ds_lab_case_study.c   bank_data.txt

C.V. Raman Global University
Bhubaneswar,Odisha

Department Of Computer Science

```
#include<stdio.h>
int main()
{
        printf(“Group: 6B,Team: 14\n”);
        printf(“Topic: Bank Management
        system”);
```

Dr. Adyasha Rath & Mr. Jyoti Ranjan Swain

Team Members

Samar Kumar Singh	2201020575
Soumya Ranjan Biswal	2201020576
Ankit Kumar	2201020577
Sanjib Harichandan	2201020578
Warisha Mazhar	2201020579
Ramit Kumar Sahoo	2201020580

Introduction

The Bank Management System is a software application developed in the C programming language to facilitate the management of bank accounts. It provides functionalities such as creating new accounts, updating existing account information, conducting transactions (deposits and withdrawals), checking account details, removing accounts, and viewing a list of customer accounts.

File Handling

In C, a binary file is a file that contains data in a format that is not human-readable.

Here are some characteristics and key points about binary files in C:

No Text Formatting: Unlike text files, which store data in a human-readable format (like characters and strings), binary files store data in its raw, binary form.

Read and Write Operations: Binary files are typically opened in binary mode using functions like fopen, fread, fwrite, fclose, etc. The binary mode ("rb" for reading and "wb" for writing) ensures that data is read or written as raw bytes without any character encoding.

case_study.c

bank_data.txt

Header File’s & Structure Defining

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define FILE_NAME "bank_data.txt"
typedef struct
{
    char name[50];
    char email[50];
    char phone[15];
    char account_type[10];
    char address[100];
    char dob[11];
    char aadhar_no[13];
    char pan_no[11];
    int account_no;
    float balance;
} Account;
```

The code includes three standard libraries: `stdio.h` for input/output, `stdlib.h` for memory allocation, and `string.h` for string manipulation. It defines `FILE_NAME` as "bank_data.txt", likely for specifying a file to store bank account data.

Additionally, the code defines a structure called `Account`, which encompasses various fields (e.g., name, email, phone) to represent bank account holder information. This structure covers personal details, account specifics, and unique identifiers like Aadhar and PAN numbers.

Data Structure

Account Number Generator

```c
int accountNumber(){
    FILE *fp;
    fp = fopen(FILE_NAME, "rb");
    if (fp == NULL){
        printf("\n\n\t\t\tError in opening file!");
        return 0;
    }
    int accNo = 0;
    Account account;
    while (fread(&account, sizeof(Account), 1, fp)){
        accNo = account.account_no;
    }
    if (accNo == 0){
        return 1;
    }
    else{
        accNo++;
        return accNo;
    }
}
```

- It first tries to open a file named "bank_data.txt" in binary read mode ("rb").
- If the file opening operation fails (due to some error or if the file doesn't exist), it displays an error message and returns 0 to indicate an error.
- If the file is successfully opened, it initializes accNo (account number) to 0.
- It then reads each existing account's data from the file and updates accNo with the account number of the last account read.
- If no accounts are found (meaning accNo remains 0), it returns 1 as the first account number.
- If there are existing accounts, it increments accNo by one and returns it, providing a unique account number for a new account.

void create_account()

The function `create_account` prompts the user to input details for a new account. It assigns a unique account number, sets initial balance to zero, and writes the account information to a file. If there's an error opening the file, it displays an error message. Finally, it confirms successful account creation with the assigned account number.

void update_account()

The "update_account" function prompts the user to input an account number. It then attempts to open a file for reading and writing. If successful, it searches for the specified account. Upon finding it, the function offers a menu to update various account details. Depending on the choice, it prompts for new information and applies the update. Finally, it saves the changes to the file and notifies the user of the successful update. If the account is not found, it displays a message accordingly.

case_study.c bank_data.txt

void transaction()

The "transaction" function facilitates account transactions. It prompts for an account number and choice (deposit or withdraw). It then reads the account file, locates the specified account, and executes the chosen transaction. If successful, it updates the balance, saves it to the file, and displays the new balance. If the account is not found or there's an error, it provides appropriate feedback.

void check_account_details()

The function "check_account_details" prompts the user to enter an account number. It then searches for that account number in the file. If found, it prints out the associated account details, including name, email, phone, account type, address, date of birth, Aadhar number, PAN number, account number, and current balance. If the account is not found, it displays an error message. Finally, it closes the file.

case_study.c bank_data.txt

void remove_account()

The function "remove_account" prompts the user to input an account number for removal. It then opens the bank data file and a temporary file. It iterates through the accounts, copying all except the specified one to the temporary file. Afterward, it closes both files, replaces the original with the updated version, and notifies the user of successful account removal.

void view_customer_list()

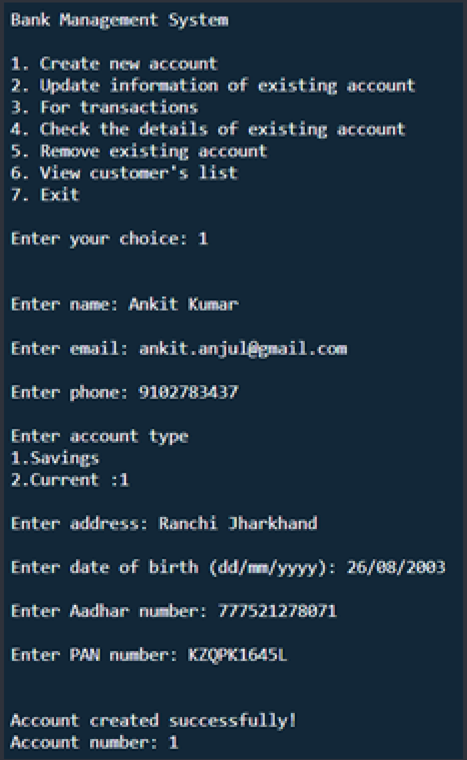
The function "view_customer_list" opens the file for reading, checks for errors, then iterates through each account record. It prints a formatted list of customer names, account numbers, and balances. Finally, it closes the file.

Main Function()

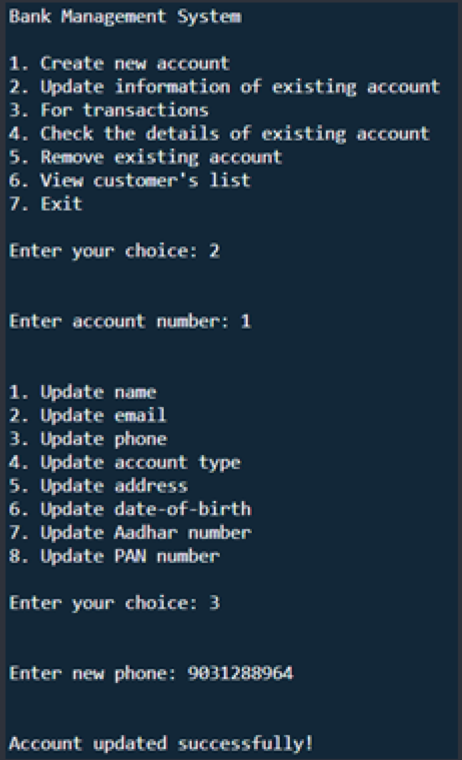
```
Bank Management System

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

Enter your choice:
```

This code defines the main function of a Bank Management System. It presents a menu to the user with options like creating accounts, updating information, conducting transactions, checking details, removing accounts, viewing customer list, and exiting. The program continues to prompt the user for choices in a loop until they choose to exit.

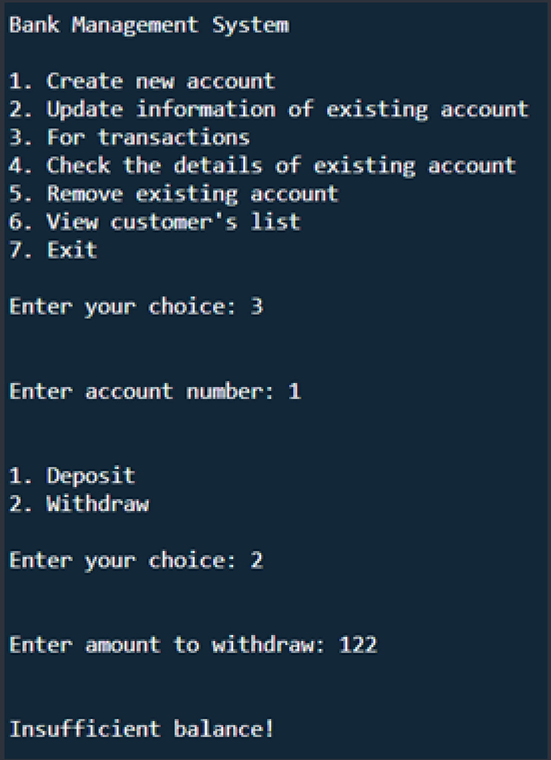
OUTPUT!!



```
Bank Management System

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

Enter your choice: 1


Enter name: Ankit Kumar

Enter email: ankit.anjul@gmail.com

Enter phone: 9102783437

Enter account type
1.Savings
2.Current :1

Enter address: Ranchi Jharkhand

Enter date of birth (dd/mm/yyyy): 26/08/2003

Enter Aadhar number: 777521278071

Enter PAN number: KZQPK1645L


Account created successfully!
Account number: 1
```



```
Bank Management System

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

Enter your choice: 2


Enter account number: 1


1. Update name
2. Update email
3. Update phone
4. Update account type
5. Update address
6. Update date-of-birth
7. Update Aadhar number
8. Update PAN number

Enter your choice: 3


Enter new phone: 9031288964


Account updated successfully!
```

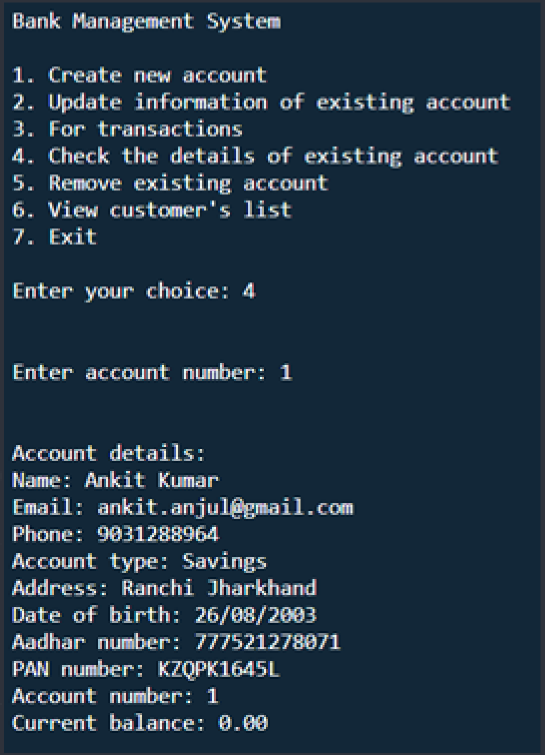


```
Bank Management System

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

Enter your choice: 3


Enter account number: 1


1. Deposit
2. Withdraw

Enter your choice: 2


Enter amount to withdraw: 122


Insufficient balance!
```

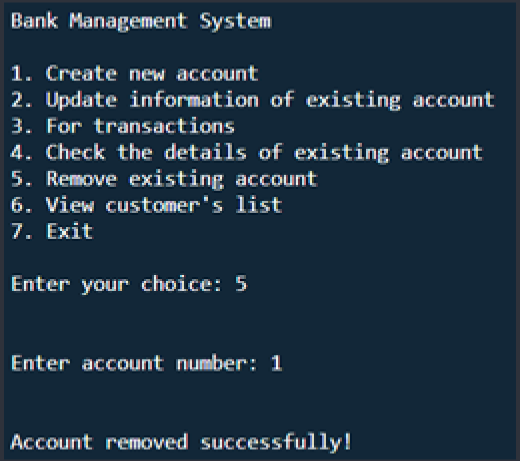
case_study.c

bank_data.txt

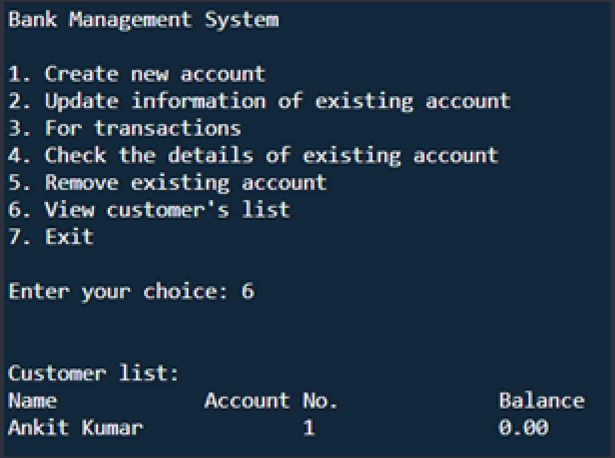
OUTPUT!!



```
Bank Management System

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

Enter your choice: 4


Enter account number: 1


Account details:
Name: Ankit Kumar
Email: ankit.anjul@gmail.com
Phone: 9031288964
Account type: Savings
Address: Ranchi Jharkhand
Date of birth: 26/08/2003
Aadhar number: 777521278071
PAN number: KZQPK1645L
Account number: 1
Current balance: 0.00
```



```
Bank Management System

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

Enter your choice: 5


Enter account number: 1


Account removed successfully!
```



```
Bank Management System

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

Enter your choice: 6


Customer list:
Name                Account No.              Balance
Ankit Kumar                   1              0.00
```

case_study.c

bank_data.txt

```
    Printf(“Thank You!!”);
}
```