



ASSALAMUALAIKUM

Course : CSE - 1122

Team Members

Kazi Md Kamrul Islam Shawn

C241257

Atiqur Rahman

C241262

Raquim Rahman

C241254

Submitted To -

Mohammad Shahin Uddin

Lecturer, Adjunct Faculty

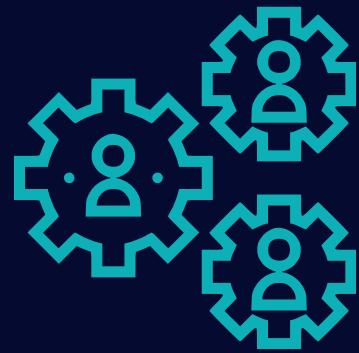
Dept. of CSE, IIUC

A dark, moody photograph of a person's hands typing on a laptop keyboard. A spiral-bound notebook is visible in the background. The image is overlaid with a dark gradient and a teal accent bar in the top right corner.

SHAHRIAH BASED

Bank Account

MANAGEMENT SYSTEM



Introduction

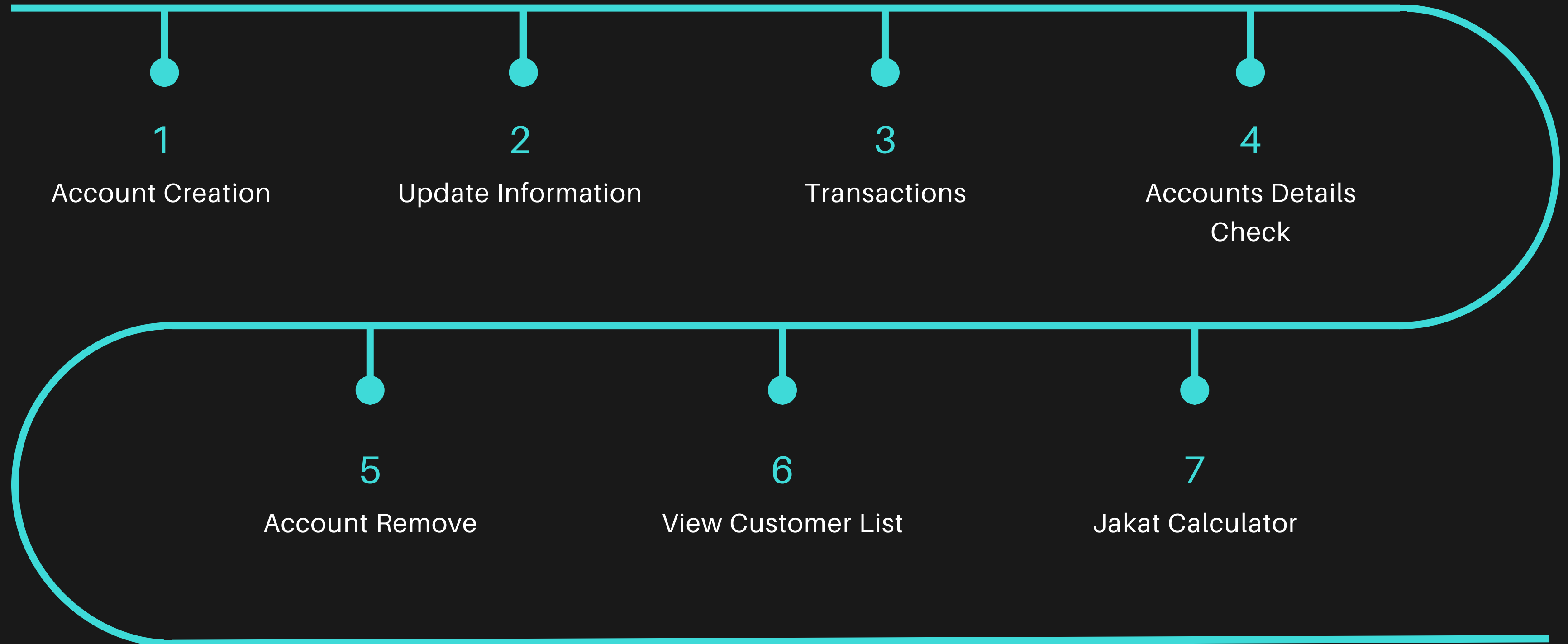
The ****Bank Management System**** is a software application designed to streamline and automate banking operations. It allows users to efficiently manage bank accounts by providing functionalities such as:

- Creating new accounts
- Updating account information
- Checking account details
- Removing accounts
- Handling deposits and withdrawals

This system aims to enhance accuracy, reduce operational time, and ensure secure management of customer information, ultimately improving the overall banking experience.



Key Features and Functionalities



User - Friendly Interface

SHAHRIAHA BASED BANK ACCOUNT MANAGEMENT SYSTEM

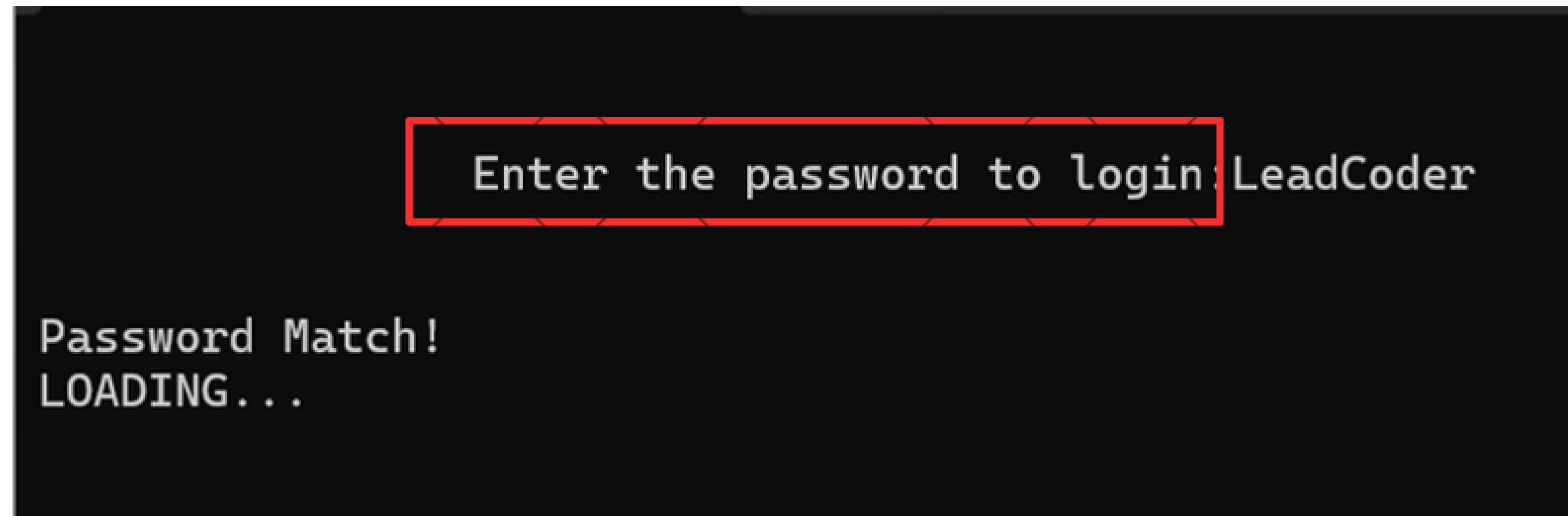
WELCOME TO THE MAIN MENU

- 1.Create new account
- 2.Update information of existing account
- 3.For transactions
- 4.Check the details of existing account
- 5.Jakat Calculator
- 6.Removing existing account
- 7.View customer's list
- 8.Exit

Enter your choice:|

Special Feature: Login with Password

Our Bank Management System includes a secure Login with Password feature. This ensures that only authorized users can access the system. Each user must enter a unique password to log in, enhancing the security and privacy of sensitive banking information. This feature helps prevent unauthorized access and ensures that account management tasks are performed by verified individuals, thereby protecting customer data and maintaining system integrity.



Short Description

The Shahriah Based Bank Account Management System is designed to manage bank accounts efficiently. It offers functionalities such as creating new accounts, updating existing account information, checking account details, removing accounts, and handling transactions like deposits and withdrawals and jakat calculation. A key feature of the system is the Login with Password, ensuring secure access to authorized users only. The system provides a user-friendly interface to streamline banking operations and improve overall customer experience.

Flowchart

Let's use this flowchart to map out our actions, identify roadblocks, and streamline our activities for better efficiency.



Library Used



```
graph TD; A[Library Used] --- B[stdio.h]; A --- C[windows.h]; A --- D[stdlib.h];
```

stdio.h

Standard input/output functions for reading and writing.

windows.h

Windows API functions for controlling the console.

stdlib.h

Standard Library functions for memory allocation, conversion, and other utility functions.

Implementations

Funtion Used :

- We tried to replicate the modern rendering technique called component based rendering using functions.

[illegible]

Implementations

Funtion Used :

- We tried to replicate the modern rendering technique called component based rendering using functions.

```
//from add
void new_acc()
{
    int choice;
    FILE *ptr;

    ptr=fopen("record.dat","a+");
account_no:
    system("cls");
    printf("\n\n\t\t\t\t\tADD RECORD \t\t\t\t\t\n\n");
    printf("\nEnter today's date(mm/dd/yyyy) : ");
    scanf("%d/%d/%d",&add.deposit.month,&add.deposit.day,&add.deposit.year);
    printf("\nEnter the account number : ");
    scanf("%d",&check.acc_no);
    while(fscanf(ptr,"%d %s %d/%d/%d %d %s %f %d/%d/%d\n",&add.acc_no,&add.name,&add.dob.month,&add.dob.day,&add.dob.year,&add.age,&add.address,&add.citizenship,&add.phone,&add.acctype,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)
    {
        if(check.acc_no==add.acc_no)
        {
            printf("\Account no. already in use!\n");
            fclose(ptr);
            goto account_no;
        }
    }
    add.acc_no=check.acc_no;
    printf("\nEnter the name : ");
    scanf("%s",&add.name);
    printf("\nEnter the date of birth(mm/dd/yyyy) : ");
    scanf("%d/%d/%d",&add.dob.month,&add.dob.day,&add.dob.year);
    printf("\nEnter the age : ");
    scanf("%d",&add.age);
    printf("\nEnter the address : ");
    scanf("%s",&add.address);
    printf("\nEnter the ID card number : ");
    scanf("%s",&add.citizenship);
    printf("\nEnter the phone number : ");
    scanf("%ld",&add.phone);
    printf("\nEnter the amount to deposit( Tk ) :
```

Structures Used for Managing Data :

`struct date`

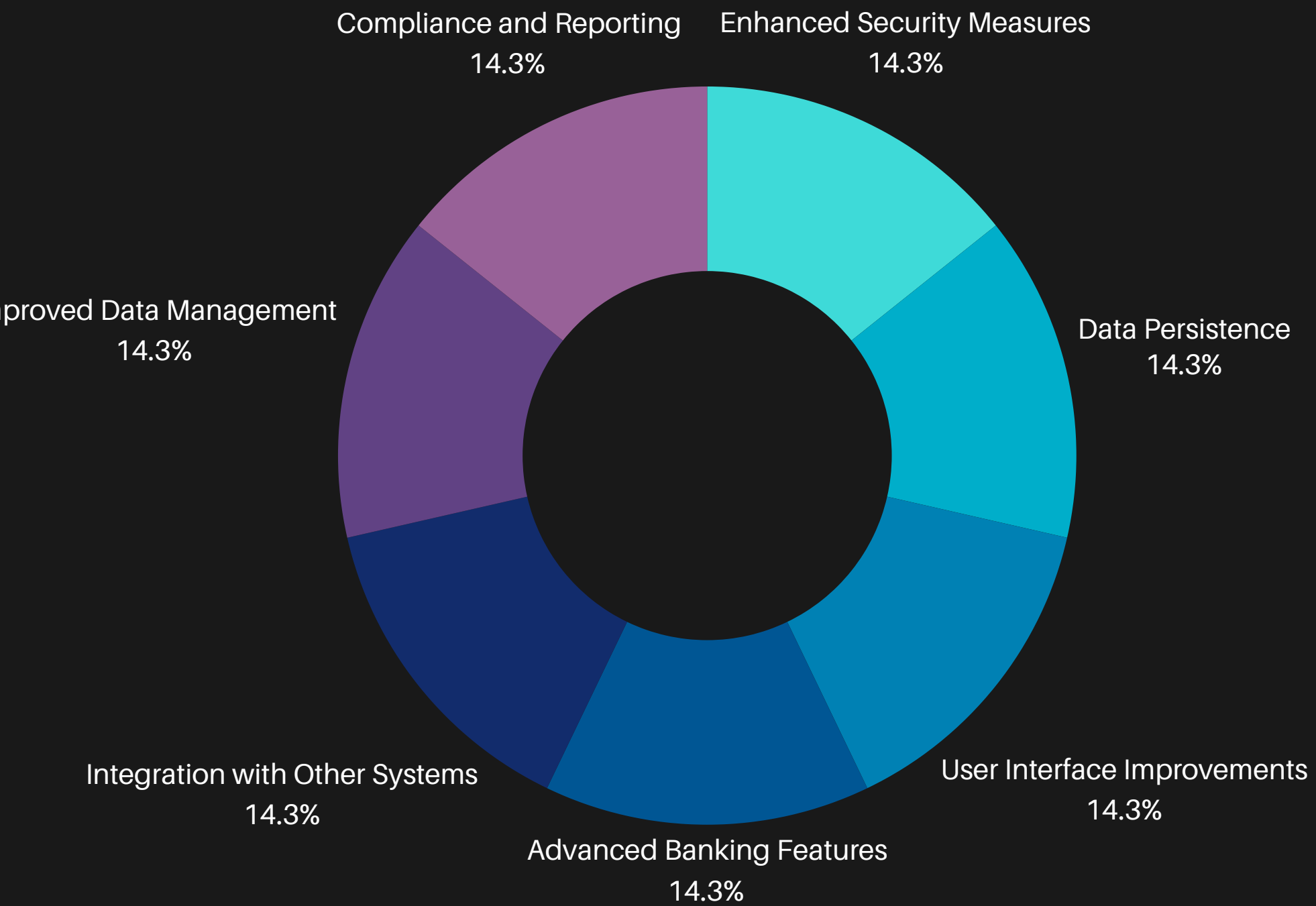
- Purpose: This structure is used to store date information, which includes the month, day, and year.
- Members:
 - ``int month``
 - ``int day``
 - ``int year``
- Usage: It is used to represent dates for account-related events like the date of birth, deposit dates, and withdrawal dates.

Structures Used for Managing Data

Main Account Structure

- Purpose: This unnamed structure is used to store all relevant details of a bank account. It organizes data related to a customer's account.
- Members:
 - ``char name[10000]``: Stores the account holder's name.
 - ``int acc_no``: Stores the account number.
 - ``int age``: Stores the account holder's age.
 - ``char address[60]``: Stores the account holder's address.
 - ``char citizenship[15]``: Stores the account holder's citizenship ID.
 - ``double phone``: Stores the account holder's phone number.
 - ``char acc_type[10]``: Stores the type of account (e.g., savings, current).
 - ``float amt``: Stores the current balance in the account.
 - ``struct date dob``: Stores the date of birth of the account holder.
 - ``struct date deposit``: Stores the date of the last deposit.
 - ``struct date withdraw``: Stores the date of the last withdrawal.
- Usage: This structure is used to manage and access all information related to a bank account. Instances of this structure (``add``, ``upd``, ``check``, ``rem``, ``transaction``) are used throughout the program to perform various operations like adding a new account, updating details, checking account information, removing accounts, and handling transactions.

Future Scopes of the Bank Management System



Conclusion

In conclusion, our project successfully developed a robust ****Shahriah Based Bank Account Management System**** using C, offering essential features such as account management, secure transactions, user authentication, and a user-friendly interface. While the current system provides a strong foundation, potential enhancements include improved error handling, data persistence, advanced security measures, a graphical user interface, advanced account management features, mobile compatibility, and integration with online banking. These improvements aim to make the system more versatile, secure, and user-friendly, ensuring a better banking experience for both employees and customers."

Always code as if the guy
who ends up maintaining
your code will be a violent
psychopath who knows
where you live.

— John Woods

THANK YOU

[SOURCE CODE](#)

