

# ***COURSE NAME: PRINCIPALS OF SOFTWARE ENGINEERING***

***COURSE CODE: SWE1003***

***PROJECT NAME :BANK MANAGEMENT SYSTEM***

<b><i>#</i></b>	<b><i>Name</i></b>	<b><i>Registration number</i></b>
<b><i>1</i></b>	<b><i>Aitha. Tarun</i></b>	<b><i>18MIS7007</i></b>
<b><i>2</i></b>	<b><i>B.Pavan Kumar Reddy</i></b>	<b><i>18MIS7130</i></b>
<b><i>3</i></b>	<b><i>Anudeep</i></b>	<b><i>18MIS7103</i></b>
<b><i>4</i></b>	<b><i>Dhanush Teja</i></b>	<b><i>18MIS7138</i></b>

## ***OBJECTIVES***

***This project aims at creation of an online banking system. This will be accessible to all customers who have a valid User Id and Password. This is an approach to provide an opportunity to the customers to make transactions from where they are at present without visiting bank. In this project we are going to develop bank functions like transactions between customer and bank, applying loans, internet banking.***

## ***TASKS PERFORMED IN LAB***

- Do the Requirement Analysis and Prepare SRS.***
- Analyse the SRS to the Proposed Project.***
- Design System Flow.***
- Draw Use cases, E-R diagrams, DFD's for the Project.***

- *Develop any one module and test.*
- *Design of Test cases based on requirements and design.*
- *Post Implementation Review*
- *Future Enhancements.*

## **INDEX**

<b>#</b>	<b>Title</b>	<b>Page No</b>
<b>1</b>	<b>Abstract (Project Profile)</b>	<b>1</b>
<b>2</b>	<b>Existing System</b>	<b>4</b>
<b>3</b>	<b>Need for new System</b>	<b>4</b>
<b>4</b>	<b>Functional Specification</b>	<b>5</b>
<b>5</b>	<b>Hardware and Software Requirement</b>	<b>6</b>
<b>6</b>	<b>System Flow Chart</b>	<b>7</b>
<b>7</b>	<b>Timeline Chart</b>	<b>7</b>
<b>8</b>	<b>UML Diagrams</b>	<b>8-13</b>
	<b>9.1 Use Case Diagram</b>	<b>8</b>
	<b>9.2 Activity Diagram</b>	<b>9</b>
	<b>9.3 Sequence Diagram</b>	<b>10</b>

	<b>9.4 Deployment Diagram</b>	<b>11</b>
	<b>9.5 Class Diagram</b>	<b>11</b>
	<b>9.6 Component Diagram</b>	<b>12</b>
	<b>9.7 State chart Diagram</b>	<b>12-13</b>
<b>9</b>	<b>Data Dictionary</b>	<b>13-15</b>
<b>10</b>	<b>Input Design</b>	<b>16-20</b>
<b>11</b>	<b>Output Design</b>	<b>20-21</b>
<b>12</b>	<b>Testing</b>	<b>22</b>
<b>13</b>	<b>Post Implementation Review</b>	<b>23</b>
<b>14</b>	<b>Bibliography</b>	<b>23</b>

## **PROJECT PROFILE**

<b>Project name</b>	<b>Bank management system</b>
<b>Objective</b>	<b>To create a desktop bank application which is useful for customers manage their bank account in a portable way</b>
<b>Platform</b>	<b>Desktop application</b>
<b>Front End</b>	<b>JAVA</b>
<b>Back End</b>	<b>JAVAFX, Microsoft SQL Server</b>
<b>Other tools</b>	<b>Scene builder, Microsoft office, E Draw max</b>
<b>Project duration</b>	<b>90 days</b>

## **EXISTING SYSTEM**

### **Infosys Finacle:**

**Finacle by Edge Verve Systems Limited, a fully-owned subsidiary of Infosys, is one of the top software's used in the banking and finance sector. Finacle is used by a cumulative customer base of about 848 million across 94 countries globally.**

**It is a financial banking software that assists several technological requirements of a bank such as core-banking, online banking, mobile banking, payments, treasury, origination, liquidity management and wealth management.**

### **Oracle Financial Services:**

**Oracle Financial Services Software Limited is an IT solution provider exclusively for the banking and finance industry. It has approximately 900 customers in over 145 countries. The banking software suite provided by Oracle covers retail, corporate and investment banking, fund, cash and wealth management. It's universal banking suite, analytical applications for customer insight, analytical applications for financial crime and compliance management are software suites that taking banking services to a completely different level.**

## **NEED OF NEW SYSTEM**

- **Portability.**
- **Better user interaction.**
- **Improved security.**

## **FUNCTIONAL SPECIFICATIONS**

### **USER SPECIFICATION:**

- ❖ **Manager:** Create new customers, create new manager, approve new customers, update customer account, delete customer account, approving loans.
- ❖ **Customer:** Deposit money, withdraw money, checking balance in account, transfer money, checking transaction history, applying for new loan, checking loan status, checking loan details.

### **MODEULE SPECIFICATION:**

- ❖ **Create new customers:** Adding new customer account to the bank database.
- ❖ **Create new manager:** Adding new manager account to the bank database.
- ❖ **Approve new customers:** Accepting or declining customer based on their details.
- ❖ **Update customer account:** Changing customer details.
- ❖ **Delete customer account:** Removing customer account from bank database.
- ❖ **Approving loans:** Accepting or declining loan request based on their details.
  
- ❖ **Deposit money:** Depositing money into account.
- ❖ **Withdraw money:** Withdrawing money from account.
- ❖ **Checking balance:** Checking amount of money present in account.
- ❖ **Transfer money:** Transfer money from one account to another which were present in the bank database.
- ❖ **Checking transaction history:** Checking user transactions.
- ❖ **Applying new loan:** Applying for new loan.
- ❖ **Checking loan status:** Checking loan status.
- ❖ **Checking loan details:** Checking loan details.

## **HARDWARE REQUIREMENT**

### **CLIENT SIDE:**

- ✓ **Ram:8 GB.**
- ✓ **Free space:10 MB.**
- ✓ **Monitor resolution:1920x1080.**
- ✓ **Processor speed:2.30 GHz.**

### **SERVER SIDE:**

- ✓ **Ram:4 GB.**
- ✓ **Free space:100 MB.**
- ✓ **Processor speed:2.30 GHz.**

## **SOFTWARE REQUIREMENT**

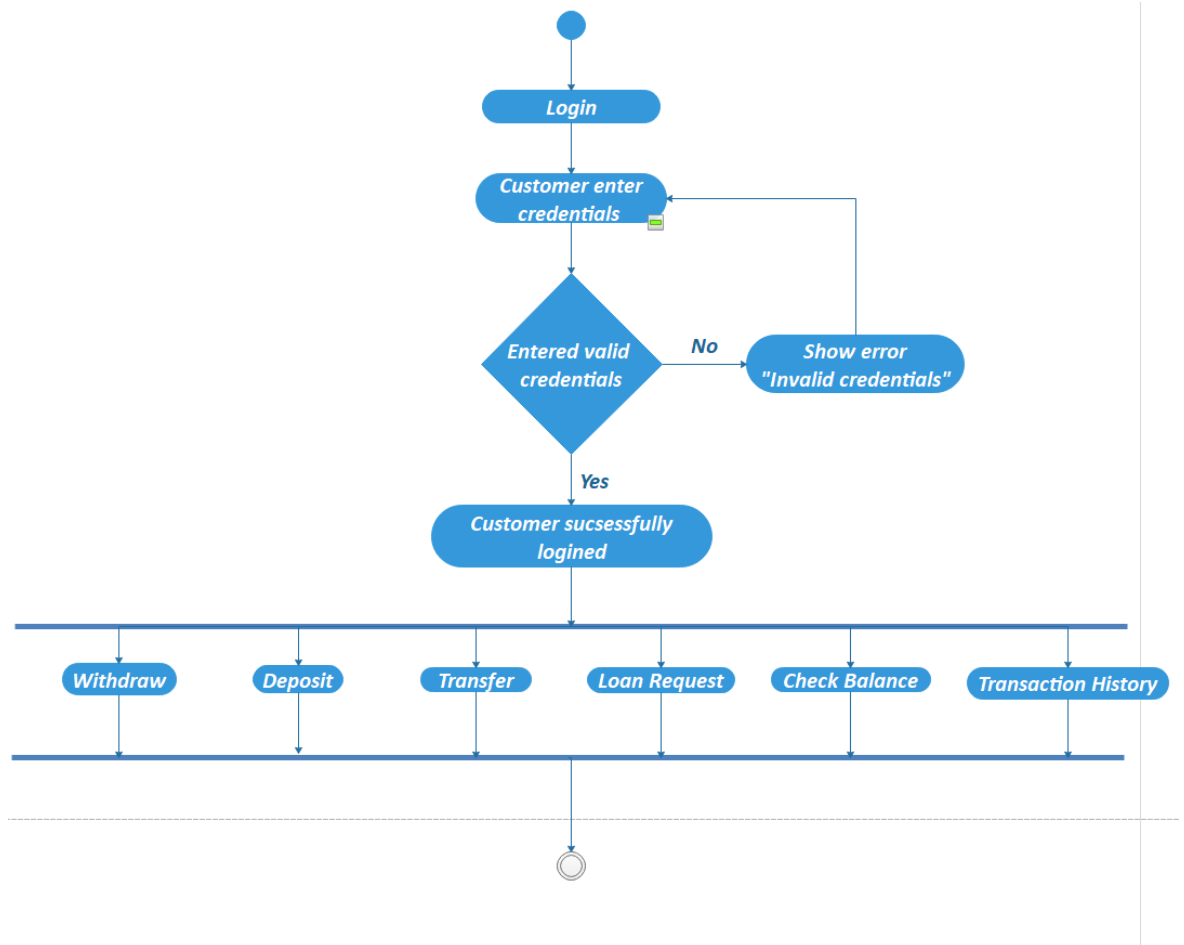
### **CLIENT SIDE:**

- ✓ **OS: Windows 10(any edition).**

### **SERVER SIDE:**

- ✓ **OS: Windows 10(any edition).**
- ✓ **Database server: Microsoft SQL server studio.**

## SYSTEM FLOWCHART

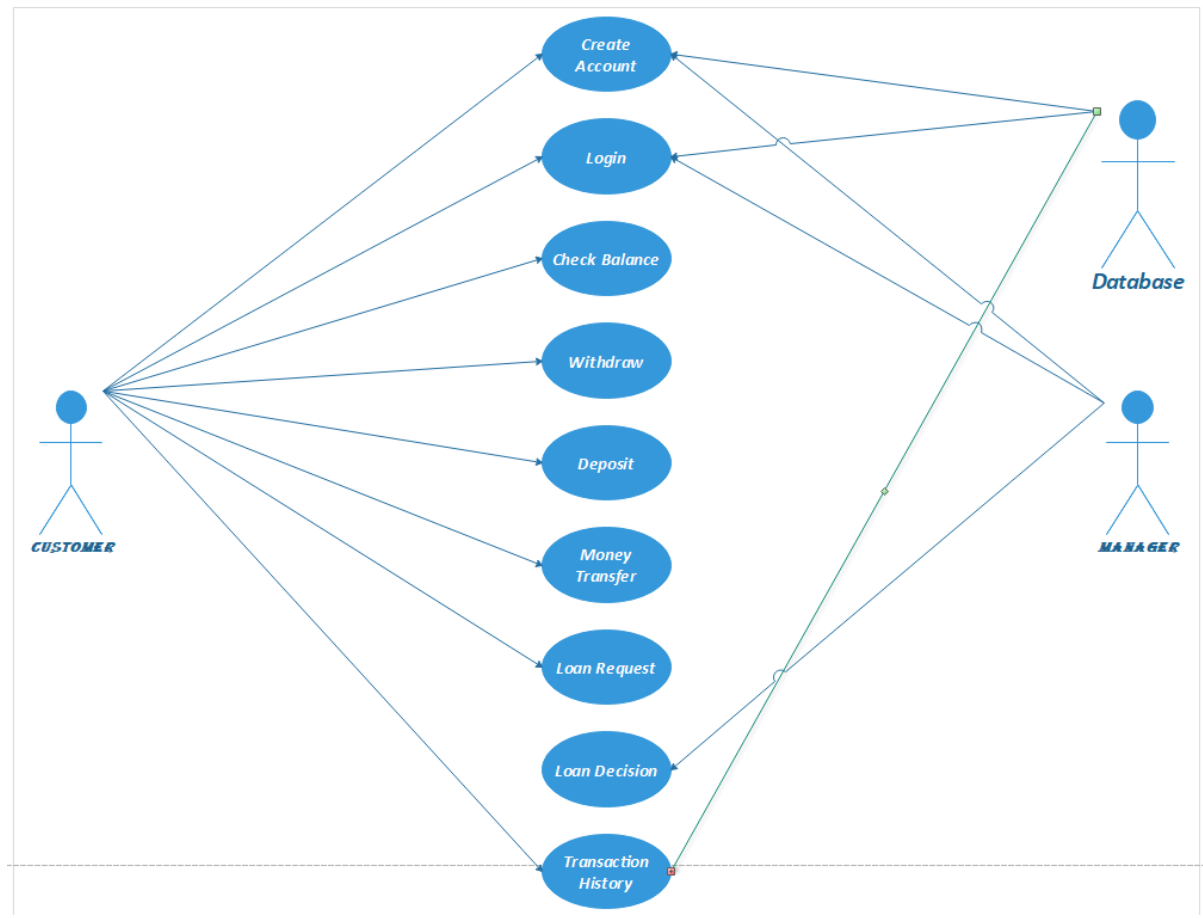


## TIMELINE CHART

Development Phase	90 Days						Duration (Day)
	0 to 15 Day	16 to 30 Day	31 to 45 Day	46 to 60 Day	61 to 75 Day	76 to 90 Day	
Requirement Gathering							10
Analysis							15
Design							30
Coding							25
Testing							12
Implementation							08
Documentation							80
Total Time (Days)							90

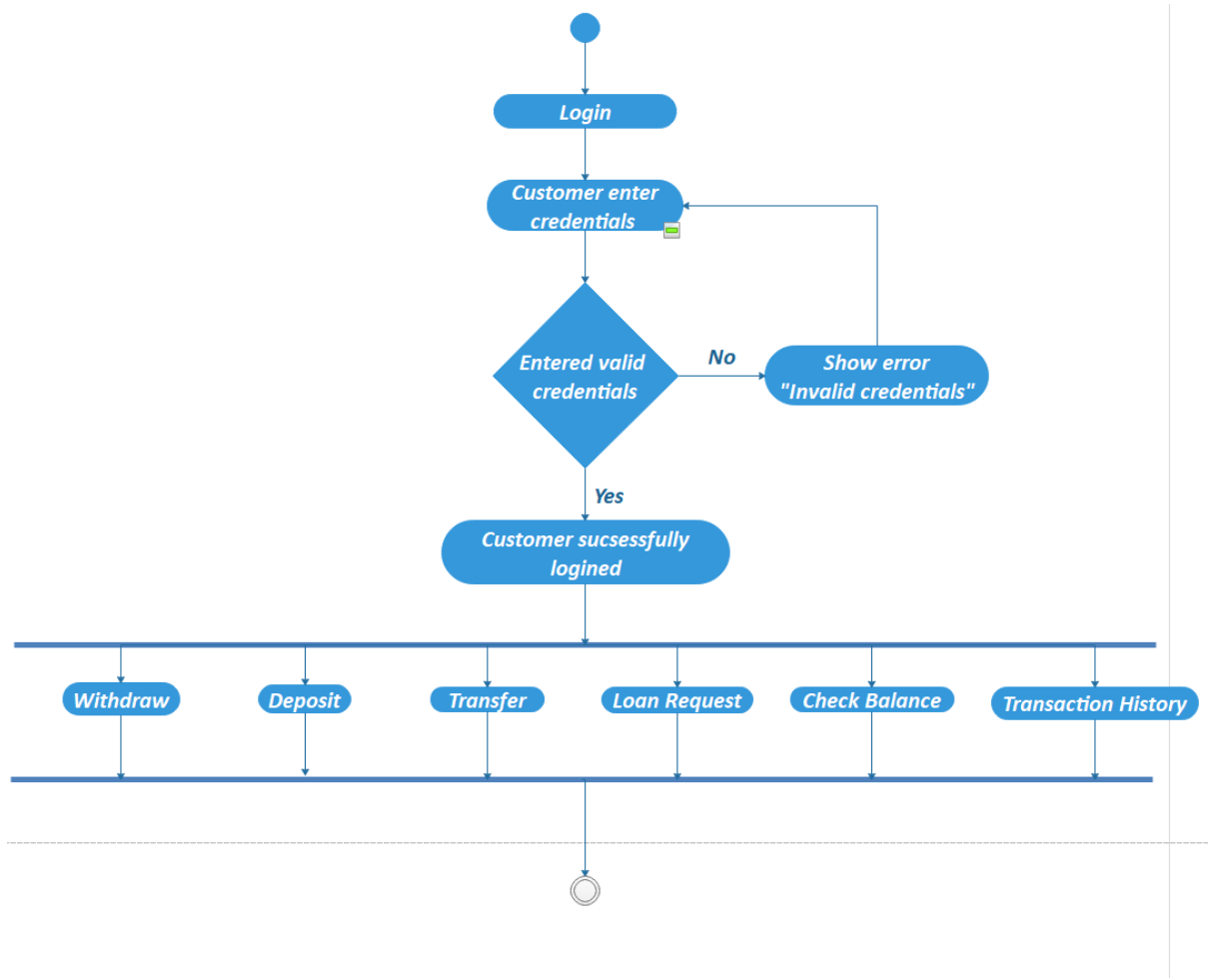
## UML DIAGRAMS

### USE CASE:

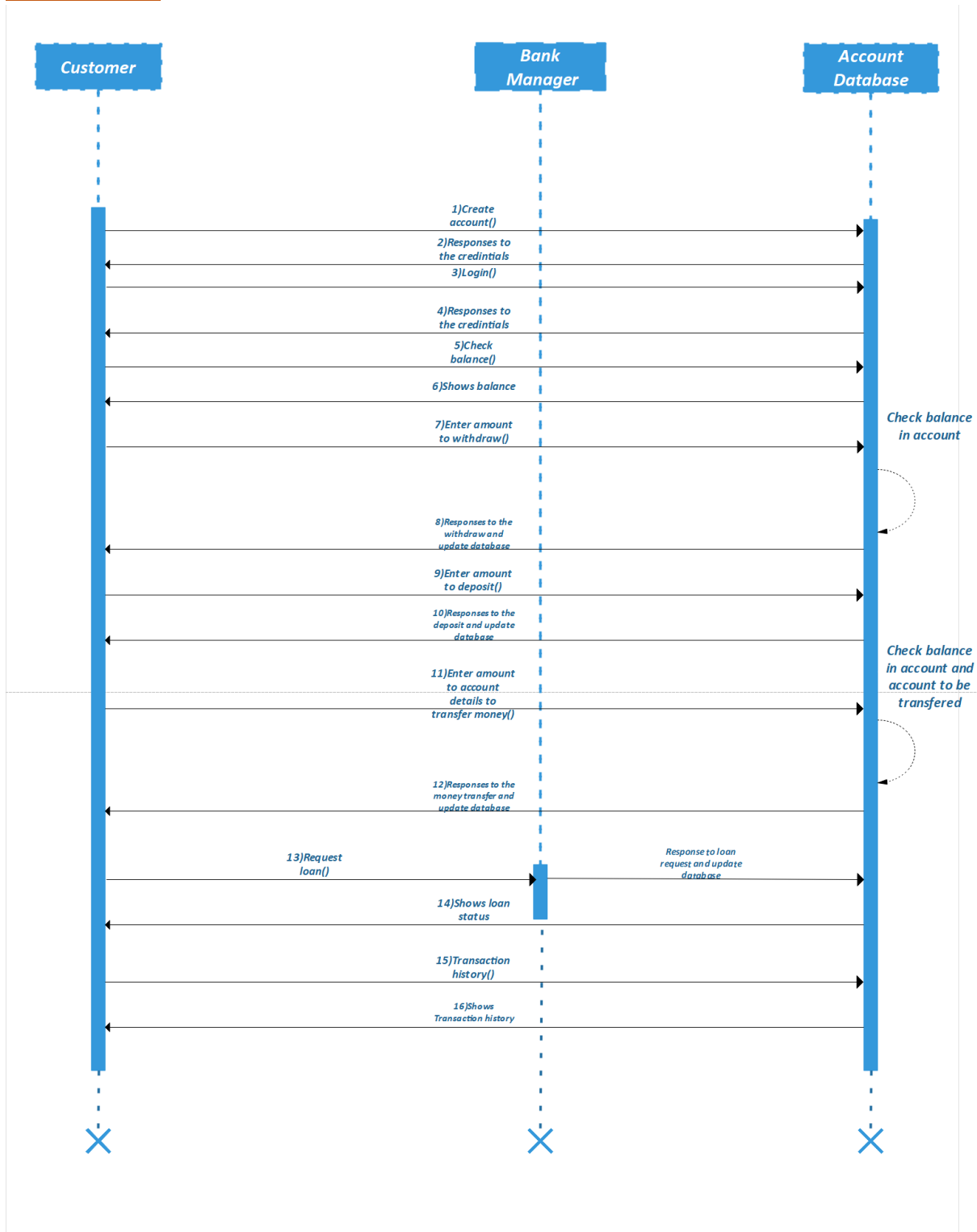




## ACTIVITY:



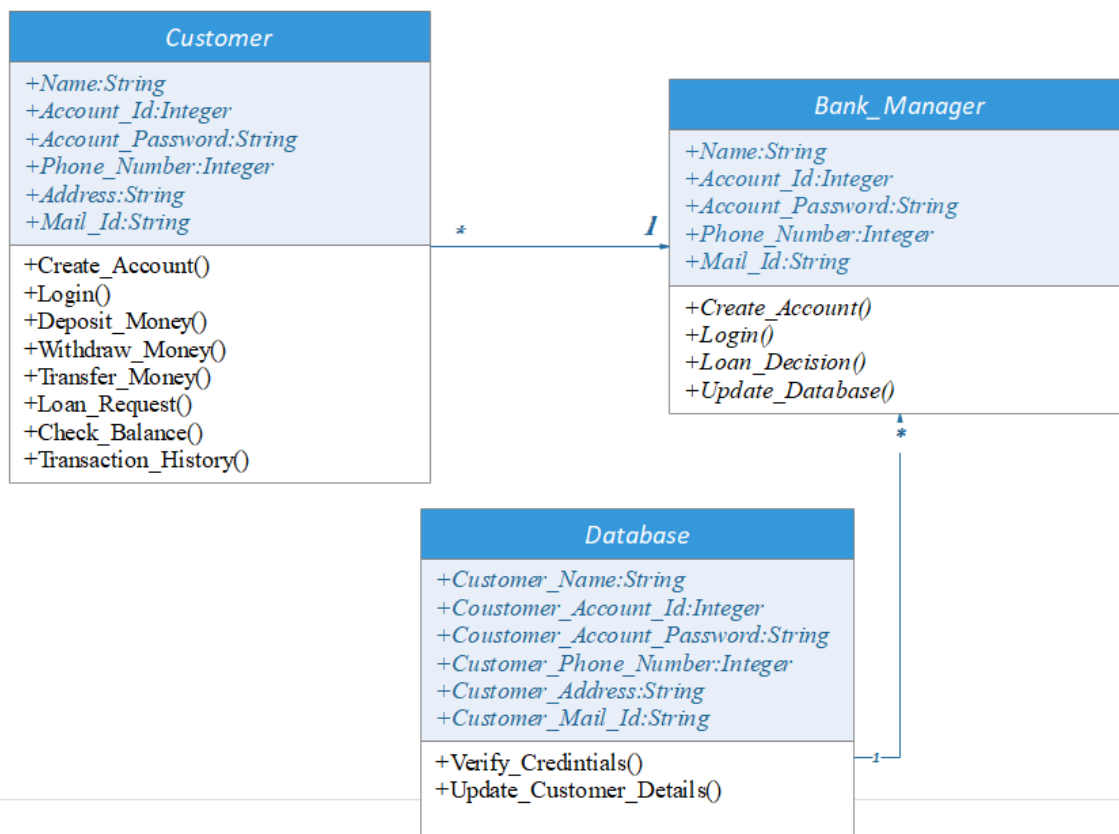
## SEQUENCE:



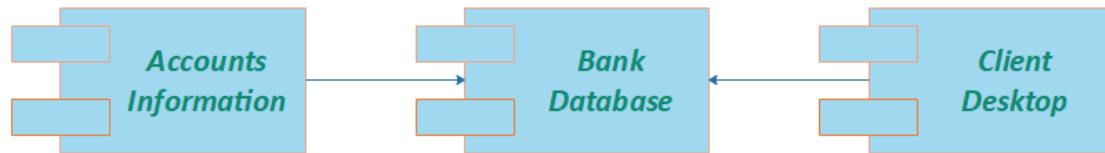
## DEPLOYMENT:



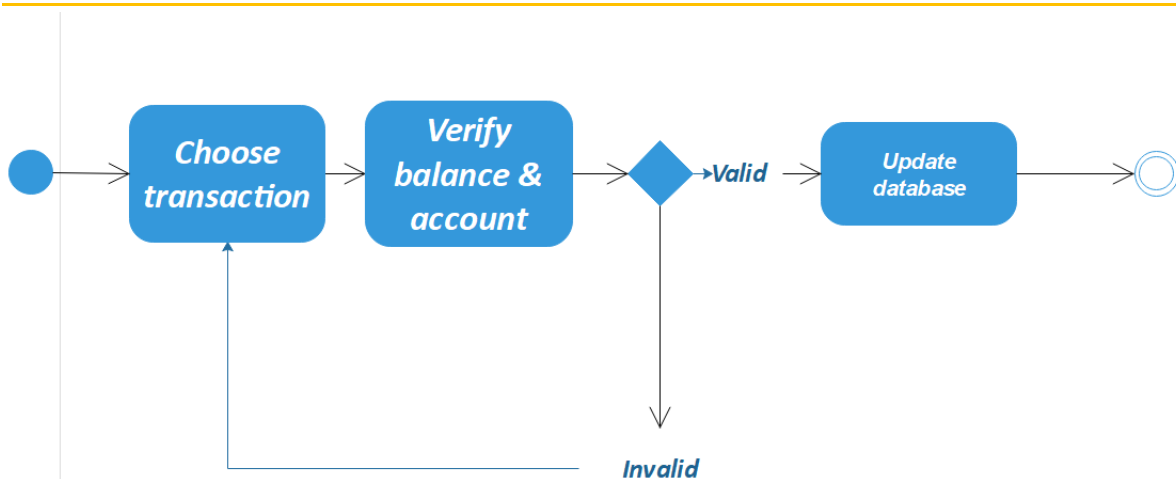
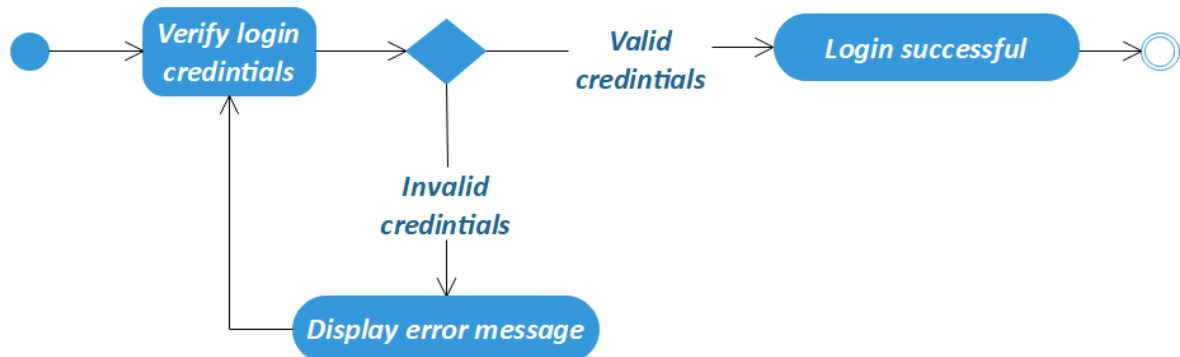
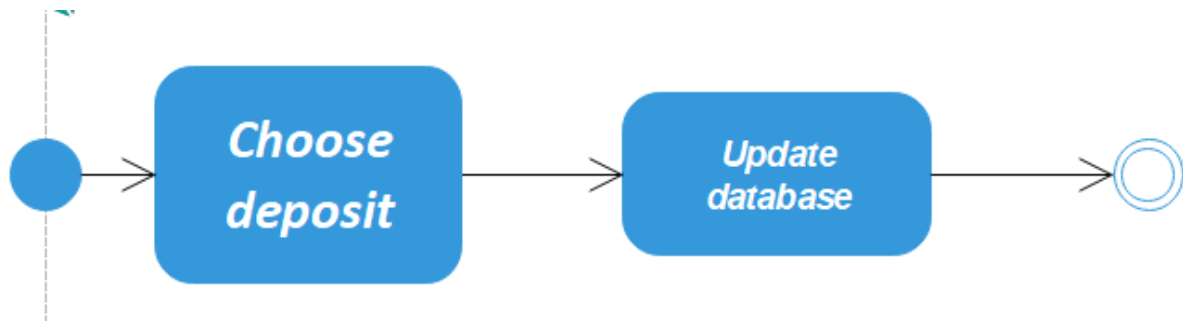
## CLASS:

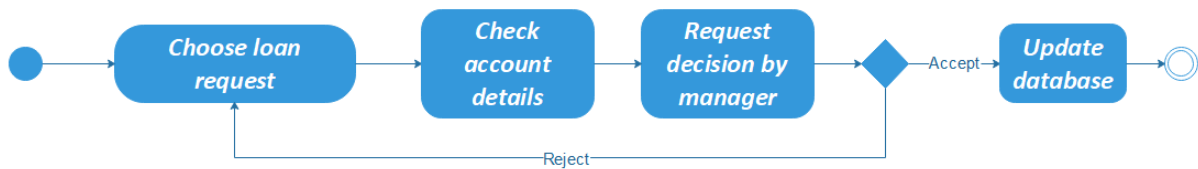
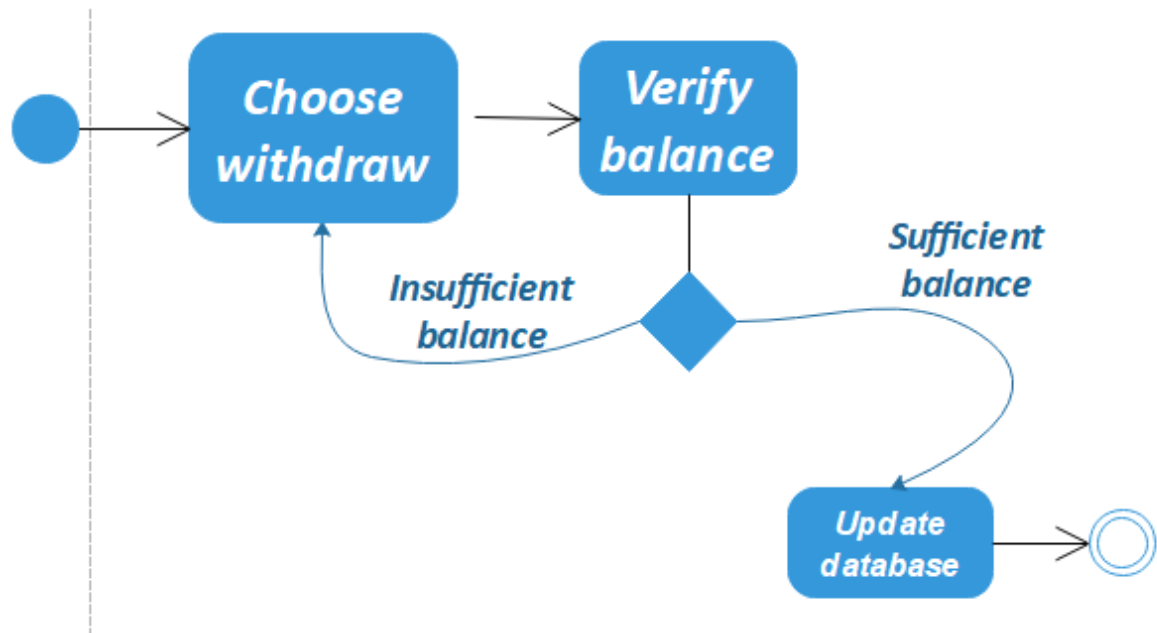


### COMPONENT:



### STATE CHART:





## **DATA DICTIONARY**

### **CUSTOMER:**

<b>Table name</b>	<b>ACCOUNTS_DETAILS</b>
<b>Description</b>	<b>Stores customer details</b>
<b>Primary key</b>	-
<b>Foreign key</b>	-

#	Field name	Data type	Constraints	Description
1	ACCOUNT_ID	Int	Not null	Store customer account ID
2	ACCOUNT_PASSWORD	VARCHAR(100)	Not null	Store customer password
3	FIRST_NAME	VARCHAR(100)	Not null	Store customer first name
4	LAST_NAME	VARCHAR(100)	Not null	Store customer last name
5	PHONE	Bigint	Not null	Store customer phone number
6	Date_Of_Birth	VARCHAR(100)	Not null	Store customer date of birth
7	EMAIL	VARCHAR(100)	Not null	Store customer email id
8	Address	VARCHAR(125)	Not null	Store customer address
9	Balance	Bigint	Null	Store customer balance
10	Status	Bit	Null	Indicate whether account is active or not
11	Login_Message	VARCHAR(256)	Null	Store message regards to login

### **TRANSACTION HISTORY:**

Table name	TRANSACTION_HISTORY_Table
Description	Stores customer transaction history details
Primary key	-
Foreign key	-

#	Field name	Data type	Constraints	Description
1	ACCOUNT_ID	Int	Not null	Store customer account ID
2	Previous_Balance	Bigint	null	Store customer previous balance
3	Withdrawn_Balance	Bigint	null	Store withdrawn balance
4	Deposited_Balance	Bigint	null	Store deposited balance
5	Withdrawn_by_Transaction	Bigint	null	Store balance withdrawn by transfer
6	Deposited_by_Transaction	Bigint	null	Store balance deposited by transfer
7	Current_Balance	Bigint	null	Store customer current balance
8	Transferred_From_Account_ID	Bigint	null	Store from transaction ID

9	Transferd_To_Account_ID	Bigint	null	Store to transaction ID
10	Operation_Date_And_Time	Datetime	null	Store from operation date

**MANAGER:**

Table name	Manager_Account
Description	Stores manager details
Primary key	-
Foreign key	-

#	Field name	Data type	Constraints	Description
1	ACCOUNT_ID	Int	Not null	Store manager account ID
2	ACCOUNT_PASSWORD	VARCHAR(100)	Not null	Store manager password
3	FIRST_NAME	VARCHAR(100)	Not null	Store manager first name
4	LAST_NAME	VARCHAR(100)	Not null	Store manager last name
5	PHONE	Bigint	Not null	Store manager phone number
6	EMAIL	VARCHAR(100)	Not null	Store manager email id

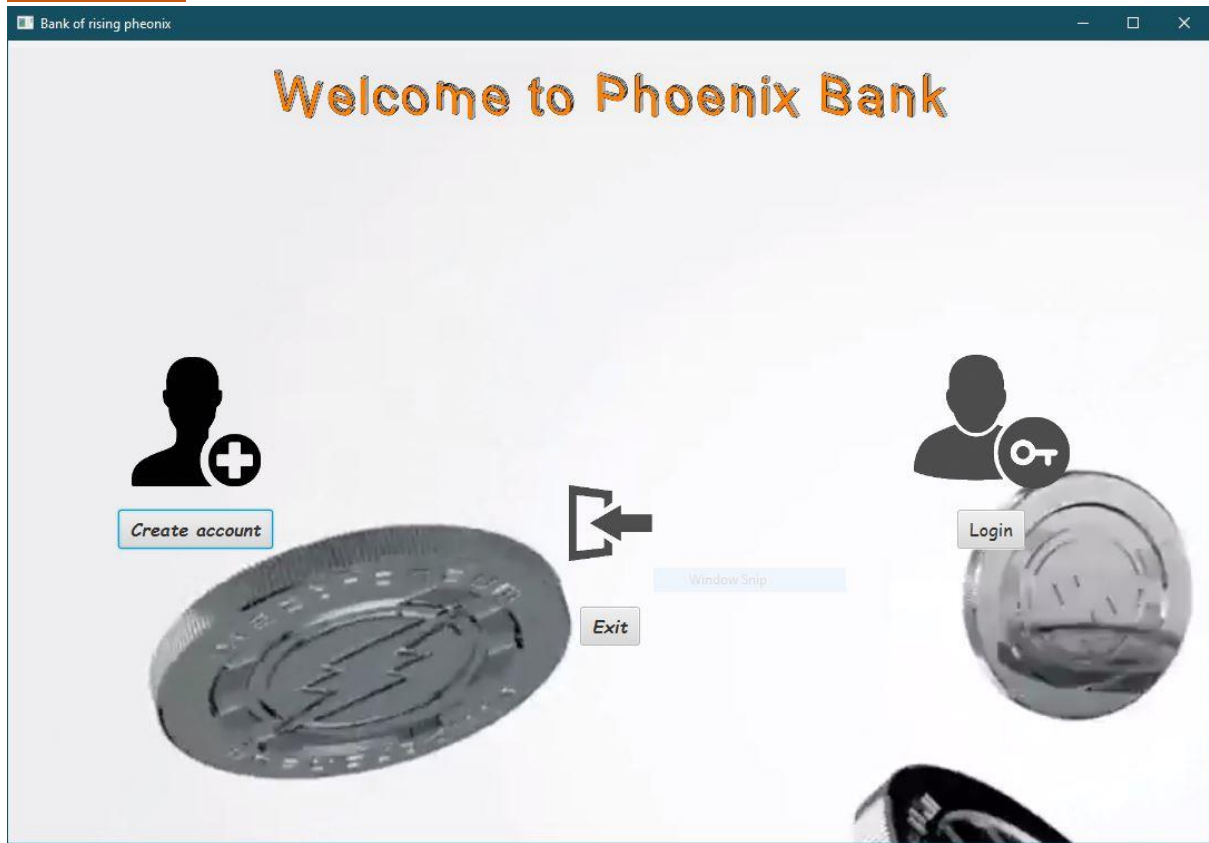
**LOAN:**

Table name	Loan
Description	Stores loan details
Primary key	-
Foreign key	-

#	Field name	Data type	Constraints	Description
1	ACCOUNT_ID	Int	Not null	Store customer account ID
2	LOAN_ID	Int	null	Store loan ID
3	LOAN_AMOUNT	Bigint	Not null	Store loan amount
4	RATE_OF_INTEREST	DOUBLE PRECISION	null	Store interest rate
5	LOAN_TENTURE	Int	Not null	Store tenture
6	STATUS	Bit	Not null	Store loan status
7	MESSAGE	VARCHAR(125)	null	Store message regards to loan

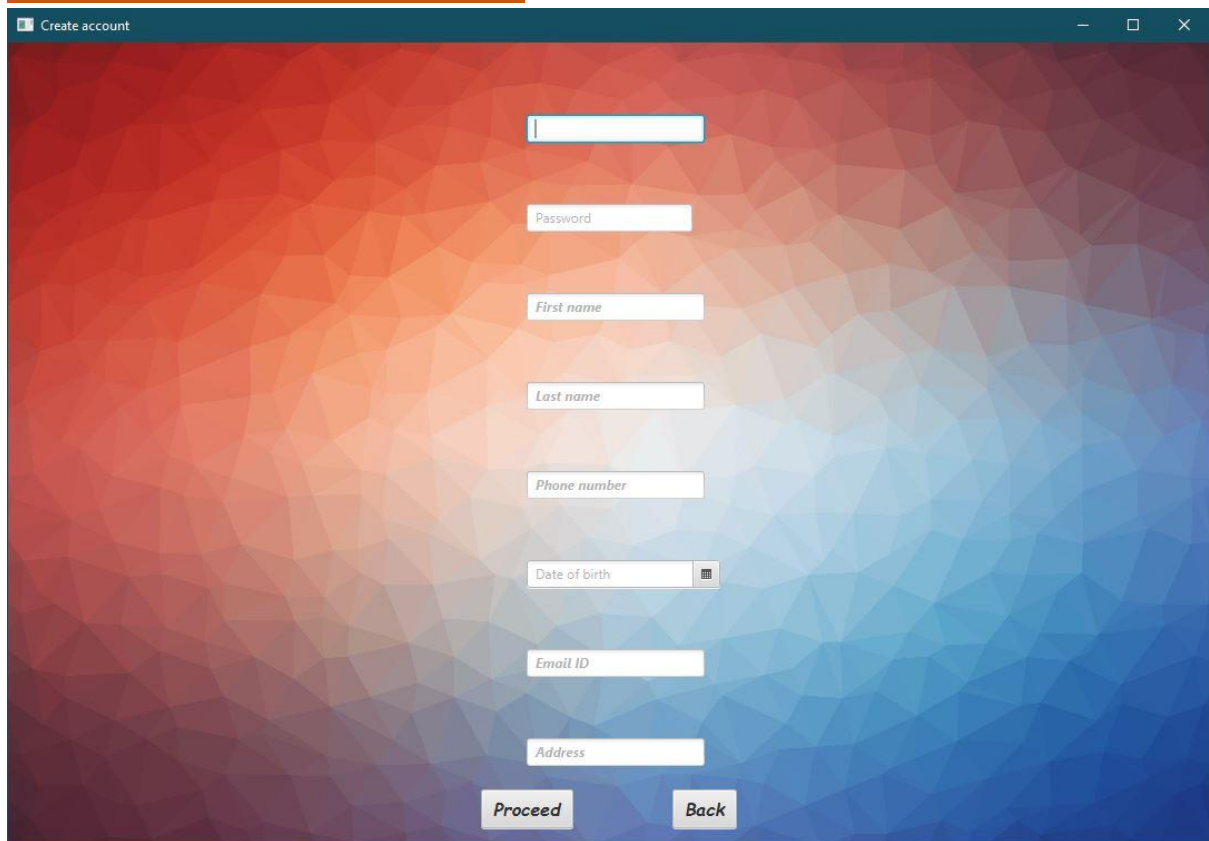
## INPUT DESIGN

### WELCOME:





## **CREATE CUSTOMER ACCOUNT:**



Create account

Password

First name

Last name

Phone number

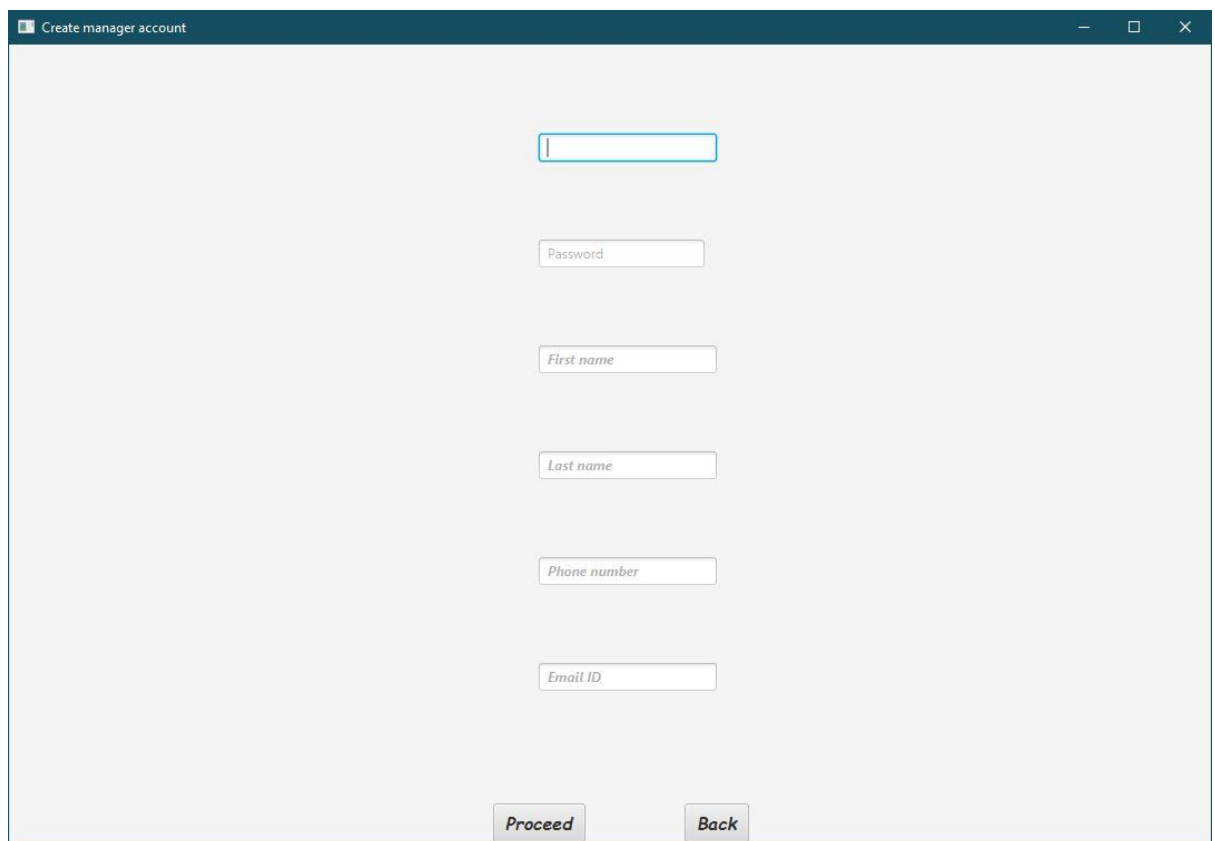
Date of birth

Email ID

Address

Proceed Back

## **CREATE MANAGER ACCOUNT:**



Create manager account

Password

First name

Last name

Phone number

Email ID

Proceed Back

## CUSTOMER FUNCTIONS:

Functions

Welcome Tarun Aitha

Deposit amount

Deposit

Check balance

Withdraw amount

Withdraw

Account ID

Account ID

Transaction history

Transfer amount

Transfer amount

Transfer money

Loan amount

Loan tenture

Loan status

Apply for loan

Loan details

Logout

## LOGIN:

Login

Manager

Customer

Password

Proceed

Back

## MANAGER FUNCTIONS:

Functions

—

□

×

Welcome Change Change

Approve customer accounts

Create customer account

Customer account ID to delete

Delete customer account

Customer account ID to update

Update customer account

Delete your account

Update your account

Create manager account

Approve loans

Logout

## UPDATE CUSTOMER ACCOUNT:

Update account

—

□

×

First name

Last name

Phone number

Date of birth

Email ID

Address

Proceed

Back

### UPDATE MANAGER ACCOUNT:

[illegible]

## OUTPUT DESIGN

### TRANSACTION HISTORY:

[illegible]

## APPROVE CUSTOMER ACCOUNTS:

[illegible]

### APPROVE LOANS:

The screenshot shows a web application window titled "Loans". The window contains a table with the following columns: "Customer account ID", "Loan ID", "Loan amount", "Rate of interest(%)", "Loan Tenture(Years)", "Status", and "Message". The table is currently empty, and the text "No content in table" is centered. At the bottom left of the window is a "Back" button, and at the bottom right is a "Proceed" button.

## TESTING

#	Data input	Expected output	Actual output	Pass/Fail
1	New account ID	Error message: Account already exists	Error message: Account already exists	Pass
2	New phone number	Error message: Enter Valid 10-Digit phone number	Error message: Enter Valid 10-Digit phone number	Pass
3	New e-mail ID	Error message: Enter Valid mail ID	Error message: Enter Valid mail ID	Pass
4	Login ID	Error message: Enter valid credentials	Error message: Enter valid credentials	Pass
		Error message: Account does not exist	Error message: Account does not exist	
		Error message: Account not approved by manager	Error message: Account not approved by manager	
5	Delete customer account	Error message: Account does not exist	Error message: Account does not exist	Pass
6	Update customer account	Error message: Account does not exist	Error message: Account does not exist	Pass
7	Transfer account ID	Error message: Account does not exist	Error message: Account does not exist	Pass
8	Withdraw amount	Error message: Insufficient amount	Error message: Insufficient amount	Pass
9	Transfer money	Error message: Insufficient amount	Error message: Insufficient amount	Pass
10	Apply new loan	Error message: Cannot apply for new loan clear previous loan	Error message: Cannot apply for new loan clear previous loan	Pass

### **POST IMPLEMENTATION**

- *The application is self –maintained.*
- *The project support already multiple customer and manger accounts.*
- *The manager can create customer account, delete customer account and update customer account.*
- *The manager can approve customers and loan requests .*

### **BIBLIOGRAPHY**

- *Online tutorials reference: Udemy, YouTube.*