Banking management system

Group no.112

Project under:-
Dr. Rahul Kala Sir
Dr. O.P Vyas Sir

Guided by:-
Bagesh Kumar Sir
Ayush Sinha Sir

Purpose::

- To allow only authorized user to access various function and processed available in the system.
- Locate any account wanted by the user.
- Reduced clerical work as most of the work done by computer.
- Provide greater speed and reduced time consumption.

System requirements::

- ❖ Software requirements

1. Programming language:: Java GUI swing
2. Database:: Oracle
3. Operating system :: Windows

Technologies to be used::

- ❖ JAVA SWING ::

Swing in Java is a lightweight GUI toolkit which has a wide variety of widgets for building optimized window based applications. It is a part of the JFC(Java Foundation Classes). It is build on top of the AWT API and entirely written in java. It is platform independent unlike AWT and has lightweight components.It becomes easier to build applications since we already have GUI components like button, checkbox etc. This is helpful because we do not have to start from the scratch.

❖ UML (Unified Modelling Language)::

UML stands for Unified Modeling Language, and it is a modeling language that is most often used for software engineering but has extended its use to business processes and other project workflows. Essentially, UML is visualizing software through diagrams, specifically one of the thirteen UML diagrams. This modeling language was created by three software engineers at the company Rational Software for their projects, and it has become the standard with very few updates over the years.

A UML diagram is typically used in software engineering and other business processes where modeling is useful. There are two main ways UML diagrams are used as a part of these processes:

- Forward design. The modeling and design are all done before coding the software application. Usually, forward design is used to help developers better see the system they are trying to create.
- Backward design. The modeling is done after the code has been written, and the UML diagrams act as documentation for the workflow of the project. This can help developers see the project development as it was, in reality, to improve in the future.

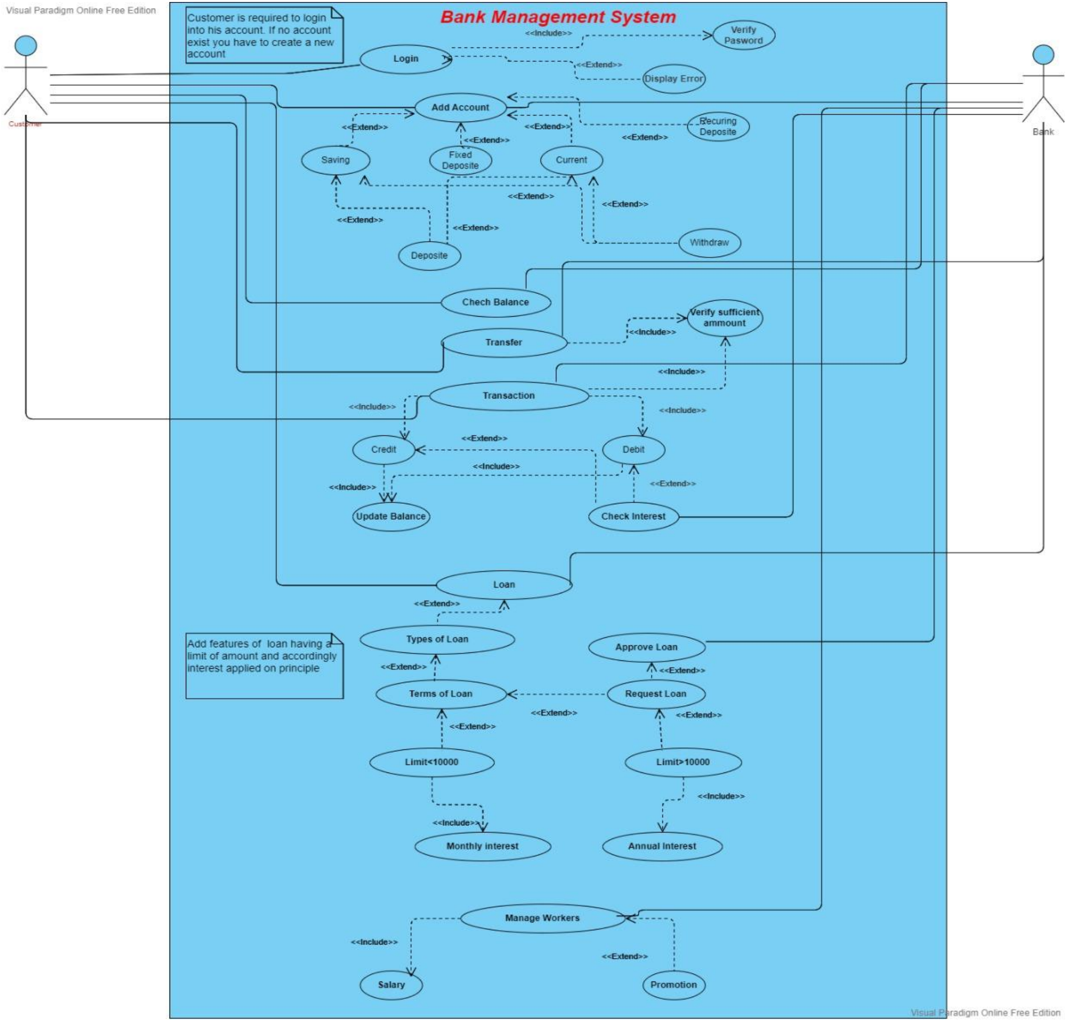
We have used UML diagrams which will give an overview of our bank management system by displaying classes , attributes,operations,interactions and relationships.

<u>We used following uml diagrams</u> –

- ➢ Use-case diagram
- ➢ Class diagram
- ➢ State machine diagram

- ❖ CRC cards (Class responsibilities collaborators)

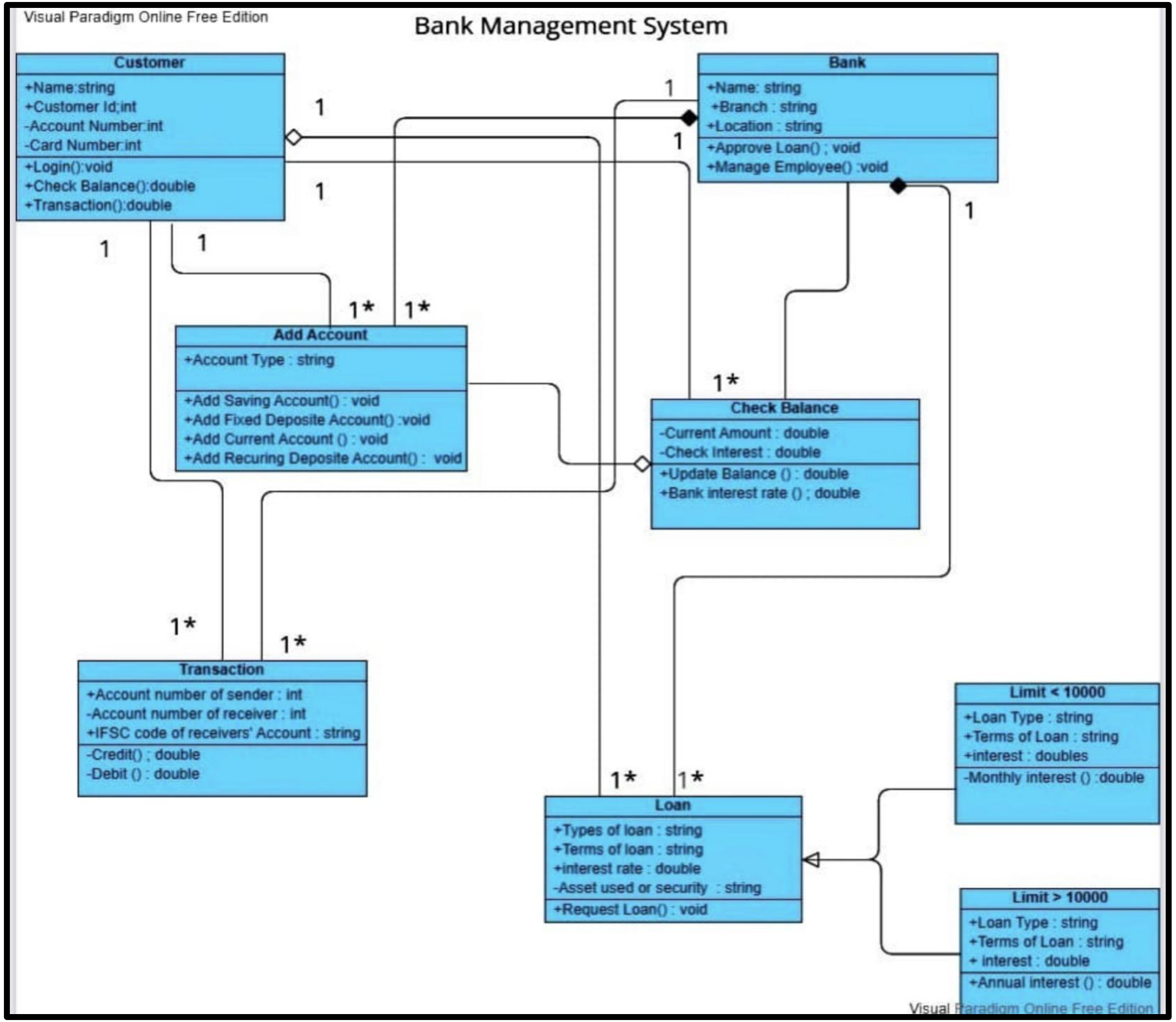
Use Case diagram

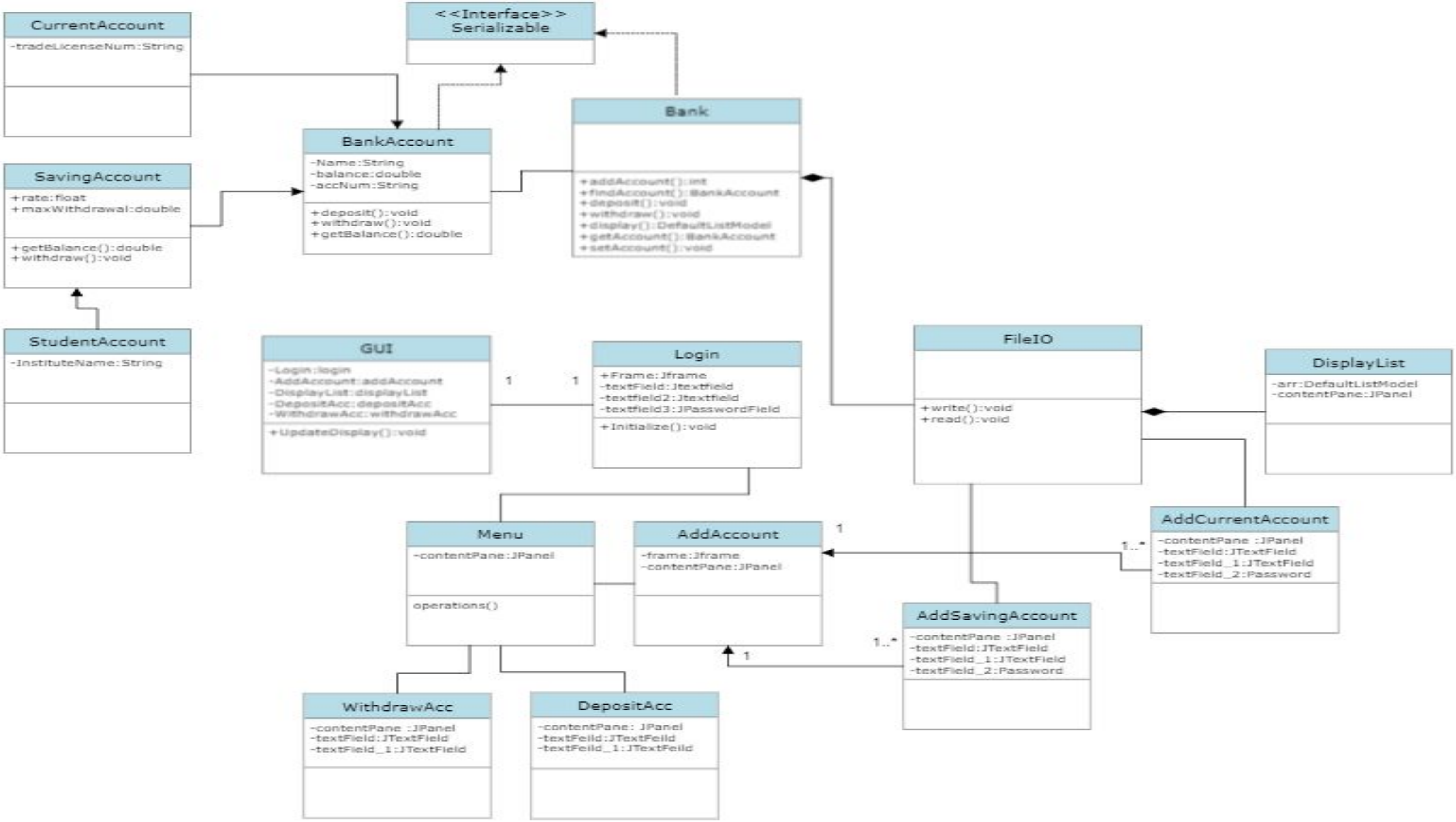


Customer is required to login into his account. If no account exist you have to create a new account

Add features of loan having a limit of amount and accordingly interest applied on principle

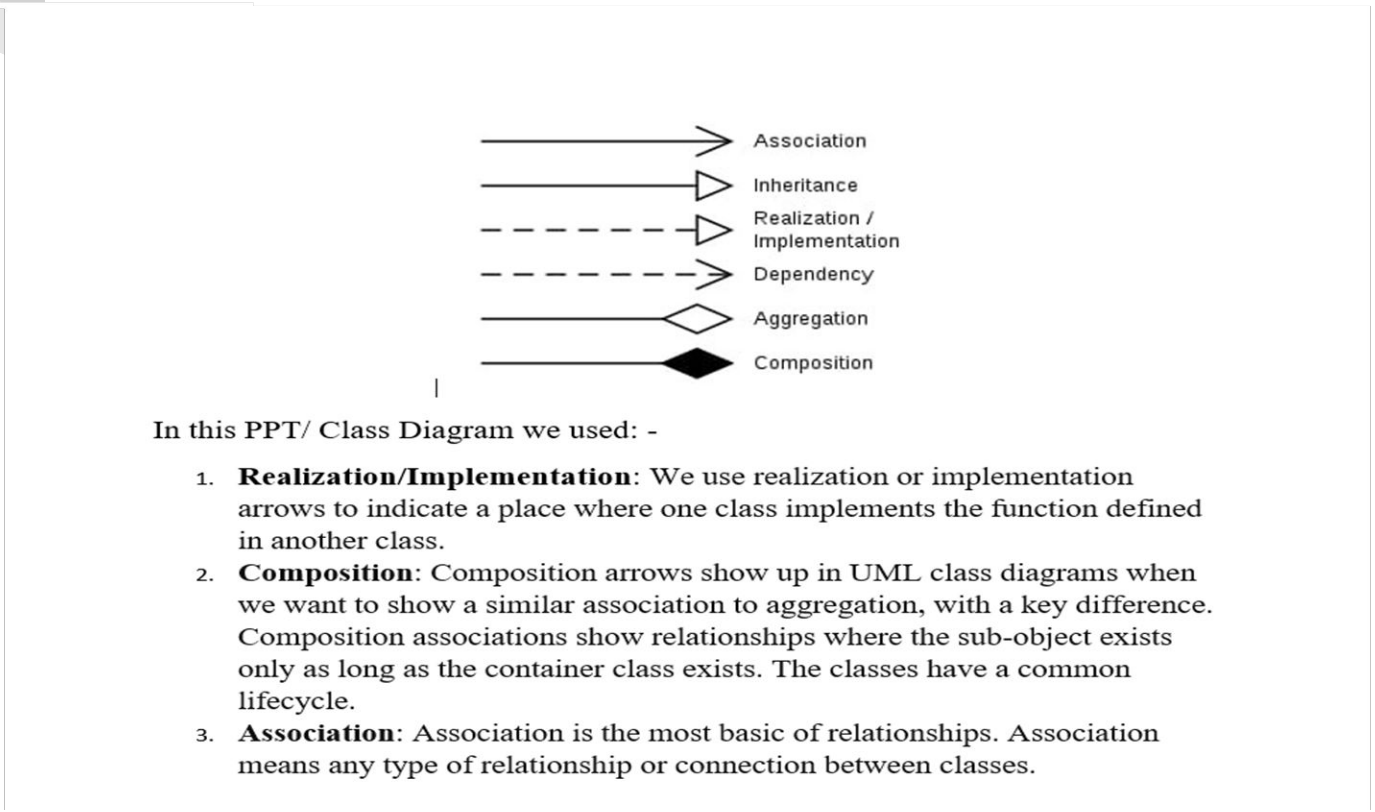
Class diagram





CurrentAccount
-tradeLicenseNum:String
<<Interface>>
Serializable
BankAccount
-Name:String
-balance:double
-accNum:String
+deposit():void
+withdraw():void
+getBalance():double
Bank
+addAccount():int
+findAccount():BankAccount
+deposit():void
+withdraw():void
+display():DefaultListModel
+getAccount():BankAccount
+setAccount():void
SavingAccount
+rate:float
+maxWithdrawal:double
+getBalance():double
+withdraw():void
StudentAccount
-InstituteName:String
GUI
-Login :login
-AddAccount:addAccount
-DisplayList:displayList
-DepositAcc:depositAcc
-WithdrawAcc:withdrawAcc
+UpdateDisplay():void
1
1
Login
+Frame:Jframe
-textField:Jtextfield
-textfield2:Jtextfield
-textfield3:JPasswordField
+Initialize():void
FileIO
+write():void
+read():void
DisplayList
-arr:DefaultListModel
-contentPane:JPanel
Menu
-contentPane:JPanel
operations()
AddAccount
-frame:Jframe
-contentPane:JPanel
1
1..*
AddCurrentAccount
-contentPane :JPanel
-textField:JTextField
-textField_1:JTextField
-textField_2:Password
AddSavingAccount
-contentPane :JPanel
-textField:JTextField
-textField_1:JTextField
-textField_2:Password
1..*
1
WithdrawAcc
-contentPane :JPanel
-textField:JTextField
-textField_1:JTextField
DepositAcc
-contentPane: JPanel
-textFeild:JTextFeild
-textFeild_1:JTextFeild

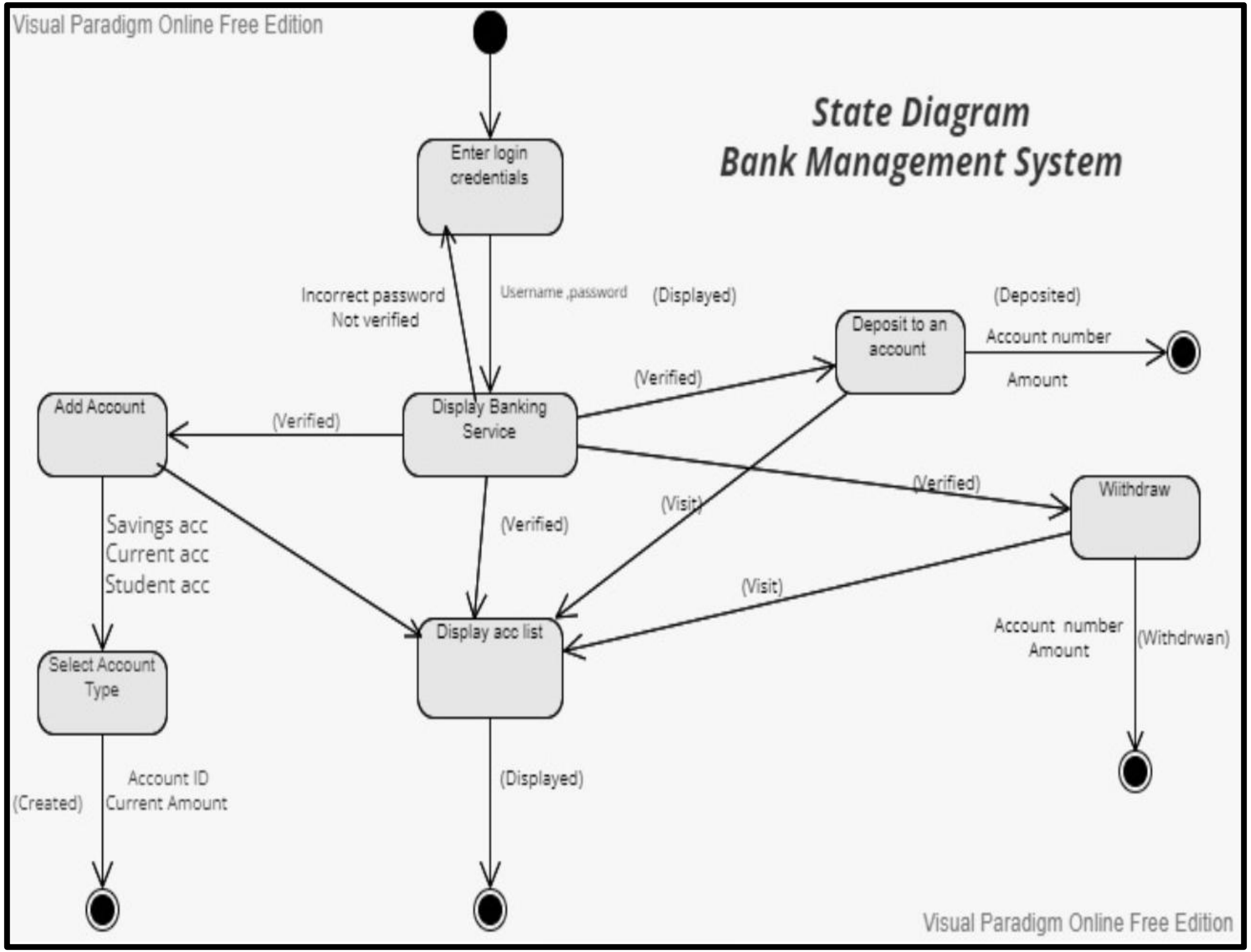
DATA FLOW DIAGRAM: CLASS DIAGRAM


Association
Inheritance
Realization / Implementation
Dependency
Aggregation
Composition

In this PPT/ Class Diagram we used: -

1. **Realization/Implementation**: We use realization or implementation arrows to indicate a place where one class implements the function defined in another class.
2. **Composition**: Composition arrows show up in UML class diagrams when we want to show a similar association to aggregation, with a key difference. Composition associations show relationships where the sub-object exists only as long as the container class exists. The classes have a common lifecycle.
3. **Association**: Association is the most basic of relationships. Association means any type of relationship or connection between classes.

State machine diagram



CRC Card

Customer	
Username	Bank
Password	
Account ID	
Login	
Check Balance	

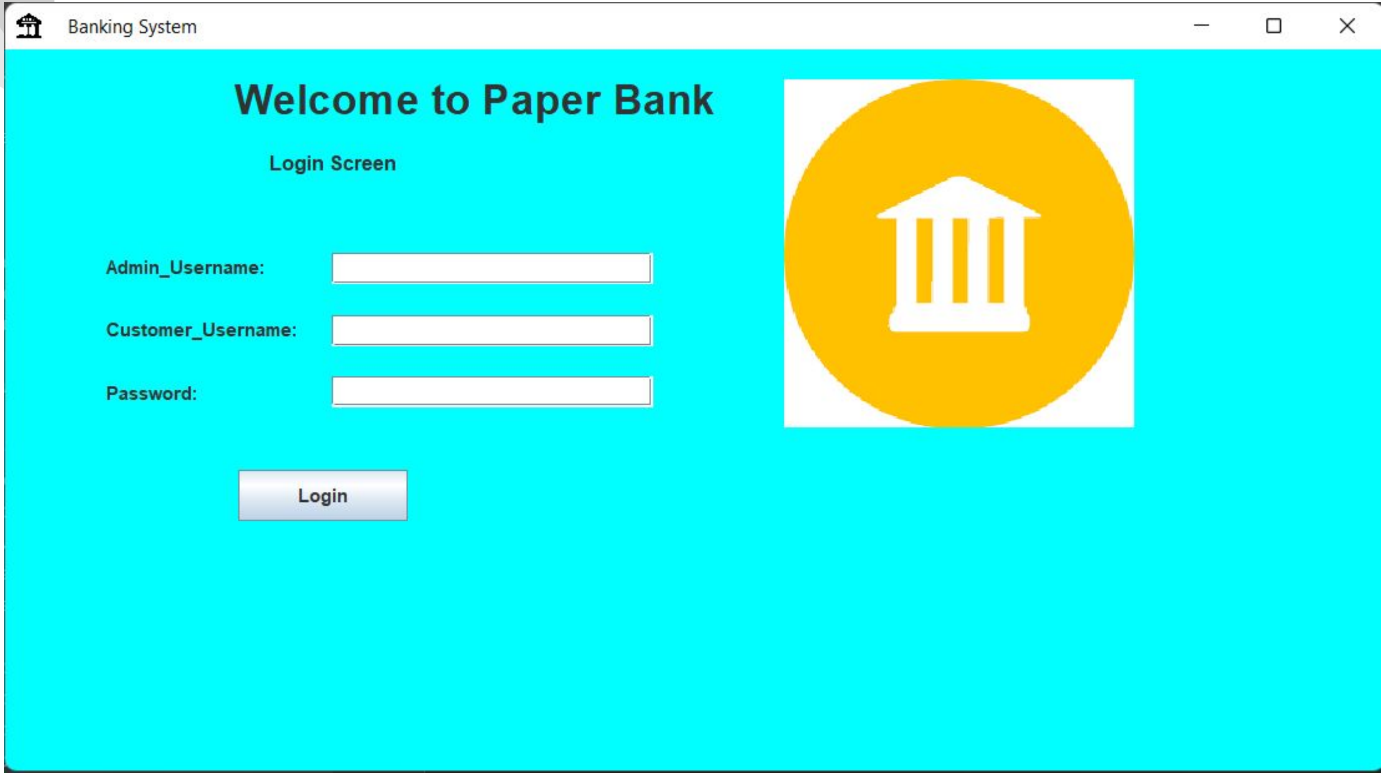
Add Account	
Account Type	Check Balance
Add Saving Account	
Add Current Account	
Add Deposit Account	

Check Balance	
Current Amount	Loan
Check Interest	
Update Balance :	

Loan	
Types Of Loan	Terms Action
Terms of Request	

Bank	
Name	Add Account
Branch	
Approve Loan	
Manage Employee	
Manage While system	

Login page

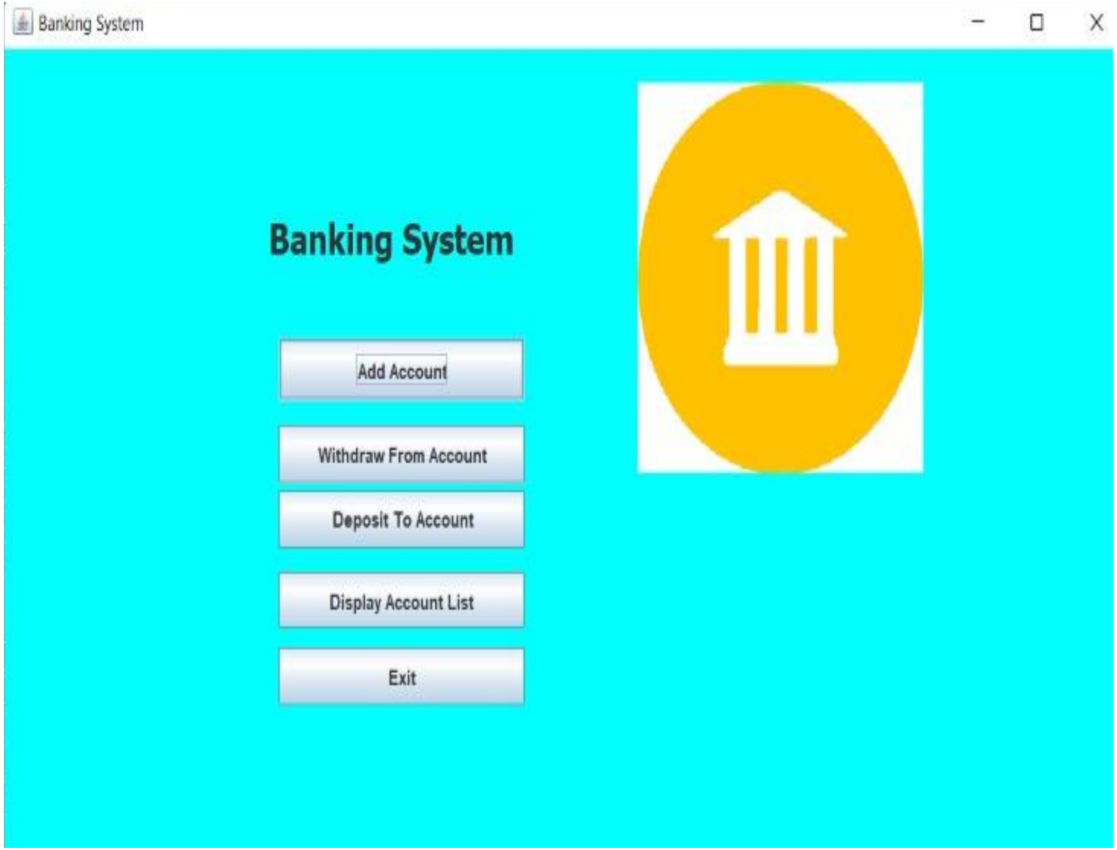


Banking System

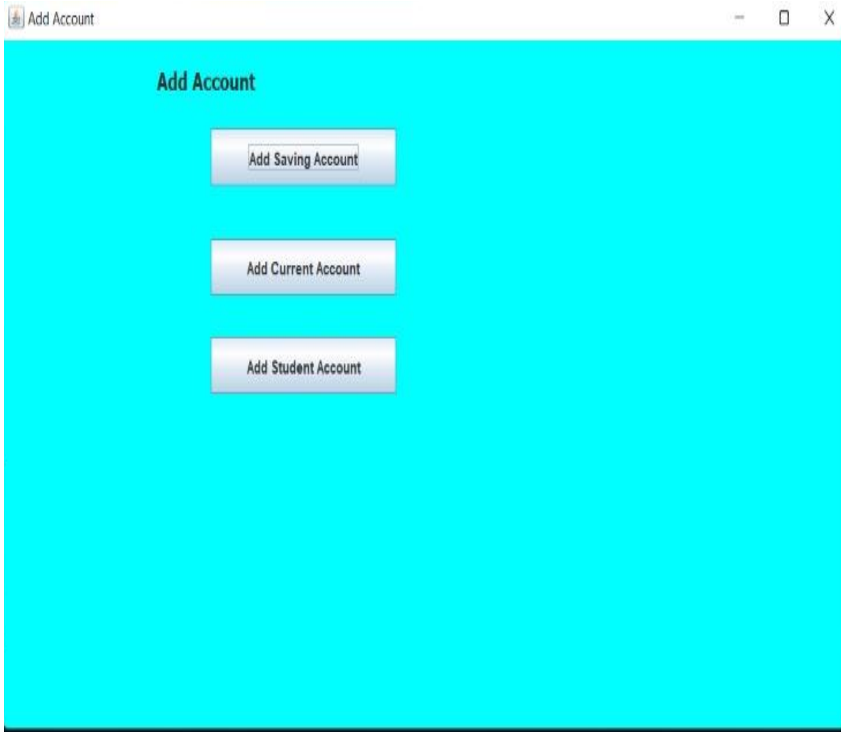
Welcome to Paper Bank

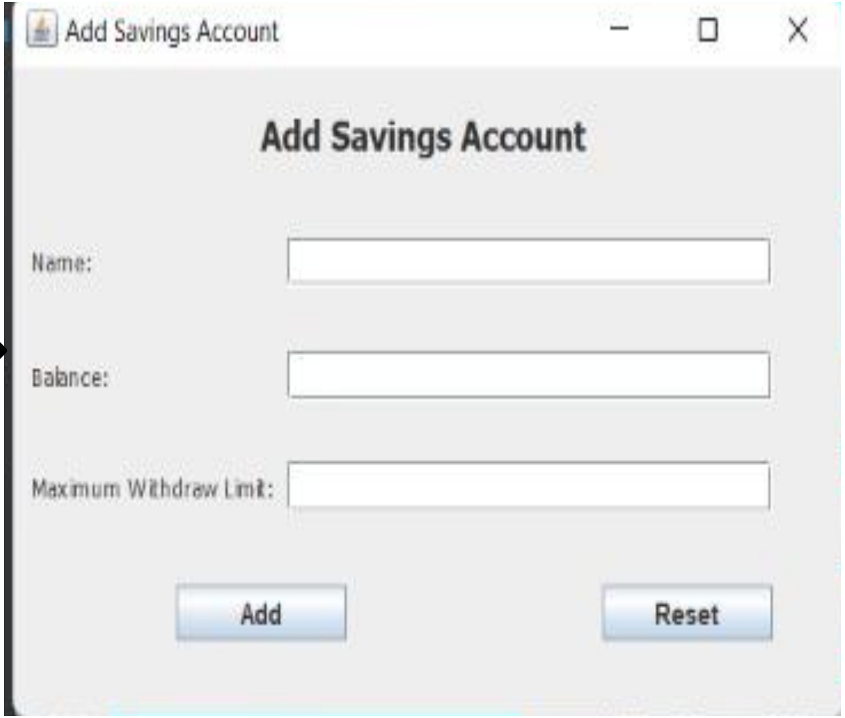
Login Screen

Banking system page

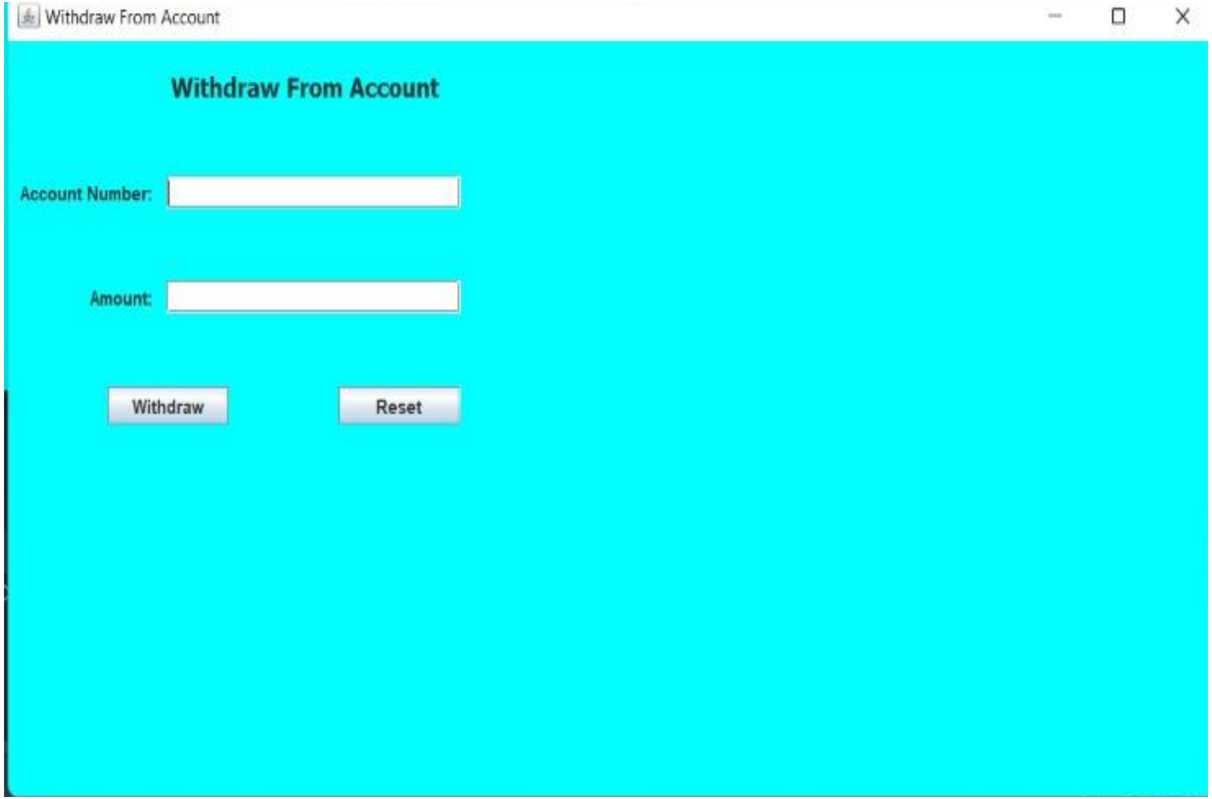


Add account
(After clicking on add saving account)





Withdraw from account



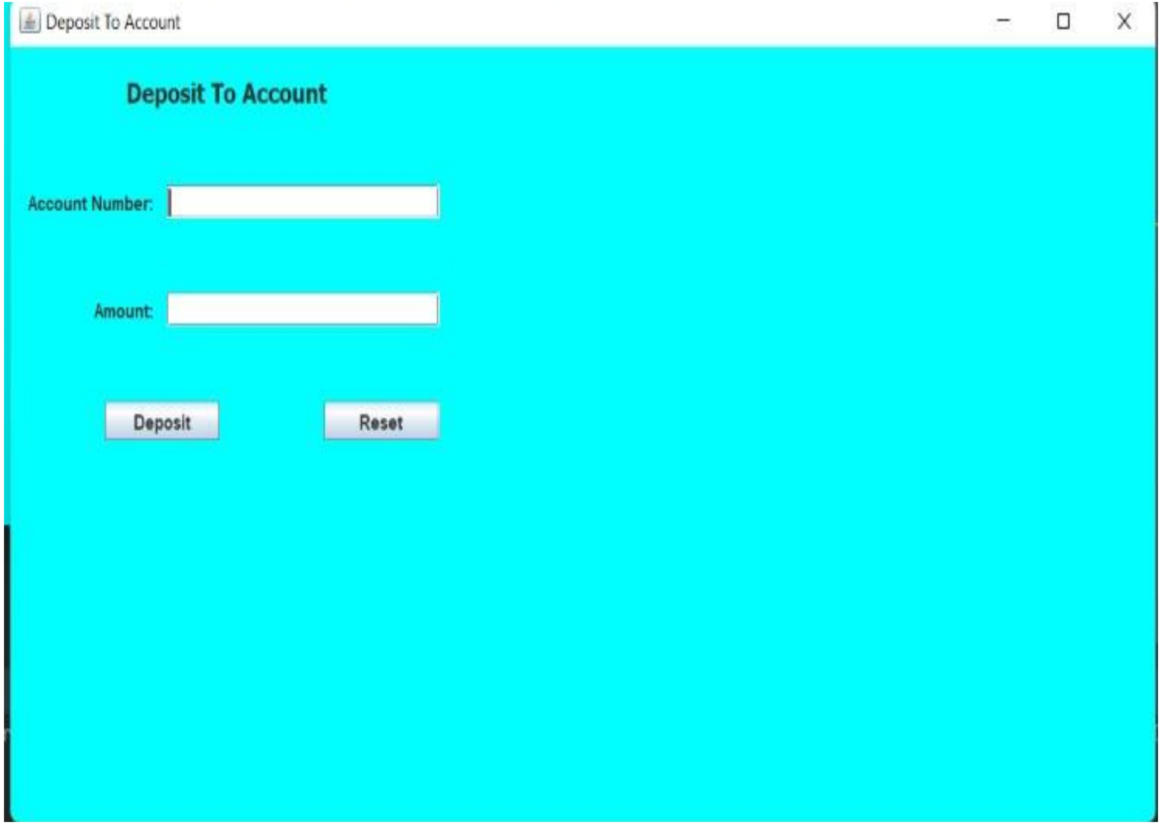
Withdraw From Account

Withdraw From Account

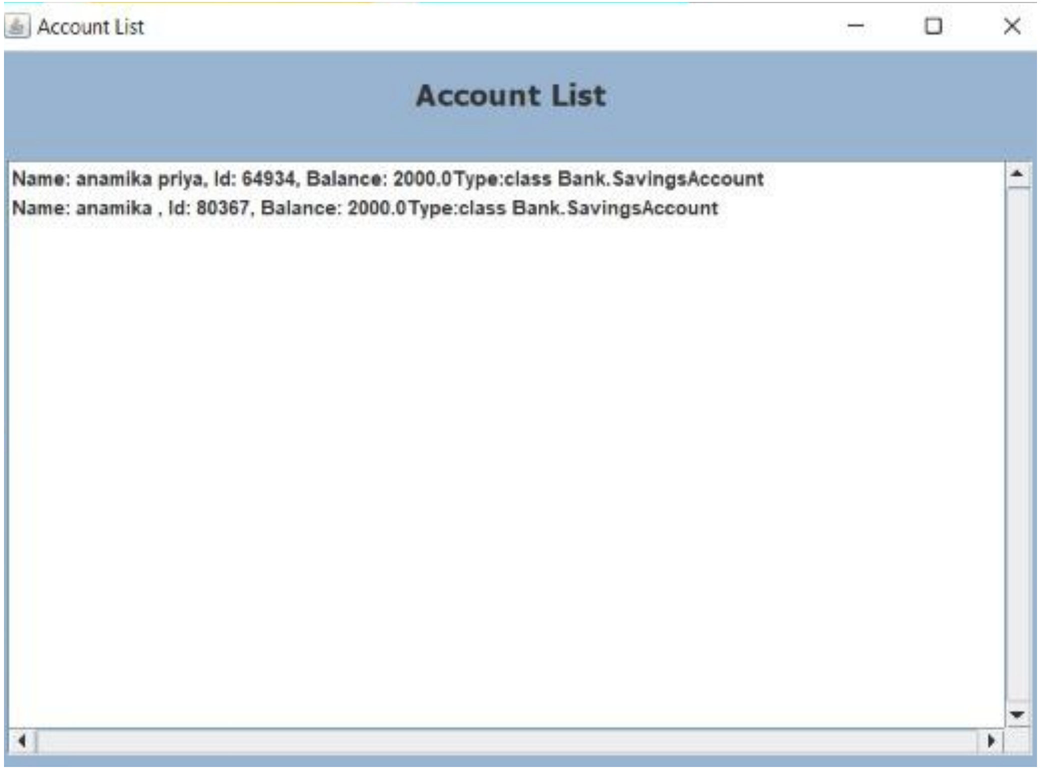
Account Number:

Amount:

Deposit to account



Display account list
(You can see information about your account details)



```
Name: anamika priya, Id: 64934, Balance: 2000.0Type:class Bank.SavingsAccount
Name: anamika , Id: 80367, Balance: 2000.0Type:class Bank.SavingsAccount
```

Future aspects of our app:-

➔ We can add QR system as like in phone pay and google pay.

➔ There should be no rush of people in the bank.

➔ It could provide the net-banking and rtgs functions.

➔ Payments of managers as well as the employee salary distributions can be managed.

➔ Attach aadhar card and their authentications.

➔ Can able to check the passbook details.

➔ Loans terminologies are transparent to the customers.

Conclusion::

This project is useful for the authorities which keep track of all the customers .Overall the project teaches us the essential skills like :GUI programming in java swing .We have learnt also unified modelling language. In JAVA ,we learnt also features of object oriented methodologies.

Thank You!