

# Low Level Document

## New And Old Book Shop

### 1. Servlets

#### i. BooksServlet:

Handles requests related to books, such as fetching all books, buying a book, and uploading a book.

##### **doGet Method:**

Endpoint `"/book/fetch"`: Retrieves all books from the database and forwards the request to `displayBooks.jsp`.

Endpoint `"/book/buy"`: Forwards the request to `buyBook.jsp` for purchasing a book.

##### **doPost Method:**

Endpoint `"/book/upload"`: Processes book upload form data, adds the book to the database, and redirects to `"/transaction/addSellTransaction"`.

Endpoint `"/book/buyBook"`: Processes book purchase request, adds the transaction to the database, and redirects to `"/ordered.jsp"`.

Endpoint `"/book/search "`: Retrieves all books which matches our search from the database and forwards the request to `"/searchBooks.jsp"`.

#### ii. CustomerServlet:

Manages customer-related operations such as registration, login, and profile viewing.

##### **doGet Method:**

Endpoint `"/customer/profile"`: Retrieves customer profile information and forwards the request to `getTransactionsHistory`.

##### **doPost Method:**

Endpoint `"/customer/registration"`: Processes customer registration form data and adds the customer to the database.

Endpoint `"/customer/login"`: Validates customer login credentials and sets session attributes.

### iii. TransactionServlet:

Handles transaction-related operations like adding a transaction and retrieving transaction history.

#### **doGet Method:**

Endpoint `"/transaction/addTransaction"`: Adds a buy transaction to the database.

Endpoint `"/transaction/addSellTransaction"`: Adds a sell transaction to the database.

Endpoint `"/transaction/getTransactionsHistory"`: Retrieves transaction history for a customer.

## 2. DAO Implementations:

### i. BooksDaoImpl:

Implements BooksDao interface for CRUD operations on books.

**addBook Method:** Inserts a new book into the database.

**getAllBooks Method:** Retrieves all books from the database.

**deleteBook Method:** Deletes a book from the database by ID.

### ii. CustomerDaoImpl:

Implements CustomerDao interface for customer-related database operations.

**addCustomer Method:** Inserts a new customer into the database.

**verifyCustomer Method:** Validates customer login credentials.

**getCustomerId Method:** Retrieves the customer ID by email.

**searchCustomer Method:** Retrieves customer details by ID.

### iii. TransactionDaoImpl:

Implements TransactionDao interface for transaction-related database operations.

**addTransaction Method:** Adds a new transaction to the database.

**getTransactionsById Method:** Retrieves transaction history by customer ID.

## 3. JSP Pages:

### **displayBooks.jsp:**

Displays a list of books fetched from the database.

Includes a form for buying each book.

**ordered.jsp:**

Confirmation page displayed after a successful book purchase.

**profileData.jsp:**

Displays customer profile details and transaction history.

**uploadBook.jsp:**

Form for sellers to upload books for sale.

**uploaded.jsp:**

Confirmation page displayed after a book is successfully uploaded for sale.

**index.jsp:**

Landing page displaying options to register or login.

**login.jsp:**

Form for customers to log in with their credentials.

**Main.jsp:**

User dashboard with options to view profile, buy a book, and sell a book.

**Registration.jsp:**

Form for new customers to register with the system.

**uploadBookFailed.jsp:**

Error page that displays when uploading of book gets failed.

**transactionFailed.jsp:**

Error page that displays when transaction gets failed.

#### 4. Database Schema:

**Tables:**

books (id, name, author, price, condition)

customers (id, name, email, password, address)

transactions (transaction\_id, customer\_id, book\_name, transaction\_type).

## 5. Technologies Used:

Java EE (Servlets, JSP)

JDBC for database connectivity

MySQL for database management

## 6. Error Handling:

Use try-catch blocks for handling database operations and exceptions.

Display appropriate error messages or redirect to error pages for user feedback.

## 7. Session Management:

Use HttpSession to manage user sessions and store session attributes like user ID.

## 8. Deployment:

Deploy the application on a web server like Apache Tomcat.

Configure database connections in a properties file for easy maintenance.

## 9. Future Enhancements:

Implement user roles and access control for admin, sellers, and buyers.

Enhance UI with AJAX for dynamic content loading.

Integrate payment gateways for online transactions.

Implement caching mechanisms for improved performance.