INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

INTRODUCTION TO C PROGRAMMING

INSTRUCTIONS TO CANDIDATES:

1 **Submit your assignment at the administrative counter**

2 **Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing)**

3 **Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld**

4 **Cases of plagiarism will be penalized**

5 **The assignment should be bound in an appropriate style (comb bound or stapled).**

6 **Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.**

7 **You must obtain 50% overall to pass this module.**

Table of Contents

Introduction

This literature is an attempt to explain the Banking System program through its functionality, by way of showcasing the features implemented and having diagrams to show how they interact, as well as explaining the process of building the program, by way of pseudocode, and explanations along with snippets of the source code.

The program is a simulation of a real world banking system with all CRUD operations allowing users to create their own profiles with their own information, which can then later be read by logging in. Users are also able to simulate depositing money into their accounts and withdrawing that money.

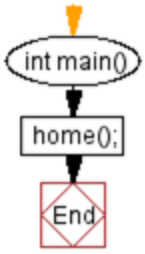
As with any information system, the program features an admin page, which allows an administrator of the system to view the transaction history recorded by the system, as well as the ability to remove any user.

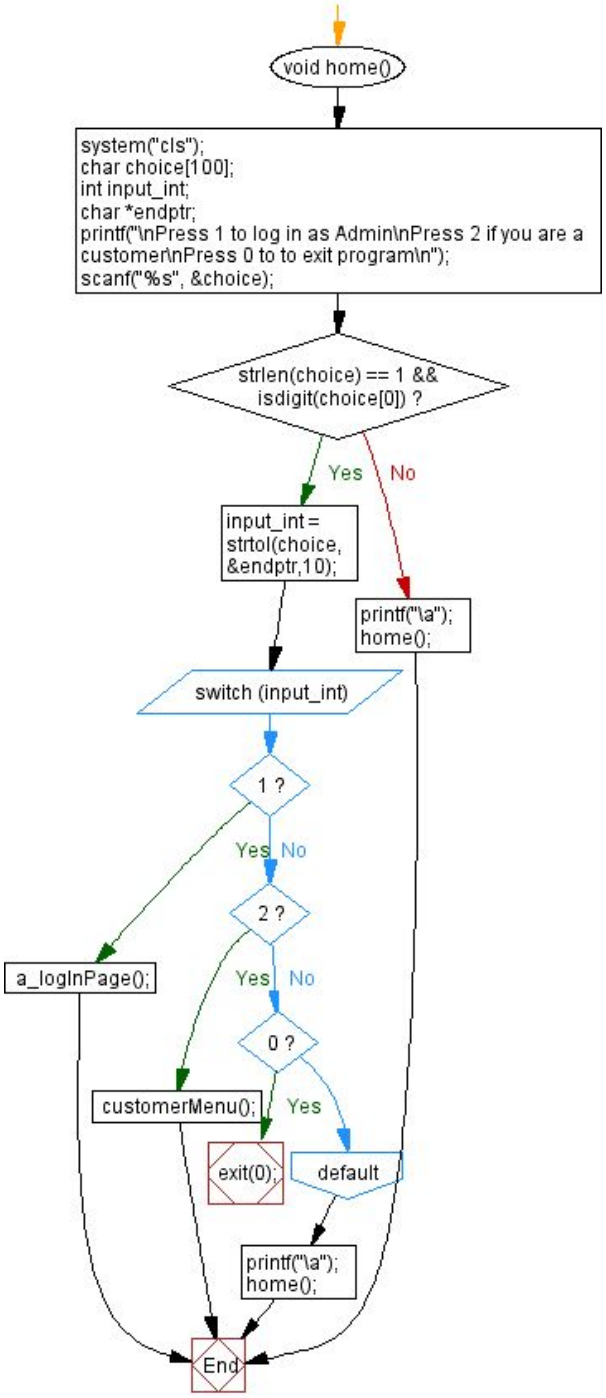
Assumptions

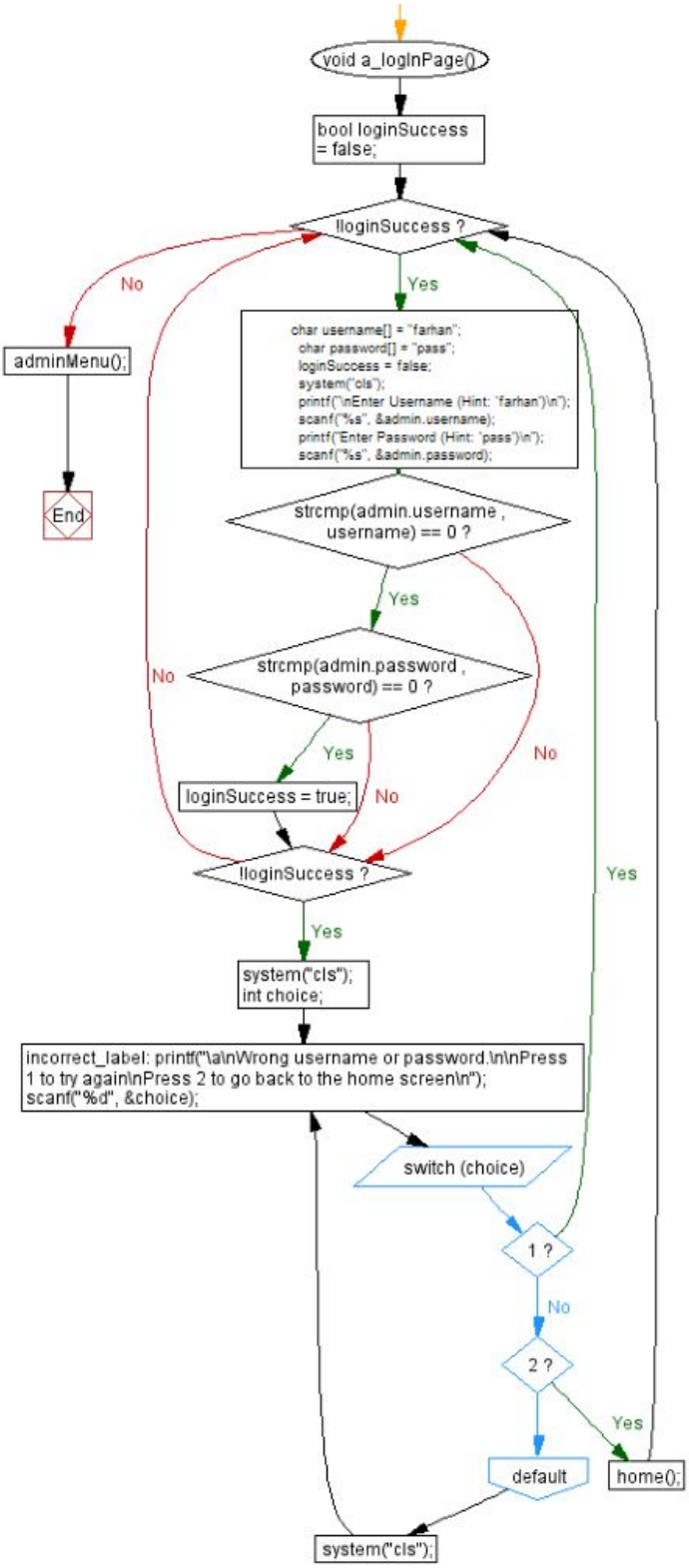
The program has been written under the assumption that no real money will be deposited or withdrawn from the system. The fiscal amounts entered into the system are arbitrary and only serve as a simulation of real world banking transactions.
The maximum value that can be deposited or withdrawn per transaction is 500rm.
The administrator credentials are hardcoded in to be “farhan” for the username, and “pass” for the password.
Users can update their name and mobile number, but not their gender.
The program is rendered on a terminal, which can be viewed through a console.
The data from the program is written to and read from a ‘.txt’ file.

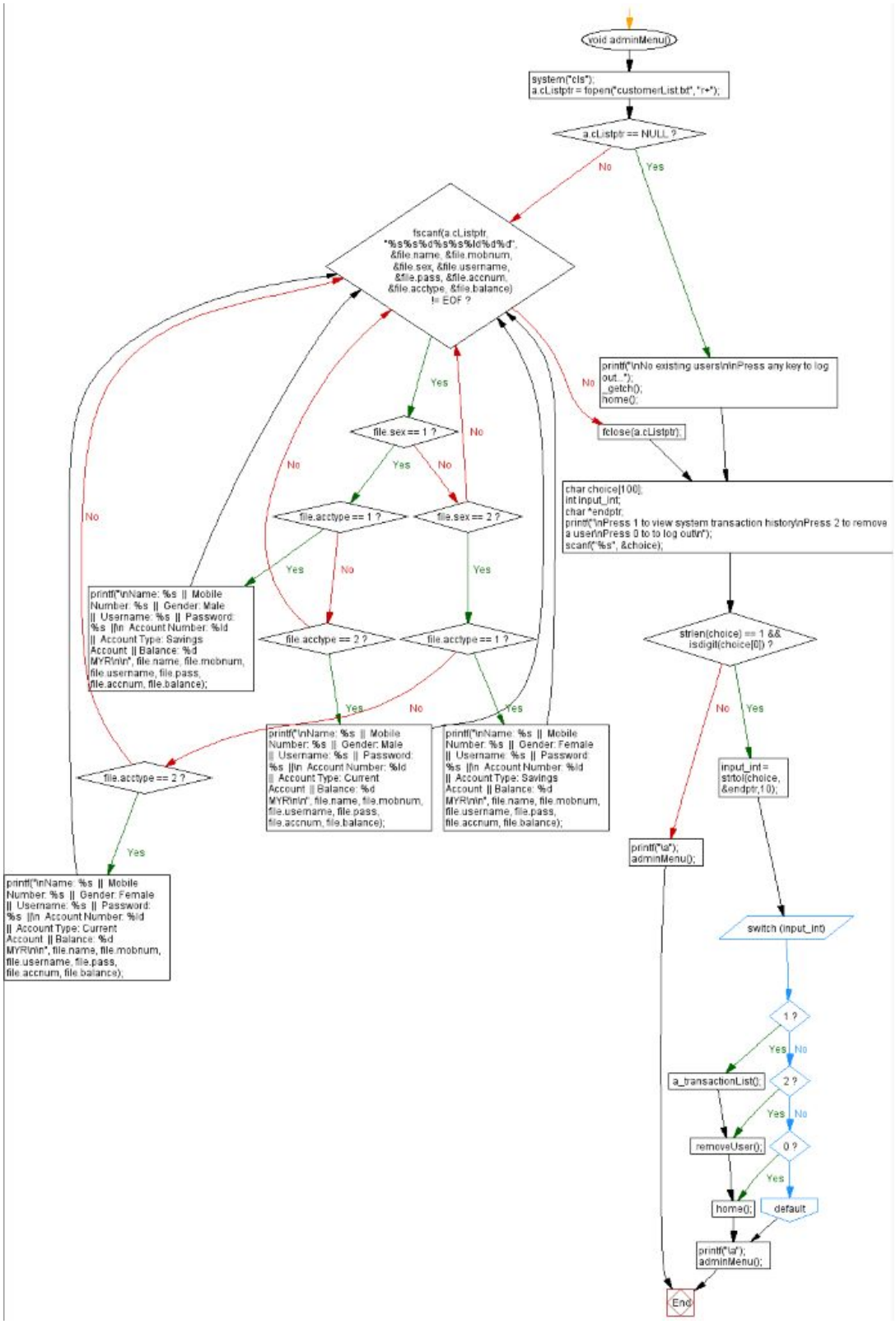
Design

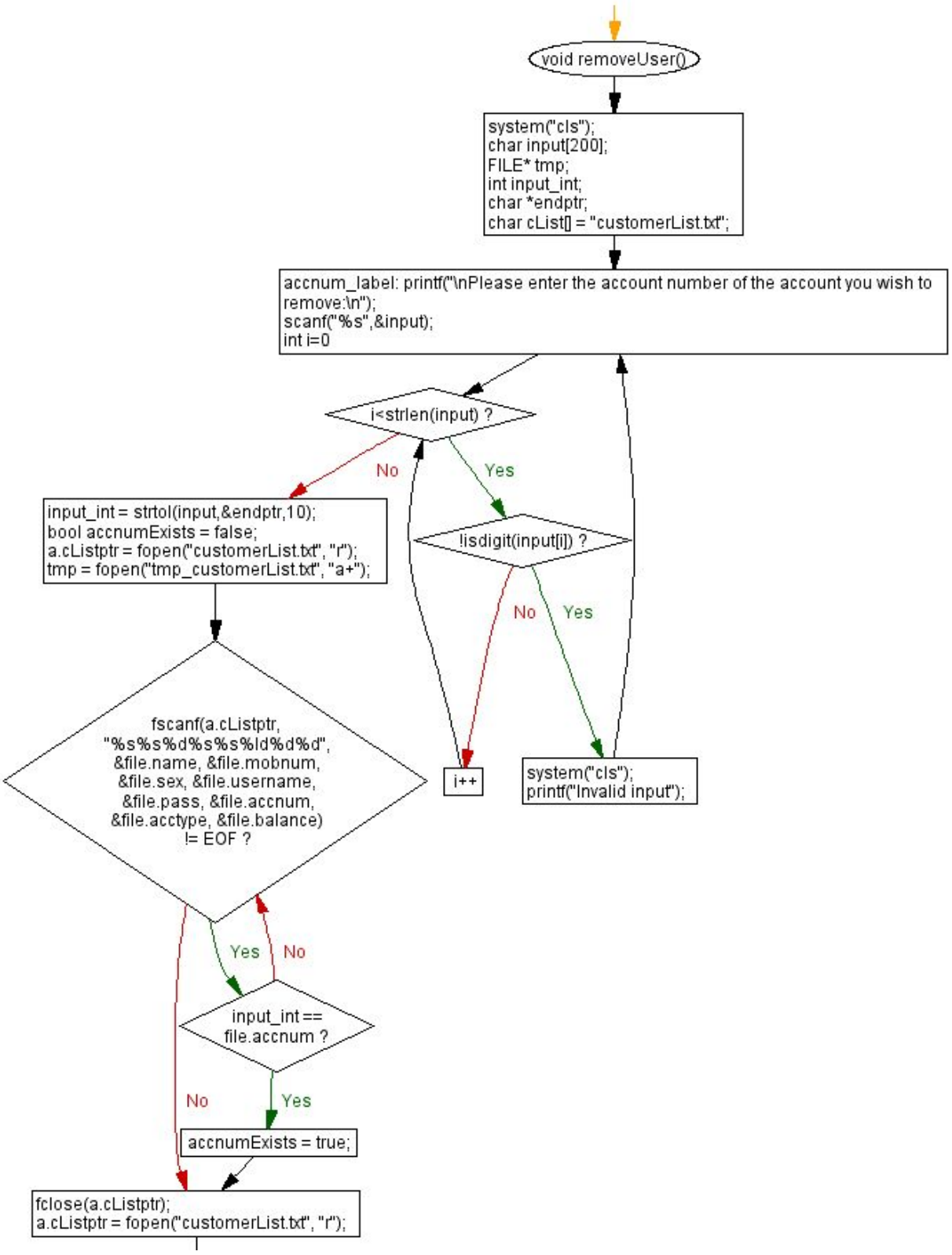
Flowchart

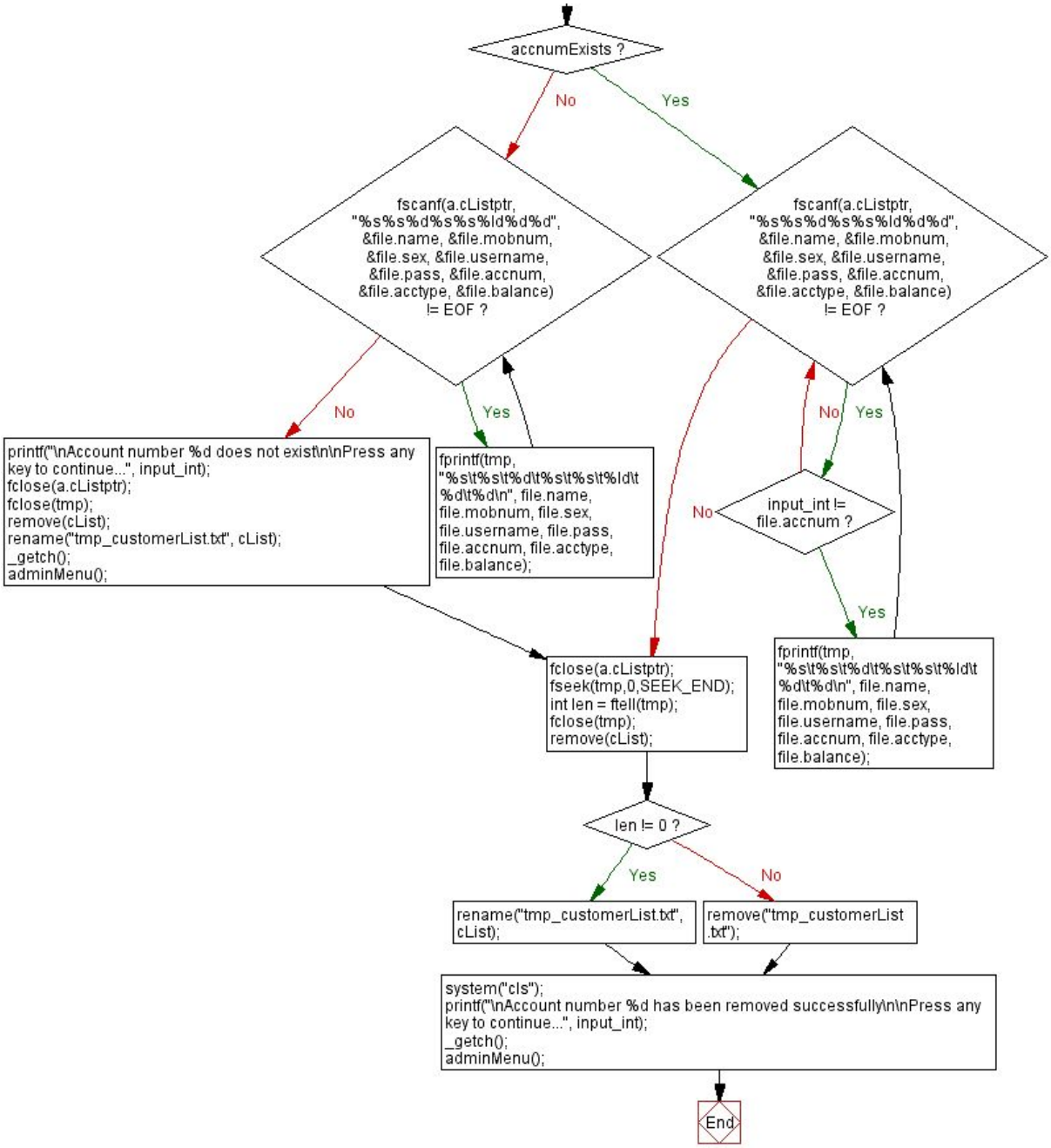


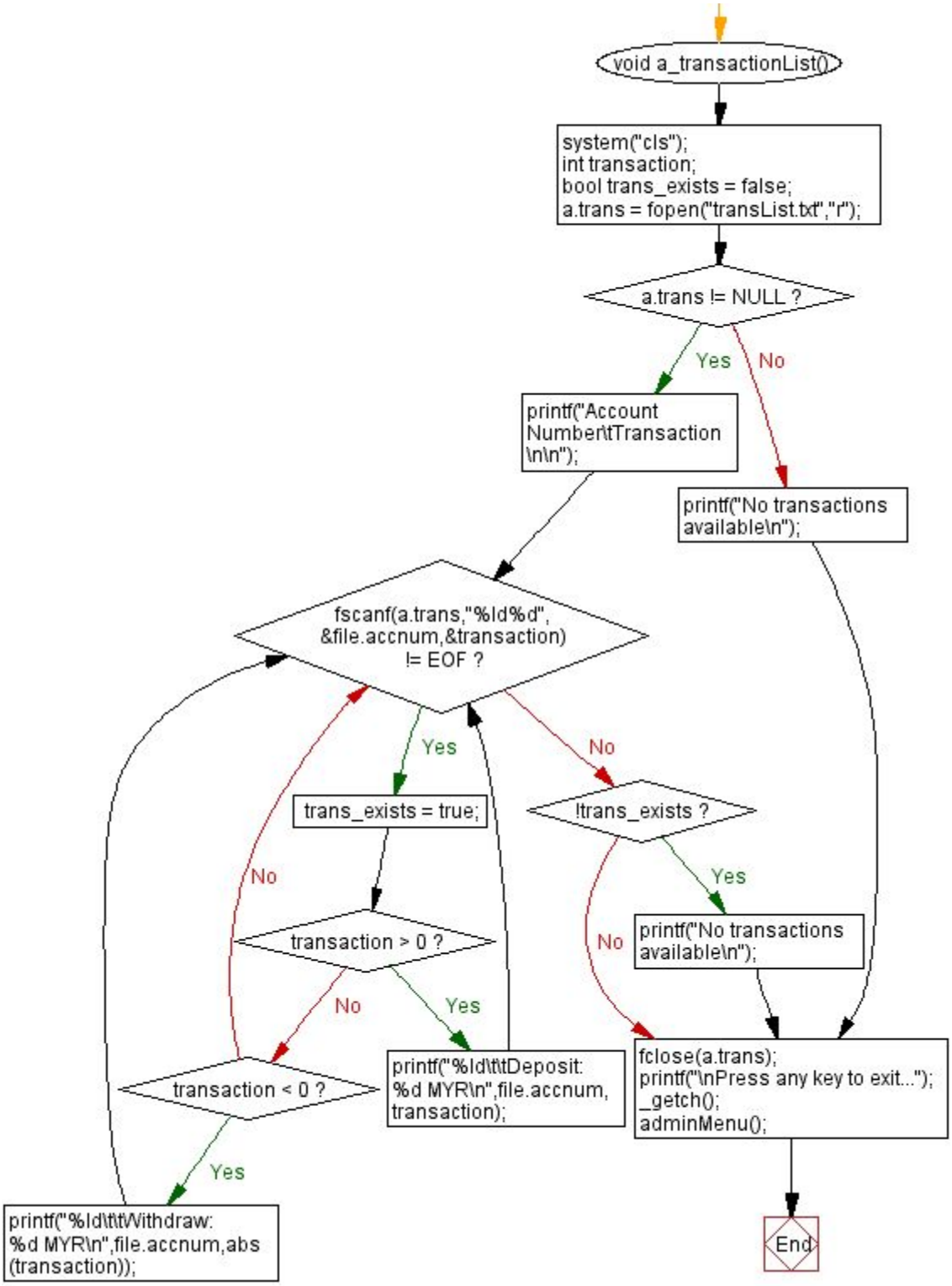


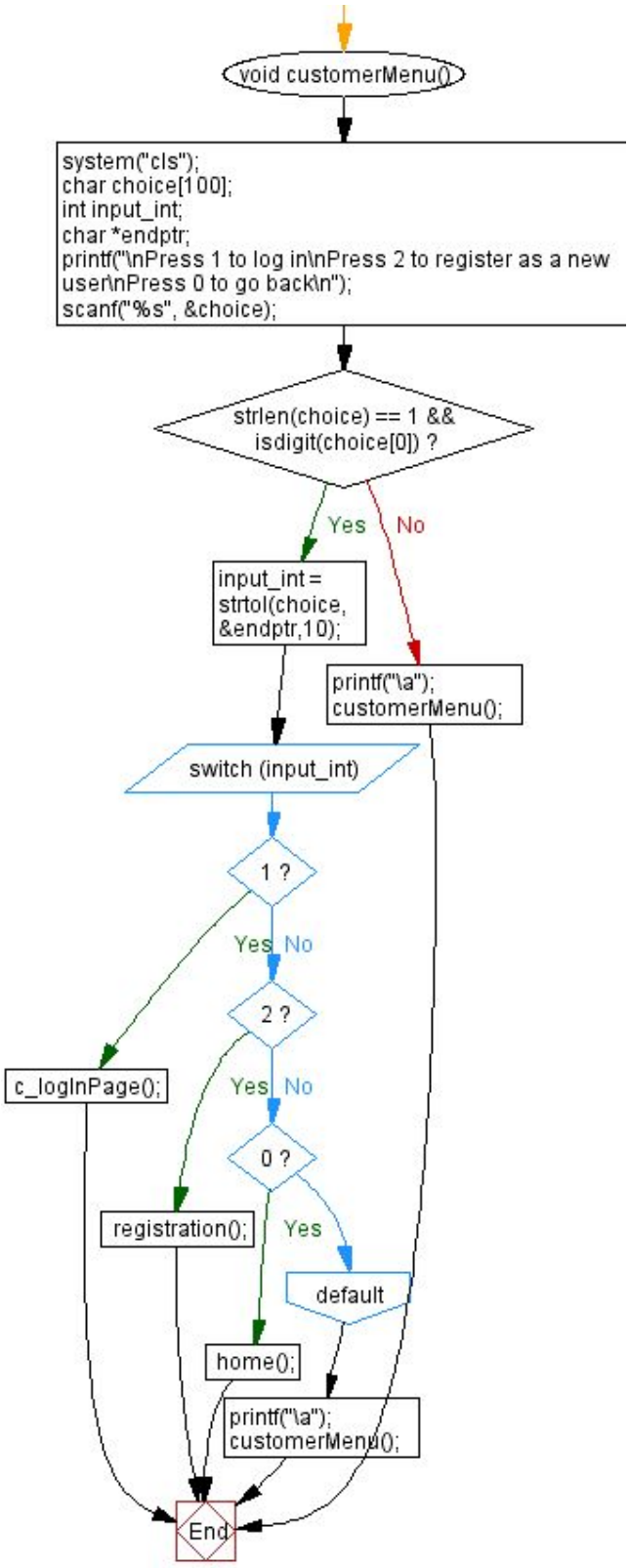

void a_logInPage()
bool loginSuccess = false;
!loginSuccess ?
No
Yes
adminMenu();
End
char username[] = "farhan";
char password[] = "pass";
loginSuccess = false;
system("cls");
printf("\nEnter Username (Hint: 'farhan')\n");
scanf("%s", &admin.username);
printf("Enter Password (Hint: 'pass')\n");
scanf("%s", &admin.password);
strcmp(admin.username , username) == 0 ?
Yes
No
strcmp(admin.password , password) == 0 ?
Yes
No
loginSuccess = true;
No
!loginSuccess ?
Yes
Yes
system("cls");
int choice;
incorrect_label: printf("\a\nWrong username or password.\n\nPress 1 to try again\nPress 2 to go back to the home screen\n");
scanf("%d", &choice);
switch (choice)
1 ?
No
2 ?
Yes
default
home();
system("cls");

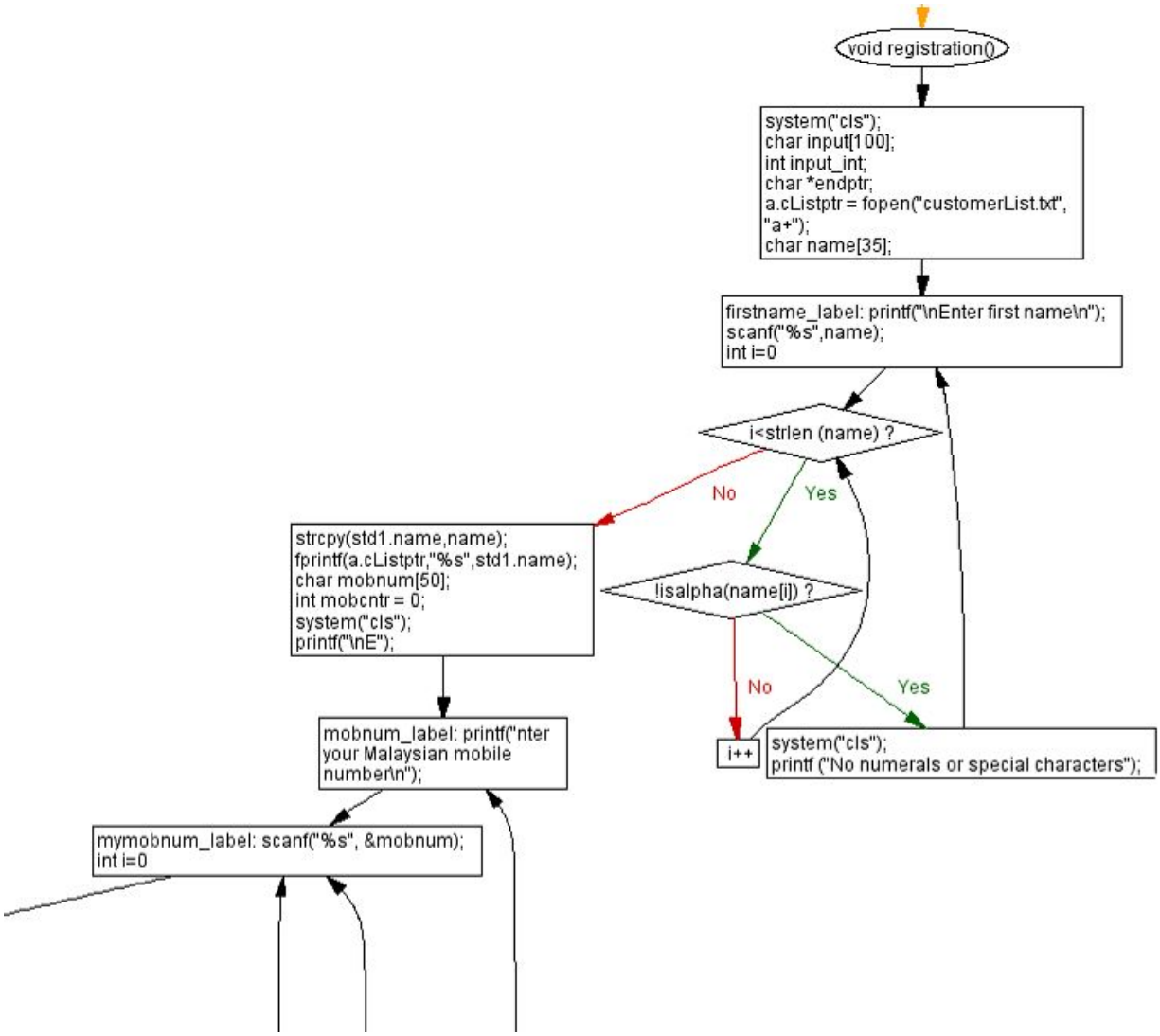

void adminMenu();
system("cls");
a.cList.ptr = fopen("customerList.txt", "r+");
a.cList.ptr == NULL ?
No
Yes
fscanf(a.cList.ptr, "%s%s%d%s%s%lc%d%d", &file.name, &file.mobnum, &file.sex, &file.username, &file.pass, &file.accnum, &file.acctype, &file.balance) != EOF ?
Yes
No
printf("\nNo existing users\n\nPress any key to log out...");
_getch();
home();
fclose(a.cList.ptr);
file.sex == 1 ?
Yes
No
file.acctype == 1 ?
file.sex == 2 ?
file.acctype == 2 ?
file.acctype == 1 ?
printf("\nName: %s || Mobile Number: %s || Gender: Male || Username: %s || Password: %s ||\n Account Number: %ld || Account Type: Savings Account || Balance: %d MYR\n\n", file.name, file.mobnum, file.username, file.pass, file.accnum, file.balance);
printf("\nName: %s || Mobile Number: %s || Gender: Male || Username: %s || Password: %s ||\n Account Number: %ld || Account Type: Current Account || Balance: %d MYR\n\n", file.name, file.mobnum, file.username, file.pass, file.accnum, file.balance);
printf("\nName: %s || Mobile Number: %s || Gender: Female || Username: %s || Password: %s ||\n Account Number: %ld || Account Type: Savings Account || Balance: %d MYR\n\n", file.name, file.mobnum, file.username, file.pass, file.accnum, file.balance);
printf("\nName: %s || Mobile Number: %s || Gender: Female || Username: %s || Password: %s ||\n Account Number: %ld || Account Type: Current Account || Balance: %d MYR\n\n", file.name, file.mobnum, file.username, file.pass, file.accnum, file.balance);
char choice[100];
int input_int;
char *endptr;
printf("\nPress 1 to view system transaction history\nPress 2 to remove a user\nPress 0 to to log out\n");
scanf("%s", &choice);
strlen(choice) == 1 && isdigit(choice[0]) ?
input_int = strtol(choice, &endptr,10);
printf("\a");
adminMenu();
switch (input_int)
1 ?
2 ?
0 ?
a_transactionList();
removeUser();
home();
default
printf("\a");
adminMenu();
End

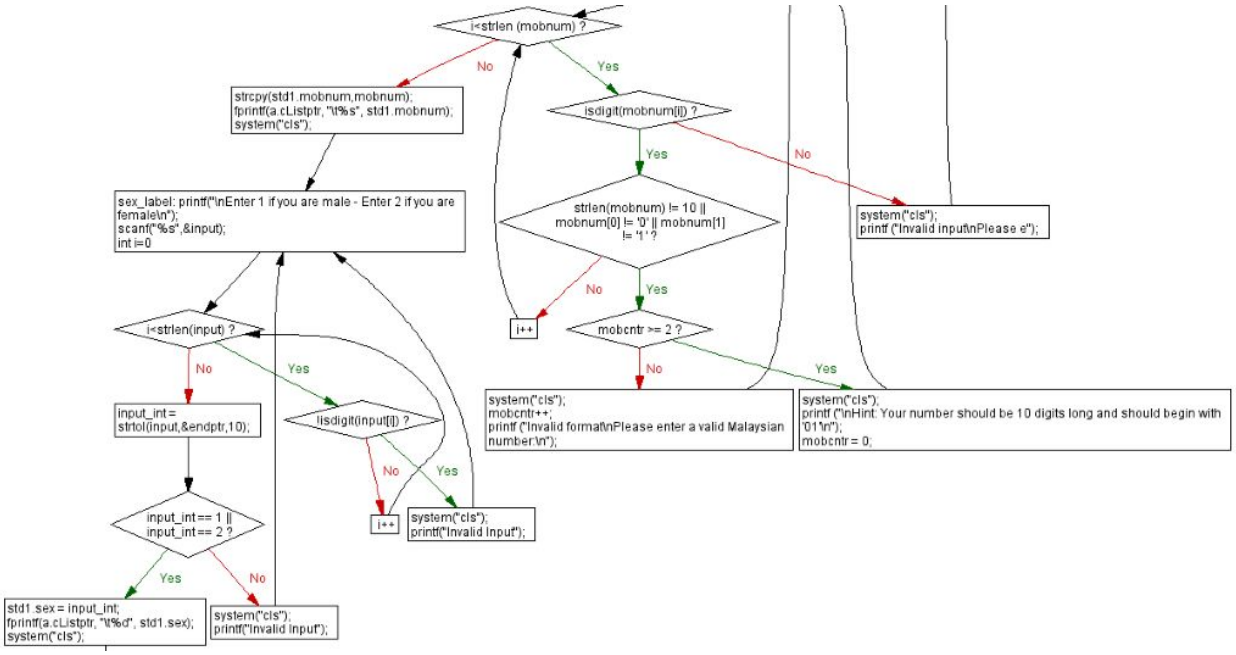

void removeUser()
system("cls");
char input[200];
FILE* tmp;
int input_int;
char *endptr;
char cList[] = "customerList.txt";
accnum_label: printf("\nPlease enter the account number of the account you wish to remove:\n");
scanf("%s",&input);
int i=0
i<strlen(input) ?
No
Yes
!isdigit(input[i]) ?
No
Yes
i++
system("cls");
printf("Invalid input");
input_int = strtol(input,&endptr,10);
bool accnumExists = false;
a.cListptr = fopen("customerList.txt", "r");
tmp = fopen("tmp_customerList.txt", "a+");
fscanf(a.cListptr, "%s%s%d%s%s%ld%d%d", &file.name, &file.mobnum, &file.sex, &file.username, &file.pass, &file.accnum, &file.acctype, &file.balance) != EOF ?
Yes
No
input_int == file.accnum ?
No
Yes
accnumExists = true;
fclose(a.cListptr);
a.cListptr = fopen("customerList.txt", "r");

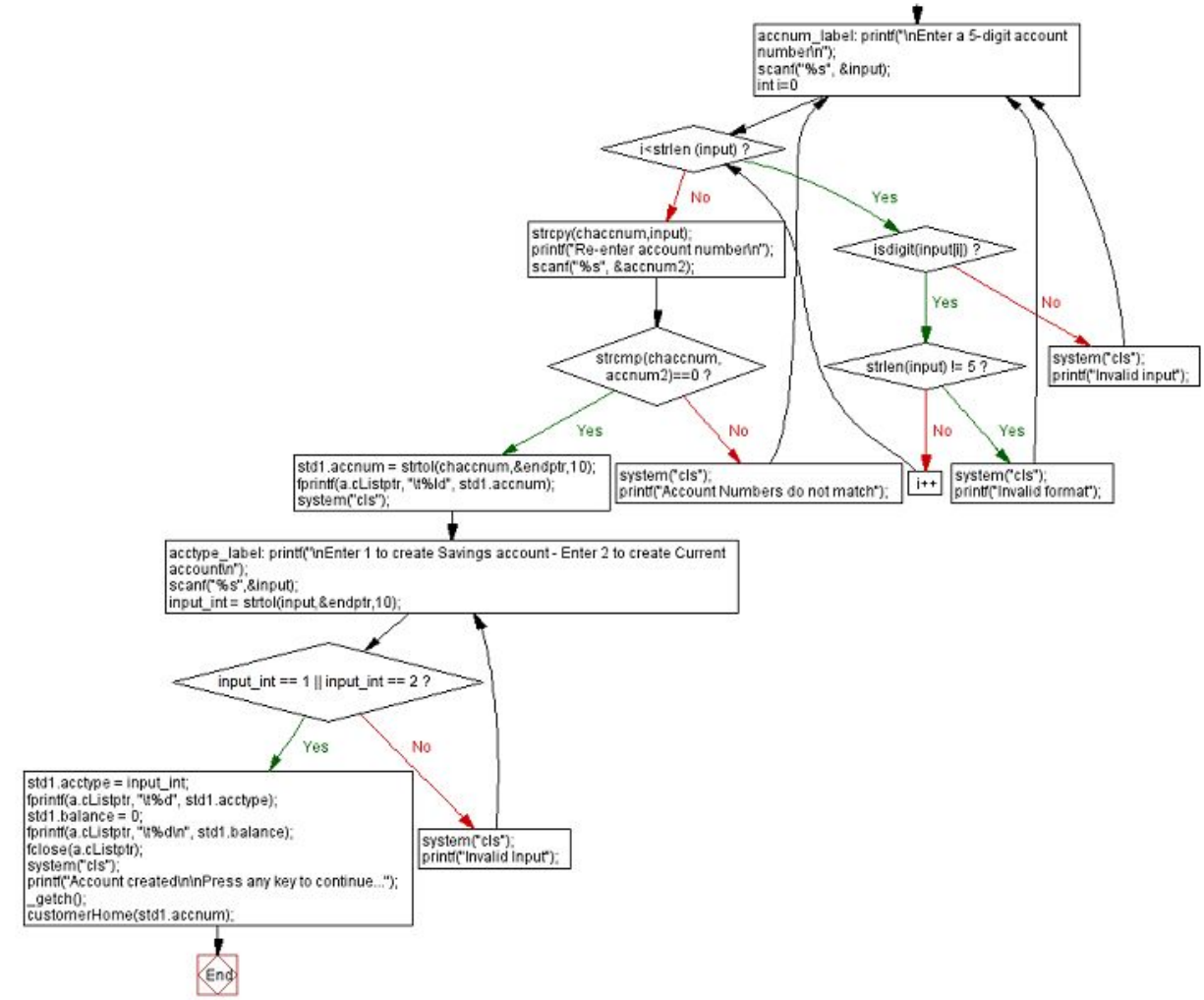

accnumExists ?
No
Yes
fscanf(a.cListptr, "%s%s%d%s%s%ld%d%d", &file.name, &file.mobnum, &file.sex, &file.username, &file.pass, &file.accnum, &file.acctype, &file.balance) != EOF ?
fscanf(a.cListptr, "%s%s%d%s%s%ld%d%d", &file.name, &file.mobnum, &file.sex, &file.username, &file.pass, &file.accnum, &file.acctype, &file.balance) != EOF ?
No
Yes
No
Yes
printf("\nAccount number %d does not exist\n\nPress any key to continue...", input_int);
fclose(a.cListptr);
fclose(tmp);
remove(cList);
rename("tmp_customerList.txt", cList);
_getch();
adminMenu();
fprintf(tmp, "%s\t%s\t%d\t%s\t%s\t%ld\t%d\t%d\n", file.name, file.mobnum, file.sex, file.username, file.pass, file.accnum, file.acctype, file.balance);
No
input_int != file.accnum ?
Yes
fprintf(tmp, "%s\t%s\t%d\t%s\t%s\t%ld\t%d\t%d\n", file.name, file.mobnum, file.sex, file.username, file.pass, file.accnum, file.acctype, file.balance);
fclose(a.cListptr);
fseek(tmp,0,SEEK_END);
int len = ftell(tmp);
fclose(tmp);
remove(cList);
len != 0 ?
Yes
No
rename("tmp_customerList.txt", cList);
remove("tmp_customerList.txt");
system("cls");
printf("\nAccount number %d has been removed successfully\n\nPress any key to continue...", input_int);
_getch();
adminMenu();
End

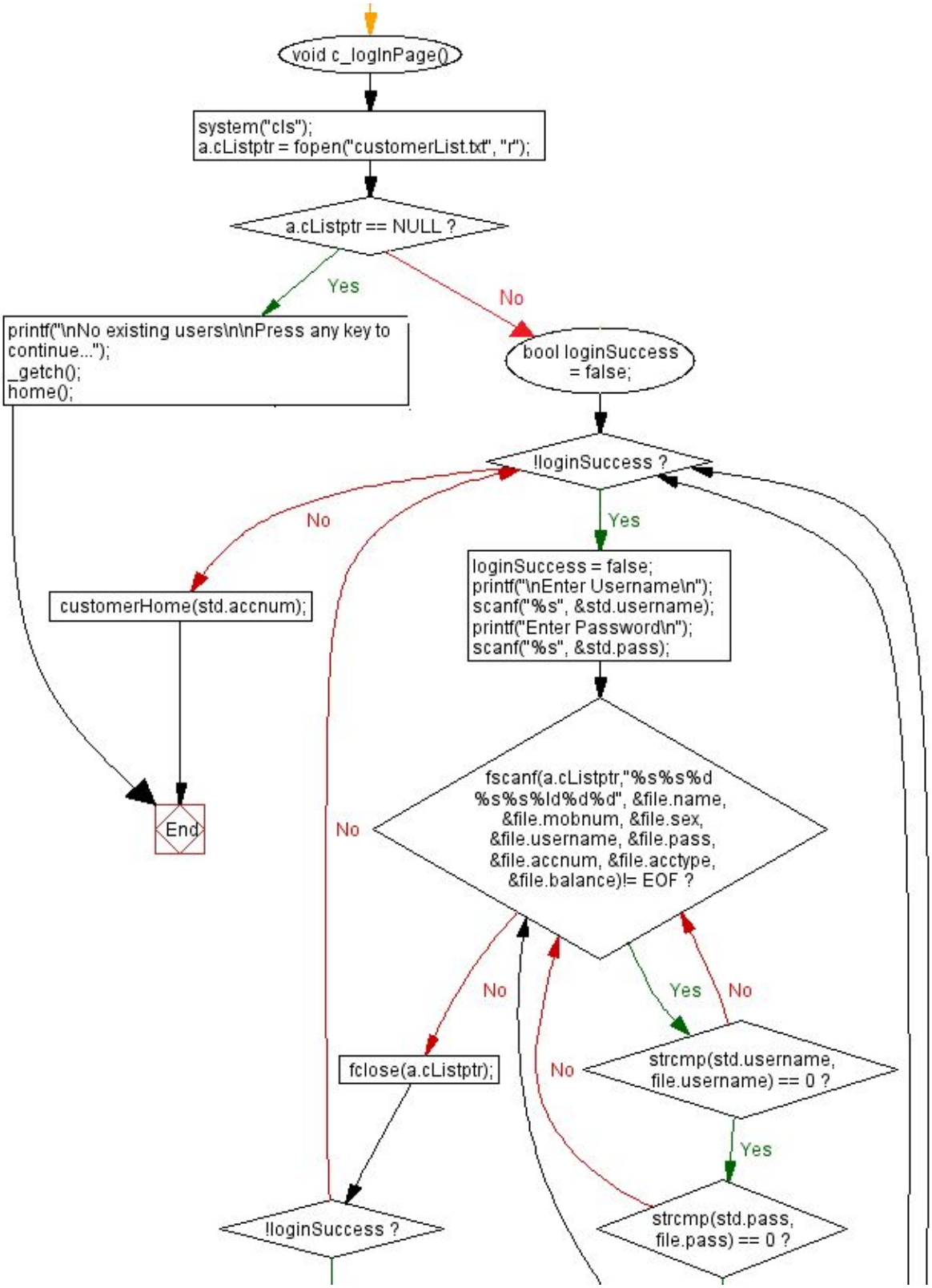

void a_transactionList()
system("cls");
int transaction;
bool trans_exists = false;
a.trans = fopen("transList.txt","r");
a.trans != NULL ?
Yes
No
printf("Account Number\tTransaction \n\n");
printf("No transactions available\n");
fscanf(a.trans,"%ld%d", &file.accnum,&transaction) != EOF ?
Yes
No
trans_exists = true;
!trans_exists ?
Yes
No
transaction > 0 ?
No
Yes
printf("No transactions available\n");
transaction < 0 ?
No
Yes
printf("%ld\t\tDeposit: %d MYR\n",file.accnum, transaction);
fclose(a.trans);
printf("\nPress any key to exit...");
_getch();
adminMenu();
Yes
printf("%ld\t\tWithdraw: %d MYR\n",file.accnum,abs (transaction));
End

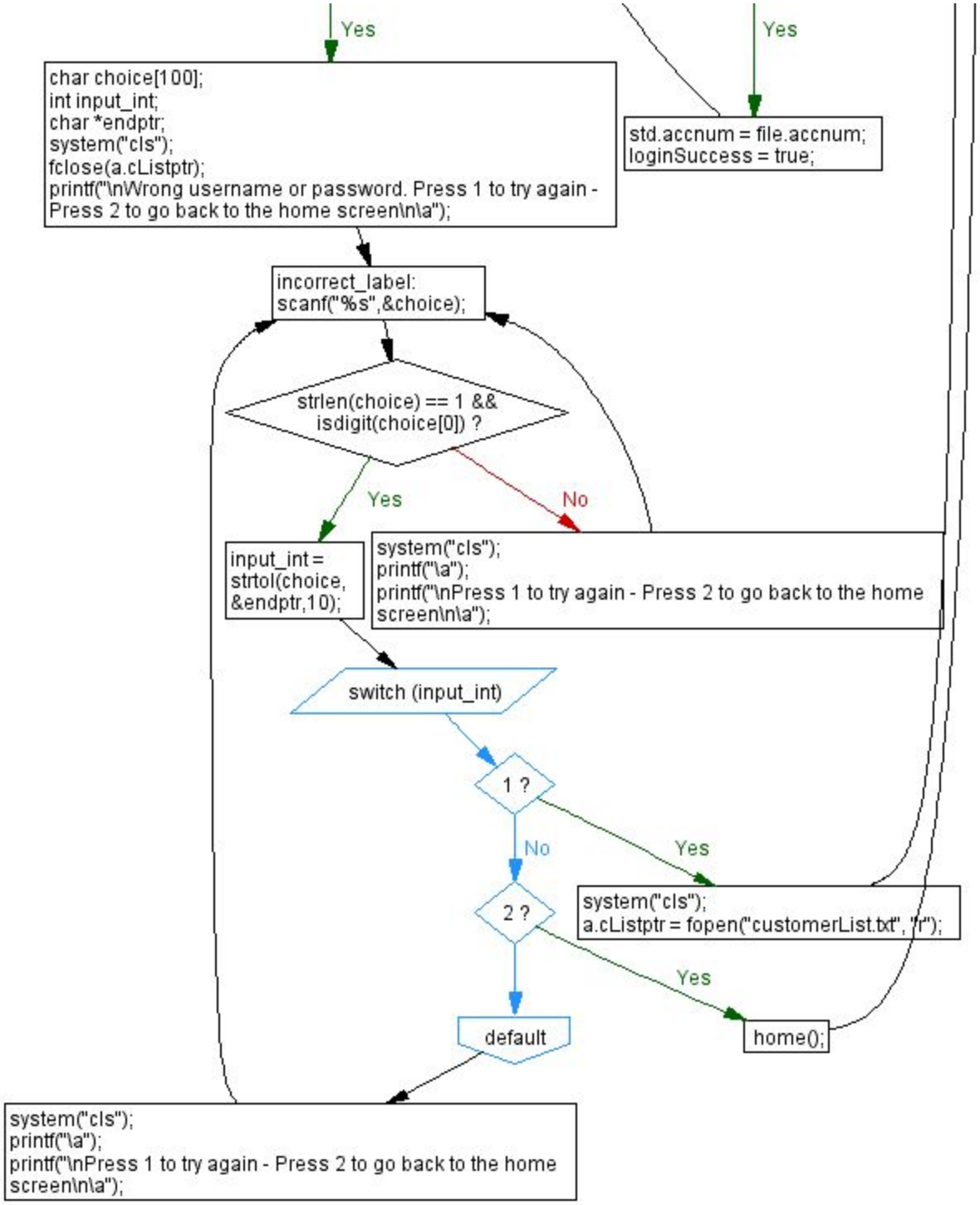

void customerMenu()
system("cls");
char choice[100];
int input_int;
char *endptr;
printf("\nPress 1 to log in\nPress 2 to register as a new user\nPress 0 to go back\n");
scanf("%s", &choice);
strlen(choice) == 1 && isdigit(choice[0]) ?
Yes
No
input_int = strtol(choice, &endptr,10);
printf("\a");
customerMenu();
switch (input_int)
1 ?
Yes
No
2 ?
Yes
No
0 ?
Yes
c_logInPage();
registration();
default
home();
printf("\a");
customerMenu();
End

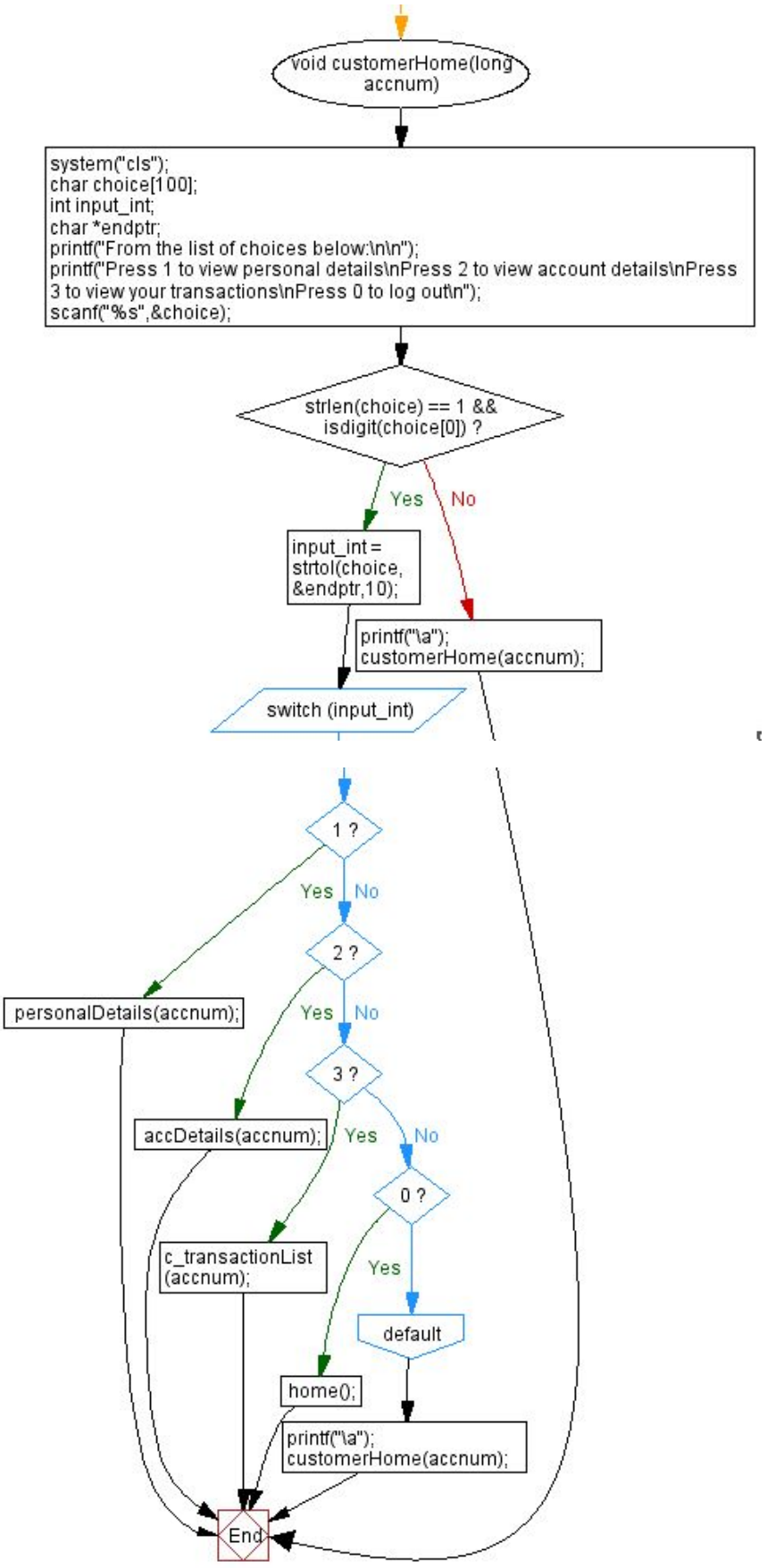

void registration()
system("cls");
char input[100];
int input_int;
char *endptr;
a.cListptr = fopen("customerList.txt",
"a+");
char name[35];
firstname_label: printf("\nEnter first name\n");
scanf("%s",name);
int i=0
i<strlen (name) ?
No
Yes
strcpy(std1.name,name);
fprintf(a.cListptr,"%s",std1.name);
char mobnum[50];
int mobcntr = 0;
system("cls");
printf("\nE");
!isalpha(name[i]) ?
No
Yes
i++
system("cls");
printf ("No numerals or special characters");
mobnum_label: printf("nter
your Malaysian mobile
number\n");
mymobnum_label: scanf("%s", &mobnum);
int i=0

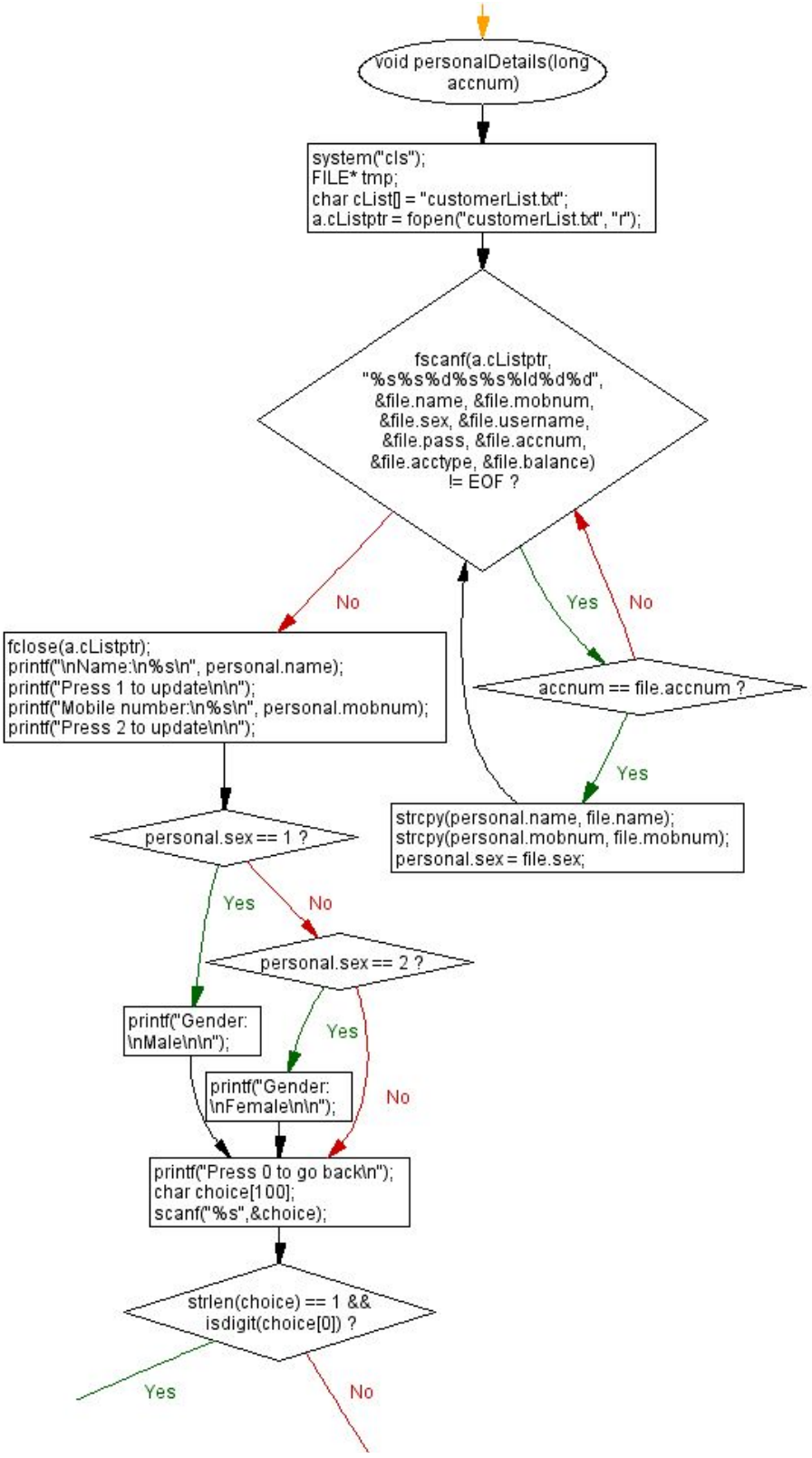

i<strlen (mobnum) ?
No
Yes
strcpy(std1.mobnum,mobnum);
fprintf(a.cListptr, "\t%s", std1.mobnum);
system("cls");
isdigit(mobnum[i]) ?
Yes
No
system("cls");
printf ("Invalid input\nPlease e");
sex_label: printf("\nEnter 1 if you are male - Enter 2 if you are female\n");
scanf("%s",&input);
int i=0
strlen(mobnum) != 10 ||
mobnum[0] != '0' || mobnum[1] != '1' ?
No
Yes
i++
mobcntr >= 2 ?
No
Yes
i<strlen(input) ?
No
Yes
system("cls");
mobcntr++;
printf ("Invalid format\nPlease enter a valid Malaysian number:\n");
system("cls");
printf ("\nHint: Your number should be 10 digits long and should begin with '01'\n");
mobcntr = 0;
input_int = strtol(input,&endptr,10);
isdigit(input[i]) ?
No
Yes
i++
system("cls");
printf("Invalid input");
input_int == 1 || input_int == 2 ?
Yes
No
std1.sex = input_int;
fprintf(a.cListptr, "\t%d", std1.sex);
system("cls");
system("cls");
printf("Invalid input");

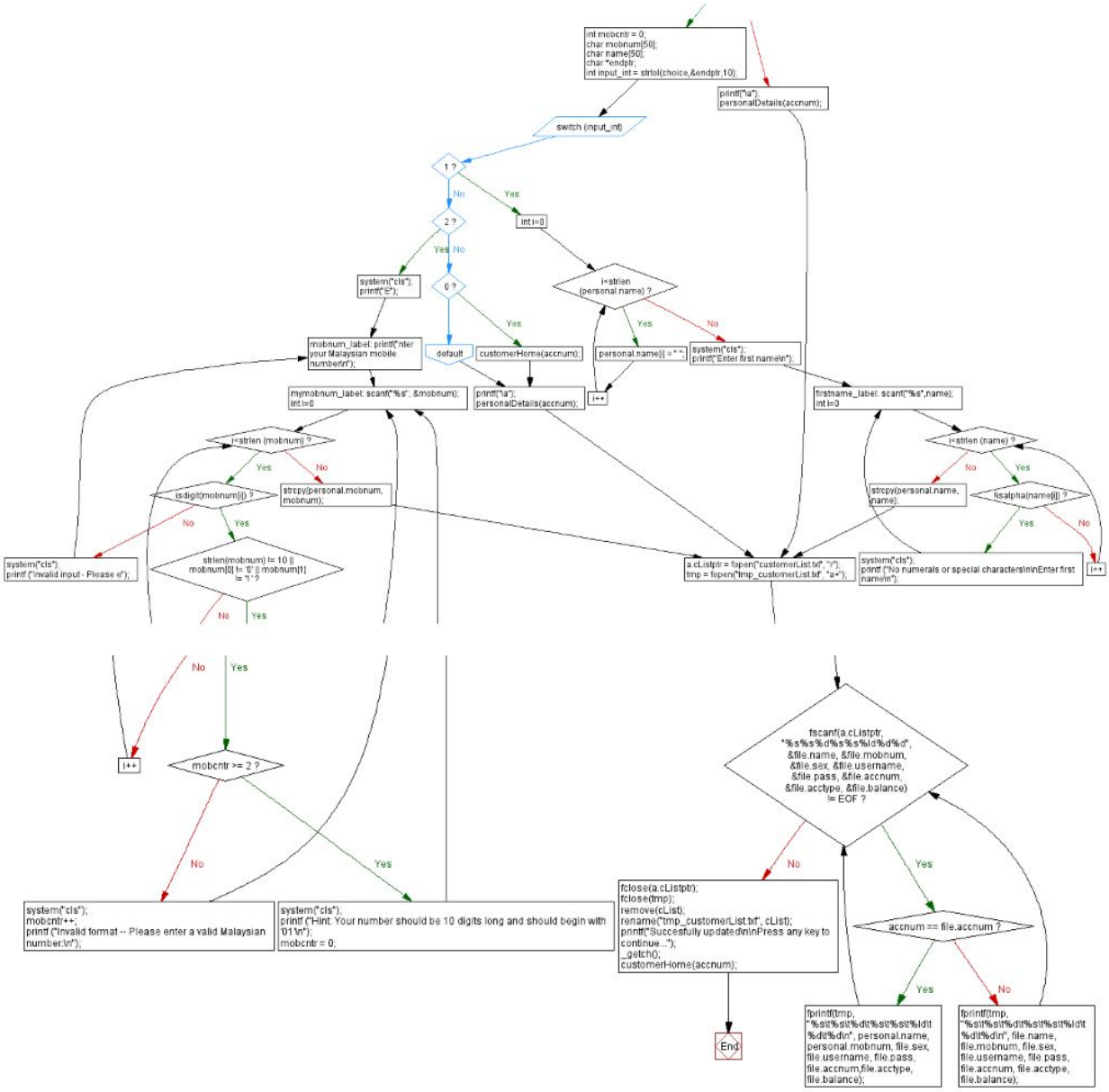

accnum_label: printf("\nEnter a 5-digit account number\n");
scanf("%s", &input);
int i=0
i<strlen (input) ?
No
Yes
strcpy(chaccnum,input);
printf("Re-enter account number\n");
scanf("%s", &accnum2);
isdigit(input[i]) ?
Yes
No
strcmp(chaccnum, accnum2)==0 ?
strlen(input) != 5 ?
system("cls");
printf("Invalid input");
Yes
No
No
Yes
std1.accnum = strtol(chaccnum,&endptr,10);
fprintf(a.cListptr, "\t%ld", std1.accnum);
system("cls");
system("cls");
printf("Account Numbers do not match");
i++
system("cls");
printf("Invalid format");
acctype_label: printf("\nEnter 1 to create Savings account - Enter 2 to create Current account\n");
scanf("%s",&input);
input_int = strtol(input,&endptr,10);
input_int == 1 || input_int == 2 ?
Yes
No
std1.acctype = input_int;
fprintf(a.cListptr, "\t%d", std1.acctype);
std1.balance = 0;
fprintf(a.cListptr, "\t%d\n", std1.balance);
fclose(a.cListptr);
system("cls");
printf("Account created\n\nPress any key to continue...");
_getch();
customerHome(std1.accnum);
system("cls");
printf("Invalid input");
End

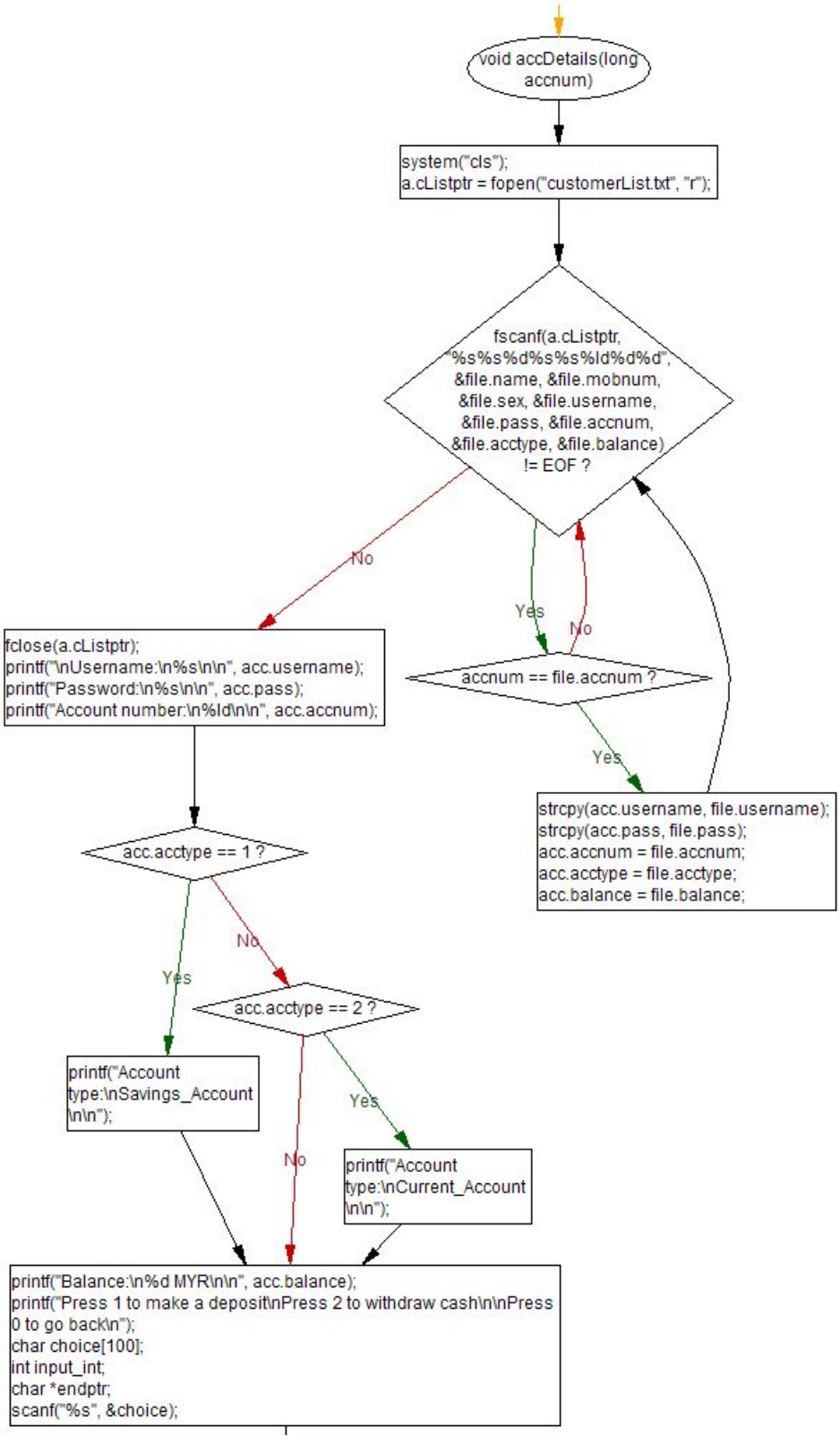

void c_logInPage()
system("cls");
a.cListptr = fopen("customerList.txt", "r");
a.cListptr == NULL ?
Yes
No
printf("\nNo existing users\n\nPress any key to continue...");
_getch();
home();
bool loginSuccess = false;
!loginSuccess ?
No
Yes
customerHome(std.accnum);
loginSuccess = false;
printf("\nEnter Username\n");
scanf("%s", &std.username);
printf("Enter Password\n");
scanf("%s", &std.pass);
End
fscanf(a.cListptr,"%s%s%d %s%s%ld%d%d", &file.name, &file.mobnum, &file.sex, &file.username, &file.pass, &file.accnum, &file.acctype, &file.balance)!= EOF ?
No
No
Yes
No
fclose(a.cListptr);
strcmp(std.username, file.username) == 0 ?
No
Yes
!loginSuccess ?
strcmp(std.pass, file.pass) == 0 ?

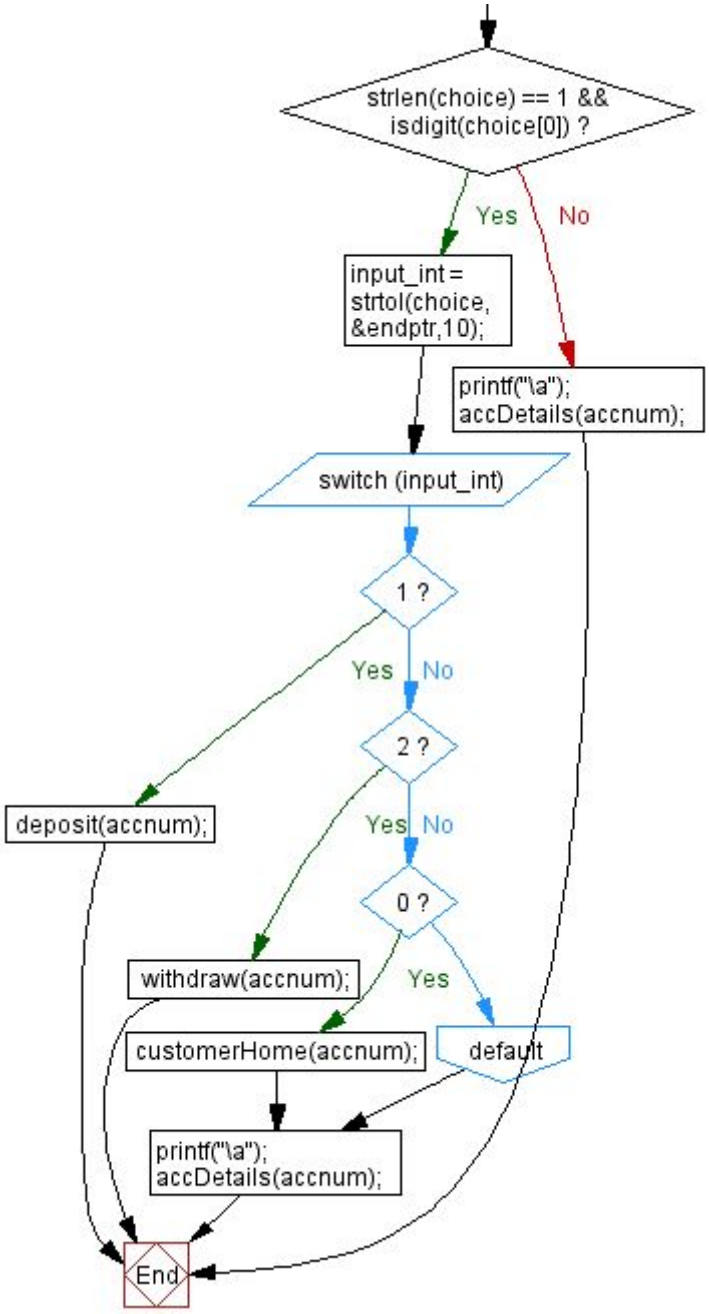

Yes
Yes
char choice[100];
int input_int;
char *endptr;
system("cls");
fclose(a.cListptr);
printf("\n\Wrong username or password. Press 1 to try again - Press 2 to go back to the home screen\n\a");
std.accnum = file.accnum;
loginSuccess = true;
incorrect_label:
scanf("%s",&choice);
strlen(choice) == 1 && isdigit(choice[0]) ?
Yes
No
input_int = strtol(choice, &endptr,10);
system("cls");
printf("\a");
printf("\nPress 1 to try again - Press 2 to go back to the home screen\n\a");
switch (input_int)
1 ?
No
Yes
2 ?
system("cls");
a.cListptr = fopen("customerList.txt", "r");
Yes
default
home();
system("cls");
printf("\a");
printf("\nPress 1 to try again - Press 2 to go back to the home screen\n\a");

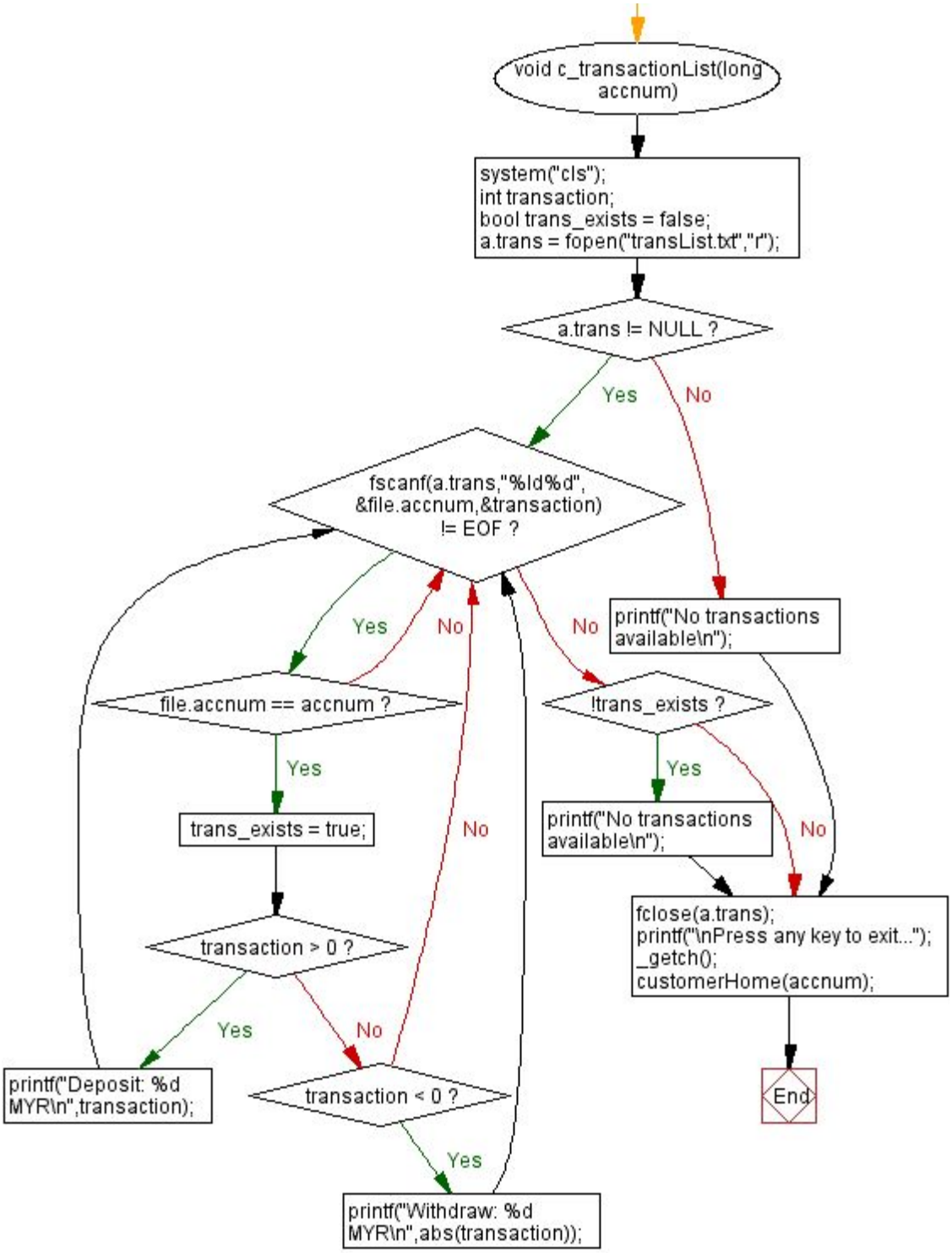

void customerHome(long accnum)
system("cls");
char choice[100];
int input_int;
char *endptr;
printf("From the list of choices below:\n\n");
printf("Press 1 to view personal details\nPress 2 to view account details\nPress 3 to view your transactions\nPress 0 to log out\n");
scanf("%s",&choice);
strlen(choice) == 1 && isdigit(choice[0]) ?
Yes
No
input_int = strtol(choice, &endptr,10);
printf("\a");
customerHome(accnum);
switch (input_int)
1 ?
Yes
No
2 ?
Yes
No
3 ?
Yes
No
0 ?
Yes
default
personalDetails(accnum);
accDetails(accnum);
c_transactionList (accnum);
home();
printf("\a");
customerHome(accnum);
End

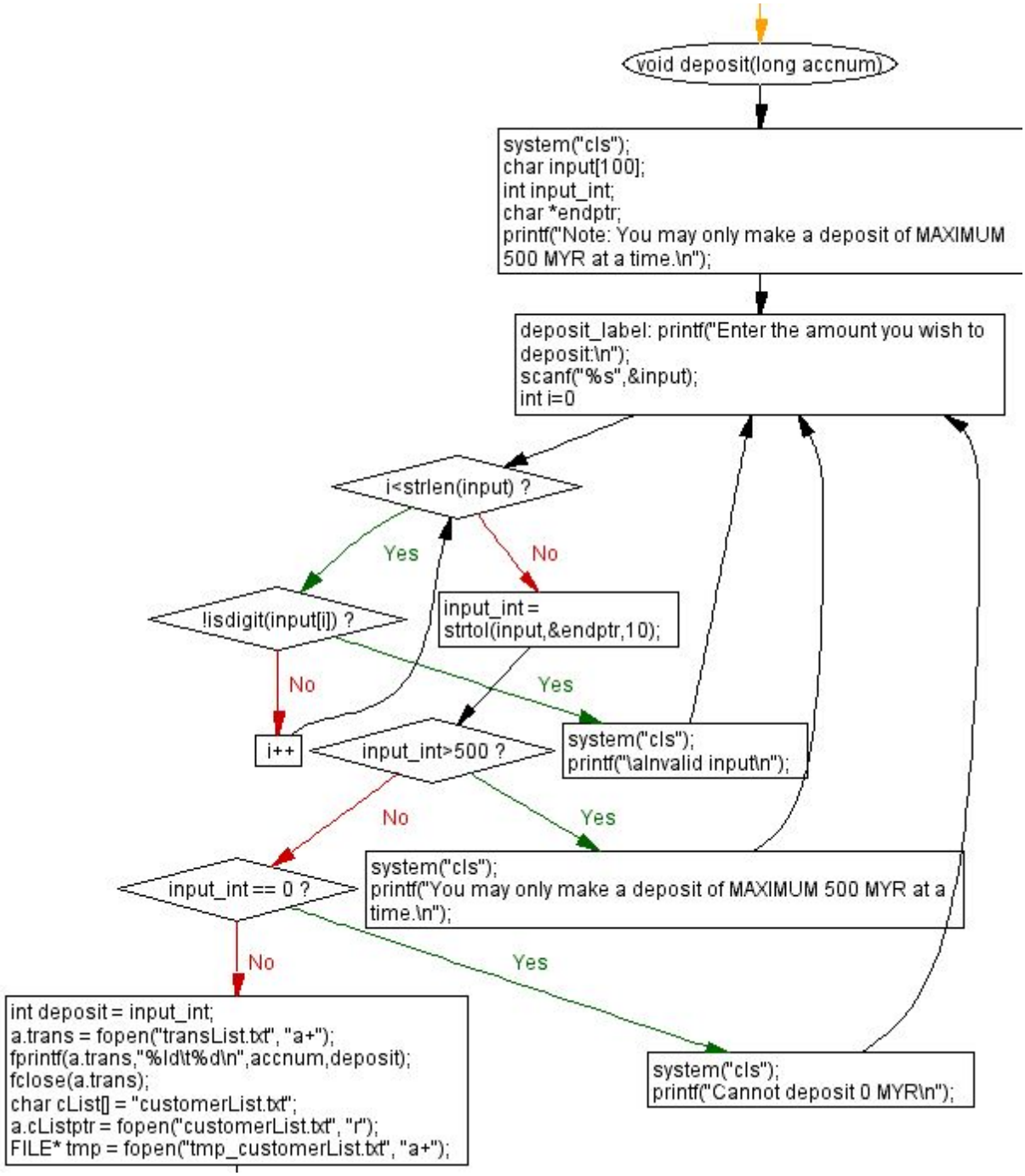

void personalDetails(long accnum)
system("cls");
FILE* tmp;
char cList[] = "customerList.txt";
a.cListptr = fopen("customerList.txt", "r");
fscanf(a.cListptr, "%s%s%d%s%s%ld%d%d", &file.name, &file.mobnum, &file.sex, &file.username, &file.pass, &file.accnum, &file.acctype, &file.balance) != EOF ?
No
Yes
No
fclose(a.cListptr);
printf("\nName:\n%s\n", personal.name);
printf("Press 1 to update\n\n");
printf("Mobile number:\n%s\n", personal.mobnum);
printf("Press 2 to update\n\n");
accnum == file.accnum ?
Yes
strcpy(personal.name, file.name);
strcpy(personal.mobnum, file.mobnum);
personal.sex = file.sex;
personal.sex == 1 ?
Yes
No
personal.sex == 2 ?
printf("Gender:\nMale\n\n");
Yes
printf("Gender:\nFemale\n\n");
No
printf("Press 0 to go back\n");
char choice[100];
scanf("%s",&choice);
strlen(choice) == 1 && isdigit(choice[0]) ?
Yes
No

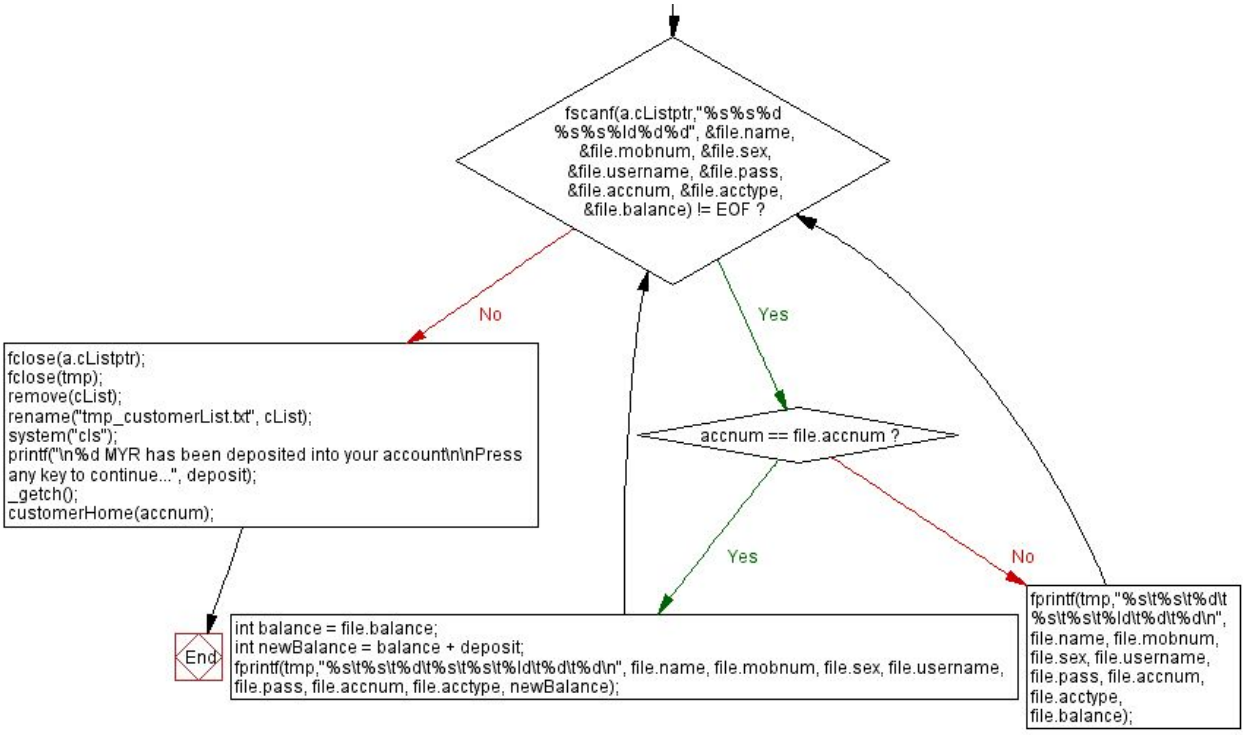

int mobcntr = 0;
char mobnum[50];
char name[50];
char *endptr;
int input_int = strtol(choice,&endptr,10);
printf("\a");
personalDetails(accnum);
switch (input_int)
1 ?
No
Yes
int i=0
2 ?
Yes
No
system("cls");
printf("\t");
0 ?
i<strlen (personal.name) ?
Yes
No
mobnum_label: printf("nter your Malaysian mobile number\n");
default
customerHome(accnum);
personal.name[i] = " ";
system("cls");
printf("Enter first name\n");
mymobnum_label: scanf("%s", &mobnum);
int i=0
printf("\a");
personalDetails(accnum);
i++
firstname_label: scanf("%s",name);
int i=0
i<strlen (mobnum) ?
i<strlen (name) ?
isdigit(mobnum[i]) ?
strcpy(personal.mobnum, mobnum);
strcpy(personal.name, name);
isalpha(name[i]) ?
system("cls");
printf ("Invalid input - Please e");
strlen(mobnum) != 10 || mobnum[0] != '0' || mobnum[1] != '1' ?
a.cListptr = fopen("customerList.txt", "r");
tmp = fopen("tmp_customerList.txt", "a+");
system("cls");
printf ("No numerals or special characters\n\nEnter first name\n");
i++
No
Yes
i++
mobcntr >= 2 ?
system("cls");
mobcntr++;
printf ("\nInvalid format -- Please enter a valid Malaysian number:\n");
system("cls");
printf ("\nHint: Your number should be 10 digits long and should begin with '01'\n");
mobcntr = 0;
fscanf(a.cListptr, "%s%s%d%s%s%ld%d%d", &file.name, &file.mobnum, &file.sex, &file.username, &file.pass, &file.accnum, &file.acctype, &file.balance) != EOF ?
fclose(a.cListptr);
fclose(tmp);
remove(cList);
rename("tmp_customerList.txt", cList);
printf("Successfully updated\n\nPress any key to continue...");
_getch();
customerHome(accnum);
accnum == file.accnum ?
fprintf(tmp, "%s\t%s\t%d\t%s\t%s\t%ld\t%d\t%d\n", personal.name, personal.mobnum, file.sex, file.username, file.pass, file.accnum,file.acctype, file.balance);
fprintf(tmp, "%s\t%s\t%d\t%s\t%s\t%ld\t%d\t%d\n", file.name, file.mobnum, file.sex, file.username, file.pass, file.accnum, file.acctype, file.balance);
End

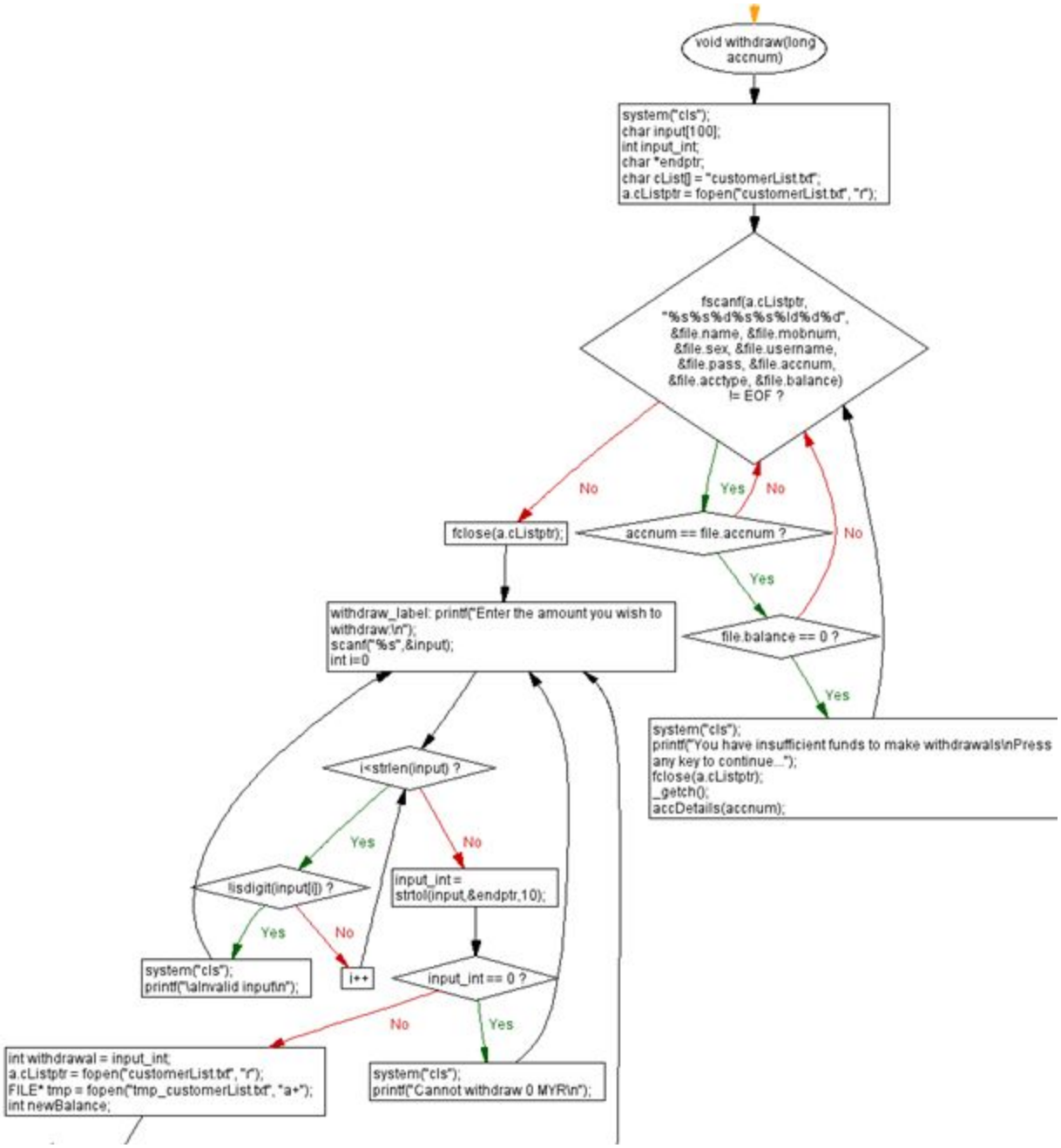

void accDetails(long accnum)
system("cls");
a.cListptr = fopen("customerList.txt", "r");
fscanf(a.cListptr, %s%s%d%s%s%ld%d%d", &file.name, &file.mobnum, &file.sex, &file.username, &file.pass, &file.accnum, &file.acctype, &file.balance) != EOF ?
No
Yes
No
accnum == file.accnum ?
Yes
strcpy(acc.username, file.username);
strcpy(acc.pass, file.pass);
acc.accnum = file.accnum;
acc.acctype = file.acctype;
acc.balance = file.balance;
fclose(a.cListptr);
printf("\nUsername:\n%s\n\n", acc.username);
printf("Password:\n%s\n\n", acc.pass);
printf("Account number:\n%ld\n\n", acc.accnum);
acc.acctype == 1 ?
Yes
No
acc.acctype == 2 ?
Yes
No
printf("Account type:\nSavings_Account \n\n");
printf("Account type:\nCurrent_Account \n\n");
printf("Balance:\n%d MYR\n\n", acc.balance);
printf("Press 1 to make a deposit\nPress 2 to withdraw cash\n\nPress 0 to go back\n");
char choice[100];
int input_int;
char *endptr;
scanf("%s", &choice);

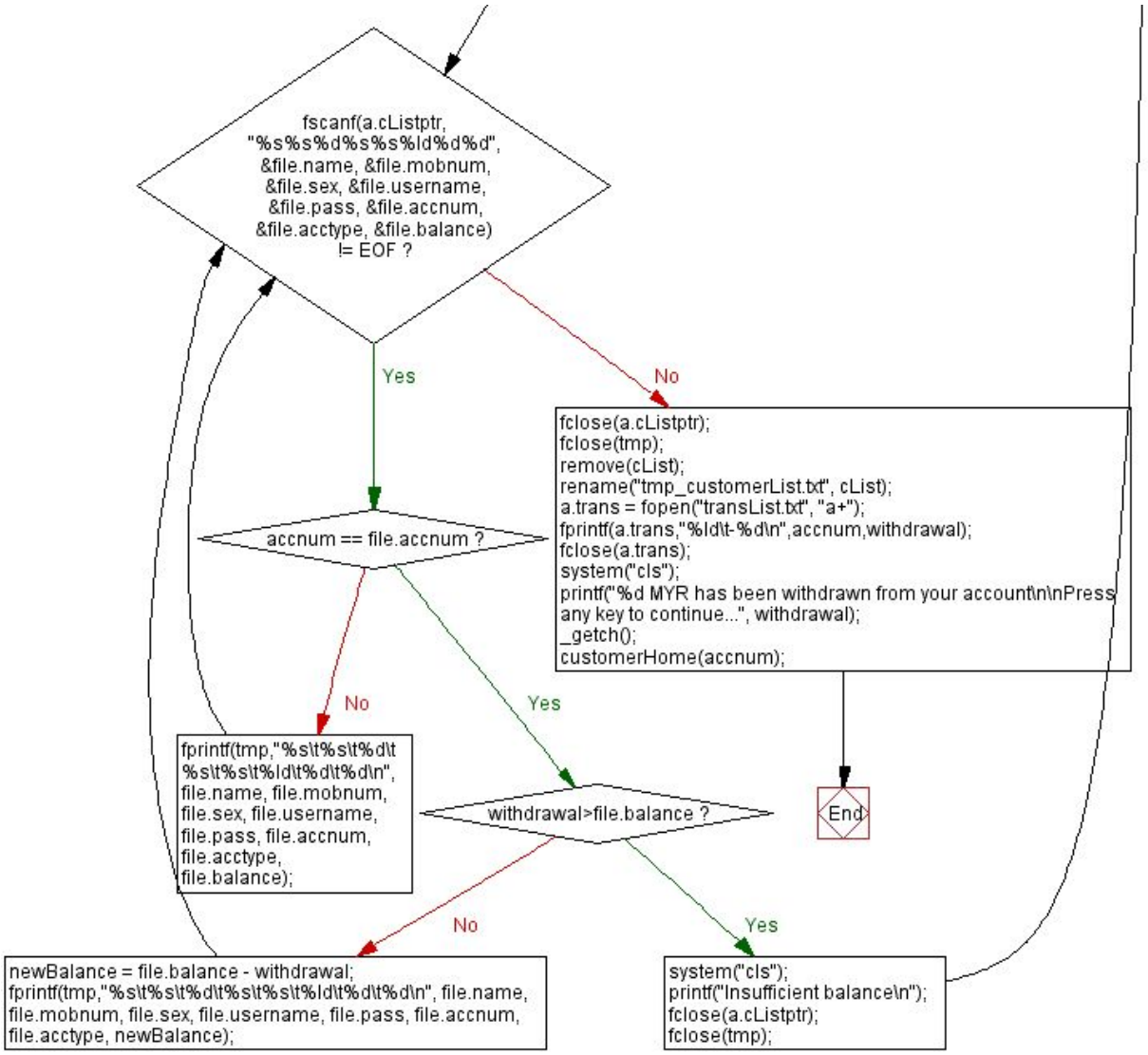

strlen(choice) == 1 &&
isdigit(choice[0]) ?
Yes
No
input_int =
strtol(choice,
&endptr,10);
printf("\a");
accDetails(accnum);
switch (input_int)
1 ?
Yes
No
2 ?
deposit(accnum);
Yes
No
0 ?
withdraw(accnum);
Yes
customerHome(accnum);
default
printf("\a");
accDetails(accnum);
End


void c_transactionList(long accnum)
system("cls");
int transaction;
bool trans_exists = false;
a.trans = fopen("transList.txt","r");
a.trans != NULL ?
Yes
No
fscanf(a.trans,"%ld%d", &file.accnum,&transaction) != EOF ?
Yes
No
No
printf("No transactions available\n");
file.accnum == accnum ?
!trans_exists ?
Yes
Yes
trans_exists = true;
printf("No transactions available\n");
No
No
fclose(a.trans);
printf("\nPress any key to exit...");
_getch();
customerHome(accnum);
transaction > 0 ?
Yes
No
End
printf("Deposit: %d MYR\n",transaction);
transaction < 0 ?
Yes
printf("Withdraw: %d MYR\n",abs(transaction));


void deposit(long accnum)
system("cls");
char input[100];
int input_int;
char *endptr;
printf("Note: You may only make a deposit of MAXIMUM 500 MYR at a time.\n");
deposit_label: printf("Enter the amount you wish to deposit:\n");
scanf("%s",&input);
int i=0
i<strlen(input) ?
Yes
No
isdigit(input[i]) ?
input_int = strtol(input,&endptr,10);
No
Yes
i++
input_int>500 ?
system("cls");
printf("\aInvalid input\n");
No
Yes
input_int == 0 ?
system("cls");
printf("You may only make a deposit of MAXIMUM 500 MYR at a time.\n");
No
Yes
int deposit = input_int;
a.trans = fopen("transList.txt", "a+");
fprintf(a.trans,"%Id\t%d\n",accnum,deposit);
fclose(a.trans);
char cList[] = "customerList.txt";
a.cListptr = fopen("customerList.txt", "r");
FILE* tmp = fopen("tmp_customerList.txt", "a+");
system("cls");
printf("Cannot deposit 0 MYR\n");


fscanf(a.cListptr,"%s%s%d
%s%s%ld%d%d", &file.name,
&file.mobnum, &file.sex,
&file.username, &file.pass,
&file.accnum, &file.acctype,
&file.balance) != EOF ?
No
Yes
fclose(a.cListptr);
fclose(tmp);
remove(cList);
rename("tmp_customerList.txt", cList);
system("cls");
printf("\n%d MYR has been deposited into your account\n\nPress
any key to continue...", deposit);
_getch();
customerHome(accnum);
accnum == file.accnum ?
Yes
No
End
int balance = file.balance;
int newBalance = balance + deposit;
fprintf(tmp,"%s\t%s\t%d\t%s\t%s\t%ld\t%d\t%d\n", file.name, file.mobnum, file.sex, file.username,
file.pass, file.accnum, file.acctype, newBalance);
fprintf(tmp,"%s\t%s\t%d\t
%s\t%s\t%ld\t%d\t%d\n",
file.name, file.mobnum,
file.sex, file.username,
file.pass, file.accnum,
file.acctype,
file.balance);


void withdraw(long accnum)
system("cls");
char input[100];
int input_int;
char *endptr;
char cList[] = "customerList.txt";
a.cListptr = fopen("customerList.txt", "r");
fscanf(a.cListptr, "%s%s%d%s%s%ld%d%d", &file.name, &file.mobnum, &file.sex, &file.username, &file.pass, &file.accnum, &file.acctype, &file.balance) != EOF ?
No
Yes
No
fclose(a.cListptr);
accnum == file.accnum ?
No
Yes
file.balance == 0 ?
Yes
withdraw_label: printf("Enter the amount you wish to withdraw:\n");
scanf("%s",&input);
int i=0
system("cls");
printf("You have insufficient funds to make withdrawals\nPress any key to continue...");
fclose(a.cListptr);
_getch();
accDetails(accnum);
i<strlen(input) ?
Yes
No
!isdigit(input[i]) ?
input_int = strtol(input,&endptr,10);
Yes
No
system("cls");
printf("\aInvalid input\n");
i++
input_int == 0 ?
No
Yes
int withdrawal = input_int;
a.cListptr = fopen("customerList.txt", "r");
FILE* tmp = fopen("tmp_customerList.txt", "a+");
int newBalance;
system("cls");
printf("Cannot withdraw 0 MYR\n");


fscanf(a.cListptr, "%s%s%d%s%s%ld%d%d", &file.name, &file.mobnum, &file.sex, &file.username, &file.pass, &file.accnum, &file.acctype, &file.balance) != EOF ?
Yes
No
fclose(a.cListptr);
fclose(tmp);
remove(cList);
rename("tmp_customerList.txt", cList);
a.trans = fopen("transList.txt", "a+");
fprintf(a.trans,"%ld\t-%d\n",accnum,withdrawal);
fclose(a.trans);
system("cls");
printf("%d MYR has been withdrawn from your account\n\nPress any key to continue...", withdrawal);
_getch();
customerHome(accnum);
accnum == file.accnum ?
No
Yes
fprintf(tmp,"%s\t%s\t%d\t %s\t%s\t%ld\t%d\t%d\n", file.name, file.mobnum, file.sex, file.username, file.pass, file.accnum, file.acctype, file.balance);
withdrawal>file.balance ?
End
No
Yes
newBalance = file.balance - withdrawal;
fprintf(tmp,"%s\t%s\t%d\t%s\t%s\t%ld\t%d\t%d\n", file.name, file.mobnum, file.sex, file.username, file.pass, file.accnum, file.acctype, newBalance);
system("cls");
printf("Insufficient balance\n");
fclose(a.cListptr);
fclose(tmp);

Pseudocode

```
--------------------
//Start execution from here
----------------------Define method "Home()"
//clear screen by "system("cls")";
//print "
      Press 1 to log in as Admin
      Press 2 if you are a customer
      Press 0 to to exit program
      ";

//take string Input to "choice";

//if "choice" lenght=1 and "choice[0]" is digit,exp:isdigit(choice[0])
      //set value to "_int" from function:"strtol(choice,&endptr,10);"

   //apply switch case on "input_int"

         //when case value 1 then
          call method "logInPage()";

         //when case  2 then
          call method "customerMenu()";

       //when case  value 0 then
          call method "exit(0)";

       //default
          call method "home()";

else
      call method "home()";

----------------------End method "Home()"


----------------------Define method "a_logInPage()"
```

```
set loginSuccess = false;

//while login not success >system will try to login by bellow information.
    set username[] = "farhan";
    set password[] = "pass";
    set loginSuccess = false;

 //clear screen;
    // print "
                        Enter Username (Hint: 'farhan')
                        ";

    // take string input and set value to >"admin.username";

    //print "Enter Password (Hint: 'pass')";
    // get input for password  & set value to >admin.password;


    //if string is match successfully by function:strcmp(admin.username , username) then

            // if string is match with "admin.password" and "password" by function
>strcmp(admin.password , password) then
                    set loginSuccess = true;

    // if "loginSuccess" is false then
    //clear screen;
    //Defining a <Label>"incorrect_label:"
    //"notification sound
    //Print "
            Wrong username or password.
            Press 1 to try again
            Press 2 to go back to the home screen
            ";

       //get numeric input and set to>"choice"


    //apply switch case on "choice"
       //when case  1 then no activity
```

```
            //when case  2 then
            call method "home()";

            // default
            Screen clear
            jump to level "incorrect_label section";

//call method "adminMenu()"

----------End method



----------Define method "adminMenu()"
//Clear screen
      a.cListptr = fopen("customerList.txt", "r+");
//read data from file "customerList.txt" and set to >"a.cListptr";
      if (a.cListptr == NULL) {
       printf("\nNo existing users\n\nPress any key to log out...");

//if "a.cListptr" is null then
   //print "existing users

         Press any key to log out...";

      //call method "home()";

//else

         //reads  data from "a.cListptr" and set to (file.name, file.mobnum, file.sex, file.username,
file.pass, file.accnum, file.acctype, file.balance);

                  //if "file.sex" == 1 and "file.acctype" == 1 then
                        //print "
                              Name:<file.name>  Mobile Number:<file.mobnum>    Gender:
Male   Username:<file.username>    Password:<file.pass>
                              Account Number:<file.accnum>  Account Type: Savings Account
Balance:<file.balance> MYR
```

```
                //else if "file.acctype" equal 2 then
                        //print "
                                        Name:<file.name>  Mobile Number:<file.mobnum>
Gender: Male            Username:<file.username>    Password:<file.pass>
                                        Account Number:<file.accnum>  Account Type: Current
Account  Balance:<file.balance> MYR
                                        ";

                //else if "file.sex" equal 2 and "file.acctype" equal 1
                                //print "
                                        Name:<file.name>  Mobile Number: <file.mobnum>
Gender:         Female  Username:<file.username>  Password:<file.pass>
                                        Account Number:<file.accnum>  Account Type: Savings
Account         Balance:<file.balance> MYR
                                        ";

                //else if "file.acctype" equal 2 then
                        //print "
                                        Name:<file.name>  Mobile Number:<file.mobnum>
Gender:Female  Username:<file.username>  Password:<file.pass>
                                        Account Number:<file.accnum>  Account Type: Current
Account Balance:<file.balance> MYR

                                        ";

        //close stream "fclose(a.cListptr)";


        //print "
            Press 1 to view system transaction history
         Press 2 to remove a user
         Press 0 to to log out
            ";


        //take string input and set to>"choice";

        //if "choice" lenght=1 and "choice[0]" is disit
```

```
        input_int = strtol(choice,&endptr,10);
            // set value to "_int" from  function:"strtol(choice,&endptr,10)";

            //apply switch case on variable>_int:
                    //when case:1 then call method>a_transactionList();
                    //when case:2 then call method>removeUser();
                    //when case:2 then call method>home();
                    //default call method>adminMenu();
    //Else
                    //call method>adminMenu();

----------End method;

----------Define method "removeUser()"
    //Clear Screen
    //set "customerList.txt" to>cList[]

        //Defining a <Label>"accnum_label:"


    print "
                    Please enter the account number of the account you wish to remove:
                    ";

    //take string input and set to >"input";
    //For loop counter 0 to lenth-1 of <strlen(input)
        //if input[i] is not digit  check by function< "isdigit(input[i])" then
            //Print "Invalid input"
            //jump to <Label>:"accnum_label";

    input_int = strtol(input,&endptr,10);
    //take integer input & set to "_int" from function<"strtol(input,&endptr,10)";
    //set accnumExists = false

    //Read data from file:"customerList.txt" and set to>"a.cListptr";

     //Apend data to file:"tmp_customerList.txt" by function>fopen("tmp_customerList.txt", "a+")
(N.B:option for file create if file does not exist)
```

```
        //reads  data from "a.cListptr" and set to (file.name, file.mobnum, file.sex, file.username,
file.pass, file.accnum, file.acctype, file.balance);
            if(input_int == file.accnum){
            //if "input_int" equal "file.accnum" then
                    //set "accnumExists"=true;


      //close stream "fclose(a.cListptr)";

      //Read data from file:"customerList.txt" and set to>"a.cListptr";

      //if accnumExists is true then
          //reads  data from "a.cListptr" and set to (file.name, file.mobnum, file.sex, file.username,
file.pass, file.accnum, file.acctype, file.balance);
              //if input_int Not Equal file.accnum then
                            //sends formatted output(file.name, file.mobnum, file.sex, file.username,
file.pass, file.accnum,file.acctype, file.balance) to a stream>"tmp"
      //else
                    //reads  data from "a.cListptr" and set to (file.name, file.mobnum, file.sex,
file.username, file.pass, file.accnum, file.acctype, file.balance);
              //sends formatted output(file.name, file.mobnum, file.sex, file.username, file.pass,
file.accnum,file.acctype, file.balance) to a stream>"tmp"

      //print "
                    Account number <input_int> does not exist
                    Press any key to continue...";


      //close stream  "fclose(a.cListptr)";

      //close FILE stream "fclose(tmp)";

      //remove "remove(cList)";


      //rename file name "tmp_customerList.txt" to <cList>


      //Call method "adminMenu()";
```

```
      //close stream "fclose(a.cListptr)";

      //check stream "fseek(tmp,0,SEEK_END)";

      //set length of streama "ftell(tmp)" and set to >"len";

      //close stream "fclose(tmp)";
   //remove stream "remove(cList)";


      //if "len" not equal 0 then
       //rename file name "tmp_customerList.txt" to <cList>;
   //else
      //remove stream "remove(cList)";

  //Clear Screen;
  //print "
         Account number <input_int> has been removed successfully

         Press any key to continue...";

   //Call method "adminMenu()";
----------End method;

----------Define method "a_transactionList()"
   //Clear Screen
   // set "trans_exists" = false;
   //Read data from file:"transList.txt" and set to>a.trans

   //if a.trans not equal null then
      //print "Account Number   Transaction

                           ";

      //Take User input for (file.name,transaction)

               //set trans_exists = true;
```

```
            //if "transaction" greater than 0
               //print "<file.accnum>      Deposit:<transaction> MYR
                                            ";

                     //else if "transaction" equal 0 then
                  //print "<file.accnum>      tWithdraw:<transaction> MYR
                        ";

        //if "trans_exists" is false then
            //print "No transactions available
                        ";

        //else
            //print "No transactions available
                  ";


    //Close stream "fclose(a.trans)";

    //print   "
            Press any key to exit..."

    //call method "adminMenu()";
-----------------------End method;

-----------------------Define method "customerMenu()"
     //Clear Screen;
    //printf "
            Press 1 to log in
            Press 2 to register as a new user
            Press 0 to go back
            ";

    //take string input to>"choice""

    //if length of "choice" equal 1 && "choice[0]" is digit then
         //take input to>"_int" from function"strtol(choice,&endptr,10)";
        //Apply switch case on "input_int""
```

```
        //when case 1 then  call method>c_logInPage()
        //when case 2 then  call method>registration()
        //when case 0 then  call method>home()
        //default call method>customerMenu()

    // Else
        //call method "customerMenu()" ;

----------End method;

----------Define method "registration()"
    //Clear Screen

    //Read data from file:"customerList.txt" and set to>"a.trans";

    //print "
        Enter first name
        ";
    //take string input and set to>"name";


    // for loop intial counter i=o to lenght of "name"<strlen(name)
        //if not equal !isalpha(name[i])
        // clear screen
        //print "No numerals or special characters"
        //jump to <label> "firstname_label";


    //copy "name">"std1.name";

    //sends formatted output "std1.name" to a stream>"a.cListptr".;

    //set "mobcntr" = 0;

    //Clear Screen

      mobnum_label:
      printf("Enter your Malaysian mobile number\n");
    //print "Enter your Malaysian mobile number "
```

```
      //Define a <Label>"mymobnum_label:"

   //take string input to>"mobnum";


   // for loop intial counter i=o to lenght of "mobnum"<strlen(mobnum)
      //if  "mobnum[i]" is digit ,Exp:isdigit(mobnum[i])
                  //if "mobnum" length not equal 10 or "mobnum[0]" not equal '0' or mobnum[1]
not equal '1' then
             //if "mobcntr" grater or equal 2 than
                //clear screen;
                //print "
                           Hint: Your number should be 10 digits long and should begin with '01'
                      ";
                //set mobcntr = 0;
                goto mymobnum_label;
                //jump to <lebel> "mymobnum_label";


        //clear screen;
        //increae counter "mobcntr",exp:mobcntr++;
        //print "Invalid format
               Please enter a valid Malaysian number:";

        //jump to <lebel> "mymobnum_label";

     //else
       //clear screen

       //print "Invalid input
                Please e";
      //jump to <lebel> "mobnum_label";


   //copy "mobnum" to "std1.mobnum",exp:strcpy(std1.mobnum,mobnum);
   //sends formatted output "std1.mobnum" to a stream>"a.cListptr";
   //clear screen;
      //Define a <Label>"sex_label:"
```

```
//print "
        Enter 1 if you are male - Enter 2 if you are female
        ";

//take string input to>"input"


// for loop intial counter i=o to lenght of "input"<strlen(input)
     //if  "input[i]" is digit ,Exp:isdigit(input[i])
        //clear screen;
       //print "Invalid Input";
      //jump to <lebel> "sex_label",Exp:goto sex_label;


//take input to "input_int" from function:"strtol",exp:strtol(input,&endptr,10);

//if "input_int" is 1 or 2 then
   //set  "input_int" to "std1.sex";
//else
   //clear screen;
   //Print "Invalid Input";
   //jump to <lebel> "sex_label",Exp:goto sex_label;


//sends formatted output "sstd1.sex" to a stream>"a.cListptr";

//clear screen;

   //Define a <Label>"uname_label1:"
//Print "
        Enter username
        ";

   //Define a <Label>"uname_label2:"
//take string input to>"input";
// for loop intial counter i=o to lenght of "input"<strlen(input)
    //if  "input[i]" is string or digit then,Exp:isalpha(input[i]) || isdigit(input[i])
        //if "input" length less than 5 or greter than 20 then
            //Clear Screen;
```

```
        //Print "Username must be 5 to 20 characters long
              Try again:
              ";
        //jump to <lebel> "uname_label2";

     //else
                        //Clear Screen;
                        //Print "No special characters"
                        //jump to <lebel> "uname_label1";


//Copy data from "input" to>"std1.username";
//sends formatted output "std1.username" to a stream>"a.cListptr";
 //Clear Screen;

   //Define a <Label>"password_label:"

//print "
       Enter password
       ";

//take string input for Password to>"input";

//Print "Re-enter password
       ";
//take string input for Re-enter password to>"pass2";

//if string is match with "input" and  "pass2"  then
   //Copy data from "input" to>"std1.pass";
   //sends formatted output "std1.pass" to a stream>"a.cListptr";

//else
   //Clear Screen;
   //print "Passwords do not match";
   //jump to <lebel> "password_label";


//Clear Screen;
   //Define a <Label>"accnum_label:"
```

```
//print "
        Enter a 5-digit account number
        ";
//take string input for account number to>"input";

   // for loop intial counter i=o to lenght of "input"<strlen(input)
    //if  "input[i]" digit then,Exp:isdigit(input[i])
       //if "input" length not equal than 5  then
           //Clear Screen;
          //print "Invalid format";
          //jump to <lebel> "accnum_label",Exp:goto accnum_label;

  //else
       //Clear Screen;
       //print "Invalid input";
       //jump to <lebel> "accnum_label",Exp:goto accnum_label;


//Copy data from "input" to>"chaccnum";
//print "Re-enter account number
         ";
//take string input for accnum2 to>"accnum2";

//if string is match  with "chaccnum" and  "accnum2"  then
    //set data to "std1.accnum"  from "strtol(chaccnum,&endptr,10)" ;
    //sends formatted output "std1.accnum" to a stream>"a.cListptr";

//else
    //Clear Screen;
    //print "Account Numbers do not match";
    //jump to <lebel> "accnum_label";

//Clear Screen;
   //Define a <Label>"acctype_label:"
//print   "
        Enter 1 to create Savings account - Enter 2 to create Current account
        ";

//take string input for Savings account to>"input";
```

```
    //set data to "input_int" from "strtol(input,&endptr,10)";

    //if input_int equal 1 or 2
        std1.acctype = input_int;
        //set data to "std1.acctype" from "input_int";
    //else
        //Clear Screen;
        //print "Invalid Input";
         //jump to <lebel> "acctype_label";

    //sends formatted output "std1.acctype" to a stream>"a.cListptr";
       //set "std1.balance" = 0;
    //sends formatted output "std1.balance" to a stream>"a.cListptr";
    //Close Streem >"a.cListptr";
    //Clear Screen;
    //print   "Account created

            Press any key to continue...";
 //call method>"customerHome(std1.accnum);"

-----------------------End method;

-----------------------Define method "c_logInPage()"
    //Clear Screen;
    //read data   from text file>"customerList.txt" and set to >"a.cListptr";
    //if "a.cListptr" is null then
        //print "
                No existing users

                Press any key to continue...";
        //call method "home()";
    //else
        //set "false" to "loginSuccess" variable

        //while not loginSuccess
            //set "false" to "loginSuccess" variable ;
            //print "
                  Enter Username
                  ";
```

```
        //take string input for username to>"std.username";
        //print "Enter Password
                ";
        //take string input for password to>"std.pass";

        //reads  data from "a.cListptr" and set to (file.name, file.mobnum, file.sex, file.username,
file.pass, file.accnum, file.acctype, file.balance);
          //if String compare succesfully for "std.username" and "file.username"  by function
"strcmp" then
            //String compare succesfully for "std.pass" and "file.pass" by function "strcmp" then
               //set data to "std.accnum" from "file.accnum"
               // set "False" to "loginSuccess" variable


        //cloase stream "a.cListptr";

        //if "loginSuccess" is false;
           //clear Screen;
           //cloase stream "a.cListptr"
           //print "
                   Wrong username or password. Press 1 to try again - Press 2 to go back to the
home screen
                  ";

                          //Define a <Label>"incorrect_label:"

           //take string input to>"choice";

           //if "choice" length equal 5 and "choice" is digit then
              //set data to "input_int" from function "strtol(choice,&endptr,10)";
              //Apply switch case on "input_int"
                 //when case 1 then
                    //clear screen;
                    //read data from file "customerList.txt" and set to "a.cListptr ";
                 //when case 2 then
                    //call method "home()";
                 //default
                    //clear screen;
                    //print  "Press 1 to try again - Press 2 to go back to the home screen";
```

```
                    //Jump to <lebel> "incorrect_label";
            //else
                //clear screen;
                //print  "Press 1 to try again - Press 2 to go back to the home screen";
                //Jump to <lebel> "incorrect_label";
      //call method "customerHome(std.accnum)"
----------End method;

----------Define method "customerHome(long accnum)"
   //clear screen;
   //print "From the list of choices below:

         ";
   //print "Press 1 to view personal details
         Press 2 to view account details
         Press 3 to view your transactions
         Press 0 to log out";
   //takse user input to "choice";
   //if "choice" lenght is 1 and "choice" is digit ,check by function "isdigit(choice[0])" then
      //set data to >"input_int" from function "strtol(choice,&endptr,10)";

      //apply switch case on "input_int"
         //when case 1 then
            //call method "personalDetails(accnum)";
         //when case 2 then
            //call method "accDetails(accnum)";
         //when case 3 then
            //call method "c_transactionList(accnum)";
         //default
            //call method "customerHome(accnum)";

  //else

    //call method "customerHome(accnum)"
----------End method;

----------Define method "personalDetails(long accnum)"
   //clear screen;
   //set file name "customerList.txt" to "cList[]" ;
```

```
   //read file data from "customerList.txt" to "a.cListptr";

      //reads  data from "a.cListptr" and set to (file.name, file.mobnum, file.sex, file.username,
file.pass, file.accnum, file.acctype, file.balance)
       //if "accnum" equal "file.accnum" then
          //copy data "file.name" to "personal.name";
          //copy data "file.mobnum" to "personal.mobnum";
          // set "file.sex" data to > "personal.sex";


   //close stream "a.cListptr";

   //print "
          Name:
             <personal.name>
          ";
   //print "Press 1 to update

          ";
   //print "Mobile number:
                     <personal.mobnum>
          ";

   //print  "Press 2 to update

          ";
   //if "personal.sex " equal 1 then
      //print "Gender:
                  Male
            ");
   //if "personal.sex " equal 2 then
      //print "Gender:
                  Female
                  ");

   //print "Press 0 to go back
          ";

   //take string input to>"choice";
```

```
  //if "choice" length is 1 and "choice" is digit then
    //set data to >"input_int" from function "strtol(choice,&endptr,10)";
    //apply switch case on "input_int"
            //when case 1 then
                for (int i=0;i<strlen (personal.name); i++){
                    personal.name[i] = " ";
                }
                // for loop intial counter i=o to lenght of "input"<strlen(personal.name)
                    //set "personal.name[i]" = " ";

                //clear screen
                //print "Enter first name
                            ";
                //Define a <Label>"firstname_label:"

                //take string input to>"name"
                // for loop intial counter i=o to lenght of "input"<strlen(name)
                    //if name[i] is string then (check by isalpha(name[i]))
                        //screen clear
                        //print "No numerals or special characters
                            Enter first name
                            ";
                //copy data to "personal.name" from "name";


    //when case 2 then
      //screen clear;
            //Define a <Label>"mobnum_label:";
      //print "enter your Malaysian mobile number
          ";
            //Define a <Label>"mymobnum_label:";
      //take string input to>"mobnum";

      // for loop intial counter i=o to lenght of "input"<strlen(mobnum)
        //if mobnum[i] is digit then
          //if "mobnum" length not equal 10 or "mobnum[0]" not equal "0" or "mobnum[1]"
not equal "1" then
          //has to start with 01, no special characters
            //if "mobcntr" grater than or equal 2 then
```

```
                    // clear screen
                    //print "Hint: Your number should be 10 digits long and should begin with '01'
                          ";
                    //set 0 to >"mobcntr";
                    //jump to section "mymobnum_label"
                 //clear screen;
                 //increase counter for "mobcntr++"
                 //print "Invalid format -- Please enter a valid Malaysian number:
                       ";
                 //jump to section "mobnum_label"

        //copy data to "personal.mobnum" from "mobnum";

      //when case 0 then
        //call method "customerHome(accnum)";

      //default
        //call method "personalDetails(accnum)";

   //else
      //call method "personalDetails(accnum)";

   //read data from file "customerList.txt" to >"a.cListptr";
   //apend data to "tmp_customerList.txt";
     //reads  data from "a.cListptr" and set to (file.name, file.mobnum, file.sex, file.username,
file.pass, file.accnum, file.acctype, file.balance)
      //if "accnum" equal "file.accnum" then
        //sends formatted output(personal.name, personal.mobnum, file.sex, file.username,
file.pass, file.accnum,file.acctype, file.balance) to a stream>"tmp"
      //else
        //sends formatted output(personal.name, personal.mobnum, file.sex, file.username,
file.pass, file.accnum,file.acctype, file.balance) to a stream>"tmp"

   //close stream >"a.cListptr";
   //close stream >"tmp";
   //remove >"cList";
   //rename file "tmp_customerList.txt" to  <cList>;

   //print "Succesfully updated
```

```
        Press any key to continue...";

   //call method "customerHome(accnum)";
----------End method;

----------Define method "accDetails(long accnum)"
   //clear screen;
   //read data from "customerList.txt" and set to >"a.cListptr";

   //reads  data from "a.cListptr" and set to (file.name, file.mobnum, file.sex, file.username,
file.pass, file.accnum, file.acctype, file.balance);

     //if "accnum" equa "file.accnum"
        //copy data to "acc.username" from "file.username";
        //copy data to "acc.pass" from "file.pass";
        //set "file.accnum" to "acc.accnum";
        //set "file.acctype" to "acc.acctype";
        //set "file.balance" to "acc.balance";


   //close strwam >"a.cListptr";
   //print "
        Username:
             acc.username
       ";
   //print "
        Password:
             acc.pass
        ";

   //print "
        Account number:
             acc.accnum
        ";

   //if "acc.acctype" equal 1 then
      //print "Account type:
                Savings_Account
          ";
```

```
   //else if "acc.acctype" equal 2 then
      //print "Account type:
                     Current_Account
            ";

   //print "Balance:
               "acc.balance"
               ";
   //print   "Press 1 to make a deposit
         Press 2 to withdraw cash
         Press 0 to go back
         ");


   //take string input to>"choice";

   //if "choice" equal 1 and "choice"" is disgit then
      //set data to "input_int" from "strtol(choice,&endptr,10)";

      //Apply switch case on "input_int"
                  //when case 1 then call method "deposit(accnum)";

                  //when case 2 then call method "withdraw(accnum)";

                  //when case 0 call method "customerHome(accnum)";

                  //when default then call method "accDetails(accnum)"

   //else
      //call method "accDetails(accnum)";
----------------------End method;

----------------------Define method "c_transactionList(long accnum)"
   //clear screen;
   //set "trans_exists" = false;
   //read data from "transList.txt" to "a.trans"";

   //if "a.trans" not null then
```

```
        //reads  data from "a.trans" and set to (file.accnum, transaction);
            //if file.accnum equal accnum then
                //set "trans_exists" to "true"
                //if transaction greater than 0 then
                    //print "Deposit: <transaction> MYR
                            ";
                //else if "transaction" less than 0 then
                    //print "Deposit: <abs(transaction)> MYR
                            ";
        //if trans_exists is false
            //print "No transactions available
                    ");
            //else
                        //print "No transactions available
                                            ";
    //close stream "a.trans";
    //print "
            Press any key to exit...";

    //call method "customerHome(accnum)";
-----------------------End method;

-----------------------Define method "deposit(long accnum)"
    //clear screen;
    //print "Note: You may only make a deposit of MAXIMUM 500 MYR at a time.
            ";

        //Define a <Label>"deposit_label:";
    //print "Enter the amount you wish to deposit:
            ";
    //takse string input to "input";

    // for loop intial counter i=o to lenght of "input"<strlen(input)
        //if "input[i]" is digit
            //clear screen
            //print "Invalid input
                    ");
            //jump to <lebel> "deposit_label";
```

```
   //set data to "input_int" from "strtol(input,&endptr,10)";
   //if "input_int" grater than 500 then
      //clear screen
      //print "You may only make a deposit of MAXIMUM 500 MYR at a time.
            ";
      //jump to <lebel> "deposit_label"

   //else if "input_int" equal 0 then
      //clear screen ;
      //print "Cannot deposit 0 MYR
            ";
      //jump to <lebel> "deposit_label"


   //set "input_int" data to "deposit";


   //read data from file "transList.txt" and set to "a.trans";
   //sends formatted output(accnum,deposit) to a stream >"a.trans"";
   //close stram "a.trans";
   //write data to file "transList.txt"


      //reads  data from "a.cListptr" and set to (file.name, file.mobnum, file.sex, file.username,
file.pass, file.accnum, file.acctype, file.balance) ;
      //if "accnum" equal"file.accnum" then
         //set "file.balance" to "balance"
         //set (balance + deposit) to newBalance;
         //sends formatted output(file.name, file.mobnum, file.sex, file.username, file.pass,
file.accnum,file.acctype, file.balance) to a stream>"tmp"      ;

         //else
         //sends formatted output(file.name, file.mobnum, file.sex, file.username, file.pass,
file.accnum,file.acctype, file.balance) to a stream>"tmp";

   //close stream "a.cListptr";

   //close stream "tmp";
```

```
      remove(cList);
   //remove "cList";
   //rename "tmp_customerList.txt" to <cList>;
   //clear screen;
   print   "
         <deposit> MYR has been deposited into your account

         Press any key to continue...";
   //call method "customerHome(accnum)";
----------End method;

----------Define method "withdraw(long accnum)"
   //clear screen;
   //read data from file "customerList.txt" and set to >"a.cListptr";

      //reads  data from "a.cListptr" and set to (file.name, file.mobnum, file.sex, file.username,
file.pass, file.accnum, file.acctype, file.balance)
       //if "accnum" equal "file.accnum" then
          //if "file.balance" equal 0 then
             //clear screen;
             //print "You have insufficient funds to make withdrawals
                   Press any key to continue...";
             //close stream "a.cListptr";
             //call method "accDetails(accnum)";



   //close stream "a.cListptr";

      //Define a <Label>"withdraw_label:";

   //print "Enter the amount you wish to withdraw:
         ";

   //take string input to "input";

   // for loop intial counter i=o to lenght of "input"<strlen(input)
      //if "input[i]" is digit
         //clear screen
         //print "Invalid input
```

```
                ";
          //jump to <lebel> "withdraw_label";

   //set data to "input_int" from "strtol(input,&endptr,10)";
   //if "input_int" equal 0 then
      //clear screen;
      //print "Cannot withdraw 0 MYR
             ";
      //jump to <lebel> "withdraw_label";

  //set data "input_int" to "withdrawal"
  //read data from file "customerList.txt" and set to "a.cListptr";
  //write data to file "tmp_customerList.txt"

      //reads  data from "a.cListptr" and set to (file.name, file.mobnum, file.sex, file.username,
file.pass, file.accnum, file.acctype, file.balance)
      //if "accnum" equal "file.accnum" then
         //if "withdrawal" equal "file.balance" then
            //clear screen;
            //print "Insufficient balance
                   ";
            //close stream "a.cListptr";
            //close stream "tmp";
            //jump to <lebel> "withdraw_label";


         //set "newBalance" by (file.balance - withdrawal)
                  //sends formatted output(file.name, file.mobnum, file.sex, file.username, file.pass,
file.accnum,file.acctype, file.balance) to a stream>"tmp"

         //else
         //sends formatted output(file.name, file.mobnum, file.sex, file.username, file.pass,
file.accnum,file.acctype, file.balance) to a stream>"tmp";

   //close stream "a.cListptr";
   //close stream "tmp";
   //remove "remove(cList)"
   //rename"tmp_customerList.txt" to <cList>;
   //write data to "transList.txt";
```

```
    //sends formatted output(accnum,withdrawal) to a stream>"a.trans";
    //close sltream "a.trans";
    //clear screen;
    //print   "<withdrawal> MYR has been withdrawn from your account
              Press any key to continue...";

    //call method "customerHome(accnum)"
----------------------End method
```

Concepts Used

In this project, a number of C programming concepts (Brain & and Crawford, 2011) were used to make the Banking System, and they are as follows:

- Functions:
  Standard library methods such as:
  fscanf() and printf() :-

```
while (fscanf(a.cListptr, "%s%s%d%s%s%ld%d%d", &file.name, &file.mobnum, &file.sex, &file.username, &file.pass, &file.accnum, &file.acc
    if (file.sex == 1) {
        if (file.acctype == 1) {
            printf("\nName: %s  ||  Mobile Number: %s  ||  Gender: Male  ||  Username: %s  ||  Password: %s  ||\n  Account Number: %ld
        }
        else if (file.acctype == 2) {
            printf("\nName: %s  ||  Mobile Number: %s  ||  Gender: Male  ||  Username: %s  ||  Password: %s  ||\n  Account Number: %ld
        }
    }
    else if (file.sex == 2) {
        if (file.acctype == 1) {
            printf("\nName: %s  ||  Mobile Number: %s  ||  Gender: Female  ||  Username: %s  ||  Password: %s  ||\n  Account Number: %l
        }
        else if (file.acctype == 2) {
            printf("\nName: %s  ||  Mobile Number: %s  ||  Gender: Female  ||  Username: %s  ||  Password: %s  ||\n  Account Number: %l
        }
```

  user defined methods such as a_logInPage() and home() :-

```
void home() { //Main menu
    system("cls");
    char choice[100];
    int input_int;
    char *endptr;
    printf("\nPress 1 to log in as Admin\nPress 2 if you are
    scanf("%s", &choice);
    if(strlen(choice) == 1 && isdigit(choice[0])){
        input_int = strtol(choice,&endptr,10);
        switch (input_int)
        {
        case 1:
            a_logInPage();
            break;
        case 2:
            customerMenu();
            break;
        case 0:
            exit(0);
        default:
            printf("\a");
            home();
        }

    } else {
        printf("\a");
        home();
    }
}

void a_logInPage() { //Admin log in
    struct {
        char username[30];
        char password[30];
```

as well as 3rd party library methods such as strtol() were used :-

```
scanf("%s", &choice);
if(strlen(choice) == 1 && isdigit(choice[0])){
    input_int = strtol(choice,&endptr,10);
    switch (input_int)
    {
    case 1:
        a_transactionList();
```

- Variables:
  These were used to store values in blocks of memory so that it could be used later, as can be seen in the snippets shown above.
- Data Types:
  A number of data types were used: ranging from “int” (integer) to “char[ ]” (string) to store the different types of values input by the customer/admin such as their name, password or deposit amount. Examples of these can be seen in the examples shown above. Pointers were also used, as can be seen here :-

```
char cList[] = "customerList.txt";
a.cListptr = fopen("customerList.txt", "r");
FILE* tmp = fopen("tmp_customerList.txt", "a+");
while (fscanf(a.cListptr,"%s%s%d%s%s%ld%d%d", &fil
    if (accnum == file.accnum) {
```

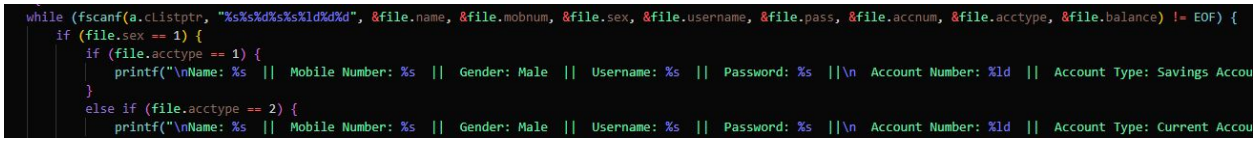
  and here :-

```
int input_int;
char *endptr;
printf("Note: You may only make a depos
deposit_label:
printf("Enter the amount you wish to de
scanf("%s",&input);
for(int i=0;i<strlen(input);i++){
    if(!isdigit(input[i])) {
        system("cls");
        printf("\aInvalid input\n");
        goto deposit_label;
    }
}
input_int = strtol(input,&endptr,10);
if (input_int>500){
```

- Operations:
  In this program, arithmetic operations were used to calculate the amount of money each customer has in their account after making a deposit or withdrawal :-

```
while (fscanf(a.cListptr,"%s%s%d%s%s%ld%d%d", &fi
    if (accnum == file.accnum) {
        int balance = file.balance;
        int newBalance = balance + deposit;
        fprintf(tmp,"%s\t%s\t%d\t%s\t%s\t%ld\t%d\
    } else {
        fprintf(tmp,"%s\t%s\t%d\t%s\t%s\t%ld\t%d\
```

Logical operations were also used for various scenarios, such as to check if the end of a text file has been reached while reading it :-



or to match a user's login credentials with their respective account details found in the text file, which is also done in the example shown immediately above.

- Loops:

  While loops were often used to repeat a set of instructions until the conditions for running that while loop stopped being met. An example is the file reader reading every line of text in the text file until the end of file is reached. The example shown immediately above is a good example of this.

  For loops were also used to loop through user inputs and check if they are valid, among other use cases :-

```
scanf("%s", &mobnum);
for (int i=0;i<strlen (mobnum); i++){
    if (isdigit(mobnum[i]))
    {
        if(strlen(mobnum) != 10 || mobnum[0] != '0' || mobnum[1] != '1') //has to start with 01, no spec
        {
            if (mobcntr >= 2){
                system("cls");
                printf ("\nHint: Your number should be 10 digits long and should begin with '01'\n");
                mobcntr = 0;
                goto mymobnum_label;
            }
            system("cls");
            mobcntr++;
            printf ("Invalid format\nPlease enter a valid Malaysian number:\n");
            goto mymobnum_label;
        }
    } else {
        system("cls");
        printf ("Invalid input\nPlease e");
        goto mobnum_label;
    }
}
```

- Data structures:
  Character arrays were used to store strings input by the customer. Examples of this can be seen in the snippets above.
- Preprocessor operations:
  Code from C libraries were used using the #include operator :-

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h> // for various functions such as the strtok() and strtol() function
#include <string.h> //to use strcmp(), strcpy() etc.
#include <stdbool.h> //for bool datatype
#include <ctype.h> //to use isalpha() and isdigit() functions
```

Additional Features Used

- The administrator's menu page has been formatted to make it easy to distinguish between each customer as well as each customer's individual account data :-

```
Name: Farhan  ||  Mobile Number: 0172839276  ||  Gender: Male  ||  Username: farhanalvi01  ||  Password: farhanpass  ||
  Account Number: 12345  ||  Account Type: Current Account  || Balance: 0 MYR


Name: Ahsan  ||  Mobile Number: 0182736479  ||  Gender: Male  ||  Username: ahsankhan02  ||  Password: ahsanpass  ||
  Account Number: 54321  ||  Account Type: Savings Account  || Balance: 0 MYR


Press 1 to view system transaction history
Press 2 to remove a user
Press 0 to to log out
```

This has been done by formatting the output around string literals, as well as using a while loop to print out all available customers on the system :-

```
while (fscanf(a.cListptr, "%s%s%d%s%s%ld%d%d", &file.name, &file.mobnum, &file.sex, &file.username, &file.pass, &file.accnum, &file.acctype, &file.balance) != EOF) {
    if (file.sex == 1) {
        if (file.acctype == 1) {
            printf("\nName: %s  ||  Mobile Number: %s  ||  Gender: Male  ||  Username: %s  ||  Password: %s  ||\n  Account Number: %ld  ||  Account Type: Savings Account
        }
        else if (file.acctype == 2) {
            printf("\nName: %s  ||  Mobile Number: %s  ||  Gender: Male  ||  Username: %s  ||  Password: %s  ||\n  Account Number: %ld  ||  Account Type: Current Account
        }
    }
    else if (file.sex == 2) {
        if (file.acctype == 1) {
            printf("\nName: %s  ||  Mobile Number: %s  ||  Gender: Female  ||  Username: %s  ||  Password: %s  ||\n  Account Number: %ld  ||  Account Type: Savings Accou
        }
        else if (file.acctype == 2) {
            printf("\nName: %s  ||  Mobile Number: %s  ||  Gender: Female  ||  Username: %s  ||  Password: %s  ||\n  Account Number: %ld  ||  Account Type: Current Accou
        }
    }
```

- As you can see from the section of the program shown above, there is also a feature to view the entire system's transaction history. This can be viewed in a table format where each user's transactions are distinctly shown by having their account number visible next to the transaction. The program tells you whether the transaction was a deposit or a withdrawal, and also tells you the amount transferred. :-

```
Account Number  Transaction

12345           Deposit: 500 MYR
12345           Deposit: 500 MYR
12345           Deposit: 250 MYR
12345           Withdraw: 60 MYR
12345           Withdraw: 300 MYR
54321           Deposit: 500 MYR
54321           Withdraw: 40 MYR
54321           Withdraw: 5 MYR
54321           Deposit: 400 MYR
54321           Withdraw: 85 MYR

Press any key to exit...
```

This is the source code behind that :-

```
void a_transactionList(){
    system("cls");
    int transaction;
    bool trans_exists = false;
    a.trans = fopen("transList.txt","r");
    if(a.trans != NULL){
        printf("Account Number\tTransaction\n\n");
        while(fscanf(a.trans,"%ld%d",&file.accnum,&transaction) != EOF){
            trans_exists = true;
            if(transaction > 0){
                printf("%ld\t\tDeposit: %d MYR\n",file.accnum,transaction);
            } else if(transaction < 0){
                printf("%ld\t\tWithdraw: %d MYR\n",file.accnum,abs(transaction));
            }
        }
        if(!trans_exists){
            printf("No transactions available\n");
        }
    } else {
        printf("No transactions available\n");
    }
    fclose(a.trans);

    printf("\nPress any key to exit...");
    _getch();
    adminMenu();
}
```

- The program has been designed with user experience in mind, as can be seen in the following example :-

```
Username:
farhanalvi01

Password:
farhanpass

Account number:
12345

Account type:
Current_Account

Balance:
890 MYR

Press 1 to make a deposit
Press 2 to withdraw cash

Press 0 to go back
```

```
12 MYR has been deposited into your account

Press any key to continue...
```

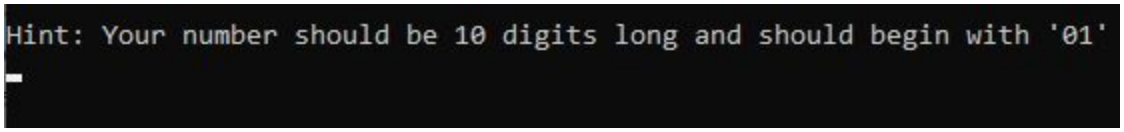
As you can see, by having it such that the user can quickly perform a transaction and then be directed back to view their account balance, it makes for a responsive experience. The code behind this example is fairly arbitrary and is simply a display of the creative decision behind it.

- There is also a validation system that prevents new or existing customers from entering a mobile number that doesn't conform to Malaysia's standard. When the user types something incorrect, either one of these outputs are shown, depending on the input :-

```
Invalid format
Please enter a valid Malaysian number:
```

```
Invalid input
Please enter your Malaysian mobile number
```

If the user makes too many mistakes, this is the output :-

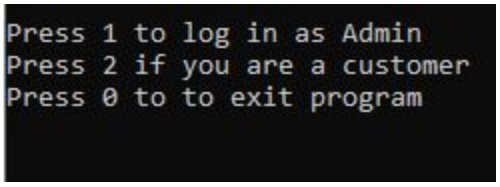


Source code behind this validation system :-

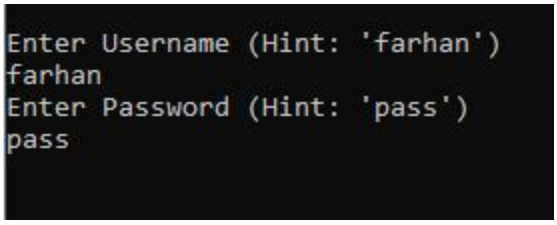
```
char mobnum[50];
int mobcntr = 0;
system("cls");
printf("\nE");
mobnum_label:
printf("nter your Malaysian mobile number\n");
mymobnum_label:
scanf("%s", &mobnum);
for (int i=0;i<strlen (mobnum); i++){
    if (isdigit(mobnum[i]))
    {
        if(strlen(mobnum) != 10 || mobnum[0] != '0' || mobnum[1] != '1') //has to start with 01, no special characters
        {
            if (mobcntr >= 2){
                system("cls");
                printf ("\nHint: Your number should be 10 digits long and should begin with '01'\n");
                mobcntr = 0;
                goto mymobnum_label;
            }
            system("cls");
            mobcntr++;
            printf ("Invalid format\nPlease enter a valid Malaysian number:\n");
            goto mymobnum_label;
        }
    } else {
        system("cls");
        printf ("Invalid input\nPlease e");
        goto mobnum_label;
    }
}
```

Sample Outputs

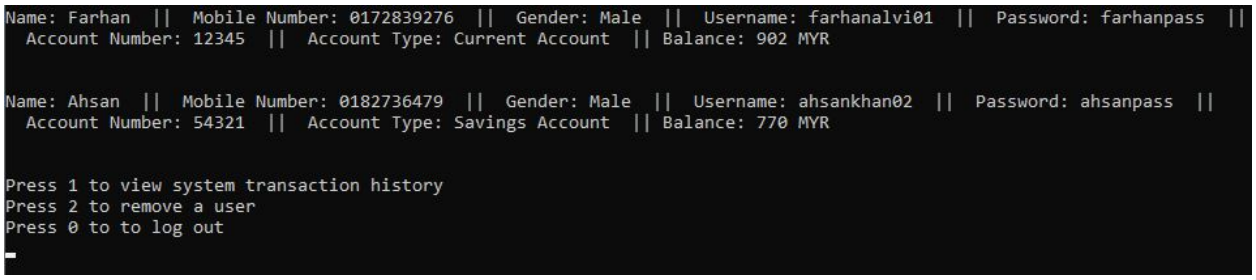
This is the home screen:



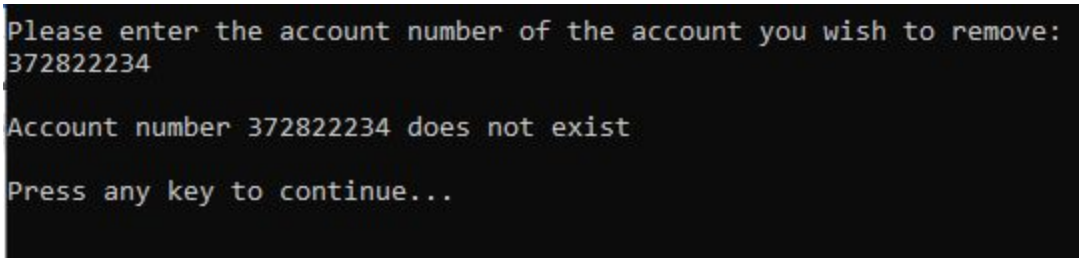
Entering 1 takes you to an admin log-in page, where you have to type in the administrator's credentials:



After logging in, the admin is greeted by a list of users and their details, along with some extra functionalities:



The system transaction history feature has already been explained in the "Additional Features Used" section, so the following snippet instead shows the screen where the admin can remove a user from the system:



As you can see, there is a validation system here that checks the data store to see if such an account even exists.

On the customer side of things, entering 2 on the home page takes you to the customer menu, which looks like this:

```
Press 1 to log in
Press 2 to register as a new user
Press 0 to go back
```

The user enters their credentials:

```
Enter Username
farhanalvi01
Enter Password
farhanpass
```

And after a successful log-in, they are greeted with this menu:

```
From the list of choices below:

Press 1 to view personal details
Press 2 to view account details
Press 3 to view your transactions
Press 0 to log out
```

From here, users can view their personal details:

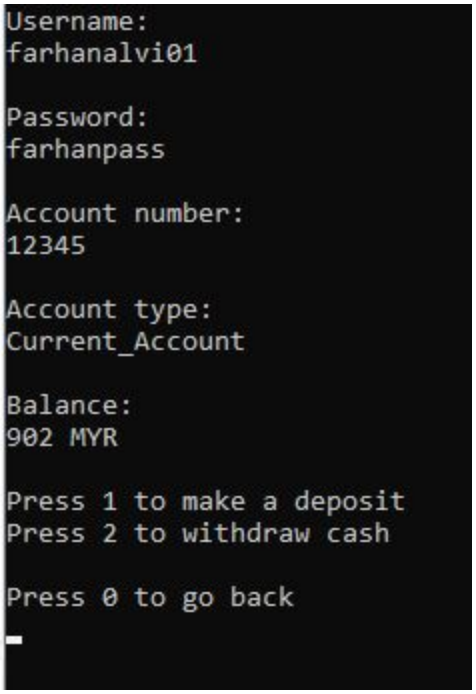
```
Name:
Farhan
Press 1 to update

Mobile number:
0172839276
Press 2 to update

Gender:
Male

Press 0 to go back
```

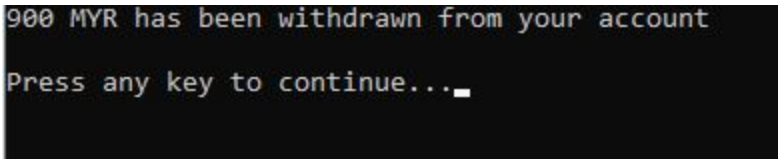
Update their name or mobile number:

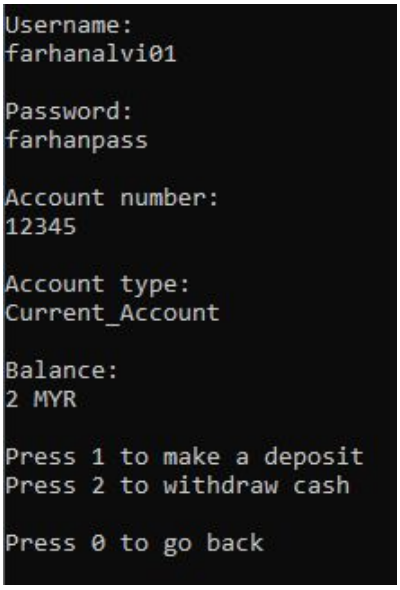
```
Enter first name
Mark
Succesfully updated

Press any key to continue...
```

```
Name:
Mark
Press 1 to update

Mobile number:
0172839276
Press 2 to update

Gender:
Male

Press 0 to go back
```

They can view their account details, as well as make transactions which affect their account balance:







And they can view their previous transactions made:

Conclusion

This project has given me strong fundamentals in functional programming and developing a fully operating system with all CRUD operations. It has also familiarized me with the C programming language and it’s subtleties and nuances. I have formed a fundamental understanding of the development process of a program and it will aid me in future endeavours.

References

&, M. B. and Crawford, S. (2011) *How C Programming Works*, *HowStuffWorks*. HowStuffWorks. Available at: https://computer.howstuffworks.com/c-programming.htm (Accessed: 22 June 2020).