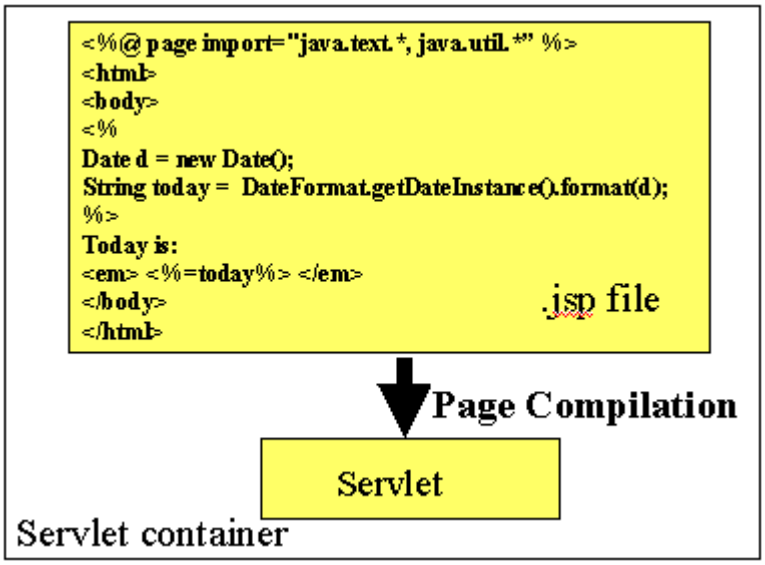


FIGURE 3.2.1 SERVLET CONTAINER

The translation phase is typically carried out by the JSP engine itself, when it receives an incoming request for the JSP page for the first time. Note that the JSP 1.1 specification also allows for JSP pages to be precompiled into class files. Precompilation may be especially useful in removing the start-up lag that occurs when a JSP page delivered in source form receives the first request from a client. Many details of the translation phase, like the location where the source and class files are stored are implementation dependent.

The JSP page implementation class file extends HttpJspBase, which in turn implements the Servlet interface. Observe how the service method of this class, _jspService(), essentially inlines the contents of the JSP page. Although _jspService() cannot be overridden, the developer can describe initialization and destroy events by providing implementations for the jspInit() and jspDestroy() methods within their JSP pages.

Once this class file is loaded within the servlet container, the _jspService() method is responsible for replying to a client's request. By default, the _jspService() method is dispatched on a separate thread by the servlet container in processing concurrent client requests, as shown below:

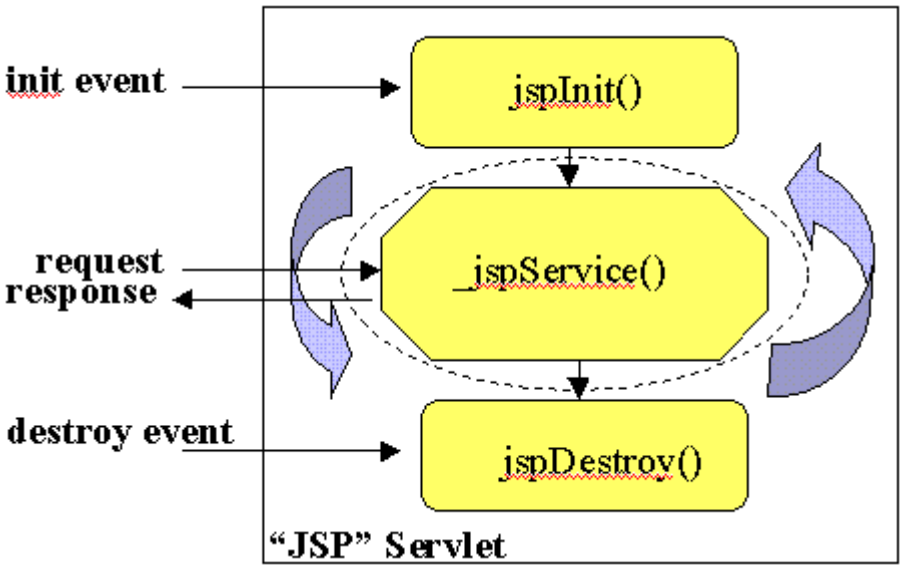


FIGURE 3.2.2 "JSP SERVLET"

3.2.1 JSP Access Models

The early JSP specifications advocated two philosophical approaches, popularly known as Model 1 and Model 2 architectures, for applying JSP technology. These approaches differ essentially in the location at which the bulk of the request processing was performed, and offer a useful paradigm for building applications using JSP technology.

Consider the Model 1 architecture, shown below:

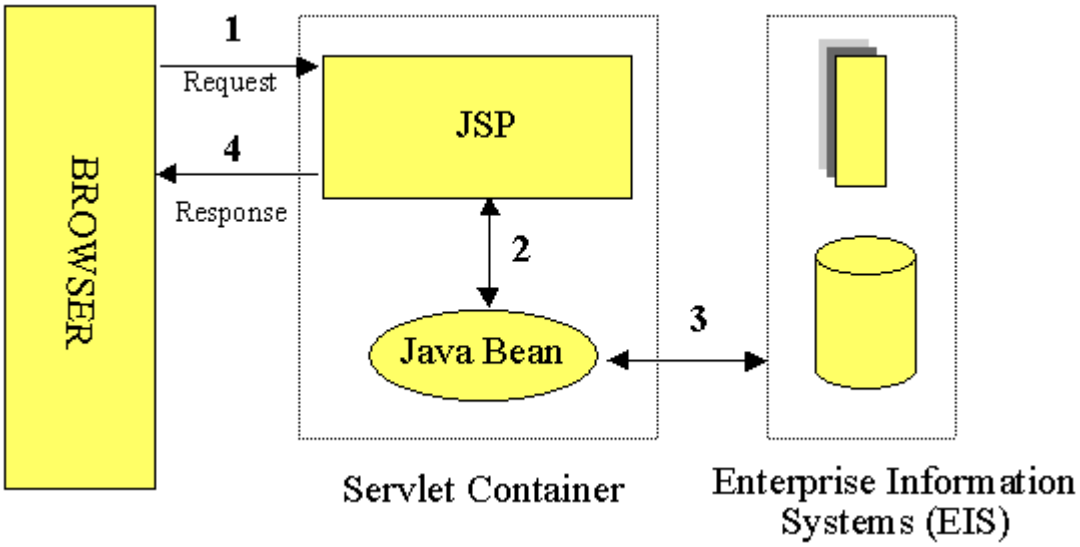


FIGURE 3.2.1.1 "JSP SERVLET"

In the Model 1 architecture, the incoming request from a web browser is sent directly to the JSP page, which is responsible for processing it and replying back to the client. There is still separation of presentation from content, because all data access is performed using beans.

Although the Model 1 architecture is suitable for simple applications, it may not be desirable for complex implementations. Indiscriminate usage of this architecture usually leads to a significant amount of scriptlets or Java code embedded within the JSP page, especially if there is a significant amount of request processing to be performed. While this may not seem to be much of a problem for Java developers, it is certainly an issue if your JSP pages are created and maintained by designers--which is usually the norm on large projects. Another downside of this architecture is that each of the JSP pages must be individually responsible for managing application state and verifying authentication and security.

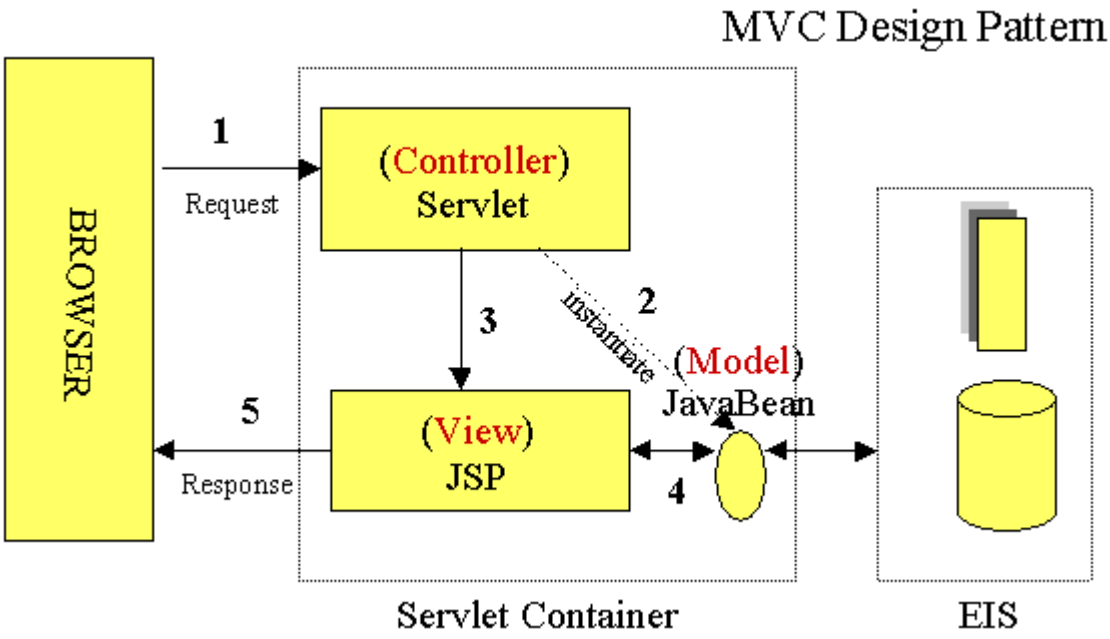


FIGURE 3.2.1.2 MVC DESIGN PATTERN

The Model 2 architecture, shown above, is a server-side implementation of the popular Model/View/Controller design pattern. Here, the processing is divided between presentation and front components. Presentation components are JSP pages that generate the HTML/XML response that determines the user interface when rendered by the browser. Front components (also known as controllers) do not handle

any presentation issues, but rather, process all the HTTP requests. Here, they are responsible for creating any beans or objects used by the presentation components, as well as deciding, depending on the user's actions, which presentation component to forward the request to. Front components can be implemented as either a servlet or JSP page.

The advantage of this architecture is that there is no processing logic within the presentation component itself; it is simply responsible for retrieving any objects or beans that may have been previously created by the controller, and extracting the dynamic content within for insertion within its static templates. Consequently, this clean separation of presentation from content leads to a clear delineation of the roles and responsibilities of the developers and page designers on the programming team. Another benefit of this approach is that the front components present a single point of entry into the application, thus making the management of application state, security, and presentation uniform and easier to maintain.

3.3. MySQL Database

In this project, MySQL is used as the backend database. MySQL is an open source database management system. The features of MySQL are given below:

· MySQL is a relational database management system. A relational database stores information in different tables, rather than in one giant table. These tables can be referenced to each other, to access and maintain data easily.

· MySQL is open source database system. The database software can be used and modify by anyone according to their needs.

· It is fast, reliable and easy to use. To improve the performance, MySQL is multithreaded database engine. A multithreaded application performs many tasks at the same time as if multiple instances of that application were running simultaneously.

In being multithreaded MySQL has many advantages. A separate thread handles each incoming connection with an extra thread that is always running to manage the connections. Multiple clients can perform read operations simultaneously, but while writing, only hold up another client that needs access to the data being updated. Even though the threads share the same process space, they execute individually and because of this separation, multiprocessor machines can spread the thread across

many CPUs as long as the host operating system supports multiple CPUs. Multithreading is the key feature to support MySQL's performance design goals. It is the core feature around which MySQL is built. MySQL database is connected to JSP using an ODBC driver. Open Database Connectivity (ODBC) is a widely accepted application-programming interface (API) for database access. The ODBC driver is a library that implements the functions supported by ODBC API. It processes ODBC function calls, submits SQL requests to MySQL server, and returns results back to the application. If necessary, the driver modifies an application's request so that the request conforms to syntax supported by MySQL.

3.4 Integrating the Website and Database

Customers ordering from an e-commerce website need to be able to get information about a vendor's products and services, ask questions, select items they wish to purchase, and submit payment information. Vendors need to be able to track customer inquiries and preferences and process their orders. So a well organized database is essential for the development and maintenance of an e-commerce site [3].

In a static Web page, content is determined at the time when the page is created. As users access a static page, the page always displays the same information. Example of a static Web page is the page displaying company information. In a dynamic Web page, content varies based on user input and data received from external sources. We use the term "data-based Web pages" to refer to dynamic Web pages deriving some or all of their content from data files or databases.

A data-based Web page is requested when a user clicks a hyperlink or the submit button on a Web page form. If the request comes from clicking a hyperlink, the link specifies either a Web server program or a Web page that calls a Web server program. In some cases, the program performs a static query, such as "Display all items from the Inventory". Although this query requires no user input, the results vary depending on when the query is made. If the request is generated when the user clicks a form's submit button, instead of a hyperlink, the Web server program typically uses the form inputs to create a query. For example, the user might select five books to be purchased

and then submit the input to the Web server program. The Web server program then services the order, generating a dynamic Web page response to confirm the transaction. In either case, the Web server is responsible for formatting the query results by adding HTML tags. The Web server program then sends the program's output back to the client's browser as a Web page.

3. Web Page Programming Options

An e-commerce organization can create data-based Web pages by using server side and client-side processing technologies or a hybrid of the two. With server-side processing, the Web server receives the dynamic Web page request, performs all processing necessary to create the page, and then sends it to the client for display in the client's browser. Client-side processing is done on the client workstation by having the client browser execute a program that interacts directly with the database.

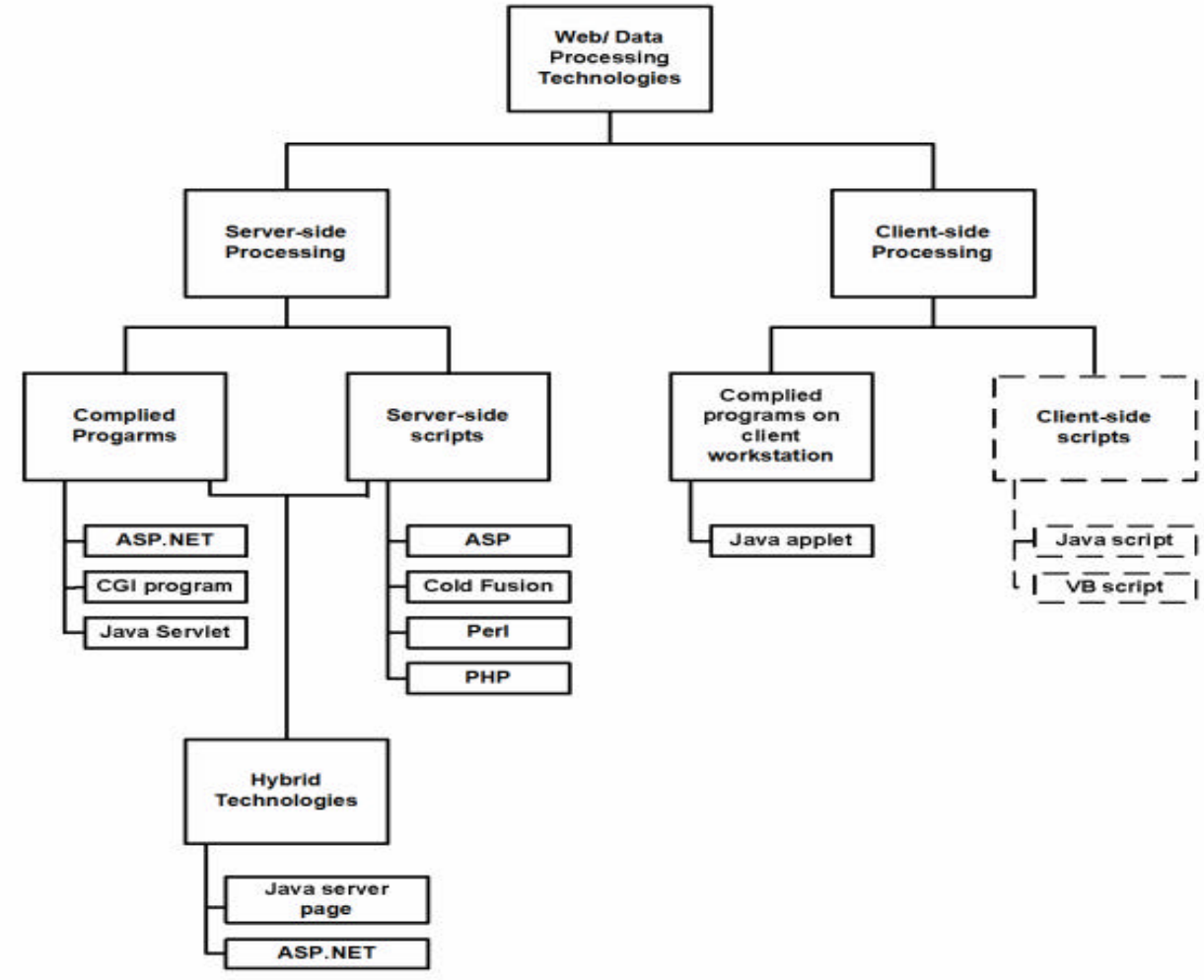


FIGURE 4.1 WEB PAGE PROGRAMMING OPTION

Figure 22 (reproduced from [3]) outlines commonly used server-side, client-side, and hybrid Web and data processing technologies; client-side scripts are in dashed lines to indicate they are unable to interact directly with a database or file but are used to validate user input on the client, then send the validated inputs to the server for further processing.

5. The Shopping Cart Application

The objective of this application is to provide the user an online website where they can buy books from the comfort of their home. A shopping cart is used for the purpose. The user can select the desired books, place them in the shopping cart and purchase them using a Credit Card. The user's order will be shipped according to the type of shipping selected at the time of placing the order.

Website consists of the following web pages:

1. AdminBooks.jsp
2. AdminMenu.jsp
3. AdvSearch.jsp
4. AdvSearch.jsp
5. BookMaint.jsp
6. Books.jsp
7. CardTypesGrid.jsp
8. CardTypesRecord.jsp
9. CategoriesGrid.jsp
10. CategoriesRecord.jsp
11. Common.jsp
12. Default.jsp
13. dirinfo.cci
14. EditorialCatGrid.jsp
15. EditorialCatRecord.jsp
16. EditorialsGrid.jsp
17. EditorialsRecord.jsp
18. Footer.jsp
19. Header.jsp

20. Index.html
21. Login.jsp
22. MembersGrid.jsp
23. MembersInfo.jsp
24. MembersRecord.jsp
25. MyInfo.jsp
26. OrdersGrid.jsp
27. OrdersRecord.jsp
28. Registration.jsp
29. ShoppingCart.jsp
30. ShoppingCartRecord.jsp

6. Project Design

In order to design a web site, the relational database must be designed first. Conceptual design can be divided into two parts: The data model and the process model. The data model focuses on what data should be stored in the database while the process model deals with how the data is processed. To put this in the context of the relational database, the data model is used to design the relational tables. The process model is used to design the queries that will access and perform operations on those tables.

6.1 Data Model

A data model is a conceptual representation of the data structures that are required by a database. The first step in designing a database is to develop an Entity-Relation Diagram (ERD). The ERD serves as a blue print from which a relational database maybe deduced. Figure 6.1.1 shows the ERD for the project and later we will show the transformation from ERD to the Relational model.

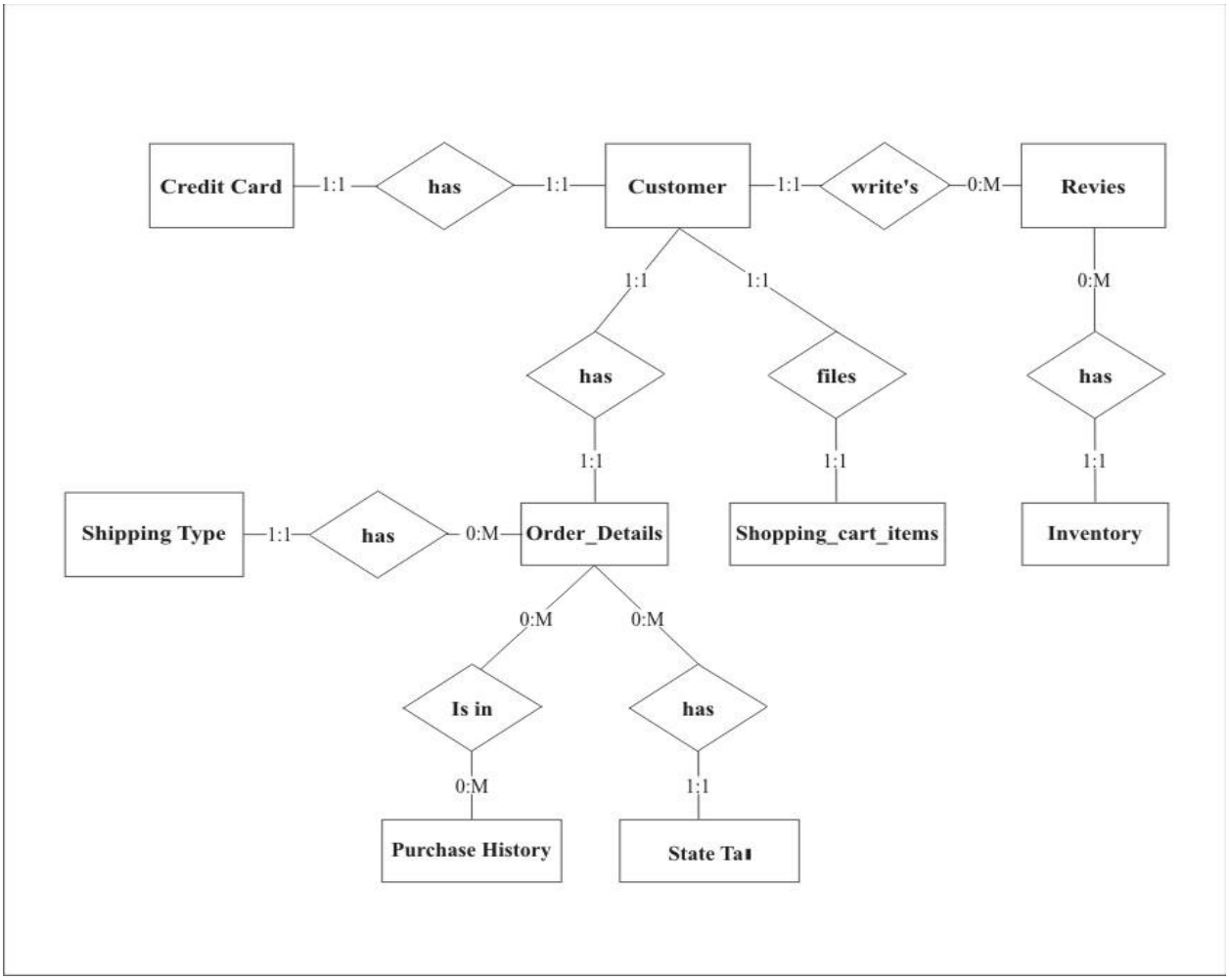


Figure6.1.1 Entity relationship diagram

Entity A matches exactly one record in entity B and every record in B matches exactly one record in A. One to many means that every record in A matches zero or more records in B and every record in B matches exactly one record in A. If there is a one to many relationships between two entities, then these entities are represented as Associative Entities. In the Relational Database model, each of the entities will be transformed into a table. The tables are shown below along with the attributes.

6.1.1 Database Design

In this section, the basic structure of the tables composing the database for the project are shown along with information about primary and foreign keys.

Table 6.1.1.1: card_types

Columns Name	Type	Size
card_type_id	Long Integer	4
name	Text	50

Table 6.1.1.2 : categories

Columns Name	**Type**	**Size**
category_id	**Long Integer**	**4**
name	**Text**	**50**

Relationships

Categories ---- items

categories		items
category_id	1 ∞	category_id
Attributes:		Enforced
RelationshipType:		One-To-Many

Table 6.1.1.3 : editorial_categories

Columns Name	Type	Size
editorial_cat_id	Long Integer	4
editorial_cat_name	Text	50

Relationships

editorial_categories -- editorials

editorial_categories	editorials
editorial_cat_id	editorial_cat_id

Attributes:	Not Enforced
RelationshipType:	One-To-Many

Table 6.1.1.4 : editorials

Columns Name	Type	Size
article_id	Long Integer	4
editorial_cat_id	Long Integer	4
article_title	Text	200
article_desc	Memo	
item_id	Long Integer	4

Relationships

editorial_categories --- editorials

editorial_categories	editorials
editorial_cat_id	editorial_cat_id
Attributes:	Not Enforced
RelationshipType:	One-To-Many

Table 6.1.1.5 : items

Columns Name	Type	Size
item_id	Long Integer	4
category_id	Long Integer	4
name	Text	255
author	Text	100
price	Double	8
product_url	Text	255
image_url	Text	100
notes	Memo -	
is_recommended	Byte	1
rating	Integer	2
rating_count	Long Integer	4

Relationships

Categories -- items

categories			**items**
category_id	1	∞	category_id

Attributes: Enforced

RelationshipType: One-To-Many

Items -- orders

items			orders
item_id	1	∞	item_id

Attributes: Enforced

RelationshipType: One-To-Many

Table 6.1.1.6 : members

Columns Name	**Type**	**Size**
member_id	Long Integer	4
member_login	Text	20
member_password	Text	20
member_level	Long Integer	4
first_name	Text	50
last_name	Text	50
email	Text	50
phone	Text	50
address	Text	50
notes	Memo	-
card_type_id	Long Integer	4
card_number	Text	50

Relationships

Members -- orders

members			orders
member_id	1	∞	member_id
Attributes:			Enforced
RelationshipType:			One-To-Many

Table 6.1.1.7 : orders

ColumnsName	Type	Size
order_id	Long Integer	4
member_id	Long Integer	4
item_id	Long Integer	4
quantity	Long Integer	4

Relationships

Items -- orders

items			orders
item_id	1	∞	item_id
Attributes:			Enforced
RelationshipType:			One-To-Many

Members -- orders

members			orders
member_id	1	∞	member_id
Attributes:			Enforced
RelationshipType:			One-To-Many

Relationships

Categories -- items

categories			items
category_id	1	∞	category_id

Attributes: Enforced

RelationshipType: One-To-Many

Items -- orders

items			orders
item_id	1	∞	item_id

Attributes: Enforced

RelationshipType: One-To-Many

Members -- orders

members			orders
member_id	1	∞	member_id

Attributes: Enforced

RelationshipType: One-To-Many

editorial_categories -- editorials

editorial_categories			editorials
editorial_cat_id			editorial_cat_id

Attributes: Not Enforced

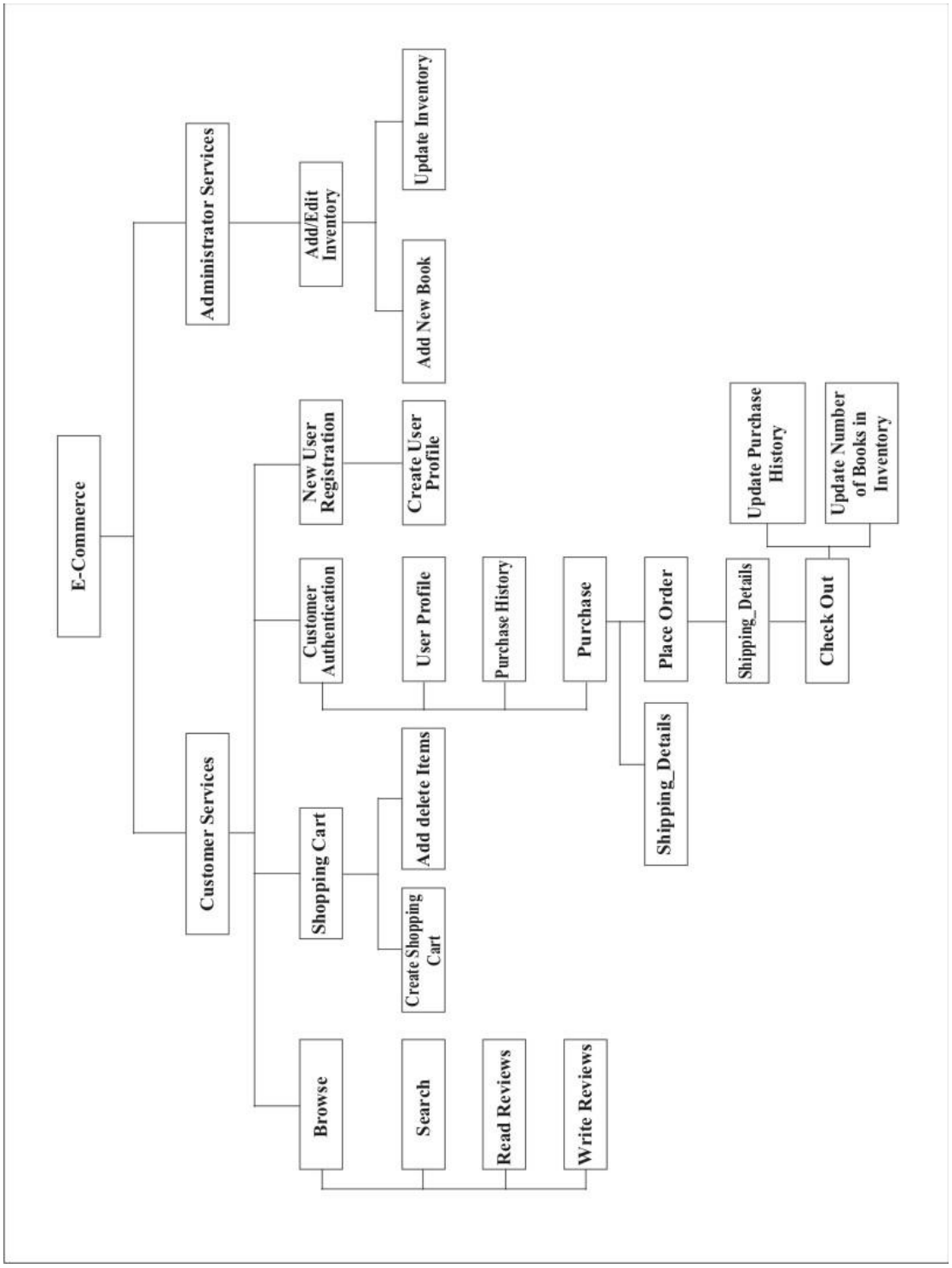
RelationshipType: One-To-Many

6.2. Process Model

A Process Model tells us about how the data is processed and how the data flows from one table to another to gather the required information. This model consists of the Functional Decomposition Diagram and Data Flow Diagram.

6.2.1. Functional Decomposition Diagram

A decomposition diagram shows a top-down functional decomposition of a system and exposes the system's structure. The objective of the Functional Decomposition is to break down a system step by step, beginning with the main function of a system and continuing with the interim levels down to the level of elementary functions. The diagram is the starting point for more detailed process diagrams, such as data flow diagrams (DFD). Figure 6.2.1.1 shows the Functional Decomposition Diagram for this project.

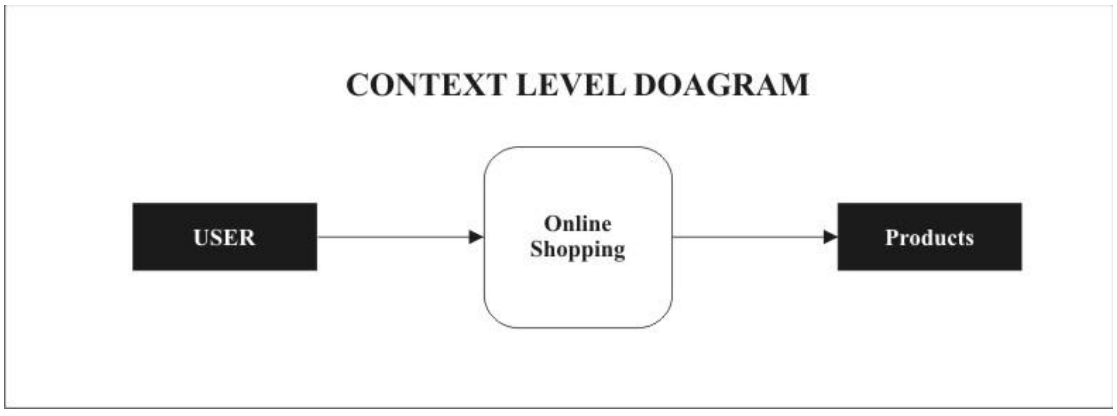

E-Commerce
Administrator Services
Add/Edit Inventory
Update Inventory
Add New Book
Customer Services
New User Registration
Create User Profile
Customer Authentication
User Profile
Purchase History
Purchase
Place Order
Shipping_Details
Shipping_Details
Update Purchase History
Update Number of Books in Inventory
Check Out
Shopping Cart
Add delete Items
Create Shopping Cart
Browse
Search
Read Reviews
Write Reviews

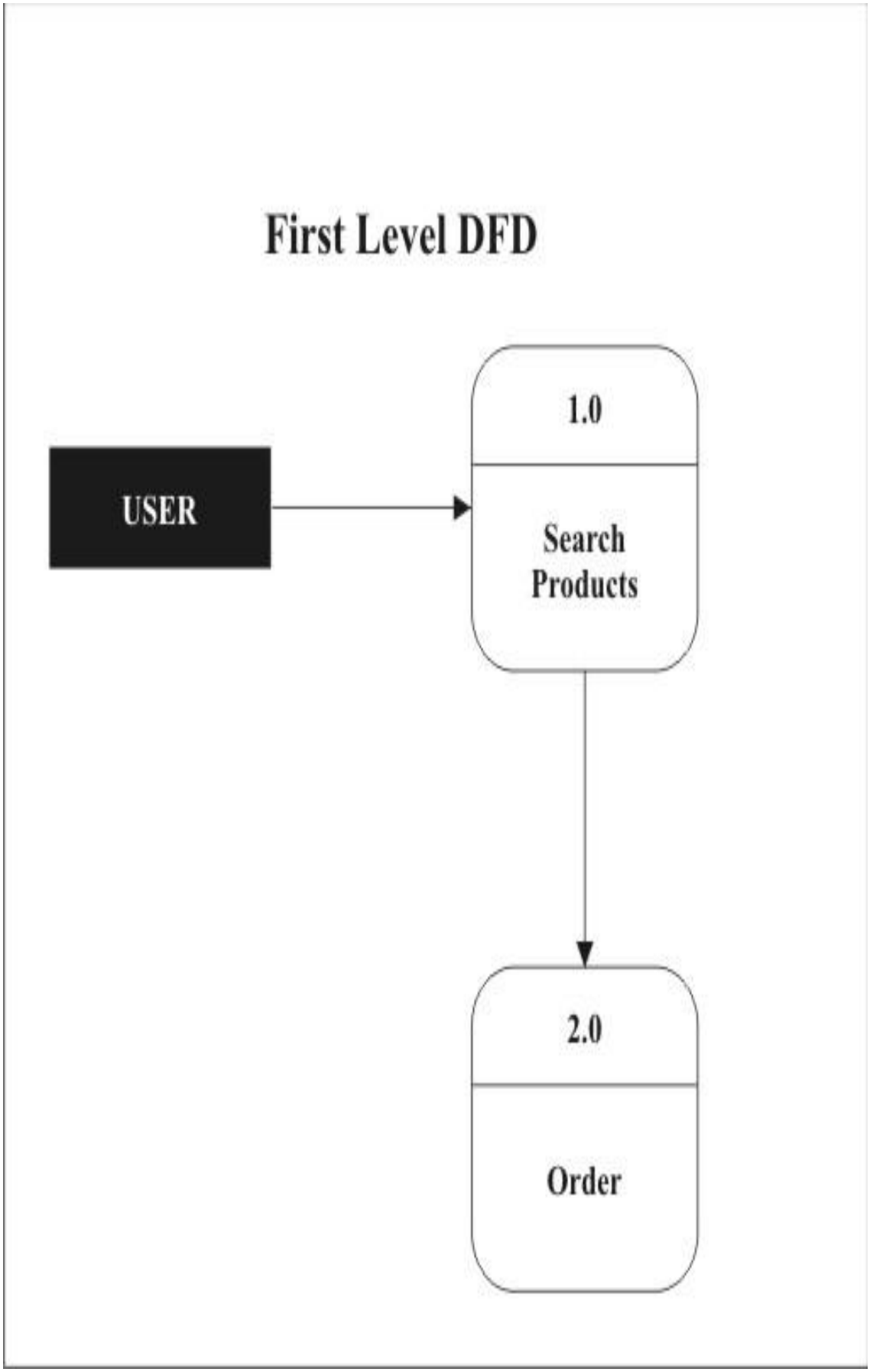
6.2.2 Data Flow Diagram (DFD)

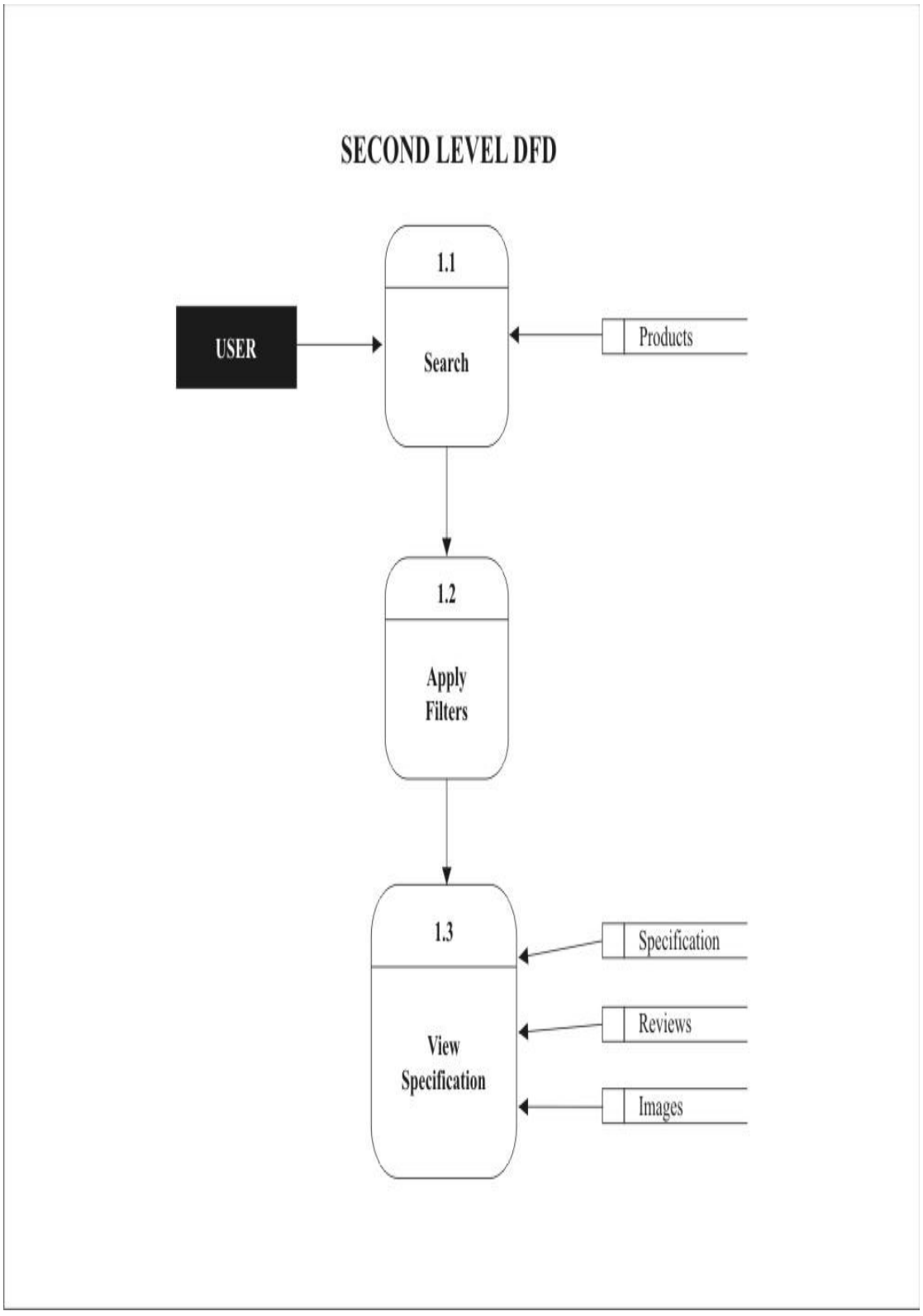
Data Flow Diagrams show the flow of data from external entities into the system, and from one process to another within the system. There are four symbols for drawing a DFD:

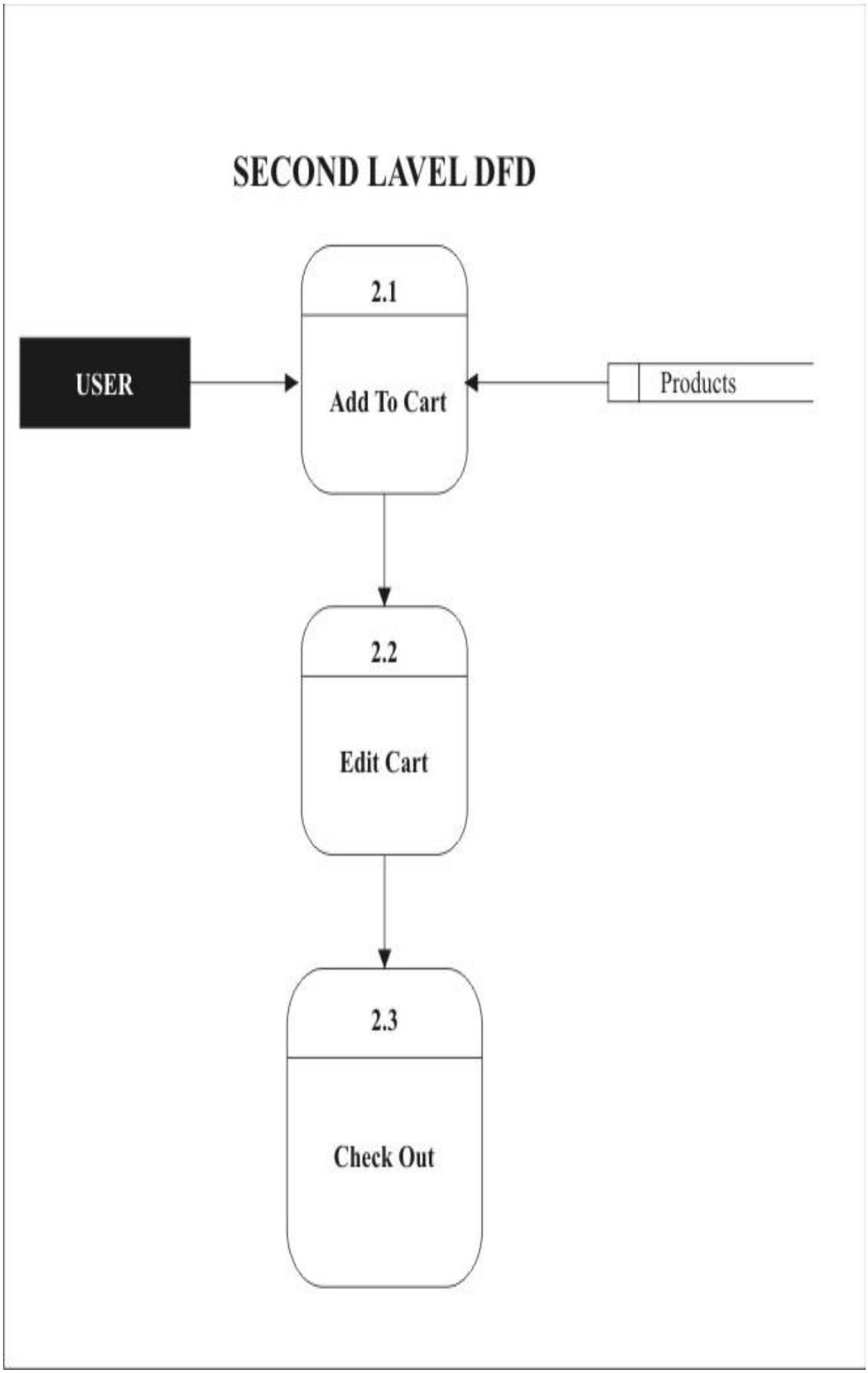
1. Rectangles representing external entities, which are sources or destinations of data.
2. Ellipses representing processes, which take data as input, validate and process it and output it.
3. Arrows representing the data flows, which can either, be electronic data or physical items.
4. Open-ended rectangles or a Disk symbol representing data stores, including electronic stores such as databases or XML files and physical stores such as filing cabinets or stacks of paper.

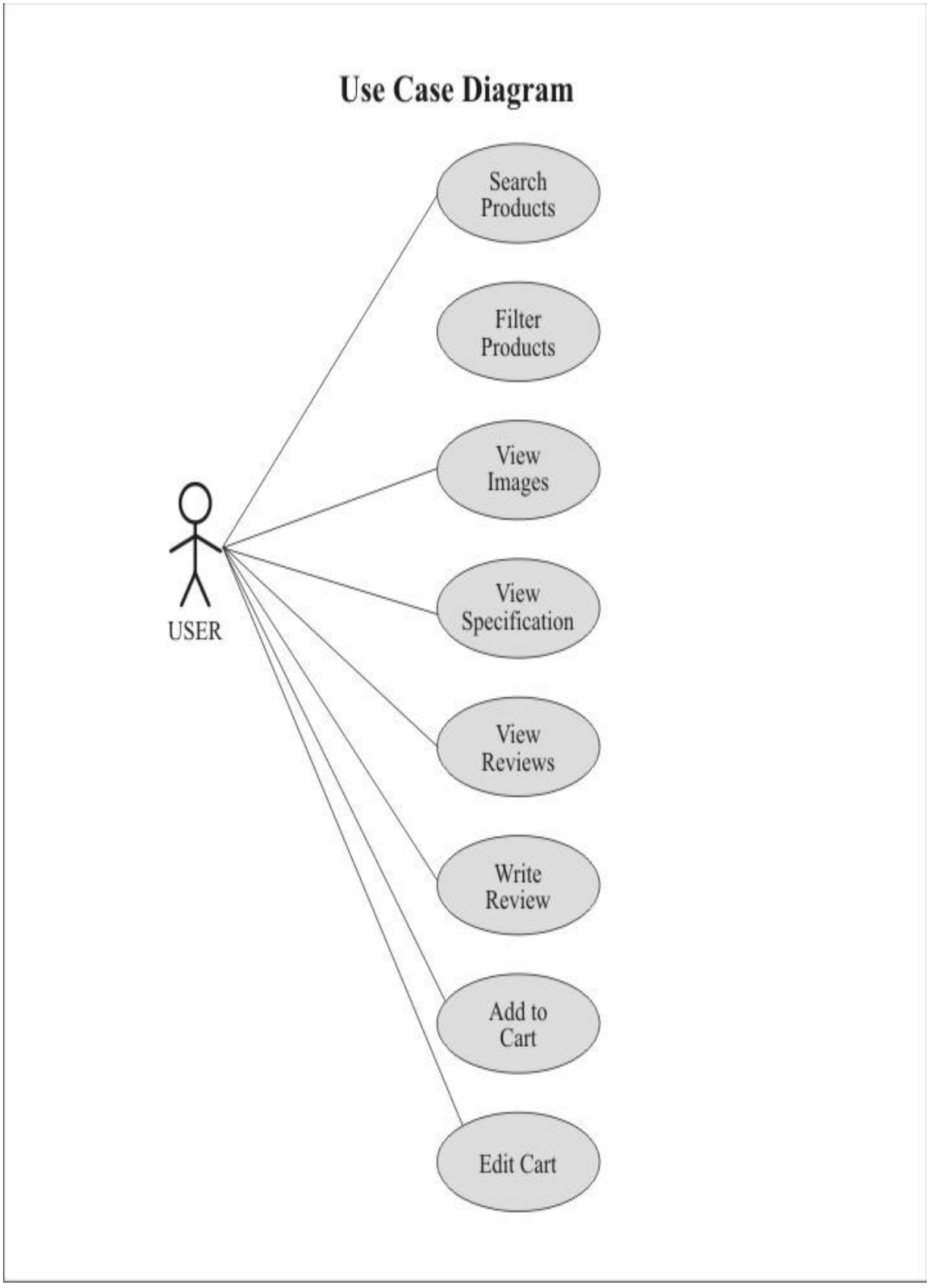
Figures 6.2.2.1 – 6.2.2.14 are the Data Flow Diagrams for the current system. Each process within the system is first shown as a Context Level DFD and later as a Detailed DFD. The Context Level DFD provides a conceptual view of the process and its surrounding input, output and data stores. The Detailed DFD provides a more detailed and comprehensive view of the interaction among the sub-processes within the system.



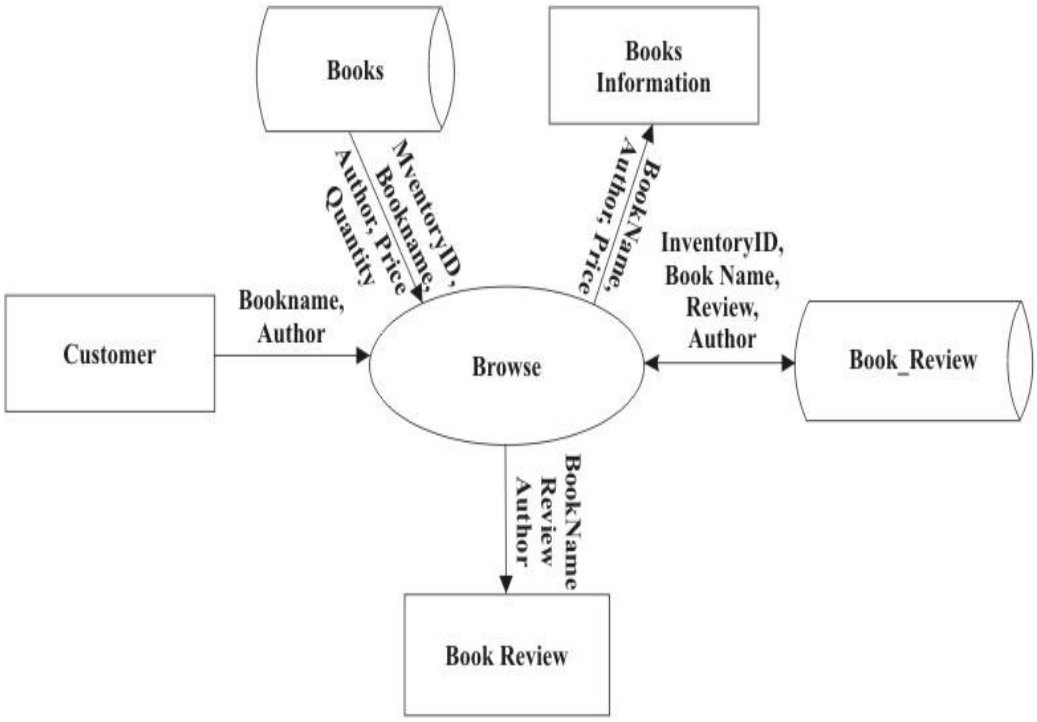

First Level DFD
USER
1.0
Search Products
2.0
Order


SECOND LEVEL DFD
1.1
Search
USER
Products
1.2
Apply
Filters
1.3
View
Specification
Specification
Reviews
Images

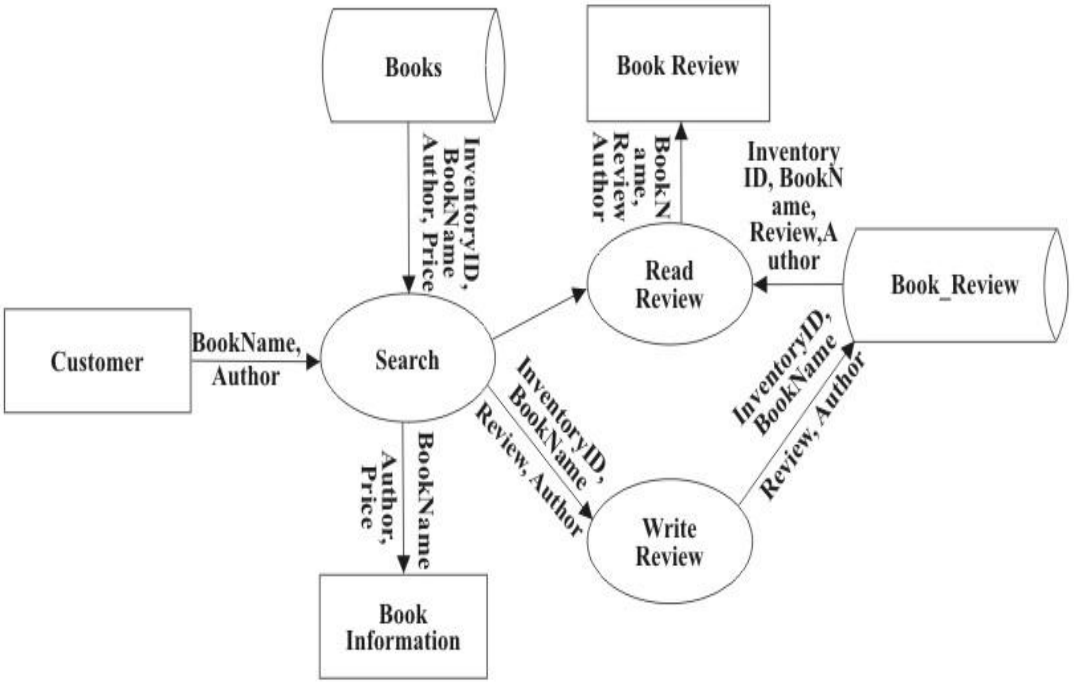

SECOND LAVEL DFD
2.1
USER
Add To Cart
Products
2.2
Edit Cart
2.3
Check Out


Use Case Diagram
Search Products
Filter Products
View Images
View Specification
View Reviews
Write Review
Add to Cart
Edit Cart
USER

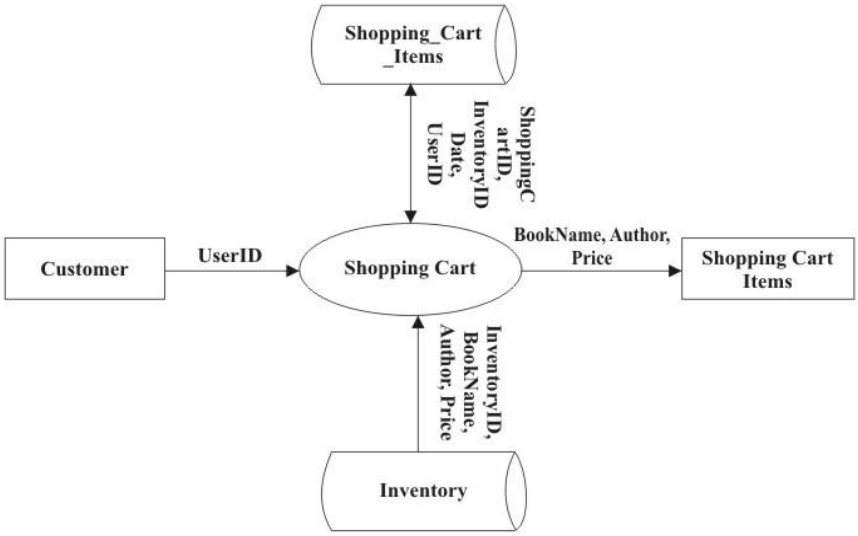
Customer-Browse Context DFD



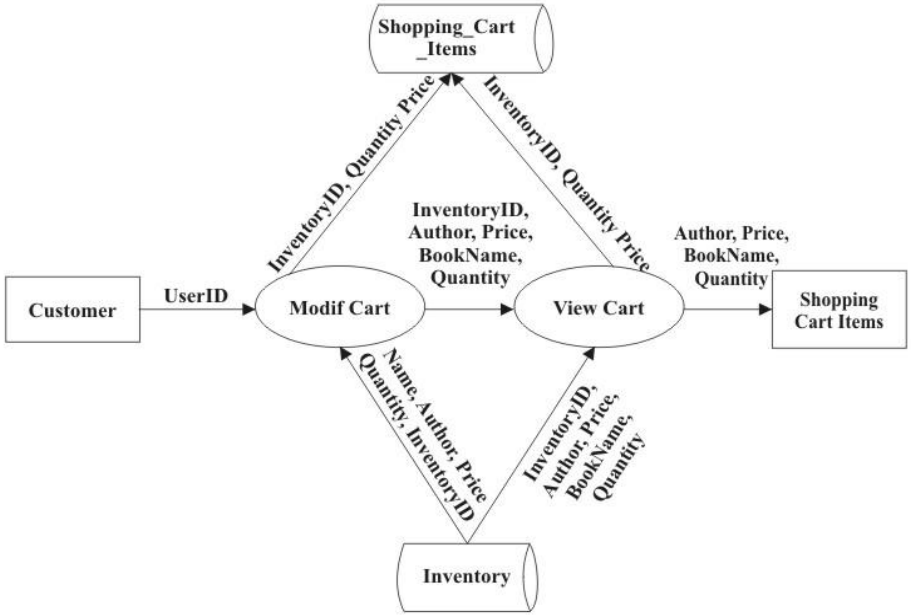
Customer-Browse Detailed DFD



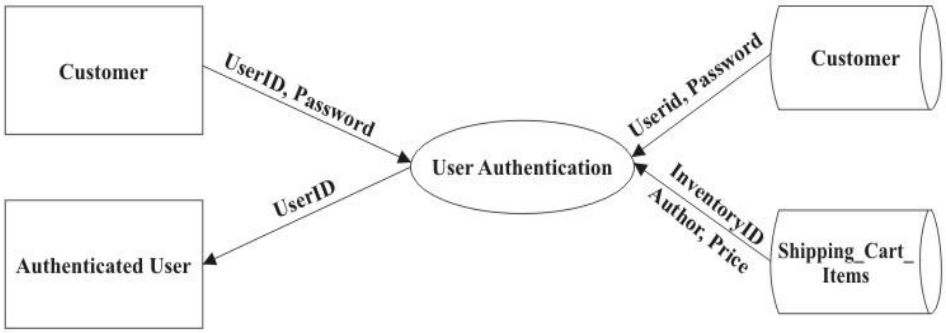
Customer-ShoppingCart Context DFD



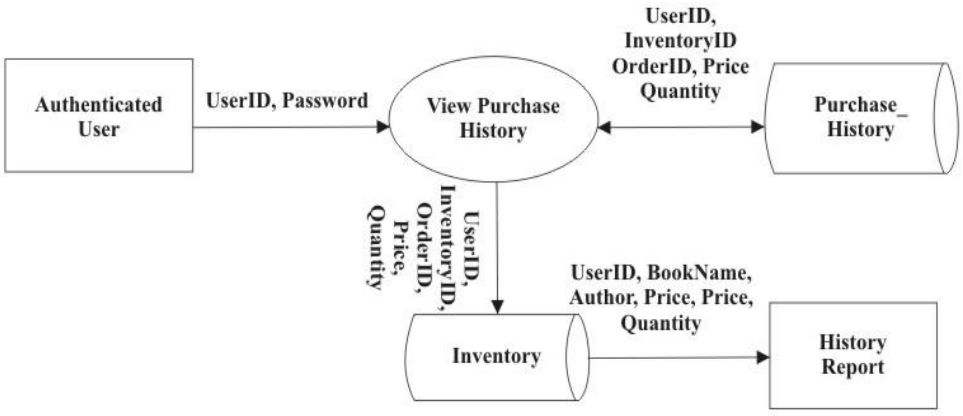
Customer - ShoppingCart Detailed DFD



Customer-Authentication Context DFD

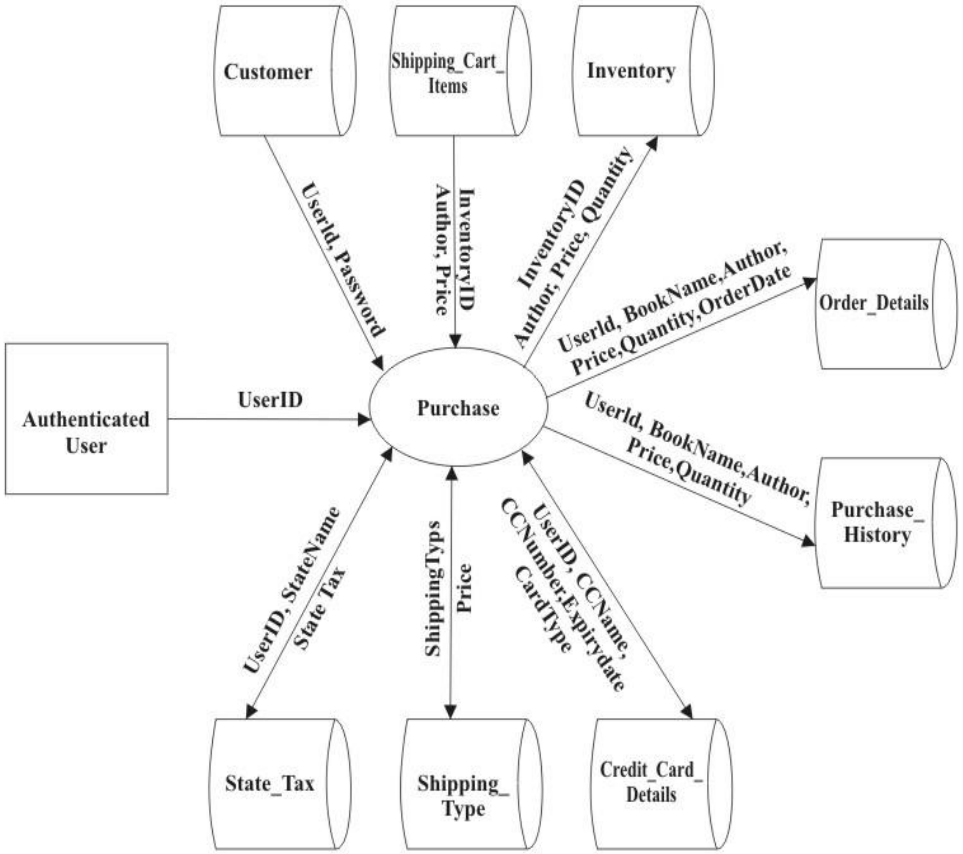


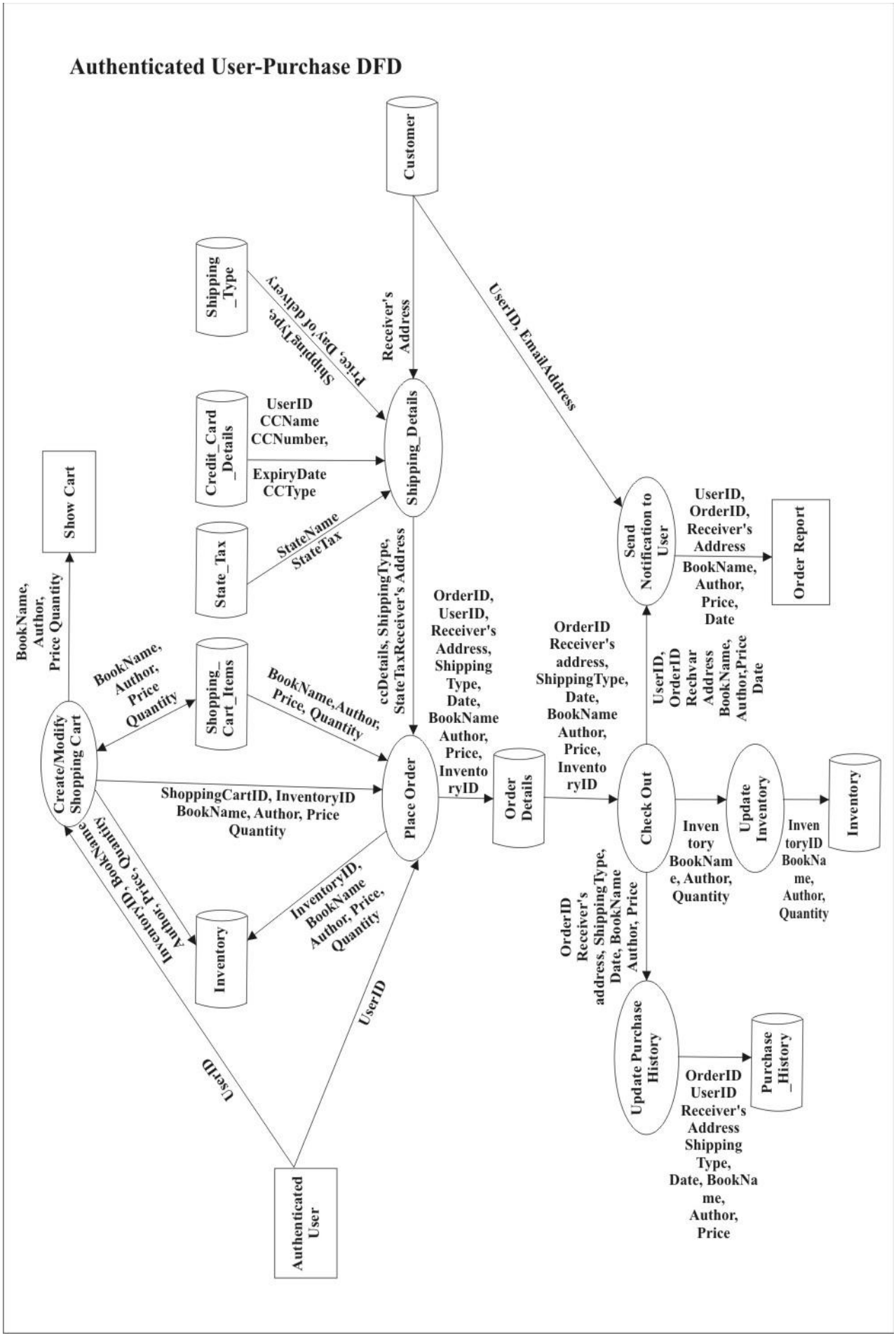
Customer-Authentication-PurchaseHistory DFD

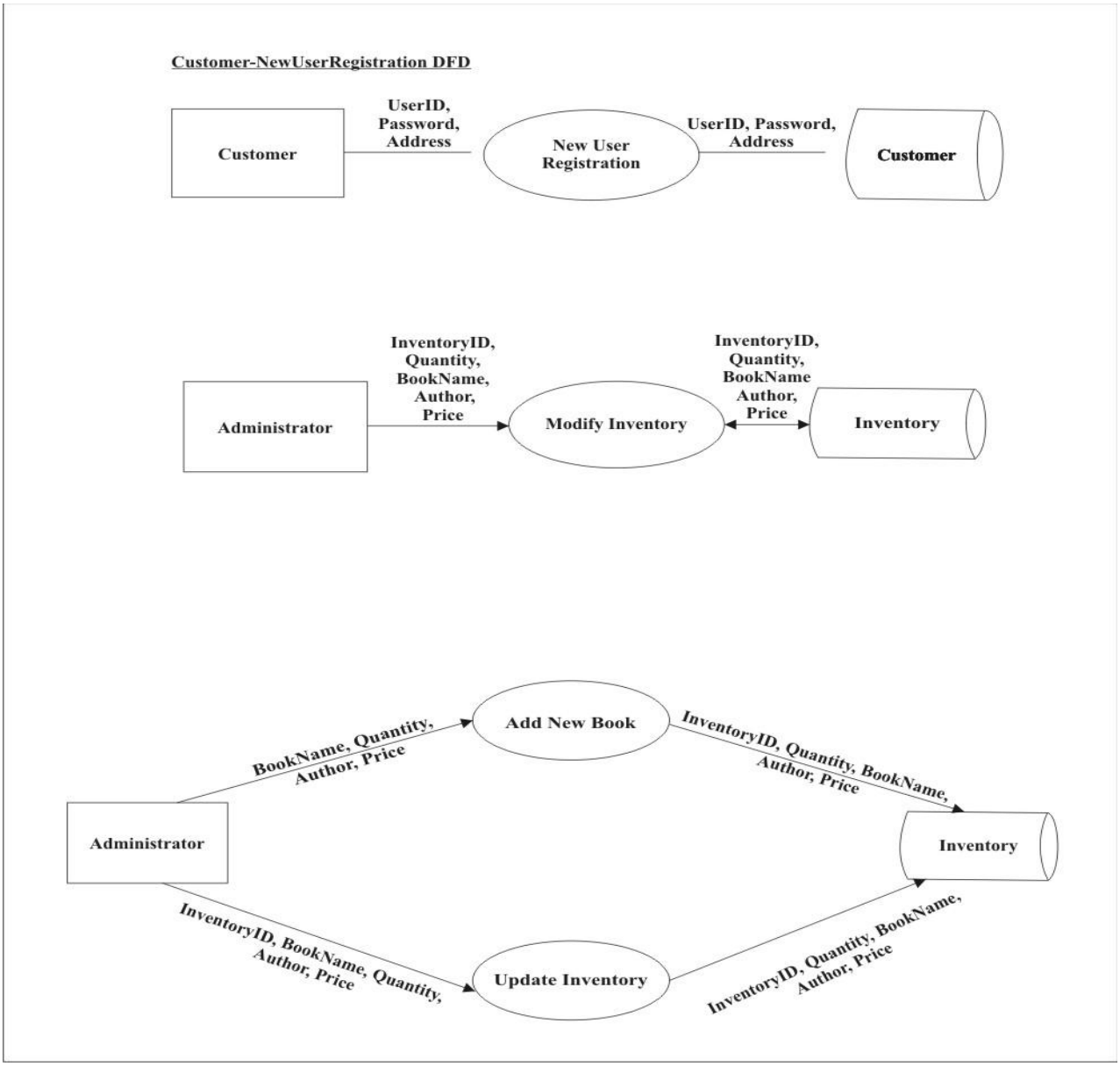


Customer-Authentication-UserProfile DFD

Authenticated User-Purchase Context DFD




Authenticated User-Purchase DFD
Customer
Shipping_Type
ShippingType, Price, Day of delivery
Receiver's Address
UserID, EmailAddress
Credit_Card_Details
UserID
CCName
CCNumber,
ExpiryDate
CCType
Shipping_Details
Show Cart
State_Tax
State/Name
StateTax
Send Notification to User
UserID, OrderID, Receiver's Address
BookName, Author, Price, Date
Order Report
BookName, Author, Price Quantity
BookName, Author, Price Quantity
Shopping_Cart_Items
BookName,Author, Price, Quantity
ccDetails, ShippingType, StateTaxReceiver's Address
OrderID, UserID, Receiver's Address, Shipping Type, Date, BookName Author, Price, InventoryID
OrderID Receiver's address, ShippingType, Date, BookName Author, Price, InventoryID
UserID, OrderID Rechvar Address BookName, Author,Price Date
Create/Modify Shopping Cart
ShoppingCartID, InventoryID BookName, Author, Price Quantity
Place Order
Order Details
Check Out
Update Inventory
Inventory
Inventory BookName, Author, Quantity
InventoryID, BookName, Price, Quantity
InventoryID, BookName, Author, Price, Quantity
Inventory
InventoryID BookName, Author, Quantity
OrderID Receiver's address, ShippingType, Date, BookName Author, Price
Update Purchase History
Purchase_History
OrderID UserID Receiver's Address Shipping Type, Date, BookName, Author, Price
UserID
UserID
Authenticated User



6.3 User Interface Design

Before implementing the actual design of the project, a few user interface designs were constructed to visualize the user interaction with the system as they browse for books, create a shopping cart and purchase books. The user interface design will closely follow our Functional Decomposition Diagram (Figure 2). Figures 15 – 20 show the initial designs of the web pages.

Designed by Er sanjay agal

As Minor project of Mtech

Mewar University , chittor

FIGURE 6.3.1 SPLASH SCREEN

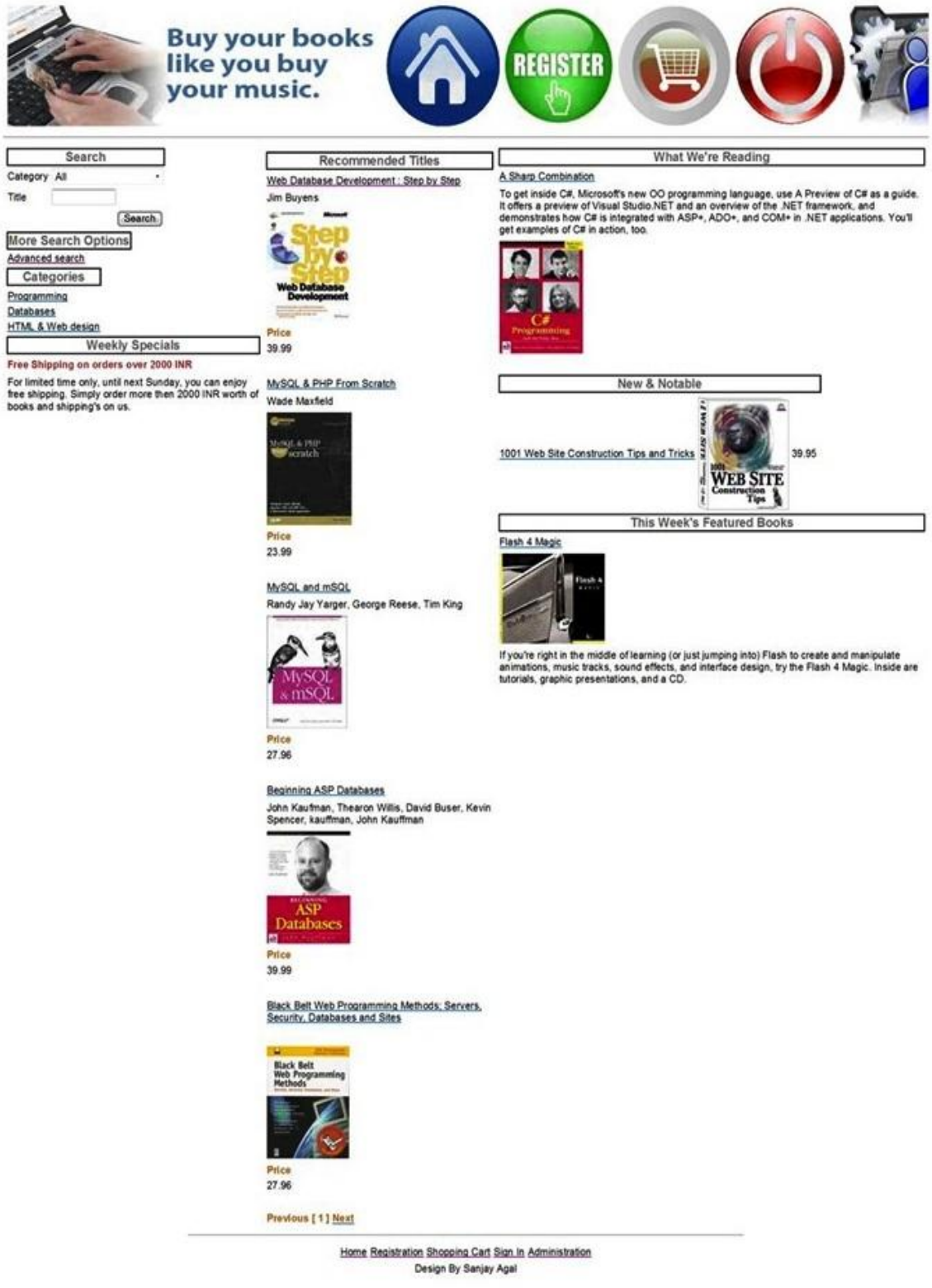


Figure 6.3.2 Menu

Figure 6.3.3 Book Details

Figure 6.3.4 Advanced Search

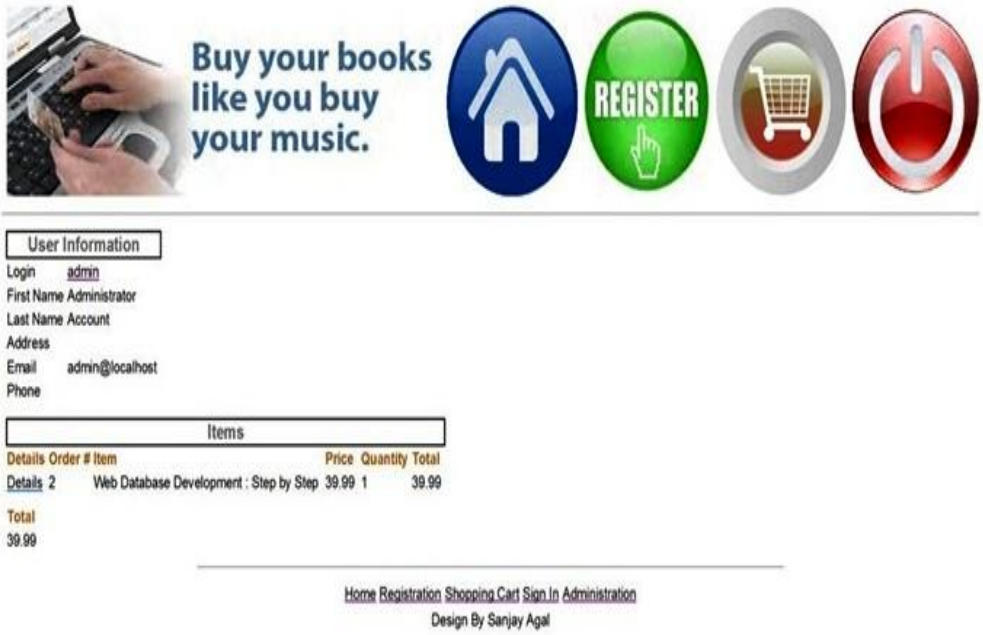


Figure 6.3.5 user information

Figure 6.3.6 registration

Figure 6.3.7 login

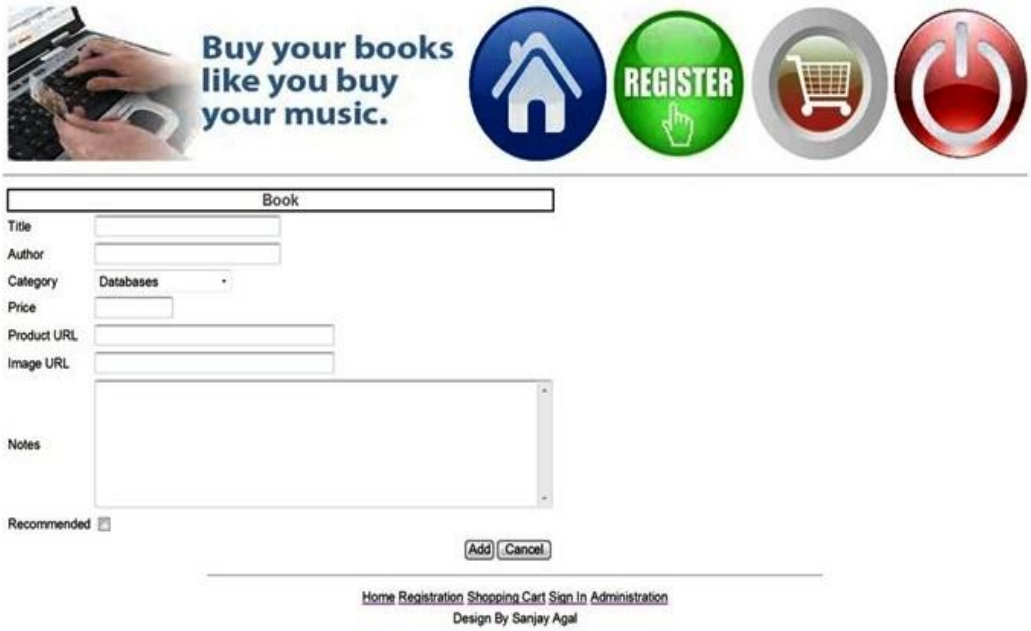


Figure 6.3.8 book

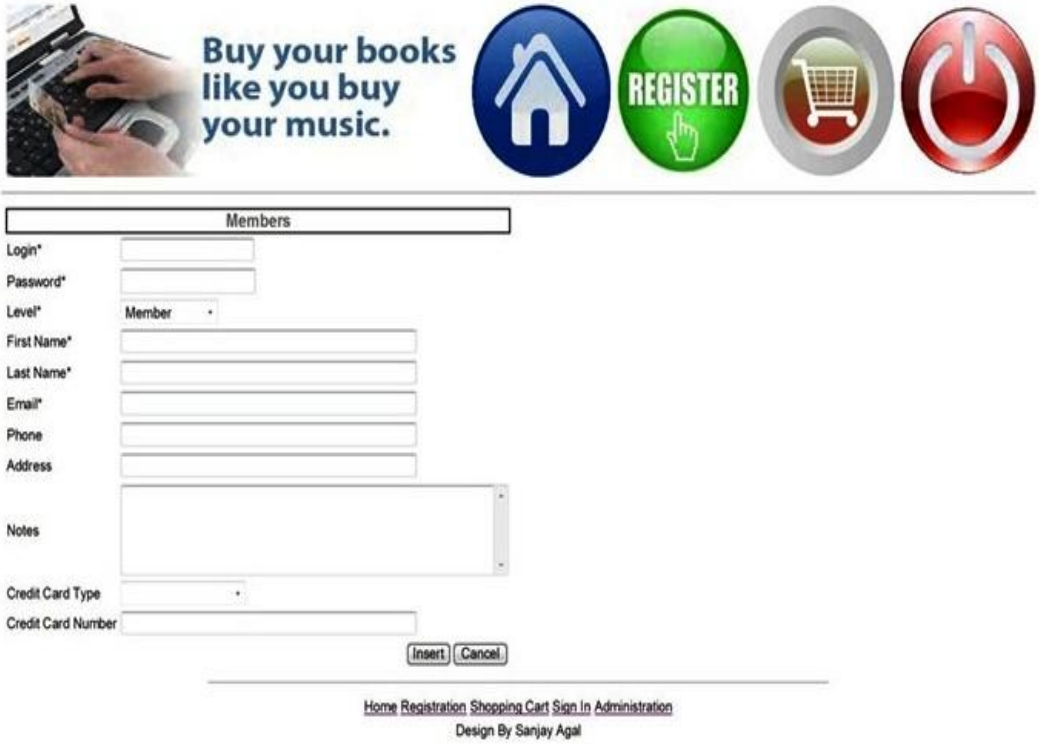


Figure 6.3.9 members

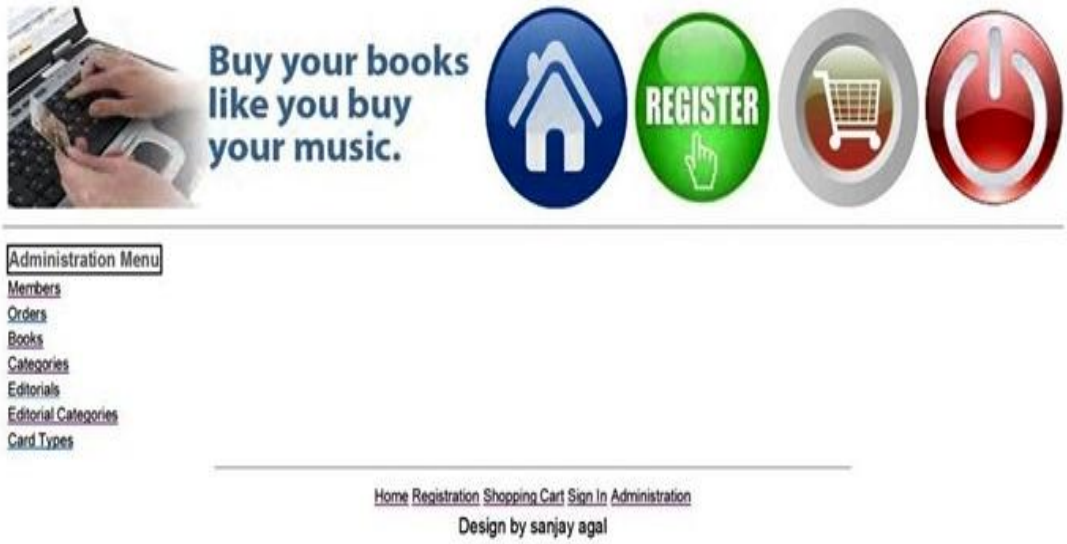


Figure 6.3.10 administration menu

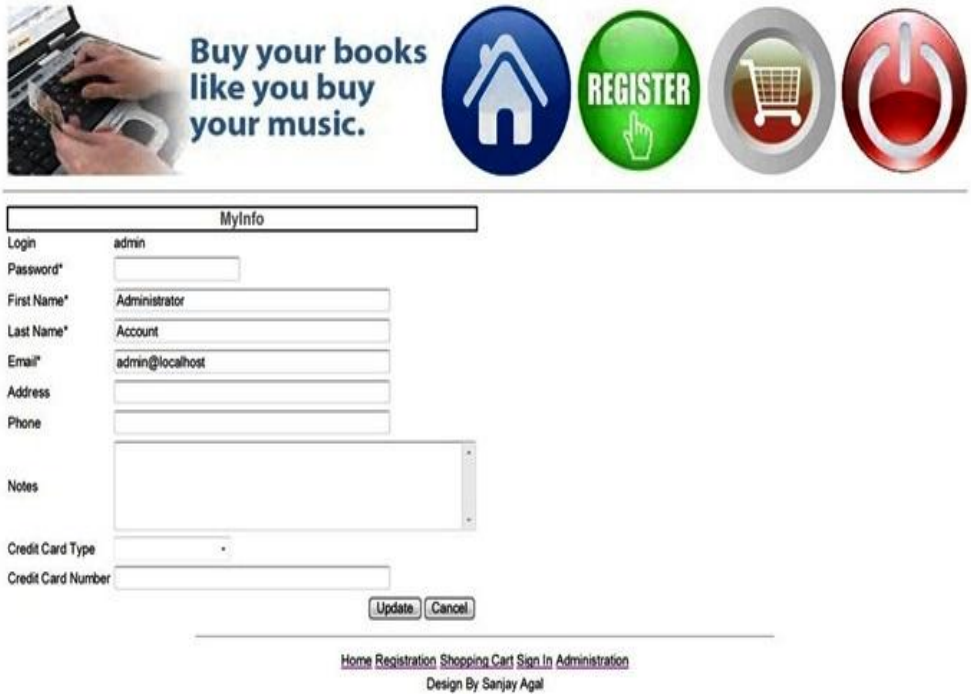


Figure 6.3.11 my info

Figure 6.3.12 categories

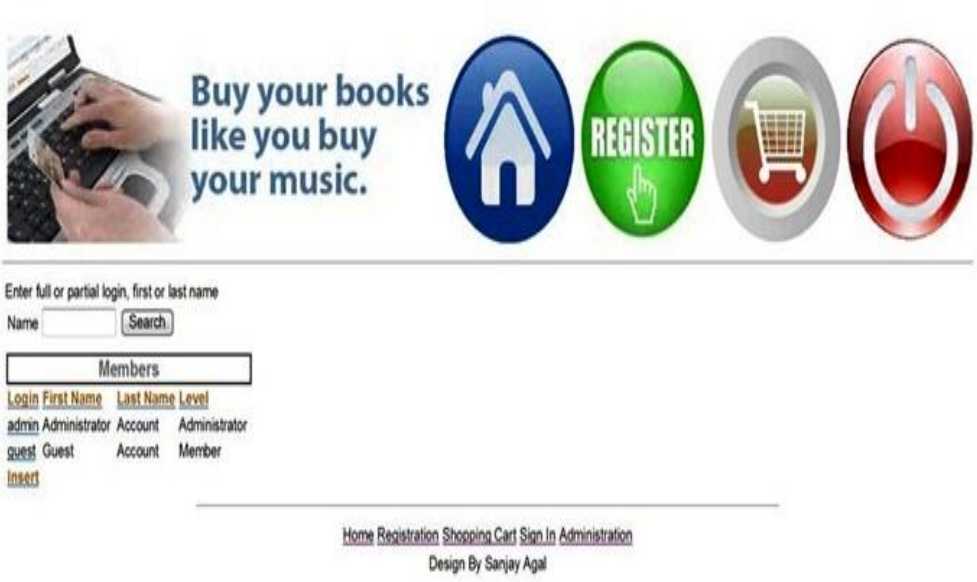


Figure 6.3.13 add member

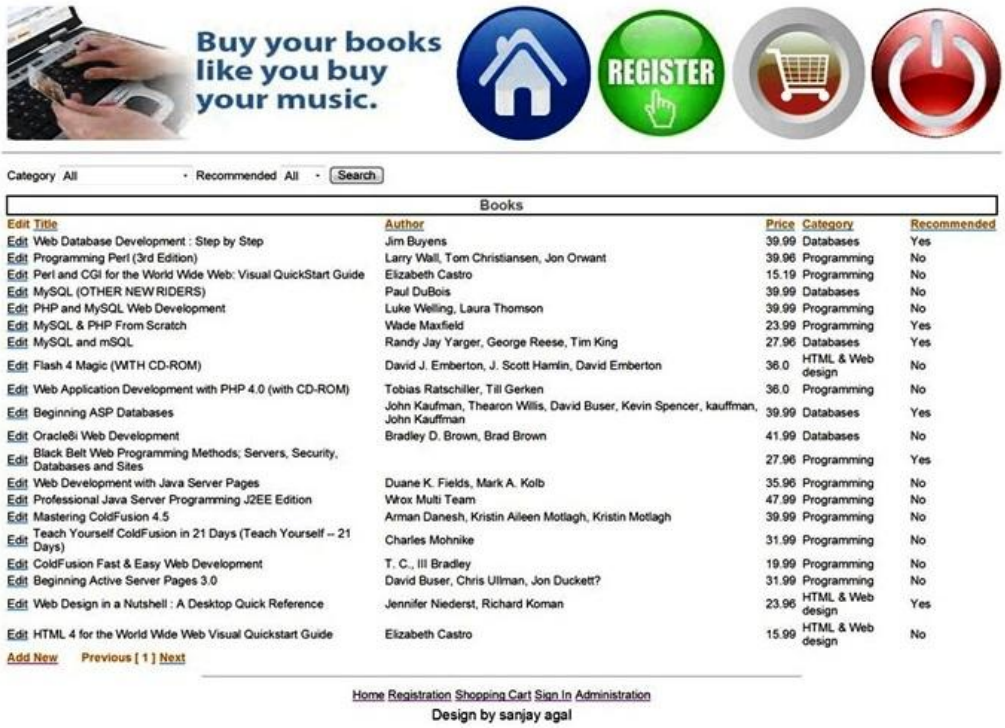


Figure 6.3.14 books

Figure 6.3.15 card types

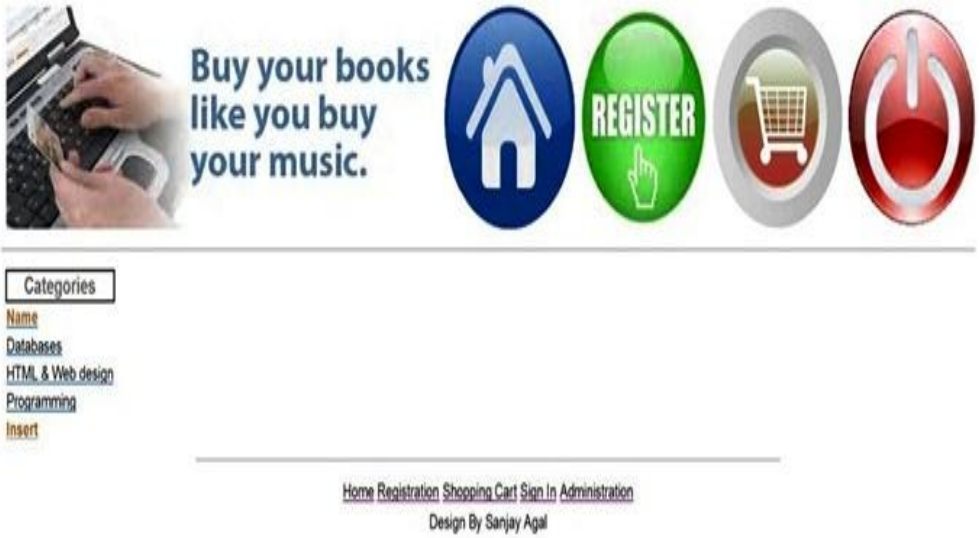


Figure 6.3.16 add categories

Figure 6.3.17 book edit

Figure 6.3.18 editorial

Figure 6.3.19 orders

Figure 6.3.20 edit editorials

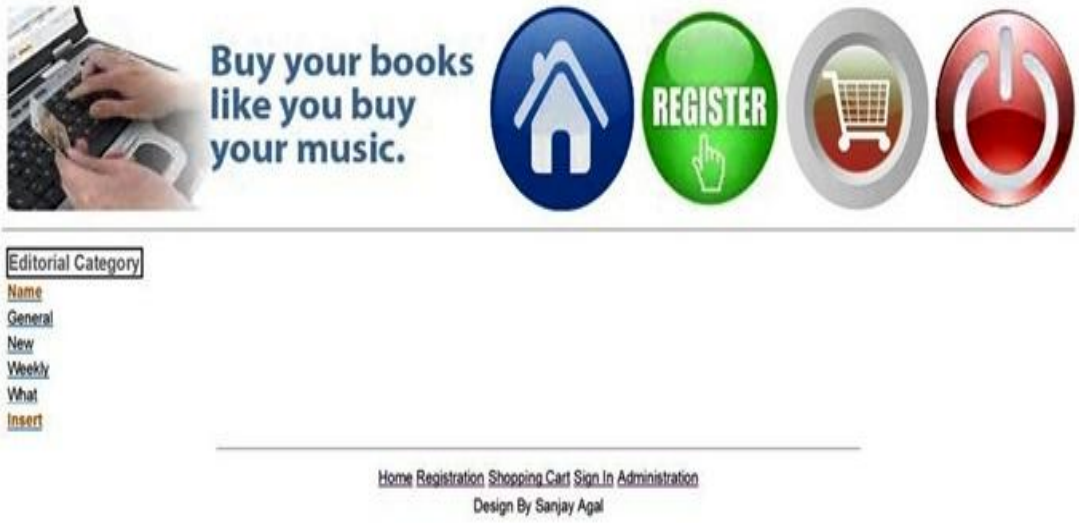


Figure 6.3.21 editorial categories

Figure 6.3.22 total orders

7. Limitations and Future Development

There are some limitations for the current system to which solutions can be provided as a future development:

1. The system is not configured for multi- users at this time. The concept of transaction can be used to achieve this.

2. The Website is not accessible to everyone. It can be deployed on a web server so that everybody who is connected to the Internet can use it.

3. Credit Card validation is not done. Third party proprietary software can be used for validation check.

As for other future developments, the following can be done:

1. The Administrator of the web site can be given more functionality, like looking at a specific customer's profile, the books that have to be reordered, etc.

2. Multiple Shopping carts can be allowed.

8. Conclusion

The Internet has become a major resource in modern business, thus electronic shopping has gained significance not only from the entrepreneur's but also from the customer's point of view. For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible. As per a survey, most consumers of online stores are impulsive and usually make a decision to stay on a site within the first few seconds. "Website design is like a shop interior. If the shop looks poor or like hundreds of other shops the customer is most likely to skip to the other site"[16]. Hence we have designed the project to provide the user with easy navigation, retrieval of data and necessary feedback as much as possible. In this project, the user is provided with an e-commerce web site that can be used to buy books online. To implement this as a web application we used JSP as the Technology. JSP has several advantages such as enhanced performance, scalability, built- in security and simplicity. To build any web application using JSP we need a programming language such as JAVA, HTML # and so on. JAVA was the language used to build this application. For the client browser to connect to the JSP engine we used J2EE 1.4 Application Server as the Web Server. JSP uses sql driver to interact with the database as it provides in-memory caching that eliminates the need to contact the database server frequently and it can easily deploy and maintain an JSP application. MySQL was used as back-end database since it is one of the most popular open source databases, and it provides fast data access, easy installation and simplicity.

A good shopping cart design must be accompanied with user-friendly shopping cart application logic. It should be convenient for the customer to view the contents of their cart and to be able to remove or add items to their cart. The shopping cart application described in this project provides a number of features that are designed to make the customer more comfortable. This project helps in understanding the creation of an interactive web page and the technologies used to implement it. The design of the project which includes Data Model and Process Model illustrates how the database is built with different tables, how the data is accessed and processed from the tables. The building of the project has given me a precise knowledge about how JSP is used to develop a website, how it connects to the database to access the data and

how the data and web pages are modified to provide the user with a shopping cart application.

9. Bibliography

9.1 Articles

1. Dijkstra, E.W. 1982. On the role of scientific thought. In Selected Writings on Computing: A Personal Perspective, 60-66. Springer-Verlag.

2. Craig Walls, Ryan Breidenbach. Spring in Action 2nd Edition. Manning Publishing Co. 2008

3. Laufer, K. A hike through post-EJB J2EE Web application architecture. Computing in Science & Engineering. Volume 7, Issue 5, Sept.-Oct. 2005 Page(s):80 - 88

4. Laufer, K. A hike through post-EJB J2EE Web application architecture. Part II. Computing in Science & Engineering. Volume 8, Issue 2, March-April 2006 Page(s):79 – 87

5. Arthur,J., Azadegan,S. Spring framework for rapid open source J2EE Web application development: a case study. Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on 23-25 May 2005 Page(s):90 – 95

9.2 Books

6. Johnson, R. J2EE development frameworks. Computer Volume 38, Issue 1, Jan. 2005 Page(s):107 – 110

7. Chris Richardson. Untangling enterprise Java. Queue. Volume 4, Issue 5 (June 2006). Component Technologies. Pages:36 – 44. 2006. ISSN: 1542-7730

8. Elizabeth J. O'Neil. Object/relational mapping 2008: hibernate and the entity data model (edm). SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. June 2008

9.3 Websites

9. The Java EE 5 Tutorial.
http://java.sun.com/javaee/5/docs/tutorial/doc/
10. Yu Ping, Kontogiannis, K., Lau, T.C. Transforming legacy Web applications to the MVC architecture. Software Technology and Engineering Practice, 2003. Eleventh Annual International Workshop on 19-21 Sept. 2003 Page(s):133 – 142
11. The Spring Framework Reference Documentation.
http://static.springframework.org/spring/docs/2.5.x/reference/index.html
12. Object-relational impedance mismatch
http://en.wikipedia.org/wiki/Object-Relational_impedance_mismatch
13. Apache Maven
http://maven.apache.org/