

Darshan University

A Project Report on

**“Loan Management System”**

Under the subject

**Software Engineering (2101CS503)**

B. Tech, Semester – VI

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Jay Ramani | Enrollment No.: 22010101478 |
| Academic Year  (2023-2024) | |
| Internal Guide  Prof. Rajkumar 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** **(2101CS503)** 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 **Prof. Rajkumar Gondaliya** and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

Jay Ramani

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 **Jay Ramani** (22010101478) under my guidance in the fulfillment of the course **Software Engineering (2101CS503)** work during the academic year 2023-2024.

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

**Acknowledgement**

I wish to express my sincere gratitude to my project guide **Prof.** **Rajkumar Gondaliya** and all the faculty members for helping me through my project by giving me the necessary suggestions and advice 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

**Jay Ramani**

**ABSTRACT**

This project aims to develop a Loan Management System (LMS) to streamline and automate loan processing, management, and record-keeping for financial institutions. The proposed system will address the limitations of manual processes by establishing a centralized digital platform for loan applications, approvals, tracking repayments, and generating reports. The LMS will offer functionalities for both customer and administrator users. Customers can submit loan applications, access loan details, and schedule payments. Administrators will manage user accounts, loan products, interest rates, process applications, track delinquencies, and generate comprehensive reports for informed decision-making. This user-friendly and secure system will enhance operational efficiency, improve customer service, and minimize errors associated with manual loan management.

Develop an LMS to automate loan processes for financial institutions, boosting profitability through efficiency.

**Table of Contents**

[List of Figures I](#_Toc169968310)

[List of Tables II](#_Toc169968311)

[1 Introduction 1](#_Toc169968312)

[1.1 Product perspective 1](#_Toc169968313)

[1.2 Product features 1](#_Toc169968314)

[1.2.1 User Types 1](#_Toc169968315)

[1.2.2 Loan Applicant Features 1](#_Toc169968316)

[1.2.3 Loan Officer Features 1](#_Toc169968317)

[1.2.4 Loan Supervisor Features 1](#_Toc169968318)

[1.2.5 Admin Features 1](#_Toc169968319)

[1.3 Functional Requirement 2](#_Toc169968320)

[1.3.1 Librarian 2](#_Toc169968321)

[1.3.2 Member 2](#_Toc169968322)

[1.4 Non-Functional Requirement 2](#_Toc169968323)

[1.4.1 Usability: 2](#_Toc169968324)

[1.4.2 Accuracy: 2](#_Toc169968325)

[1.4.3 Availability: 2](#_Toc169968326)

[1.4.4 Maintainability: 2](#_Toc169968327)

[2 Design and Implementation Constraints 3](#_Toc169968328)

[2.1 Use case diagram 3](#_Toc169968329)

[2.2 Activity diagram and Swimlane diagram 4](#_Toc169968330)

[2.3 Sequence diagram 5](#_Toc169968331)

[2.4 State diagram 6](#_Toc169968332)

[2.5 Class diagram 7](#_Toc169968333)

[2.6 Data flow diagram 8](#_Toc169968334)

[2.6.1 Context diagram (level-0) 8](#_Toc169968335)

[2.6.2 DFD Level-1 8](#_Toc169968336)

[2.6.3 DFD Level-2 9](#_Toc169968337)

[3 External interface requirement (Screens) 10](#_Toc169968338)

[3.1 Screen-1: Registration Form 10](#_Toc169968339)

[3.2 Screen-2: Login Form 11](#_Toc169968340)

[3.3 Screen-3: Add borrower book 12](#_Toc169968341)

[4 Database design 13](#_Toc169968342)

[4.1 List of Tables 13](#_Toc169968343)

[5 Stories and Scenario 14](#_Toc169968344)

[5.1 Story-1: Add New Book in Library Catalogue 14](#_Toc169968345)

[5.1.1 Scenario# S1.1 14](#_Toc169968346)

[5.1.2 Scenario# S1.2 14](#_Toc169968347)

[5.1.3 Scenario# S1.3 14](#_Toc169968348)

[5.2 Story-2: Search Book 15](#_Toc169968349)

[5.3 Story-3: Manage due date for borrowed book 15](#_Toc169968350)

[5.4 Story-4: Renew book 15](#_Toc169968351)

[5.5 Story-5: Generate a report on book usage and availability 15](#_Toc169968352)

[6 Test cases 16](#_Toc169968353)

[7 References 20](#_Toc169968354)

# List of Figures

[Figure 2.1‑1 Use case diagram for library management system 3](#_Toc146706848)

[Figure 2.2‑1 Activity diagram for Book Issue 4](#_Toc146706849)

[Figure 2.2‑2 Swimlane diagram for Book Issue 4](#_Toc146706850)

[Figure 2.3‑1 Sequence diagram for Book Issue 5](#_Toc146706851)

[Figure 2.4‑1 State diagram of Book 6](#_Toc146706852)

[Figure 2.4‑2 State diagram for Librarian 6](#_Toc146706853)

[Figure 2.5‑1 Class diagram for Library management system 7](#_Toc146706854)

[Figure 2.6‑1 Context diagram for Library management system 8](#_Toc146706855)

[Figure 2.6‑2 DFD level-1 for Library management system 8](#_Toc146706856)

[Figure 2.6‑3 DFD level-2 for Issue book 9](#_Toc146706857)

[Figure 3.1‑1 Screen-1: Registration Form 10](#_Toc146706858)

[Figure 3.2‑1 Screen-2: Login Form 11](#_Toc146706859)

[Figure 3.3‑1 Screen-3: Add borrower book 12](#_Toc146706860)

# List of Tables

[Table 3.1‑1 Screen element of Registration form 10](#_Toc146706861)

[Table 3.2‑1 Screen element of Login form 11](#_Toc146706862)

[Table 3.3‑1 Screen element of Add borrower book 12](#_Toc146706863)

[Table 4.1‑1 Table: Book 13](#_Toc146706864)

[Table 4.1‑2 Table: Borrower 13](#_Toc146706865)

[Table 4.1‑3 Table: Staff 13](#_Toc146706866)

[Table 4.1‑4 Table: Student 13](#_Toc146706867)

# Introduction

## Product perspective

This project aims to develop a Loan Management System (LMS) to automate and centralize loan processing, management, and record-keeping for financial institutions. Replacing manual processes with a secure web application, the LMS improves efficiency, reduces errors, and enhances customer service. The system caters to four user roles: Loan Applicants, Loan Officers, Loan Supervisors, and Admin, streamlining the loan lifecycle from application to repayment.

## Product features

### User Types

* **Loan Applicant:**
  + Individuals applying for and managing loans.
* **Loan Officer:**
  + Processes loan applications, conducts initial assessments, and makes recommendations.
* **Loan Manager:**
  + Reviews and approves/rejects loan applications based on officer recommendations and established criteria.
* **Admin:**
  + Manages system configuration, user accounts, and security protocols.

### Loan Applicant Features

* Secure online loan application submission with document upload.
* Real-time application status tracking.
* Access to loan details, including interest rate, repayment schedule, and remaining balance (once a loan is approved).
* Online payment processing for convenient loan repayments (once a loan is approved).
* Secure communication portal for inquiries and support.

### Loan Officer Features

* User management for creating borrower profiles.
* Configurable loan product selection based on borrower eligibility.
* Access to credit bureau reports and other borrower data for initial assessment.
* Recommendation system for loan approval/rejection with supporting documentation.

### Loan Manager Features

* Review loan applications submitted by Loan Officers, including borrower information, financial details, and recommendations.
* Authority to approve or reject loan applications based on established criteria.
* Access to detailed decision reports for each loan application.

### Admin Features

* System configuration, including setting up loan products, interest rates, and user access controls.
* User management for creating and managing Loan Officer and Loan Supervisor accounts.
* Robust reporting and analytics for overall loan portfolio performance and risk management.
* Secure data storage and access control to ensure borrower privacy and regulatory compliance.

## Functional Requirement

### Loan Applicant Requirements

* **Login/Logout**: Applicants can log in and log out of the system.
* **Submit Application:** Electronically submit loan applications with clear instructions.
* **Upload Documents:** Attach required documents (proof of income, ID) for loan application.
* **Track Status:** Monitor application status in real-time for informed decisions.
* **View Loan Details:** Access loan details (interest rate, repayment schedule, remaining balance) upon approval.
* **Make Payments:** Initiate loan repayments online using various payment options (e.g., bank transfer, debit card).
* **Contact Loan Officer:** Securely communicate with Loan Officers for inquiries and support.
* **Update Profile:** Maintain profile accuracy by updating personal information within the system.
* **Explore Products:** View available loan products and eligibility criteria to choose the best fit.
* **Receive Updates:** Get notified regarding application status updates (e.g., approval, rejection, additional information needed).
* **Download Statements:** Download loan statements and repayment history for record-keeping.
* **Request Restructuring:** If needed, request loan restructuring or extension for unforeseen circumstances.
* **Submit Multiple Applications:** Submit multiple loan applications simultaneously (if applicable).
* **Set Up Auto-Pay:** Set up automatic loan repayments for convenience and timely payments.
* **Dispute Information:** Dispute loan information or fees if discrepancies are found.
* **Provide Feedback:** Help improve the system by providing feedback on functionality and user experience.

### Loan Officer Requirements

* **Review Applications:** Access borrower profiles and application details for comprehensive review.
* **Select Loan Product:** Choose appropriate loan products based on borrower eligibility and financial situation.
* **Verify Financials:** Access credit bureau reports and other borrower financial data for informed decision-making.
* **Recommend Action:** Utilize a recommendation system to suggest loan approval/rejection with supporting documentation.
* **Collaborate with Supervisor:** Collaborate securely with Loan Supervisors through the system for clarification or guidance.
* **Assign Applications:** Assign loan applications to other Loan Officers for review and workload distribution.
* **Generate Reports:** Generate customized reports on loan applications and borrower data for analysis.
* **Communicate with Borrower:** Initiate communication with borrowers regarding application status or missing information.
* **Update Contact Info:** Update borrowers contact information to ensure accurate communication.
* **Access Loan Terms:** Access pre-defined loan terms and conditions for different loan products.
* **Utilize Workflows:** Utilize automated workflows for efficient processing of loan applications.
* **Assess Creditworthiness:** Conduct basic creditworthiness assessments to gauge borrower risk.
* **Credit Check**: Loan officers can perform credit checks on applicants.
* **Customer Feedback Review**: Loan managers can review feedback provided by applicants.

### Loan Manager Requirements

* **Approve Applications:** Review loan applications submitted by Loan Officers.
* **Make Decisions:** Approve or reject loan applications based on established criteria.
* **Access Decision Reports:** Access detailed decision reports for each loan application.
* **Collaborate with Loan Officer:** Securely communicate with Loan Officers for clarification or guidance.
* **Escalate Applications:** Escalate complex loan applications to Admin for review.
* **Analyse Performance:** Access historical loan performance data for trend analysis.
* **View Reports:** View reports on overall loan portfolio performance and risk assessment.
* **Define Workflows:** Define and manage loan approval workflows and escalation procedures.
* **Analyse Officer Performance:** Analyse Loan Officer performance metrics and identify areas for improvement.
* **Control User Accounts:** Manage user accounts and access permissions for Loan Officers.
* **Conduct Audits:** Conduct periodic system audits and ensure compliance with regulations.
* **Generate Supervisor Reports:** Generate customized reports on Loan Supervisor activities and loan performance.
* **Request System Changes:** Initiate system configuration changes with Admin approval.

### Admin Requirements

* **Backups & Recovery:** Implement system backup and disaster recovery procedures.
* **Define User Roles**: Define user roles, permissions, and access levels within the system.
* **User Handling**: Admins can create, update, and delete user accounts.
* **Audit Logs**: Admins can access and manage system audit logs.
* **System Maintenance**: Admins can schedule and perform system maintenance tasks.
* **API**: Admins can manage API access and integrations with external systems.
* **Usage Analytics**: Admins can analyse system usage and generate insights for improvement.

## 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.

### **Security Audits**:

* Conduct system audits and ensure adherence to security protocols.

# Design and Implementation Constraints

## Use case diagram

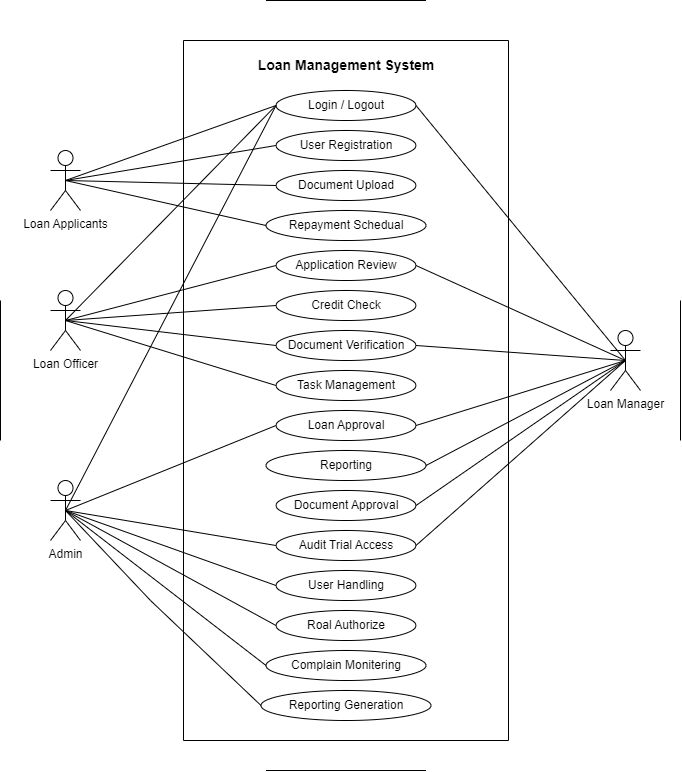
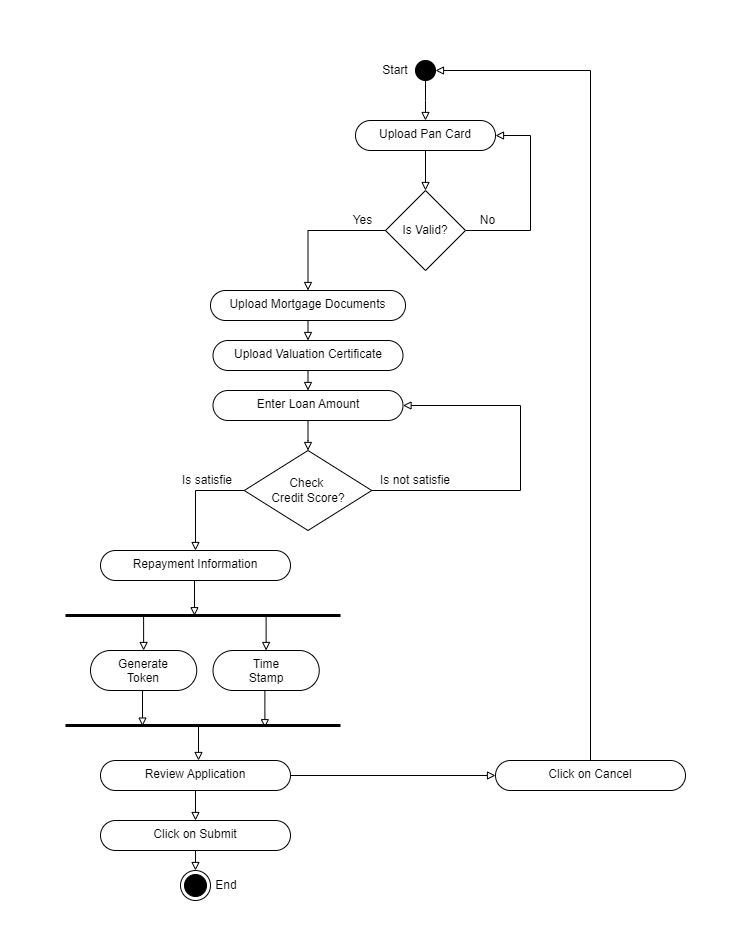
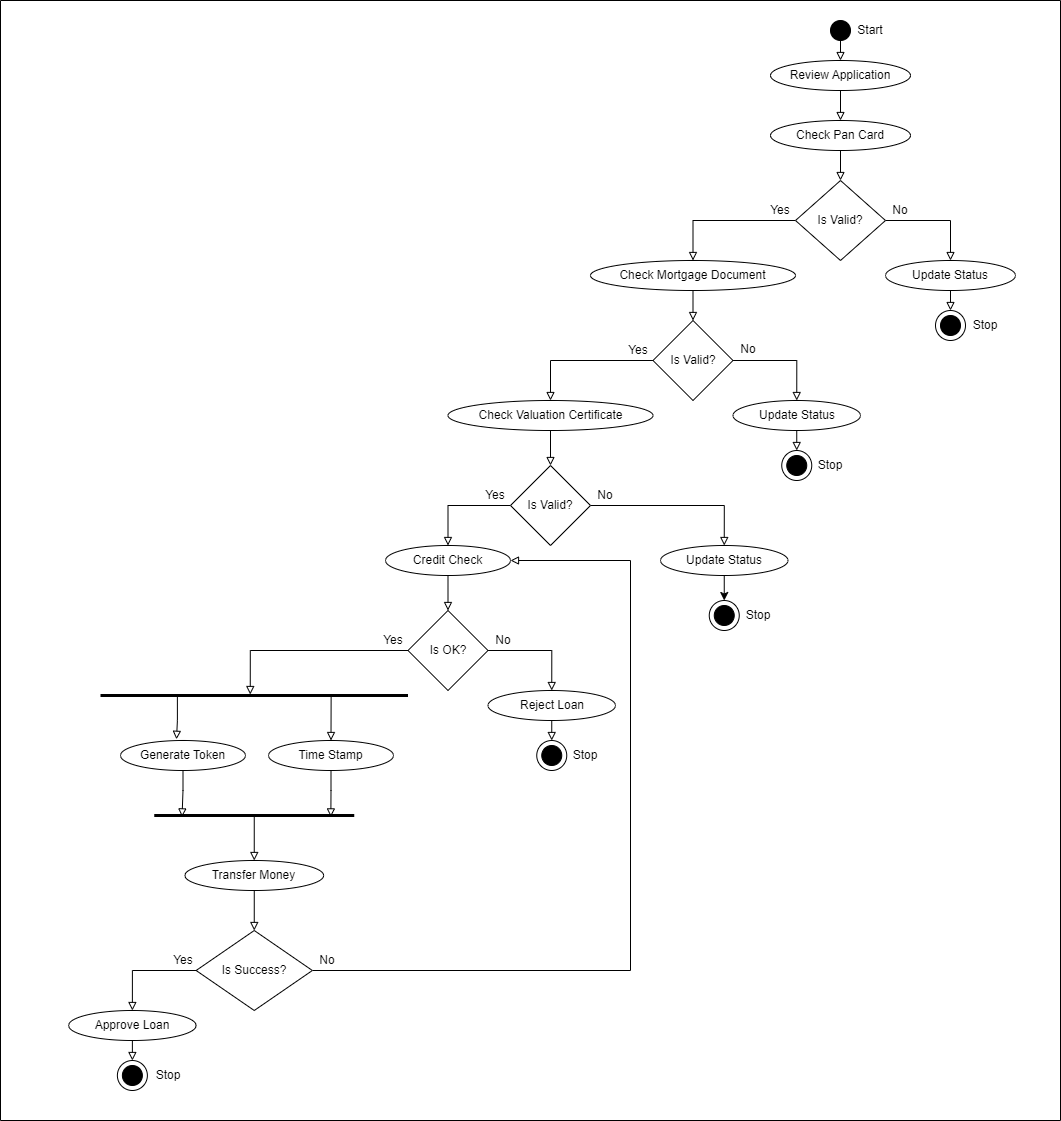


Figure 2.1‑1 Use case diagram for loan management system

**Activity Diagram for Loan Application**

****

**Activity Diagram for Loan Approval**



## Activity diagram and Swimlane diagram

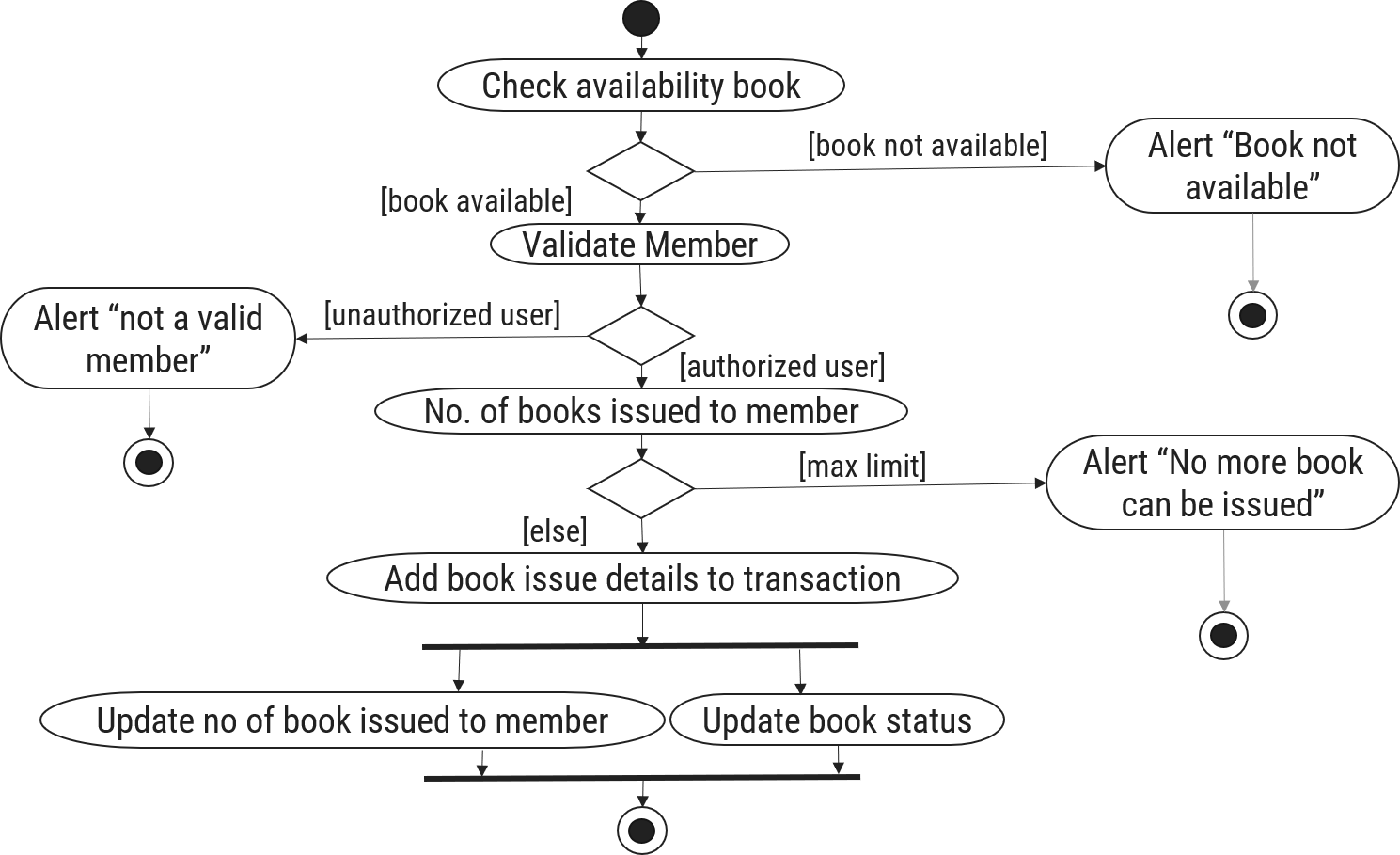


Figure 2.2‑1 Activity diagram for Book Issue

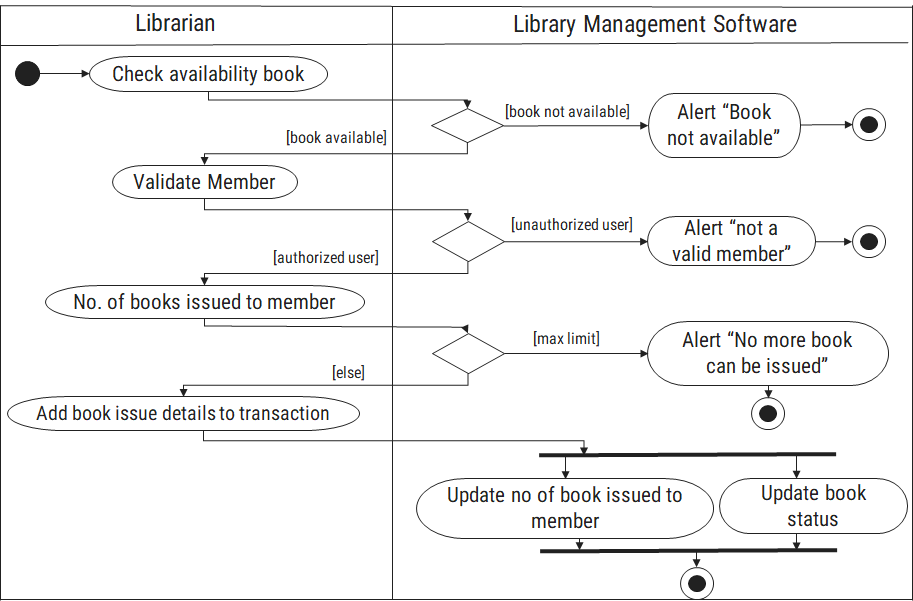


Figure 2.2‑2 Swimlane diagram for Book Issue

## 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

* 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