SYNOPSIS

Problem Statement: Simple ATM Interface: The Simple ATM Interface allows users to perform basic banking operations like checking balances, depositing, and withdrawing money. The project performs transactions or to make a balance inquiry one should require to enter a Personal Identification Number (PIN) using a C programming language.

Project Id: P-44

Name- Divyanshu Dobhal

University Roll number. – 2418477

Section- B1

- Problem Description

The main objective of this project is to develop a Simple ATM Interface using C. It enables the user to perform basic banking functions such as viewing their account balance, making deposits and making withdrawals. However, before performing any of these, the user has to enter the correct PIN to gain access to the system. The aim is to mimic how a real ATM works on a basic level, incorporating concepts from the C language.

ATM are everywhere around us, and they've become a major part of our daily life when it comes to banking. This project may look simple, but it actually represents the base of how real ATMs operate.

Here's how it connects to real life:

(i) Security is a big deal. The PIN system we added is just like the one used in actual ATMs to keep users' money safe.
(ii) It teaches how banks handle things like cash deposits, withdrawals, and balance inquiries—but in a digital way.
(iii) Also, it gives a clear idea of how the backend of an ATM might function—even if it's just the basics.
(iv) This kind of logic is also used in real banking apps, so it's like a mini version of that.

It lays the groundwork for more intricate systems. It helped me grasp how menu-driven programs operate, how to handle user inputs, and how to construct a system that responds according to user actions.

- **STEPS OF IMPLEMENTATION**

User Authentication:

The system verifies the correct PIN entry of the user before enabling ATM functionality. The system provides three attempts before it blocks user access similar to a standard ATM.

Record Handling:

The record handling system of my ATM project requires accurate balance tracking through variable or file storage after each transaction.

String Manipulation:

The system uses strings to handle and compare user inputs such as the PIN. The comparison of PIN entries was achieved through the use of strcmp() functions.

File Handling :

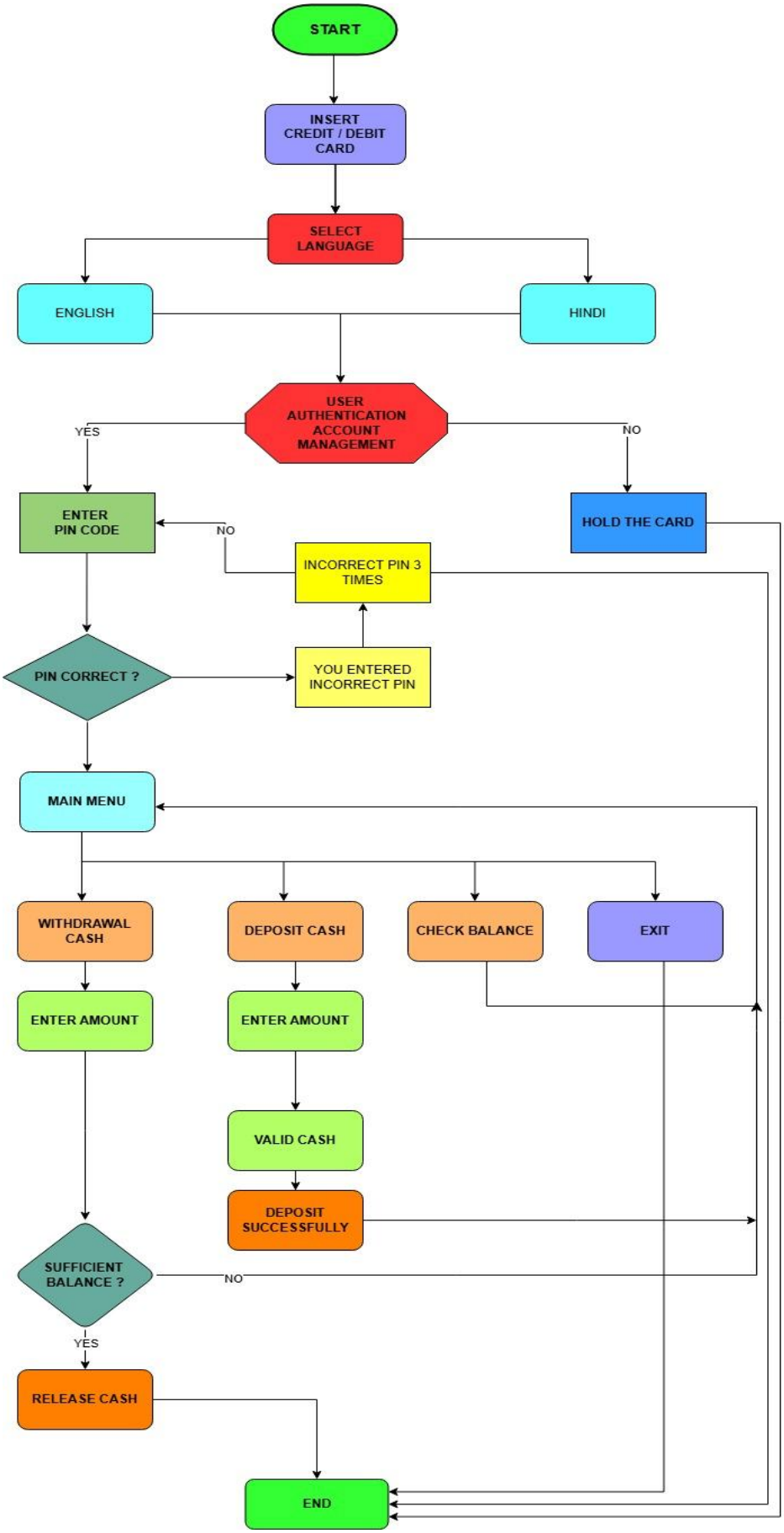
The system uses file handling to store and retrieve information about user balances. The system maintains data consistency between program sessions because it saves information when it closes and retrieves it when it restarts.

Main Driver Module:

This is the heart of the program that connects all other parts like login, deposit, and withdrawal.
It controls the flow using menus and keeps the program running until the user exits.

Report and Presentation:

This part includes all the project details like flowcharts, code explanation, and output screenshots.
It helps in clearly presenting how the ATM system works, both on paper and while explaining.


START
INSERT CREDIT / DEBIT CARD
SELECT LANGUAGE
ENGLISH
HINDI
USER AUTHENTICATION ACCOUNT MANAGEMENT
YES
NO
ENTER PIN CODE
NO
HOLD THE CARD
INCORRECT PIN 3 TIMES
PIN CORRECT ?
YOU ENTERED INCORRECT PIN
MAIN MENU
WITHDRAWAL CASH
DEPOSIT CASH
CHECK BALANCE
EXIT
ENTER AMOUNT
ENTER AMOUNT
VALID CASH
DEPOSIT SUCCESSFULLY
SUFFICIENT BALANCE ?
NO
YES
RELEASE CASH
END

- **Proposed Modules:**

Authentication- It asks the user to enter a PIN (just like a real ATM). If the PIN is correct, the user gets access to the system. If not, the program gives only 3 chances before locking the user out.

Balance Inquiry Module- This module lets the user check how much money they have available in their account. It's similar to checking your balance at an ATM — you just enter a command, and the shell will show you the current amount in your account.

Deposit Module- This feature lets the user deposit money into their account. After entering the amount, it gets added to the balance, and the updated balance is shown on the screen.

Withdrawal Module- In this module, the user can withdraw money. It first checks if there's enough balance available. If yes, the amount is deducted; if not, an error message is shown. This helps avoid overdrawing.

Main Driver Module- This is the brain of the program. It controls the entire flow—starting from PIN entry, to showing the menu, and then calling the correct module based on what the user chooses (Check Balance / Deposit / Withdraw / Exit).

Report & Output Module- After every action, this part of the code shows a message on the screen. Whether the transaction was successful, failed, or if there's an error—it keeps the user informed and improves the overall experience.

- Required topics from the subjects

1. Strings- I used strings mainly to handle things like PIN input or showing messages to the user. They make it easier to work with words and text inside the program.

2. Pointers- Pointers helped me pass values between functions and manage memory better. They're useful when I want to access or update data directly.

3. File Handling - This is used to save and load data like balance or transaction history even after the program ends.

4. Loops (while, for, do-while)- Loops were super useful for repeating actions, like showing the menu until the user exits. I also used them to give the user 3 chances to enter the correct PIN.

5. Conditional Statements (if-else)- These helped me make decisions in the program—like whether to allow a withdrawal or show an error. I used them in almost every module to control the program flow.

6. Switch Case- Switch case made it easy to handle the user's choices from the main menu. It looks cleaner and works better than using too many if-else statements.

7. Functions- Functions helped me break the whole program into smaller parts. So each task—like deposit, withdraw, or checking balance—has its own section, which keeps things neat and clear.

- Platform required :
  Turbo C, Code::Blocks with GCC compiler, Visual Studio Code with MinGW or GCC

- Books and Link Sources:
  C IN DEPHT (S.K. Srivastava, Deepali Srivastava)
  Expert C programming by Peter Van Der Linden
  Learn-C.org
  Programiz
  Code Chef