­



Darshan University

A Project Report on

**“Bank management system”**

Under the subject

**Software Engineering (2301CS405)**

B. Tech, Semester – IV

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Padsumbiya Darshan | Enrollment No.: 23010101181 |
| 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 “**Bank management system**” submitted in partial fulfilment for the Semester-5 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Department to 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.

Padsumbiya Darshan

Student’s Signature

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

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

**CERTIFICATE**

This is to certify that the SRS on “Bank Management System” has been satisfactorily prepared by Padsumbiya Darshan (23010101181) 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. Gondaliya and 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

**Padsumbiya Darshan**

**ABSTRACT**

This project is about the Banking System. The project has the complete information regarding the account details (including create, withdrawal, deposit, summary etc). It also contains the information about the different customers opening their account in the bank. This project also helps to keep the information about all the details of the various customers who have opened their account. This project is stand alone and the various center's of banking can implement connected through the network (only WAN) and only authenticated Person of the Bank can check the details of the account holder, he can also create saving, fix, current account of the customer. Due to centralized database only authenticated Person of The Bank can handle the application.

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

**Table of Contents**

[List of Figures I](#_Toc177666596)

[List of Tables II](#_Toc177666597)

[1 Introduction 1](#_Toc177666598)

[1.1 Product perspective 1](#_Toc177666599)

[1.2 Product features 1](#_Toc177666600)

[1.2.1 There are three different users who will be using this product: 1](#_Toc177666601)

[1.2.2 The features that are required for the Manager are: 1](#_Toc177666602)

[1.2.3 The features that are required for the Member are: 1](#_Toc177666603)

[1.3 Functional Requirement 1](#_Toc177666604)

[1.3.1 Customer 1](#_Toc177666605)

[1.3.2 Bank staff 2](#_Toc177666606)

[1.3.3 Manager 2](#_Toc177666607)

[1.3.4 Admin 2](#_Toc177666608)

[1.4 Non-Functional Requirement 2](#_Toc177666609)

[1.4.1 Usability: 2](#_Toc177666610)

[1.4.2 Accuracy: 3](#_Toc177666611)

[1.4.3 Availability: 3](#_Toc177666612)

[1.4.4 Maintainability: 3](#_Toc177666613)

[2 Design and Implementation Constraints 4](#_Toc177666614)

[2.1 Use case diagram 4](#_Toc177666615)

[2.2 Activity diagram and Swimlane diagram 5](#_Toc177666616)

[2.3 Sequence diagram 8](#_Toc177666617)

[2.4 State diagram 10](#_Toc177666618)

[2.5 Class diagram 10](#_Toc177666619)

[2.6 Data flow diagram 11](#_Toc177666620)

[2.6.1 Context diagram (level-0) 11](#_Toc177666621)

[2.6.2 DFD Level-1 12](#_Toc177666622)

[3 External interface requirement (Screens) 13](#_Toc177666623)

[3.1 Screen-1: Login Form 13](#_Toc177666624)

[3.2 Screen-2: Data change request 14](#_Toc177666625)

[3.3 Screen-3: Email verification 15](#_Toc177666626)

[3.4 Screen-4: Search lesson 16](#_Toc177666627)

[3.5 Screen-4: Add employee 17](#_Toc177666628)

[4 Database design 18](#_Toc177666629)

[4.1 List of Tables 18](#_Toc177666630)

[5 Stories and Scenario 19](#_Toc177666631)

[5.1 Story-1: Add New Book in Library Catalogue 19](#_Toc177666632)

[5.1.1 Scenario# S1.1 19](#_Toc177666633)

[5.1.2 Scenario# S1.2 19](#_Toc177666634)

[5.1.3 Scenario# S1.3 20](#_Toc177666635)

[5.2 Story-2: Search Book 20](#_Toc177666636)

[5.3 Story-3: Manage due date for borrowed book 20](#_Toc177666637)

[5.4 Story-4: Renew book 21](#_Toc177666638)

[5.5 Story-5: Generate a report on book usage and availability 21](#_Toc177666639)

[6 Test cases 22](#_Toc177666640)

[7 References 25](#_Toc177666641)

# List of Figures

[+Figure 2.1‑1 Use case diagram for Bank management system 4](#_Toc177730481)

[Figure 2.2‑1 Activity diagram for Invest in FDs 5](#_Toc177730482)

[Figure 2.2‑1 Activity diagram for deposit money 6](#_Toc177730483)

[Figure 2.2‑2 Swimlane diagram for Invest money in FDs 7](#_Toc177730484)

[Figure 2.2‑2 Swimlane diagram for deposit money 8](#_Toc177730485)

[Figure 2.3‑1 Sequence diagram Invest in FDs 9](#_Toc177730486)

[Figure 2.3‑2 Sequence diagram Account Details 10](#_Toc177730486)

[Figure 2.4‑1 State diagram of Customer 1](#_Toc177730487)1

[Figure 2.5‑1 Class diagram for Bank management system 1](#_Toc177730488)1

[Figure 2.6‑1 Context diagram for Library management system 1](#_Toc177730489)2

[gure 2.6‑2 DFD level-1 for Library management sys 1](#_Toc177730490)3

[Figure 3.1‑1 Screen-1: Registration Form 1](#_Toc177730491)4

[Figure 3.2‑1 Screen-2: Data change request form 1](#_Toc177730492)5

# 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 is basically updating the manual library system into an internet-based application so that the users can know the details of their account, availability of books etc. It is a multi-user version and can take care of all the fundamental functions of a Library like Cataloguing, Circulation, Accessioning and Housekeeping. It can satisfactorily cater to all the basic functions of a small library.

## Product features

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

* Customer who will see their account details.
* Member who will be accessing the account.
* Guest who will request for membership.

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

* Control the movement of books and other material and avoid losing the same.
* Search if you have a specific book in your collection based on the title, author etc.
* Print the spine labels for the book.
* Find what a specific person has borrowed from you. It offers the following modules Cataloguing, Circulation, and Queries.
* Can issue a book to the student.
* Can view the list of books available in each category.
* Can take the book returned from students.
* Add books and their information of the books to the database.
* Edit the information of the existing books.
* Can check the report of the issued Books.
* Can access all the accounts of the students.

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

* Can view the different categories of books available in the library.
* Can view the List of books available in each category.
* Can own an account in the library.
* Can view the books issued to him.
* Can put a request for a new book.

## Functional Requirement

### Customer

* See account details:Customers should see their account details.
* Apply for documents: Customer can apply for documents like statement, chequebook etc.
* Login: Customer should login by their A/C No, Mobile No or Email.
* Online payment:Customer should do online payments from system.
* Apply for card renew: Customer should apply for card renew.
* View statement: Customer should view statements of their account.
* Change personal details: Customer should change their personal details like phone No, email, address, etc.
* Apply for loans: Customer should view and apply for loans.
* Pay credit card bills: Customer should view credit card purchases and Pay bill in the system.
* Change passwords and pin: Customer should change net banking password and ATM pin online.
* Track cheque status: Customer should track cheque status by cheque No.
* Stop payment of cheque: Customer should request for stop payment of any given cheque.
* Invest money in FDs: Customer should view and invest in different FDs.
* Apply for lockers: Customer should apply for bank locker service
* Disable payment on card: Customer should disable payment service on card.
* Open new account: Customer should open new account.

### Bank staff

* Login: Staff members should login in system by their id’s.
* Access of account:Casher should deposit and credit amount of account.
* Apply for leave: Staff should apply for leave in system.
* View daily cash collection: Casher should view daily cash collection of particular branch.
* View loan application: loan department should view loan application and documents.
* Search customers: Staff should search customers by their name, AC No and mobile No
* View document applications: Staff members should view that Customers would apply for which documents
* Approve document application: Staff members should approve document application of customer.
* Approve loan application: Loan department should approve loan application.
* View locker application: Locker department should view locker applications of customers.
* View list of securities: Loan department should view list of securities was given by loan takers.
* Open new account: Staff member should open a new account of person.
* Close account: Staff member should remove existing account.

### Manager

* Login: Manager should login in system by their id’s.
* View leave applications: Managers should view leave application of their respective department.
* Grant leave: Manager should grant a leave of staff member.
* View daily collection: Manager should view daily cash collection of branches.
* View loan details: Managers should view given loan details and their document.
* View Details of ATMs: Managers should view details of ATMs of their respective branch area.
* View details of loan EMIs: Managers should view EMI details of loan taker. Weather they are paying EMIs on time or not.
* View details of credit card bills: Managers should view pending and paid bill details of credit cards.
* View details of FDs: Managers should view that how many customers are Inves
* ting in FDs.
* View performance of employee: Managers should view overall performance of their staff.

### Admin

* View performance branch: Admin should view details of branches in their region.
* Chek the cashflow: Admin should view overall cashflow of bank.
* View total assets: Admin should view total assets of the bank.
* View investment details: Admin should view investment details of the bank that bank are investing money in which bonds, schemes, stocks, etc.
* View performance of managers: Admin should view performance of branch manager.
* Promote manager: Admin should give promotion of the managers.

## 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:

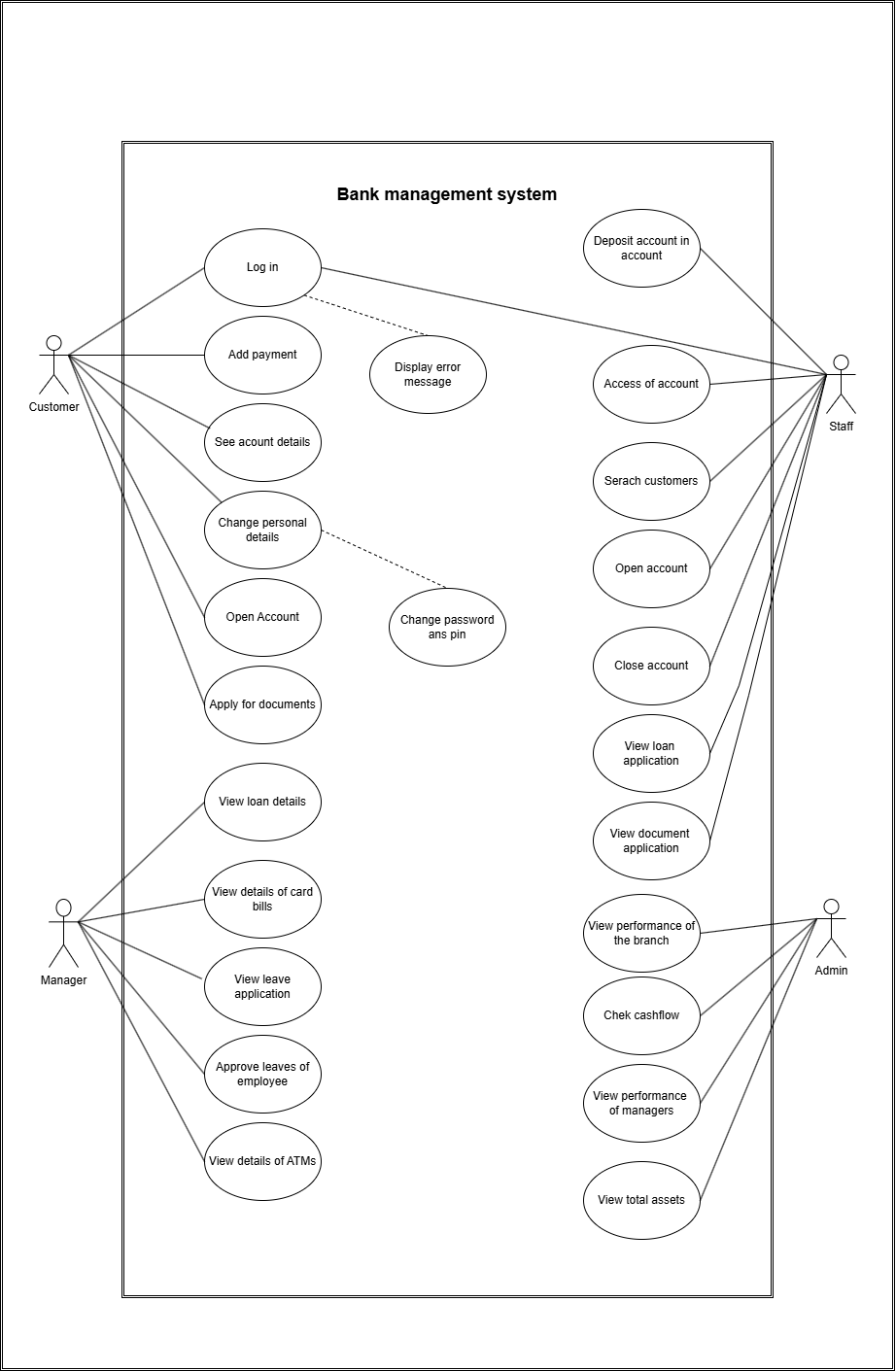
* The System should be available for the duration when the library operates and must be recovered within an hour or less if it fails. The system should respond to the requests within two seconds or less.

### 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 Bank management system

## Activity diagram and Swimlane diagram

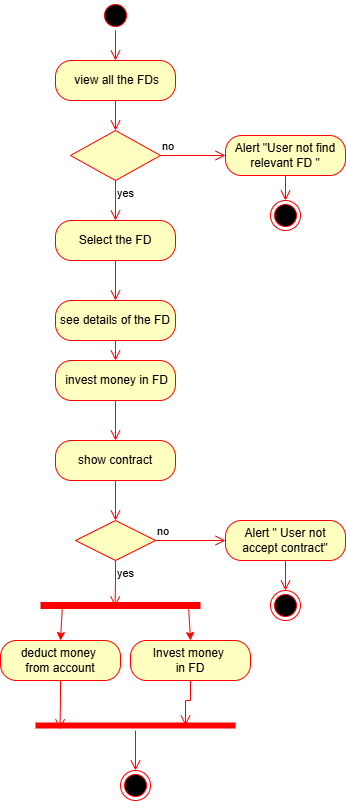


Figure 2.2‑1 Activity diagram for Invest in FDs

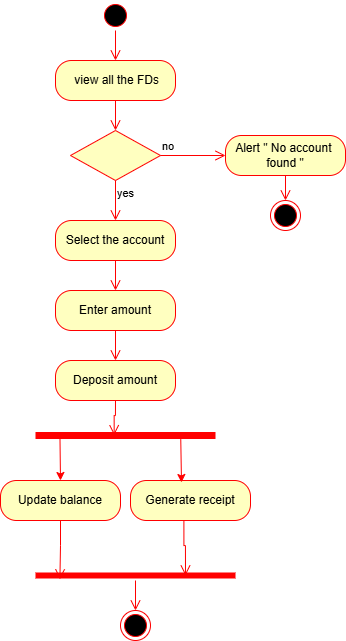


Figure 2.2‑1 Activity diagram for deposit money

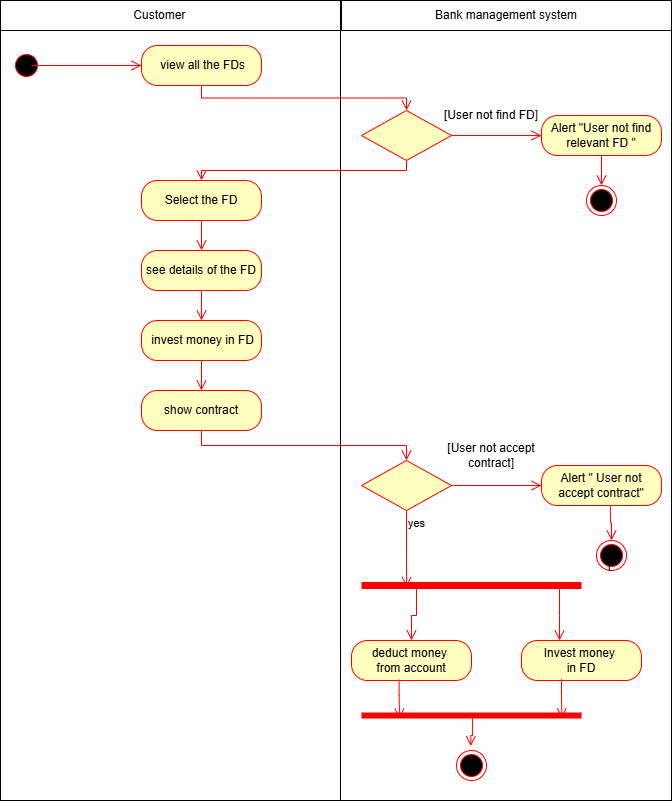


Figure 2.2‑2 Swimlane diagram for Invest money in FDs

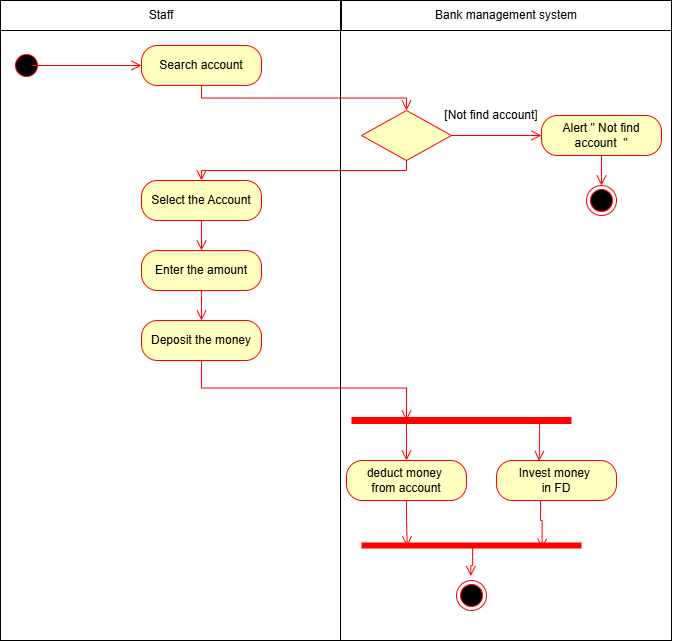


Figure 2.2‑2 Swimlane diagram for deposit money

## Sequence diagram

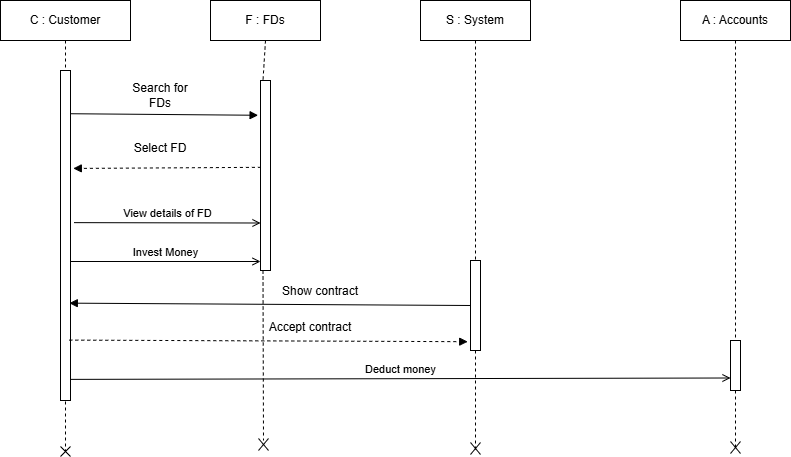
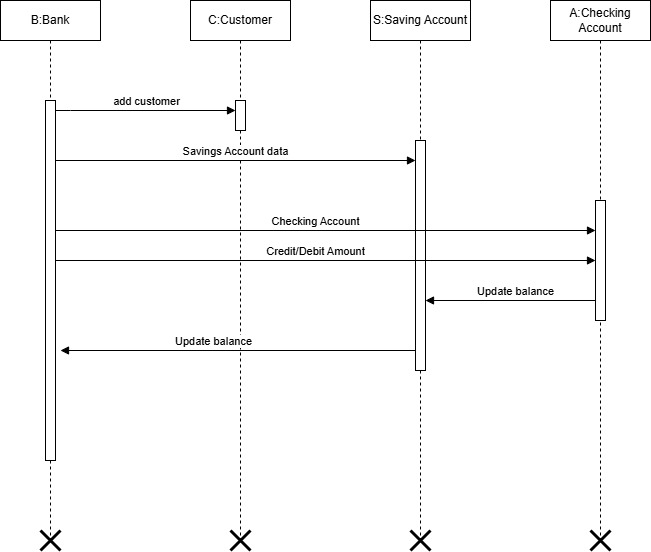


Figure 2.3‑1 Sequence diagram Invest in FDs

 Figure 2.3‑2 Sequence diagram Account Details

## State diagram

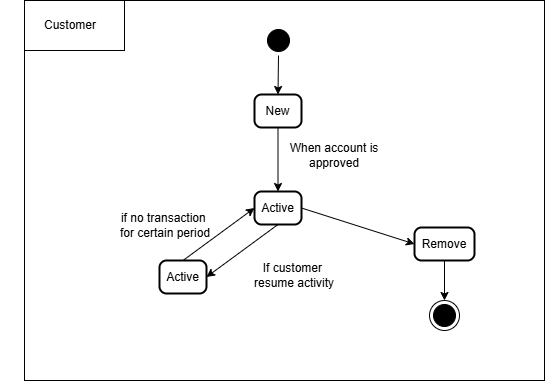


Figure 2.4‑1 State diagram of Customer

## Class diagram

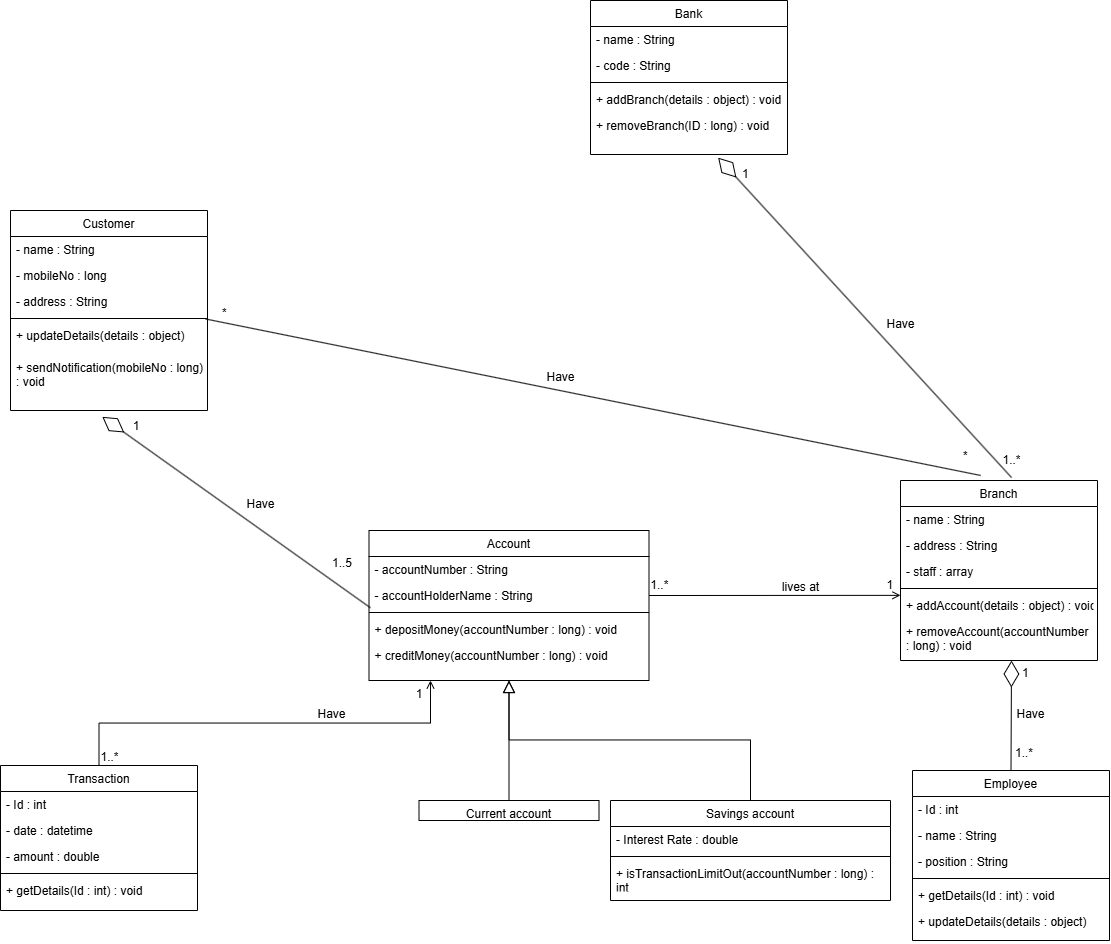


Figure 2.5‑1 Class diagram for Bank management system

## Data flow diagram

### Context diagram (level-0)

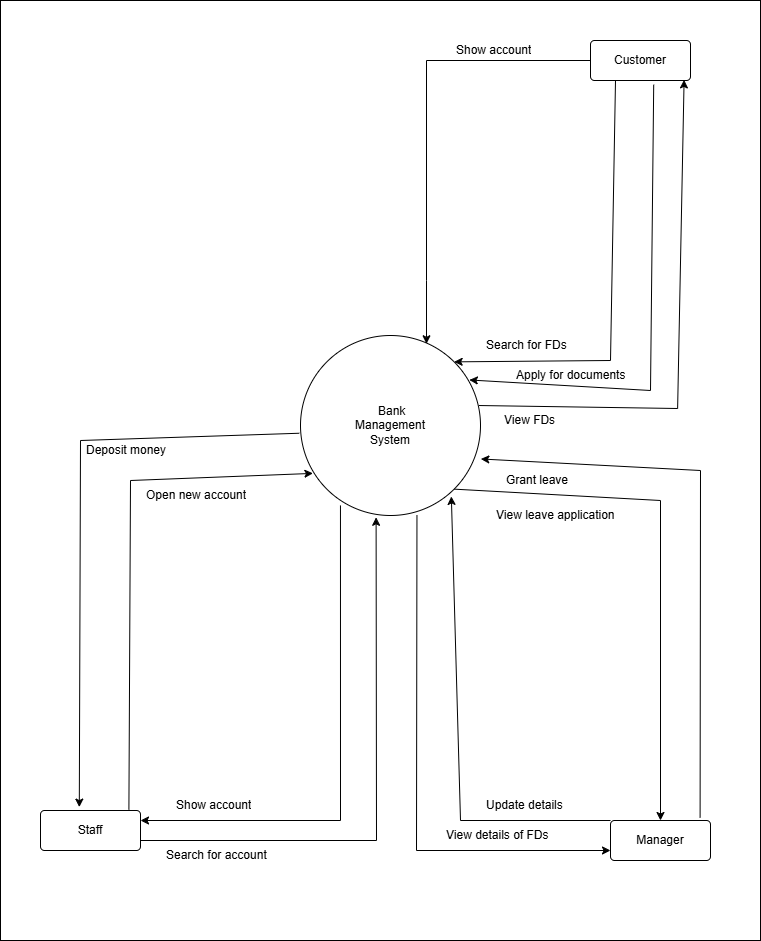
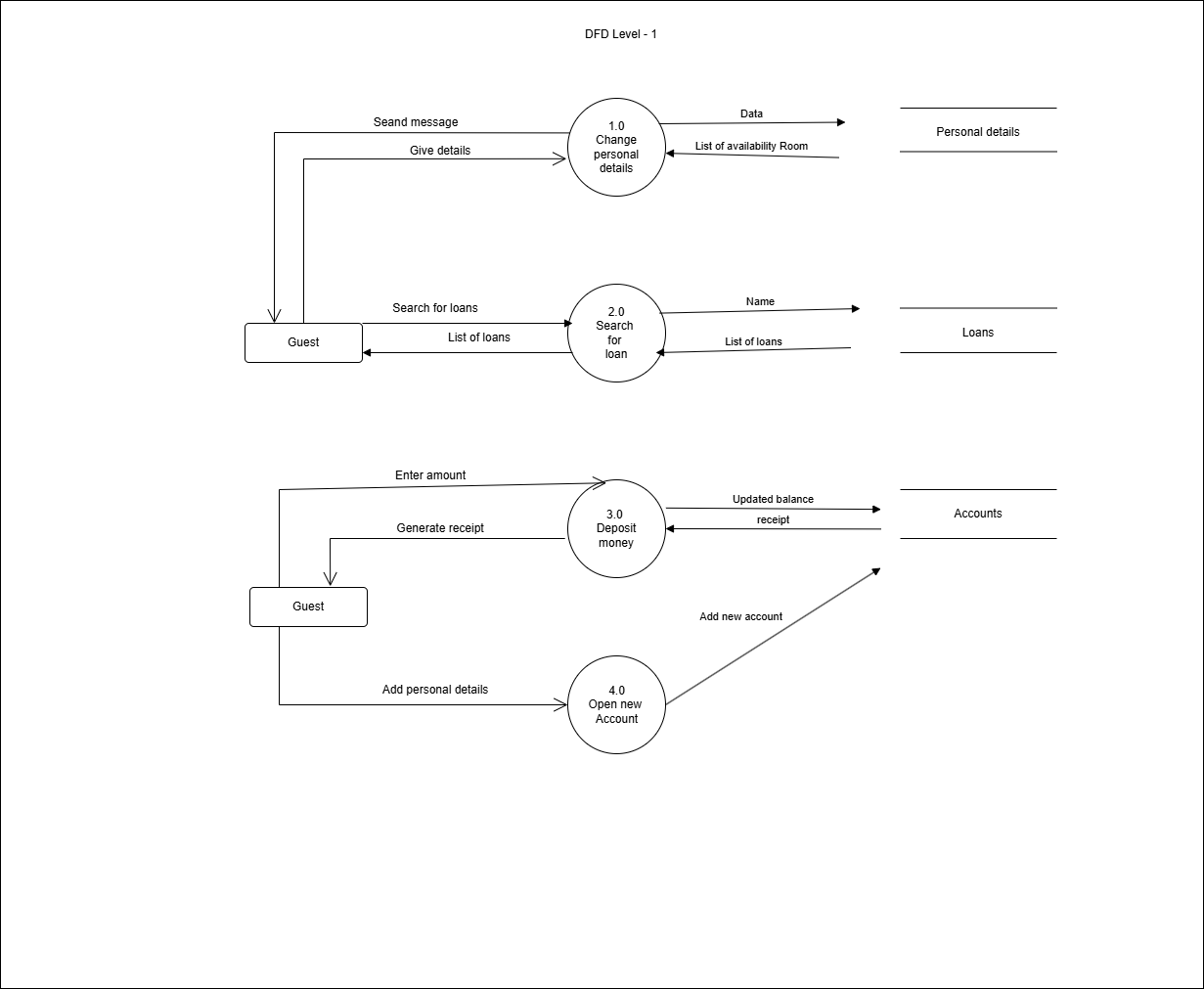


Figure 2.6‑1 Context diagram for Bank management system

### DFD Level-1



gure 2.6‑2 DFD level-1 for Library management sys

# External interface requirement (Screens)

## Screen-1: Login Form

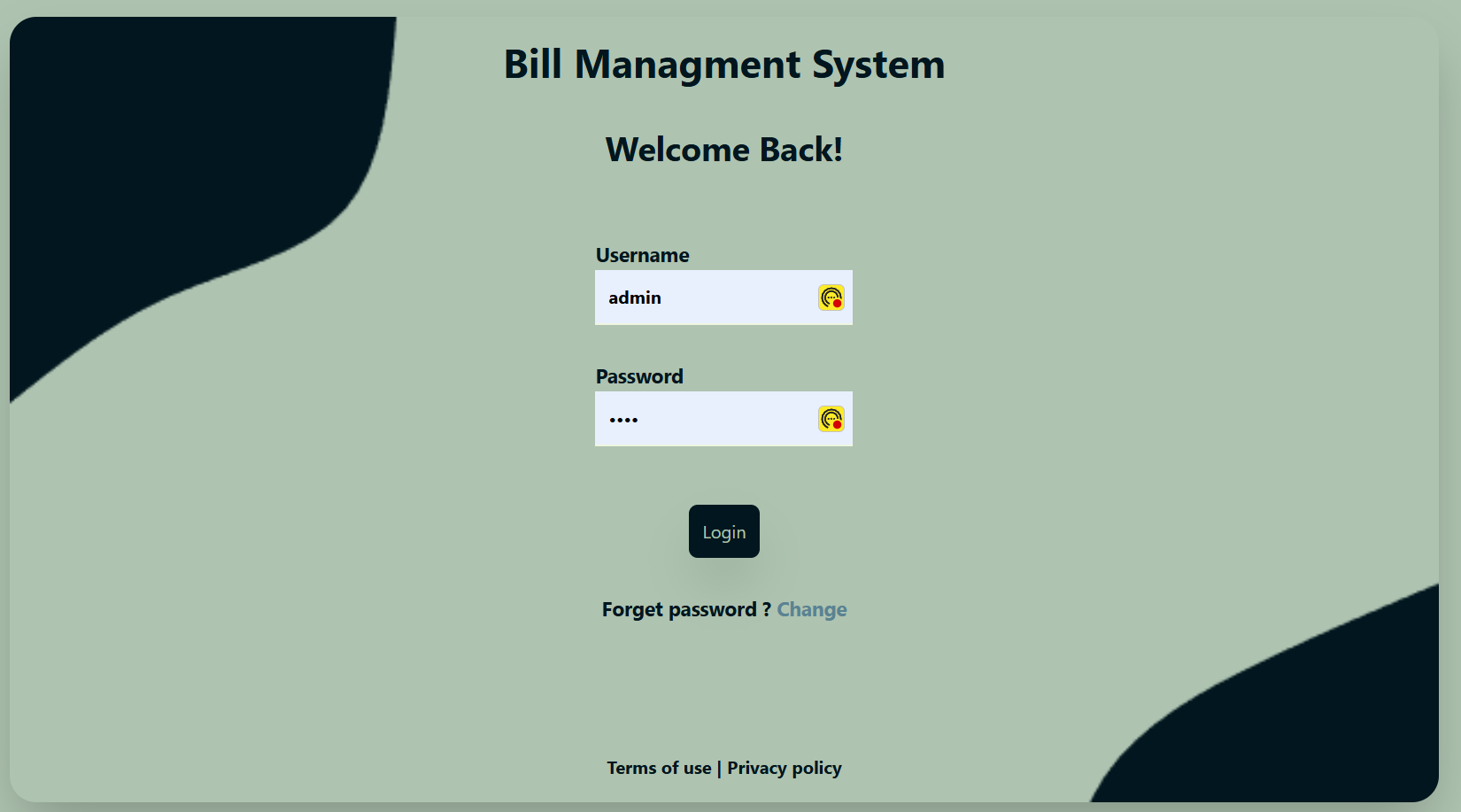


Figure 3.1‑1 Screen-1: Registration Form

**Purpose:** This form will allow the target end-users to login 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 | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| 3 | Terms of use | Link | ------ | ------ | Link to Terms of use page to read terms |
| 4 | Privacy policy | Link | M | 1 | Link to Privacy policy page to read Privacy policy |
| 5 | Change | Link | ------ | ------ | Link for forget password page to changepassword |
| 6 | Login | Button | ------ | ------ | Login is a button for store and check the entered data into database. |

## Screen-2: Data change request

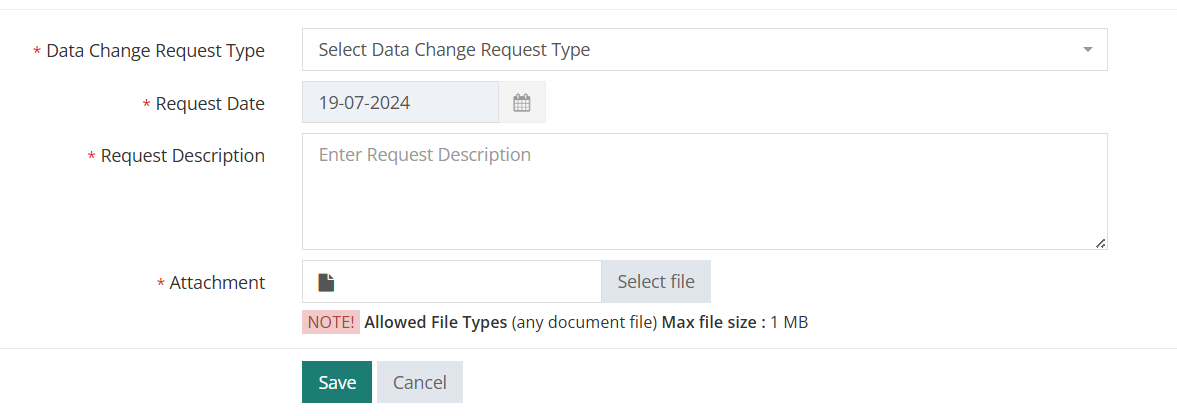


Figure 3.2‑1 Screen-2: Data change request form

**Purpose:** This form will be used by the students to make request to change the data. Add request type, date, description and attachments.

Table 3.2‑1 Screen element of Login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Data change request type | Selection List | M | 1 | Username field should be editable and accept the Username. |
| 2 | Request Date | Date | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| 3 | Request Description | Text Area | M | 1 | Saving login credentials through remember me checkbox |
| 4 | Attachment | File | M | 1 | Link for navigate to Forgot password page for allows users to recover password. |
| 5 | Save | Button | ------ | ------ | Save button will save form data in database and navigate to Home page |
| 6 | Cancel | Button | ------ | ------ | Cancel button will cancel the request and navigate to home page |

## Screen-3: Email verification

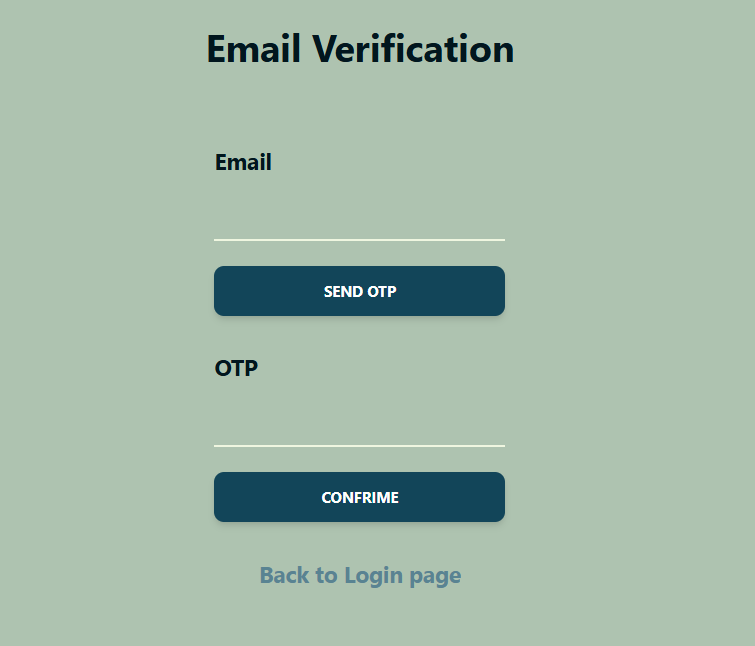


Table 3.3‑1 Screen element of Add borrower book

**Purpose:** This module will allow Customer to verify Email by OTP.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Email | Textbox | M | 1 | Email field should be editable and accept the Username. |
| 2 | OTP | Number | M | 1 | OTP field should be editable and accept the OTP and display as star or dot. |
| 3 | Send OTP | Button | ------ | ------ | Send OTP button will send the OTP in Email |
| 4 | Confirm | Button | ------ | ------ | Confirm Button will check the OTP and navigate to forget password page |
| 5 | Back to Login Page | Link | ------ | ------ | Link back toLogin page |

## Screen-4: Search lesson

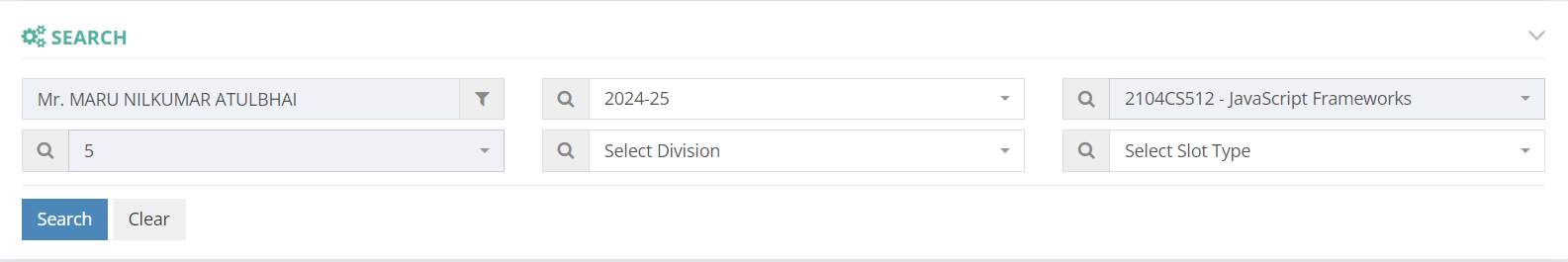


Table 3.3‑1 Screen element of Add borrower book

**Purpose:** This module will help faculties to search their lessons for particular division.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Name | Textbox | M | 1 | Name field will auto fill from login token |
| 2 | Academic year | Selection | M | 1 | OTP field should be editable and accept the OTP and display as star or dot. |
| 3 | Subject | Selection | M | 1 | Select the subject from the list |
| 4 | Semester | Selection | M | 1 | Select the semester from 1 to 8 |
| 5 | Division | Selection | M | 1 | Select the Division from the list |
| 6 | Slot | Selection | M | 1 | Select the slot from 1 to 4 |
| 7 | Search | Button | ------ | ------ | Search the lesson from database by following field and return. |
| 8 | Clear | Button | ------ | ------ | Clear button clear the selected items from all the fields |

## Screen-4: Add employee

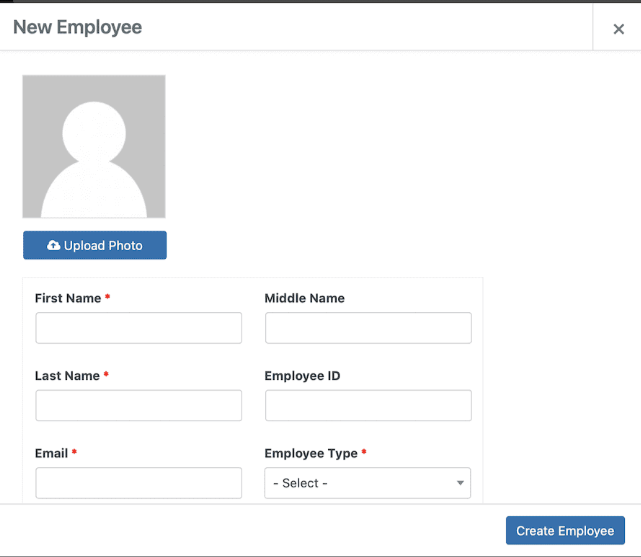


Table 3.3‑1 Screen element of Add borrower book

**Purpose:** This module will Add new User with photo and necessary fields.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | First Name | Textbox | M | 1 | First Name field should be editable and accept the First Name. |
| 2 | Middle Name | Textbox | O | 1 | Middle Name field should be editable and accept the Middle Name. |
| 3 | Last Name | Textbox | M | 1 | Last Name field should be editable and accept the Last Name. |
| 4 | Employee ID | Textbox | M | 1 | Employee ID field should be editable and accept the Employee ID. |
| 4 | Email | Textbox | M | 1 | Email field should be editable and accept the Email. |
| 5 | Employee Type | Selection | M | 1 | Select the type from the list |
| 6 | Upload photo | Button | ------ | ------ | Open folder to select the photo. |
| 7 | Create Employee | Button | ------ | ------ | Create Employee button will add employee details to databse |

# Database design

## List of Tables

* Admin
* Staff
* Customer
* Manager

Table 4.1‑1 Table: Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Null** | **Constraints** | **Description** |
| Admin\_ID | INT | NOT NULL | PRIMARY KEY | Unique Identifier for each admin. |
| Name | VARCHAR(255) | NOT NULL |  | Name of the admin. |
| Email | VARCHAR(255) | NOT NULL | UNIQUE | Email address of the admin. |
| Password | VARCHAR(255) | NOT NULL |  | Password for the admin account. |
| Additional\_Info | TEXT | NULL |  | Any additional information about the admin. |
| Contact\_Number | VARCHAR(20) | NULL |  | Contact number of the admin. |

Table 4.1‑2 Table: Staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Null** | **Constraints** | **Description** |
| Staff\_ID | INT | NOT NULL | PRIMARY KEY | Unique Identifier for each staff member. |
| Name | VARCHAR(255) | NOT NULL |  | Name of the staff member. |
| Email | VARCHAR(255) | NOT NULL | UNIQUE | Email address of the staff member. |
| Password | VARCHAR(255) | NOT NULL |  | Password for the staff account. |
| Position | VARCHAR(100) | NOT NULL |  | Job position of the staff member. |
| Contact\_Number | VARCHAR(20) | NULL |  | Contact number of the staff member. |

Table 4.1‑3 Table: Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Null** | **Constraints** | **Description** |
| Customer\_ID | INT | NOT NULL | PRIMARY KEY | Unique Identifier for each customer. |
| Name | VARCHAR(255) | NOT NULL |  | Name of the customer. |
| Email | VARCHAR(255) | NOT NULL | UNIQUE | Email address of the customer. |
| Password | VARCHAR(255) | NOT NULL |  | Password for the customer account. |
| Address | TEXT | NULL |  | Address of the customer. |
| Contact\_Number | VARCHAR(20) | NULL |  | Contact number of the customer. |
| Account\_Number | VARCHAR(20) | NOT NULL | UNIQUE | Bank account number of the customer. |
| Balance | DECIMAL(10,2) | NOT NULL |  | Current balance of the customer's account. |

Table 4.1‑4 Table: Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Null** | **Constraints** | **Description** |
| Manager\_ID | INT | NOT NULL | PRIMARY KEY | Unique Identifier for each manager. |
| Name | VARCHAR(255) | NOT NULL |  | Name of the manager. |
| Email | VARCHAR(255) | NOT NULL | UNIQUE | Email address of the manager. |
| Password | VARCHAR(255) | NOT NULL |  | Password for the manager account. |
| Department | VARCHAR(100) | NOT NULL |  | Department the manager oversees. |
| Contact\_Number | VARCHAR(20) | NULL |  | Contact number of the manager. |

# Stories and Scenario

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

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As aCustomer,  I want toview my account details  So thatI can track my balance and recent transactions easily.. |
| Priority | **:** | High |
| Estimate | **:** | S |
| Reason | **:** | This is a core functionality that allows customers to track their account status, including their balance and transactions. It needs to be quick, accessible, and reliable, as customers frequently need to view their account details for various banking activities. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Viewing Account Details with Valid Credentials |
| Prerequisite | **:** | The customer is registered and logged into the Bank Management System |
| Acceptance Criteria | **:** | **Given:** The customer is logged in to the system and navigates to the "Account Details" section.  **When:**  The customer selects the "View Account Details" option.  **Then**  The system should display the customer’s account number, balance, recent transactions, and other relevant details |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Attempting to View Account Details with Invalid Credentials |
| Prerequisite | **:** | The customer is not logged into the system or has entered invalid login credentials. |
| Acceptance Criteria | **:** | **Given:**The customer tries to access their account details without logging in or by using invalid credentials.  **When:** The customer enters incorrect login credentials or attempts to bypass login.  **Then t**he system should display an error message such as "Invalid username or password" and prevent access to account details. |

### Scenario# S1.3

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.3* | : | Viewing Account Details During System Maintenance |
| Prerequisite | **:** | The Bank Management System is undergoing scheduled maintenance |
| Acceptance Criteria | **:** | **Given**: The customer tries to access their account details during a period when the system is unavailable due to maintenance.  **When**: The customer logs in and attempts to view their account details. **Then**: The system should display a maintenance notification or a message like "Service is temporarily unavailable" and prevent access to account details.  **Then**: Generate unique book id, barcode and spine label for various book of same title. |

## Story-2: Search Book

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a Customer,  I want to apply for a loan online,  So thatI can submit my application without visiting the branch. |
| Priority | **:** | Medium |
| Estimate | **:** | M |
| Reason | **:** | Applying for loans online is an essential feature for customers, especially for convenience and accessibility. However, since this feature involves form filling and approval processes, it requires more development and testing efforts compared to simply viewing account details |

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

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As Customer,  I wantpay my credit card bill online  So thatI can clear my outstanding balance quickly and conveniently. |
| Priority | **:** | High |
| Estimate | **:** | L |
| Reason | **:** | Paying credit card bills is a critical feature that customers expect to work seamlessly. Any issues here could lead to customer dissatisfaction, financial penalties, or compliance risks. Due to the complex integrations with payment gateways and security considerations, this feature requires more time and effort to develop and test.. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Bank Management system | Test Designed by: | P. U. Jadeja |
| Module Name: | **Apply for loan** | **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 Bank Management System with valid credentials | Functional | Login to Bank management system web application through valid credential | TC\_001 |
| TC\_002 | Login to Bank Management System with invalid credentials | Functional | Login to Bank management system web application through invalid credential | TC\_002 |
| TC\_003 | Apply for a loan with valid data | Functional | Apply for loan in bank management system | TC\_003 |

|  |  |
| --- | --- |
| **Test Case Title** | Login to Bank Management System with valid credentials |
| **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://bankmanagementsystem.com/login |  |
| 2 | Enter valid Username in username field | Username field should be editable and accept the Username | Username input accepted | Pass |  | Username: johndoe@example.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: John@123 |  |
| 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 | Captcha from the image |  |
| 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 Bank Management System 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 | Error: "Invalid username or password" |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Apply for a loan with valid data |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | The user must be logged into the Bank Management System |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Navigate to the "Apply for Loan" section | The loan application form should be displayed | Form displayed | Pass |  |  |  |
| 2 | Enter valid loan amount, duration, and purpose | Fields should accept valid inputs and display them correctly | Inputs accepted | Pass |  | Loan amount: 100000 Duration: 12 months Purpose: Home renovation |  |
| 3 | Upload valid required documents (e.g., proof of income) | The system should accept the uploaded files | Documents uploaded | Pass |  | File: income\_proof.pdf |  |
| 4 | Click on the "Submit" button properly or not | The system should process the application and display a confirmation | Application submitted | Pass | A confirmation message should appear: "Your loan application has been submitted successfully." |  |  |

# References

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