

**Submitted By:**

Ahmad Razzaq (FC-094)

Abdul Rehman (FC-089)

**Submitted To:**

Dr. Naeem Raza

**Project:**

Bank Management System

**Semester:**

BSCS-5th

**Subject:**

Assembly Language

**Banking Management System**

**Introduction:**

The Banking Management System is a simple console-based application developed in x86 assembly language using Irvine32 library. This system allows users to perform basic banking transactions such as creating a new account, logging in as an existing customer, depositing money, withdrawing money, and checking the account balance.

**Features:**

1. **User Authentication**

* New users can create an account by providing a unique username and password.
* Existing customers can log in by entering their username and password.

2. **Account Transactions**

* Users can deposit money into their accounts.
* Users can withdraw money from their accounts (if the balance is sufficient).
* Users can check their account balance.
* Users can view their account details and exit the system.

3. **File Handling**

* User File
* Money File
* Password File

**Code Structure:**

The code is organized into sections, each serving a specific purpose:

**.data Section:**

* Defines constant strings for messages, error messages, file names, and user inputs.

**main PROC:**

* Entry point of the program.
* Displays a welcome message and a menu with options for new user, existing customer, and exit.
* Handles user input and performs corresponding actions.

**File Operations:**

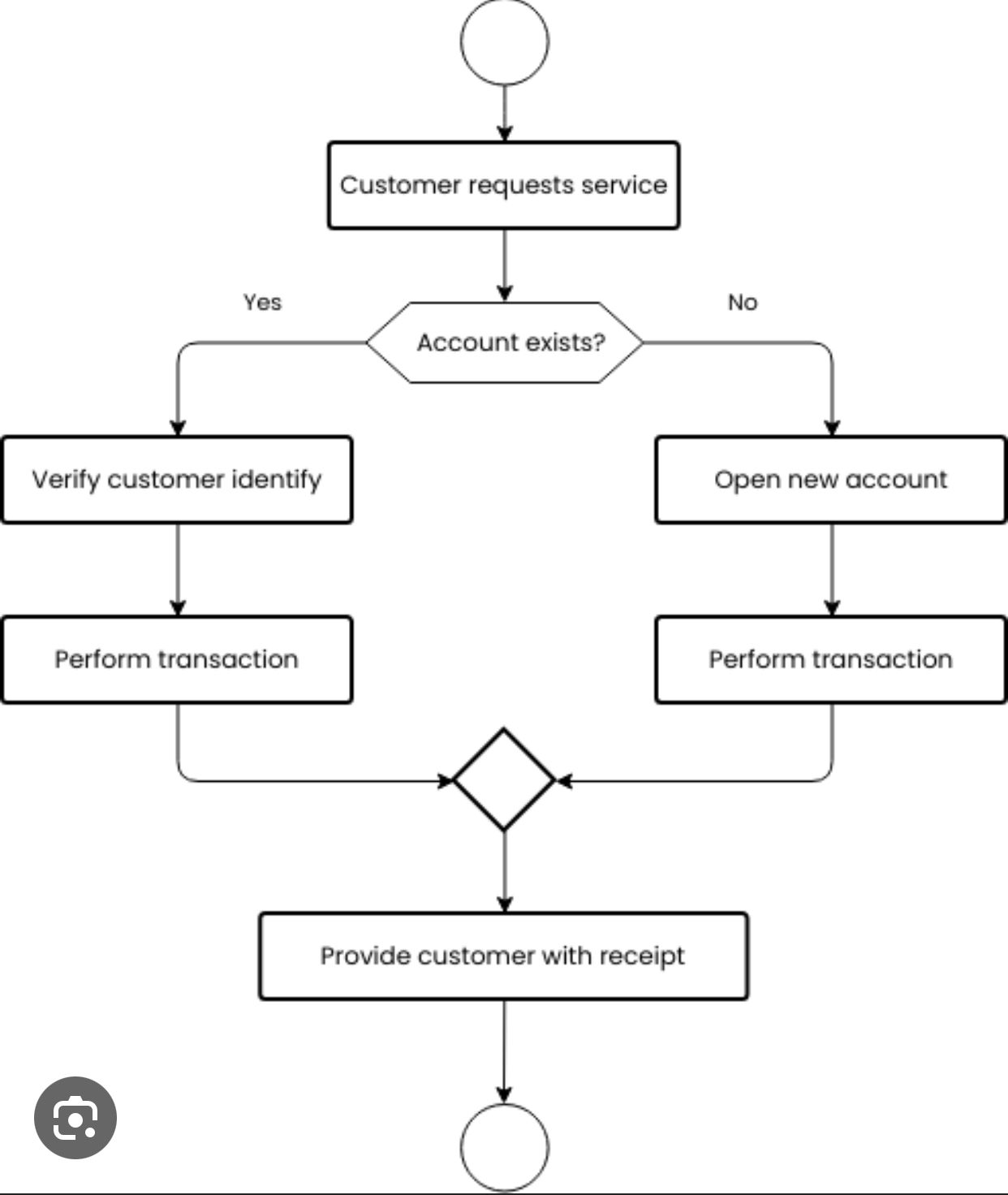
* Opens a file for input.
* Creates a new file for output.
* Writes data to the currently open file.
* Reads data from the currently open file.
* Closes the currently open file.

**Conclusion**

The Banking Management System provides a basic yet functional interface for managing user accounts and performing banking transactions. Users can easily navigate through the system to create accounts, log in, and perform various transactions while maintaining the security of their credentials.

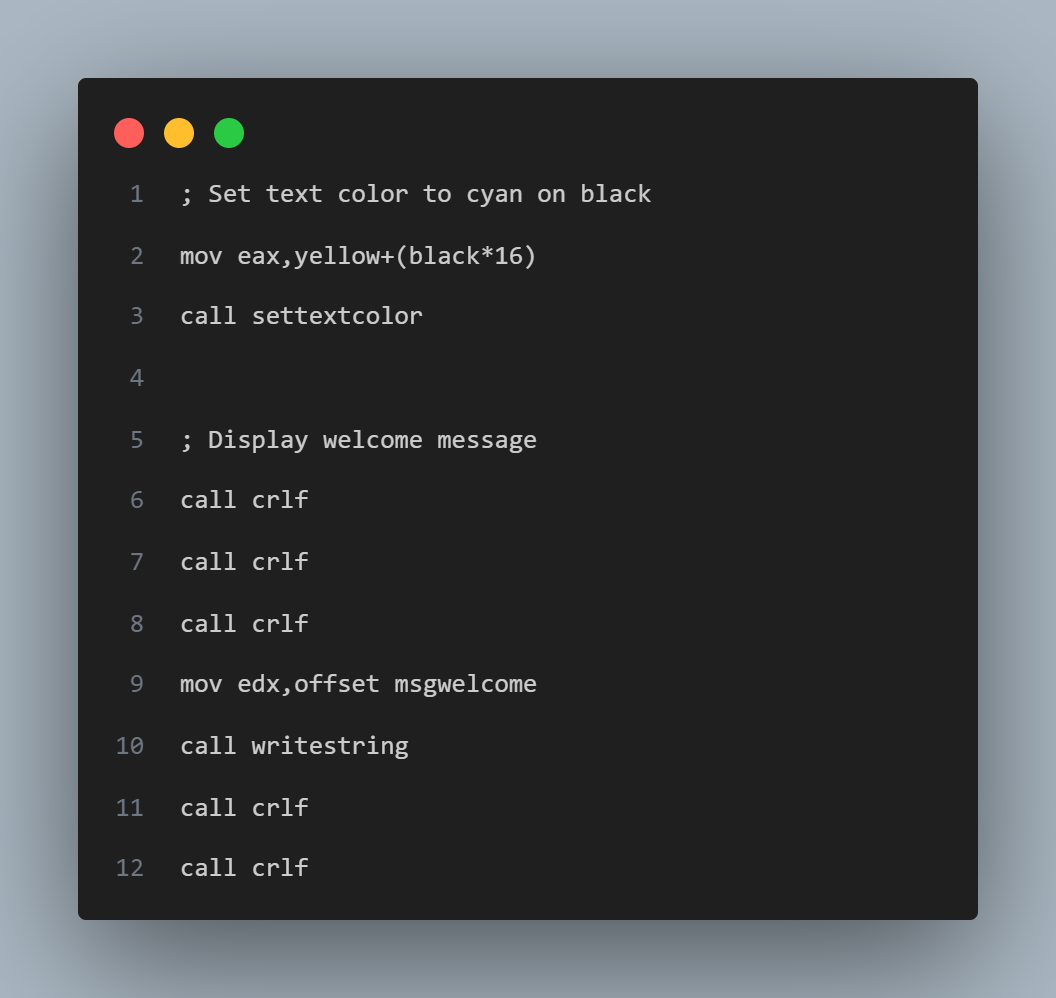
**Flow Chart**

Start

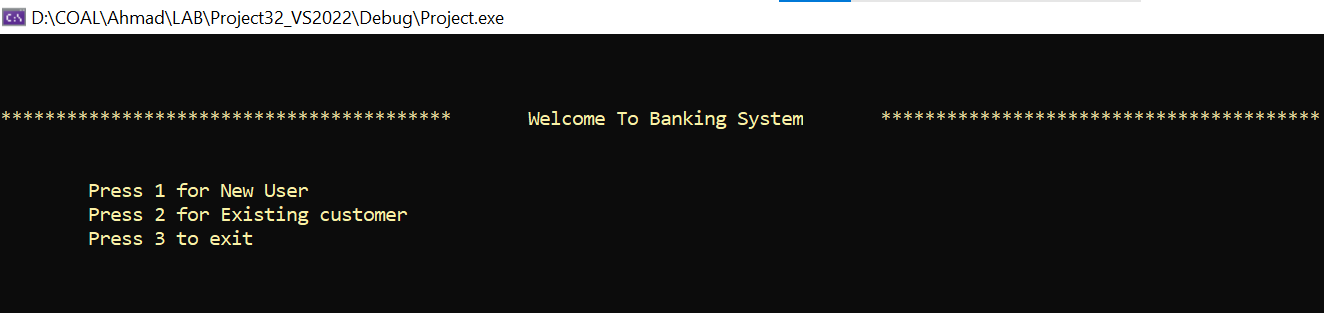


**End**

**Code**

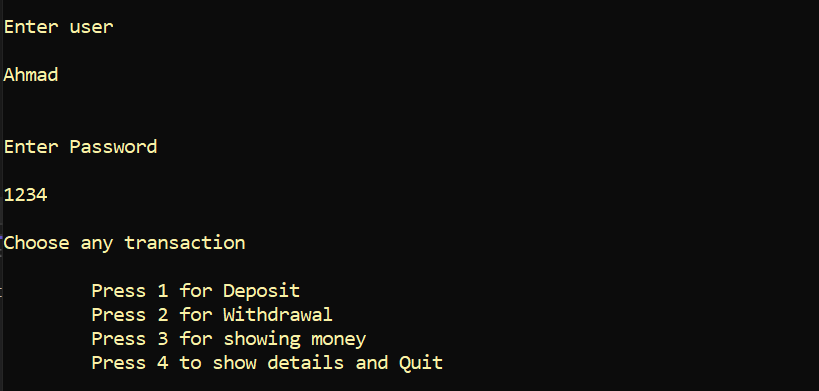


**Output**

****

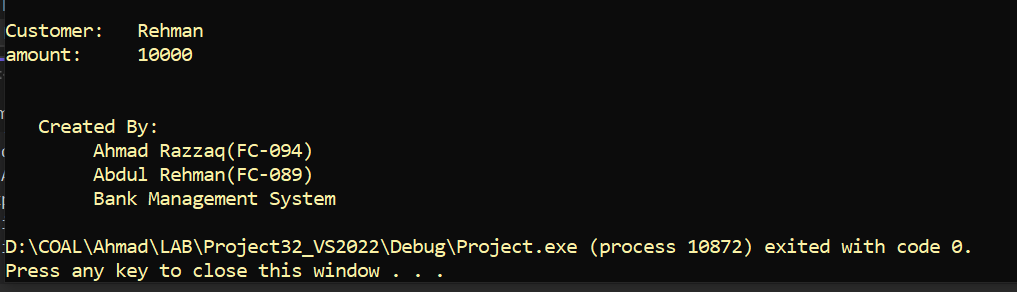
****

**Output**

****

****

**Output**

****