1. Introduction

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 list their books through this software. 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 project with many features using a database.

• To increase my knowledge horizon in technologies like JSP, MySQL, CSS, HTML, and JavaScript.

• To gain good experience in JAVA before joining in a full time job.

2. OBJECTIVES

All the Book Shop Keepers can use our project to run their store easily. By using this project they can store all the information Related with the books available in the store. They can also search a particular book easily by getting the information from the project. Because of this computerization, any book store owner can run their store easily in User friendly Environment.

The Main objectives of this project are:

• Reduction of paper work

• Automation of existing manual information systems.

• Reduction of manual processing

• Keep track of daily information exchange at the server by the administrator.

• Increase in processing and transfer speeds of information over the network.

• Decrease in processing time

• Fast retrieval of all type of information

• Good efficiency and response time

• More consistent data handling

• A user-friendly system which do not require any special training or expertise of computer

3. Project Overview

Background

Through a web browser the customers can search for a book by its title or author. The user can login using his account details or new customers can set up an account very quickly. They should give the details. The user can also give feedback to a book by giving ratings. The books are divided into many categories based on subject Like Software, Database, English, Architecture etc.

This project has the following functionalities:

1) Search

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

2) 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 to get the book review.

5) User Review

The user can give Review to a book based on his interest. He can Review it by giving 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 review.

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

8) Administration

The Administrator will be provided with special functionalities like Add or delete a book.

4. SYSTEM REQUIREMENTS

Product Definition

The project is Computerization of Book Store. This project is made for any organization to manage books information in store. The main features of this package are- Information of All the books available in Store.

Processing Environment:

The programming language and development tools are the critical factors in the functioning of the developed system, so proper attention has to be paid while selecting these tools. Following are the minimum requirements for installation and making use of the software package: Computerization of Book Store in JAVA

Hardware requirements:

* Processor: Pentium
* RAM: 1GB
* HARD DISK: 40GB
* Printer: Any compatible printer

Software requirements:

* JDK
* Tomcat-7/8

JAVA being the platform independent language to generate the user-friendly Software system (JSP) is used as Front-end system and MySql for database system, this will facilitate user in operating the system successfully. A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000/XP, Linux, Solaris, and Mac OS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

* The Java Virtual Machine (Java VM)
* The Java Application Programming Interface (Java API)

5. Feasibility Analysis

Feasibility study describes and evaluates candidate systems and provides for the selection of the best candidate system that meets the system performance requirements. Three key considerations are involved in the feasibility analysis:

1. Economic feasibility
2. Technical
3. Behavioral

1. Economic Feasibility

Economic feasibility determines the benefits and saving that are expected from the system and compare them with costs. Cost/Benefit analysis has been done on the basis of total cost of the system and direct and indirect benefits derived from the system. The total cost for the proposed system comprises of hardware costs and software costs. The main aim of economic feasibility is to check whether the system is financially affordable or not. The cost for the proposed system can be divided into two parts given below:

* Hardware Costs

The hardware cost for the proposed system can be calculated from cost of hardware needed for the development of the proposed system. The hardware specifications for the system are given below:

Personal Computer: The cost of the PC depends upon the configuration of the PC. The minimum specification assumed for the Pc is given below:

* + Pentium IV processor
  + 1 GB RAM
  + 40 GB free disk space
  + MS Windows
* Software Costs:

The Software costs for the proposed system can be calculated from the cost of software tools needed from the development of the proposed system. The software tools needed for the development of the system are given below:-

* + JDK
  + Tomcat7/8.

2. Technical Feasibility:

Technical Feasibility centers on the existing system and to what extent it can support the proposed system. In this part of feasibility analysis we determined the technical possibilities for the implementation of the system. Two major benefits are:

* + Improving the performance
  + Minimizing the cost of processing

The performance category emphasis improvements in the accuracy and access to the information. It also makes easier to access the system by the authorized users.

3. Behavioral Feasibility:

Computerization of Book Store in JAVA Behavioral feasibility estimates the reaction of the User staff towards the development of the computerized system. For the successful implementation of any system, the users must be impressed that the new system is for his benefit. So, the behavioral feasibility plays a very important role in the development of new system It reveals that whether the system is acceptable by user or not. If the user does not ready to use it, then it doesn’t matter how best the system is or how much effort you are putting in its development.

6. Implementation Technologies

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.

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.

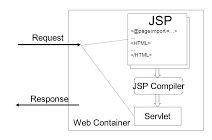


FIGURE 3.2.1 SERVLET CONTAINER

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. Pre-compilation 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:

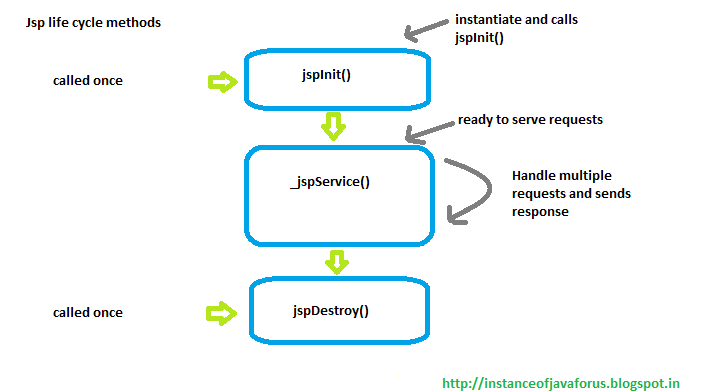


FIGURE 3.2.2 “JSP SERVLET Lifecycle”

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:

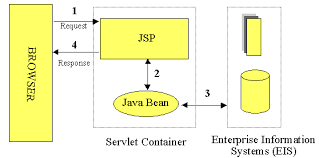


FIGURE 3.2.1.1 “JSP SERVLET”

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 are 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.

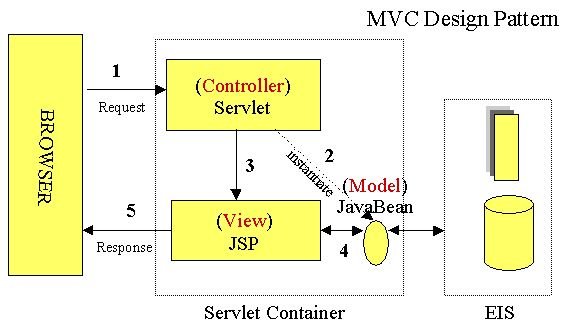


FIGURE 3.2.1.2 MVC DESIGN PATTERN

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. 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.

7. 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.

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.

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.

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).

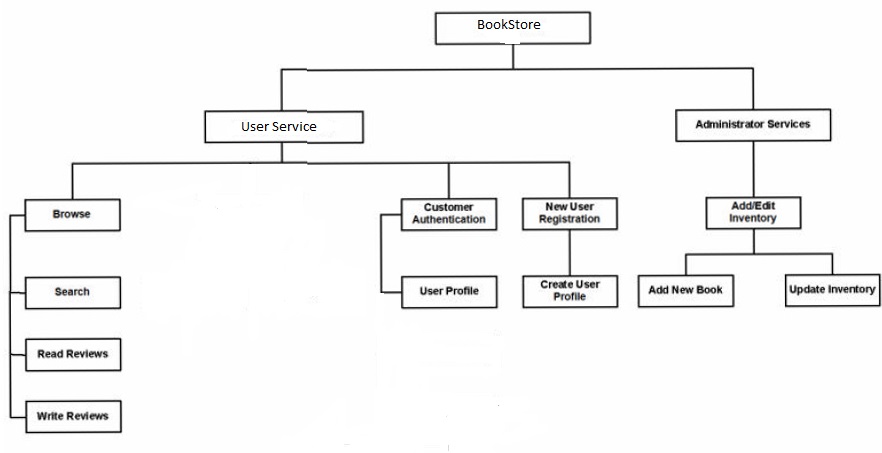


Figure 2 Functional Decomposition Diagram

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 & give output.
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.

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.

User-Browse Context DFD:

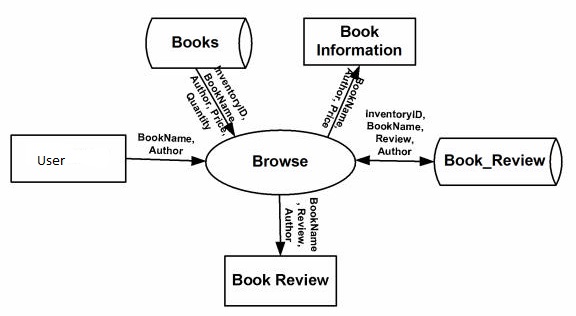


Figure: User-Browse Context DFD

User-Browse Detailed DFD:

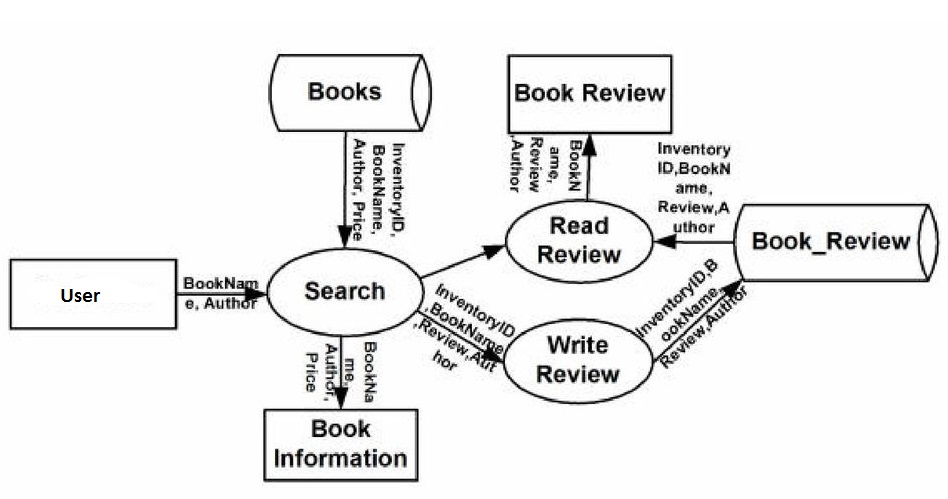


Figure: User-Browse Detailed DFD

User-Authentication Context DFD:

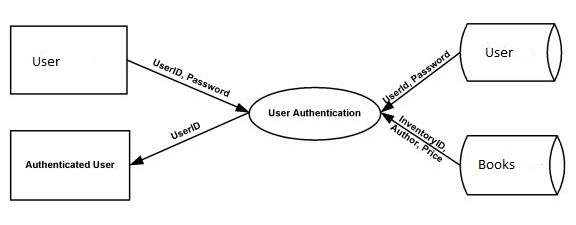


Figure: User-Authentication Context DFD

User-New User Registration DFD

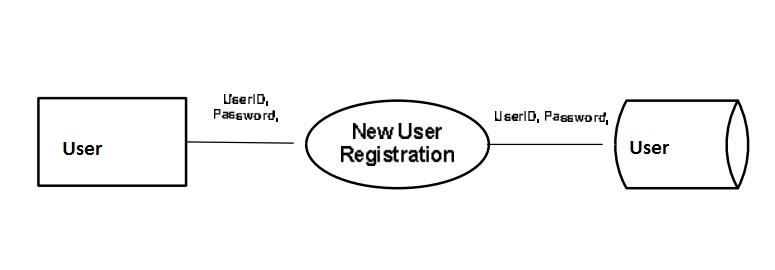
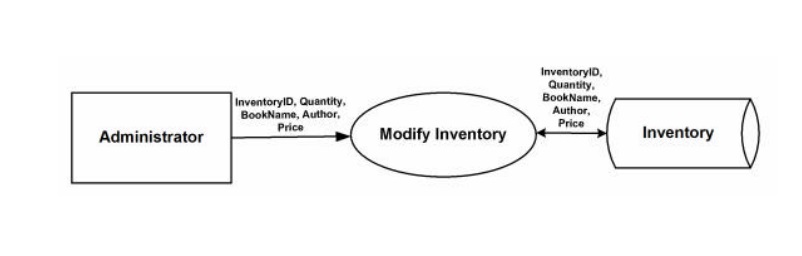


Figure: User-New User Registration DFD

Administrator Update Inventory Context-DFD:

Figure: Administrator Update Inventory DFD



Administrator Update Inventory Detailed-DFD:

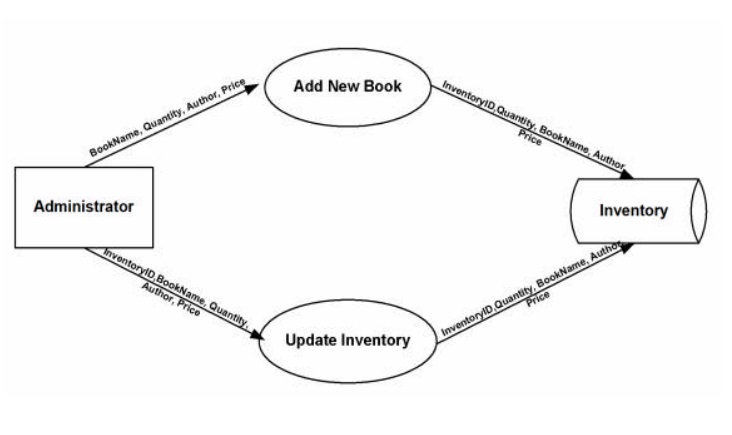


Figure: Administrator Update Inventory Detailed-DFD

8. 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, add review on books and list the books. The user interface design will closely follow our Functional Decomposition Diagram. Below snapshots show the initial designs of the web pages.

Snapshots………………………