

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  Prof. R. B. Gondaliya  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  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](#_Toc189087928)

[List of Tables II](#_Toc189087929)

[1 Introduction 1](#_Toc189087930)

[1.1 Product perspective 1](#_Toc189087931)

[1.2 Product features 1](#_Toc189087932)

[1.2.1 There are four different users who will be using this product: 1](#_Toc189087933)

[1.2.2 The features that are required for the Customer are: 1](#_Toc189087934)

[1.2.3 The features that are required for the Manager are: 1](#_Toc189087935)

[1.2.4 The features that are required for the admin are: 1](#_Toc189087936)

[1.3 Functional Requirement 1](#_Toc189087937)

[1.3.1 Customer 1](#_Toc189087938)

[1.3.2 Manager 2](#_Toc189087939)

[1.3.3 Cashier 2](#_Toc189087940)

[1.3.4 Admin 2](#_Toc189087941)

[1.4 Non-Functional Requirement 2](#_Toc189087942)

[1.4.1 Usability: 2](#_Toc189087943)

[1.4.2 Accuracy: 2](#_Toc189087944)

[1.4.3 Availability: 2](#_Toc189087945)

[1.4.4 Maintainability: 2](#_Toc189087946)

[2 Design and Implementation Constraints 3](#_Toc189087947)

[2.1 Use case diagram 3](#_Toc189087948)

[2.2 Activity diagram and Swimlane diagram 4](#_Toc189087949)

[2.3 Sequence diagram 7](#_Toc189087950)

[2.4 State diagram 8](#_Toc189087951)

[2.5 Class diagram 9](#_Toc189087952)

[2.6 Data flow diagram 10](#_Toc189087953)

[2.6.1 Context diagram (level-0) 10](#_Toc189087954)

[2.6.2 DFD Level-1 10](#_Toc189087955)

[2.6.3 DFD Level-2 11](#_Toc189087956)

[3 External interface requirement (Screens) 12](#_Toc189087957)

[3.1 Screen-1: Give Feedback 12](#_Toc189087958)

[3.2 Screen-2: Payment Methods 13](#_Toc189087959)

[3.3 Screen-3: Assign Roles 14](#_Toc189087960)

[3.4 Screen-4: Cancel Transactions 15](#_Toc189087961)

[3.5 Screen-5: Remove Roles 16](#_Toc189087962)

[4 Database design 17](#_Toc189087963)

[4.1 List of Tables 17](#_Toc189087964)

[5 Stories and Scenario 18](#_Toc189087965)

[5.1 Story-1: Add New Book in Library Catalogue 18](#_Toc189087966)

[5.1.1 Scenario# S1.1 18](#_Toc189087967)

[5.1.2 Scenario# S1.2 18](#_Toc189087968)

[5.1.3 Scenario# S1.3 18](#_Toc189087969)

[5.2 Story-2: Search Book 19](#_Toc189087970)

[5.3 Story-3: Manage due date for borrowed book 19](#_Toc189087971)

[5.4 Story-4: Renew book 19](#_Toc189087972)

[5.5 Story-5: Generate a report on book usage and availability 19](#_Toc189087973)

[6 Test cases 20](#_Toc189087974)

[7 References 24](#_Toc189087975)

# List of Figures

[Figure 2.1‑1 Use case diagram for supermarket management system 3](#_Toc189087911)

[Figure 2.2‑1 Activity diagram for purchasing product 4](#_Toc189087912)

[Figure 2.2‑2 Swimlane diagram for purchasing product 4](#_Toc189087913)

[Figure 2.2‑3 Activity Diagram for Adding Product 5](#_Toc189087914)

[Figure 2.2‑4 Swimlane Diagram for Adding Product 6](#_Toc189087915)

[Figure 2.3‑1 Sequence diagram for Book Issue 7](#_Toc189087916)

[Figure 2.4‑1 State diagram of Book 8](#_Toc189087917)

[Figure 2.4‑2 State diagram for Librarian 8](#_Toc189087918)

[Figure 2.5‑1 Class diagram for Library management system 9](#_Toc189087919)

[Figure 2.6‑1 Context diagram for Library management system 10](#_Toc189087920)

[Figure 2.6‑2 DFD level-1 for Library management system 10](#_Toc189087921)

[Figure 2.6‑3 DFD level-2 for Issue book 11](#_Toc189087922)

[Figure 3.1‑1 Screen-1: Give Feedback 12](#_Toc189087923)

[Figure 3.2‑1 Screen-2: Payment Methods 13](#_Toc189087924)

[Figure 3.3‑1 Screen-3: Assign Roles 14](#_Toc189087925)

[Figure 3.4‑1 Screen-4: Cancel Transactions 15](#_Toc189087926)

[Figure 3.5‑1 Screen-5: Remove Roles 16](#_Toc189087927)

# List of Tables

[Table 3.1‑1 Screen element of Give Feedback form 12](#_Toc189087976)

[Table 3.2‑1 Screen element of Payment Methods 13](#_Toc189087977)

[Table 3.3‑1 Screen element of Assign Roles 14](#_Toc189087978)

[Table 3.4‑1 Screen element of Cancel Transactions 15](#_Toc189087979)

[Table 3.5‑1 Screen element of Remove Roles 16](#_Toc189087980)

[Table 4.1‑1 Table: Book 17](#_Toc189087981)

[Table 4.1‑2 Table: Borrower 17](#_Toc189087982)

[Table 4.1‑3 Table: Staff 17](#_Toc189087983)

[Table 4.1‑4 Table: Student 17](#_Toc189087984)

# 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

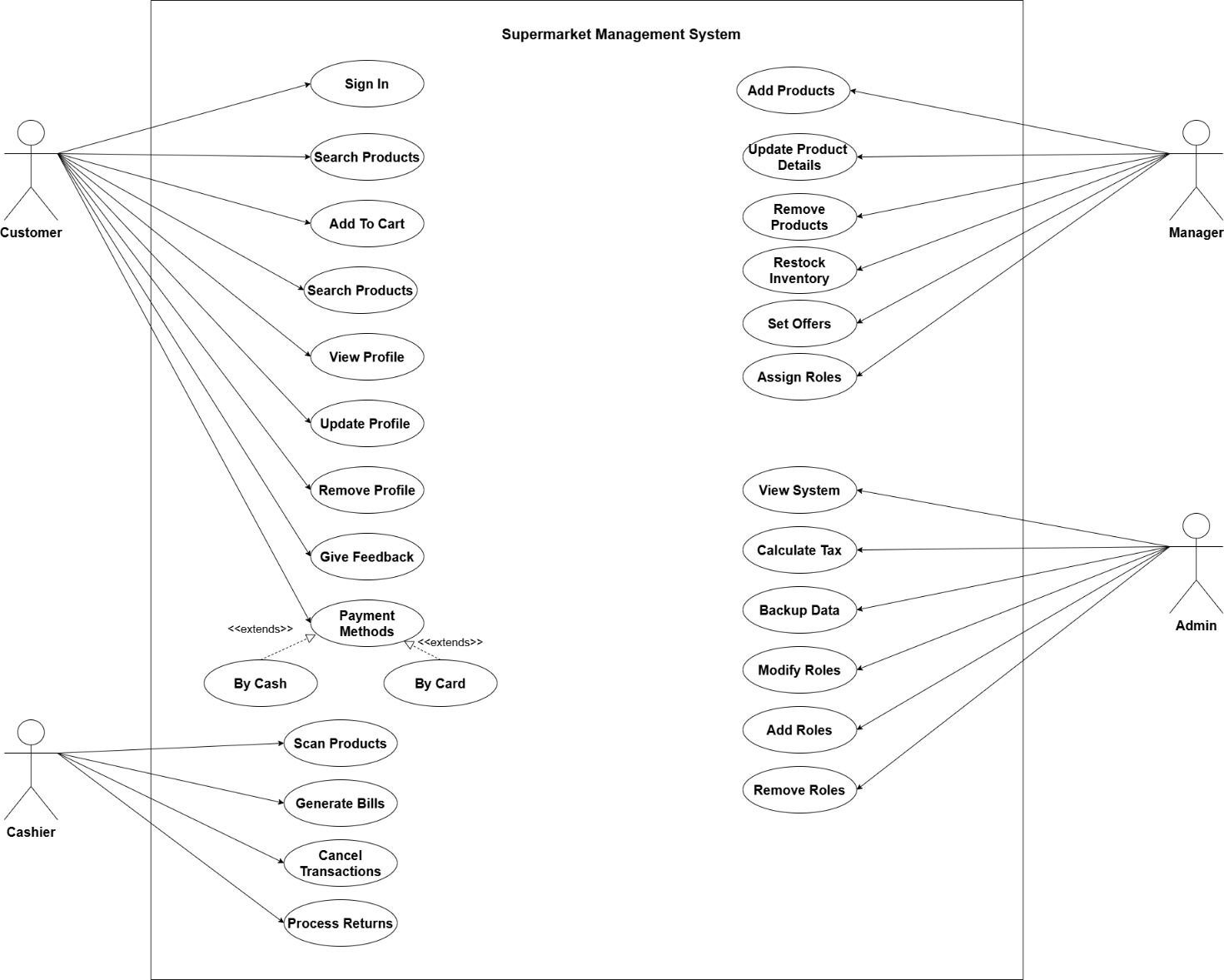


Figure 2.1‑1 Use case diagram for supermarket management system

## Activity diagram and Swimlane diagram

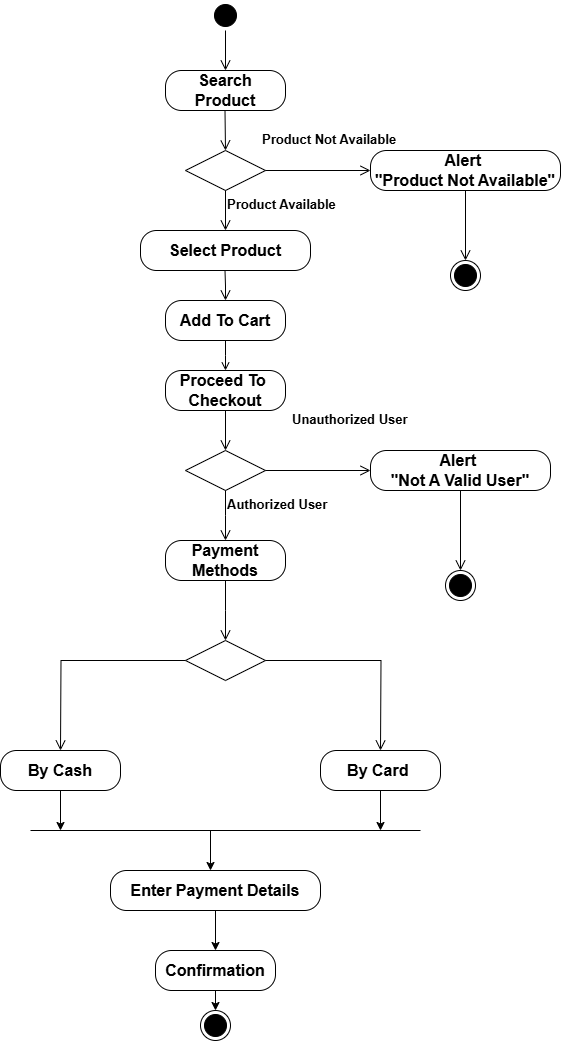


Figure 2.2‑1 Activity diagram for purchasing product

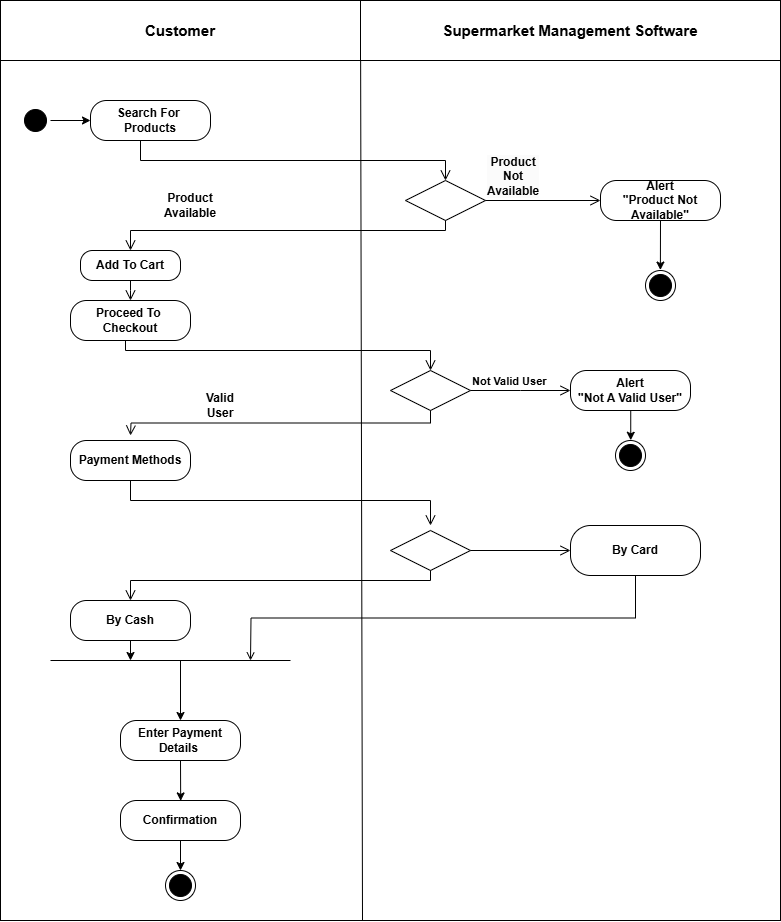


Figure 2.2‑2 Swimlane diagram for purchasing product

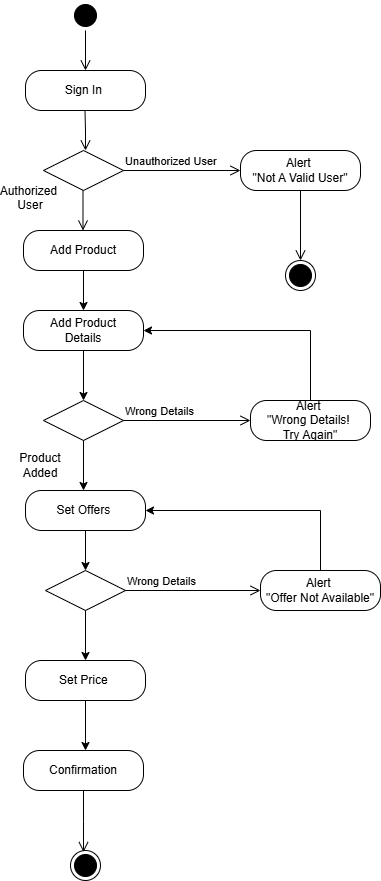


Figure ‑ Activity Diagram for Adding Product

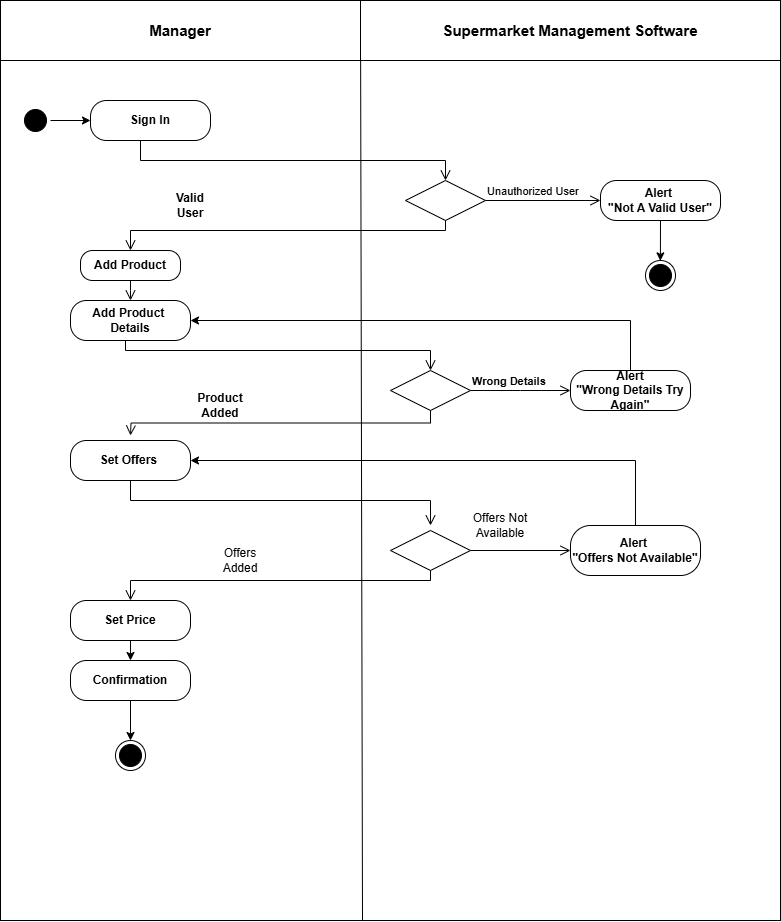
**

Figure ‑ 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: Give Feedback

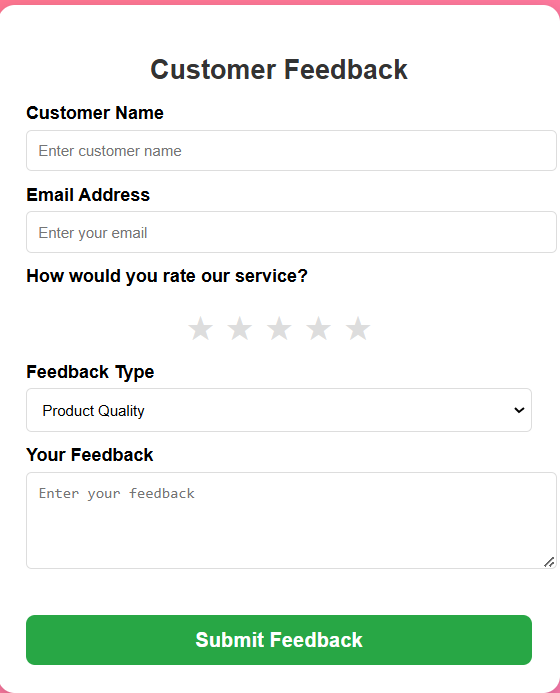


Figure 3.1‑1 Screen-1: Give Feedback

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

Table 3.1‑1 Screen element of Give Feedback form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Customer Name | Textbox | M | 1 | Customer Name field should be editable and accept the Customer Name. |
| 2 | Email Address | Textbox | M | 1 | Email field should be editable and accept the email with proper format. |
| 3 | Rating | Radio Button | M | 1 | Rating field should be editable and accept the rating. |
| 4 | Feedback Type | Dropdown  Menu | M | 1 | Feedback Type field should be editable and accept the user’s choice in which they want to give feedback |
| 5 | Feedback | Textbox | M | 1 | Feedback field should be editable and accept the feedback which user want to give on particular topic |
| 7 | Submit | Button | ------ | ------ | Submit is a button for store the entered data into database. |

## Screen-2: Payment Methods

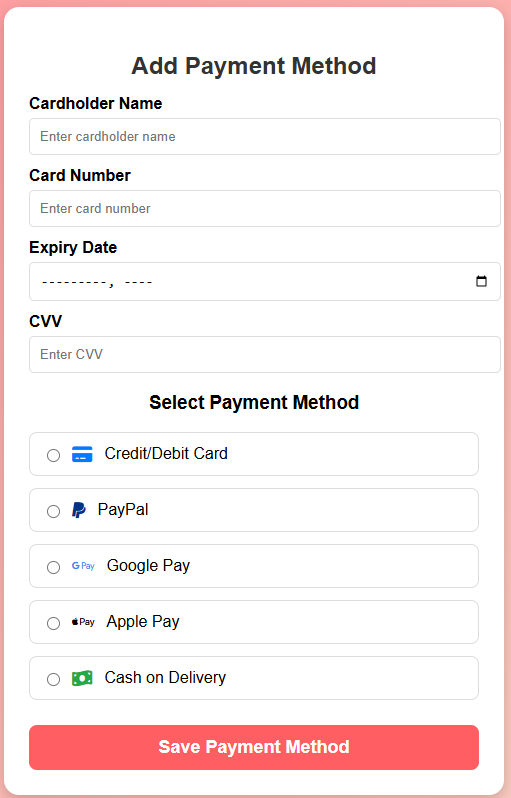


Figure 3.2‑1 Screen-2: Payment Methods

**Purpose:** This form will be used by the system’s users to select payment methods like Credit / Debit Card, Gpay, etc. 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 Payment Methods

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Cardholder Name | Textbox | M | 1 | Cardholder Name field should be editable and accept the Cardholder Name. |
| 2 | Card Number | Textbox | M | 1 | Card Number field should be editable and accept the Card Number. |
| 3 | Expiry Date | Datetime | M | 1 | Expiry Date field should accept Expiry Date and should be checked the valid date user enters. |
| 4 | CVV | Textbox | M | 1 | CVV should be accept Card Number and should be unique for all customers. |
| 5 | Select Payment Methods | Radio | M | 1 | This field should be accepted by user in which he/she wants to do payment. |
| 6 | Save | Button | ------ | ------ | Save is a button for store the entered data into database. |

## Screen-3: Assign Roles

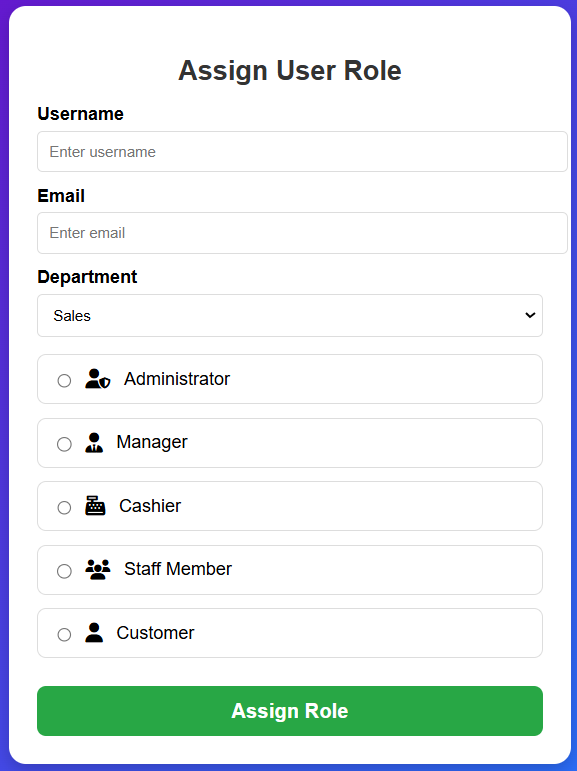


Figure 3.3‑1 Screen-3: Assign Roles

**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 Assign Roles

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | Department | Dropdown | M | 1 | Department field should be editable and accept the department choices. |
| 4 | Assign | Radio | M | 1 | Assign field should be editable and can assign role. |
| 5 | Assign Role | Button | ----- | ----- | Assign Role is a button for store the entered data into database. |

## Screen-4: Cancel Transactions

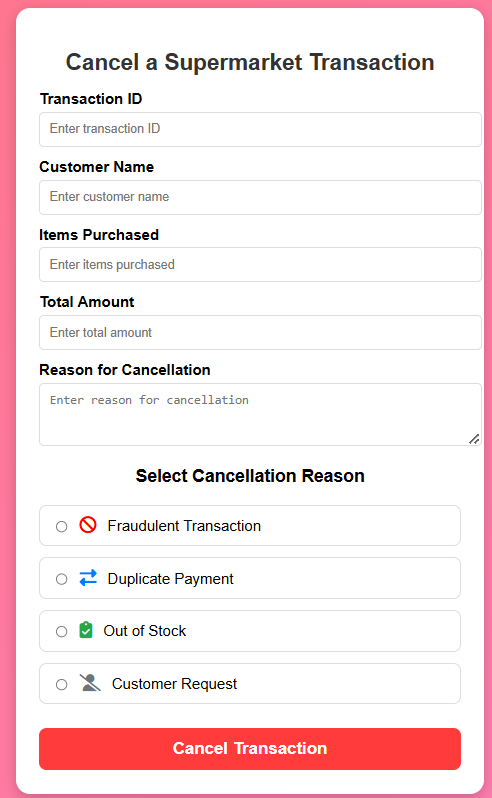


Figure 3.4‑1 Screen-4: Cancel Transactions

**Purpose:** This module will allow the system users to cancel transaction at any time.

Table 3.4‑1 Screen element of Cancel Transactions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Transaction ID | Textbox | M | 1 | Transaction Id should be editable and accept the Id. |
| 2 | Customer Name | Textbox | M | 1 | Customer Name field should be editable and accept the Customer Name. |
| 3 | Items Purchased | Textbox | M | 1 | This field should be editable and accept the Item Purchased details. |
| 4 | Total Amount | Textbox | M | 1 | This field should be editable and accept the Item Purchased details. |
| 5 | Reason For Cancellation | Textbox | M | 1 | This field should be editable and accept the Item Purchased details. |
| 6 | Select Reason | Radio | M | 1 | User can select cancellation reason in this field. |
| 7 | Cancel Transaction | Button | ---- | ---- | Cancel Transaction is a button for store the entered data into database. |

## Screen-5: Remove Roles

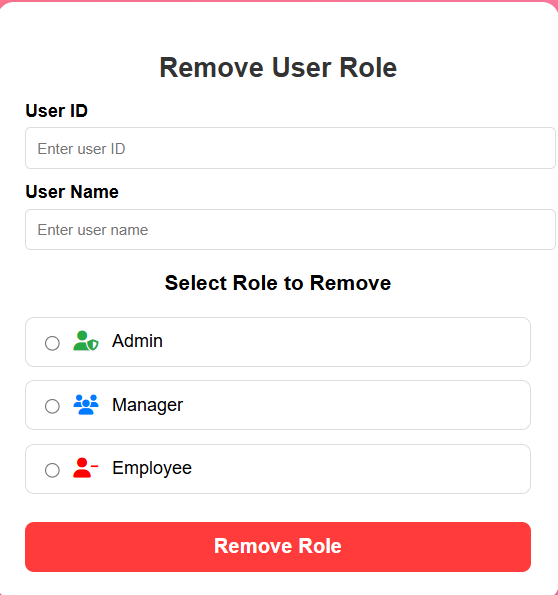


Figure 3.5‑1 Screen-5: Remove Roles

**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 3.5‑1 Screen element of Remove Roles

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | User ID | Textbox | M | 1 | User Id should be editable and accept the Id. |
| 2 | User Name | Textbox | M | 1 | User Name field should be editable and accept the User Name. |
| 3 | Remove | Radio | M | 1 | User can select what user he want to remove. |
| 4 | Remove Role | Button | ---- | ---- | Cancel Transaction is a button for store the entered data into database. |

# Database design

## List of Tables

* Book
* Borrower
* Student
* Staff

Table 4.1‑1 Table: Book

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| BookID | int | NN | PK (Auto Increment) |  |
| BookName | varchar(100) | NN |  |  |
| ISBN | varchar(100) | AN |  |  |
| Publication\_year | int | AN |  |  |
| Language | varchar(50) | AN |  |  |

Table 4.1‑2 Table: Borrower

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| BorrowerID | int | NN | PK (Auto Increment) |  |
| BookID | varchar(100) | NN | FK | Reference of Book Table |
| BorrowedFromDate | DateTime | AN |  |  |
| BorrowedToDate | DateTime | AN |  |  |
| ActualReturnDate | DateTime | AN |  |  |
| IssuedBy | int | NN | FK | Reference of Student and Staff Table |

Table 4.1‑3 Table: Staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| StaffID | int | NN | PK (Auto Increment) |  |
| StaffName | varchar(100) | AN |  |  |
| IsAdmin | Boolean | AN |  |  |
| Designation | varchar(100) | AN |  |  |

Table 4.1‑4 Table: Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| StudentID | int | NN | PK (Auto Increment) |  |
| StudentName | varchar(100) | AN |  |  |
| Gender | varchar(100) | AN |  |  |
| DOB | DateTime | AN |  |  |
| Department | varchar(100) | AN |  |  |
| Contact | 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/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 required when human action validation perform | get captcha from image which is near by captcha field |  |
| 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 invalid Username and Valid Password | Should be display an error message User not found | Display an error Username not found | Pass |  |  |  |
| 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.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 “Username” text box on the page load | Cursor is focused in Username textbox | Cursor focus in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionality 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 placeholder | All text fields have proper placeholder | 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