

A Mini Project Report On
“Simple Bank Management System”

submitted to Savitribai Phule Pune University, Pune



In partial fulfillment for the Java ESD Training 2022-23

Submitted by

Mr.Kiran Ramrao Kuyate

MS.Gayatri Arun Pendhari

Ms.Swati Navnath Jagtap

Ms.Amruta Sanjay Potphode



2022-2023

Department of Computer Engineering

Matoshri College of Engineering & ResearchCenter, Nashik

Academic Year 2022-2023

Certificate



Department of Computer Engineering
Matoshri College of Engineering and Research Centre, Nashik

This is to certify that,

Mr.Kiran Ramraov Kuyate
MS.Gayatri Arun Pendhari
Ms.Swati Navnath Jagtap
Ms.Amruta Sanjay Potphode

has successfully completed his/her project work on **Simple Bank Management System** at Matoshri college of Engineering and research Centre, Nashik in partial fulfillment of under Graduate course T.E Computer in academic year 2022-23 as prescribed by Savitribai Phule Pune University.

Place: Nashik

Date: / / 2023

Ms. P. R. Dholi
Project Guide

Dr. V.H. Patil
Head of Department

Dr. G. K. Kharate
Principal

ACKNOWLEDGEMENT

With deep sense of gratitude, we would like to thank all the people who have lit our path with their kind guidance. We are very grateful to these intellectuals who did their best to help during our project work. It is our proud privilege to express a deep sense of gratitude to **Prof. Dr. G.K. Kharate**, Principal of Matoshri College of Engineering and Research Center Nashik, for his comments and kind permission to complete this project. We remain indebted to **Dr. V. H. Patil**, Vice-Principal & H.O.D. Computer Engineering Department for his timely suggestion and valuable guidance.

The special gratitude goes to **Ms.P.R.Dholi** excellent and precious guidance in completion of this work. We thank to all the colleagues for their appreciable help for our working project. With various industry owners or lab technicians to help, it has been our endeavor throughout our work to cover the entire project work.

We are also thankful to our parents who provided their wishful support for our project completion successfully. And lastly, we thank our all friends and the people who are directly or indirectly related to our project work.

Project Members:

Mr.Kiran Ramrao Kuyate

MS.Gayatri Arun Pendhari

Ms.Swati Navnath Jagtap

Ms.Amruta Sanjay Potphode

ABSTRACT

The project is designed for developing the medical store management system whose purpose is to reduce the complexities of record keeping and documentation in the inventory bank management, Transactions and account creation. This work discusses the scope, objective and aim of the project along with the analysis of the risks and constraints in the project. The entire project plan is also discussed in this work with the Gantt chart and WBS diagram. The staffing required for completing is project is also mentioned in this work. The estimated cost has also been provided in this report.

The traditional approach to bank is manually. These days there is often more emphasis on software, which may consist of withdrawal, Deposit, Transfer of money, view balance and get mini statement and create new account option. It is a modern method of Bank management widely accepted and hastily increasing environment of daily life by using Simple bank management system application. It will considerably replace the traditional paper and pen-based, Manual transactions. It supports and helps the society and senior citizen to save time and energy and complete their work from home using this application.

TABLE OF CONTENT

Sr.No	TITLE	Page Number
1	Introduction_____	06
2	Literature Review_____	08
3	Requirement Analysis_____	10
4	Proposed System_____	13
5	Hardware Requirements_____	15
6	Software Requirements_____	11
7	Design of project	
	a.Data Flow Diagram_____	16
	b.ER Diagram_____	17
8	Implementation_____	18
9	Results_____	21
10	Conclusion_____	30
11	References_____	31

1.INTRODUCTION

The Project Banking system has been made to automate the Banking system. Through this bank management system user can manage all bank account activity like deposit money, withdraw money, transfer money from one account to another account, online payment etc. Using this bank management system user can check his account detail online like balance in account, bank statement etc.

The Administrator can check bank account with a login can work out with A/C holders of the bank can withdraw/ deposit cash / cheque /DD to/from their accounts. This system is also help bank user to create New account easily. The project makes a sincere effort to provide all the below-mentioned features to meet the requirements of the bank. In this project we have automate the bank process like Account Opening, Daily Transactions, Loan Sanctions, Account Maintenance. In this bank management system use can also search record of a particular Account Holder.

Introduction is based on the software program which is a low-level Bank System which can handles the low level bank management easily. It can be use by the business men who wants to remember their Account Status everyday. It going to be very handy for all of them as working in it very easy as compare to the complex programs like it which is not only difficult to understand but also most of people feel

difficulty while using these programs. This software program is very useful to maintain the information about the customer account.

This bank management system also allow user to add new customer account, delete account and user can also modify existing user account information. Using this system user can also search any individual account in few seconds. Using our bank management system user can also check any translation in any account. Our system also provide security check to reduce fraud. The system will check the user's existence in the database and provide the set of services with respect to the role of the user.

2.LITERATURE REVIEW

Existing System of Banking Management System:

The existing system work manually. The existing system has got lot of intricacies within itself and need lot of human effort and paper works. All above the data need to be maintained on ledgers and maintaining this is a tedious and risky process. As the transactions increases, so the data too. So the task of maintaining them increases exponentially. To view a data may need lot of paper to be searched.

The existing bank system is slow as every task is being performed by the human being and comparing the computer task speed with a computer is not fair. The complexity of this system is increased when an increase in the number of customers and with that there will be a number of transactions will be performed now everything needs to log in to a file for reference in the future which is simply not the kind of scenario we need at this time.

Some other drawbacks of the existing system:

- Less security of customer and bank information.
- Require more physical work and manpower.
- All the manual entry and editing will take more time.
- No level of clearance for the different levels of employees.
- Safety of paper documents from the disaster.
- No backup of the information.

The by looking at disadvantages these are pretty serious for any banking system as they are capable of bringing down the whole system. By digitalization in the banking system, it will not only achieve its goals and also will give some benefits like less manual calculation will be required.

The existing system work manually. The existing system has got lot of intricacies within itself and need lot of human effort and paper works. All above the data need to be maintained on ledgers and maintaining this is a tedious and risky process. As the transactions increases, so the data too. So the task of maintaining them increases exponentially. To view a data may need lot of paper to be searched.

Some of the negative aspects of the existing system are as follows:

1) Time Consuming: There is a lot of time consumes in the bank, whenever we open account, deposit, withdraw or pass a loan than because of many customers with his/her different purpose, than we wait for our turn sometimes 2 to 3 hours.

2) Reliability: This banking system is not fully reliable whenever the computer system is create a problem and not work properly than sometime our data is damaged or lost.

3) Man Power: In this project man power is fully used. A number of employee need to manage the banking system.

4) Less Accurate:

This system is not fully accurate, because sometime computrised system create a problem in working, than the computer system also give us wrong results.

3.REQUIREMENT ANALYSIS

There are many different important factors of having a computer within the banking sector. It makes life a lot quicker and easier and there will no longer have to be manual inputs of data on paper and within records, it can all be taken care of by the computer and be stored and sent on accordingly. Previous transactions of certain customers can be found in seconds as opposed to waiting for a long period of time, and therefore problems and issues can be dealt with a lot quicker as the information can be found instantly. You can also see the importance of the computer within the banking sector as you can now make transactions from your home with the technology and keep an eye on what you have within your account when you are waiting for a statement to come through or want to buy something online. With in the computer being used within the banking sector, the banking system will still be as long and as manual as it was many years ago and you would not be able to transfer money and stay ahead of statements successfully.

Electronic banking is now more popular than ever and this is due to the introduction of computers within the banking sector. Being able to communicate with different banks from all over the world is also possible due to the new injection of technology and there are much less problems with data imports and exports now that everything can be done by a few clicks of a button.

Some improvements by executing the proposed system:

- More secure information will give a layer of security of authentication and authorization.
- Required very little manpower.
- Simplify the problem of editing.
- Maintain the clearance level by the hierarchy.

- The information will be secure from the different types of disasters as there will be an automatic backup system for the customer and bank information.
- Maintain data integrity Validate the manual calculations avoid calculation error.
- Safeguard the data accuracy.
- More reliable and efficient.
- More user-friendly interface.

4.PROPOSED SYSTEM

The main objective of the project is to develop online Banking system for banks. In present system all banking work is done manually. User have to visit bank to Withdrawal or Deposit amount. In present bank system it is also difficult to find account information of account holder. In this bank management system we will automate all the banking process. In our bank management system user can check his balance online and he can also transfer money to other account online. In this Software you can keep record for daily Banking transactions. The main purpose of developing bank management system is to design an application, which could store bank data and provide an interface for retrieving customer related details with 100% accuracy.

This bank management system also allow user to add new customer account, delete account and user can also modify existing user account information. Using this system user can also search any individual account in few seconds. Using our bank management system user can also check any translation in any account. Our system also provide security check to reduce fraud. The system will check the user's existence in the database and provide the set of services with respect to the role of the user.

To develop a software for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the users workspace to have additional functionalities which are not provided under a conventional banking software.

In this project we are going to explain about Banking Management System. This project have facility to opening account, depositing and withdrawing money. The Bank management system is an application for maintaining a person's account

in a bank. The system provides the access to the customer to create an account, deposit/withdraw the cash from his account, also to view reports of all accounts present. The following presentation provides the specification for the system.

There are seven modules in this System:

- 1). Deposit Module
- 2). Withdraw Module
- 3). Cancel Account Module
- 4). Create File Module
- 5). Open Account Module
- 6). Search Module
- 7). Report Module

- Deposit: Provides options to deposit amount from the given account number.
- Withdraw: Provides options to withdraw amount from the given account number.
- Cancel Account: Cancels the selected account from the bank.
- Create File: Selecting this creates a new file for the user by accepting input such as account number, name and amount.
- Open account: Opens a new account for the user by accepting input such as account number, name and minimum balance
- Search: Enables to search for the details of the given account number. Displays only one account detail at a time.
- Report: Displays the list of all account & Details comprising of account number, name and balance amount.

5.HARDWARE REQUIREMENTS:

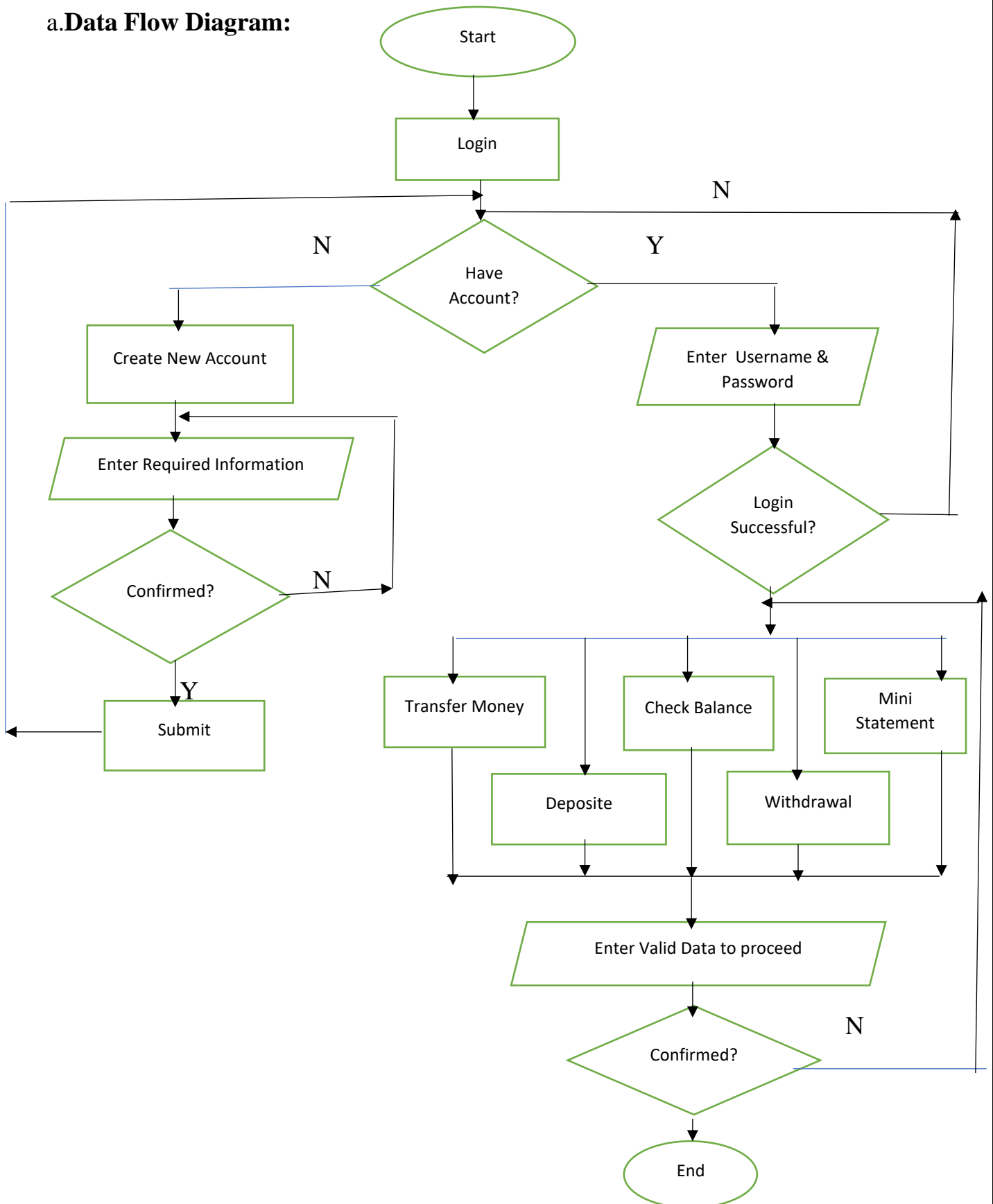
- INTEL CORE I3 PROCESSOR
- 8GB Ram
- 100GB HDD

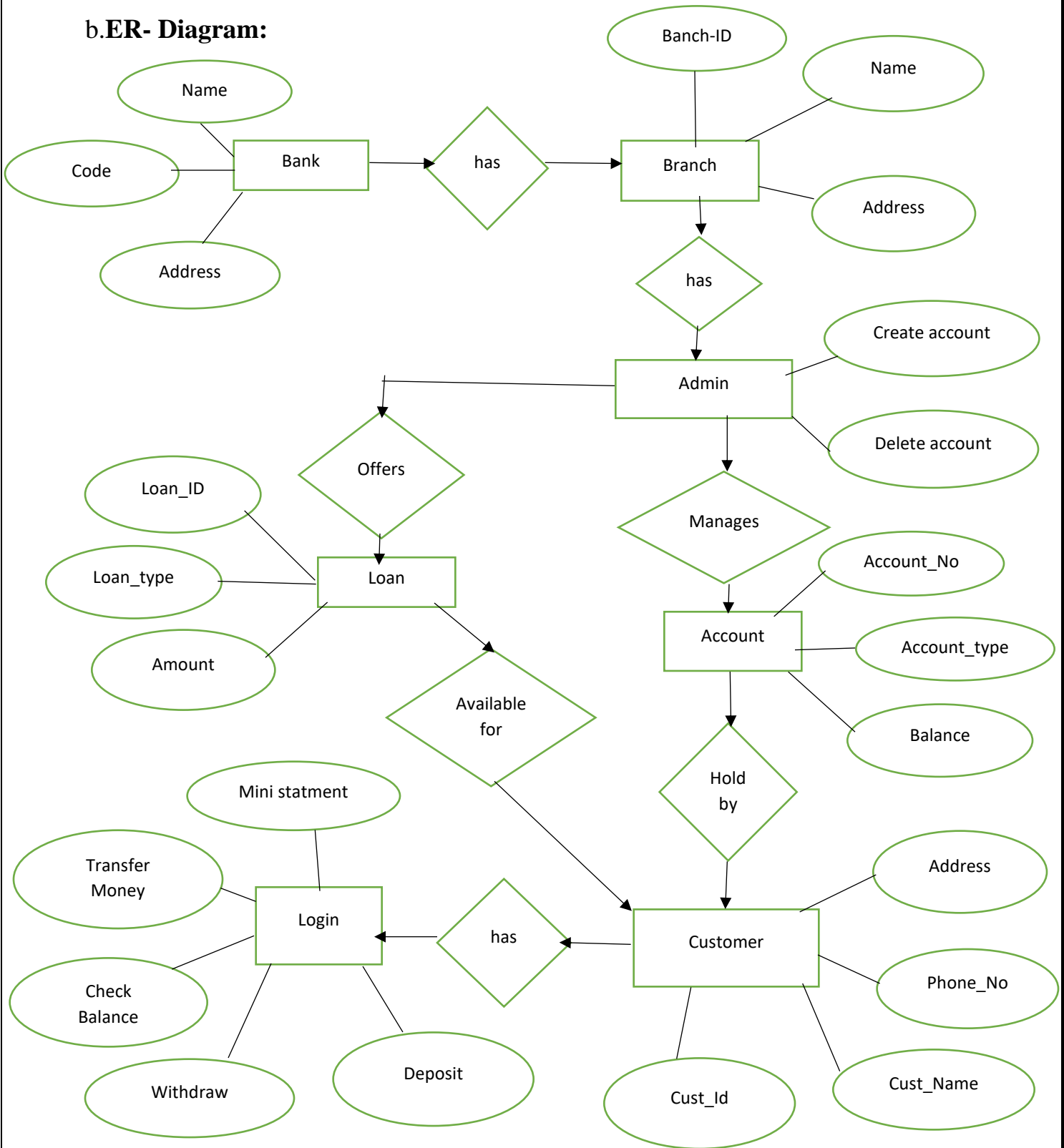
6.SOFTWARE REQUIREMENTS:

- Windows 10
- Java Compiler
- Text editor

7.DESIGN OF PROJECT:

a.Data Flow Diagram:



b.ER- Diagram:**Figure 3.1: ER-Diagram for Simple Bank Management System**

8.IMPLEMENTATION

- In this project, we had defined a Java class named as **Account** that represents a bank account. The class has several methods for depositing and withdrawing money, retrieving transaction history, and updating the balance of an account.
- The Account class has a constructor that takes a Login object, account number, and balance as parameters to initialize an instance of the class. It also has a default constructor that prints a message to prompt the user to provide account details.
- The **deposit method** allows a user to add money to their account by increasing the balance of their account. It also saves the transaction details to a file using the saveTransactionDetails method.
- The **withdraw method** allows a user to withdraw money from their account if their balance is sufficient. It also saves the transaction details to a file using the **SaveTransactionDetails method**.
- The **miniStatement method** retrieves the last 5 transaction details of an account and displays them to the user.
- The saveTransactionDetails method saves the transaction details to a file named **AccountData.txt**. The numberOfTransactions method counts the number of transactions made by an account and returns the count.

- The **isLoanable method** checks if an account is eligible for a loan based on the number of transactions made and whether the account has an active loan.
- The **readLastNTransactions method** reads the last n transaction details of an account from the AccountData.txt file and returns them as a list.
- The **updateBalance method** updates the balance of an account in the userData.txt file. The updateLoanBalance method updates the loan balance and loan status of an account in the **userData.txt** file.
- We had also defined a class named **Login** that represents a banking application login system. The class has several private member variables such as username, password, email, phone number, birthdate, account number, account type, loan status, balance, and several Boolean flags for account features like debit card, credit card, international transactions, and chequebook.
- The class also has several public methods that can be used to retrieve information about the user's account such as **getAccount()**, **getBalance()**, **isLoanActive()**, and **isChqueBook()**. It also has several methods to check if the user has certain features enabled such as **isCreditCard()**, **isDebitCard()**, and **isInternationalTransaction()**.
- The constructor of the class takes in two arguments, username and password, and calls the **isUserExist()** method to check if the

provided credentials are valid or not. If they are valid, it prints "Login successful", otherwise, it prints "Login failed".

- The **isUserExist() method** reads user data from a file named "UserData.txt" and compares the provided username and password with the stored data. If the provided credentials match with the stored data, it sets the member variables of the class with the user's information and returns true. Otherwise, it returns false.
- The **createAccount() method** is used to create a new account for a user. It prompts the user to enter their name, email address, phone number, birth date, account type, password, and opening balance. It then generates a unique account number for the user and sets the member variables of the class accordingly. It also writes the user's account data to the "UserData.txt" file.
- Overall, this code provides the functionality for users to login to their account and create a new account. It also allows users to retrieve information about their account and check if certain features are enabled or not.

9.Result Snapshots

9.1 Registration Page and Login Page

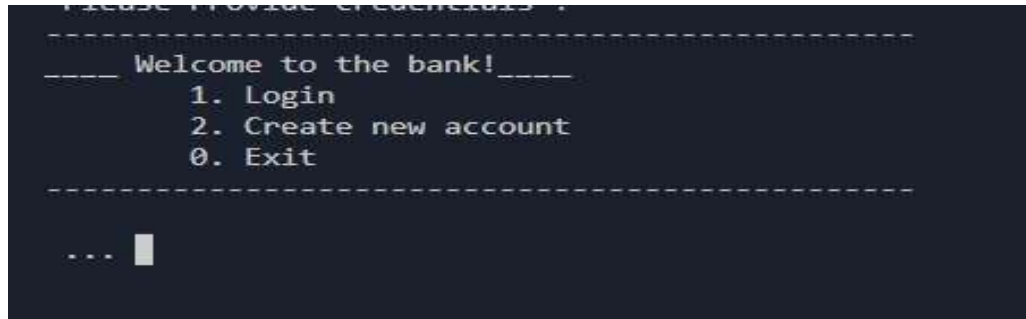


Fig 9.1:Registration and login Page

9.2Login Page

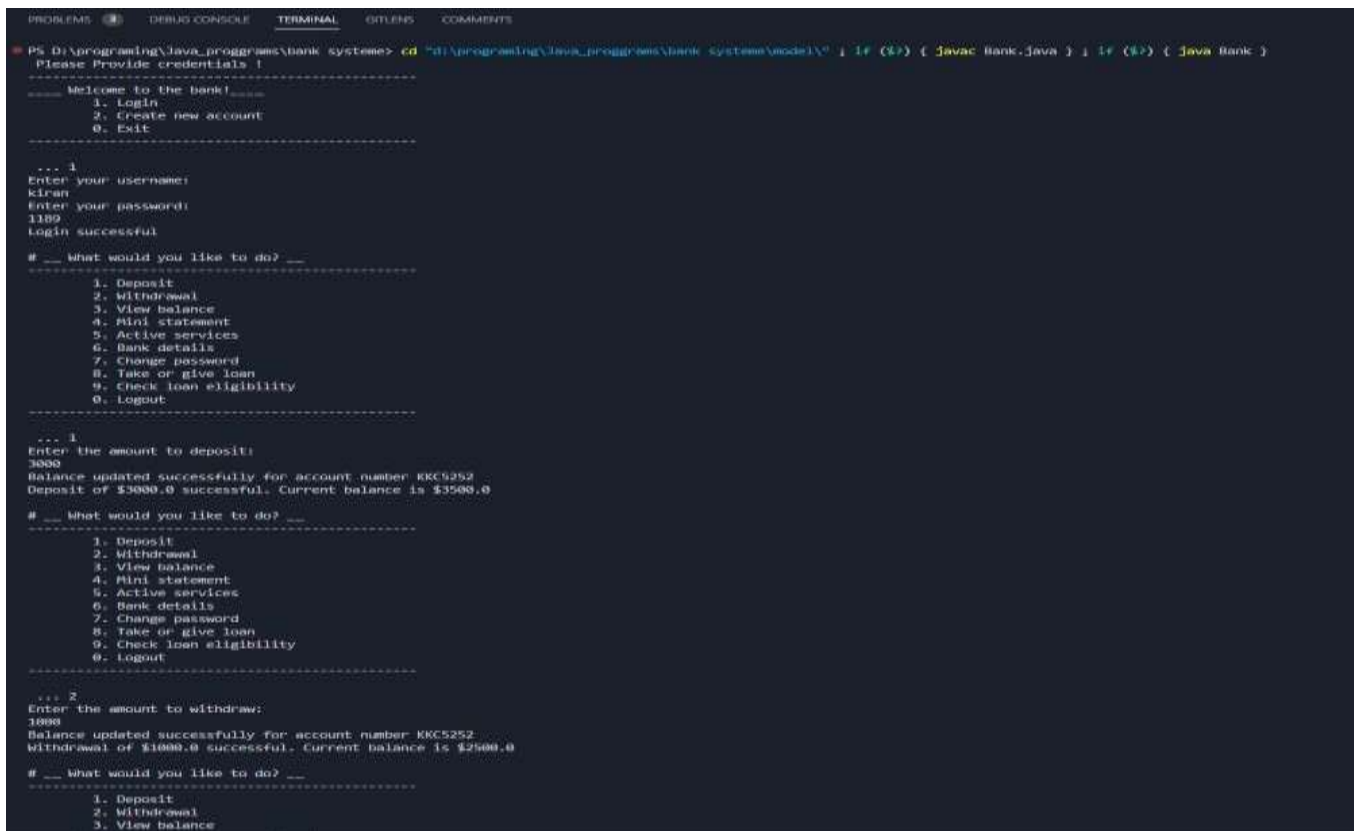


Fig 9.2:Login page & menu after Successful login

9.3 Deposit Amount in account

```

PROBLEMS ② DEBUG CONSOLE TERMINAL OUTPUTS COMMENTS
PS D:\programming\java_programs\bank_system> cd "D:\programming\java_programs\bank_system\src" & if ($?) { javac Bank.java } & if ($?) { java Bank }
Please Provide credentials :
-----
Welcome to the bank!
1. Login
2. Create new account
0. Exit
-----
... 1
Enter your username:
siran
Enter your password:
1189
Login successful

# __ What would you like to do? __
-----
1. Deposit
2. Withdrawal
3. View balance
4. Mini statement
5. Active services
6. Bank details
7. Change password
8. Take or give loan
9. Check loan eligibility
0. Logout
-----
... 1
Enter the amount to deposit:
2000
Balance updated successfully for account number KKC5252
Deposit of $2000.0 successful. Current balance is $5000.0

```

Fig 9.3:Deposit Amount

9.4 Withdrawal of Money

```

# __ What would you like to do? __
-----
1. Deposit
2. Withdrawal
3. View balance
4. Mini statement
5. Active services
6. Bank details
7. Change password
8. Take or give loan
9. Check loan eligibility
0. Logout
-----
... 2
Enter the amount to withdraw:
1000
Balance updated successfully for account number KKC5252
Withdrawal of $1000.0 successful. Current balance is $2500.0

# __ What would you like to do? __

```

Fig 9.4: WithDrawal Of Money

9.5 View Balance

```

2. Withdrawal
3. View balance
4. Mini statement
5. Active services
6. Bank details
7. Change password
8. Take or give loan
9. Check loan eligibility
0. Logout
-----

... 3
Your balance is: 2500.0

# __ What would you like to do? __
-----

```

Fig9.5:View Account Balance

9.6 Mini Statement

```

# __ What would you like to do? __
-----

1. Deposit
2. Withdrawal
3. View balance
4. Mini statement
5. Active services
6. Bank details
7. Change password
8. Take or give loan
9. Check loan eligibility
0. Logout
-----

... 4
Last 2 transaction details transactions :
Deposit of $3000.0 on 2023-02-25T15:26:16.668485100
Withdrawal of $1000.0 on 2023-02-25T15:26:34.240986800

+-----+
| Date/Time          | Transaction Type | Amount  | Balance |
+-----+
| 2023-02-25T15:26:34.240986800 | Withdrawal      | $1000.00 | 2500.0  |
| 2023-02-25T15:26:16.668485100 | Deposit         | $3000.00 | 3500.0  |
+-----+

# __ What would you like to do? __
-----

```

Fig 9.6:Get Mini Statement

9.7 Check Active Services of Your Account

```
# __ What would you like to do? __
-----
1. Deposit
2. Withdrawal
3. View balance
4. Mini statement
5. Active services
6. Bank details
7. Change password
8. Take or give loan
9. Check loan eligibility
0. Logout
-----

... 5
The following services are active:
- emails
- SMS
- debit card

# __ What would you like to do? __
-----
```

Fig 9.7:Displaying Active Services of Account

9.8 Get Bank Details

```
# __ What would you like to do? __
-----
1. Deposit
2. Withdrawal
3. View balance
4. Mini statement
5. Active services
6. Bank details
7. Change password
8. Take or give loan
9. Check loan eligibility
0. Logout
-----

... 6
Welcome to Komal Secure Bank Pvt. Ltd.
We are one of the leading banks in India.
Our services include: savings accounts, current accounts, fixed deposits, loans, and credit cards.
Our branches are located in various cities across India, including Mumbai, Delhi, Bangalore, Chennai, Kolkata, and Hyderabad.
For more information, visit our website at www.komalsecurebank.com.

# __ What would you like to do? __
-----
```

Fig 9.8:Bank Details

9.9 Change Password

```

8. Take or give loan
9. Check loan eligibility
0. Logout
-----

... 6
Welcome to Komsal Secure Bank Pvt. Ltd.
We are one of the leading banks in India.
Our services include: savings accounts, current accounts, fixed deposits, loans, and credit cards.
Our branches are located in various cities across India, including Mumbai, Delhi, Bangalore, Chennai, Kolkata, and Hyderabad.
For more information, visit our website at www.komsalsecurebank.com.

# __ What would you like to do? __
-----
1. Deposit
2. Withdrawal
3. View balance
4. Mini statement
5. Active services
6. Bank details
7. Change password
8. Take or give loan
9. Check loan eligibility
0. Logout
-----

... 7
Enter old password: 1234
Enter new password:
7246
7246
password updated successfully

# __ What would you like to do? __
-----

```

Fig 9.9:After losing old password ,getting new password

9.10 Check Loan Eligibility

```

# __ What would you like to do? __
-----
1. Deposit
2. Withdrawal
3. View balance
4. Mini statement
5. Active services
6. Bank details
7. Change password
8. Take or give loan
9. Check loan eligibility
0. Logout
-----

... 8
-----
1.take loan
2.give loan
-----

... 2
Your not eligible for Loan

# __ What would you like to do? __
-----

```

Fig 9.10:If you are not eligible for loan

9.11 Apply for Loan

```
# __ What would you like to do? __
-----
1. Deposit
2. Withdrawal
3. View balance
4. Mini statement
5. Active services
6. Bank details
7. Change password
8. Take or give loan
9. Check loan eligibility
0. Logout
-----

... 9
Congratulations! You are eligible

# __ What would you like to do? __
-----
1. Deposit
2. Withdrawal
3. View balance
4. Mini statement
5. Active services
6. Bank details
7. Change password
8. Take or give loan
9. Check loan eligibility
0. Logout
-----

... 8
-----
1.take loan
2.give loan
-----

... 1
Approved loan amount is : 20000.0
if you want to continue give positive number :) ... 5
Loan Balance updated successfully for account number KKC83479

# __ What would you like to do?
```

Fig 9.11:If you Eligible for loan

9.12 Exit

```
# __ What would you like to do? __
-----
1. Deposit
2. Withdrawal
3. View balance
4. Mini statement
5. Active services
6. Bank details
7. Change password
8. Take or give loan
9. Check loan eligibility
0. Logout
-----

... 0
-----

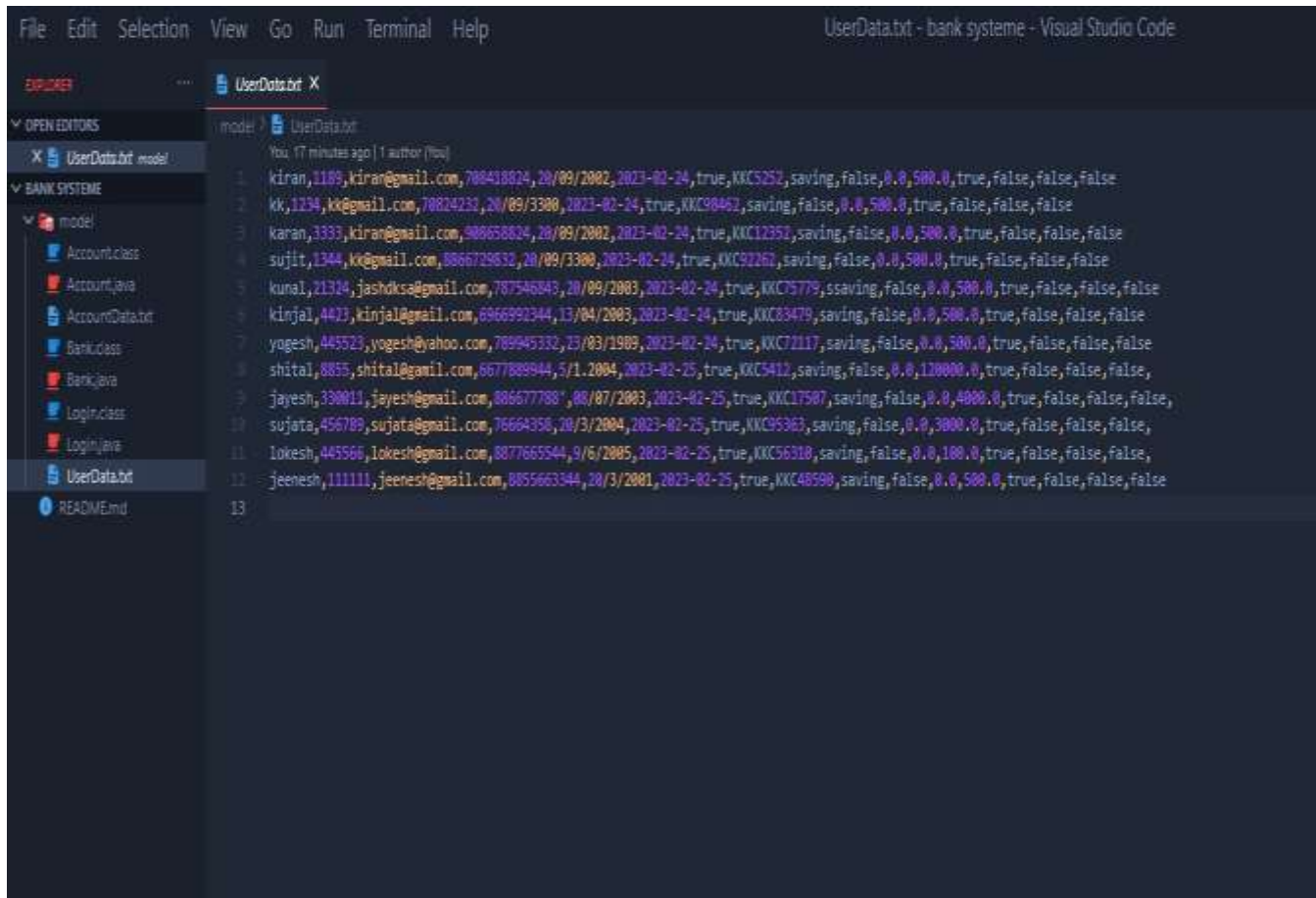
___ Welcome to the bank! ___
1. Login
2. Create new account
0. Exit
-----

... 0
----- Goodbye! :) -----
```

Fig 9.12:Exit

9.13 After Performing different Transaction updated data in files

9.13A Updated data in UserData.txt



```

File Edit Selection View Go Run Terminal Help
UserData.txt - bank systeme - Visual Studio Code

EXPLORER
OPEN EDITORS
  X UserData.txt model
BANK SYSTEME
  model
    Account.class
    Account.java
    AccountData.txt
    Bank.class
    Bank.java
    Login.class
    Login.java
    UserData.txt
    README.md

model > UserData.txt
You: 17 minutes ago | 1 author (You)
1 kiran,1189,kiran@gmail.com,788438824,28/09/2002,2023-02-24,true,KKC5252,saving,false,0.0,500.0,true,false,false,false
2 kk,1134,kk@gmail.com,78824232,20/09/3300,2023-02-24,true,KKC98462,saving,false,0.0,500.0,true,false,false,false
3 karan,3333,kiran@gmail.com,988658824,28/09/2002,2023-02-24,true,KKC12352,saving,false,0.0,500.0,true,false,false,false
4 sujit,1344,kk@gmail.com,8866729832,20/09/3300,2023-02-24,true,KKC92262,saving,false,0.0,500.0,true,false,false,false
5 kunal,21324,jashoksa@gmail.com,787546843,20/09/2003,2023-02-24,true,KKC75779,ssaving,false,0.0,500.0,true,false,false,false
6 kinjal,4423,kinjal@gmail.com,6966992344,13/04/2003,2023-02-24,true,KKC83479,saving,false,0.0,500.0,true,false,false,false
7 yogesh,445523,yogesh@yahoo.com,789945332,23/03/1909,2023-02-24,true,KKC72117,saving,false,0.0,500.0,true,false,false,false
8 shital,8855,shital@gmail.com,6677889944,5/1.2004,2023-02-25,true,KKC5412,saving,false,0.0,120000.0,true,false,false,false,
9 jayesh,330011,jayesh@gmail.com,886677788,08/07/2003,2023-02-25,true,KKC17507,saving,false,0.0,4000.0,true,false,false,false,
10 sujata,456789,sujata@gmail.com,76664358,20/3/2004,2023-02-25,true,KKC95363,saving,false,0.0,3000.0,true,false,false,false,
11 lokesh,445566,lokesh@gmail.com,8877665544,9/6/2005,2023-02-25,true,KKC56310,saving,false,0.0,100.0,true,false,false,false,
12 jeenesh,111111,jeenesh@gmail.com,8855663344,20/3/2001,2023-02-25,true,KKC48590,saving,false,0.0,500.0,true,false,false,false
13
  
```

Fig 9.13A: Data updated by isUserExist() method, createAccount() method, updateBalance method in UserData.txt

9.13B Updated Data in AccountData.txt

```

File Edit Selection View Go Run Terminal Help
AccountData.txt - bank system - Visual Studio Code

EXPLORER
  OPEN EDITORS
    AccountData.txt model
  BANK SYSTEM
    model
      AccountData.txt
      Bank.java
      Login.java
      README.md

model / AccountData.txt
14 KKC3479,Withdrawal,200.0,2023-02-24T20:30:10.56383200,balance : 12300.0
15 KKC3479,Deposit,500.0,2023-02-24T21:53:34.50810900,balance : 12300.0
16 KKC3479,Withdrawal,200.0,2023-02-24T21:53:34.52454700,balance : 12300.0
17 KKC3479,Deposit,500.0,2023-02-24T21:55:20.83924400,balance : 12300.0
18 KKC3479,Withdrawal,200.0,2023-02-24T21:55:20.85975600,balance : 12300.0
19 KKC3479,Deposit,500.0,2023-02-24T21:24:09.79400300,balance : 20500.0
20 KKC3479,Withdrawal,200.0,2023-02-24T22:14:01.83277700,balance : 20300.0
21 KKC3479,Deposit,500.0,2023-02-24T22:20:29.09680200,balance : 20500.0
22 KKC3479,Withdrawal,200.0,2023-02-24T22:24:30.01067000,balance : 20300.0
23 KKC3479,Deposit,300.0,2023-02-24T22:27:34.611806500,balance : 20600.0
24 KKC3479,Withdrawal,200.0,2023-02-24T22:27:34.64262900,balance : 20400.0
25 KKC3479,Deposit,300.0,2023-02-24T22:27:50.71331700,balance : 20400.0
26 KKC3479,Withdrawal,200.0,2023-02-24T22:27:50.801370,balance : 20200.0
27 KKC3479,Deposit,300.0,2023-02-24T22:27:54.01130000,balance : 20500.0
28 KKC3479,Withdrawal,200.0,2023-02-24T22:27:54.03130300,balance : 20300.0
29 KKC3479,Deposit,300.0,2023-02-24T22:27:58.14967900,balance : 20600.0
30 KKC3479,Withdrawal,200.0,2023-02-24T22:27:58.17043400,balance : 20400.0
31 KKC3479,Deposit,300.0,2023-02-24T22:28:01.01402100,balance : 20700.0
32 KKC3479,Withdrawal,200.0,2023-02-24T22:28:01.90016300,balance : 20500.0
33 KKC3479,Loan ,20000.0,2023-02-24T22:28:16.03576400,balance : 20500.0
34 KKC3479,Deposit,300.0,2023-02-24T22:28:16.07262600,balance : 20800.0
35 KKC3479,Withdrawal,200.0,2023-02-24T22:28:20.00065900,balance : 20600.0
36 KKC3479,Loan ,20000.0,2023-02-24T22:33:06.32450500,balance : 20600.0
37 KKC3479,Deposit,300.0,2023-02-24T22:33:06.545163,balance : 20900.0
38 KKC3479,Withdrawal,200.0,2023-02-24T22:33:06.56240000,balance : 20700.0
39 KKC3479,Loan ,20000.0,2023-02-24T22:37:04.01574500,balance : 20700.0
40 KKC3479,Deposit,300.0,2023-02-24T22:37:04.03016500,balance : 21000.0
41 KKC3479,Withdrawal,200.0,2023-02-24T22:37:04.05022100,balance : 20800.0
42 KKC3479,Deposit,300.0,2023-02-24T22:38:51.22764900,balance : 20800.0
43 KKC3479,Withdrawal,200.0,2023-02-24T22:38:51.15167100,balance : 20600.0
44 KKC3479,Loan ,20000.0,2023-02-24T22:39:23.67023700,balance : 20600.0
45 KKC3479,Deposit,300.0,2023-02-24T22:39:23.70006700,balance : 20400.0
46 KKC3479,Withdrawal,200.0,2023-02-24T22:39:23.720934,balance : 20200.0
47 KKC3479,Loan ,20000.0,2023-02-24T22:40:53.75066100,balance : 20000.0
48 KKC3479,Loan ,20000.0,2023-02-24T22:43:15.62543900,balance : 20000.0
49 KKC12351,Pin changed ,0.0,2023-02-25T00:33:04.07504000,balance : 40000.0
50 KKC48930,Deposit,150.0,2023-02-25T13:09:43.73820600,balance : 1150.0
51 KKC48930,Withdrawal,150.0,2023-02-25T13:10:34.61067800,balance : 1000.0
52 KKC48930,Withdrawal,500.0,2023-02-25T13:11:10.54645000,balance : 500.0
53 KKC48930,Pin changed ,0.0,2023-02-25T13:35:13.03614100,balance : 500.0
54 KKC3479,Loan ,20000.0,2023-02-25T14:04:04.59196100,balance : 30000.0
55 KKC3479,Loan ,20000.0,2023-02-25T14:13:46.61478000,balance : 30000.0
56 KKC3479,Loan ,20000.0,2023-02-25T14:19:16.09591300,balance : 30000.0
57 KKC3479,Loan ,20000.0,2023-02-25T14:22:07.51977000,balance : 50000.0
58 KKC3479,Loan cleared ,20000.0,2023-02-25T14:24:04.59425500,balance : 50000.0
59 KKC3479,Loan taken ,20000.0,2023-02-25T14:25:41.065454,balance : 10000.0
60 KKC3479,Loan cleared ,20000.0,2023-02-25T14:26:29.36411100,balance : 30000.0
61 KKC3479,Loan taken ,20000.0,2023-02-25T14:32:22.64067600,balance : 30000.0
62

```

Fig 9.13B: Data updated by saveTransactionDetails, getAccount(), getBalance(), isLoanActive(), and isChqueBook(),readLastNTransactions methods in AccountData.txt

10.CONCLUSION

We have implemented the Simple Bank Management System successfully. We used Core Java as programming language and class, objects, constructors, all JDBC concepts.

The system is very good at saving the precious time and efforts of account users as well as reduces the work load of Bank staff. It is a modern method of Online Banking widely accepted and hastily increasing environment of conducting managing Bank Transactions through software. It will considerably replace the traditional paper and pen-based information storing, Queue methods with Simple Bank Management System Application.

11.REFERENCES

- [1] <https://leetcode.com/problems/simple-bank-system/>
- [2] <https://github.com/topics/basic-banking-system>
- [3] <https://ieeexplore.ieee.org/document/8077021>
- [4] <https://www.engpaper.com/mobile-banking.htm>