

Darshan University

A Project Report on

**“Supermarket Management System”**

Under the subject

**Software Engineering (2301CS405)**

B. Tech, Semester – IV

Computer Science & Engineering Department

Submitted By

Student Name: Jasmin Chauhan Enrollment No.: 23010101045

Academic Year (2024-2025)

|  |  |
| --- | --- |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Computer Science & Engineering Department Darshan University**

**DECLARATION**

We hereby declare that the SRS, submitted along with the **Software Engineering** **(2301CS405)** for entitled **“Supermarket Management System”** submitted in partial fulfilment for the Semester-4 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Departmentto Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of R. B. Gondaliya and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

Jasmin Chauhan

Student’s Signature

Date: \_\_\_\_\_\_\_\_\_\_

**Computer Science & Engineering Department Darshan University**

**CERTIFICATE**

This is to certify that the SRS on **“Supermarket Management System” has** been satisfactorily prepared by **Jasmin Chauhan (23010101045)** under my guidance in the fulfillment of the course **Software Engineering (2301CS405)** work during the academic year 2024-2025.

Internal Guide Dean-DIET

Prof. R. B. Gondaliya Dr. Gopi Sanghani

Darshan University Darshan University

**ACKNOWLEDGEMENT**

I wish to express my sincere gratitude to my project guide Prof. R. B. Gondaliyaand all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me the permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

**Jasmin Chauhan**

**ABSTRACT**

The Supermarket Management System is a system which aims in developing a computerized system to maintain all daily work of supermarket. This system will act as a tool to transfer traditional supermarket to digital supermarket. This project has many features which are generally not available in general market system like facility of user login, products search on one click etc. It also has a facility of admin login through which the admin can monitor the whole system. This system will be designed with the basic features such as manager can add/view/update/delete product details and in it. The Customer can add their products into cart and can order it. Manager has facility in software that he can track the records of inventory and restock it. Cashier can generate bill easily by clicking one button.

Main purpose of this system is to reduce human efforts as much as possible.

**Table of Contents**

List of Figures ....................................................................................................................................................... I

List of Tables ....................................................................................................................................................... II

[1 Introduction 1](#_Toc188520819)

[1.1 Product perspective 1](#_Toc188520820)

[1.2 Product features 1](#_Toc188520821)

[1.2.1 There are four different users who will be using this product: 1](#_Toc188520822)

[1.2.2 The features that are required for the Customer are: 1](#_Toc188520823)

[1.2.3 The features that are required for the Manager are: 1](#_Toc188520824)

[1.2.4 The features that are required for the admin are: 1](#_Toc188520825)

[1.3 Functional Requirement 1](#_Toc188520826)

[1.3.1 Customer 1](#_Toc188520827)

[1.3.2 Manager 2](#_Toc188520828)

[1.3.3 Cashier 2](#_Toc188520829)

[1.3.4 Admin 2](#_Toc188520830)

[1.4 Non-Functional Requirement 2](#_Toc188520831)

[1.4.1 Usability: 2](#_Toc188520832)

[1.4.2 Accuracy: 2](#_Toc188520833)

[1.4.3 Availability: 2](#_Toc188520834)

[1.4.4 Maintainability: 2](#_Toc188520835)

[2 Design and Implementation Constraints 2](#_Toc188520836)

[2.1 Use case diagram 2](#_Toc188520837)

[2.2 Activity diagram and Swimlane diagram 4](#_Toc188520838)

[2.3 Sequence diagram 7](#_Toc188520839)

[2.4 State diagram 8](#_Toc188520840)

[2.5 Class diagram 9](#_Toc188520841)

[2.6 Data flow diagram 10](#_Toc188520842)

[2.6.1 Context diagram (level-0) 10](#_Toc188520843)

[2.6.2 DFD Level-1 10](#_Toc188520844)

[2.6.3 DFD Level-2 11](#_Toc188520845)

[3 External interface requirement (Screens) 12](#_Toc188520846)

[3.1 Screen-1: Registration Form 12](#_Toc188520847)

[3.2 Screen-2: Login Form 13](#_Toc188520848)

[3.3 Screen-3: Add borrower book 14](#_Toc188520849)

[4 Database design 15](#_Toc188520850)

[4.1 List of Tables 15](#_Toc188520851)

[5 Stories and Scenario 16](#_Toc188520852)

[5.1 Story-1: Add New Book in Library Catalogue 16](#_Toc188520853)

[5.1.1 Scenario# S1.1 16](#_Toc188520854)

[5.1.2 Scenario# S1.2 16](#_Toc188520855)

[5.1.3 Scenario# S1.3 16](#_Toc188520856)

[5.2 Story-2: Search Book 17](#_Toc188520857)

[5.3 Story-3: Manage due date for borrowed book 17](#_Toc188520858)

[5.4 Story-4: Renew book 17](#_Toc188520859)

[5.5 Story-5: Generate a report on book usage and availability 18](#_Toc188520860)

[6 Test cases 18](#_Toc188520861)

[7 References 22](#_Toc188520862)

**List of Figures**

Figure 2.1-1 Use case diagram for supermarket management system ............................................................. 3

Figure 2.2-1 Activity diagram for purchasing product ....................................................................................... 4

Figure 2.2-2 Swimlane diagram for purchasing product .................................................................................... 4

Figure 2.2-3 Activity Diagram for Adding Product ............................................................................................. 5

Figure 2.2-4 Swimlane Diagram for Adding Product .......................................................................................... 6

Figure 2.3-1 Sequence diagram for Book Issue .................................................................................................. 7

Figure 2.4-1 State diagram of Book .................................................................................................................... 8

Figure 2.4-2 State diagram for Librarian ........................................................................................................... 8

Figure 2.5-1 Class diagram for Library management system ............................................................................. 9

Figure 2.6-1 Context diagram for Library management system ...................................................................... 10

Figure 2.6-2 DFD level-1 for Library management system ............................................................................... 10

Figure 2.6-3 DFD level-2 for Issue book ........................................................................................................... 11

Figure 3.1-1 Screen-1: Registration Form ........................................................................................................ 12

Figure 3.2-1 Screen-2: Login Form ................................................................................................................... 13

Figure 3.3-1 Screen-3: Add borrower book ...................................................................................................... 14

**I**

**List of Tables**

Table 3.1-1 Screen element of Registration form ............................................................................................ 12

Table 3.2-1 Screen element of Login form ....................................................................................................... 13

Table 3.3-1 Screen element of Add borrower book ......................................................................................... 14

Table 4.1-1 Table: Book .................................................................................................................................... 15

Table 4.1-2 Table: Borrower ............................................................................................................................. 15

Table 4.1-3 Table: Staff ..................................................................................................................................... 15

Table 4.1-4 Table: Student ............................................................................................................................... 15

**II**

# Introduction

## Product perspective

This project is basically updating the manual supermarket system into an internet-based application so that the users can order products from home, give feedback in application so that we can improve user experience in it, etc. It is a multi-user version and can take care of all the fundamental functions of a Supermarket like Purchasing, viewing product details, Get heavy discounts and offers, etc. It can satisfactorily cater to all the basic functions of a small supermarket.

## Product features

### There are four different users who will be using this product:

* Customer who will be using this product for shopping and all these things.
* Manager who will be restocking inventory and assigning roles to staff members.
* Cashier who will scan products and generate bills.
* Admin who will keep backup of whole system and set taxes in products.

### The features that are required for the Customer are:

* Search Products they want.
* Add Products to cart if they want to buy multiple things.
* See Offers if they want to buy something.
* Give Feedback if either they like it user interface or not as well as they can share their experience for it.
* Create their profile
* Delete their profile if they want.

### The features that are required for the Manager are:

* Can add the products in market.
* Can update the product details in case of price change.
* Can remove products.
* Can set offers in products.
* Can restock inventory.

### The features that are required for the admin are:

* Can Calculate the tax.
* Backup the data.
* Can change the role of staff members.
* Can Hire and remove the staff.

## Functional Requirement

### Customer

* Sign In: Let Customer create new account
* Search Product: Customers can find products by searching it.
* Add to Cart: It allows customer to add items in cart and buy it all items together.
* See Offers: It shows offers ongoing in market like discounts and all.
* View Profile: Customer can view their profile.
* Remove Profile: Customer can delete their profile. • Update Profile: Customer can update their profile.
* Give Feedback: Customer can give feedback of their experience.
* Payment Methods: Customer can choose methods of payment like cash or credit / debit card.

### Manager

* Add Products: Manager can add Details about new products like their name, price, rating, etc.
* Update Product Details: Manager can update details of products.
* Remove Products: Manager can remove products that is not available in supermarket now.
* Restock Inventory: He keeps track of stock and restock inventory of market.
* Set Offers: He sets offers and discounts on products.
* Assign Roles: He assigns roles like cashier, staff members.

### Cashier

* Scan Products: Use barcode scanner to add products details in a bill.
* Generate Bills: Create bills and handle payments in cash or card.
* Cancel Transactions: He cancels transactions at any time of payment failures.
* Process Returns: He can handle refunds or exchanges easily.

### Admin

* View Whole System: He can view whole details about cashiers and managers and all staff members.
* Set Tax: He sets taxes on products.
* Backup Data: He makes sure that all important data will be backed up in case of software problem.
* Update Roles: He can change roles of manager or cashiers or any staff members.
* Add Roles: He can add more cashiers and managers.
* Remove Roles: He can remove roles of any staff members.

## Non-Functional Requirement

### Usability:

• The UI should be simple enough for everyone to understand and get the relevant information without any special training. Different languages can be provided based on the requirements.

### Accuracy:

• The data stored about the books and the fines calculated should be correct, consistent, and reliable.

### Availability:

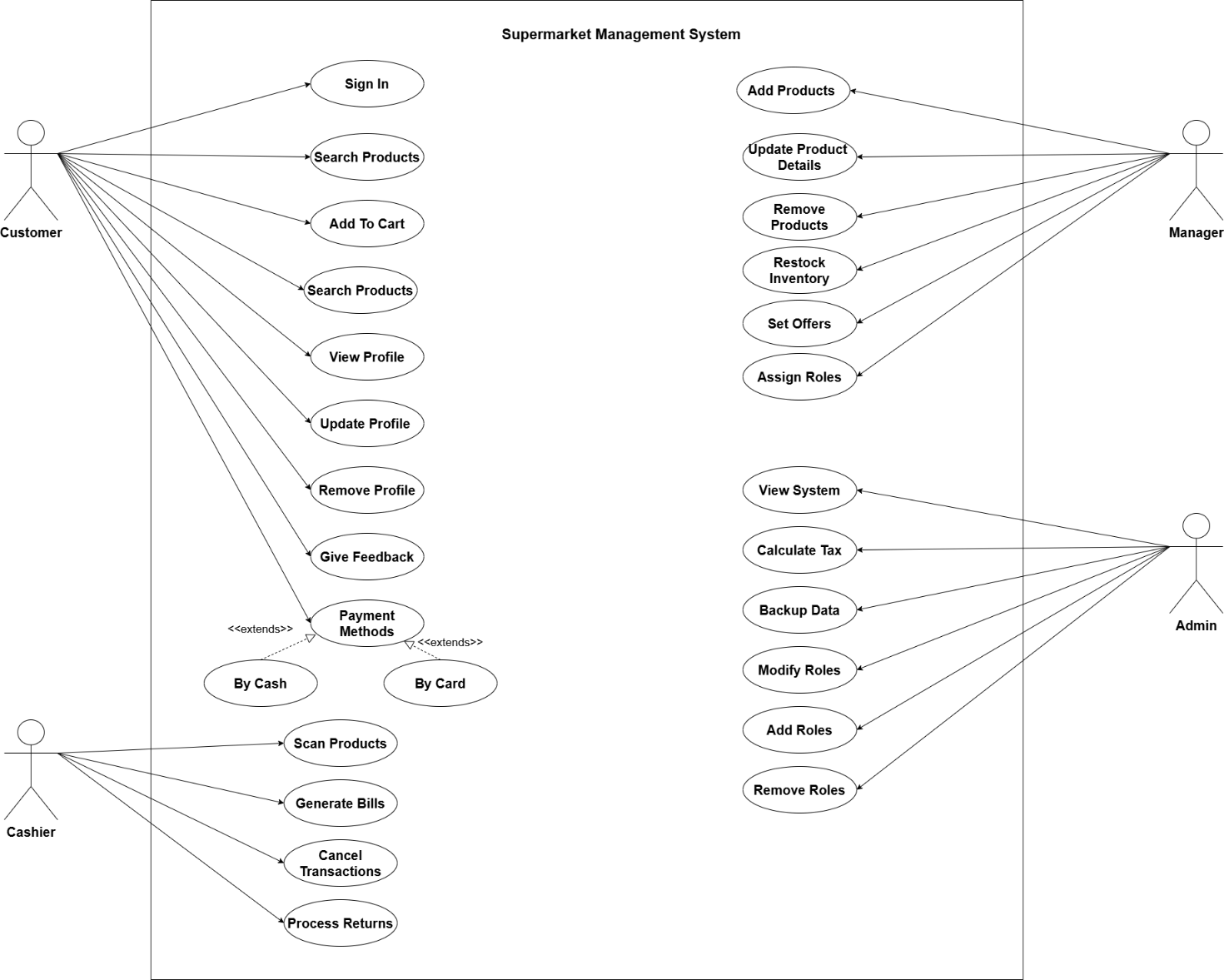
• The System should be available for the duration when the library operates and must be recovered within an hour or less if it fails. The system should respond to the requests within two seconds or less.

### Maintainability:

• The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible. In addition to this, the software must also be portable.

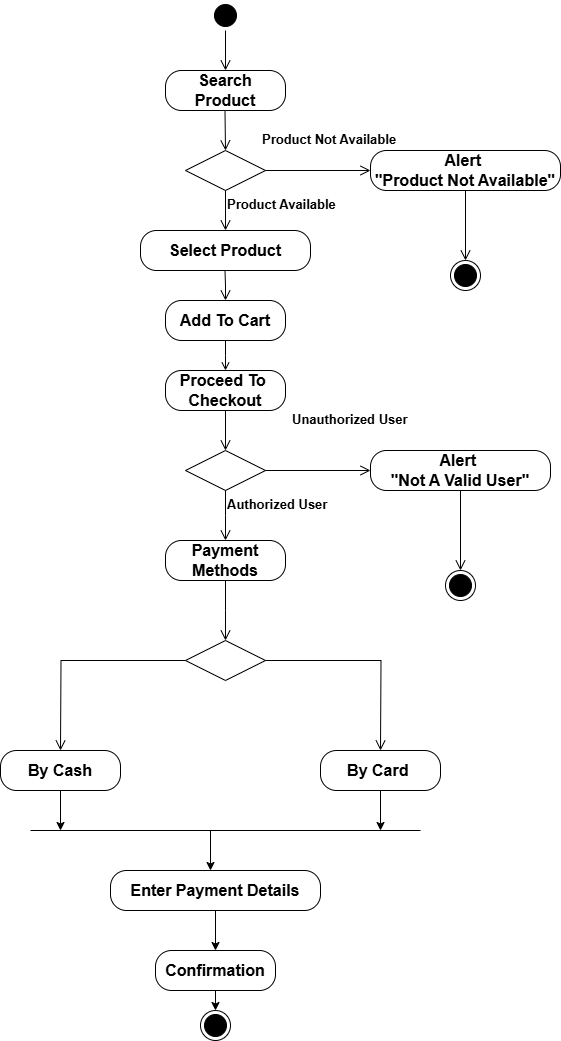
# Design and Implementation Constraints

## Use case diagram

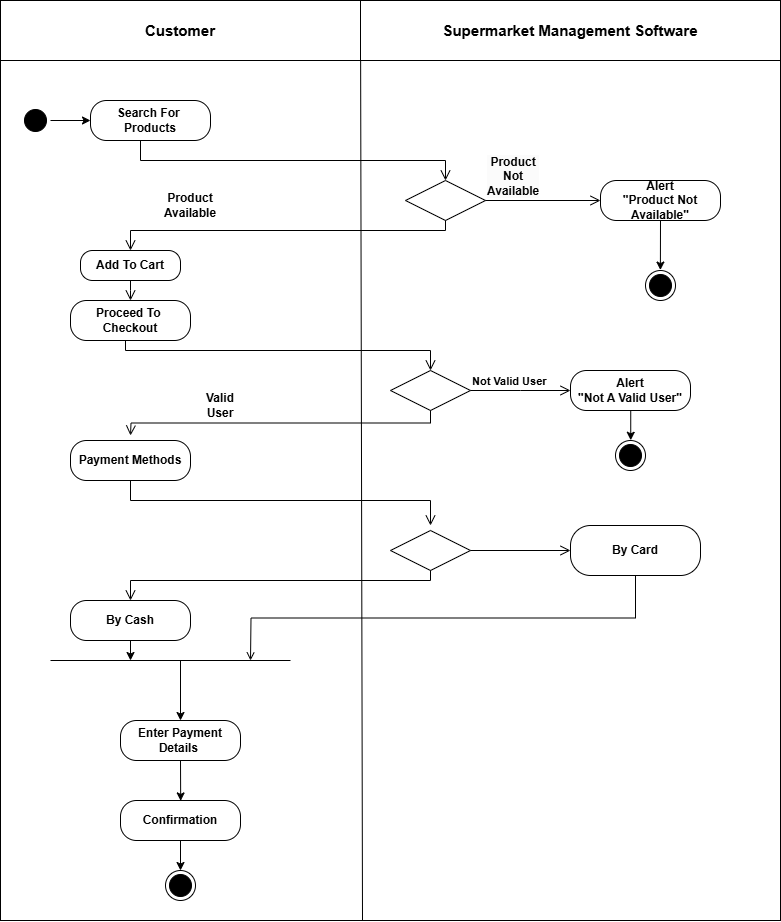


*Figure 2.1-1 Use case diagram for supermarket management system*

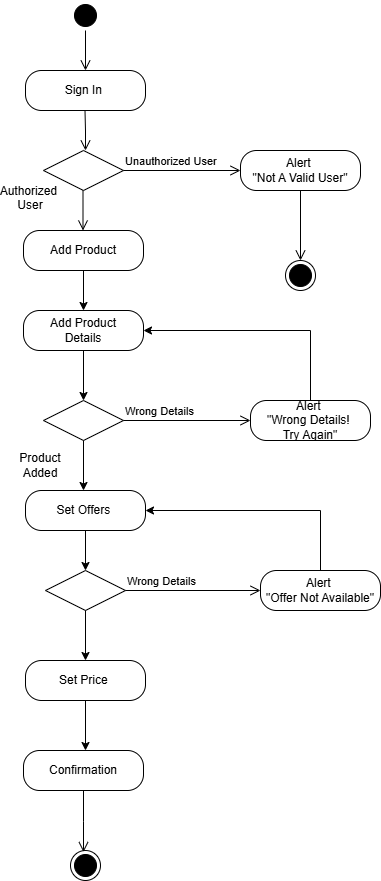
## Activity diagram and Swimlane diagram



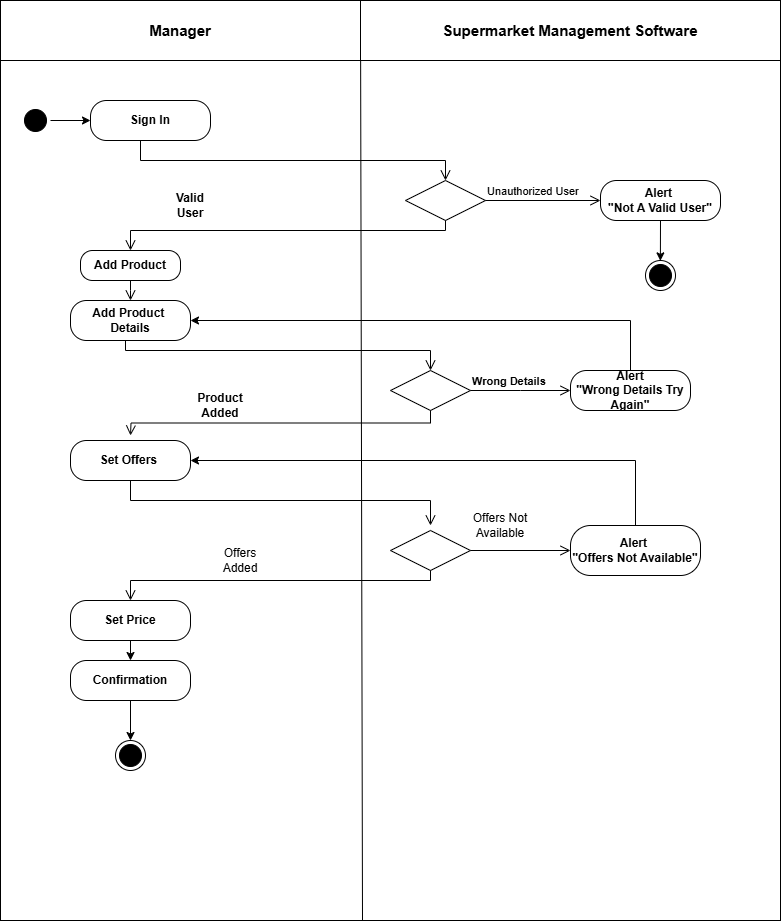
*Figure 2.2-1 Activity diagram for purchasing product*



*Figure 2.2-2 Swimlane diagram for purchasing product*

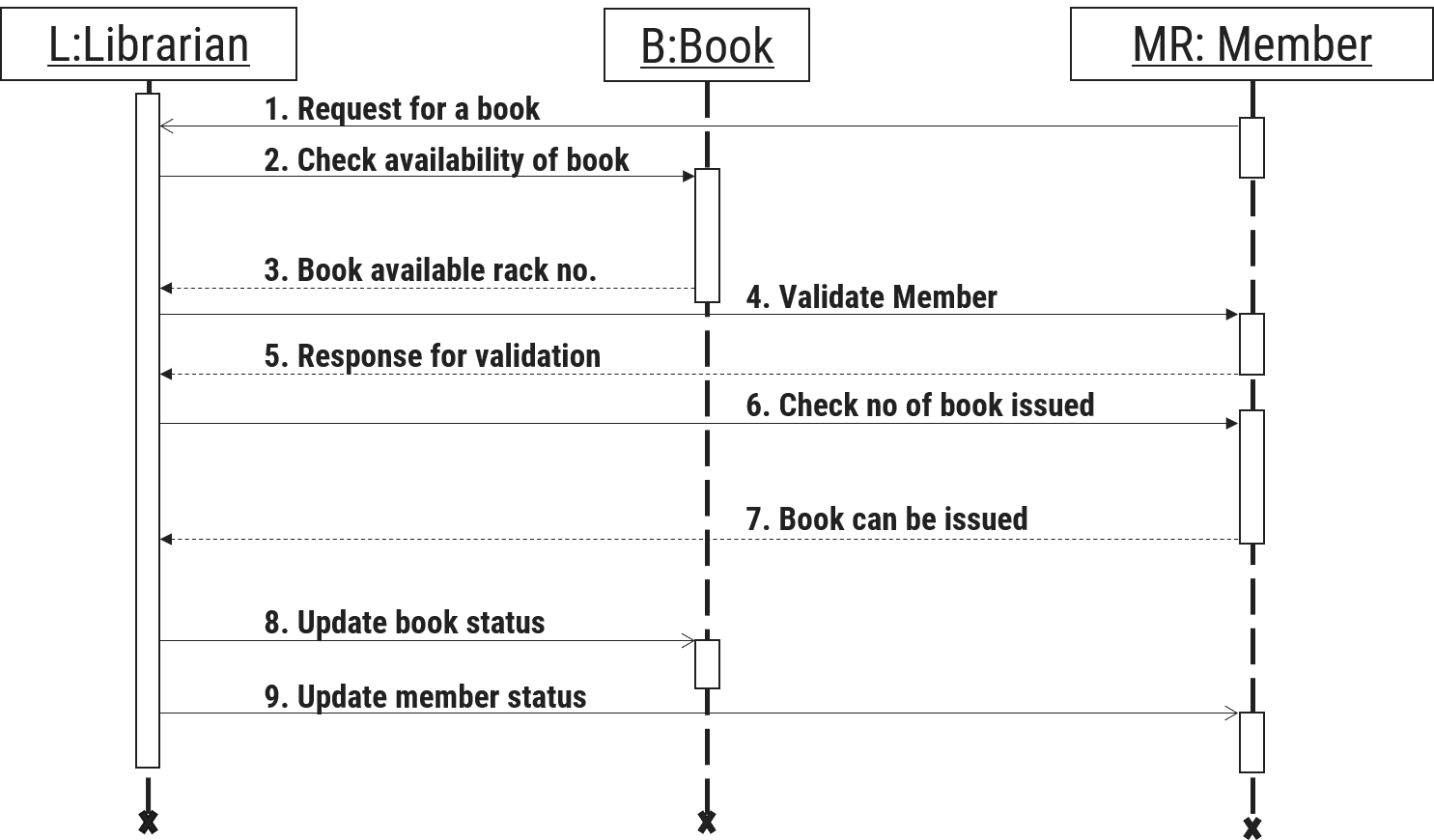


*Figure 2.2-3 Activity Diagram for Adding Product*



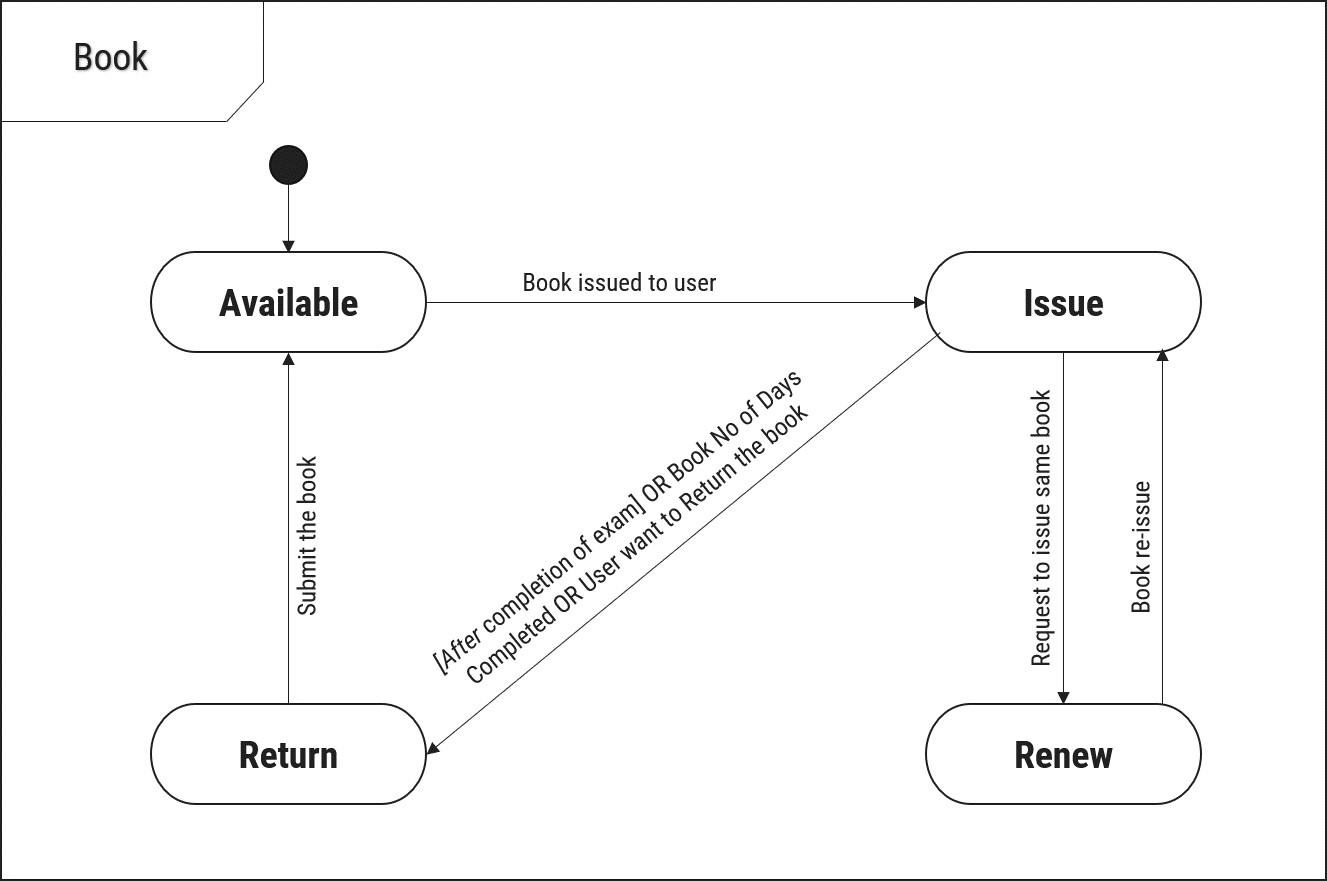
*Figure 2.2-4 Swimlane Diagram for Adding Product*

## Sequence diagram

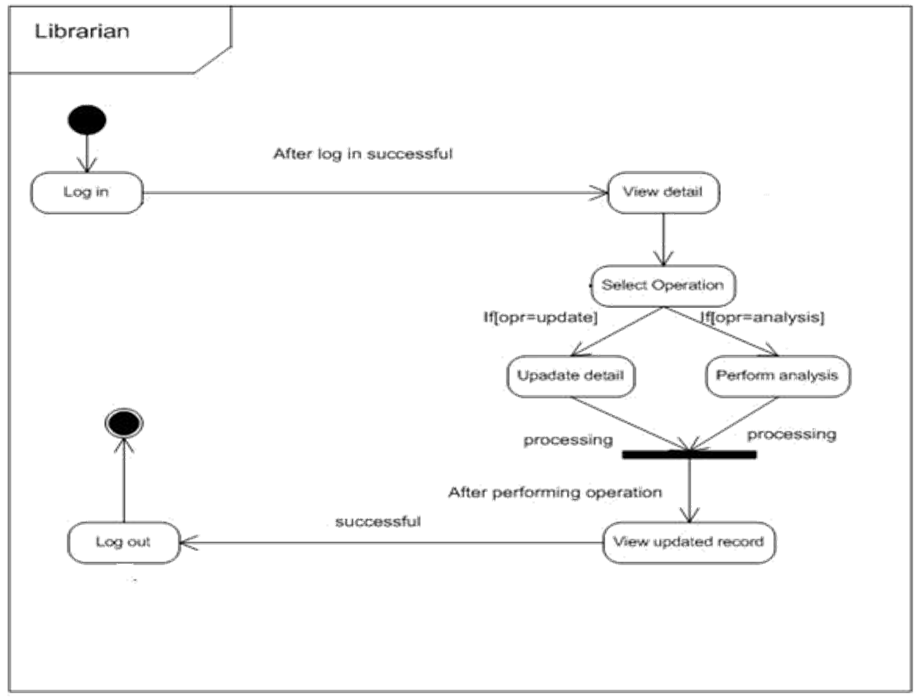


*Figure 2.3-1 Sequence diagram for Book Issue*

## State diagram

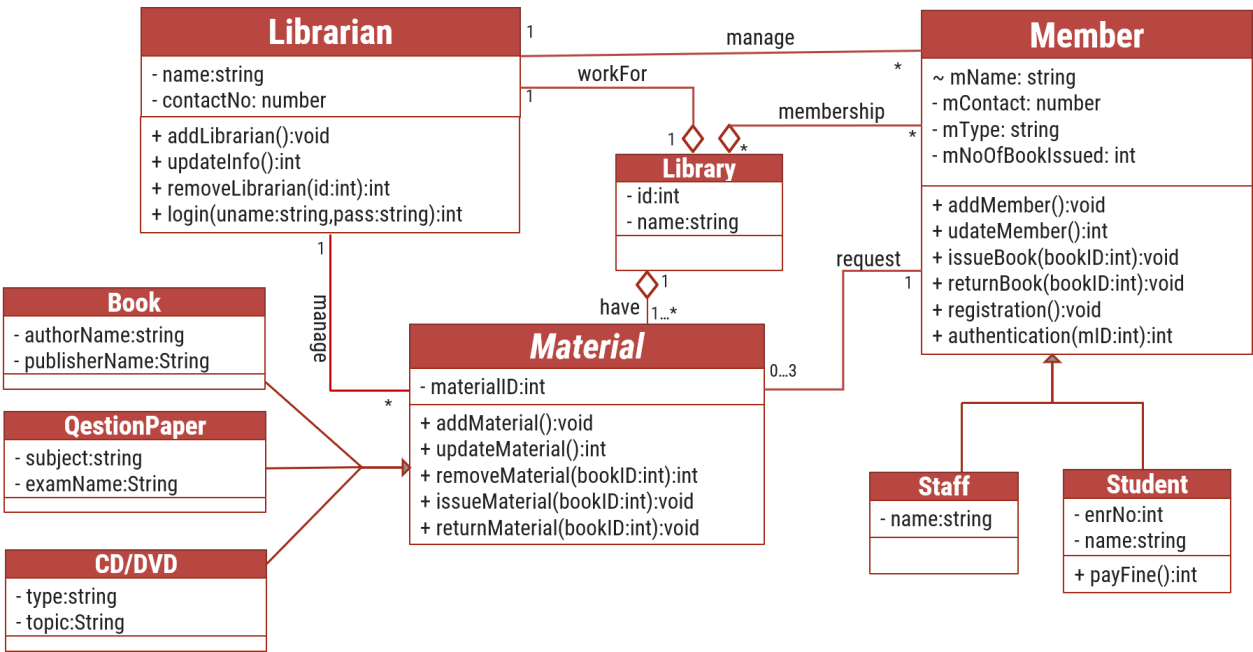


*Figure 2.4-1 State diagram of Book*



*Figure 2.4-2 State diagram for Librarian*

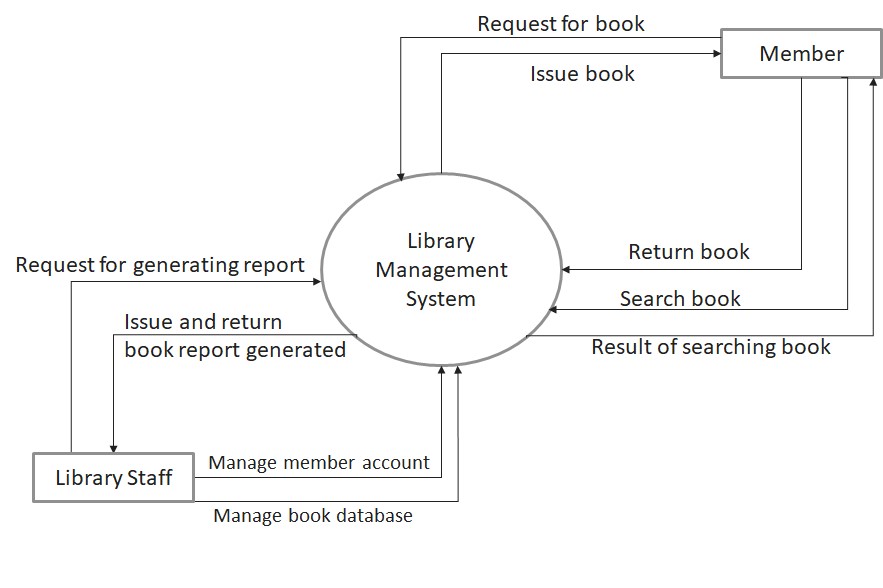
## Class diagram



*Figure 2.5-1 Class diagram for Library management system*

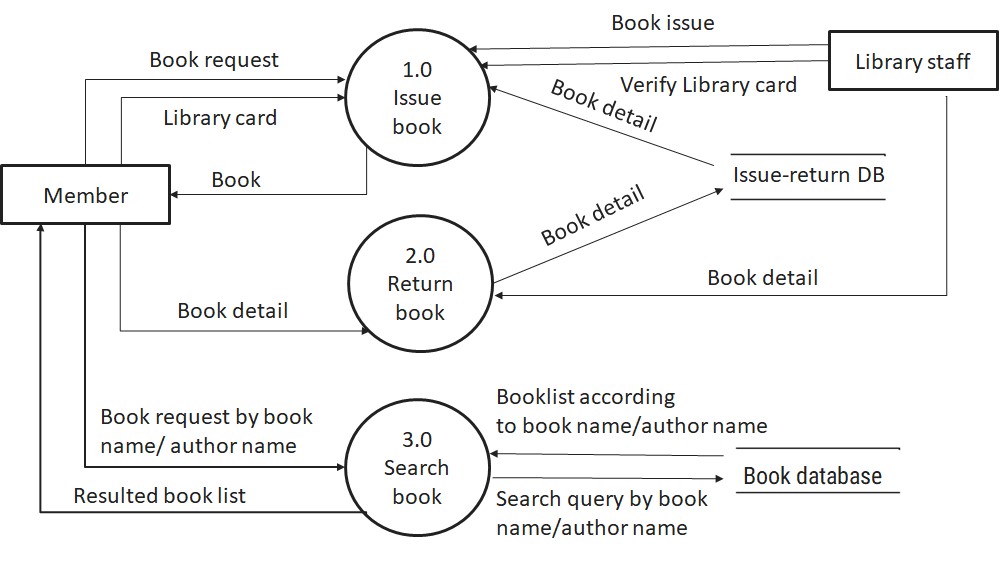
## Data flow diagram

### Context diagram (level-0)



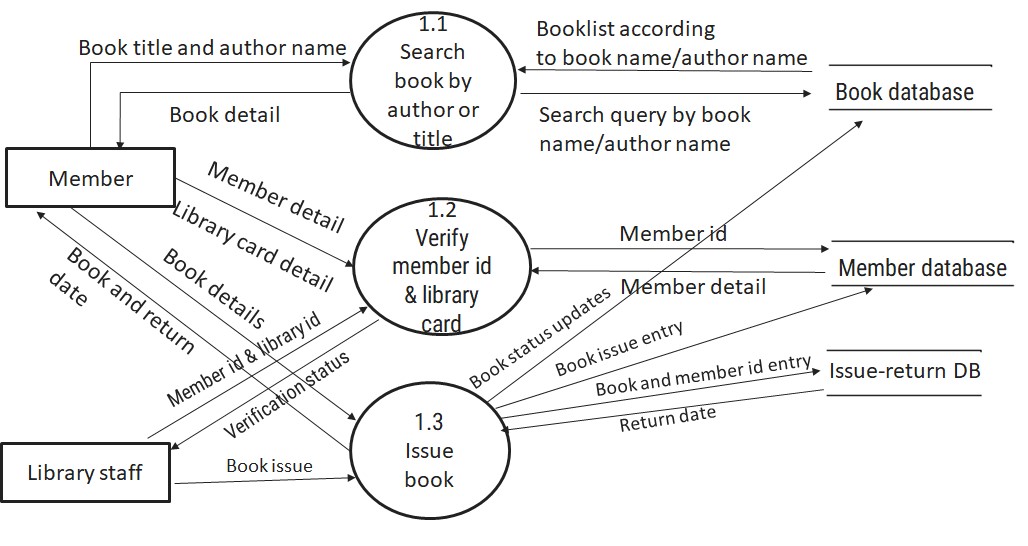
*Figure 2.6-1 Context diagram for Library management system*

### DFD Level-1



*Figure 2.6-2 DFD level-1 for Library management system*

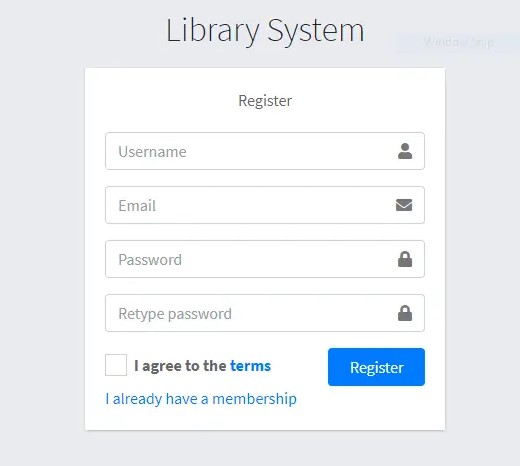
### DFD Level-2



*Figure 2.6-3 DFD level-2 for Issue book*

# External interface requirement (Screens)

## Screen-1: Registration Form



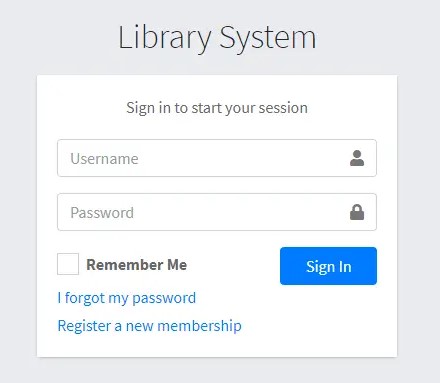
*Figure 3.1-1 Screen-1: Registration Form*

**Purpose:** This form will allow the target end-users to register in the system. To register , the following information will be encoded in the system.

*Table 3.1-1 Screen element of Registration form*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Username | Textbox | M | 1 | Username field should be editable and accept the Username. |
| **2** | Email | Textbox | M | 1 | Email field should be editable and accept the email with proper format. |
| **3** | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| **4** | Retype password | Password | M | 1 | Retype Password field should be editable and accept the password and display as star or dot. |
| **5** | I agree to the terms | Checkbox | M | 1 | Checkbox for accepting terms and condition. |
| **6** | I already have a membership | Link | ------ | ------ | Link for navigate to login page for registered user. |
| **7** | Register | Button | ------ | ------ | Register is a button for store the entered data into database. |

## Screen-2: Login Form



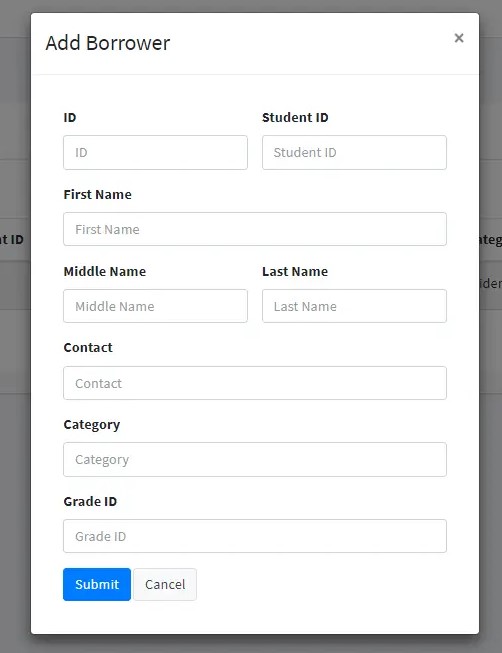
*Figure 3.2-1 Screen-2: Login Form*

**Purpose:** This form will be used by the system’s users to access records and features of the system. The users will input the correct combination of their username and password to be able to login to the system.

*Table 3.2-1 Screen element of Login form*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Username | Textbox | M | 1 | Username field should be editable and accept the Username. |
| **2** | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| **3** | Remember Me | Checkbox | M | 1 | Saving login credentials through remember me checkbox |
| **4** | I forgot my  password | Link | ------ | ------ | Link for navigate to Forgot password page for allows users to recover password. |
| **5** | Register a new membership | Link | ------ | ------ | Link for navigate to membership registration. |
| **6** | Sign in | Button | ------ | ------ | Login button navigates to another page even if valid login credentials. |

## Screen-3: Add borrower book



*Figure 3.3-1 Screen-3: Add borrower book*

**Purpose:** This module will allow the system administrator to add, edit, update or delete borrowers of book. The admin can add borrower information and manage it.

*Table 3.3-1 Screen element of Add borrower book*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Borrower ID | Textbox | M | 1 | Borrower ID field should be editable and fill autogenerated Borrower ID. |
| **2** | Student ID | Textbox | M | 1 | Student ID field should be editable and accept Student barcode id. |
| **3** | First Name | Textbox | M | 1 | First Name field should be editable and accept only string as a borrower first name. |
| **4** | Middle Name | Textbox | O | 1 | Middle Name field should be editable and accept only string as a borrower middle name. |
| **5** | Last Name | Textbox | M | 1 | Last Name field should be editable and accept only string as a borrower last name. |
| **6** | Contact | Textbox | M | 1 | Contact field should be editable and accept only 10-digit mobile number. |
| **7** | Grade Id | Textbox | O | 1 | Auto fetch detail from the database if proper student id is entered. |
| **8** | Submit | Button | ----- | ----- | Submit is a button for store the entered data into database. |
| **9** | Cancel | Button | ----- | ----- | Cancel is a button for close add borrower dialog. |

# Database design

## List of Tables

* Product
* Customer
* Manager
* Admin

*Table 4.1-1 Table: Product*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **ProductID** | int | NN | PK (Auto Increment) |  |
| **ProductName** | varchar(100) | NN |  |  |
| **ProductPrice** | int | NN |  |  |
| **ExpiryDate** | datetime | NN |  |  |
| **Discount** | Decimal(4,2) | AN |  |  |

*Table 4.1-2 Table: Customer*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **Customer­ID** | int | NN | PK (Auto Increment) |  |
| **CustomerName** | Varchar(100) | NN |  |  |
| **PurchaseDate** | Datetime | NN |  |  |
| **Address** | Varchar(250) | NN |  |  |
| **Cart** | Nvarchar | AN |  |  |

*Table 4.1-3 Table: Manager*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **ManagerID** | int | NN | PK (Auto Increment) |  |
| **ManagerName** | varchar(100) | NN |  |  |
| **IsInventoryAvailable** | Boolean | NN |  |  |

*Table 4.1-4 Table: Admin*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **AdminID** | int | NN | PK (Auto Increment) |  |
| **AdminName** | varchar(100) | NN |  |  |
| **ContactNumber** | number(10,0) | AN |  |  |

# Stories and Scenario

## Story-1: Add New Book in Library Catalogue

|  |  |  |
| --- | --- | --- |
| *Story #* ***S1*** | **:** | **As a** Librarian,  **I want to** add a new book in library catalogue  **So that** everyone can easily find and borrow it. |
| **Priority** | **:** | High |
| **Estimate** | **:** | XL |
| **Reason** | **:** | The addition of a new book to the library catalogue is crucial for ensuring that the library's collection is up-to-date and accessible to everyone. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S1.1*** | **:** | Adding a New Book with Valid Information |
| **Prerequisite** | **:** | Librarian is logged in to the Library management system. |
| **Acceptance**  **Criteria** | **:** | **Given:**  The Librarian is navigated to the library catalog management page. Valid book information, including title, author, ISBN, and other relevant details is added.  **When:**  The librarian selects the "Add New Book" option  And The librarian enters valid book details  The librarian clicks the "Save" button to add the book to the catalog.    **Then t**he system successfully adds the book to the catalog and the librarian receives a confirmation message with the book's identification number. |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S1.2*** | **:** | Adding a New Book with Invalid Information. |
| **Prerequisite** | **:** | The librarian is logged into the library management system. |
| **Acceptance**  **Criteria** | **:** | **Given:** The librarian is on the library catalogue management page  **When:** The librarian selects the "Add New Book" option and the librarian enters an incomplete or incorrect book details and librarian clicks the "Save" button to add the book to the catalogue.  **Then t**he system displays error messages for the incorrect or missing information and the book is not added to the catalogue. |

### Scenario# S1.3

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S1.3*** | **:** | Attempting to Add a Duplicate Book |
| **Prerequisite** | **:** | The librarian is logged into the library management system and the librarian is on the library catalogue management page |
| **Acceptance**  **Criteria** | **:** | **Given**: The book information, including title, author, ISBN, and other relevant details, is available and the book with the same ISBN is already in the catalogue.  **When**: User Clicks on “Add book” button. Enter a number of copies with the same book detail mentioned in the field. |
|  | | **Then**: Generate unique book id, barcode and spine label for various book of same title. |

## Story-2: Search Book

|  |  |  |
| --- | --- | --- |
| *Story #* ***S2*** | : | **As a** Librarian or member,  **I want to** search for books by title, author, or keyword,  **So that** I can quickly find books that match my interests. |
| **Priority** | **:** | High |
| **Estimate** | **:** | M |
| **Reason** | **:** | Implementing a search functionality is essential for enhancing the user experience, as it allows librarian and member to efficiently discover and access the library's resources. |

## Story-3: Manage due date for borrowed book

|  |  |  |
| --- | --- | --- |
| *Story #* ***S3*** | **:** | **As** Librarian,  **I want** to manage due dates for borrowed books  **So that** I can ensure that books are returned on time and avoid overdue fines. |
| **Priority** | **:** | High |
| **Estimate** | **:** | M |
| **Reason** | **:** | Proper due date management is crucial for maintaining the library's collection and ensuring that books are available for all members. |

## Story-4: Renew book

|  |  |  |
| --- | --- | --- |
| *Story #* ***S3*** | **:** | **As** Librarian,  **I want** to renew a book that I have borrowed  **So that** I can extend my borrowing period if needed. |
| **Priority** | **:** | Medium |
| **Estimate** | **:** | M |
| **Reason** | **:** | Book renewal functionality is a convenience feature for librarian, allowing them to keep a book for an extended period if no one else has requested it. |

## Story-5: Generate a report on book usage and availability

|  |  |  |
| --- | --- | --- |
| *Story #* ***S3*** | **:** | **As** Librarian,  **I want** to generate reports on library usage and book availability **So that** I can make informed decisions about library’s books. |
| **Priority** | **:** | Medium |
| **Estimate** | **:** | L |
| **Reason** | **:** | Reporting functionality helps librarians track the usage of library resources and make data-driven decisions to improve services and collections. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name:** | **EMI Calculator** | **Test Designed by:** | **P. U. Jadeja** |
| **Module Name:** | **Login** | **Test Designed date:** | 01-10-2023 |
| **Release Version:** | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 15-01-2023 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pre-condition**: **Web application should be accessible** | | | |  |
| **Test Case ID** | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| **TC\_001** | Login to web application with valid credential | Functional | Login to Library management system web application through valid credential | TC\_001 |
| **TC\_002** | Login to web application with  invalid credential | Functional | Login to Library management system web application through invalid credential | TC\_002 |
| **TC\_003** | Varify login page elements | GUI | varify that all elements are availabe on login page | TC\_003 |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with valid credential |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **BUG ID** |
| 1 | Access Web application URL | The site  launched properly | Site launched successfully | Pass |  | [https://accounts. google.com/Servi ceLogin](https://accounts.google.com/ServiceLogin) |  |
| 2 | Enter valid Username  in username  field | Username field should be editable and accept the  Username | Username input accepted | Pass |  | Username: Rbgondaliya@ gmail.com |  |
| 3 | Enter valid Password in Password  field | Password field should be editable and accept the password and display as star or dot | Password input displayed in dot and  accepted | pass |  | Password: rbgondaliya |  |
| 4 | Enter valid captcha  code in  captch field | Captch field should editable and accept captcha and captcha is case sensitive | Captcha input accepted | Pass | Step require d when human  action validati on | get captcha from image which is near by captcha  field |  |
|  |  |  |  |  | perfor m |  |  |
| 5 | Click on  login button | User should login into site and navigated to dashboard | User navigated to dashboard and  username should br display in top of the right side. | pass |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with invalid credential |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Verify that User is not able to Login with  invalid  Username and  invalid  Password | Should be display an error message enter wrong  username  or password | Display an error of wrong username and password | Pass |  |  |  |
| 2 | Verify that User is not able to Login with Valid Username and  invalid  Password | Should be display an error message enter wrong password | Display an error of wrong password | Pass |  |  |  |
| 3 | Verify that User is not able to Login with | Should be display an error message | Display an error  Username not found | Pass |  |  |  |
|  | invalid  Username and Valid Password | User not found |  |  |  |  |  |
| 4 | Verify that User is not able to Login with  blank  Username  or  Password | Set required field validation  message  for  Username and Password | Display an error of wrong username and password | Fail | Not performa  a validation  function  fix it |  | Bug\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Varify login page elements |
| **Test Type** | GUI |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case**  **Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug**  **ID** |
| 1 | Launch application with the  given url | The site launched properly | Site launched successfully | Pass |  | [https://accounts.google.](https://accounts.google.com/ServiceLogin)  [com/ServiceLogin](https://accounts.google.com/ServiceLogin) |  |
| 2 | Verify that the login screen contains elements  such as  Username,  Password,  Sign in  button,  Remember password check box, Forgot password link, and Create an account link. | All listed control displayed  properly  on the  page | Login page loaded successfully | Pass |  |  |  |
| 3 | Verify that cursor is focused on  “Usernam  e” text box on the  page load | Cursor is  focused  in  Usernam  e textbox | Cursor  focus in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionali  ty is  working properly or not | When tab pressed cursor  move in  next control | Cursor moving in next control | Pass |  |  |  |
| 5 | Verify that all the fields such as  Username, Password has a valid  placeholde  r | All text  fields have proper  placehold er | All text fields have proper placeholder | Pass |  |  |  |
| 6 | Verify that the labels float  upward  when the text field is in focus or filled (In case of  floating label) | When field is focused or filled, label display on top of the  filled | When field is focus or filled, label display on top of the  filled | Pass | step required when  fields with floating label |  |  |
| 7 | verify that forgot password  link  working properly | when click on forgot password load forgot passworg page | forgot  password link not  working | Fail |  |  |  |

# References

* http://www.w3schools.com/html/html\_intro.asp
* https://www.w3schools.com/php/default.asp
* https://www.javatpoint.com/uml

*Figure 2.2-3 1*