PAGE COVER  
  
  
  
RBE 104  
  
NAMA   
KELAS

NIM

**Program Design**

The **Community Bank Simulator** is a console-based C application that simulates simple banking operations through a text-based menu.  
It allows users to create and manage accounts, perform deposits, withdrawals, transfers, and preview monthly interest.

**A. Data Model**

The program uses a structure to represent each customer account:

|  |  |
| --- | --- |
| **Variabel** | **Type** |
| accountNo | int |
| pin | int |
| balance | Double |
| name | Char |
| failed\_logins | Int |
| withdrawals\_today | Int |
| Int | locked |

All accounts are stored in fixed-array:

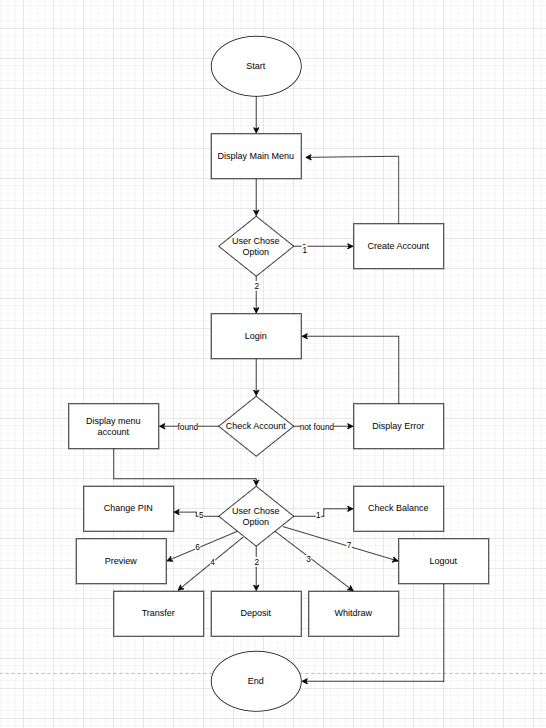
#define MAX\_ACCOUNTS 50

**B. Functional Modules**

The program follows modular design principles:

|  |  |
| --- | --- |
| **Function** | **Description** |
| create\_account() | Registers a new account |
| login\_prompt() | Handles login with account number and PIN |
| deposit\_flow() | Adds money to balance |
| withdraw\_flow() | Deducts money with daily limit rules. |
| transfer\_flow() | Transfers funds between accounts |
| simulate\_new\_day() | Resets withdrawal limits |
| show\_accounts\_summary() | Displays summary (for testing). |

**C. Flowchart**

The our programs Flow in diagram:

**Test Result (Sample Output)**

The following section presents the test results from running the Community Bank Simulator program. These results show how each main function performs in different scenarios, including account creation, transactions, and login validation.

|  |  |
| --- | --- |
| Description | Sample Output |
| Main Menu |  |
| Create Account |  |
| Login |  |
| Check Balance |  |
| Deposit |  |
| Withdraw |  |
| Change PIN |  |

## ****Analysis Comments****

The Community Bank Simulator demonstrates **modular programming** and **structured design** in C.  
Each module handles a specific functionality, making the code easier to maintain and extend.

**Strengths:**

* Proper input validation (avoids invalid or crash-causing inputs).
* Modular structure using .c and .h files.
* Realistic banking logic (withdrawal limits, account locking).

**Limitations:**

* Data is stored only in memory — lost after program exit.
* No file storage (can be added for persistence).

**Enhancements Possible:**

* File-based account saving (e.g., using fwrite / fread).
* GUI version or web-based interface in the future.

## ****Conclusion****

The Community Bank Simulator successfully meets the coursework objectives:

* Implements account management, deposits, withdrawals, transfers, and daily simulation.
* Demonstrates good coding practices in modular C programming.
* Follows clear user interface design and validation rules.

This project enhances understanding of **C structures, functions, arrays, and logic flow**, which are essential foundations for more advanced programming concepts.