**BANKING APP**

A

Mini Project Report

*Submitted in partial fulfilment of the*

*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

**By**

RAKESH HOTKER-1602-19-737-151

KAMASANI RAGHUPATHI-1602-19-737-148



**Department of Information Technology**

**Vasavi College of Engineering(Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahim Bagh,Hyderabad-31**

**2020**

**Vasavi College of Engineering(Autonomous)**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**

**DECLARATION BY THE CANDIDATE**

We, RAKESH HOTKER AND KAMSANI RAGHUPATHI bearing hall ticket

Numbers, 1602-19-737-151 and 1602-19-737-148,hereby declare that the project report

Entitled “BANKING APP” is submitted in partial fulfilment of the requirement for the award of the

Degree of Bachelor Of Engineering in Information Technology.

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma

RAKESH HOTKER

1602-19-737-151

KAMASANI RAGHUPATHI

1602-19-737-148

(Faculty In-Charge) (Head.Dept IT)

**ACKNOWLEDGEMENTS**

We are overwhelmed in all humbleness and gratefulness to acknowledge our debt to all those

Who have helped us put these ideas, well above the level of simplicity and into something concrete. We would like to thank our college who gave us the golden opportunity to do this wonderful project “Banking Application” which helped us in learning many new things.

We are really obliged to them.

We wish to express our affectionate gratitude to our friend Siddharth for his idea of using structures.

RAKESH HOTKER

1602-19-737-151

KAMASANI RAGHUPATHI

1602-19-737-148

Date:19-12-2020

**ABSTRACT**

This Mini Project in C, tries to mimic the way finance works in today’s world by allowing users to deposit,withdraw, transfer funds which in real are just numbers typed in by the bank.

We wanted to clone services like PhonePe,Google Pay.

TABLE OF CONTENTS

1. Introduction…………………..7

1.1 Problem Domain in general

1.2 Project Introduction

1.3 List of features in the project

1. Technology

2.1 Software Requirements

2.2 Hardware Requirements

1. Proposed work

3.1 Design

3.1.1 Customer Use Cases

3.1.1.1 Create New Account

3.1.1.1.1 Aadhar Verification

3.1.1.2 Login

3.1.1.3 Deposit

3.1.1.4 Withdraw

3.1.1.5 Check Account Balance

3.1.1.6 Transfer Funds

3.2 Implementation

3.2.1 Module-Wise Code

3.2.1.1 Create New Account Function

3.2.1.1.1 Aadhar Verification function

3.2.1.2 Login Function

3.2.1.3 Deposit Function

3.2.1.4 Withdraw Function

3.2.1.5 Check Account Balance

3.2.1.6 Transfer Funds

3.2.2 Github Folder Structure

3.3 Testing

3.3.1 Customer Test Cases

Create New Account

Aadhar Verification

Login

Deposit

Withdraw

Check Balance

Transfer Funds

1. Results(Output Screenshots)......................27

4.1 Main Menu

4.2 Customer Test Cases

1. Additional knowledge gained…………….32
2. Conclusion and future work……………...33
3. References…...........................34

**1.INTRODUCTION**

**1.1 Problem domain in general**

Cash payments aren’t that safe anymore with the onset of covid this was clear as crystal

Digital transactions are safe and secure and also solve the transfer of decimals

In payments.

**1.2 Project Introduction**

We, tried to clone already pre-existing applications in the market and tried to build

C-version of it.

Digital currency as we know it is just virtual bond that the bank creates,so our application

Tries to hold bank as a security entity.

**1.3 List of features in the Project**

Our Project tries to look at the customer side, with bank being the facilitator.

Features that customer can use are

Create an Account,using aadhar Number

Login using password

Deposit Amount

Withdraw Amount

Check account Balance

Transfer Funds to another account

**2.TECHNOLOGY**

**2.1 Software Requirements**

Operating System: windows 10

C compiler: GNU Compiler Collection

Editor: Any Editor(Code Blocks preferably)

**2.2 Hardware Requirements**

Processor: intel core i3,ryzen 3 and above

Memory: 4GB RAM and above

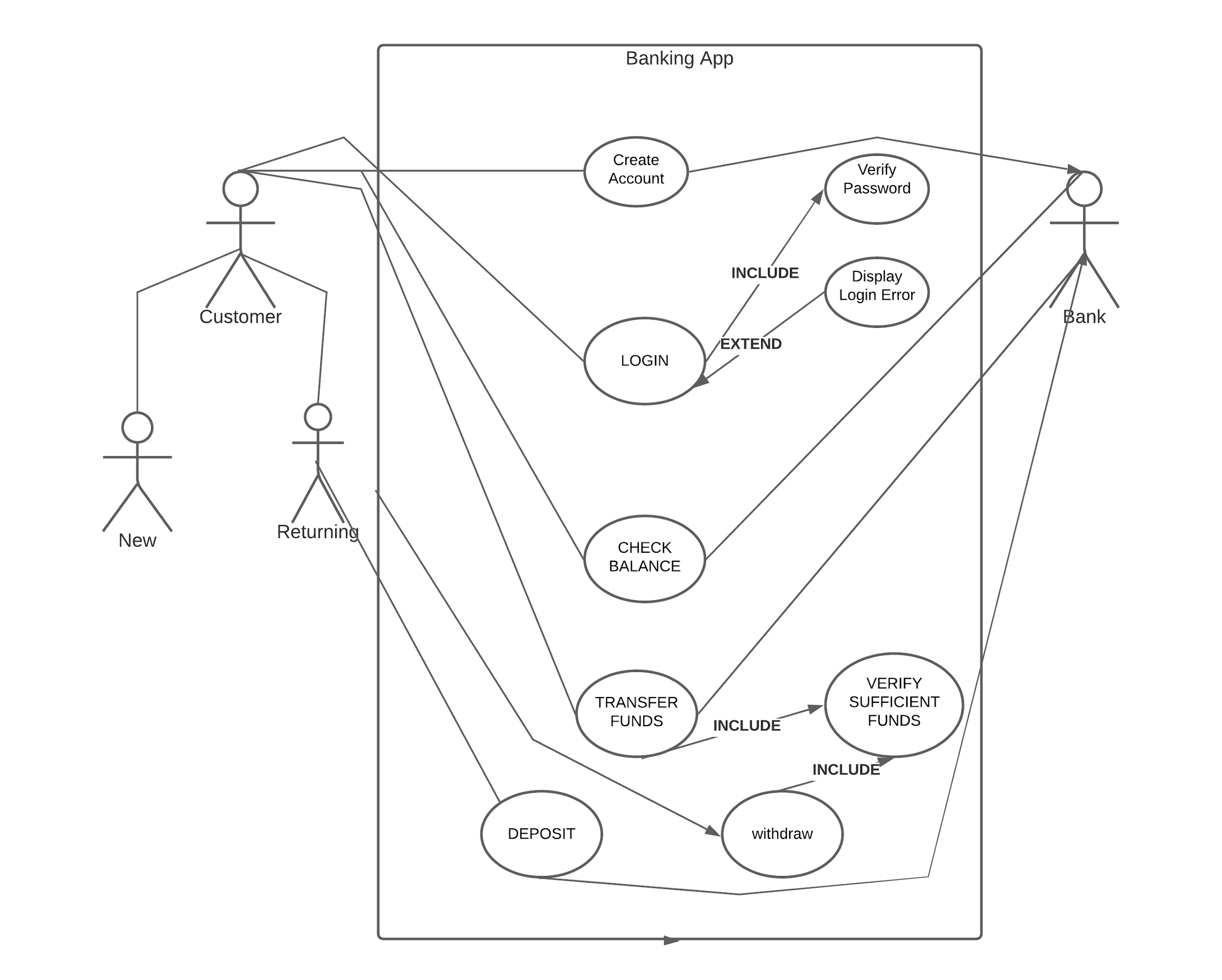
**3.PROPOSED WORK**

**3.1 DESIGN**

**3.1.1 CUSTOMER USE-CASES**

A customer can create account,login,deposit amount,withdraw amount

Transfer money,Check Balance



**3.1.1.1. Create New Account**

This function allows user to enter aadhar number and verifies the

Aadhar and if it’s correct allows user to set a password

3.1.1.1.1 Aadhar Number of the user is verified using the Verhoeff’s algorithm

For error detection.

**3.1.1.2 Login**

This function allows user to enter the account number and the password set

Above to be able to access the transaction features

**3.1.1.3 Deposit**

This function allows the user logged in, to enter the amount that he/she want to deposit into their account.

**3.1.1.4 Withdraw**

This function allows user to withdraw from their account,throws an error if the

User wants to withdraw more than what is present in his account

**3.1.1.5 Check Account Balance**

This function allows user to check their account Balance

**3.1.1.6 Transfer Funds**

This function allows user to transfer funds to other users,having the account in

The same bank(virtual)

**3.2 IMPLEMENTATION**

**3.2.1. Module-Wise Code**

**3.2.1.1 Create New Account**

**Function name:** new\_acc()

**Functionality:** This prompts the user to enter Aadhar Number and if found orrect, asks the user to set password and creates new Account

void new\_acc()

{

char aadharNumber[12];

printf("Enter your aadhar Number(Remember it's 12 digits)");

scanf("%s",&aadharNumber);

int a=validate\_verhoeff(aadharNumber);

if(a==0)

{

printf("Incorrect aadhar Number!");

menu();

}

printf("\nAadhar Verified Successfully!");

for(int i=0;i<12;i++)

{

create.aadharNumber[i]=aadharNumber[i];

}

for(int i=0;i<10;i++)

{

create.fileName[i]=create.aadharNumber[i];

}

printf("Set your password:");

for(int i=0;i<8;i++)

{

create.password[i]=getch();

printf("\*");

}

FILE \*ptr;

ptr=fopen(create.fileName,"w");

fwrite(&create,sizeof(struct customer),1,ptr);

fclose(ptr);

struct customer showcase;

FILE \*read;

read=fopen(create.fileName,"r");

printf("\n\nAccount created successfully\t(Note down your account number and remember your password for future reference)");

while(fread(&showcase,sizeof(struct customer),1,read))

{

printf("\nAccountNumber=%s",showcase.fileName);

}

fclose(read);

int choice;

printf("\nChoose 1 to return to main menu or 0 to exit:\n");

scanf("%d",&choice);

if(choice==1)

{

menu();

}

else

{

printf("Be aware bank never calls you to ask for any details\n");

exit(0);

}

}

**3.2.1.1.1 Aadhar verification Code**

**Function Name:** validate\_verhoeff()

**Functionality:** This algorithm verifies if the number entered is an aadhar Number,

If it is, then allows user to set password, else denies creation

static int verhoeff\_d[10][10] = {

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9},

{1, 2, 3, 4, 0, 6, 7, 8, 9, 5},

{2, 3, 4, 0, 1, 7, 8, 9, 5, 6},

{3, 4, 0, 1, 2, 8, 9, 5, 6, 7},

{4, 0, 1, 2, 3, 9, 5, 6, 7, 8},

{5, 9, 8, 7, 6, 0, 4, 3, 2, 1},

{6, 5, 9, 8, 7, 1, 0, 4, 3, 2},

{7, 6, 5, 9, 8, 2, 1, 0, 4, 3},

{8, 7, 6, 5, 9, 3, 2, 1, 0, 4},

{9, 8, 7, 6, 5, 4, 3, 2, 1, 0}

};

static int verhoeff\_p[8][10] = {

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9},

{1, 5, 7, 6, 2, 8, 3, 0, 9, 4},

{5, 8, 0, 3, 7, 9, 6, 1, 4, 2},

{8, 9, 1, 6, 0, 4, 3, 5, 2, 7},

{9, 4, 5, 3, 1, 2, 6, 8, 7, 0},

{4, 2, 8, 6, 5, 7, 3, 9, 0, 1},

{2, 7, 9, 3, 8, 0, 6, 4, 1, 5},

{7, 0, 4, 6, 9, 1, 3, 2, 5, 8}

};

static int verhoeff\_inv[] = {0, 4, 3, 2, 1, 5, 6, 7, 8, 9};

static int validate\_verhoeff(const char\* num)

{

int c;

int len;

c=0;

len=strlen(num);

for(int i=0;i<len;i++)

{

c=verhoeff\_d[c][verhoeff\_p[(i%8)][num[len-i-1]-'0']];

}

return (c==0);

}

**3.2.1.2 Login Function**

**Function name:** login()

**Functionality:** Allows user to login using the account Number and the password set

By the user

void login()

{

char accountNumber[20];

char password[8];

printf("\t\t\t\t\tEnter your 10 digit account Number:");

scanf("%s",&accountNumber);

printf("\t\t\t\t\tEnter your password:");

for(int i=0;i<8;i++)

{

password[i]=getch();

printf("\*");

}

FILE \*ptr;

ptr=fopen(accountNumber,"r");

fread(&retrieve,sizeof(struct customer),1,ptr);

if(strncmp(accountNumber,retrieve.fileName,sizeof(retrieve.fileName))==0 && strncmp(password,retrieve.password,sizeof(retrieve.password))==0)

{

printf("\n\t\t\t\tLogin successful\n");

int choice;

printf("\n\t\t\t\tChoose 1 to do transactions or 0 to go back to main menu:\n");

scanf("%d",&choice);

if(choice==1)

{

transact(accountNumber);

}

else

{

menu();

}

}

else

{

printf("\n\t\t\t\tWrong Credentials mate! you got to go!");

exit(0);

}

}

3.2.1.3 Deposit, 3.2.1.4 Withdraw, 3.2.1.5 Check Account Balance,

Function Name:transact()

Functionality: Allows user to deposit amount into the account,withdraw

From the account,check account balance,transfer funds

void transact(char \*accountNumber)

{

int deposit,withdraw;

FILE \*ptr;

ptr=fopen(accountNumber,"r");

if(ptr==NULL)

{

printf("Account doesn't exist!");

return;

}

fclose(ptr);

int choice;

printf("\nChoose:\n\4 1 for deposit\n\4 2 for withdraw\n\4 3 for account\_balance\n\4 4 for transferring funds\n");

scanf("%d",&choice);

switch(choice)

{

case 1:

ptr=fopen(accountNumber,"r");

printf("\nEnter the amount you want to deposit");

scanf("%d",&deposit);

fread(&create,sizeof(struct customer),1,ptr);

create.balance+=deposit;

create.deposit+=deposit;

fclose(ptr);

ptr=fopen(accountNumber,"w");

fwrite(&create,sizeof(struct customer),1,ptr);

fclose(ptr);

Break;

case 2:

ptr=fopen(accountNumber,"r");

printf("\nEnter the amount you want to withdraw");

scanf("%d",&withdraw);

fread(&create,sizeof(struct customer),1,ptr);

if(withdraw>create.balance)

{

printf("\nYou don't have sufficient balance, try something below %d",create.balance);

}

else{

create.balance-=withdraw;

printf("\n\t%d Withdrawn successfully",withdraw);

fclose(ptr);

ptr=fopen(accountNumber,"w");

fwrite(&create,sizeof(struct customer),1,ptr);

fclose(ptr);

}

break;

case 3:

ptr=fopen(accountNumber,"r");

printf("\nBalance in your account");

fread(&create,sizeof(struct customer),1,ptr);

printf("\t$%d",create.balance);

fclose(ptr);

break;

case 4:

transfer\_funds(accountNumber);

break;

}

int options;

printf("\n\t\t\t\tChoose 1 to continue transacting or 0 to return to main menu.");

scanf("%d",&options);

if(options==1)

{

transact(accountNumber);

}

else

{

menu();

}

}

3.2.1.6 Transfer Funds

Function name:transfer\_funds()

Functionality:This function allows user to transfer funds to another account

void transfer\_funds(char \* User\_accountNumber)

{

char accountNumber[10];

printf("\n\t\t\t\tEnter the account number, you want to transfer funds to.");

scanf("%s",&accountNumber);

FILE \*ptr;

ptr=fopen(accountNumber,"r");

if(ptr==NULL)

{

printf("\t\t\t\tAccount doesn't exist");

menu();

}

else

{

int amount;

printf("\n\t\t\t\tEnter the amount you want to transfer:\n");

scanf("%d",&amount);

/\*Need to check if the user account has enough balance\*/

FILE \*ptr2;

ptr2=fopen(User\_accountNumber,"r");

fread(&retrieve,sizeof(struct customer),1,ptr2);

if(retrieve.balance<=amount)

{

printf("\t\t\t\tInsufficient balance in your account:");

fclose(ptr2);

return;

}

else

{

/\*adding amount to the other user\*/

fread(&create,sizeof(struct customer),1,ptr);

create.balance+=amount;

fclose(ptr);

FILE \*ptr1;

ptr1=fopen(accountNumber,"w");

fwrite(&create,sizeof(struct customer),1,ptr);

fclose(ptr1);

/\*substracting from the current\_user\*/

retrieve.balance-=amount;

fclose(ptr2);

ptr2=fopen(User\_accountNumber,"w");

fwrite(&retrieve,sizeof(struct customer),1,ptr2);

fclose(ptr2);

}

}

int choice;

printf("\n\t\t\t\tChoose 1 to return to menu\t2 to exit");

scanf("%d",&choice);

if(choice==1)

{

menu();

}

else

{

printf("\n\t\t\t\tHave a nice day!");

}

}

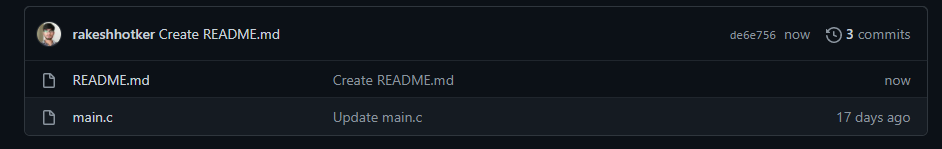
**3.2.2 GitHub structure/folder**

The project code file name main.c is in the repository named Mini\_Project

The repository also has README.md file which provides the overview of the project

**GitHub Repository link:**

<https://github.com/rakeshhotker/Mini_Project>

**3.3 Testing**

**3.3.1 Customer Test Cases**

**Create New Account**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC01 | | | Use case ID:  **UC01** |
| **Test case title**: Create New Account | | |
| **Test case description**: This case doesn’t allows user to create account | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The app prompts the user    To enter aadhar Number  User enters wrong aadhar  Number | A message saying  “Incorrect Aadhar Number”  Is displayed. | A message saying “Incorrect Aadhar Number”  Is displayed | |
|  |  |  | |
|  |  |  | |
|  |  |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC02 | | | Use case ID:  **UC01** |
| **Test case title**: Create New Account | | |
| **Test case description**: This case allows user to set password and thus creates new account | | |
| **Test steps** | **Expected result** | **Actual result** | |
| App prompts the user  To enter aadhar Number  User enters right Aadhar Number, app prompts user  To set password  User sets password. | Message saying  “Account created  Successfully” along with account number is displayed. | Message saying “Account created successfully” along  With account number is  displayed. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC03 | | | Use case ID:  **UC01** |
| **Test case title**: Create New Account | | |
| **Test case description**: This case doesn’t allow user to create New Account | | |
| **Test steps** | **Expected result** | **Actual result** | |
| User enters an aadhar  Number that pre-exists  In the records. | Message saying “Aadhar  Verified successfully, Account already exists” is shown | “Aadhar Verified Successfully, Account already exists” is displayed | |

**LOGIN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC04 | | | Use case ID:  **UC02** |
| **Test case title**: Login | | |
| **Test case description**: This case doesn’t allow the user to login | | |
| **Test steps** | **Expected result** | **Actual result** | |
|  |  |  | |
| The app prompts user to enter the account Number  User enters wrong account  Number | Message saying”Wrong  Credentials mate” is displayed on the screen | Message saying “Wrong  Credentials mate” is displayed on the screen | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC05 | | | Use case ID:  **UC02** |
| **Test case title**: Login | | |
| **Test case description**: This case doesn’t allow the user to login | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The app prompts the user  To enter the account Number and the password  User enters Correct Account Number but wrong  Password | Message saying “Wrong Credentials mate!” is displayed on the screen. | Message saying “Wrong Credentials mate” is displayed on the screen. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC06 | | | Use case ID:  **UC02** |
| **Test case title**: Login | | |
| **Test case description**: This Case allows user to Login | | |
| **Test steps** | **Expected result** | **Actual result** | |
| App prompts the user  To enter account Number  And password  User enters correct  Account Number and Correct Password. | Message saying “Login Successful”  Choose 1 to do transaction 0 to go back to main menu  Is displayed. | Message saying  “Login Successful” Choose 1  To do transaction 0 to go back to main menu is displayed. | |

**Deposit**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC07 | | | Use case ID:  **UC03** |
| **Test case title**: Deposit amount | | |
| **Test case description**: This case allows user to enter the amount  He wants to deposit. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| App prompts the user to enter the amount  User enters the amount | Message saying “amount deposit successfully” is displayed | Message saying “amount deposited successfully is displayed | |

**Withdraw**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC08 | | | Use case ID:  **UC04** |
| **Test case title**: Withdraw amount | | |
| **Test case description**: This case allows user to enter the amount  He wants to withdraw. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| App prompts the user to enter the amount  User enters the amount | Message saying “amount withdrawn successfully” is displayed. | Message saying “amount withdrawn successfully is displayed. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC09 | | | Use case ID:  **UC04** |
| **Test case title**: Withdraw amount | | |
| **Test case description**: This case doesn’t allow user to withdraw | | |
| **Test steps** | **Expected result** | **Actual result** | |
| App prompts the user to enter the amount  User enters the amount | Message saying “You don’t have sufficient balance”  Try something below account balance is displayed | Message saying “You don’t have sufficient balance”  Try something below account balance is displayed. | |

**Transfer Funds**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC10 | | | Use case ID:  **UC05** |
| **Test case title**: Transfer Funds | | |
| **Test case description**: This Case doesn’t allow user to transfer funds | | |
| **Test steps** | **Expected result** | **Actual result** | |
| App prompts the user to enter the account number to which amount should be transferred  User enters wrong account Number | Message saying “Account doesn’t exist” is displayed  On the screen. | Message saying “Account doesn’t exist” is displayed  On the screen. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC11 | | | Use case ID:  **UC05** |
| **Test case title**: Transfer Funds | | |
| **Test case description**: This case doesn’t allow user to transfer amounts | | |
| **Test steps** | **Expected result** | **Actual result** | |
| App prompts the user to enter the account number  User enters own account number. | Message saying “you have entered your own number” is displayed. | Message saying “you have entered your own number”  is displayed. | |

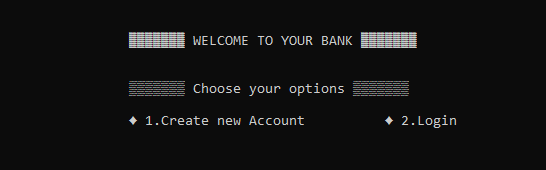
|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC12 | | | Use case ID:  **UC05** |
| **Test case title**: Transfer Funds | | |
| **Test case description**: This case allows user to transfer amounts | | |
| **Test steps** | **Expected result** | **Actual result** | |
| App prompts the user to enter the account number  User enters account number of the other user. | Message saying “Funds transferred successfully is displayed”. | Message saying “Funds transferred successfully is displayed”. | |

**Check Account Balance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID**: TC13 | | | Use case ID:  **UC06** |
| **Test case title**: Check Balance | | |
| **Test case description**: This case allows user to check the amount balance in the account. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| User chooses the option  To check balance | Displays the account balance | Displays the account balance | |

**4. Results(Output Screenshots)**

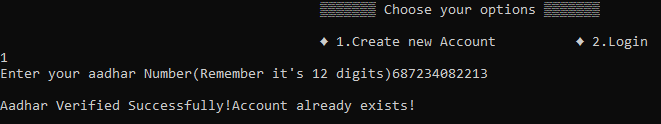
**4.1 Main Menu**



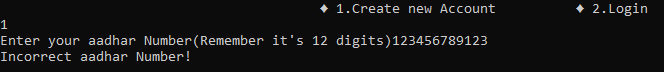
**4.2 User Test Cases**

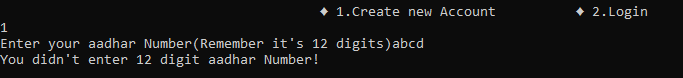
**Create New Account**

**Pre-existing Aadhar Number entered**

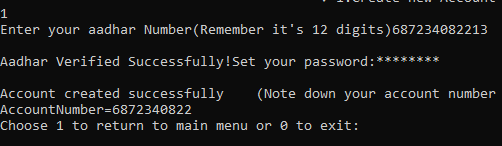


**Wrong aadhar Number**



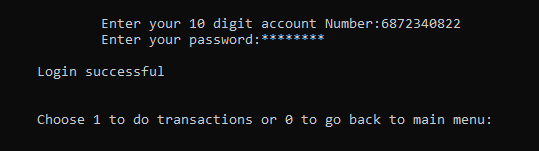


**Correct aadhar Number**

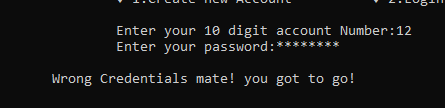


**Login**

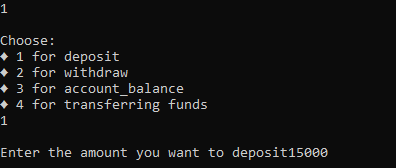
**Correct Login Credentials**



**Wrong Login Credentials**

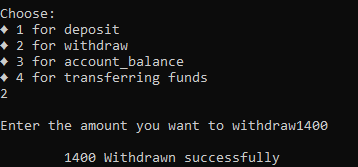


**Deposit**

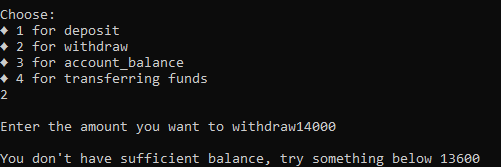


**Withdraw**

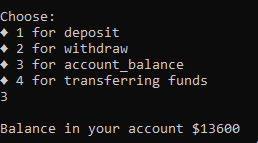
**Correct amount entered**



**Incorrect amount entered**



**Check Account Balance**



**Transfer Funds**

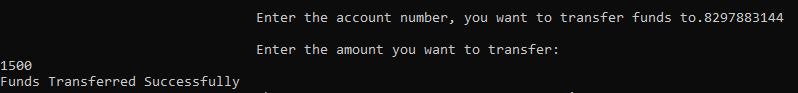
**Incorrect account number entered**



**Self account number entered**



**Correct account Number entered**



**5 ADDITIONAL KNOWLEDGE GAINED**

We understood the usage of Verhoeff’s algorithm in generation of Aadhar

Numbers and also in the API’s suggested by UIDAI in verifying the aadhar

Number.

We, understood how C can be used as a backend-programming language.

We, felt the need of databases and how they are better than using files for

Storing and retrieving data.

**6. CONCLUSION AND FUTURE WORK**

We, through this project understood the importance of Algorithms and their

Application in the various softwares around us.

We, would like to extend the scope of the project by building a web-based or a

Native windows application using react-native for UI and Firebase.

**7. References**

WEBSITE: <https://www.geeksforgeeks.org/readwrite-structure-file-c/>

USED FOR: Writing and Reading structures to a file

WEBSITE:

<https://www.quora.com/How-do-I-write-a-code-for-a-hidden-password-without-using-getch-in-C-language>

USED FOR: Hiding the password entered by the user.