# **Requirements and Design Document**

# **TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **Content** | **Page No** |
| Application description | 2 |
| Application Design | 2 |
| Web Page Design (View) | 2 |
| Business Logic (Control) | 3 |
| Business Layer Design (Model) | 3 |
| Site Map | 4 |
| Details for each page  Home Page  Login page  Register User  Forgot password  Product List  Cart Page  Update product Add ProductDelete ProductAssign role to user | 5  5  5  5  6  6  6  7  7  7 |
| ER DIAGRAM | 8 |
| TABLES User tableTempuser tableResetpass tableProductlist table | 9  9  9  9 |

## **Application description:**

The ‘BOOK WORLD’ is an application that allows users to purchase books online. This application allows the admins to manage the data of the website with the help of various operations and the customers to buy the books of their choice from the wide variety available. This application is developed for people who love to read and would like to buy books online.

# **Application Design:**

## MVC Model

This architectural pattern used to build this application is the Model View Controller pattern. It has Java bean classes, JSP pages and Servlet pages.

## JSP Pages:

JSP pages are used to represent view in the browsers. Each JSP page corresponds to a view in the browser. It renders the data from servlet class and displays it and also passes the data entered in the browser to the servlet.

## Servlet Pages:

Servlet pages control data from and to the browser. It passes data between JSP pages and Java bean classes.

## Java bean:

It implements the business layer of the application. It gets data from the database as well as sends data to the database. The database logic is written here. Connection to the database happens in Java bean class.

1. **Web Page Design (View)**

The basic coding of the web pages is done in HTML 5. To make the web pages of this application responsive Bootstrap has been used. All the CSS is kept in a separate file. The user is allowed to choose the number of pages he wants to see. A responsive search option has been given to better searching. It also consists of a reset password option.

# **Business Logic (Control)**

The controller is helps in controlling data flow between JSP pages and bean classes. This is done by sending and receiving parameters using the GET/POST http request. The GET methods help to pass data from controller to JSP when a data wants to see any data on the web page and when a user enters any data into a web page the POST method gets hit and the data from the JSP pages flows to the controller. In this way controller populate JSP with dynamic data. Controller also helps to navigate between different pages. On the click of a button the desired parameters are passed which are then used further for some logic or passed to some other pages by setting the attribute for a request.

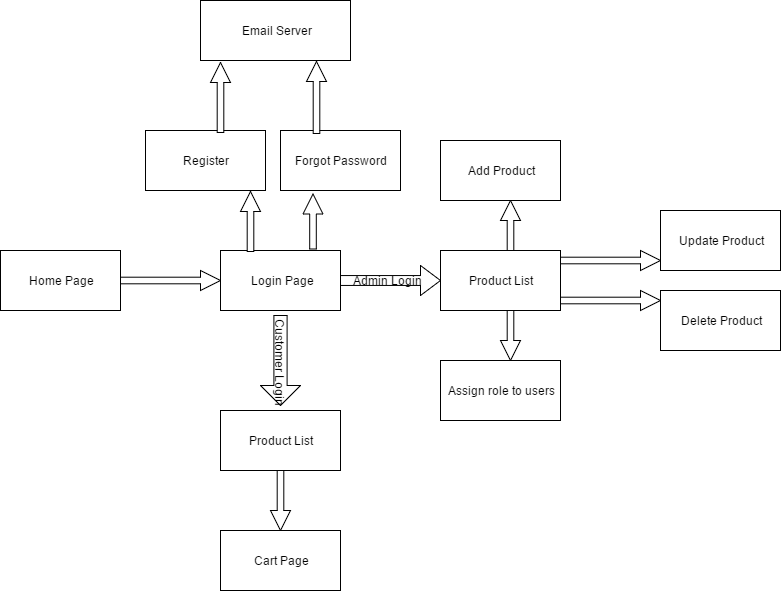
To send a password reset email Murach’s java Servlets and JSP’ email class has been used. This class takes in parameters like the email id of the respective user and then sends a link on that email id. On clicking this link, the user can set a new password. The access of various web pages according to the user role(customer/admin) is also carried out using controller. If an admin logs in, a variable is set in the controller which identifies the current user session as an admin user session. Further this variable is passed to each JSP page based on which a decision is taken as to which pages and functionality of web pages should be shown. The admin is permitted to use ADD, UPDATE and DELETE operations on the various books available on the database. If the user logs in as a customer a respective session variable is set indicating the same. Controller is responsible to tell each JSP page that a customer is currently logged in and then the user is allowed to use the operations and functionalities meant for a customer such as ADD TO CART.

# **Business Layer Design (Model)**

Java beans classes contain private variables and getter and setter methods which are used to set and get values from these private variables.

To store data, MySQL has been used, bean class sends a query to fetch or alter data. The tables in the database are normalized. To interact with the database prepared statements have been used. Bean classes send and receive data from servlets. In order to secure passwords, hashing and salting has been used. The code for hashing and salting has been used from ‘Murach’s java Servlets and JSP’. By using hashing and salting, an encrypted password is set in the database instead of the original password thus protecting it.

1. **Site Map**



1. **Details for each page**

### **7.1 Home Page**

Purpose: To display home screen, with header and footer

Audience: Customer and Admin

Data fields: Welcome ‘username’ – only if a user is logged in.

Buttons and hyperlinks: Login button - Opens login page on click

Product List button - Opens product list page on click (Visible if a user is logged in)

### **7.2 Login page**

Purpose: To display login screen that allows a user to login

Audience: Customer and Admin

Data fields: Username field and password field. Both are input fields.

Validations: User Id and password must be correct to login successfully.

Buttons and hyperlinks: Login button - Opens login page on click

Forgot password- Opens forgot password screen on click

Register – Opens new user page on click

### **7.3 Register User**

Purpose: To show new user registration screen, where a new user can register.

Audience: Customer and Admin

Data fields: Username, password, confirm password and email Id field.

Validations: password, confirm password must match, User ID must be unique.

Buttons and hyperlinks: Create Account button - Email with link is sent on click

### **7.4 Forgot password**

Purpose: To display screen where a user can enter his/her email to reset password

Audience: Customer and Admin

Data fields: email Id field.

Validations: User ID must be an email and must be present in the system.

Buttons and hyperlinks: Reset Password button - Email with link is sent on click

### **7.5 Product List**

Purpose: To display screen where details of all the products are listed.

Audience: Customer and Admin.

Data fields for Admin: Product image, name, status, price, search and pagination.

Data fields for customer: Product image, name, status, price, search and pagination.

Validations of customer: If an item is out of stock, the ‘ADD TO CART’ button will be disabled

Buttons and hyperlinks for customer: A new page cart opens on click of ‘ADD TO CART’ button

Buttons and hyperlinks for Admin: ‘UPDATE’ button - a new page named update product opens on click.

‘DELETE’ - a prompt comes up for confirmation of deletion of that product on click of this button.

‘ADD PRODUCT’ - add product page opens on click.

‘ASSIGN ROLE TO USER’ – a page where roles can be assigned to each user of the system opens on click of this button

### **7.6 Cart Page**

Purpose: To display a screen where a customer can see the items added in their cart and can add or remove those as desired.

Audience: Customer

Data fields: Product name, price, quantity, amount.

Validations: Quantity cannot be set to zero. Amount is the total price calculated based on quantity.

Buttons and hyperlinks: REMOVE ITEM button - Removes the item in cart on click

CONTINUE SHOPPING button - takes the customer back to the product list page

CHECKOUT button - Takes user to the checkout page

### **7.7 Update product**

Purpose: To display screen where an admin can update a selected product

Audience: Admin

Data fields: Product image, name, price, availability

Validations: Product name and price cannot be blank

Buttons and hyperlinks: SUBMIT button – takes user back to the product list page with the new updated details being displayed now

GO BACK button - takes user back to the product list page on click

### 

### **7.8 Add Product**

Purpose: To display screen where an admin can add a new product

Audience: Admin

Data fields: Product name, price, availability

Validations: Product name and price cannot be blank.

Buttons and hyperlinks: ADD button - a new product gets added to the database and the user is taken to the product list page with the new product now being displayed

GO BACK button - takes the user back to the product list page on click

### **7.9 Delete Product**

Purpose: To display the screen where an admin can delete a product.

Audience: Admin

Data fields: a popup appears asking for a confirmation for the deletion of product

Buttons and hyperlinks: Yes button - the product is deleted on click and the user is taken to the product list page

No button - user is taken to the product list page

### **7.10 Assign role to user**

Purpose: To display screen where an admin can assign distinct roles to users

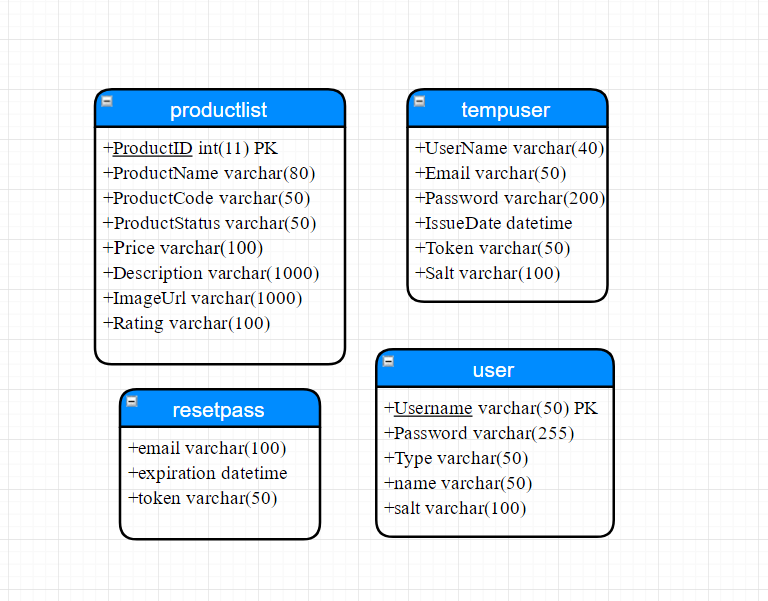
Audience: Admin

Data fields: User name, role

Buttons and hyperlinks: CHANGE THE ROLE button - role of the user is changed based on the selected value from the dropdown list

GO BACK button - ‘product list’ page appears on click

1. **ER DIAGRAM**



## 

1. **TABLES**

## **user table**

User table stores data of all the registered users

Username is the primary key

Column ‘Type’ stores the user role as ‘Admin’ or ‘Customer’

Password column saves encrypted password.

## **tempuser table**

tempuser saves temporary user data. When a user fills up the registration form, their details are saved in this table

When user clicks on the link from the email to activate his/her account, the details of that user get deleted from tempuser table.

## **resetpass table**

When a user requests for reset of his password, data is entered in this table

The ‘email’ column stores the email id of the user

## **productlist table**

ProductID is the primary key of this table

This table stores all the products which are displayed to the users

‘ProductStatus’ column saves the availability of that each product.