DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING

DATABASE MANAGEMENT SYSTEMS
(CS331P)

B. Tech in Information Technology

School of Engineering and Technology, CHRIST (Deemed to be University),

Kumbalagodu, Bengaluru-560 074

Nov 2020

REPORT

On

DESIGN AND IMPLEMENTATION OF BANKING SYSTEM

By

ADARSH SINGH NISHEN 1961055

SARTHAK SWAPNIL DUBEY 1961056

REHMAN AHMAD 1961057

JOSNA MARIA JOSE 1961058

CONTENTS

INTRODUCTION

Banking is one of the most regulated businesses in the world. It is vital system for developing economy for the nation.

It is the place where customers feel the sense of safety for their property. In the bank, customers deposit and withdraw their money. Transaction of money also is a part where customer takes shelter of the bank. Now to keep the belief and trust of customers, there is the positive need for management of the bank, which can handle all this with comfort and ease. Smooth and efficient management affects the satisfaction of the customers and staff members, indirectly. And of course, it encourages management committee in taking some needed decision for future enhancement of the bank.

ABOUT THE PROJECT

The project that we have undertaken aims to develop a banking system that is clean, user-friendly and multi-functional. Development of this application includes a number of fields such that the user feels comfortable and the system appears as dynamic to him. The project "Banking System" includes the following functionalities.

- Create an account: A user can create an account by providing the name of the account holder, account number, select amount type whether its Saving account or Current account and providing an initial amount.
- Deposit amount: The user can deposit money just by providing his/her account, then the system displays his/her profile and entering an amount.
- Withdraw amount: The user can also withdraw money just by providing his/her account, then the system displays his/her profile and entering an amount.
- Balance Enquiry: For certain purpose, he/she can also check for the balance inquiry which displays the account holder's name with account number type and amount.
- List account holder's detail: He/she can also check for all the account holder's list. Another feature is that the user can also close their account by providing their account number and he/she can modify their account detail and type if they want to.

RULES GOVERNING THE PROJECT

- All the customers of the bank have a **unique** account number.
- The account numbers are not nullable i.e., they cannot take **null** values.
- A person is eligible to get a loan from the bank if he has an account in the bank.
- The percentage of interest imposed on the loan depends on the Company's policies.

ER Diagram

ER diagram of Bank has the following description :

- Bank have Customer.
- Banks are identified by a name, code, address of main office.
- Banks have branches.
- Branches are identified by a branch_no., branch_name, address.
- Customers are identified by name, cust-id, phone number, address.
- Customer can have one or more accounts.
- Accounts are identified by acc_no., acc_type, balance.
- Customer can avail loans.
- Loans are identified by loan_id, loan_type and amount.
- Account and loans are related to bank's branch.

Entities and their Attributes are :

Bank Entity : Attributes of Bank Entity are Bank Name, Code and Address.

Code is Primary Key for Bank Entity.

Customer Entity : Attributes of Customer Entity are Customer_id, Name, Phone Number and Address.

Customer_id is Primary Key for Customer Entity.

Branch Entity : Attributes of Branch Entity are Branch_id, Name and Address.

Branch_id is Primary Key for Branch Entity.

Account Entity : Attributes of Account Entity are Account_number, Account_Type and Balance.

Account_number is Primary Key for Account Entity.

Loan Entity : Attributes of Loan Entity are Loan_id, Loan_Type and Amount.

Loan_id is Primary Key for Loan Entity.

Relationships are :

Bank has Branches => 1 : N

One Bank can have many Branches but one Branch can not belong to many Banks, so the relationship between Bank and Branch is one to many relationship.

Branch maintain Accounts => 1 : N

One Branch can have many Accounts but one Account can not belong to many Branches, so the relationship between Branch and Account is one to many relationship.

Branch offer Loans => 1 : N

One Branch can have many Loans but one Loan can not belong to many Branches, so the relationship between Branch and Loan is one to many relationship.

Account held by Customers => M : N

One Customer can have more than one Accounts and also One Account can be held by one or more Customers, so the relationship between Account and Customers is many to many relationship.

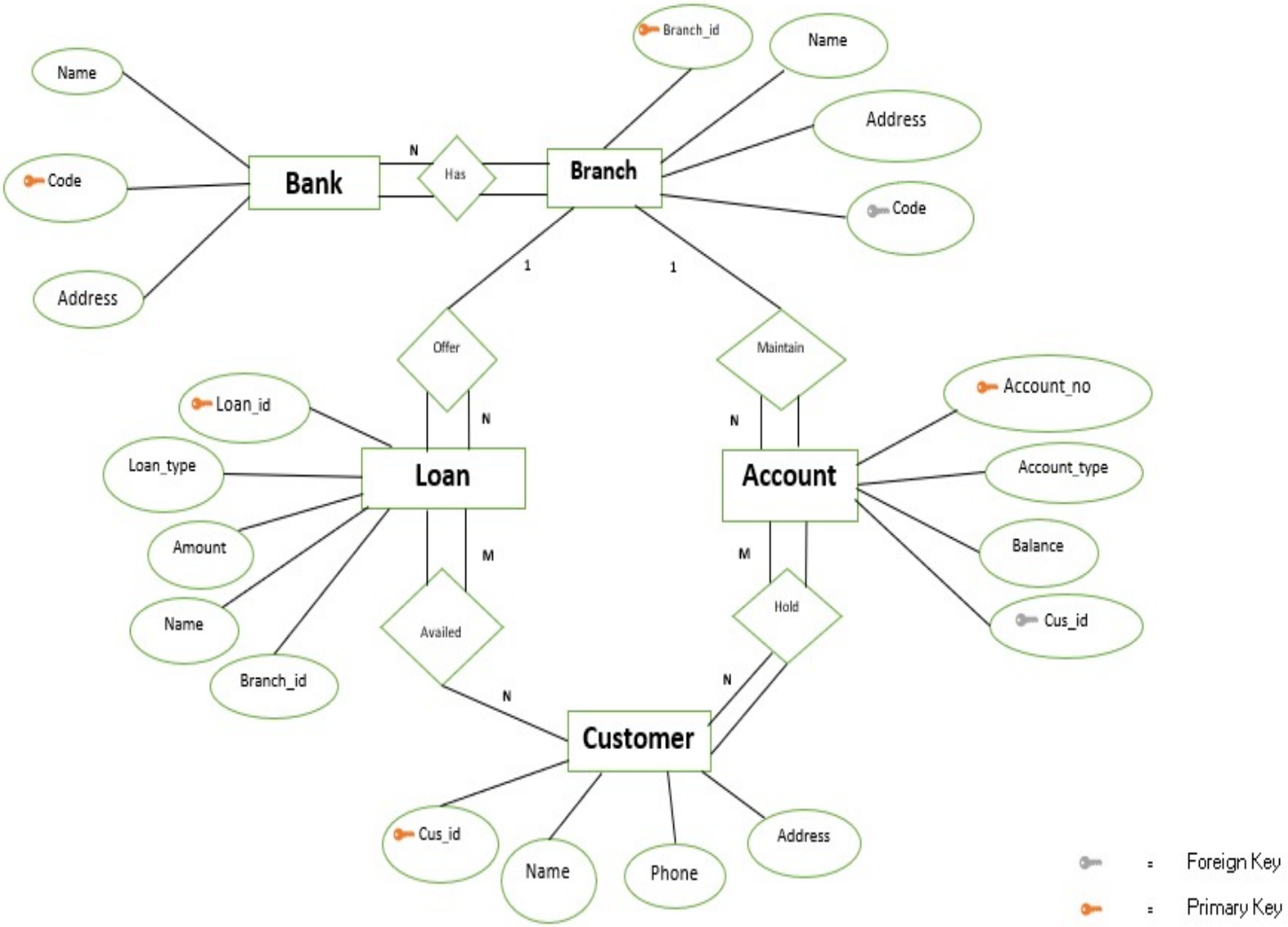
Loan availed by Customer => M : N

(Assume loan can be jointly held by many Customers).

One Customer can have more than one Loans and also One Loan can be availed by one or more Customers, so the relationship between Loan and Customers is many to many relationship.

The following bank ER diagram illustrates key information about bank, including entities such as branches, customers, accounts, and loans. It allows us to understand the relationships between entities.

BANKING SYSTEM



ER Diagram Converted into Relations

TABLE DESCRIPTION

```
project=# \d bank;
                    Table "public.bank"
 Column  |         Type          | Collation | Nullable | Default
---------+-----------------------+-----------+----------+---------
 name    | character varying(20) |           |          |
 code    | integer               |           | not null |
 address | character varying(30) |           |          |
Indexes:
    "bank_pkey" PRIMARY KEY, btree (code)
Referenced by:
    TABLE "branch" CONSTRAINT "branch_code_fkey" FOREIGN KEY (code) REFERENCES bank(code)


project=# \d branch;
                    Table "public.branch"
  Column   |         Type          | Collation | Nullable | Default
-----------+-----------------------+-----------+----------+---------
 branch_id | integer               |           | not null |
 name      | character varying(20) |           |          |
 address   | character varying(30) |           |          |
 code      | integer               |           |          |
Indexes:
    "branch_pkey" PRIMARY KEY, btree (branch_id)
Foreign-key constraints:
    "branch_code_fkey" FOREIGN KEY (code) REFERENCES bank(code)


project=# \d customer;
                   Table "public.customer"
 Column  |         Type          | Collation | Nullable | Default
---------+-----------------------+-----------+----------+---------
 cus_id  | integer               |           | not null |
 name    | character varying(20) |           |          |
 phone   | numeric(14,0)         |           |          |
 address | character varying(30) |           |          |
Indexes:
    "customer_pkey" PRIMARY KEY, btree (cus_id)
Referenced by:
    TABLE "account" CONSTRAINT "account_cus_id_fkey" FOREIGN KEY (cus_id) REFERENCES customer(cus_id)
```

```
project=# \d account;
                    Table "public.account"
    Column    |         Type          | Collation | Nullable | Default
--------------+-----------------------+-----------+----------+---------
 account_no   | integer               |           | not null |
 account_type | character varying(20) |           |          |
 balance      | integer               |           |          |
 cus_id       | integer               |           |          |
 branch_id    | integer               |           |          |
Indexes:
    "account_pkey" PRIMARY KEY, btree (account_no)
Foreign-key constraints:
    "account_cus_id_fkey" FOREIGN KEY (cus_id) REFERENCES customer(cus_id)


project=# \d loan;
                     Table "public.loan"
  Column   |         Type          | Collation | Nullable | Default
-----------+-----------------------+-----------+----------+---------
 loan_id   | integer               |           | not null |
 loan_type | character varying(20) |           |          |
 amount    | numeric(12,2)         |           |          |
 name      | character varying(20) |           |          |
 branch_id | integer               |           |          |
Indexes:
    "loan_pkey" PRIMARY KEY, btree (loan_id)
```

DATA IN TABLES

```
project=# select * from bank;
 name | code |          address
------+------+---------------------------
 SBI  |   11 | Nariman_Point_Mumbai
 HDFC |   22 | Worli_Mumbai
 PNB  |   33 | Sector10_Dwarka_New Delhi
 AXIS |   44 | Worli_Mumbai
(4 rows)


project=# select * from branch;
 branch_id |       name        |        address        | code
-----------+-------------------+-----------------------+------
      1124 | SBI Kolkata       | Kolkata Main          |   11
      1127 | SBI Borivili      | Borivili West         |   11
      1158 | SBI Dehradun      | Dehradun Main         |   11
      2211 | HDFC Adarsh Nagar | Adarsh Nagar Delhi    |   22
      2243 | HDFC Calicut      | G H Roda Calicut      |   22
      2278 | HDFC Assam        | Guwahati              |   22
      3389 | PNB Delhi         | Greater Kailash Delhi |   33
      3357 | PNB Gurgaon       | M G Road Gurgaon      |   33
      4423 | AXIS West Bengal  | Akra Fatak Kolkata    |   44
      4477 | AXIS Maharastra   | Byculla Mumbai        |   44
(10 rows)


project=# select * from customer;
 cus_id |  name   |   phone    | address
--------+---------+------------+---------
      1 | Sarthak | 9856733456 | Delhi
      2 | Joshna  | 8865472217 | Calicut
      3 | Adarsh  | 7762864320 | Kolkata
      4 | Rehmaan | 6574893023 | Mumbai
(4 rows)
```

```
project=# select * from account;
 account_no |    account_type    | balance | cus_id | branch_id
------------+--------------------+---------+--------+-----------
       1687 | checking account   |   28000 |      1 |      3389
       1710 | credit account     |   26000 |      1 |      1158
       2911 | investment account |   55000 |      2 |      4423
       2662 | checking account   |   24000 |      2 |      2278
       2877 | deposit account    |   27000 |      2 |      1124
       3943 | investment account |   49000 |      3 |      2243
       3891 | deposit account    |   17000 |      3 |      1127
       3799 | credit account     |   21000 |      3 |      3357
       4855 | deposit account    |   35000 |      4 |      4477
       4745 | credit account     |   29000 |      4 |      2211
(10 rows)


project=# select * from loan;
 loan_id |      loan_type      |  amount   |  name   | branch_id
---------+---------------------+-----------+---------+-----------
     227 | Credit Card Loan    |  17000.00 | Sarthak |      1158
     456 | Two Wheeler Loan    |  60000.00 | Sarthak |      3389
     101 | Small Business Loan | 870000.00 | Adarsh  |      4423
     499 | Credit Card Loan    |  23000.00 | Adarsh  |      2278
     789 | Home Loan           | 300000.00 | Adarsh  |      1124
     201 | Small Business Loan | 500000.00 | Joshna  |      2243
     435 | Car Loan            | 400000.00 | Joshna  |      1127
     765 | Personal Loan       |  56000.00 | Joshna  |      3357
     666 | Home Loan           | 750000.00 | Rehmaan |      4477
     555 | Personal Loan       |  50000.00 | Rehmaan |      2211
(10 rows)
```

POSSIBLE QUERIES TO SHOW THE WORKING OF THE SYSTEM

1. *Selecting customer with minimum balance*

```
SELECT * FROM CUSTOMER, ACCOUNT

WHERE ACCOUNT.BALANCE =

(

SELECT MIN(BALANCE) FROM ACCOUNT

)

AND CUSTOMER.CUS_ID=ACCOUNT.CUS_ID;
```

```
Project=# select * from customer,account
Project-# where account.balance=(select min(balance) from account) and customer.cus_id = account.cus_id;
 cus_id |  name  |   phone    | address | account_no |  account_type   | balance | cus_id | branch_id
--------+--------+------------+---------+------------+-----------------+---------+--------+-----------
      3 | Adarsh | 7762864320 | Kolkata |       3891 | deposit account |   17000 |      3 |      1127
(1 row)
```

2. *Selecting customer with maximum balance*

```
SELECT * FROM CUSTOMER, ACCOUNT

WHERE ACCOUNT.BALANCE =

(

SELECT MAX (BALANCE) FROM ACCOUNT

)

AND CUSTOMER.CUS_ID=ACCOUNT.CUS_ID;
```

```
Project=# select * from customer,account
Project-# where account.balance=(select max(balance) from account) and customer.cus_id = account.cus_id;
 cus_id |  name  |   phone    | address | account_no |    account_type    | balance | cus_id | branch_id
--------+--------+------------+---------+------------+--------------------+---------+--------+-----------
      2 | Joshna | 8865472217 | Calicut |       2911 | investment account |   55000 |      2 |      4423(1 row)
```

3. *Selecting loans taken by each customer*

```
SELECT CUS_ID, LOAN.NAME, ADDRESS, LOAN_ID, LOAN_TYPE,
AMOUNT
FROM CUSTOMER, LOAN
WHERE CUSTOMER.NAME = LOAN.NAME;
```

```
Project=# select cus_id,loan.name,phone,address,loan_id,loan_type,amount,branch_id from customer,loan
Project-# where customer.name = loan.name;
 cus_id |  name   |   phone    | address | loan_id |      loan_type      |  amount   | branch_id
--------+---------+------------+---------+---------+---------------------+-----------+-----------
      1 | Sarthak | 9856733456 | Delhi   |     227 | Credit Card Loan    |  17000.00 |      1158
      1 | Sarthak | 9856733456 | Delhi   |     456 | Two Wheeler Loan    |  60000.00 |      3389
      3 | Adarsh  | 7762864320 | Kolkata |     101 | Small Business Loan | 870000.00 |      4423
      3 | Adarsh  | 7762864320 | Kolkata |     499 | Credit Card Loan    |  23000.00 |      2278
      3 | Adarsh  | 7762864320 | Kolkata |     789 | Home Loan           | 300000.00 |      1124
      2 | Joshna  | 8865472217 | Calicut |     201 | Small Business Loan | 500000.00 |      2243
      2 | Joshna  | 8865472217 | Calicut |     435 | Car Loan            | 400000.00 |      1127
      2 | Joshna  | 8865472217 | Calicut |     765 | Personal Loan       |  56000.00 |      3357
      4 | Rehmaan | 6574893023 | Mumbai  |     666 | Home Loan           | 750000.00 |      4477
      4 | Rehmaan | 6574893023 | Mumbai  |     555 | Personal Loan       |  50000.00 |      2211
(10 rows)
```

4. *Selecting deposit accounts*

```
SELECT * FROM ACCOUNT
WHERE ACCOUNT_TYPE=’DEPOSIT ACCOUNT’;
```

```
Project=# select * from account where account_type = 'deposit account';
 account_no |  account_type   | balance | cus_id | branch_id
------------+-----------------+---------+--------+-----------
       2877 | deposit account |   27000 |      2 |      1124
       3891 | deposit account |   17000 |      3 |      1127
       4855 | deposit account |   35000 |      4 |      4477
(3 rows)
```

5. *Selecting customers with account balance more than 30000*
   SELECT * FROM CUSTOMER, ACCOUNT
   WHERE BALANCE>30000 AND
   ACCOUNT.CUS_ID=CUSTOMER.CUS_ID;

```
Project=# select * from customer, account
Project-# where balance>30000 and account.cus_id= customer.cus_id;
 cus_id |  name   |   phone    | address | account_no |    account_type    | balance | cus_id | branch_id
--------+---------+------------+---------+------------+--------------------+---------+--------+-----------
      2 | Joshna  | 8865472217 | Calicut |       2911 | investment account |   55000 |      2 |      4423
      3 | Adarsh  | 7762864320 | Kolkata |       3943 | investment account |   49000 |      3 |      2243
      4 | Rehmaan | 6574893023 | Mumbai  |       4855 | deposit account    |   33000 |      4 |      4477
(3 rows)
```

6. SELECT * FROM BANK.BRANCH
   WHERE BANK.CODE=BRANCH.CODE;

```
Project=# select * from bank,branch
Project-# where bank.code =branch.code;
 name | code |          address          | branch_id |       name        |        address        | code
------+------+---------------------------+-----------+-------------------+-----------------------+------
 SBI  |   11 | Nariman_Point_Mumbai      |      1124 | SBI Kolkata       | Kolkata Main          |   11
 SBI  |   11 | Nariman_Point_Mumbai      |      1127 | SBI Borivili      | Borivili West         |   11
 SBI  |   11 | Nariman_Point_Mumbai      |      1158 | SBI Dehradun      | Dehradun Main         |   11
 HDFC |   22 | Worli_Mumbai              |      2211 | HDFC Adarsh Nagar | Adarsh Nagar Delhi    |   22
 HDFC |   22 | Worli_Mumbai              |      2243 | HDFC Calicut      | G H Roda Calicut      |   22
 HDFC |   22 | Worli_Mumbai              |      2278 | HDFC Assam        | Guwahati              |   22
 PNB  |   33 | Sector10_Dwarka_New Delhi |      3389 | PNB Delhi         | Greater Kailash Delhi |   33
 PNB  |   33 | Sector10_Dwarka_New Delhi |      3357 | PNB Gurgaon       | M G Road Gurgaon      |   33
 AXIS |   44 | Worli_Mumbai              |      4423 | AXIS West Bengal  | Akra Fatak Kolkata    |   44
 AXIS |   44 | Worli_Mumbai              |      4477 | AXIS Maharastra   | Byculla Mumbai        |   44
(10 rows)
```

7. *Selecting customers with loan greater than 70000*

```
SELECT CUS_ID, CUSTOMER.NAME, PHONE, ADDRESS, LOAN_ID,
LOAN_TYPE, AMOUNT
FROM LOAN, CUSTOMER
WHERE CUSTOMER.NAME=LOAN.NAME AND AMOUNT>70000;
```

```
Project=# select cus_id,customer.name,phone,address,loan_id,loan_type,amount from loan,customer
Project-# where customer.name= loan.name and amount>70000;
 cus_id |  name   |   phone    | address | loan_id |      loan_type      |  amount
--------+---------+------------+---------+---------+---------------------+-----------
      3 | Adarsh  | 7762864320 | Kolkata |     101 | Small Business Loan | 870000.00
      3 | Adarsh  | 7762864320 | Kolkata |     789 | Home Loan           | 300000.00
      2 | Joshna  | 8865472217 | Calicut |     201 | Small Business Loan | 500000.00
      2 | Joshna  | 8865472217 | Calicut |     435 | Car Loan            | 400000.00
      4 | Rehmaan | 6574893023 | Mumbai  |     666 | Home Loan           | 750000.00
(5 rows)
```

8. *Selecting all details of all customers*

```
SELECT * FROM CUSTOMER, BANK, ACCOUNT, BRANCH
WHERE ACCOUNT.CUS_ID = CUSTOMER.CUS_ID
AND BRANCH.BRANCH_ID=ACCOUNT.BRANCH_ID
AND BRANCH.CODE=BANK.CODE;
```

```
Project=# select * from customer,bank, account,branch
Project-# where account.cus_id = customer.cus_id and branch.branch_id = account.branch_id and branch.code = bank.code;
 cus_id |  name   |   phone    | address | name | code |          address          | account_no |    account_type    | balance | cus_id | branch_id | branch_id |       name        |        address        | code
--------+---------+------------+---------+------+------+---------------------------+------------+--------------------+---------+--------+-----------+-----------+-------------------+-----------------------+------
      1 | Sarthak | 9856733456 | Delhi   | PNB  |   33 | Sector10_Dwarka_New Delhi |       1687 | checking account   |   28000 |      1 |      3389 |      3389 | PNB Delhi         | Greater Kailash Delhi |   33
      2 | Joshna  | 8865472217 | Calicut | AXIS |   44 | Worli_Mumbai              |       2911 | investment account |   55000 |      2 |      4423 |      4423 | AXIS West Bengal  | Akra Fatak Kolkata    |   44
      2 | Joshna  | 8865472217 | Calicut | HDFC |   22 | Worli_Mumbai              |       2662 | checking account   |   24000 |      2 |      2278 |      2278 | HDFC Assam        | Guwahati              |   22
      2 | Joshna  | 8865472217 | Calicut | SBI  |   11 | Nariman_Point_Mumbai      |       2877 | deposit account    |   27000 |      2 |      1124 |      1124 | SBI Kolkata       | Kolkata Main          |   11
      3 | Adarsh  | 7762864320 | Kolkata | HDFC |   22 | Worli_Mumbai              |       3943 | investment account |   49000 |      3 |      2243 |      2243 | HDFC Calicut      | G H Roda Calicut      |   22
      3 | Adarsh  | 7762864320 | Kolkata | SBI  |   11 | Nariman_Point_Mumbai      |       3891 | deposit account    |   17000 |      3 |      1127 |      1127 | SBI Borivili      | Borivili West         |   11
      3 | Adarsh  | 7762864320 | Kolkata | PNB  |   33 | Sector10_Dwarka_New Delhi |       3799 | credit account     |   21000 |      3 |      3357 |      3357 | PNB Gurgaon       | M G Road Gurgaon      |   33
      4 | Rehmaan | 6574893023 | Mumbai  | HDFC |   22 | Worli_Mumbai              |       4745 | credit account     |   29000 |      4 |      2211 |      2211 | HDFC Adarsh Nagar | Adarsh Nagar Delhi    |   22
      4 | Rehmaan | 6574893023 | Mumbai  | AXIS |   44 | Worli_Mumbai              |       4855 | deposit account    |   33000 |      4 |      4477 |      4477 | AXIS Maharastra   | Byculla Mumbai        |   44
      1 | Sarthak | 9856733456 | Delhi   | SBI  |   11 | Nariman_Point_Mumbai      |       1710 | credit account     |   27000 |      1 |      1158 |      1158 | SBI Dehradun      | Dehradun Main         |   11
(10 rows)
```

9. *Transanction to update balance of Customers with Account number 4855 and 1710.*

```
BEGIN;
UPDATE ACCOUNT
SET
BALANCE=BALANCE-1000
WHERE ACCOUNT_NO=4855;
UPDATE ACCOUNT
SET
BALANCE=BALANCE+1000
WHERE ACCOUNT_NO=1710;
COMMIT;
SELECT * FROM ACCOUNT;
```

```
Project=# begin;
BEGIN
Project=# update account
Project-# set
Project-# balance = balance-1000
Project-# where account_no=4855;
UPDATE 1
Project=# update account
Project-# set
Project-# balance = balance+1000
Project-# where account_no=1710;
UPDATE 1
Project=# commit;
COMMIT
Project=# select * from account;
 account_no |    account_type    | balance | cus_id | branch_id
------------+--------------------+---------+--------+-----------
       1687 | checking account   |   28000 |      1 |      3389
       2911 | investment account |   55000 |      2 |      4423
       2662 | checking account   |   24000 |      2 |      2278
       2877 | deposit account    |   27000 |      2 |      1124
       3943 | investment account |   49000 |      3 |      2243
       3891 | deposit account    |   17000 |      3 |      1127
       3799 | credit account     |   21000 |      3 |      3357
       4745 | credit account     |   29000 |      4 |      2211
       4855 | deposit account    |   33000 |      4 |      4477
       1710 | credit account     |   27000 |      1 |      1158
(10 rows)


Project=#
```

10. *Transaction to update balance of one customer and delete account of another customer*

```
BEGIN;
 UPDATE ACCOUNT
 SET
BALANCE=BALANCE + (
                        SELECT BALANCE FROM ACCOUNT
                        WHERE ACCOUNT_NO=2911
                    )
WHERE ACCOUNT_NO = 2662;
DELETE FROM ACCOUNT WHERE ACCOUNT_NO=2911;
COMMIT;
```

```
Project=# begin;
BEGIN
Project=# update account
Project-# set
Project-# balance = balance+(select balance from account where account_no =2911)
Project-# where account_no = 2662;
UPDATE 1
Project=# delete from account where account_no = 2911;
DELETE 1
Project=# commit;
COMMIT
Project=# select * from account;
 account_no |    account_type    | balance | cus_id | branch_id
------------+--------------------+---------+--------+-----------
       1687 | checking account   |   28000 |      1 |      3389
       2877 | deposit account    |   27000 |      2 |      1124
       3943 | investment account |   49000 |      3 |      2243
       3891 | deposit account    |   17000 |      3 |      1127
       3799 | credit account     |   21000 |      3 |      3357
       4745 | credit account     |   29000 |      4 |      2211
       4855 | deposit account    |   33000 |      4 |      4477
       1710 | credit account     |   27000 |      1 |      1158
       2662 | checking account   |   79000 |      2 |      2278
(9 rows)


Project=#
```

FRONT END

Bank System is based on a concept of recording customer's account details.

Here the user can perform all the tasks like creating an account, deposit amount, withdraw amount, check balance, view all account holders' detail, close an account and modify an account.

There's no login system for this project.

Synopsis

Project Name: Banking System = C++ + SQL

Features:

1.Create an account: A user can create an account by providing the name of the account holder, account number, select amount type whether its Saving account or Current account and providing an initial amount.

2.Deposit amount: The user can deposit money just by providing his/her account, then the system displays his/her profile and entering an amount.

3.Withdraw amount: The user can also withdraw money just by providing his/her account, then the system displays his/her profile and entering an amount.

4.Balance Enquiry: For certain purpose, he/she can also check for the balance inquiry which displays the account holder's name with account number type and amount.

5.List account holder's detail: He/she can also check for all the account holder's list. Another feature is that the user can also close their account by providing their account number and he/she can modify their account detail and type if they want to.

C++:

1.Classes

2.File Handling

In order to store all the user's data, an external file (DAT file) is created by the system, so every time we get into the system we can operate with the existing accounts.

Bank System is developed using C++ Programming Language and different variables, strings have been used for the development of it.

Recommended IDEs:

1.Dev C++

2.Code Blocks

Software Requirements:

1. C++

2. SQL

Hardware Requirements:

1. Processor: 1.8 GHz or faster

2. RAM: 2 GB of RAM; 8 GB of RAM recommended (2.5 GB minimum if running on a virtual machine)

3. Hard disk space: Minimum of 800MB up to 210 GB of available space, depending on features installed; typical installations require 20-50 GB of free space.

REFERENCES

https://www.slideshare.net/unsajawaid/documentation-on-bank-management-system

https://www.researchgate.net/publication/301293322_Bank_Account_Management_System