**INTRODUCTION**

This mini project implements the lpg booking. The system will help the customers by providing a simple user interactive interface for booking gas which will save the time. The system will display the user the number of gas they booked . It also gives the ease by helping them make the booking process faster and easier to maintain. There are various steps to book a gas. The main objective of this project is to manage the details of the consumers who registered themselves. It enables the consumers to book the lpg very easily. The admin provide username and password to login. It handles the booking details and keep track the queries/complaints.Authorised consumer of that system first register and gets the username and password with which he/she will be able to book and mention the complaints if any and then the admin check the issue by mentioning the particular date. The admin can easily store, update and delete the customer details. In simple language the project provides such user friendliness and easy understandability that even a novice find it easy to use.

**We in our system focused on the following keywords:**

* Consumer Record is maintained.
* Edit, update, and deletion of the record.
* Check consumer is valid to book a gas.
* Booking records should be maintaining.

**OBJECTIVES**

The main objective of this project is to manage the details of the consumers who registered themselves. It enables the consumers to book the lpg very easily. The admin provide username and password to login. It handles the booking details and keep track the queries/complaints.

* It is used to save the time.
* It reduces the human effects.
* It manages the detail of the students properly.

**TOOLS / ENVIRONMENT**

|  |  |
| --- | --- |
| **System** | **Hardware/Software Requirements** |
| Client/ Server System Operating System | Window 10 |
| Language Used | C++ |
| Compiler Used | CodeBlocks |
| RAM | Minimum 512 mbs |
| Disk Space | Minimum 500 mbs |

**ANALYSIS DOCUMENT**

**ENTITY RELATIONSHIP DIAGRAM**

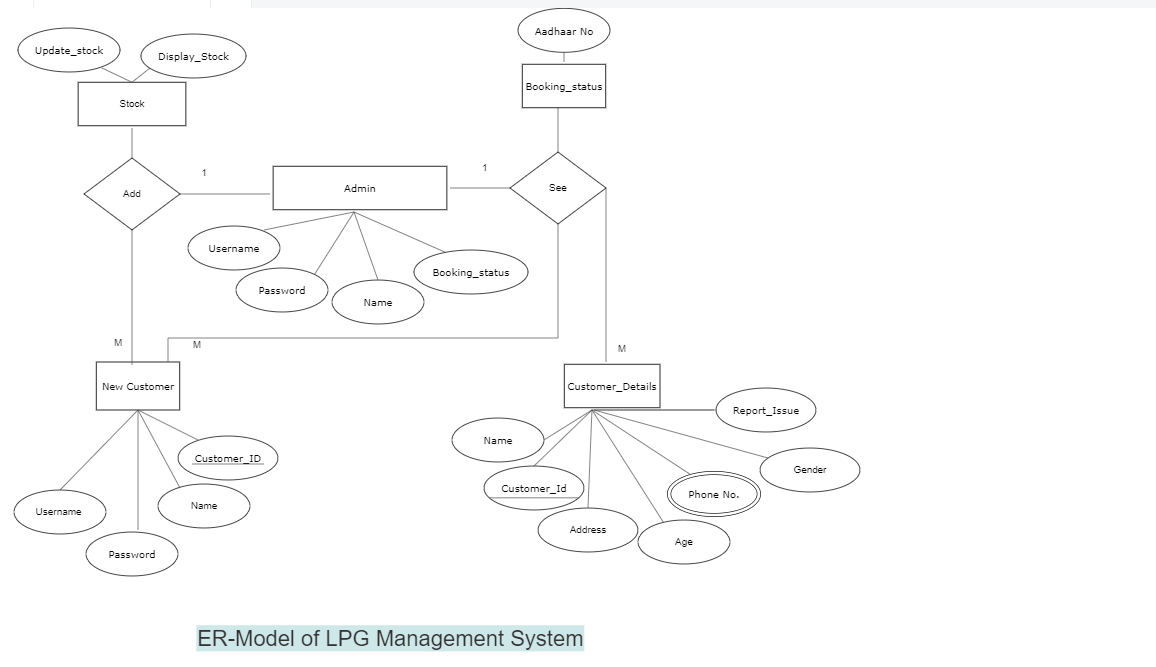


Figure 4.2 ER Diagram

**DATA FLOW DIAGRAM**

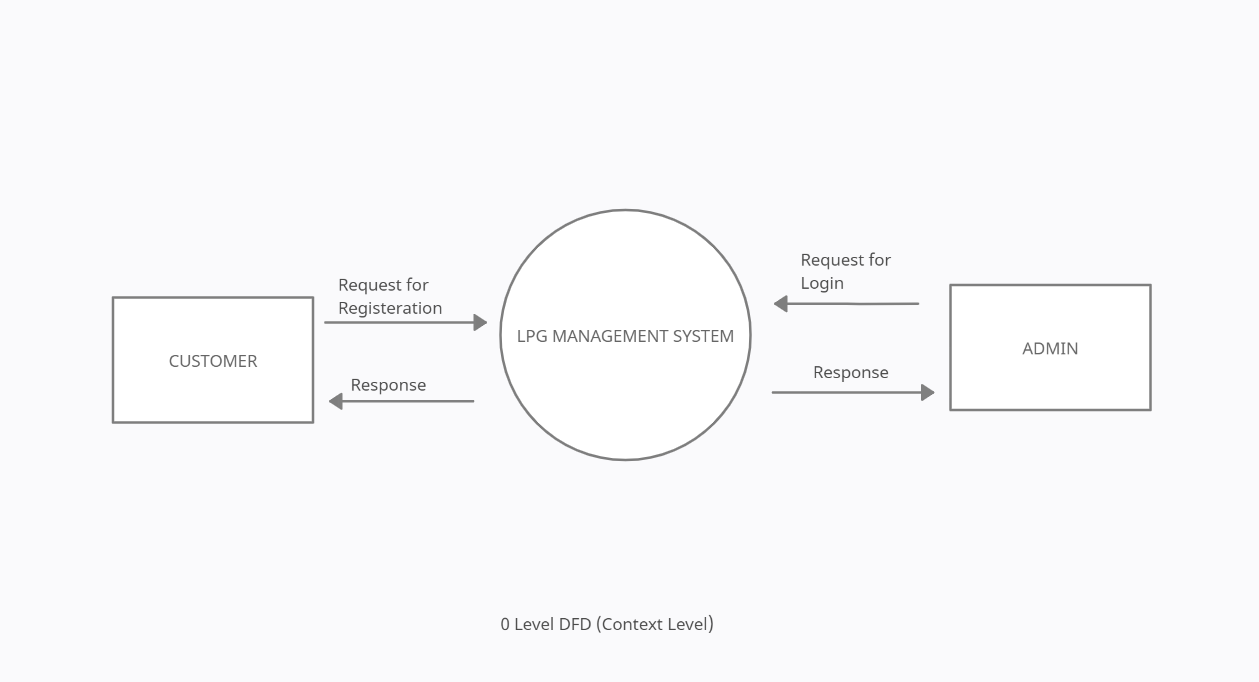


Figure 4.3 DFD ( 0 Level)

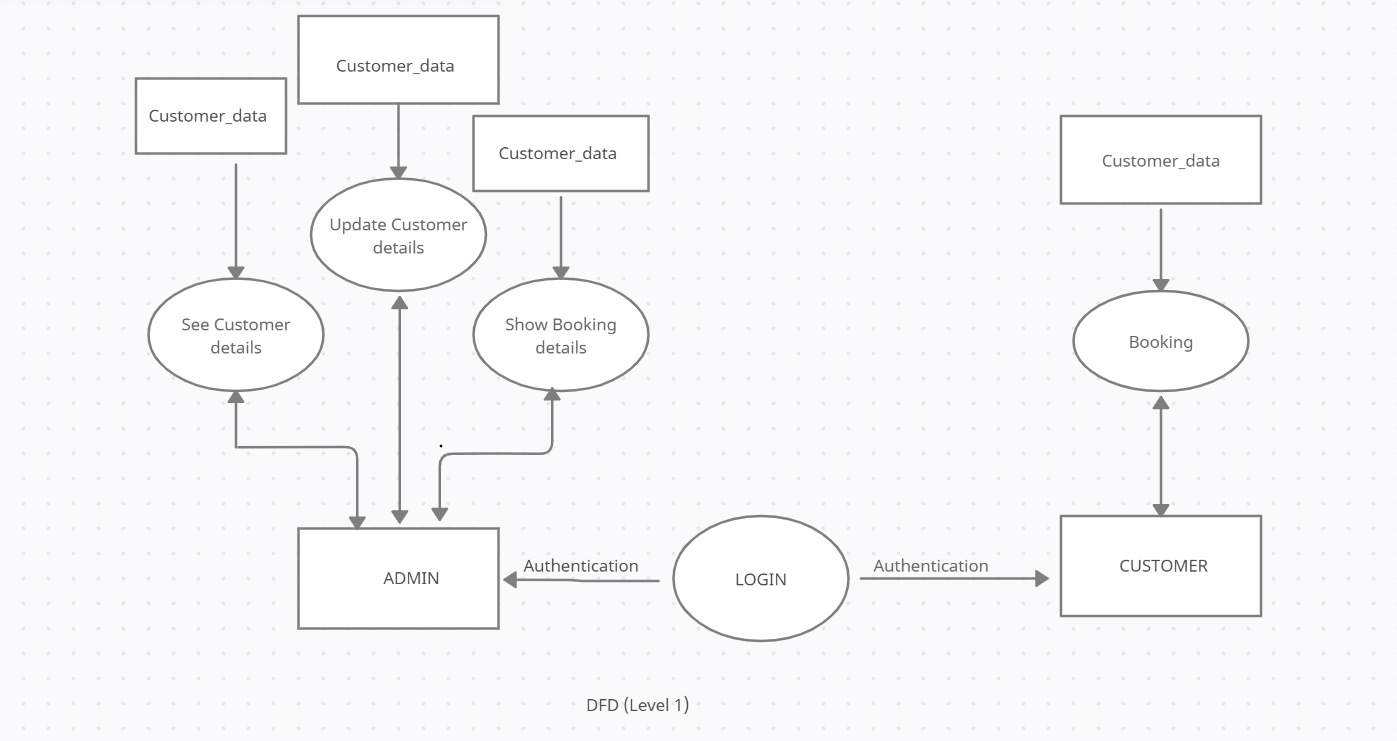


Figure 4.3 DFD (1 Level)

**DATA DICTIONARY**

**Admin**

|  |  |  |  |
| --- | --- | --- | --- |
| Data Name | Data Type | Data Length | Description |
| UserName | Char | 20 | Admin user name |
| Password | Char | 20 | Password of  Admin |
| **Customer** |

|  |  |  |  |
| --- | --- | --- | --- |
| Data Name | Data Type | Data Length | Description |
| Customer\_Name | Char | 100 | Name Of Customer |
| Customer\_Age | Int | 10 | Age Of Customer |
| Customer\_Complaints | Char | 1000 | Complaints |
| Customer\_Address | Char | 70 | Address Of Customer |
| Customer\_Gender | Char | 5 | Gender Of Customer |

**DESIGN DOCUMENTS**

**MODULARIZATION DETAILS**

Designs of this document are done with various functions where each function forms some tasks.

* **Void Menu ()** : The Front Panel which includes 3 options Admin , Customer and Exit .
* **Void admin ()** : This Module Displays the main menu of admin and performs all the Menu options available in the Function.
* **Void admin\_password ()** : This Module Accepts the Username and password and checks if it is right or not . It is the login page of Admin.
* **Void customer ()** : This Module Displays the main menu of customer which includes only Booking and Report Issues .
* **Void cust\_password ()** : This Module Asks if you are registered or not and it works as the login page of customer .
* **Void stock ()** : This Module is perfomed by Admin to Update the stock details or display the stock details.
* **Void new\_connection ()** : This Module is perfomed by admin for registering new customer.
* **Void display\_data ()** : This Module is perfomed by admin for displaying the details of registered customer.
* **Void show\_data ()** : This module is used in display\_data () module for displaying details.
* **Void update\_customer\_data ()** : This module is used by admin for modifying the Details of registered customers.
* **Void delete\_customer\_data ()** : This module is used by admin for deleting the data of a specific Customer.
* **Void show\_bookings ()** : This Module checks if the customer have booked the Gas or not.
* **Void show\_complaints ()** : This Module shows the complaints of customers to admin by fetching dates.
* **Void issue ()** : This Module is used in show\_complaints () module for displaying Complaints.
* **Void booking ()** : This Module is used by Customer for booking Gas .
* **Void report\_issue ()** : This Module is used by customer for Reporting the issues they faced .

**SOURCE CODE**

// LPG MANAGEMENT SYSTEM BY - Harsh Sharma & Himanshi Bisht //

#include<iostream>

#include<iomanip>

#include<string>

#include<windows.h>

#include<fstream>

#include<conio.h>

#define TOTAL\_GAS 400

using namespace std;

class lpg\_management

{

// ===> FOR CUSTOMER LOGIN <=== //

protected:

string username;

string password;

string usn;

// ===> FOR DATE <=== //

int dd,mm,yy;

// ===> FOR ADMIN TO UPDATE STOCK <=== //

int sum\_stock = 0;

int add, comp;

// ===> FOR REGISTRATION THROUGH ADMIN <=== //

char name[100];

char gender;

char complaint[1000];

char address[70];

int age;

string adhaar;

string phone\_no;

public:

// ===> LOGIN MODULES <=== //

void menu(); // 61-96

void admin(); // 98-152

void admin\_password(); // 154-206

void customer(); // 208-242

void cust\_password(); // 244-325

// ADMIN MODULES //

void Stock(); // 329-384

void new\_connection(); // 386-499

void display\_data(); // 501-539

void show\_data(); // 541-555

void update\_customer\_data();// 557-709

void delete\_customer\_data();// 711-759

void show\_bookings(); // 762-817

void show\_complaints(); // 820-878

void show\_issue(); // 880-891

// CUSTOMER MODULES //

void booking(); // 895-945

void report\_issue(); // 947-989};

// Menu Module //

void lpg\_management::menu()

{

system("cls");

int choice;

do

{

system("cls");

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

// Main Menu //

cout << "\n\n\t\t -->> MAIN MENU <<--";

cout << "\n\n\t\t -->>1. ADMIN";

cout << "\n\n\t\t -->>2. CUSTOMER";

cout << "\n\n\t\t -->>3. EXIT";

cout << "\n\n\t\tEnter Choice: ";

cin >> choice;

switch(choice)

{

case 1:

admin();

break;

case 2:

customer();

break;

case 3:

cout << "\n\n\t\t\t\t\t YOU'VE BEEN EXITED..! \n\n\t\t\t";

exit(0);

default:

cout << "\n\n\t\t\t Invalid Choice... Press Try Again.. \n ";

}

cin.ignore();

cin.get();

}while(choice!=3);

}

// ADMIN MENU //

void lpg\_management::admin()

{

system("cls");

admin\_password();

int admin\_choice;

do{

system("cls");

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

//----------ADMIN MENU OPTIONS----------//

cout << "\n\n\t\t-->> ADMIN MENU <<--";

cout << "\n\n\t\t 1. Stock Details";

cout << "\n\n\t\t 2. Apply For A Connection ";

cout << "\n\n\t\t 3. Show Customer Details ";

cout << "\n\n\t\t 4. Update/Modify Details ";

cout << "\n\n\t\t 5. Delete Customer Data";

cout << "\n\n\t\t 6. Show Bookings";

cout << "\n\n\t\t 7. Show Customer Complaints";

cout << "\n\n\t\t 8. Log Out";

cout << "\n\n\t\t Enter Choice : ";

cin >> admin\_choice;

switch(admin\_choice)

{

case 1:

Stock();

break;

case 2:

new\_connection();

break;

case 3:

display\_data();

break;

case 4:

update\_customer\_data();

break;

case 5:

delete\_customer\_data();

break;

case 6:

show\_bookings();

break;

case 7:

show\_complaints();

break;

case 8:

menu();

default:

cout << "\n\n\t\t\t Invalid Choice... Please Try Again....";

getch();

break;

}

}while(admin\_choice!=8);

}

// ADMIN LOGIN //

void lpg\_management::admin\_password()

{

system("cls");

char a\_name[20];

char a\_password[20];

int ch,i=0,capt=0,otp=0;

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\n\t\t -->> ADMIN LOGIN <<--";

cout << "\n\n\t\tEnter Your Name : ";

cin >> a\_name;

cout << "\n\t\tEnter Password : ";

while((ch = getch()) != 13)

{

cout << "\*";

a\_password[i]=ch;

i++;

}

a\_password[i]='\0';

srand(time(0));

capt = rand();

cout << "\n\n\t\tOTP : "<< capt;

cout << "\n\n\t\tEnter the OTP : ";

cin >> otp;

if((strcmp(a\_name,"harsh") == 0) && (strcmp(a\_password, "admin") == 0) && (capt == otp) || (strcmp(a\_name,"himanshi") == 0) && (strcmp(a\_password, "admin") == 0) && (capt == otp))

{

cout << "\n\n\n\t\t\t\t\t| VERIFYING ADMIN |\n\t\t\t\t\t";

for (int a = 1; a < 8; a++)

{

Sleep(400);

cout << "...";

}

cout << "\n\n\t\tAccess Granted..\n\n";

system("PAUSE");

system("cls");

}

else

{

cout << "\n\n\n\t\t\t\t\t| VERIFYING ADMIN |\n\t\t\t\t\t";

for (int a = 1; a < 8; a++)

{

Sleep(400);

cout << "...";

}

cout << "\n\n\t\tAccess Not Granted..\n\n";

system("PAUSE");

system("cls");

menu();

}

}

// CUSTOMER MENU //

void lpg\_management::customer()

{

cust\_password();

system("cls");

int c;

do{

system("cls");

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

//----------CUSTOMER MENU OPTIONS--------------//

cout << "\n\n\t\t -->> Customer Main Menu <<--";

cout << "\n\n\t\t1.Booking ";

cout << "\n\n\t\t2.Report Issue ";

cout << "\n\n\t\t3.Logout ";

cout << "\n\n\t\tChoose Any Option : ";

cin >> c;

switch(c)

{

case 1:

booking();

break;

case 2:

report\_issue();

break;

case 3:

menu();

default:

cout << "\n\n\t\t Invalid Choice...Please Try Again...";

}

cin.ignore();

cin.get();

}while(c!=3);

}

// CUSTOMER LOGIN //

void lpg\_management::cust\_password()

{

system("cls");

int cust\_choice;

string fname,usern,addr,phn;

ofstream file1;

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\n\t\t -->> Customer Menu <<--";

cout << "\n\n\t\t 1. Create Account";

cout << "\n\n\t\t 2. Login";

cout << "\n\n\t\t 3. Back";

cout << "\n\n\t\t Enter Choice : ";

cin >> cust\_choice;

if(cust\_choice == 1)

{

system("cls");

cout << "\n\n\t\t -->> REGISTRATION USER <<--";

cout << "\n\n\t\tEnter Your Name : ";

cin.ignore();

getline(cin, username);

cout << "\n\t\tEnter Your Username : ";

cin >> usn;

cout << "\n\t\tEnter your Password : ";

cin >> password;

fname=usn+".txt";

file1.open(fname.c\_str());

file1<<usn<<endl

<<username<<endl

<<password<<endl;

cout << "\n\n\tYou are successfully registered.. ";

cout << "\n\n\tPress Any key to continue ..";

getch();

file1.close();

customer();

}

else if(cust\_choice == 2)

{

system("cls");

ifstream file2;

string u\_name, u\_pass;

//---------LOGIN CUSTOMER---------//

cout << "\n\n\t\t -->> LOGIN USER <<--";

cout << "\n\n\t\tEnter Your Username : ";

cin >> u\_name;

cout << "\n\t\tEnter Your Password : ";

cin >> u\_pass;

fname = u\_name + ".txt";

file2.open(fname.c\_str());

if (!file2.is\_open() && file2.fail())

{

cout << "\n\n\t\tYou are not registered, please register before logging in.\n";

file2.close();

getch();

cust\_password();

}

getline(file2, usern);

getline(file2, u\_pass);

if (u\_name == usn && u\_pass == password)

{

cout << "\n\n\t\tYou are successfully logged in..!";

cout << "\n\n\t\tPress Enter..";

Sleep(500);

getch();

customer();

}

}

else if(cust\_choice==3)

{

menu();

}

else

{

cout << "\n\n\t\t Invalid Choice...Please Try Again...";

getch();

cust\_password();

}

}

//===> ADMIN MODULES <===//

// STOCK MODULE //

void lpg\_management::Stock()

{

fstream file;

system("cls");

cout << "\n\n\t\t-->> STOCK IN COMPANY <<--";

cout << "\n\n\t\t1. Update Or Add the Stock ";

cout << "\n\n\t\t2. Display the stock ";

cout << "\n\n\t\t3. Back";

cout << "\n\n\t\tEnter Choice: ";

cin >> comp;

if(comp == 1)

{

system("cls");

file.open("company1.txt", ios::out);

cout << "\n\t\t -->> For INDANE LPG <<--";

cout << "\n\n\tEnter Number Of Gas You Want To Add To Their Warehouse : ";

cin >> add;

sum\_stock = sum\_stock + add;

file << sum\_stock;

file.close();

cout << "\n\n\tStock In Indane : " << sum\_stock;

cout << "\n\t\tSUCCESSFULLY ADDED STOCK TO THE INDANE";

cout << "\n\n\nPress Any Key To Continue..";

getch();

}

else if(comp == 2)

{

system("cls");

ifstream ind;

fstream file1;

cout << "\n\n\t\t->> Available Gas Stock <<--\n";

int i = 0;

file1.open("company1.txt"); // Open Company

file1 >> sum\_stock;

ind.open("Customer\_Data.txt"); // Data take from file

ind.seekg(0, ios::beg);

while (ind.read((char \*)this, sizeof(lpg\_management)));

{

i++;

}

ind.close();

int s = TOTAL\_GAS + sum\_stock - i;

file1.close();

cout << "\n\n\t\tAvailable Number Of GAS are: " << s;

cout << "\n\n\t\tPress Any Key To Continue !";

getch();

}

else

{

cout << "\n\n\t\t\t Invalid Choice... Please Try Again....";

getch();

Stock();

}

}

// REGISTER NEW CUSTOMER //

void lpg\_management::new\_connection()

{

int index, check = 0;

fstream file;

system("cls");

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\n\t\t-->> CUSTOMER REGISTRATION <<--";

V:

cout << "\n\n\t\tEnter Name: ";

fflush(stdin);

cin.getline(name, 100);

// for loop to traverse the whole string

for (index = 0; name[index]; index++)

{

for (index = 0; name[index]; index++)

{

// if condition to check if the string Have Alphabet

if ((name[index] >= 'A' && name[index] <= 'Z') || (name[index] >= 'a' && name[index] <= 'z') || (name[index] == 32))

{

check++;

}

else

{

check = 0;

break;

}

}//Inner For Loop Ends Here//

if (check != 0)

{

cout << "\n\t\tValid Name";

break;

}

else

{

cout << "\n\t\tInvalid Name";

goto V;

break;

}

}

C:

cout << "\n\n\t\tEnter Aadhar Number : ";

cin >> adhaar;

if (adhaar.length() != 12)

{

cout << "\n\t\tInvalid Aadhar Card Number !";

goto C;

}

else

{

// The first letter should not start with 0

if (adhaar[0] == '0')

{

cout << "\n\t\tInvalid Aadhar Card Number";

goto C;

}

else

{

cout << "\n\t\tValid Aadhar Card Number";

}

}

B:

cout << "\n\n\t\tEnter Your Mobile Number: ";

cin >> phone\_no;

if (phone\_no.length() != 10)

{

cout << "\n\t\tInvalid Phone Number";

goto B;

}

else

{

// The First letter should not start with 0 or 1

if (phone\_no[0] == '0' || phone\_no[0] == '1')

{

cout << "\n\t\tInvalid Phone Number";

goto B;

}

else

{

cout << "\n\t\tValid Phone Number";

}

}

GEN:

cout << "\n\n\t\tEnter Gender (M/F/T): ";

cin >> gender;

if( gender == 'M' || gender == 'm' || gender == 'F' || gender == 'f' || gender == 'T' || gender == 't' )

{

cout << "\n\t\tValid Gender";

}

else

{

cout << "\n\t\tInvalid Gender";

goto GEN;

}

G:

cout << "\n\n\t\tEnter Age: ";

cin >> age;

if(age < 12 || age > 100)

{

cout << "\n\n\t\tInvalid Age";

goto G;

}

cout << "\n\t\tEnter Permanent Address: ";

fflush(stdin);

cin.getline(address, 70);

file.open("customer\_Data.txt", ios::app );

file.write((char \*)this, sizeof(lpg\_management));

cout << "\n\t\tYOU HAVE SUCCESSFULLY APPLIED FOR A NEW CONNECTION !";

cout << "\n\n\t\tPress Any Key To Continue !";

file.close();

getch();

}

// DISPLAY DATA OF THE REGISTERED CUSTOMER //

void lpg\_management::display\_data()

{

int count = 0;

string vadhaar;

fstream file;

file.open("customer\_Data.txt", ios::in );

system("cls");

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\n\t\t-->> SEARCH DATA BY AADHAR <<--";

cout << "\n\n\t\tEnter Aadhar No.: ";

cin >> vadhaar;

if (!file)

{

cout << "\nFile Not Found!";

}

else

{

file.read((char \*)this, sizeof(lpg\_management));

}

while (!file.eof())

{

if (vadhaar.compare(adhaar) == 0)

{

count++;

show\_data();

}

file.read((char \*)this, sizeof(lpg\_management));

}

if (count == 0)

{

cout << "\n\t\tRecord Not Found!";

}

file.close();

cout << "\n\n\t\tPress Any Key To Continue !";

getch();

}

// MODULE TO READ DATA FROM FILE (USED IN DISPLAY MODULE) //

void lpg\_management::show\_data()

{

system("cls");

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\n\t\tThe Details are shown below..!";

cout << "\n\n\t\tName : " << name;

cout << "\n\n\t\tAadhar No.: " << adhaar;

cout << "\n\n\t\tMobile No.: " << phone\_no;

cout << "\n\n\t\tGender : " << gender;

cout << "\n\n\t\tAge : " << age;

cout << "\n\n\t\tAddress : " << address;

}

// MODULE TO MODIFY OR UPDATE THE DETAILS OF CUSTOMER //

void lpg\_management::update\_customer\_data()

{

system("cls");

int choice;

int index, check = 0;

string uadhaar;

int found = 0;

fstream file, file1;

file.open("customer\_data.txt", ios::in);

if (!file)

{

cout << "\n\t\t\tFile Not Found!";

getch();

file.close();

}

else

{

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\n\t\t-->> SEARCH DATA BY AADHAR NO. TO MODIFY <<--";

cout << "\n\n\t\tEnter Aadhar No.: ";

cin >> uadhaar;

file1.open("record.txt", ios::app | ios::out);

file.read((char \*)this, sizeof(lpg\_management));

while (!file.eof())

{

if (uadhaar!=adhaar)

{

file1.write((char \*)this, sizeof(lpg\_management));

}

else

{

cout << "\n\t\tWhat Do You Want To Update ?";

cout << "\n\t\t1.Name";

cout << "\n\t\t2.Mobile Number";

cout << "\n\t\t3.Gender";

cout << "\n\t\t4.Age";

cout << "\n\t\t5.Address";

cout << "\n\t\tEnter Choice : ";

cin >> choice;

switch (choice)

{

case 1:

NAME:

cout << "\n\n\t\tEnter Name: ";

fflush(stdin);

cin.getline(name, 100);

// for loop to traverse the whole string

for (index = 0; name[index]; index++)

{

for (index = 0; name[index]; index++)

{

// if condition to check if the string Have Alphabet

if ((name[index] >= 'A' && name[index] <= 'Z') || (name[index] >= 'a' && name[index] <= 'z') || (name[index] == 32))

{

check++;

}

else

{

check = 0;

break;

}

}

if (check != 0)

{

cout << "\n\t\tValid Name";

break;

}

else

{

cout << "\n\t\tInvalid Name";

goto NAME;

break;

}

}

break;

case 2:

PHN:

cout << "\n\t\t Enter Mobile No.: ";

cin >> phone\_no;

if (phone\_no.length() != 10)

{

cout << "\n\t\tInvalid Phone Number";

goto PHN;

}

else

{

// The First letter should not start with 0 or 1

if (phone\_no[0] == '0' || phone\_no[0] == '1')

{

cout << "\n\t\tInvalid Phone Number";

goto PHN;

}

else

{

cout << "\n\t\tValid Phone Number";

}

}

break;

case 3:

GENDER:

cout << "\n\t\t Enter Gender (M/F): ";

cin >> gender;

if( gender == 'M' || gender == 'm' || gender == 'F' || gender == 'f' || gender == 'T' || gender == 't' )

{

cout << "\n\t\tValid Gender";

}

else

{

cout << "\n\t\tInvalid Gender";

goto GENDER;

}

break;

case 4:

AGE:

cout << "\n\t\t Enter Age : ";

cin >> age;

if(age < 12 || age > 100)

{

cout << "\n\n\t\tInvalid Age";

goto AGE;

}

break;

case 5:

cout << "\n\t\t Enter Address : ";

fflush(stdin);

cin.getline(address, 70);

break;

default:

cout << "\n\t\tInvalid choice .... Please Try Again";

getch();

break;

}

cout << "\n\n\t\tSuccessfully Modified Details Of Customer !";

getch();

file1.write((char \*)this, sizeof(lpg\_management));

found++;

}

file.read((char \*)this, sizeof(lpg\_management));

}

if (found == 0)

{

cout << "\n\n\t\tRecord Not Found!";

getch();

}

file1.close();

file.close();

remove("customer\_data.txt");

rename("record.txt", "customer\_data.txt");

}

}

// MODULE TO DELETE DATA OF CUSTOMER //

void lpg\_management::delete\_customer\_data()

{

system("cls");

string dadhaar;

int found = 0;

fstream file, file1;

file.open("customer\_data.txt", ios::in);

if (!file)

{

cout << "\n\t\t\tFile Not Found!";

file.close();

getch();

}

else

{

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\n\t\t-->> SEARCH DATA BY AADHAR NO. TO DELETE <<--";

cout << "\n\n\t\tEnter Aadhar No.: ";

cin >> dadhaar;

file1.open("record.txt", ios::app | ios::out);

file.read((char \*)this, sizeof(lpg\_management));

while (!file.eof())

{

if (dadhaar!=adhaar)

{

file1.write((char \*)this, sizeof(lpg\_management));

}

else

{

found++;

cout << "\n\n\t\t SUCCESSFULLY DELETED ";

}

file.read((char \*)this, sizeof(lpg\_management));

}

if (found == 0)

{

cout << "\n\n\t\tRecord Not Found!";

}

file1.close();

file.close();

remove("customer\_data.txt");

rename("record.txt", "customer\_data.txt");

}

}

// MODULE TO DISPLAY CUSTOMER WHO HAVE BOOKED //

void lpg\_management::show\_bookings()

{

fstream file;

string mob;

int count = 0;

file.open("book\_Data.txt", ios::in );

system("cls");

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\n\t\t-->> CHECK BOOKINGS BY MOBILE NUMBER <<--";

cout << "\n\n\t\tEnter Mobile Number : ";

cin >> mob;

if (!file)

{

cout << "\n\n\t\tFile Not Found!";

}

else

{

file.read((char \*)this, sizeof(lpg\_management));

}

while (!file.eof())

{

if (mob.compare(phone\_no) == 0)

{

count++;

}

else

{

cout << "\n\n\t\t Record Doesn't Exist !! \n";

system("pause");

}

cout << "\n\n\t\t\t\t\t| CHECKING | \n\t\t\t\t\t ";

for (int a = 1; a < 5; a++)

{

Sleep(600);

cout << "...";

}

if (mob.compare(phone\_no) == 0)

{

cout << "\n\n\t\tGAS BOOKED ";

}

else

{

cout << "\n\n\t\tNOT BOOKED";

}

file.read((char \*)this, sizeof(lpg\_management));

}

if (count == 0)

{

cout << "\n\t\tRecord Not Found!";

}

file.close();

cout << "\n\n\t\tPress Any Key To Continue !";

getch();

}

// MODULE TO DISPLAY COMPLAINTS OF THE CUSTOMERS //

void lpg\_management::show\_complaints()

{

system("cls");

fstream file;

file.open("complaints.txt", ios::in );

int count = 0;

int date,month,year;

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t-->> Enter Date To Search Complaints <<--\n";

J:

cout << "\n\t\t Enter Date : ";

cin >> date;

if( date > 31)

{

cout << "\n\t\tInvalid Date ";

goto J;

}

K:

cout << "\n\t\t Enter Month : ";

cin >> month;

if( month > 12)

{

cout << "\n\t\tInvalid Month ";

goto K;

}

L:

cout << "\n\t\t Enter Year : ";

cin >> year;

if( year < 2022 )

{

cout << "\n\t\tInvalid Year ";

goto L;

}

if (!file)

{

cout << "\nFile Not Found!";

}

else

{

file.read((char \*)this, sizeof(lpg\_management));

}

while (!file.eof())

{

if (date == dd && month == mm && year == yy)

{

count++;

show\_issue();

}

file.read((char \*)this, sizeof(lpg\_management));

}

if (count == 0)

{

cout << "\n\n\t\tRecord Not Found!";

}

file.close();

}

// MODULE TO READ DATA FROM FILE (USED IN SHOW COMPLAINT MODULE) //

void lpg\_management::show\_issue()

{

system("cls");

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\n\t\t Date : " << dd << "/" << mm << "/" << yy ;

cout << "\n\n\t\t The Complaints Are Shown Below !";

cout << "\n\n\t\t ===> " << complaint << endl <<"\n\n\n\t";

system("pause");

}

// ===> CUSTOMER MODULES <=== //

// MODULE FOR CUSTOMER TO BOOK GAS //

void lpg\_management::booking()

{

system("cls");

display\_data();

char a,b;

fstream file;

cout << "\n\n\t\tDo you want to Book Gas ? [y/n] => ";

cin >> a;

if (a == 'y' || a == 'Y')

{

cout << "\n\t\tOnly 1 Gas Can be Booked at a Time !";

}

else

{

customer();

}

cout << "\n\n\t\tDo You Want to Continue ? [y/n] => ";

cin >> b;

if (b == 'y' || b == 'Y')

{

cout << "\n\n\t\t Your Gas is Booked ..!";

}

P:

cout << "\n\n\t\tEnter your Registered Mobile Number for Further Updates --> ";

cin >> phone\_no;

if (phone\_no.length() != 10)

{

cout << "\n\t\tInvalid Phone Number";

goto P;

}

else

{

// The First letter should not start with 0 or 1

if (phone\_no[0] == '0' || phone\_no[0] == '1')

{

cout << "\n\t\tInvalid Phone Number";

goto P;

}

else

{

cout << "\n\t\tValid Phone Number";

}

}

file.open("book\_Data.txt", ios::app );

file.write((char \*)this, sizeof(lpg\_management));

cout << "\n\n\t\tYOU HAVE SUCCESSFULLY BOOKED !";

cout << "\n\n\t\tPress Any Key To Continue !";

file.close();

getch();

}

// MODULE FOR CUSTOMER TO REPORT THEIR COMPLAINTS //

void lpg\_management::report\_issue()

{

fstream file;

system("cls");

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\t\t\t\t \* LPG MANAGEMENT SYSTEM \*";

cout << "\n\t\t\t\t-----------------------------------------";

cout << "\n\n\t\t-->> REPORT ISSUE <<--";

D:

cout << "\n\n\t\tEnter Today's Date : ";

cin >> dd ;

if( dd > 31)

{

cout << "\n\t\tInvalid Date ";

goto D;

}

M:

cout << "\n\t\tEnter Current Month : ";

cin >> mm ;

if( mm > 12)

{

cout << "\n\t\tInvalid Month ";

goto M;

}

Y:

cout << "\n\t\tEnter Current Year : ";

cin >> yy;

if( yy < 2022)

{

cout << "\n\t\tInvalid Year ";

goto Y;

}

cout << "\n\t\t Entered Date : " << dd << "/" << mm << "/" << yy ;

cout <<"\n\n\t\tWrite Your Complaint : ";

cin.ignore();

cin.getline(complaint,1000);

file.open("complaints.txt", ios::app );

file.write((char \*)this, sizeof(lpg\_management));

cout << "\n\n\t\tWe Are Sorry to hear about this!!!\n\t\tYour complaint has been registered successfully and We shall revert you soon !!";

cout << "\n\n\t\tPress Any Key To Continue ! ";

file.close();

}

int main()

{

system("color E");

lpg\_management system;

system.menu();

}

**TESTING & VALIDATIONS**

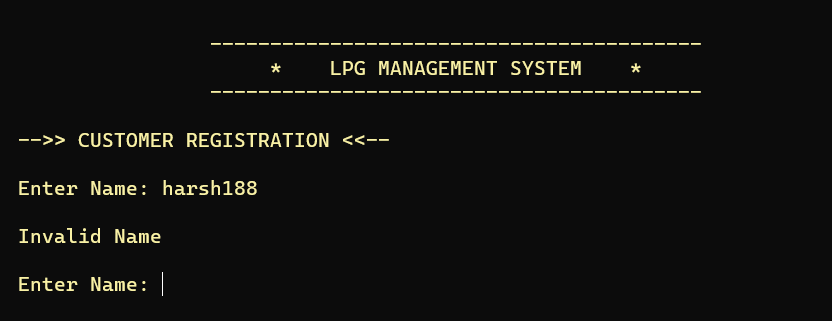
There are several testing and validations used in the code for different purposes.

**Test and validation 1: For inputting correct password of Admin.**

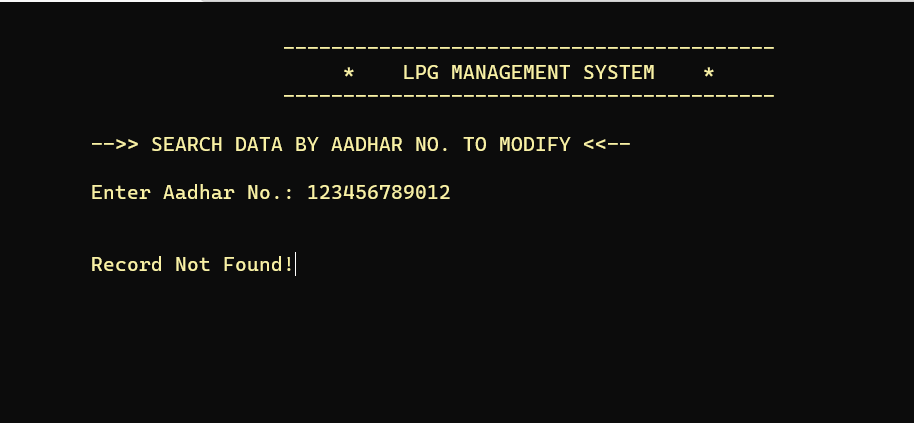


We have to provide the correct password to enter into the Admin and Customer respectively.

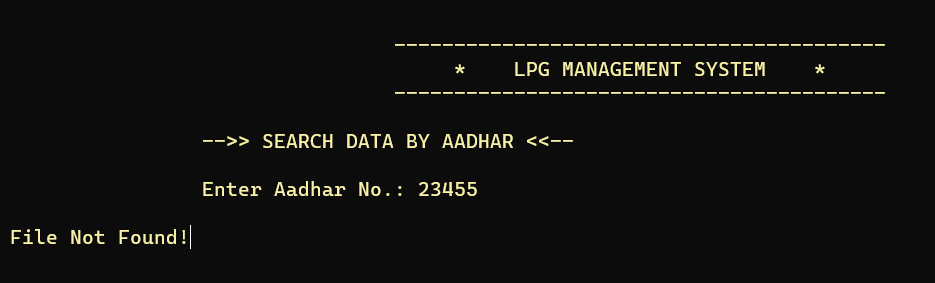
**Test and validation 2: For inputting correct data type value and refraining user to enter character or string value in Alphabets.**

****

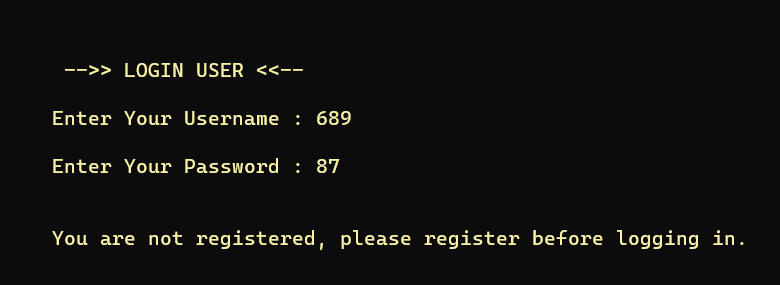
**Test and validation 3: For searching product in a file.**

****

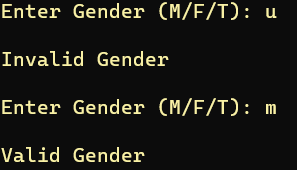
**Test and validation 4: For seeing if the file exist or not.**

****

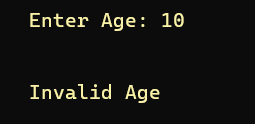
**Test and validation 5: For seeing if the customer is registered.**

****

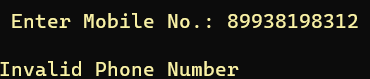
**Test and validation 6: Checking if Gender is inputted wrong.**

****

**Test and validation 7: Checking if Age is inputted wrong.**

****

**Test and validation 8: Checking if Number is inputted is wrong.**

****

**INPUT / OUTPUT SCREENS**

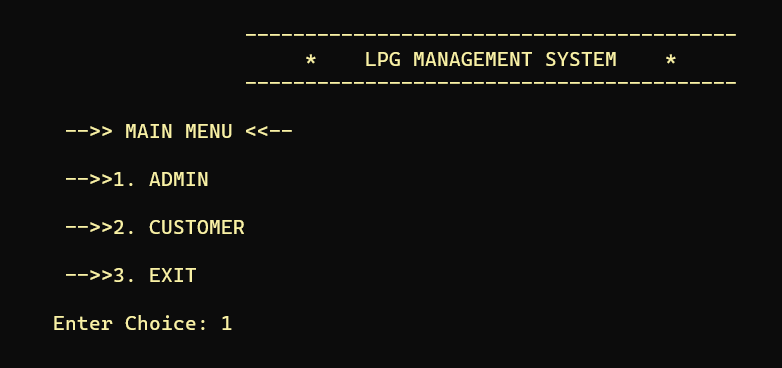


Fig 1. Main Menu

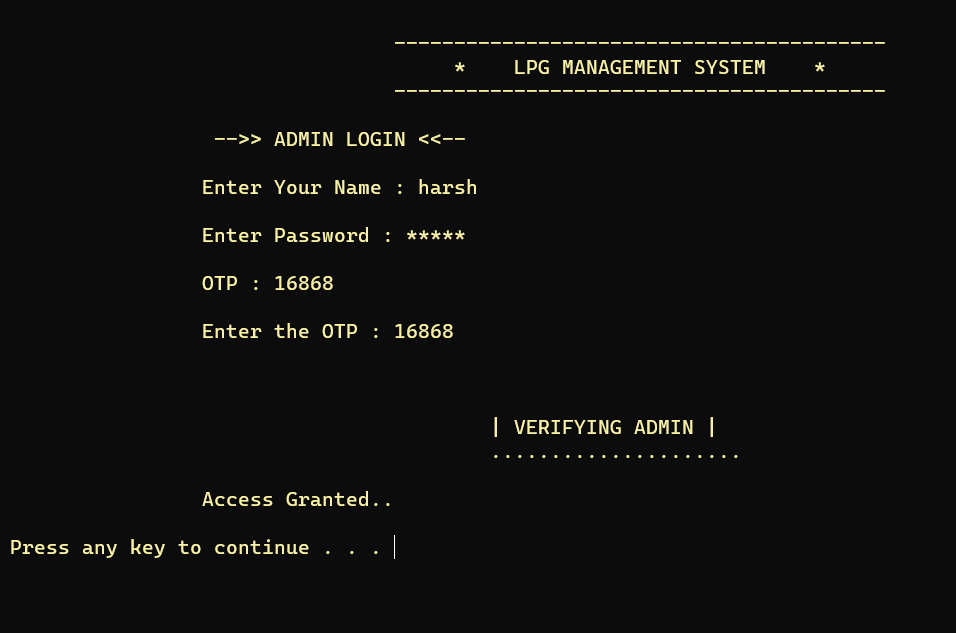


Fig 1.1 Admin Login

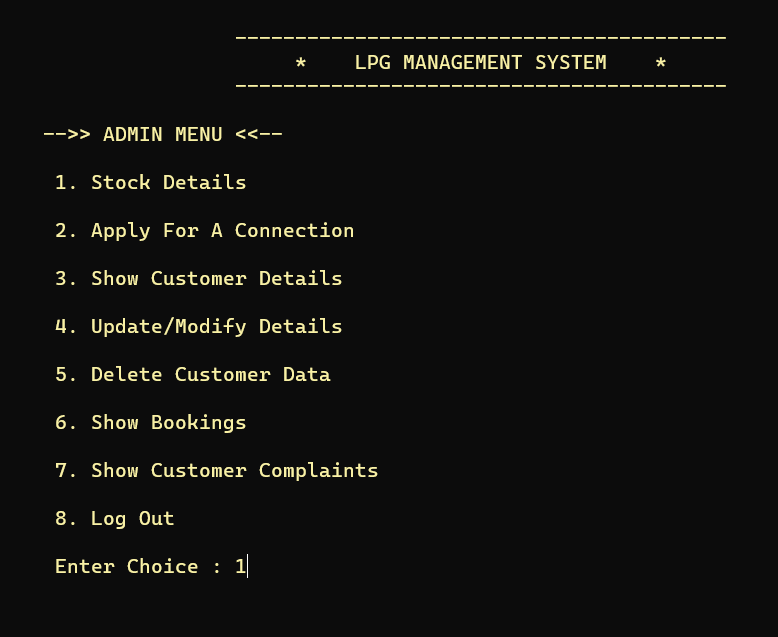


Fig 1.1 Admin Menu

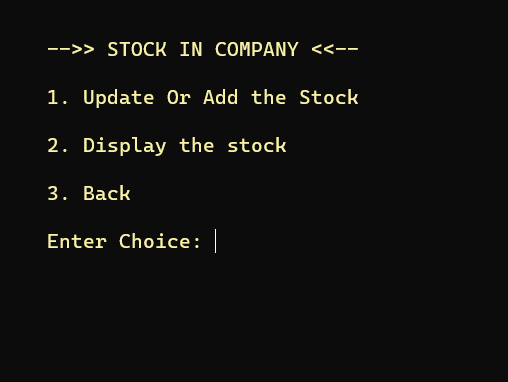


Fig 1.1.1 Stock

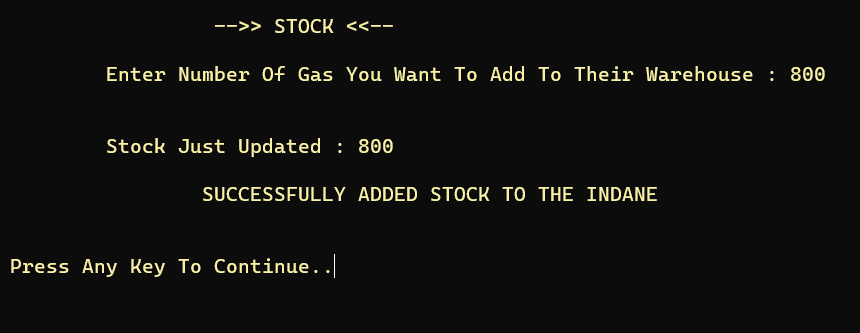


Fig 1.1.1 Stock Add

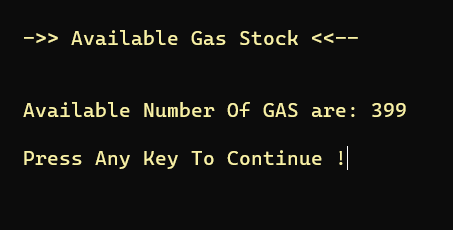


Fig 1.1.1 Stock Display

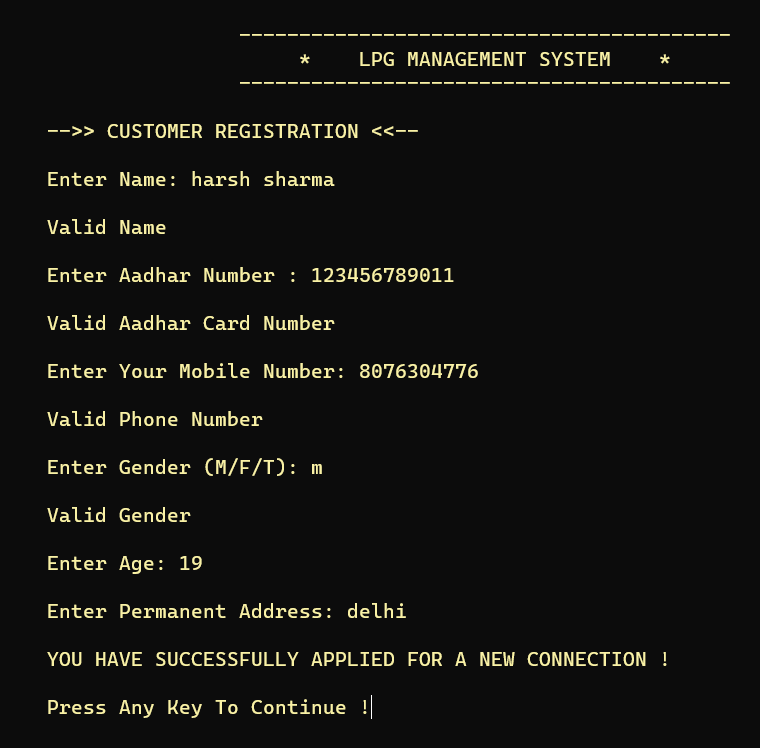


Fig 1.1.2 Customer Registration

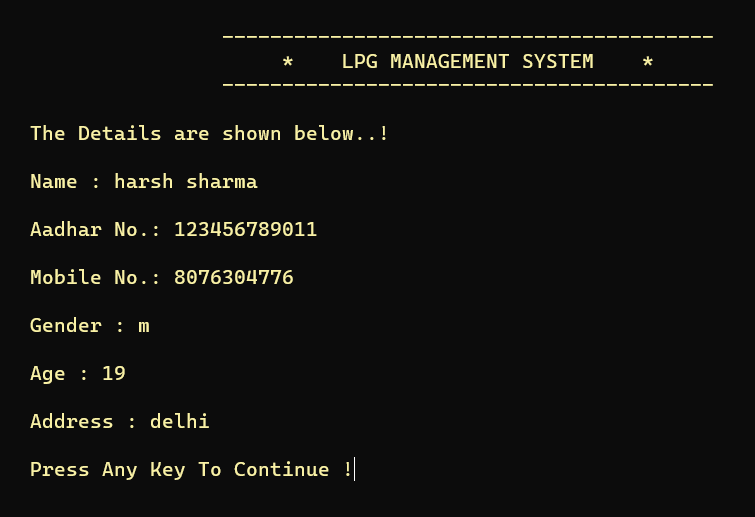


Fig 1.1.3 Display Information

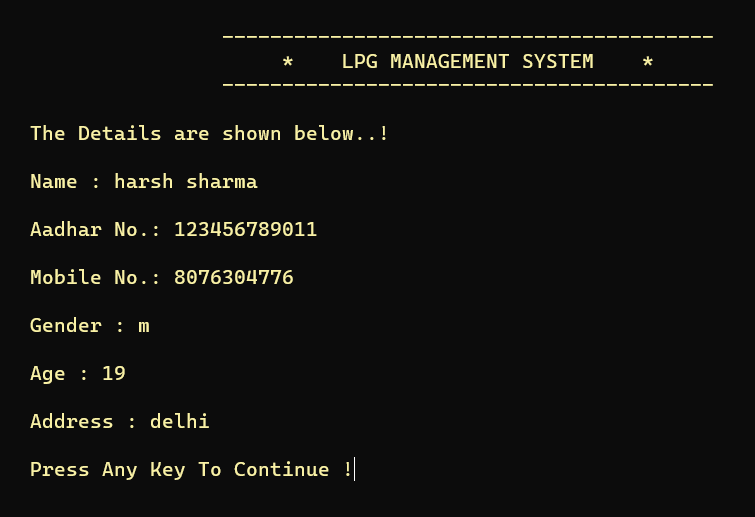


Fig 1.1.4 Update Details

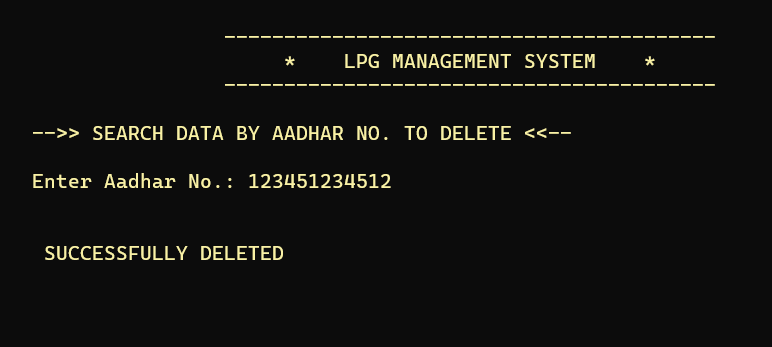


Fig 1.1.5 Delete Data

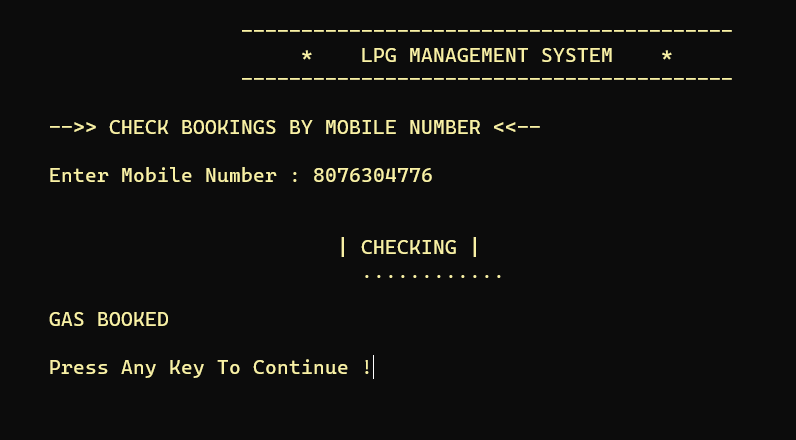


Fig 1.1.6 Show Bookings

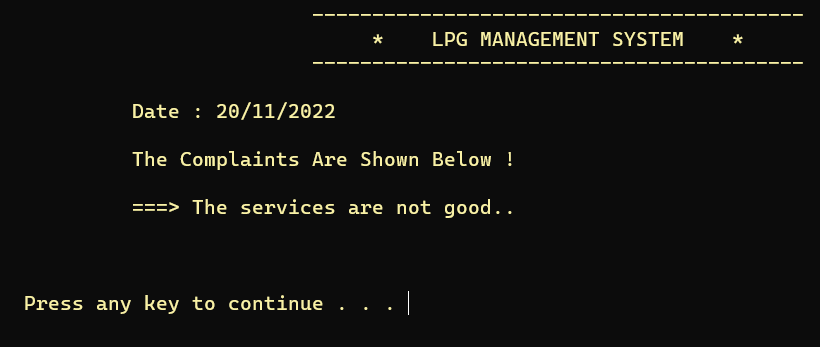


Fig 1.1.7 Show Complaints

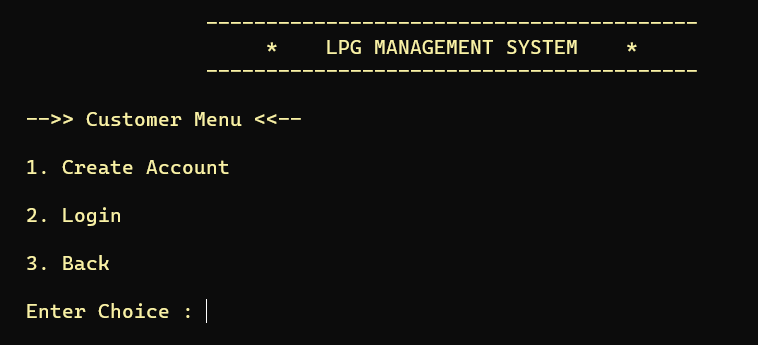


Fig 2. Customer Panel

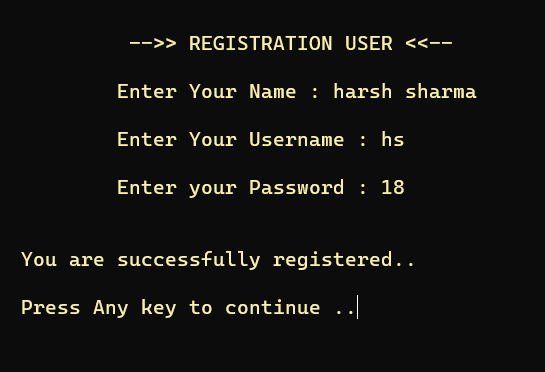


Fig 2.1 Customer Account Creation

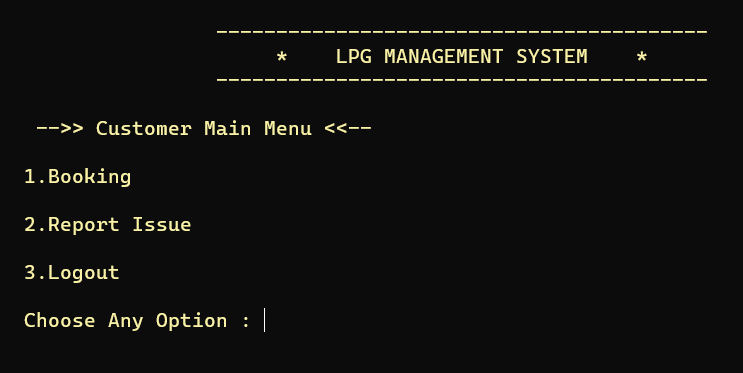


Fig 2.2 Customer Main Menu

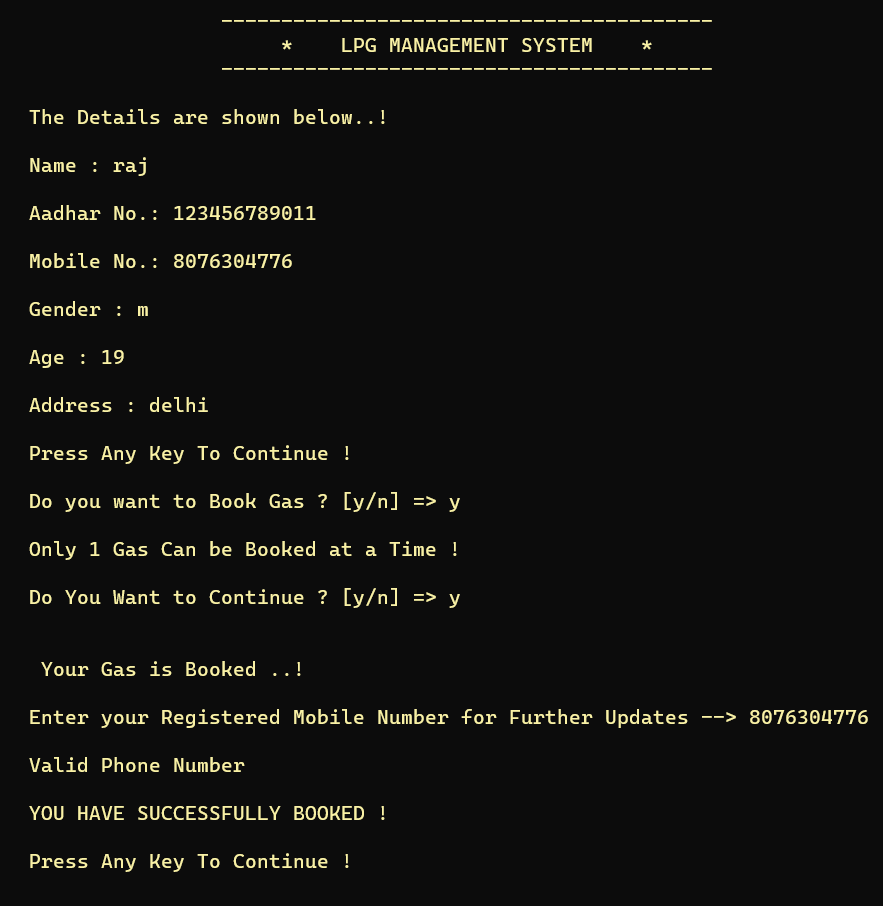


Fig 2.2.1 Customer Booking

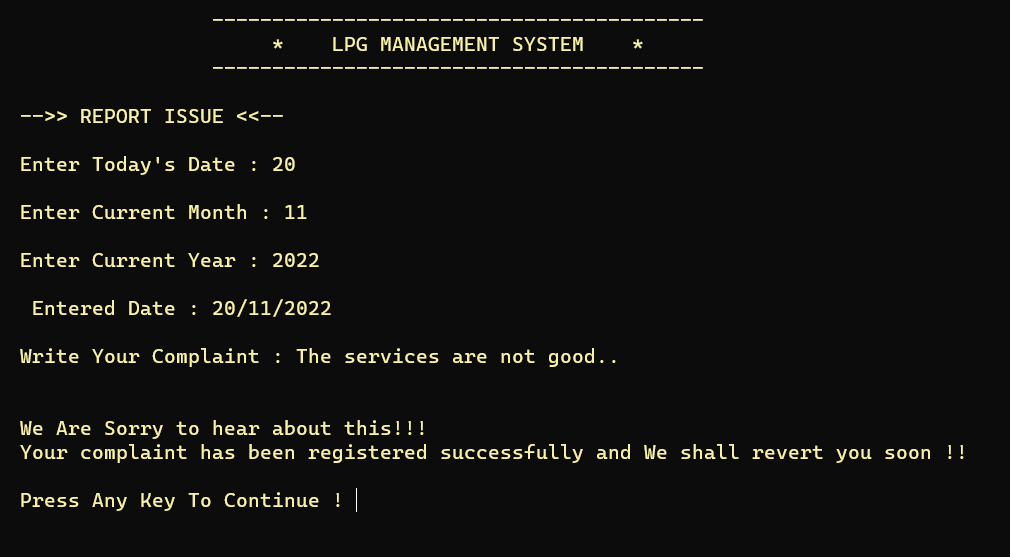


Fig 2.2.2 Customer Complaint

**LIMITATIONS**

* Anyone can manipulate data according to their own wish.
* Technical glitches.
* It does not have GUI interface.
* Lack of data security because data is stored in files not in database.
* There is no option for data backup.

**FUTURE APPLICATIONS OF THE PROJECT**

* This could be use in customer records to generate customer’s reports .
* It will help to maintain the records of the customer properly.
* In future we can connect this project with cloud servers which provide better data security and good data management.