

Darshan University

A Project Report on

**“Loan Management System”**

Under the subject

**Software Engineering (2301CS405)**

B. Tech, Semester – IV

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Kalp Vipul Viradia | Enrollment No.: 23010101299 |
| 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 **“Loan Management System”** submitted in partial fulfilment for the Semester-5 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.

(Kalp Vipul Viradia)

Student’s Signature

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

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**CERTIFICATE**

This is to certify that the SRS on **“Loan Management System” has** been satisfactorily prepared by **Kalp Vipul Viradia** (**23010101299**) 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

**Kalp Vipul Viradia**

**ABSTRACT**

The Loan Management System (LMS) is a digital solution designed to automate and streamline the entire loan process for financial institutions. This system simplifies loan application, approval, repayment, and reporting, reducing manual efforts and ensuring accuracy. It offers a user-friendly interface for both clients and loan officers, enabling clients to apply for loans securely and track their repayment schedules. Once a loan application is submitted, loan officers can review, approve, or reject it, with the system automatically updating the status. Clients can view their outstanding balances and make payments through integrated payment gateways.

Administrators can generate comprehensive reports on loan disbursements, repayments, and overdue loans, helping in effective resource management. The system’s secure login and data privacy features ensure that client information is protected. By reducing paperwork, minimizing human error, and improving operational efficiency, the LMS enhances both the loan management process and the customer experience, providing a seamless and efficient solution for loan handling.

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

**Table of Contents**

[List of Figures I](#_Toc185767868)

[List of Tables II](#_Toc185767869)

[1 Introduction 1](#_Toc185767870)

[1.1 Product perspective 1](#_Toc185767871)

[1.2 Product features 1](#_Toc185767872)

[1.2.1 There are three different users who will be using this product: 1](#_Toc185767873)

[1.2.2 The features that are required for the Admin are: 1](#_Toc185767874)

[1.2.3 The features that are required for the Loan officer are: 1](#_Toc185767875)

[1.2.4 The features that are required for the Customer are: 1](#_Toc185767876)

[1.3 Functional Requirement 2](#_Toc185767877)

[1.3.1 Admin 2](#_Toc185767878)

[1.3.2 Loan Officer 2](#_Toc185767879)

[1.3.3 Customer 2](#_Toc185767880)

[1.4 Non-Functional Requirement 3](#_Toc185767881)

[1.4.1 Usability: 3](#_Toc185767882)

[1.4.2 Accuracy: 3](#_Toc185767883)

[1.4.3 Availability: 3](#_Toc185767884)

[1.4.4 Maintainability: 3](#_Toc185767885)

[2 Design and Implementation Constraints 4](#_Toc185767886)

[2.1 Use case diagram 4](#_Toc185767887)

[2.2 Activity diagram and Swimlane diagram 5](#_Toc185767888)

[2.3 Sequence diagram 7](#_Toc185767889)

[2.4 State diagram 8](#_Toc185767890)

[2.5 Class diagram 9](#_Toc185767891)

[2.6 Data flow diagram 10](#_Toc185767892)

[2.6.1 Context diagram (level-0) 10](#_Toc185767893)

[2.6.2 DFD Level-1 10](#_Toc185767894)

[2.6.3 DFD Level-2 11](#_Toc185767895)

[3 External interface requirement (Screens) 12](#_Toc185767896)

[3.1 Screen-1: Registration Form 12](#_Toc185767897)

[3.2 Screen-2: Login Form 13](#_Toc185767898)

[3.3 Screen-3: Add borrower book 14](#_Toc185767899)

[4 Database design 15](#_Toc185767900)

[4.1 List of Tables 15](#_Toc185767901)

[5 Stories and Scenario 16](#_Toc185767902)

[5.1 Story-1: Add New Book in Library Catalogue 16](#_Toc185767903)

[5.1.1 Scenario# S1.1 16](#_Toc185767904)

[5.1.2 Scenario# S1.2 16](#_Toc185767905)

[5.1.3 Scenario# S1.3 16](#_Toc185767906)

[5.2 Story-2: Search Book 17](#_Toc185767907)

[5.3 Story-3: Manage due date for borrowed book 17](#_Toc185767908)

[5.4 Story-4: Renew book 17](#_Toc185767909)

[5.5 Story-5: Generate a report on book usage and availability 17](#_Toc185767910)

[6 Test cases 18](#_Toc185767911)

[7 References 22](#_Toc185767912)

# List of Figures

[Figure 2.1‑1 Use case diagram for loan management system 4](#_Toc185873403)

[Figure 2.2‑1 Activity diagram for taking loan 5](#_Toc185873404)

[Figure 2.2‑2 Activity diagram for approving loan 5](#_Toc185873405)

[Figure 2.2‑3 Swimlane diagram for taking loan 6](#_Toc185873406)

[Figure 2.2‑4 Swimlane diagram for approving loan 6](#_Toc185873407)

[Figure 2.3‑1 Sequence diagram for taking loan 7](#_Toc185873408)

[Figure 2.3‑2 Sequence diagram for admin user 7](#_Toc185873409)

[Figure 2.4‑1 State diagram for loan 8](#_Toc185873410)

[Figure 2.4‑2 State diagram for payment 8](#_Toc185873411)

[Figure 2.5‑1 Class diagram for Library management system 9](#_Toc185873412)

[Figure 2.6‑1 Context diagram for Library management system 10](#_Toc185873413)

[Figure 2.6‑2 DFD level-1 for Library management system 10](#_Toc185873414)

[Figure 2.6‑3 DFD level-2 for Issue book 11](#_Toc185873415)

[Figure 3.1‑1 Screen-1: Registration Form 12](#_Toc185873416)

[Figure 3.2‑1 Screen-2: Login Form 13](#_Toc185873417)

[Figure 3.3‑1 Screen-3: Add borrower book 14](#_Toc185873418)

# List of Tables

[Table 3.1‑1 Screen element of Registration form 12](#_Toc185766585)

[Table 3.2‑1 Screen element of Login form 13](#_Toc185766586)

[Table 3.3‑1 Screen element of Add borrower book 14](#_Toc185766587)

[Table 4.1‑1 Table: Book 15](#_Toc185766588)

[Table 4.1‑2 Table: Borrower 15](#_Toc185766589)

[Table 4.1‑3 Table: Staff 15](#_Toc185766590)

[Table 4.1‑4 Table: Student 15](#_Toc185766591)

# Introduction

## Product perspective

The **Loan Management System (LMS)** is designed to streamline the processes involved in managing loans, from origination to closure. It serves as a comprehensive solution for financial institutions, providing robust functionality to automate and manage loan-related operations efficiently.

## Product features

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

* **Admin**: Oversees the entire system and has full control over its functionality.
* **Loan Officer**: Handle loan operations, customer interactions, and compliance.
* **Customer**: Individuals applying for loan and view their loans.

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

* Oversee all loan operations and monitor system activities.
* Add, edit, or delete loan products, including their terms (interest rates, tenure, and eligibility criteria).
* Monitor system users (Loan Officers and Customers), including account creation, modification, and deactivation.
* Make and analyse detailed reports, including loan disbursement, overdue accounts, and revenue data.
* Configure system settings, including repayment penalties, grace periods, and automated notifications.
* Monitor compliance with regulatory requirements and generate audit reports.
* Handle system backups and data recovery processes.
* Assign roles and permissions for Loan Officers.
* Monitor the performance of Loan Officers through task and target tracking.

### The features that are required for the Loan officer are:

* Review and process loan applications, including document verification and credit assessment.
* Approve or reject loans based on customer eligibility and internal guidelines.
* Track the status of loans, including disbursement and repayments.
* View detailed customer profiles, including their loan history, repayment behaviour, and credit scores.
* Send reminders and notifications to customers regarding due dates and overdue payments.
* Generate customer-specific reports, such as outstanding balances and repayment schedules.
* Respond to customer queries or requests for loan restructuring or early closure.
* Record and track interactions with customers for future reference.
* Monitor pending tasks and customer escalations assigned by the Admin.

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

* View available loan products with details such as interest rates, tenure, and eligibility requirements.
* Apply for loans by submitting necessary documentation and filling out the application form.
* Track the status of loan applications and receive updates on approvals or rejections.
* Access their loan account to view repayment schedules, outstanding balances, and due dates.
* Make loan repayments through integrated payment gateways or other channels.
* Request changes to loan terms, such as tenure extension or restructuring.
* Access notifications and reminders for upcoming repayments or overdue amounts.
* Update personal details such as address, contact information, or bank account details.

## Functional Requirement

### Admin

* **Add Loan Product**: Add new loan products with details such as interest rate, tenure, eligibility criteria, and terms and conditions to the system.
* **Update Loan Product**: Modify existing loan products if there are changes in terms, rates, or eligibility criteria.
* **Delete Loan Product**: Remove discontinued or incorrect loan products from the system.
* **User Management**: Maintain a database of all system users, including Loan Officers and Customers, with details like names, IDs, and permissions.
* **Set User Permissions**: Assign specific roles and access levels to Loan Officers.
* **Generate Reports**: Create detailed reports on loans, overdue accounts, revenue, and regulatory compliance.
* **Backup and Recovery**: Take system backups and ensure data recovery in case of failures.
* **Audit Logs**: Track all system activities and modifications for compliance and accountability.
* **Monitor Loan Officer Performance**: Track tasks and performance metrics of Loan Officers.

### Loan Officer

* **Authenticate**: Loan Officers must log in with valid credentials to access the system.
* **Review Loan Applications**: Assess customer loan applications, verify submitted documents, and evaluate eligibility criteria.
* **Approve/Reject Loans**: Decide the approval or rejection of loan applications based on customer creditworthiness and institutional guidelines.
* **Manage Customer Profiles**: Access and update customer records, including personal details and loan histories.
* **Track Loans**: View the lifecycle of loans, including application, disbursement, and repayment.
* **Send Notifications**: Notify customers about due dates, overdue payments, or changes in loan terms.
* **Generate Customer Reports**: Create detailed reports specific to individual customers, such as repayment schedules and outstanding balances.
* **Request Loan Restructuring**: Process customer requests for changes in loan terms, such as tenure extension or repayment restructuring.
* **Handle Customer Queries**: Respond to customer inquiries about loans or system-related issues.

### Customer

* **Authenticate**: Customers must log in using valid credentials to access their accounts.
* **Apply for Loan**: Submit loan applications with all required information and supporting documents.
* **Track Loan Application**: View the status of their loan application, including updates on approvals or rejections.
* **View Loan Details**: Access detailed information about their loans, including repayment schedules, interest rates, and outstanding balances.
* **Make Repayments**: Pay EMIs or clear overdue payments through integrated payment gateways.
* **Request Loan Changes**: Submit requests for loan restructuring, early closure, or increases in loan amounts.
* **Search Loan Products**: Browse available loan products and calculate EMIs using a loan calculator.
* **Access Notifications**: View alerts for upcoming due dates, payment confirmations, and system updates.
* **Update Personal Details**: Update account information, including address, contact details, or linked bank accounts.
* **Download Statements**: Retrieve loan account statements and payment receipts for record-keeping.

## 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

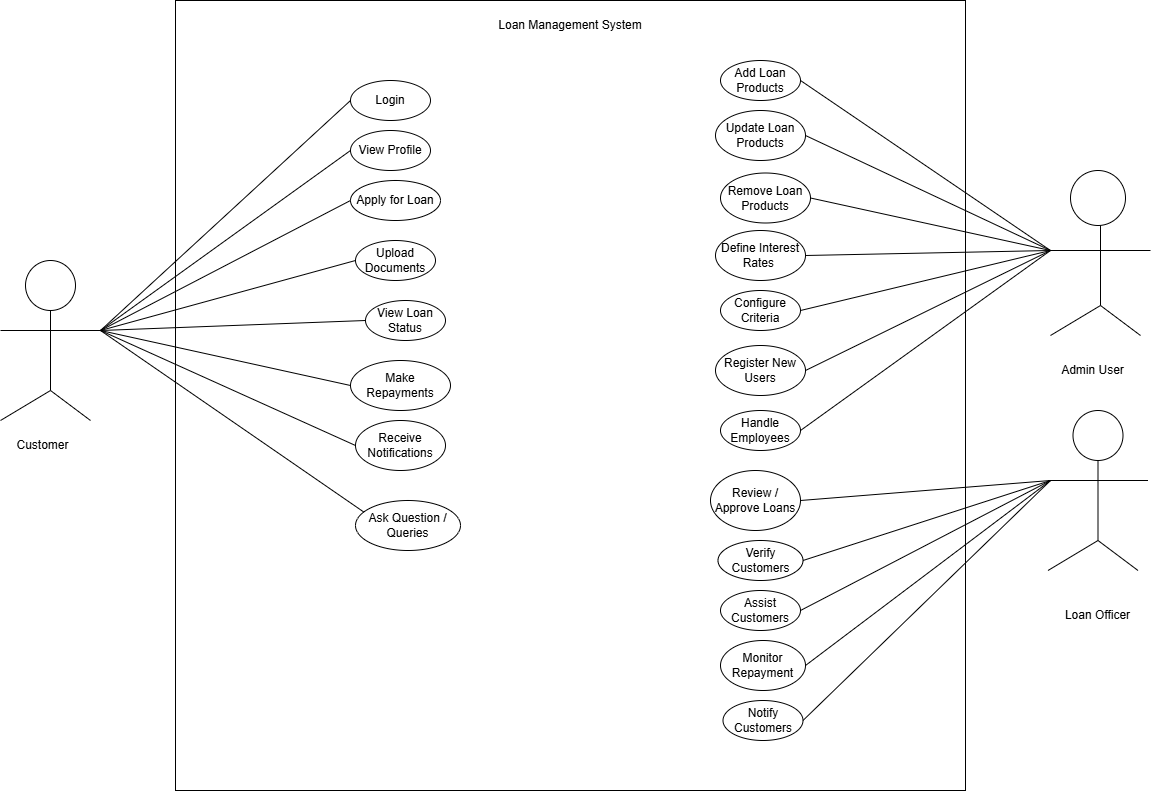


Figure ‑ Use case diagram for loan management system

## Activity diagram and Swimlane diagram

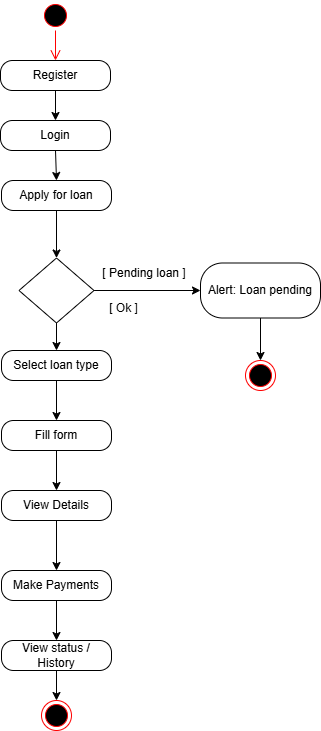


Figure ‑ Activity diagram for taking loan

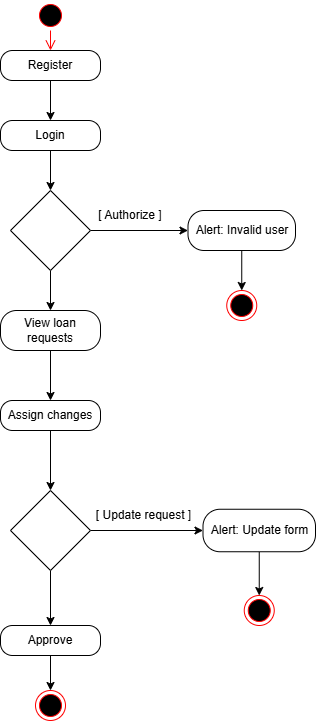


Figure ‑ Activity diagram for approving loan

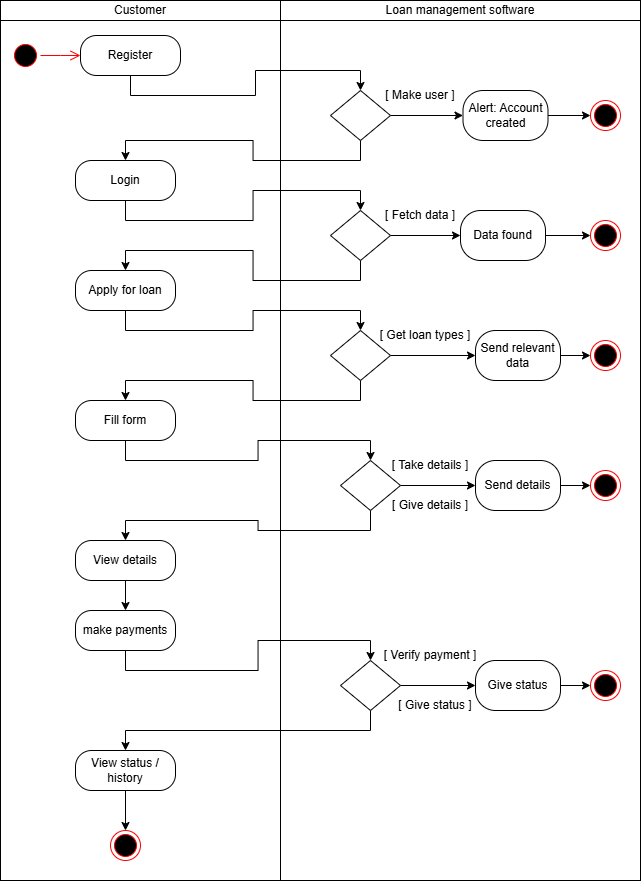


Figure ‑ Swimlane diagram for taking loan

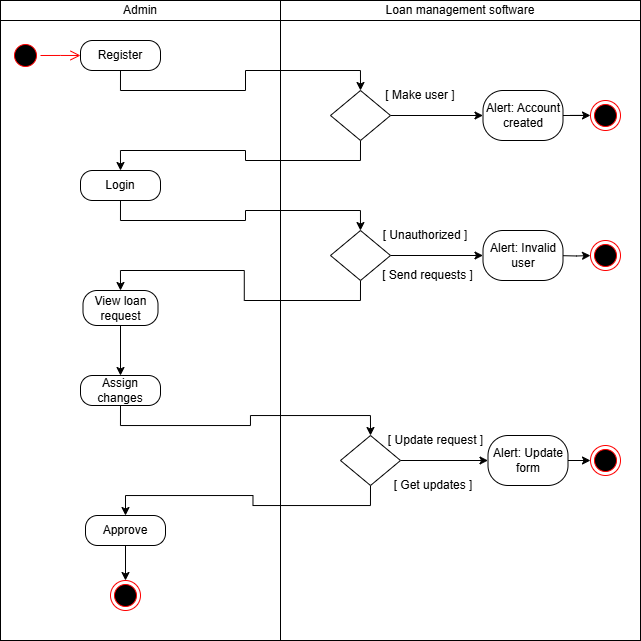


Figure ‑ Swimlane diagram for approving loan

## Sequence diagram

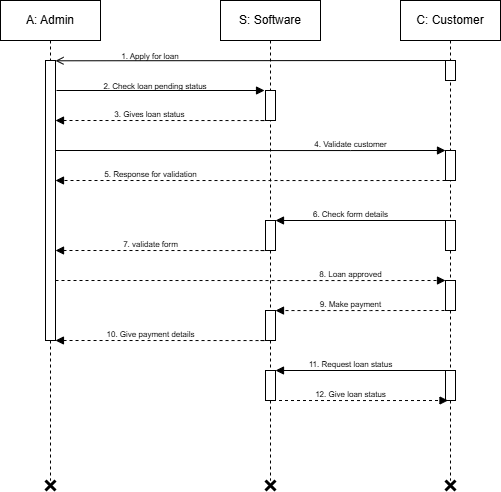


Figure ‑ Sequence diagram for taking loan

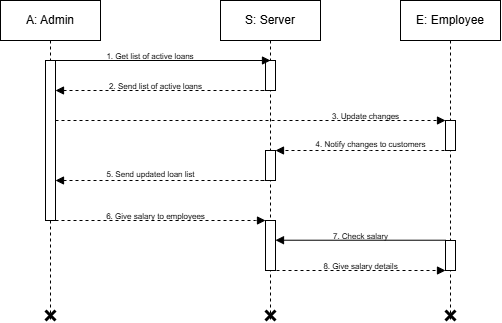


Figure ‑ Sequence diagram for admin user

## State diagram

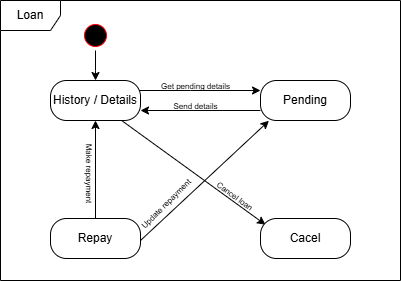


Figure ‑ State diagram for loan

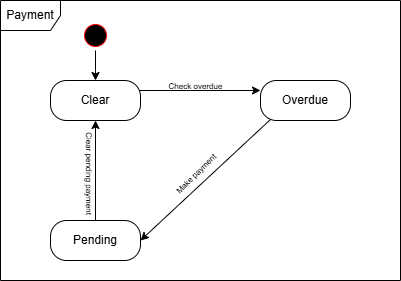


Figure ‑ State diagram for payment

## Class diagram



Figure ‑ Class diagram for Library management system

## Data flow diagram

### Context diagram (level-0)



Figure ‑ Context diagram for Library management system

### DFD Level-1



Figure ‑ DFD level-1 for Library management system

### DFD Level-2



Figure ‑ DFD level-2 for Issue book

# External interface requirement (Screens)

## Screen-1: Registration Form



Figure ‑ 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 ‑ 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 ‑ 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 ‑ 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 ‑ 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 ‑ 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

* Book
* Borrower
* Student
* Staff

Table ‑ 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 ‑ 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 ‑ 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 ‑ 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