

BANKING MANAGEMENT SYSTEM(BMS)

- **Team Members:**

S.NO.	NAME	ROLL NUMBER
1	Anuj Singh	03
2	Purushottam Raj	26
3	Rajan Neupane	27
4	Raman Rajbansi	29

- Department of Computer
- Thapathali campus, Tribhuvan University
- Date: 16th March,2025

Login System in C

This presentation covers the implementation of a secure login system in C, designed to simulate a basic authentication process using account data stored in text files.



Initial Setup and User Input

Clear Screen & Beep

`Clearscreen()` the terminal

`Beep(1000, 750);` forc clears audio feedback

User Prompt

Requests account number & password

Buffer cleared of extra characters

File System Search

1 Open Directory: Uses `opendir(".")` to access current directory

2 Iterate Files: Checks for `.txt` extension

3 Binary Mode: Opens file in `rb` mode

Data Retrieval

1

Read Structure

Reads `BankAccount` struct

2

Compare Credentials

Matches account/password with input

Successful Login



Match Found
Filename stored in
logged_in_user_file.



Close Directory
Directory closed



Return Value
Returns 1
indicating success

Failed Login

1

No Match
No matching account found

2

Directory Closed

System Overview

Input

User provides credentials.

Search

File system is searched for matching accounts.

Verify

Account details are verified for access.

Access

System grants or denies access.

Account Management System in C

This presentation outlines the core components of a basic account management system implemented in C. It covers account number generation, user input validation, and secure data storage.

pS

Account Creation Process



Unique Account Number Generation

Random Generation

The `generateAccountNumber()` function creates an 8-digit random number.

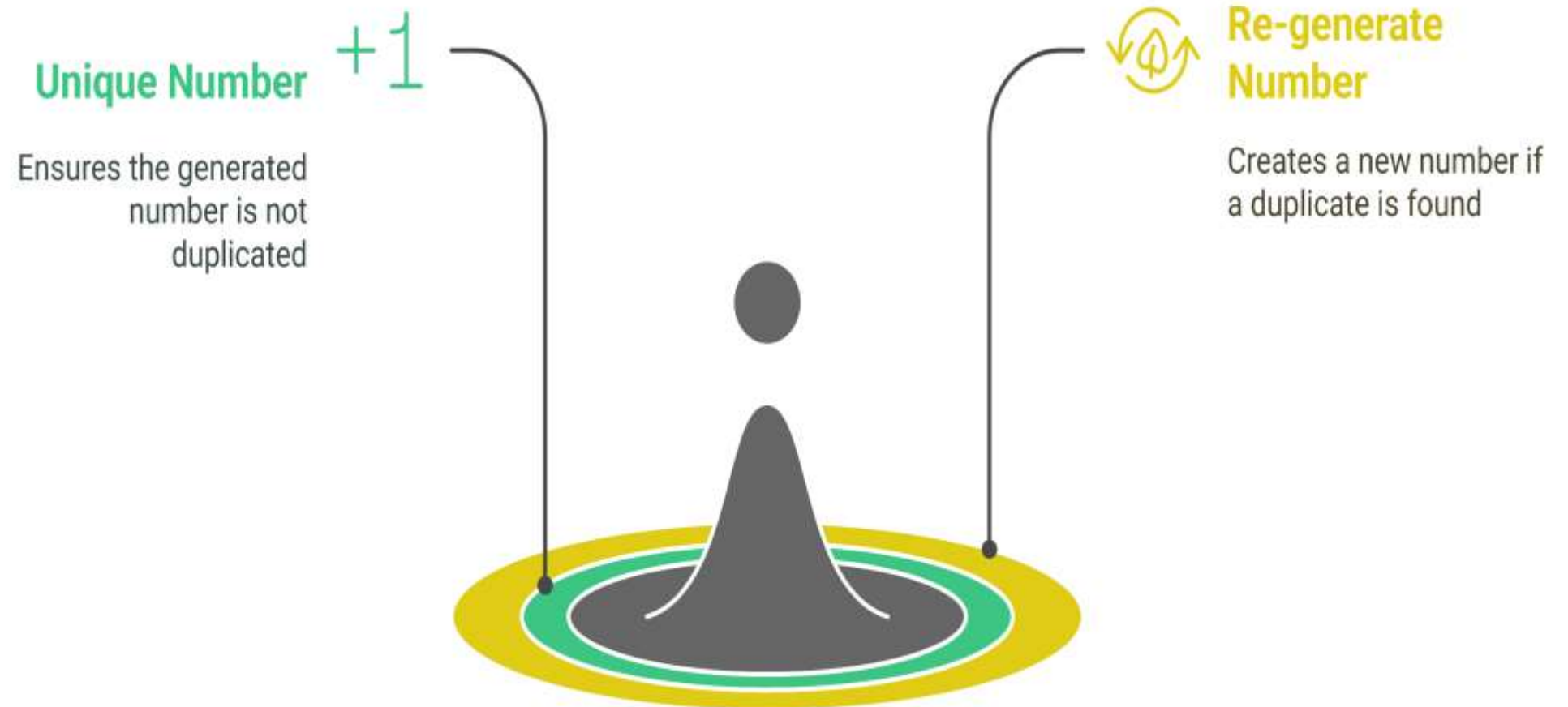
Uniqueness Check

`accountNumberExists()` scans `.txt` files to ensure the number is unique.

Re-generation

If the account number already exists, a new one is made.

Account Number Generation Process



User Input for Account Creation



Username

The user enters a desired username for the account.



Password

The user creates a password with a minimum length of 8.



Confirmation

Password re-entry ensures accuracy and prevents errors.

Password Validation

1 Minimum Length

The password must be at least 8 characters long for security.

2 Confirmation

The system verifies that both entered passwords match exactly.

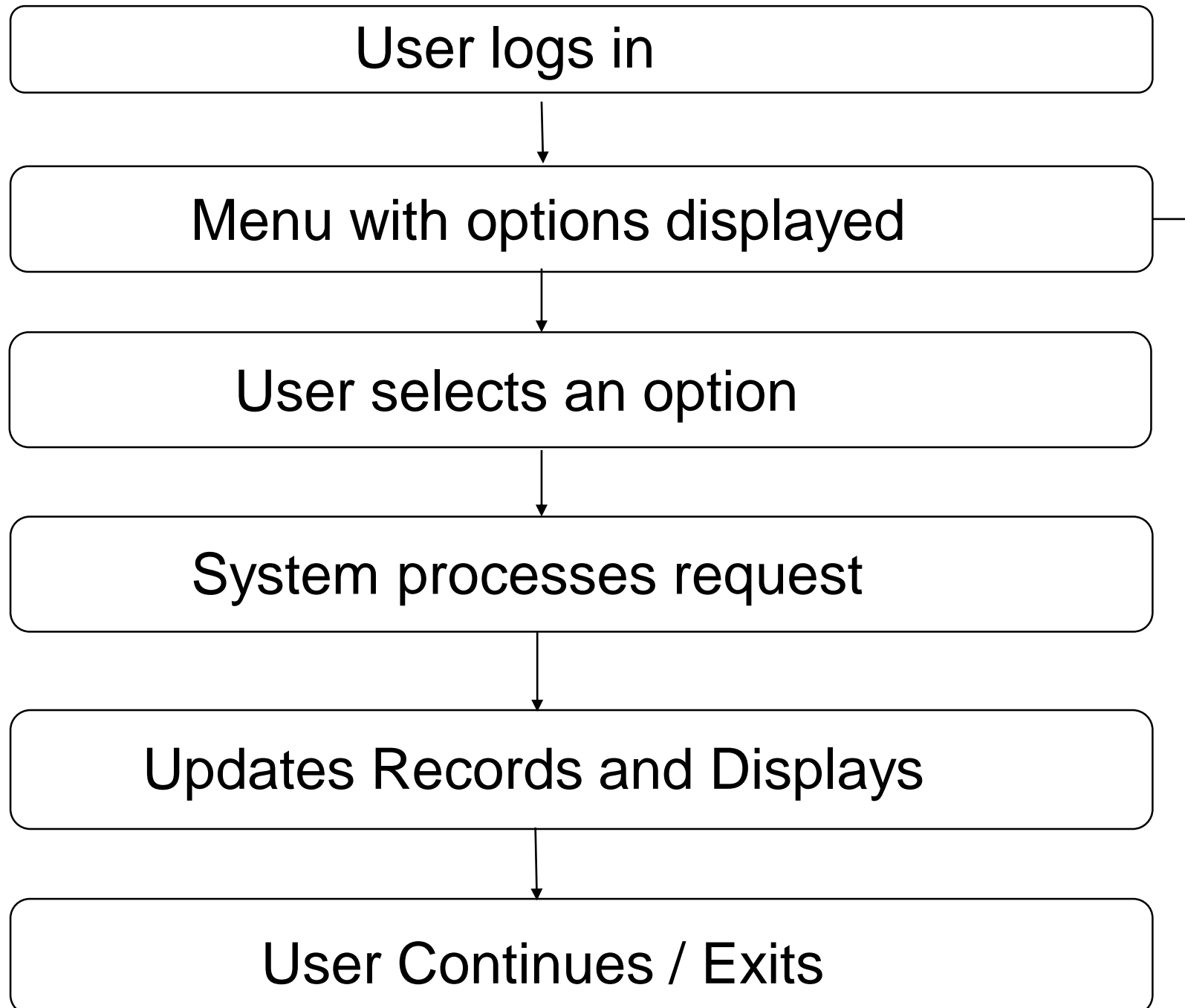
3 Re-entry Prompt

Users are prompted to fix mismatches or length issues. (Passwords do not match. Please try again)

Introduction

- Overview of the User module in Money Laundry Banking System
- Handles all user interactions related to account management
- Key functionalities: View, Deposit, Withdraw, Transfer, Delete Account

User Interaction Flow



```
Welcome YOHAN LIBERTO

Press 1 to view your account details      :
Press 2 to deposit amount to your account :
Press 3 to withdraw amount from your account :
Press 4 to transfer sum to another account :
Press 5 to delete account                  :
Press 6 to exit                            :

Enter your choice                          : █
```

Code Structure

- **neo_user()**: Main function handling user interactions.
- **account_details()**: Displays account details.
- **adding_balance()**: Allows users to deposit money.
- **withdraw_balance()**: Withdraws funds from the account.
- **transfer_balance()**: Transfers funds to another account.
- **delete_account()**: Deletes an account after confirmation.

```
void account_details(BankAccount *);  
void adding_balance(BankAccount *);  
void withdraw_balance(BankAccount *);  
void transfer_balance(BankAccount *, BankAccount *);  
int delete_account(BankAccount *);  
void neo_user();
```


Overview of neo_user() Function:

- Provides main menu for user interaction.
- Reads user data from logged_in_user_file.
- Presents options: details, deposit, withdraw, transfer, delete, exit.
- Uses a while loop for continuous menu display.
- Input validation for valid choice selection.

Sample:

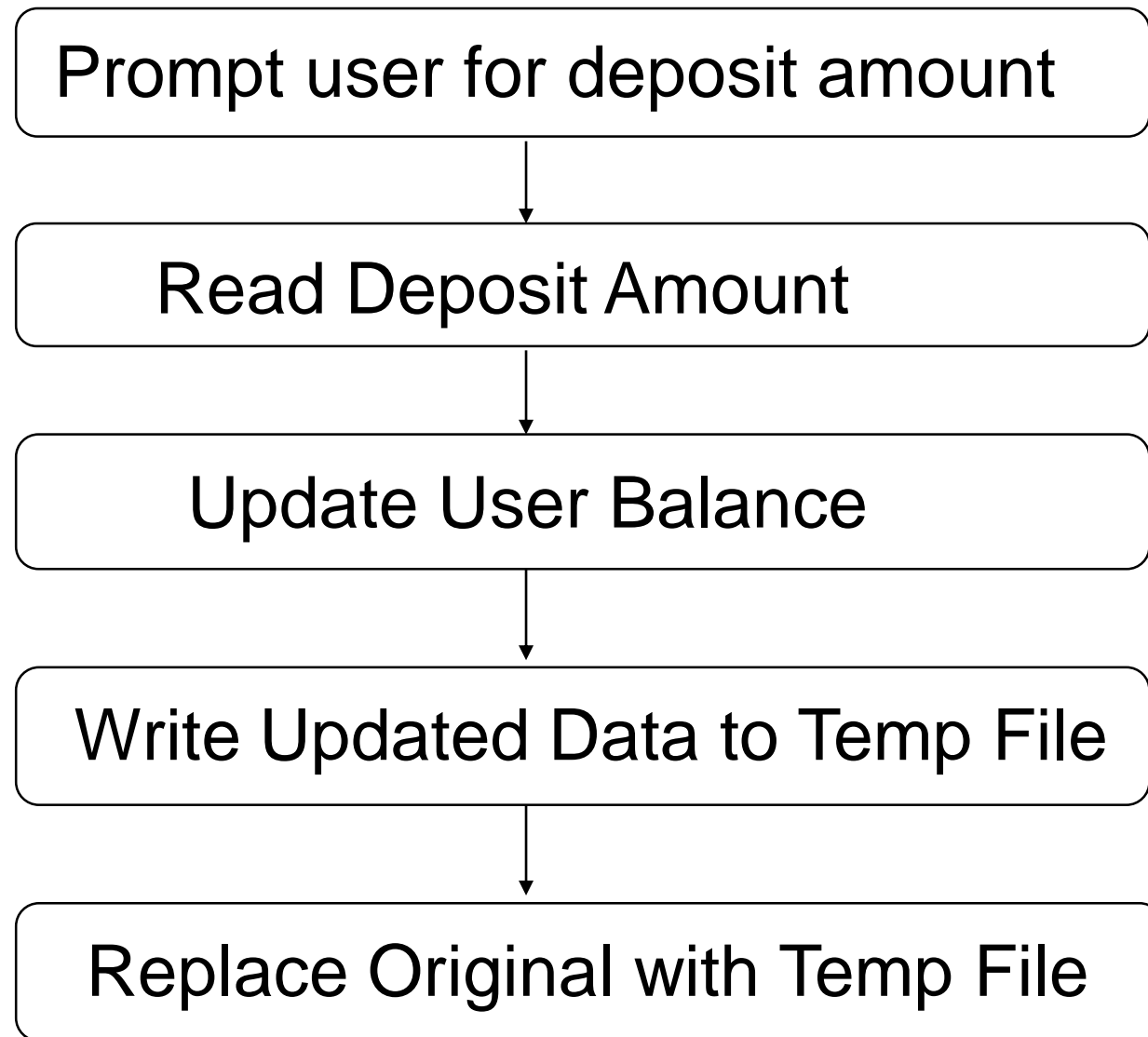
Displaying Account Information

- The `account_details()` function displays user info.
- Includes Account Number, Username, Password, Balance.
- Uses `printf` to format and display details.
- `getch()` pauses until a key is pressed.

Example Output:

```
Your account details are:  
  
Account Number : 10014168  
Username       : YOHAN LIBERTO  
Password       : yohan123  
Balance        : 0.00
```

Depositing Funds (adding_balance())

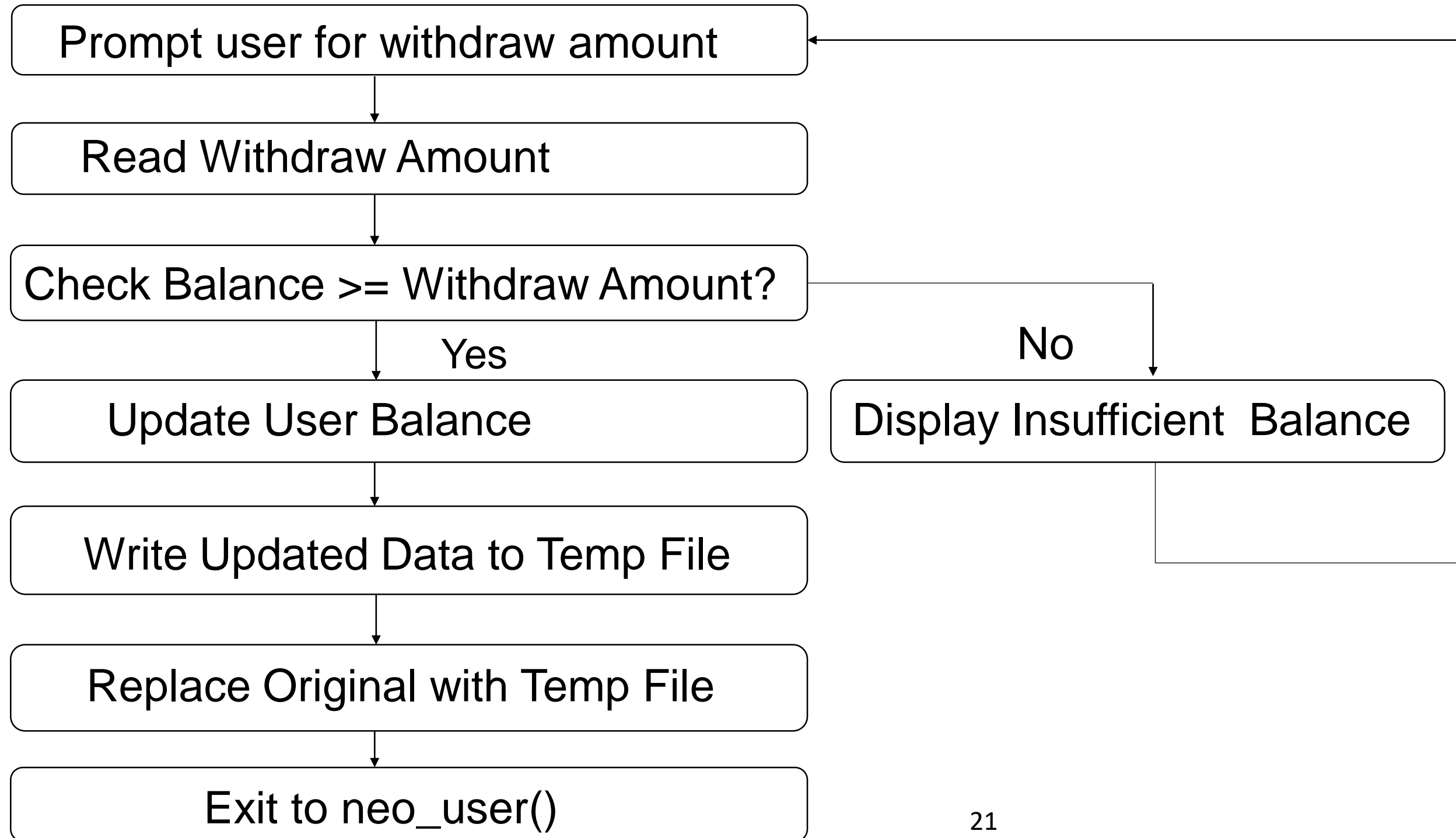


```
Enter the amount you want to deposit    : 1000

Balance is updated successfully

Your new balance is                      : 1000.00
█
```

Withdrawing Funds (withdraw_balance())



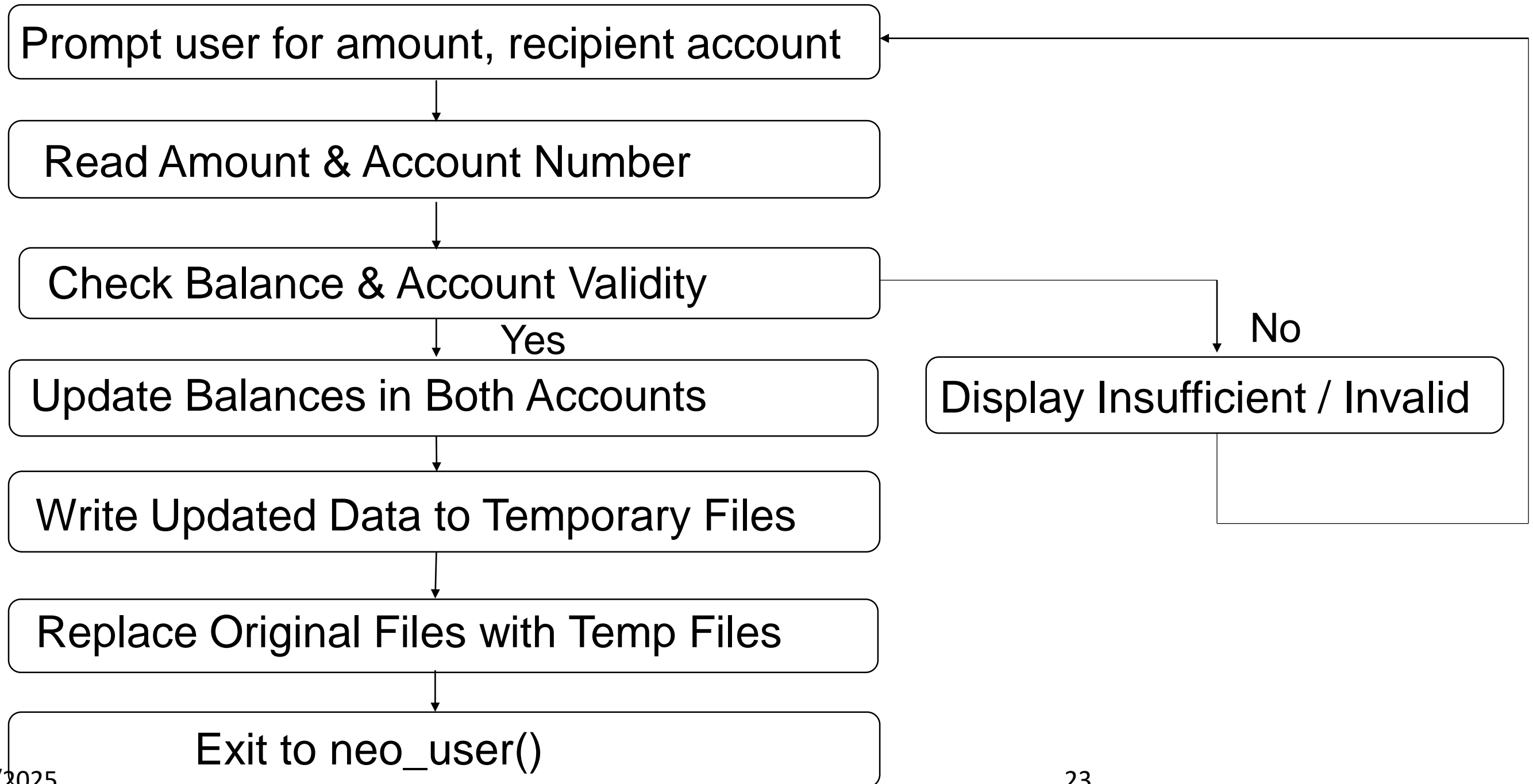
Enter the amount you want to withdraw : 1000

Amount withdrawn successfully from account.

Your new balance is : 4000.00



Transfer Balance to Another Account

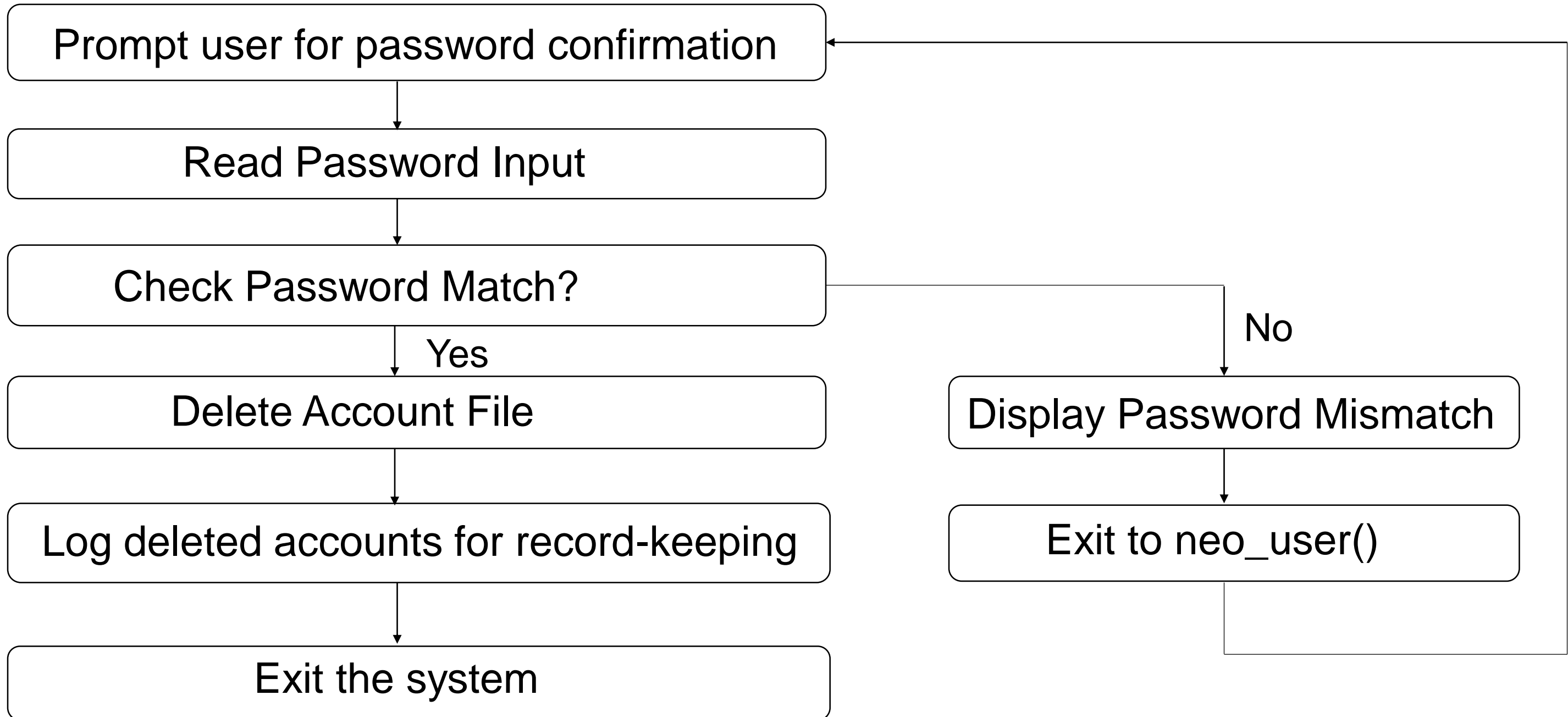


```
Enter the amount you want to transfer           :    1000
Enter the account number you want to transfer the sum to : 10029948

Amount transferred successfully.

Your new balance is                               :  3000.00
█
```


Account Deletion (delete_account())



Enter your password to confirm account deletion : yohan123

Password confirmed.

Deleting account...

Account deleted successfully.



Error Handling

- Input validation for menu selection and numeric inputs
- Ensures sufficient balance before withdrawals and transfers
- Confirms account deletion with password
- Uses file handling to store and retrieve data safely

Security Considerations

- Password verification before deletion
- Ensures correct account number for transfers

Conclusion

- Simple and efficient banking management system
- Secure and user-friendly operations
- Future improvements: UI enhancements, database integration

Questions & Discussion

- Open for queries
- Feedback and suggestions