IEEE RAS
ROBOTICS & AUTOMATION SOCIETY
Zagazig student chapter

Bank management system banque Ras

Meet our Team

Norhan yasser

Aya Ali

Abdelrahman

Ahmed

Today’s Highlights

1. main.c
2. functions.c
3. functions.h

1st Main.c

```c
int main()
{
   loadCustomers();
   int option;

    printf("\n");
   printf("\t\t\t Welcome to Banque Ras \n");
   printf("\t\t-------------------------------------\n");
   printf("\t\t1.create customer\n");
   printf("\t\t2.edit_customer \n");
   printf("\t\t3.print_customer_data \n");
   printf("\t\t4.delete_customer \n");
   printf("\t\t5.deposit\n");
   printf("\t\t6.money_transfer\n");
   printf("\t\t7.withdraw\n");
   printf("\t\t8.Exit\n\n");

   while(1){

   printf("\t\tEnter your option ");
   scanf("%d",&option);
```

```c
   switch (option) {
     case 1 :
        create_customer();
        break;
     case 2 :
        edit_customer();
        break;
     case 3 :
        print_customer_data();
        break;
     case 4 :
        delete_customer();
        break;
     case 5 :
         depositMoney();
         break;
     case 6 :
         transfer_money();
         break;
     case 7 :
         withdrawMoney();
         break;
     case 8 :
          return 0 ;
     default :
        printf("\n\t\tInvalid option , please choose from the menu\n");
        break;

   } }

    return 0 ;
};
```

2nd

Functions.c

1. Create customer
2. Edit customer
3. Print customer data
4. Delete customer
5. Deposit money
6. Transfer money
7. Withdraw money

Create customer

```c
void  create_customer(){

    if (customerCount >= 100) {
        printf("\t\tCustomer list is full.\n");
        return;
    }
    struct Customer c;
    int tryCount=0;

    while(1) {
        if (tryCount==3){
            printf("\t\tTry again later \n");
            return;
        }
        printf("\t\tEnter Customer ID: ");
        scanf("%d", &searchId);
        tryCount++;
        if (!idCheck(searchId)) {
            printf("\t\tID must be exactly 4 digits.\n");
            continue;
        }
        if (findCustomerById(searchId)!=-1){
            printf("\t\tThis ID already exists .\n");
        } else {
            c.id=searchId;
            break ;}
        }

    printf("\t\tEnter name: ");
    scanf(" %[^\n]", c.name);
```

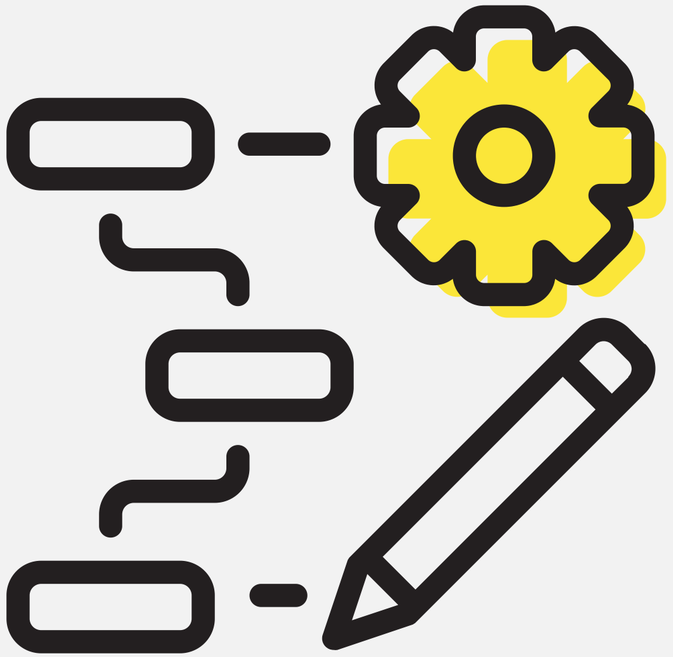
```c
    while(1) {
        printf("\t\tEnter cash: ");
        scanf("%lf", &c.balance);
        if (c.balance <= 0) {
            printf("\t\tCash must be positive.\n");
        }else { break ;}
    }

    printf("\t\tEnter phone: ");
    scanf(" %[^\n]", c.phone);

    customers[customerCount] = c;
    customerCount++;

    saveCustomers();
    printf("\t\tCustomer added successfully.\n");
}
```

Edit customer



```c
void edit_customer(){
    printf("\t\tEnter customer ID to update te data : ");
    scanf("%d", &searchId);

    int index = findCustomerById(searchId);

    if( index ==-1){
        printf("\t\tCustomer with ID %d not found.\n", searchId);
        return;
    }

    printf("\t\tEnter new name: ");
    scanf(" %[^\n]", customers[index].name);

    printf("\t\tEnter new phone: ");
    scanf(" %[^\n]", customers[index].phone);

    saveCustomers();

    printf("\t\tCustomer updated.\n");
    return;
}
```

Print customer data

```c
void print_customer_data(){
    printf("\t\tEnter customer ID to print data : ");
    scanf("%d", &searchId);
    int index = findCustomerById(searchId);

    if(index == -1){
        printf("\t\tYOUR ACCOUNT IS NOT AVAILABLE.\n");
    }else {
        for(int i = 0; i < 4; i++){
            printf("\n\t\t\t=== Account Details ===\n");
            printf("\t\tID    : %d\n", customers[index].id);
            printf("\t\tName  : %s\n", customers[index].name);
            printf("\t\tCash  : %.2f\n", customers[index].balance);
            printf("\t\tphone : %s\n", customers[index].phone);
            printf("\t\t==========================\n");
            break;
        }
    }
}
```

Delete customer

```c
void delete_customer(){
    printf("\t\tEnter customer ID to be deleted: ");
    scanf("%d", &searchId);
    int index = findCustomerById(searchId);

    if( index ==-1){
        printf("\t\tCustomer with ID %d not found.\n", searchId);
        return;
    }

    for (int j = index; j < customerCount - 1; j++) {
        customers[j] = customers[j + 1];
    }

    customerCount--;

    saveCustomers();

    printf("\t\tCustomer deleted.\n");
}
```

Deposit

```c
void depositMoney() {
    double amount;
    int tryCount=0;
    int index;
    while(1) {
        if (tryCount==3){
            printf("\t\tTry again later \n");
            return;
        }
        printf("\t\tEnter Customer ID: ");
        scanf("%d", &searchId);
        tryCount++;
        index = findCustomerById(searchId);
        if (index == -1) {
            printf("\t\tCustomer not found.\n");
            continue;
        }else {
            break;
        }
    }
    printf("\t\tEnter amount to deposit: ");
    scanf("%lf", &amount);

    if (amount <= 0) {
        printf("\t\tInvalid amount.\n");
        return;
    }
    customers[index].balance += amount;

    saveCustomers();

    printf("\t\tDeposit successful. New balance: $%.2f\n", customers[index].balance);
}
```

Transfer

```c
void transfer_money(){
    double amount;
    int indexSender;
    int tryCount ;
    while(1) {
        if (tryCount==3){
            printf("\t\tTry again later \n");
            return;
        }
        printf("\t\tEnter Customer ID: ");
        scanf("%d", &searchId);
        tryCount++;

        indexSender = findCustomerById(searchId);

        if (indexSender == -1) {
            printf("\t\tCustomer not found.\n");
            continue;
        }else {
            break;
        }
    }
    printf("\t\tEnter ID of the receiver:");
    scanf("%d", &searchId);

    int indexRecievier = findCustomerById(searchId);
    if (indexRecievier ==-1) {
        printf("\t\tCustomer not found.\n");
        return;
    }
    if (customers[indexSender].id == customers[indexRecievier].id) {
        printf("\t\tError: Cannot transfer to the same person.\n");
        return ;
    }
    printf("\t\tEnter amount to transfer: ");
    scanf("%lf", &amount);

    if (amount <= 0) {
        printf("\t\tInvalid amount.\n");
        return;
    }
    if (customers[indexSender].balance < amount){
        printf("\t\tError: You don't have enough money.\n");
        return;
    }
    customers[indexSender].balance -= amount;
    customers[indexRecievier].balance += amount;

    printf("\t\tTransfer sucsessful.\n");
    printf("\t\tFrom: %s (ID: %d)\n",  customers[indexSender].name,  customers[indexSender].id);
    printf("\t\tTo: %s (ID: %d)\n", customers[indexRecievier].name, customers[indexRecievier].id);
    printf("\t\tAmount: $%.2f\n", amount);
    printf("\t\tNew balance - %s: $%.2f\n", customers[indexSender].name, customers[indexSender].balance);
    printf("\t\tNew balance - %s: $%.2f\n", customers[indexRecievier].name, customers[indexRecievier].balance);
}
```

Withdraw


```c
void withdrawMoney() {
    double amount;
    int index;
    int tryCount=0;
    while(1) {
        if (tryCount==3){
            printf("\t\tTry again later \n");
            return;
        }
        printf("\t\tEnter Customer ID: ");
        scanf("%d", &searchId);
        tryCount++;

        index = findCustomerById(searchId);
        if (index == -1) {
            printf("\t\tCustomer not found.\n");
            continue;
        }else {
            break;
        }
    }
    printf("\t\tEnter amount to withdraw: ");
    scanf("%lf", &amount);

    if (amount <= 0) {
        printf("\t\tInvalid amount.\n");
        return;
    }
    if (customers[index].balance < amount){

        printf("\t\tError: You don't have enough money.\n");
        return;
    }
    customers[index].balance -= amount;

    saveCustomers();

    printf("\t\tWithdraw successful. New balance: $%.2f\n", customers[index].balance);
}
```

Save customers

```c
void saveCustomers() {
    FILE *customerFileP = fopen("customers.txt", "w");

    for (int i = 0; i < customerCount; i++) {
        fprintf(customerFileP, "%d %s %.2f %s\n", customers[i].id,
                customers[i].name, customers[i].balance, customers[i].phone);
    }
    fclose(customerFileP);
}
```

Find customer

```c
int findCustomerById(int id) {
    for (int i = 0; i < customerCount; i++) {

        if (customers[i].id == id) return i;
    }
    return -1;
}
```

Load customers

```c
void loadCustomers() {
    FILE *customerFileP = fopen("customers.txt", "r");
    if (!customerFileP) return;

    customerCount = 0;
    while (fscanf(customerFileP, "%d %s %lf %s", &customers[customerCount].id,
                  customers[customerCount].name,
                  &customers[customerCount].balance,&customers[customerCount].phone) == 4) {
        customerCount++;
    }
    fclose(customerFileP);
}
```

3rd functions.h

```
#ifndef FUNCTIONS_H      // If FUNCTIONS_H is not defined
#define FUNCTIONS_H      // Define it now

void create_customer();

void edit_customer();

void print_customer_data();

void  delete_customer();

void deposite();

void transfer_money();

void withdrawMoney();

void loadCustomers() ;

void saveCustomers();

void depositMoney();

bool idCheck (int id);


#endif                   // End of guard
```

Thank you!