Malaysian Urban Banking Management System

Banking Simplified for You

Sandesh Subedi

13th March, 2020

Abstract

Our project, the "Malaysian Urban Banking Management System," represents a revolution in modern banking. This assignment explores the dynamic world of banking technology, where computers have reshaped the landscape. This console-based banking application is designed to offer essential services to both clients and administrators. Its primary goal is to provide user-friendly, secure, and attractive banking experiences.

Admins play a crucial role, in managing clients and overseeing operations. Registration and login are prerequisites for admin access, ensuring security. The project employs C programming concepts, including structures, arrays, functions, loops, validation, file handling, and pointers. These concepts are woven into the application's fabric, creating a robust and efficient banking system. In a nutshell, the project showcases a sample output, demonstrating its real-world functionality. It represents the power of technology in modern banking, delivering a secure and user-friendly platform for all.

Acknowledgment

I would like to express my heartfelt gratitude to my professor, Mr. Sushil Adhikari, for his unwavering guidance and mentorship throughout this project. His expertise as well as encouragement has been invaluable.

I also want to extend my thanks to my classmates for their support, insightful discussions, and collaborative efforts, which have enhanced the quality of our work.

Lastly, I appreciate the wealth of knowledge and resources from the academic community and open-source platforms, which played a crucial role in our project's success.

Table of Contents

1. Introduction and Assumptions

Banking applications represent a category of enterprise software extensively adopted by banking institutions and held in high regard by their clientele. The contemporary appeal of these applications can be attributed to their characteristics of simplicity, security, visual appeal, and the added incentive of exclusive offers provided through the application platform.

Regarding the specific project at hand, it involves the development of a banking application derived from the Malaysian Urban Bank Management System. This software application primarily focuses on delivering an array of services related to client accounts within the banking sector. The software operates within a console-based environment, distinguishing between two key user roles: 'Admin,' responsible for system operation, and 'Clients,' who benefit from the provided services.

2. User Roles

There are two major user types in our application: Admin and Clients. Both of them have their own sets of privileges which they can use to facilitate the banking process. Their major roles and functions and roles are described below:

2.1. Admin

In a broader context, an administrator assumes the responsibility of overseeing, managing, and facilitating client interactions within the application framework. This role encompasses several distinct functions aimed at effectively handling and supporting clients. Administrators are empowered to access and review the client database, display transaction records, and execute account deletions when necessary. To engage in administrative activities within this software environment, registration, and authentication are integral prerequisites.

2.2. Clients

In a parallel vein, clients represent end-users who avail themselves of the application's services. Their roles encompass accessing and utilizing the features provided by the system. Clients are granted specific privileges that enable them to interact with their accounts. These include accessing exclusive account information, updating personal details, monitoring account balances, executing financial transactions such as deposits and withdrawals, and reviewing transaction histories. To participate as clients in this software ecosystem, they must undergo a registration and login process, thereby ensuring secure access to their designated functionalities.

3. Application Design

This section delves into the foundational structure of the application, offering a succinct yet informative description of its architectural design and operational mechanisms. The two major design blocks are pseudocode and flowcharts. Further information regarding them and their application in the design of the Malaysian Urban Banking Management System is demonstrated below:

3.1. Pseudocode

Pseudocode serves as an informal means of articulating programming logic, free from the constraints of rigid programming language syntax. It plays a pivotal role in outlining the algorithmic processes of the application.

3.2. Flowchatrs

Flowcharts, employed within this context, serve as visual representations of the program's logic. They utilize a series of geometric symbols and interconnected lines to graphically illustrate the algorithm's logical flow. These graphical aids are instrumental in enhancing comprehension of the program's intricate operational structure.

3.3. Application of Pseudocode and Flowcharts

Pseudocode for Main Menu

```
Step 1: BEGIN the Program
Step 2: Display Menu
Step 3: Ask to choose Input
Step 4: Initiate Switch
Step 5: Initiate Switch Case
        IF Choice == 0, Log Out
        IF Choice == 1, Go to Register Menu
        IF Choice == 2, Go to Login Section
        ELSE
        Return to Step 3
        END of Switch
        END of Program
```

Flowchart for Main Menu

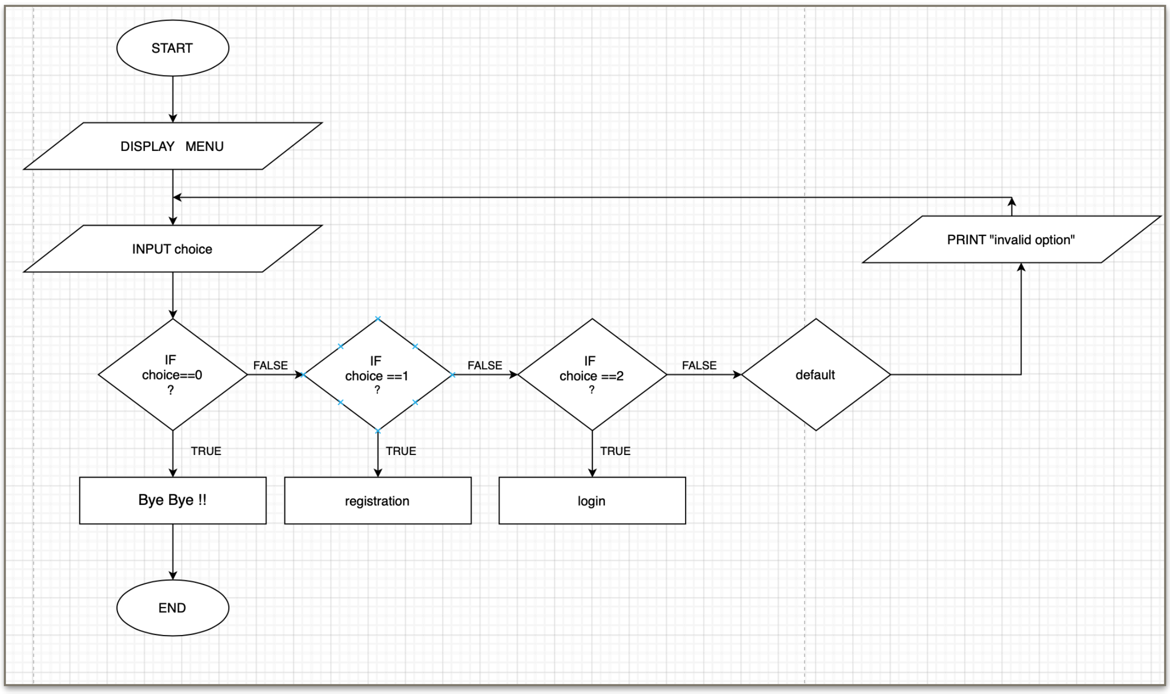


Figure 1: Flowchart for Main Menu

Pseudocode for Admin Menu

```
Step 1: BEGIN the Program
Step 2: INPUT Admin Name and Password
Step 3: IF(username= user and truepasswd=password)
            Log In to the Admin Menu
            Display Menu
            Input Control Choice
            IF (choice == 0), Log out of the Program
            IF (choice == 1), Display Clients List
            IF (choice == 2), Display Clients Transaction
            IF (choice == 3), Erase Clients Account
            ELSE
            Print 'Invalid Option' and Return to Menu
            END 'IF'
           IF (username ==!username and password!==password)
           PRINT 'Sorry, Cannot Open'
  END of Admin Menu
```

Flowchart for Admin Menu

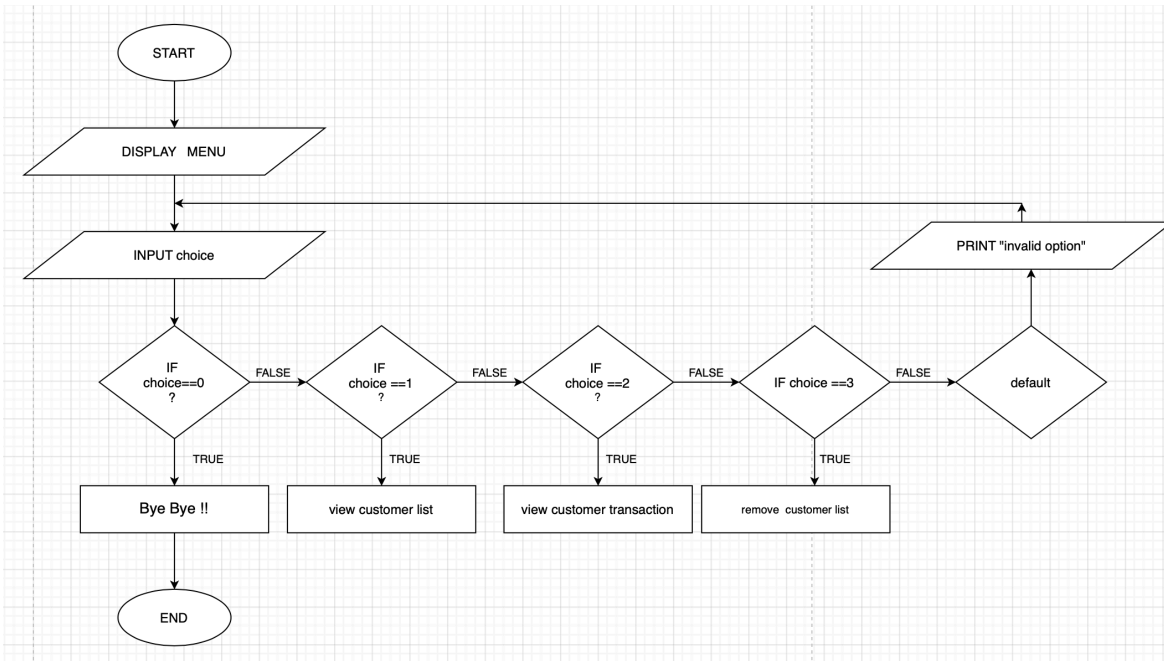


Figure 2: Flowchart for Admin Menu

Pseudocode for Client Menu

```
BEGIN
Log In to customerLogin menu
INPUT username and password
OPEN public file.txt file
IF (username==username and password==password)
ENTER inside the Client Menu
        Display Menu Inside Clients Menu
        Enter your choice
        IF (choice==0), Log Out of the program
        IF (choice==1), Display Deposit
        IF (choice==2), Display Withdraw
        IF (choice==3), Update Personal Information
        IF (choice==4), Display Personal Information
        IF (choice==5), Display Transaction
        IF (choice==6), Display Account Detail
        ELSE
        Print 'Invalid Choice'
Return to Menu
END
```

Flowchart for Client Menu

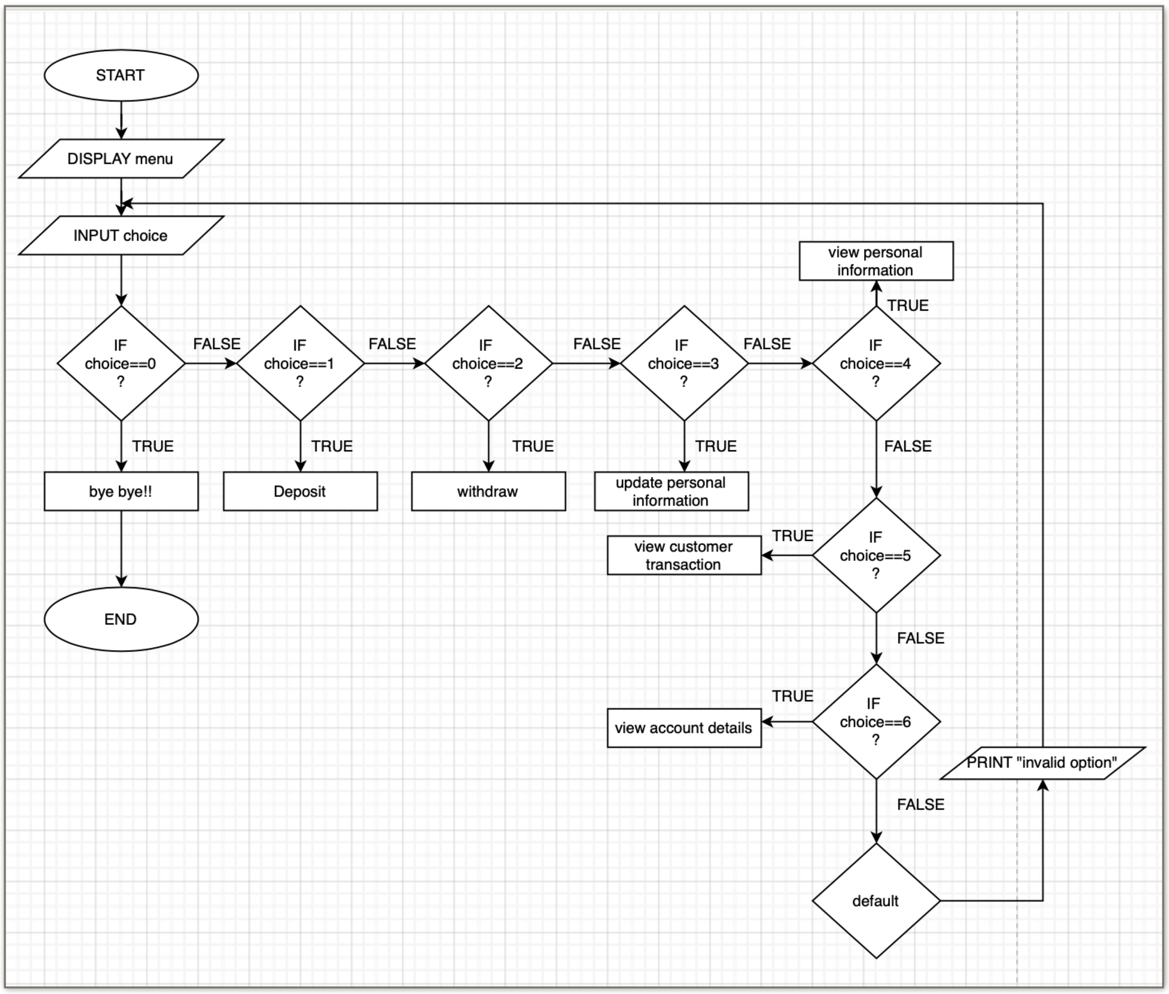


Figure 3: Flowchart for Client Menu

Pseudocode for Deposit Section

```
BEGIN
Display file transaction
Enter the amount to deposit
Start of for loop
a=list[i]
If AcNumber == tmp.AcNumber
tmp.amount= tmp.amount + deposit amount
Write in file transaction.txt
End of if
List [i]=a
End of for loop
Call function
END of deposit menu
```

Flowchart for Deposit Section

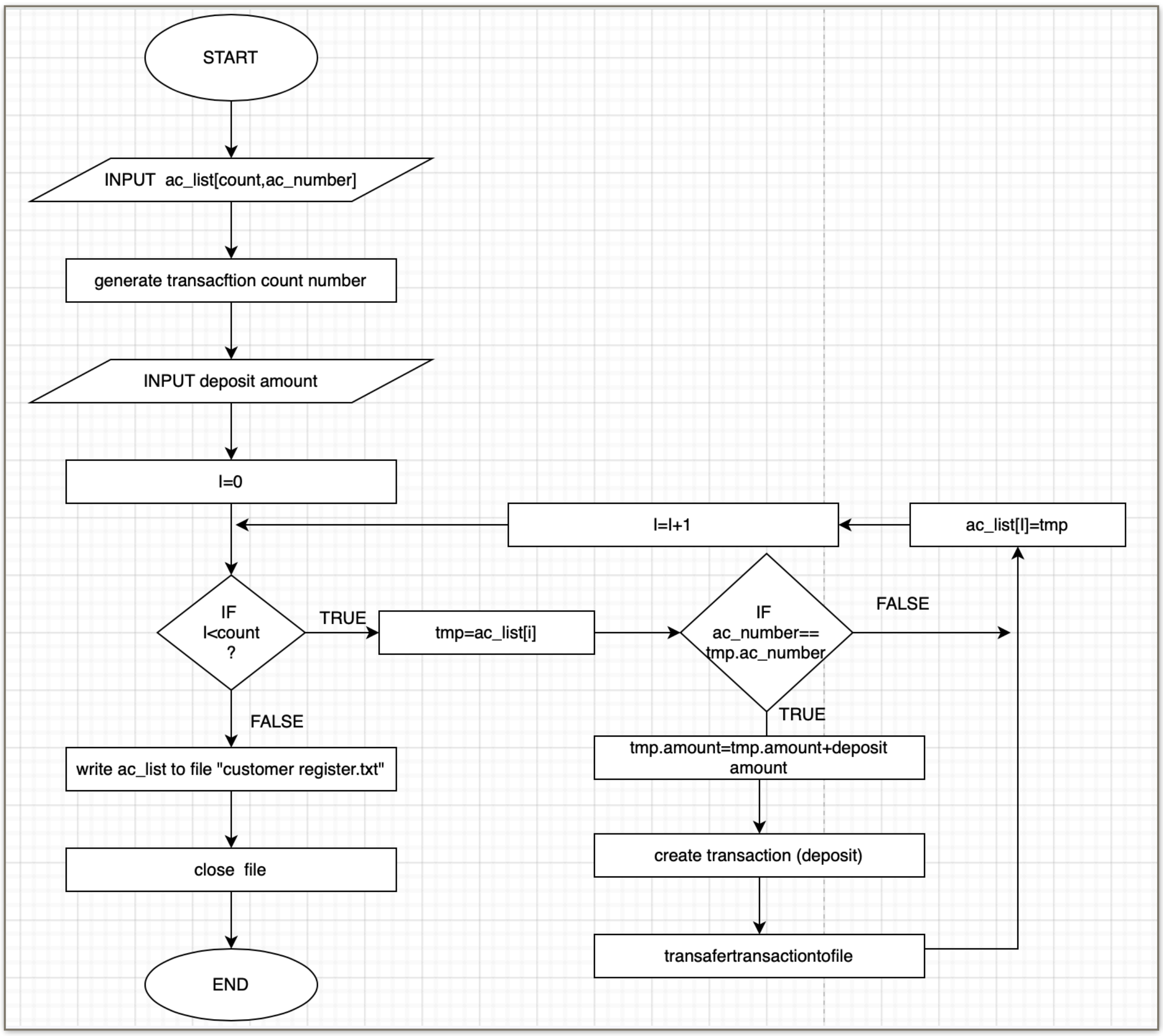
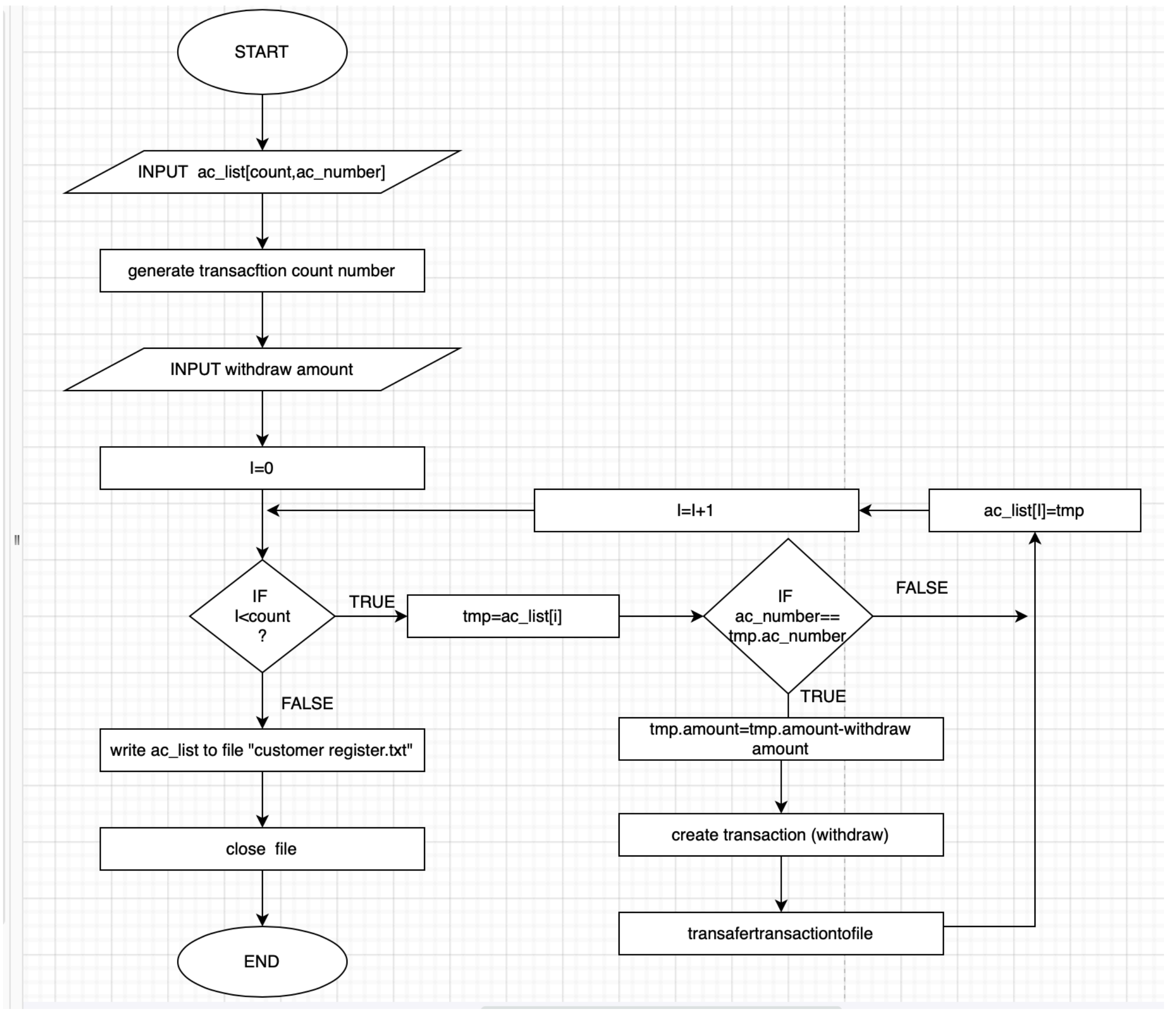

START
INPUT ac_list[count,ac_number]
generate transacftion count number
INPUT deposit amount
I=0
I=I+1
ac_list[I]=tmp
IF I<count ?
TRUE
tmp=ac_list[i]
IF ac_number== tmp.ac_number
FALSE
TRUE
FALSE
write ac_list to file "customer register.txt"
tmp.amount=tmp.amount+deposit amount
create transaction (deposit)
close file
transafertransactiontofile
END

Figure 4: Flowchart for Deposit Section

Pseudocode for Withdraw Section

```
START
Open file transaction.txt
INPUT withdrawamount //amount to withdraw
Start of for loop
a=list[i]
IF AcNumber == tmp.AcNumber
IF amount &lt; withdrawamount
Then Print ‘less amount withdraw is not possible’
END
ELSE then
tmp.amount=tmp.amount-withdrawamount
Write in file transaction.txt
END of IF
End of IF
List [i]=a
End of for loop
Function call
END of withdraw menu.
```

Flowchart for Withdraw Section



START
INPUT ac_list[count,ac_number]
generate transacftion count number
INPUT withdraw amount
I=0
I=I+1
ac_list[I]=tmp
IF I<count ?
TRUE
tmp=ac_list[i]
IF ac_number== tmp.ac_number
FALSE
TRUE
FALSE
write ac_list to file "customer register.txt"
tmp.amount=tmp.amount-withdraw amount
close file
create transaction (withdraw)
transafertransactiontofile
END

Figure 5: Flowchart for Withdraw Section

4. Application of Programming Concepts

4.1. Structure

The structure is a user-defined datatype in C language that authorizes users to combine data of distinctive types together. It assists in generating data type that seems more meaningful in a program. In my project, structure has been used multiple times like structure date, structure DOB, structure name, structure clients, and so on.

Sample Application

```
Struct customer

{

Char id [ 100 ]

    Int AcNumber;

    Int id_number;

    Int age;

    Float amount;

   Char ac_type[5]

   Longlong int phone;

   Char gender[10];

   Char nationality[100];

   Char date dob;

   Char role;

};
```

4.2. Arrays

Arrays can be defined as a kind of data structure that can store a definite-sized collection of elements of the same type. The function of the array is to store a collection of variables of the same type. Particularly in my project, the array stored the list of all accounts that are used.

Example: Struct account a_list[100]

4.3. Functions

Functions in C programming can be explained as a group of statements that provide functionality to perform a certain task together. A program in C is incomplete without the inclusion of at least one function i.e., main( ). There are two types of functions. They are :

4.3.1. Pre-defined Functions

- #include : This function shape processes and uses pre-defined functions
- <stdio.h> : This is used for input and output functions in a program
- <stdio.h> : This is used to generate a console screen

4.3.1. User-defined Functions

- **Void Customer( ) :**

This function displays menu and call function which is accessed by a client.

- **Void customerLogin ( ) :**

This function is used in logging in to the client's account to the application. This reads data from the file by receiving several information like account numbers, usernames, and passwords.

- **Void autoAcNumber ( ) :**

This function reads data from file transaction.txt and compares the account number of all transactions ultimately.

- **Void View_customers_list ( ) :**

This function reads files and assists in displaying the clients that are registered or authorized in the application. This function displays the transactions created by clients within the program. This function is called inside the admin menu, which reads data from file transaction.txt and prints all the available transactions.

- **Void Remove_customer_account ( ):**

This function erases clients from the application permanently. This function is also called the inside admin menu as only the admin has authorization to erase existing accounts.

- **Void Deposit ( ) :**

This peculiar function is for the deposition of the amount in clients' accounts. This function studies data and compares the ID of a client. This function not only accepts input of amount but also adds amount in a specific account.

- **Void Update_personal_informtion ( ) :**

The feature of this function is it aids in modifying and updating existing information into new ones. This function takes the input float amount and adds balance to a particular account. Clients can change their data for self-satisfaction and security.

- **Void View_transaction ( ) :**

This function displays the transactions done by all the clients in the banking application. This shows the deposited and withdrawn amount with addition and subtraction in an account. This is also authorized via the admin and is called inside the admin menu. It reads data from the file transaction.txt and publishes all transactions.

- **Void withdraw ( ) :**

This function compares username and ID which further deducts the withdrawn amount when executed. This only functions when the amount to be withdrawn is less than or equal to the available balance.

4.4. Iteration Statement

Iteration statements are commonly known as 'Loops'. It is the methodology where certain instructions or statements are executed repeatedly for a specified number of times until a required condition is met. In my project, a for loop is used for the purpose of iterating arrays.

Sample :

```
for(i=0; i<number; i++)
{
  tmp=a_list[i];
  if(AcNumber==tmp.AcNumber)
   {
for(i=i+1; i<number; i++)
     {
      a_list[i-1]=a_list[[i];
     }
   }
}
```

4.5. Validation

Validation is often ignored section in most of the fields. However, this is one of the indispensable factors in making applications secure, trustable, successful, and functioning appropriately. In this application, validation is used during the changing of passwords where the previous password must match to new password for that to be modified.

4.6. File Handling

More or less specific inputs are essential for data to be generated several numbers of times in programming. However, the data that is to be displayed can be huge in size, which may not be supportive. In this case, data can be stored in a local file which can be authorized anytime the user wants. So, file handling is something that enables users to create, update, and erase the files stored in that local file via the C program.
Example :

- fopen( ) - Opens new or existing files

- fprintf( ) - Writes the data in a file

For instance, in the project, file handling transaction.txt is used in the format of txt, where transactions of clients are saved and published whenever necessary.

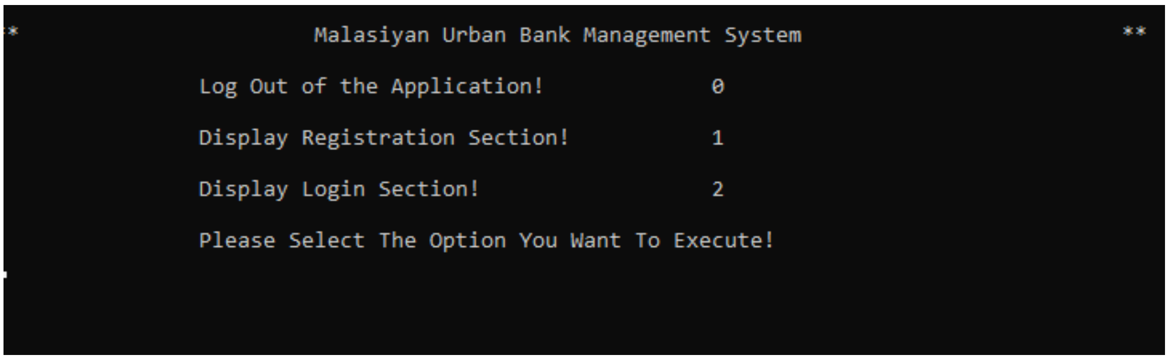
4.7. Pointers

In C programming language, the pointer can be defined as the variable that stores/ points the address of another variable. The main function of the pointer is to allocate memory dynamically at the run time. In the project, a file pointer is used, which controls as well as authorizes files from device storage.

5. Application Outcomes

Within this section, we delve into the practical application of the system, accompanied by visual aids in the form of screenshots. These screenshots serve as windows into the user interface and interactions, providing a tangible glimpse of how the system operates in real-world scenarios. By combining textual descriptions with visual representations, this section offers a comprehensive understanding of the user experience and functionality of the application.

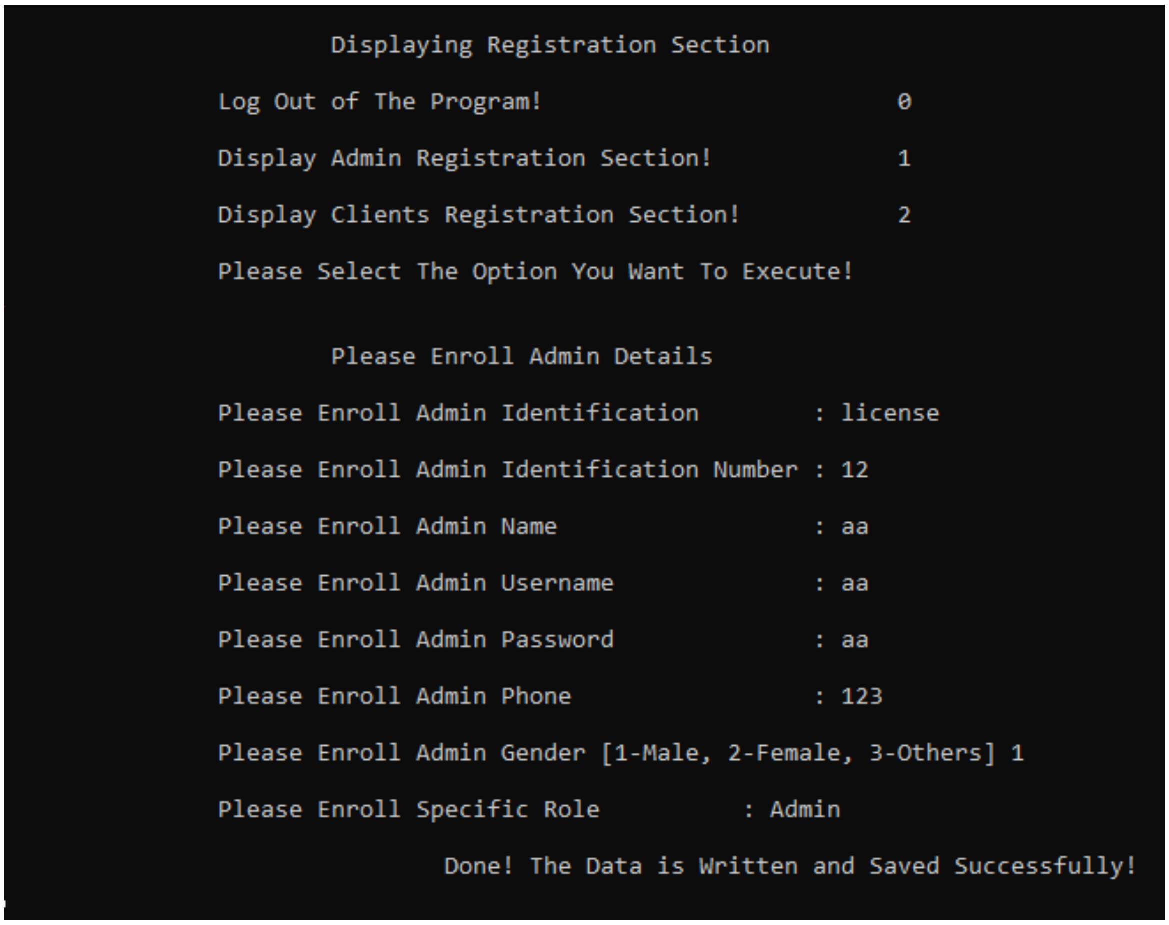
Screenshot showcasing the Main Menu Screen



```
*                  Malasiyan Urban Bank Management System                    **

            Log Out of the Application!            0

            Display Registration Section!          1

            Display Login Section!                 2

            Please Select The Option You Want To Execute!
```

Figure 6: Main Menu Screen

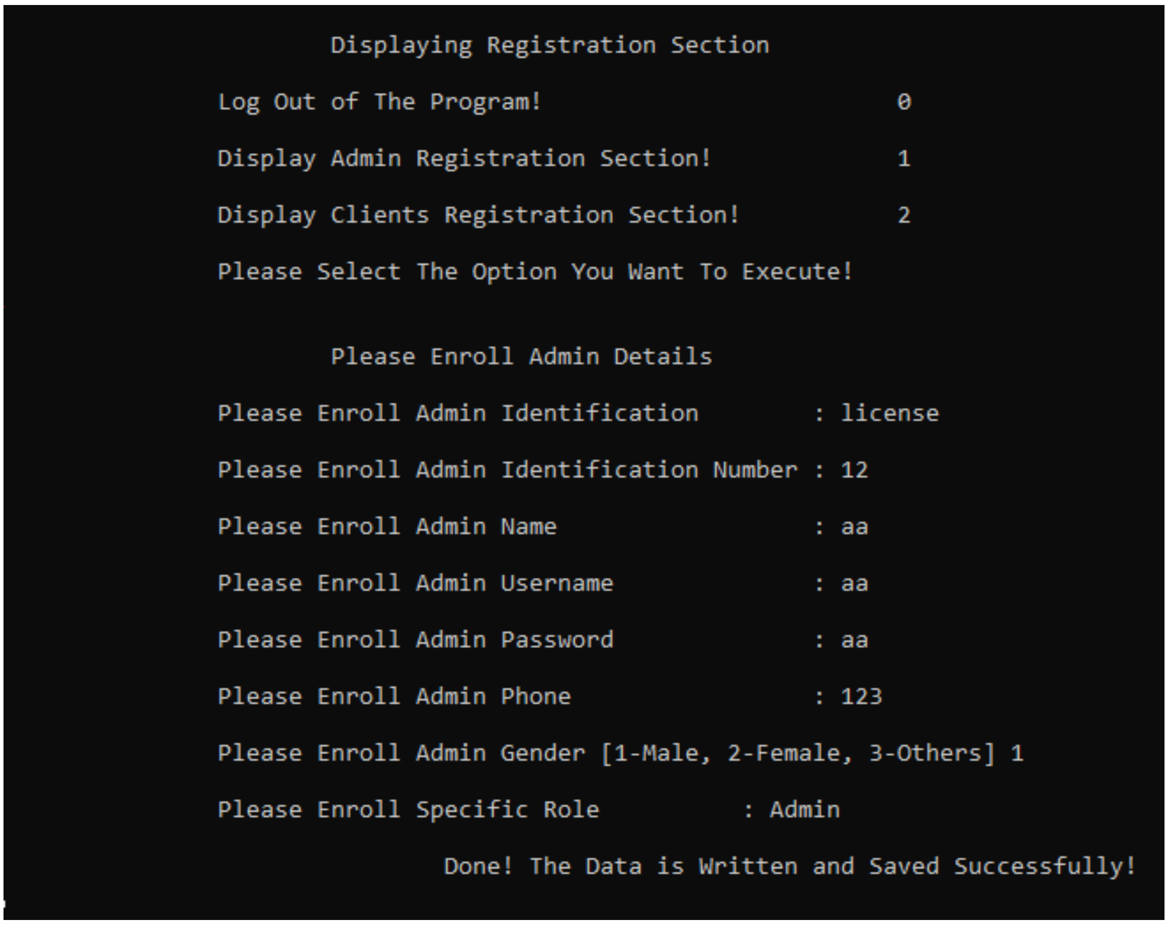
Screenshot showcasing the Admin Registration Screen



```
                Displaying Registration Section

        Log Out of The Program!                            0

        Display Admin Registration Section!                1

        Display Clients Registration Section!              2

        Please Select The Option You Want To Execute!

                Please Enroll Admin Details

        Please Enroll Admin Identification        : license

        Please Enroll Admin Identification Number : 12

        Please Enroll Admin Name                  : aa

        Please Enroll Admin Username              : aa

        Please Enroll Admin Password              : aa

        Please Enroll Admin Phone                 : 123

        Please Enroll Admin Gender [1-Male, 2-Female, 3-Others] 1

        Please Enroll Specific Role          : Admin

                        Done! The Data is Written and Saved Successfully!
```

Figure 7: Admin Registration Screen

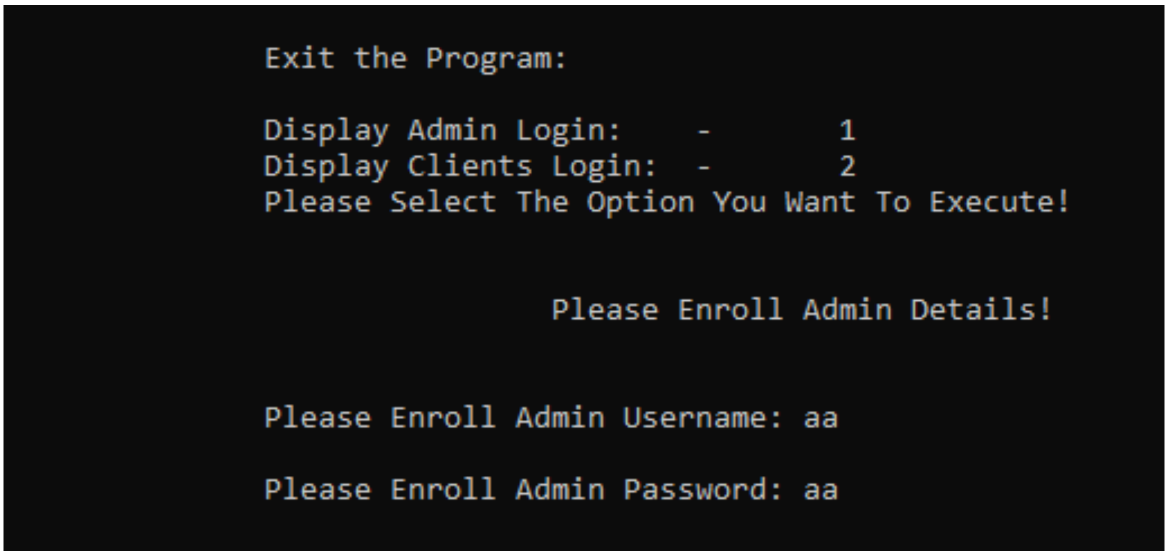
- **Screenshot showcasing the Client Registration Screen**



```
        Displaying Registration Section

Log Out of The Program!                         0

Display Admin Registration Section!             1

Display Clients Registration Section!           2

Please Select The Option You Want To Execute!


        Please Enroll Admin Details

Please Enroll Admin Identification        : license

Please Enroll Admin Identification Number : 12

Please Enroll Admin Name                  : aa

Please Enroll Admin Username              : aa

Please Enroll Admin Password              : aa

Please Enroll Admin Phone                 : 123

Please Enroll Admin Gender [1-Male, 2-Female, 3-Others] 1

Please Enroll Specific Role          : Admin

                Done! The Data is Written and Saved Successfully!
```

Figure 8: Client Registration Screen

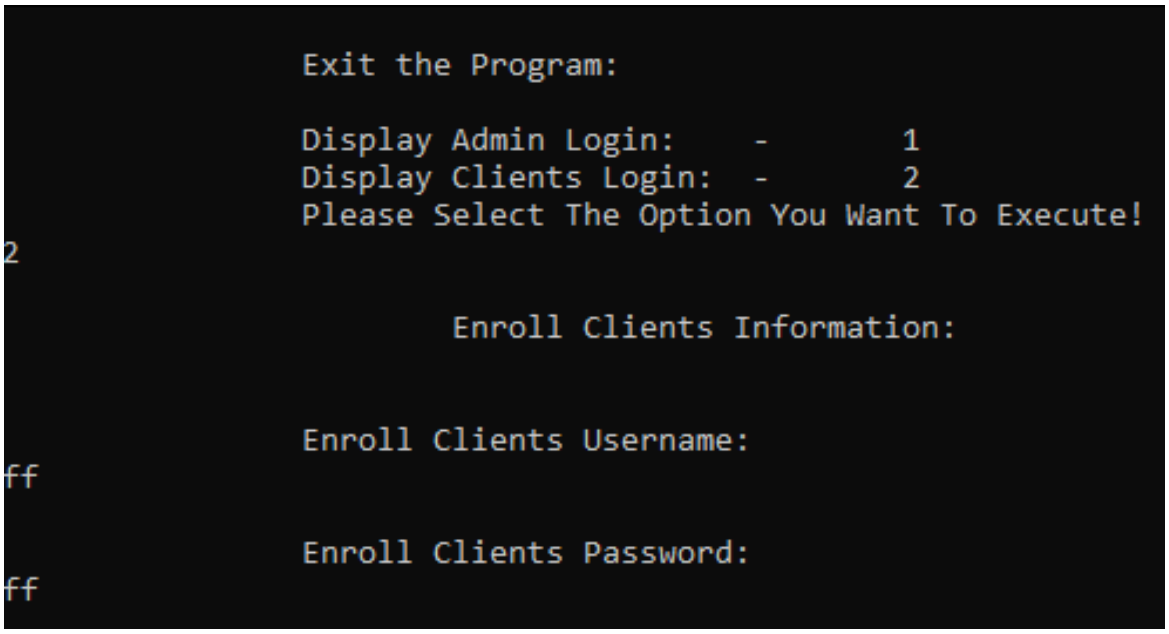
- **Screenshot showcasing the Admin Login Screen**



```
Exit the Program:

Display Admin Login:    -       1
Display Clients Login:  -       2
Please Select The Option You Want To Execute!


            Please Enroll Admin Details!


Please Enroll Admin Username: aa

Please Enroll Admin Password: aa
```

Figure 9: Admin Login Screen

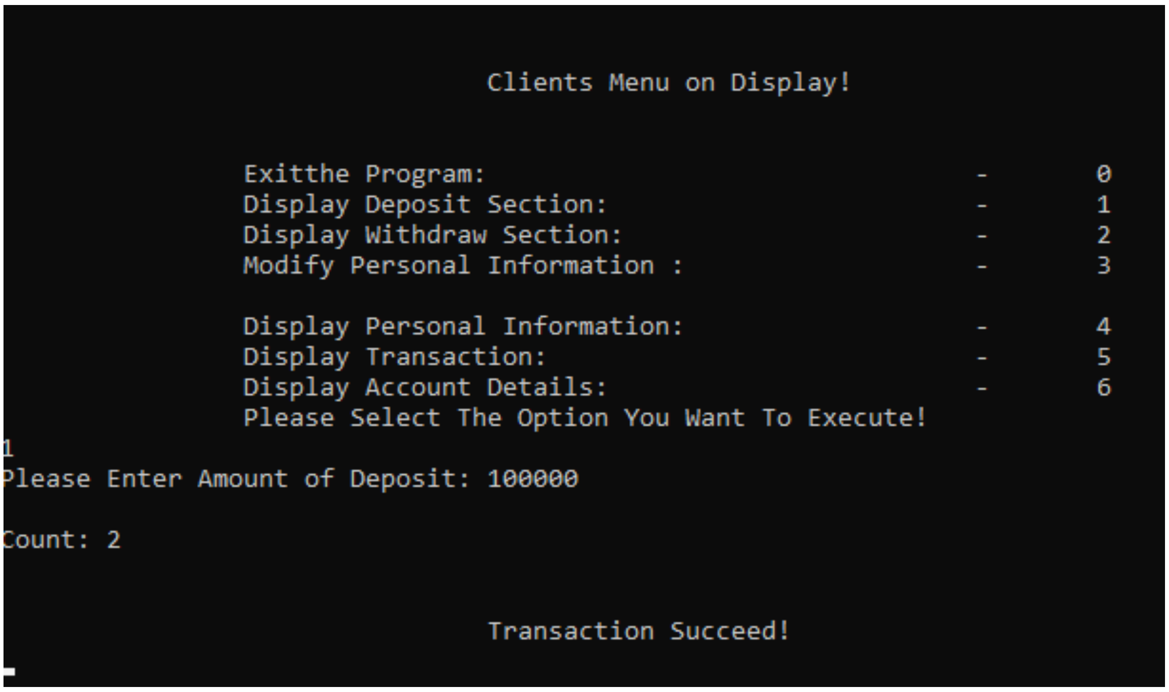
- **Screenshot showcasing the Client's Login Screen**



```
                   Exit the Program:

                   Display Admin Login:    -       1
                   Display Clients Login:  -       2
                   Please Select The Option You Want To Execute!
2

                            Enroll Clients Information:


                   Enroll Clients Username:
ff


                   Enroll Clients Password:
ff
```

Figure 10: Client's Login Screen

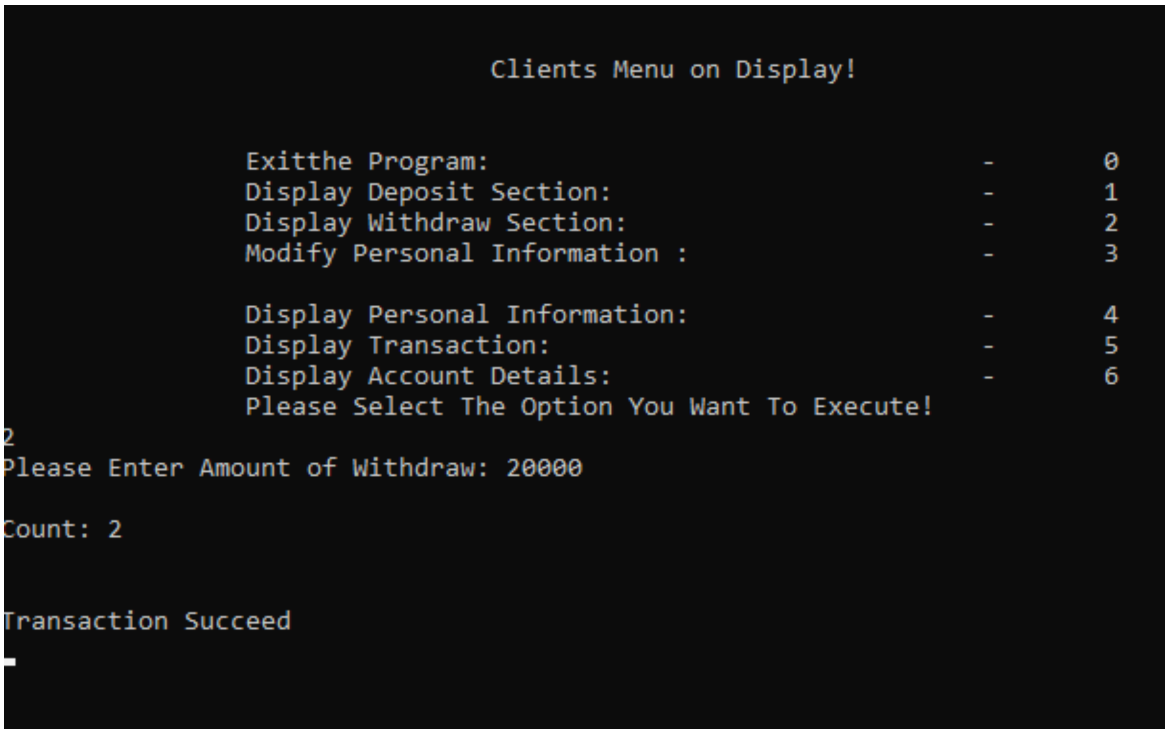
- **Screenshot showcasing the Deposit Section Screen**



```
                                Clients Menu on Display!


               Exitthe Program:                                -       0
               Display Deposit Section:                        -       1
               Display Withdraw Section:                       -       2
               Modify Personal Information :                   -       3

               Display Personal Information:                   -       4
               Display Transaction:                            -       5
               Display Account Details:                        -       6
               Please Select The Option You Want To Execute!
1
Please Enter Amount of Deposit: 100000

Count: 2


                                Transaction Succeed!
```

Figure 11: Deposit Section Screen

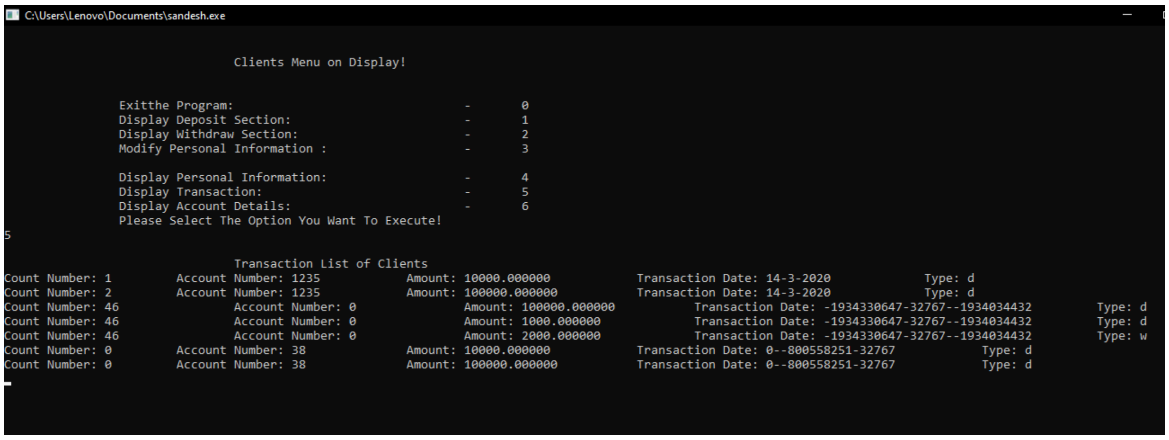
- **Screenshot showcasing the Withdraw Section Screen**



```
                         Clients Menu on Display!


            Exitthe Program:                                  -      0
            Display Deposit Section:                          -      1
            Display Withdraw Section:                         -      2
            Modify Personal Information :                     -      3

            Display Personal Information:                     -      4
            Display Transaction:                              -      5
            Display Account Details:                          -      6
            Please Select The Option You Want To Execute!
2
Please Enter Amount of Withdraw: 20000

Count: 2


Transaction Succeed
```

Figure 12: Withdraw Section Screen

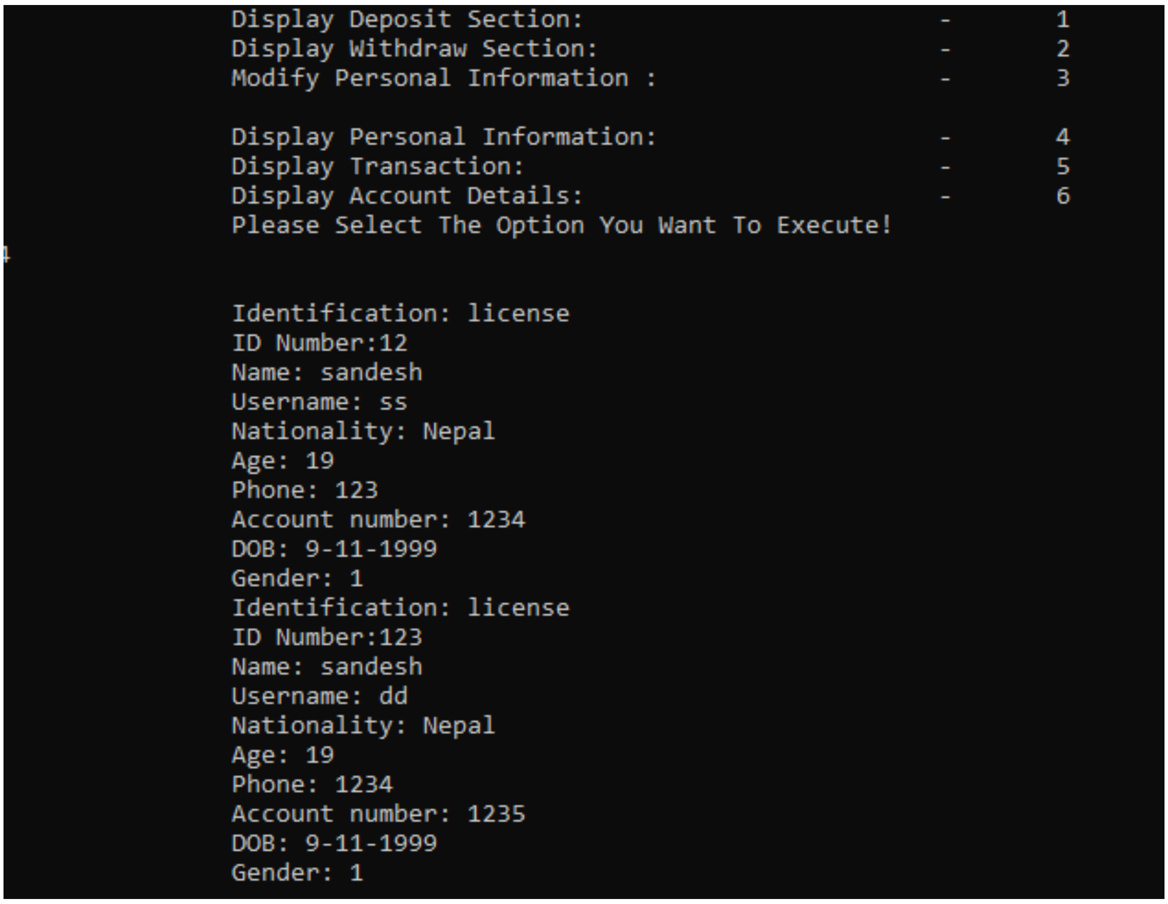
- **Screenshot showcasing the Transaction Display Screen**



```
C:\Users\Lenovo\Documents\sandesh.exe

                                Clients Menu on Display!


                Exitthe Program:                                -       0
                Display Deposit Section:                        -       1
                Display Withdraw Section:                       -       2
                Modify Personal Information :                   -       3

                Display Personal Information:                   -       4
                Display Transaction:                            -       5
                Display Account Details:                        -       6
                Please Select The Option You Want To Execute!
5

                                Transaction List of Clients
Count Number: 1         Account Number: 1235            Amount: 10000.000000            Transaction Date: 14-3-2020             Type: d
Count Number: 2         Account Number: 1235            Amount: 100000.000000           Transaction Date: 14-3-2020             Type: d
Count Number: 46                Account Number: 0               Amount: 100000.000000           Transaction Date: -1934330647-32767--1934034432         Type: d
Count Number: 46                Account Number: 0               Amount: 1000.000000             Transaction Date: -1934330647-32767--1934034432         Type: d
Count Number: 46                Account Number: 0               Amount: 2000.000000             Transaction Date: -1934330647-32767--1934034432         Type: w
Count Number: 0         Account Number: 38              Amount: 10000.000000            Transaction Date: 0--800558251-32767            Type: d
Count Number: 0         Account Number: 38              Amount: 100000.000000           Transaction Date: 0--800558251-32767            Type: d
```

Figure 13: Transaction Displaying Screen

- **Screenshot showcasing the Personal Details Screen**



```
               Display Deposit Section:                          -      1
               Display Withdraw Section:                         -      2
               Modify Personal Information :                     -      3

               Display Personal Information:                     -      4
               Display Transaction:                              -      5
               Display Account Details:                          -      6
               Please Select The Option You Want To Execute!
4


               Identification: license
               ID Number:12
               Name: sandesh
               Username: ss
               Nationality: Nepal
               Age: 19
               Phone: 123
               Account number: 1234
               DOB: 9-11-1999
               Gender: 1
               Identification: license
               ID Number:123
               Name: sandesh
               Username: dd
               Nationality: Nepal
               Age: 19
               Phone: 1234
               Account number: 1235
               DOB: 9-11-1999
               Gender: 1
```

Figure 14: Personal Details Screen

6. Conclusion

The banking application project offered a valuable learning experience, introducing us to new features and programming techniques. This application provides essential functionalities for clients and administrators, including exclusive account management, convenient deposits, and withdrawals. This project exemplifies the potential of technology to enhance banking services, setting the stage for future advancements. As the digital realm of banking continues to evolve, the Malaysian Urban Banking Management System project represents a noteworthy stride towards harnessing the power of technology to elevate financial services, setting the stage for future advancements and improved user experiences.

7. References

GeeksforGeeks. (2023). C Programming Language tutorial. *GeeksforGeeks*. https://www.geeksforgeeks.org/c-programming-language/

Goues, C. L., Holtschulte, N. J., Smith, E. K. M., Brun, Y., Dévanbu, P., Forrest, S., & Weimer, W. (2015). The ManyBugs and IntroClass benchmarks for automated repair of C programs. *IEEE Transactions on Software Engineering*, *41*(12), 1236–1256. https://doi.org/10.1109/tse.2015.2454513

Jones, R. W. M., & Kelly, P. H. J. (1997). Backwards-Compatible Bounds Checking for Arrays and Pointers in C Programs. *Arrays in C Programming*, *001*, 13–26. https://www.ep.liu.se/ea/cis/1997/009/02/cis9700902.pdf

Kim, J., & Moon, J. Y. (1998). Designing towards emotional usability in customer interfaces—trustworthiness of cyber-banking system interfaces. Interacting With Computers, 10(1), 1–29. https://doi.org/10.1016/s0953-5438(97)00037-4

Stroustrup, B. (1985). The C++ programming language. http://ci.nii.ac.jp/ncid/BA26230123