Mini-Project Design

Record Structures

Customer Record

```
struct Customer {
    int id; // 0, 1, 2 ....
    char name[25];
    char gender; // M -> Male, F -> Female, O -> Other
    int age;
    // Login Credentials
    char login[30]; // Format : name-id (name will the first word in the
structure member `name`)
    char password[30];
    // Bank data
    int account; // Account number of the account the customer owns
}
```

Account Record

```
struct Account {
    int accountNumber; // 0, 1, 2, ....
    int owners[2]; // Customer IDs
    bool isRegularAccount; // 1 -> Regular account, 0 -> Joint account
    bool active; // 1 -> Active, 0 -> Deactivated (Deleted)
    long int balance; // Amount of money in the account
}
```

Admin Functions

Admin Login

```
1. Input : Admin login ID
2. Input : Admin password
3. Compare with harcoded admin credentials
4. If matching, show admin menu
```

Get Account Details

```
1. Input : accountNumber
2. `lseek` to the accountNumber
3.1. Lock the record for reading
3.2. Read the account details from Account file into a buffer (of type `struct
Account`).
3.2 Unlock the record
4. Print account details
```

Add account

```
1. Input : Account type (Regular / Joint)
2. Generate new account number
    2.1. If Regular account then go to 3
    2.2 If Joint then go to 4
3. Input : Customer Information for 2 people
4. Input : Customer Information for 1 person
5. Add Customer information into Customer file
6. Add the customer IDs into `owner` field of account
7. Input : Opening balance of account
8. Set `active` status of account to `true`
9. Write account into Account file
10. Print customer & account info to the admin
```

Delete Account

```
1. Input : Account number
2. `lseek` into account in the Account file
3. Lock the record for writing
4. Read account into `struct Account` buffer
4. Change account `active` status to false (to simulate delete)
5. Write back update account information back into the Account file
   (Make sure to `lseek` back to the start of the record)
6. Unlock the record
```

Get Customer Details

```
1. Input : Customer ID
2. `lseek` into customer in the Customer file
3. Lock the record for reading
4. Read the customer into `struct Customer` buffer
5. Unlock the record
6. Print the details
```

Modify Customer Information

```
1. Input : Customer ID
2. `lseek` into customer in the Customer file
3. Lock the record for writing
4. Read the customer into `struct Customer` buffer
5. The admin can only alter Name, Gender & Age of the customer
    5.1 Input : Which field to alter
    5.2 Input : New value
6. Update value in the buffer
7. Write the buffer back into the file
   (Make sure to `lseek` back to the start of the record)
8. Unlock the record
```

Customer Functions

Customer Login

```
1. Input : Login ID
2. Get the customer information by `lseek`ing in to the customer record. If
found, go to next step.
3. Lock the record for reading
4. Store this customer information into a `struct customer` Buffer. This data
will persist till the end of client session.
5. Unlock the record
6. Input : Password
7. Compare the input password to the password stored in the buffer. If matching,
give the client the customer menu.
```

View Details

```
Make a call to `Get Customer Details` described above. Pass the logged in
customer's ID.
```

Deposit Money

```
Input : Money to deposit
1. Use the logged in customer's information to find the associated bank account
number.
2. `lseek` into the account in the `Account file`
3. Lock the record for writing
4. Read the account details into a `struct Account` buffer
5. Add the deposit money to the balance field
6. Write the updated account details back into the Account file.
7. Unlock the record
```

Withdraw Money

```
Input : Money to withdraw
1. Use the logged in customer' information to find the associated bank account
number.
2. `lseek` into the account in the `Account file`
3. Lock the record for writing
4. Read the account details into a `struct Account` buffer.
5. Deduct the money from the balance field (if the deduction doesn't result in a
negative amount)
6. Write the updated account details back into the Account file.
7. Unlock the record
```

Balance Enquiry

```
Make a call to the `Get Account Details` described above. Pass the logged in
customer's account number.
```

Change Password

```
1. Input : Existing password
2. Compare the input password to the password in the buffer. If it matches go to
next step.
3. Input : New Password
4. Input : Re-entered new password
5. Check if new password & re-entered new password matches. If yes, go to next
step.
6. Lock the record for writing
7. Write new password into buffer.
8. Write updated customer information into Customer file
9. Unlock the record
```