CS3101 Project: Flight Reservation System

Ronway Airlines

Ronit Bhuyan
22MS025

Sagnik Seth
22MS026

Aviyank Aryan
22MS030

Contents

1 Introduction

This project is made as part of our CS3101 course. The project implements a flight reservation system which is named as Ronway Airlines. It is written using C and compiled using gcc.

The system integrates various concepts in C like structures, pointers, file handling, etc to provide a holistic prototype interface for handling flight systems.

The system can be accessed in two modes: User and Admin.

As a User, one can search flights according to the source,destination, date and time. The user can also book or cancel a booking, view the booking history and also chat with the chatbot named Bandhu on basic questions regarding booking, cancellation and other general queries.

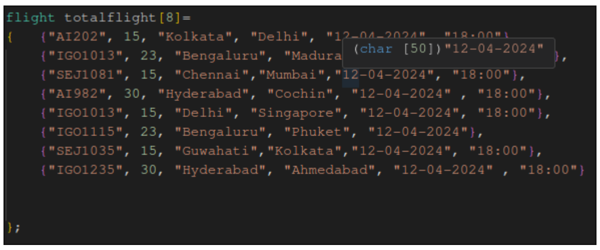
As an Admin, one can add flights, view all the flights available, change the details of a flight and delete a flight altogether.

2 Building and Dependencies

The project is built using C and compiled using gcc. To provide a more aesthetic interface, the project uses the *ncurse.h* library which has provisions like output positioning at specific coordinates, output colouring, etc. Thus, the *ncurse.h* library should be installed to run the project.

To provide an initial list of flights, we have created a C file *flight_list.c* (located in the Seat Matrix directory) which contains the details of the flights and stores it in a file Airlist.txt. Thus, it should be compiled first using

```
gcc -o AirList flight_list.c
```



```
flight totalflight[8]=
{   {"AI202", 15, "Kolkata", "Delhi", "12-04-2024", "18:00"},
    {"IGO1013", 23, "Bengaluru", "Madura (char [50])"12-04-2024" },
    {"SEJ1081", 15, "Chennai","Mumbai","12-04-2024", "18:00"},
    {"AI982", 30, "Hyderabad", "Cochin", "12-04-2024" , "18:00"},
    {"IGO1013", 15, "Delhi", "Singapore", "12-04-2024", "18:00"},
    {"IGO1115", 23, "Bengaluru", "Phuket", "12-04-2024"},
    {"SEJ1035", 15, "Guwahati","Kolkata","12-04-2024", "18:00"},
    {"IGO1235", 30, "Hyderabad", "Ahmedabad", "12-04-2024" , "18:00"}



};
```

Figure 1: Initial List of flights

The main project file is *Flight_Reservation.c* which should be compiled using:

```
gcc -o Flight_out Flight_Reservation.c -lncurses
```

The *-lncurses* flag is neccessary since we are using the *ncurse.h* library.

3 Main Structure

The figure below provides a schematic diagram of the main structure of the program. The arrows indicate the logical flow of the system.

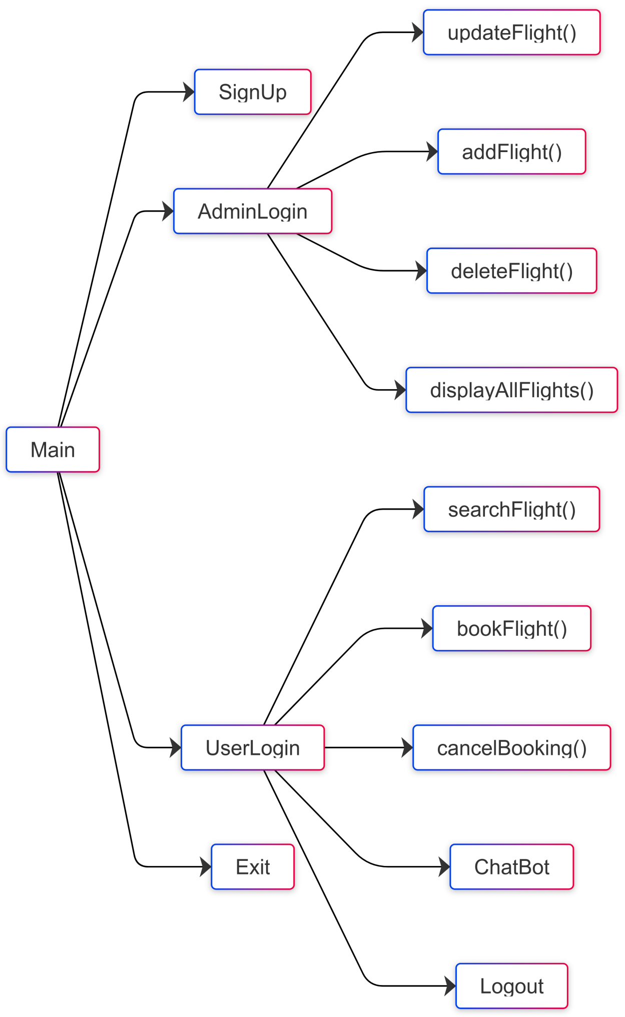


Figure 2: Main structure of the Program

Starting from the main page, we can either signup (if new user/admin) or login. Accordinly as user/admin, we have the accessibility of different utilitites.

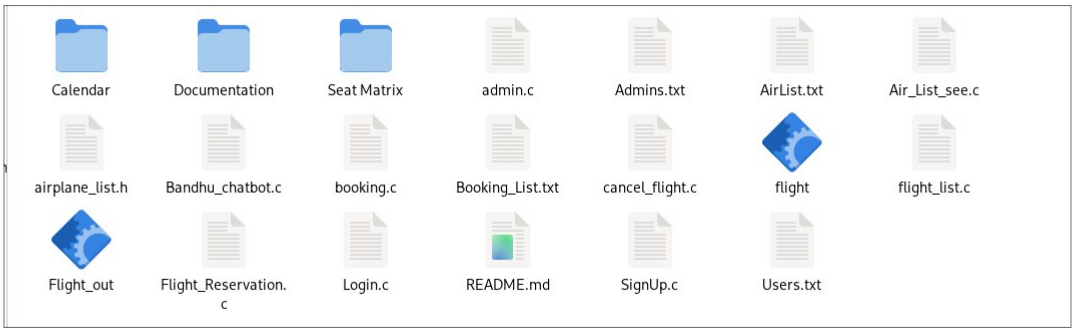


Figure 3: Main contents of the directory

4 Documentation

4.1 General Functions

Functions in Login.c file:

- `int admin_login()`: This function is used to login as an admin. The admin is asked to enter the username and password. If the username and password match, the admin is redirected to the admin utilities page.

- `int user_login()`: This function is used to login as a user. The user is asked to enter the username and password. If the username and password match, the user is redirected to the user utilities page.

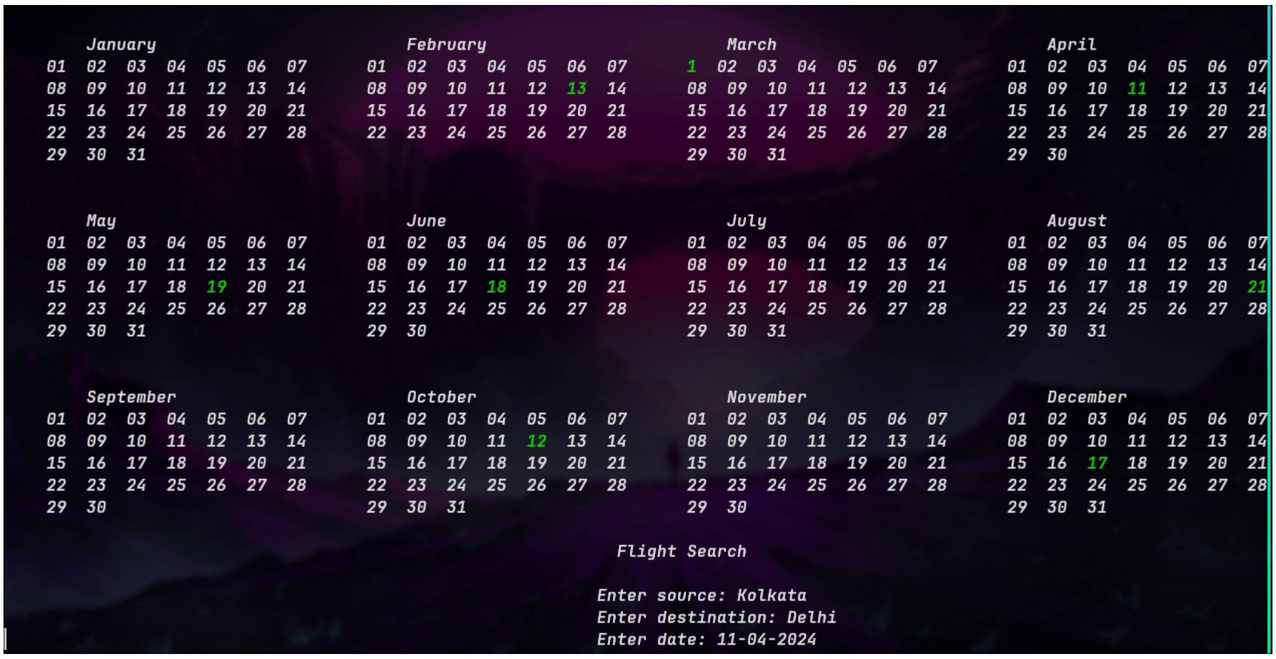
Functions in SignUp.c:

`int signup()`: This function is used to signup as a user. The user is asked to enter the username and password. The function then adds the user to the database.

4.2 Functions for User Utilities

The main Functions used in the User Utilities (contained in the booking.c file) are as follows:

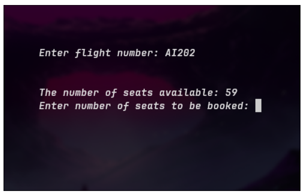
- `int searchFlight()`: This function is used to search for flights according to the source, destination, date. The function then searches for the flights according to the user's input and displays the flights available.
  The function first displays a calendar where the dates available are highlighted in green. Then the user is asked to select the source and destination of his choice and the date of travel. If a flight is found, the user is redirected to the booking page.



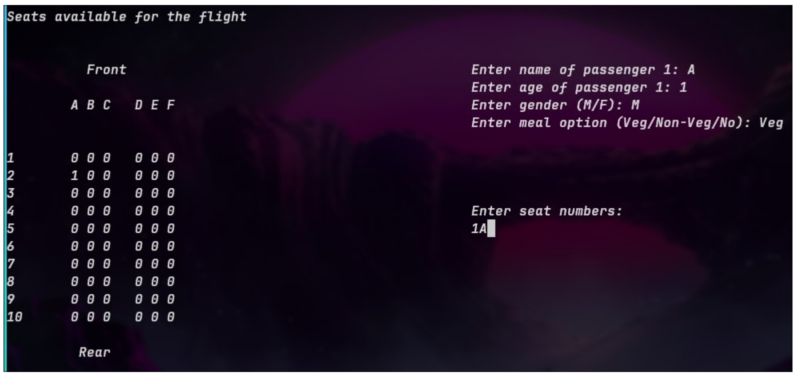
```
    January                       February                      March                         April
01  02  03  04  05  06  07    01  02  03  04  05  06  07    1   02  03  04  05  06  07    01  02  03  04  05  06  07
08  09  10  11  12  13  14    08  09  10  11  12  13  14    08  09  10  11  12  13  14    08  09  10  11  12  13  14
15  16  17  18  19  20  21    15  16  17  18  19  20  21    15  16  17  18  19  20  21    15  16  17  18  19  20  21
22  23  24  25  26  27  28    22  23  24  25  26  27  28    22  23  24  25  26  27  28    22  23  24  25  26  27  28
29  30  31                                                  29  30  31                    29  30

    May                           June                          July                          August
01  02  03  04  05  06  07    01  02  03  04  05  06  07    01  02  03  04  05  06  07    01  02  03  04  05  06  07
08  09  10  11  12  13  14    08  09  10  11  12  13  14    08  09  10  11  12  13  14    08  09  10  11  12  13  14
15  16  17  18  19  20  21    15  16  17  18  19  20  21    15  16  17  18  19  20  21    15  16  17  18  19  20  21
22  23  24  25  26  27  28    22  23  24  25  26  27  28    22  23  24  25  26  27  28    22  23  24  25  26  27  28
29  30  31                    29  30                        29  30  31                    29  30  31

    September                     October                       November                      December
01  02  03  04  05  06  07    01  02  03  04  05  06  07    01  02  03  04  05  06  07    01  02  03  04  05  06  07
08  09  10  11  12  13  14    08  09  10  11  12  13  14    08  09  10  11  12  13  14    08  09  10  11  12  13  14
15  16  17  18  19  20  21    15  16  17  18  19  20  21    15  16  17  18  19  20  21    15  16  17  18  19  20  21
22  23  24  25  26  27  28    22  23  24  25  26  27  28    22  23  24  25  26  27  28    22  23  24  25  26  27  28
29  30                        29  30  31                    29  30                        29  30  31

                                                       Flight Search

                                                     Enter source: Kolkata
                                                     Enter destination: Delhi
                                                     Enter date: 11-04-2024
```

Figure 4: Contents when search function is called

- `int bookFlight(char* flightnum, char* usr)`: This function is used to book a flight. The user is asked to enter the flight number and the number of seats to be booked. The function then books the flight and updates the booking history. The final bill is then generated and booking is successful



```
Enter flight number: AI202


The number of seats available: 59
Enter number of seats to be booked:
```



```
Seats available for the flight


         Front                                                      Enter name of passenger 1: A
                                                                    Enter age of passenger 1: 1
       A B C   D E F                                                Enter gender (M/F): M
                                                                    Enter meal option (Veg/Non-Veg/No): Veg

1      0 0 0   0 0 0
2      1 0 0   0 0 0
3      0 0 0   0 0 0
4      0 0 0   0 0 0                                                Enter seat numbers:
5      0 0 0   0 0 0                                                1A
6      0 0 0   0 0 0
7      0 0 0   0 0 0
8      0 0 0   0 0 0
9      0 0 0   0 0 0
10     0 0 0   0 0 0

        Rear
```

Figure 5: Contents when booking function is called

- `int cancelBooking(char* tick)`: This function is used to cancel a booking. The user is asked to enter the booking ID and the function then cancels the booking. If a record is found, the user is asked for confirmation of cancellation and the booking is subsequently cancelled.

- `void view_booking_history()`: This function is used to view the booking history. The function displays the booking record of the user.

- `void update_seat_matrix(char* filename, char* airnum, char seats[][10], int size, char* mode)`: This function is used to update the seat matrix. The function takes the filename, flight number, the list of seats, the number of seats, the size of the matrix and the mode as input ("add" or "remove") and updates the seat matrix.

4.3 Functions for Admin Utilities

The main Functions used in the Admin Utilities (contained in the admin.c file) are as follows:

- `void addFlight()`: This function is used to add a flight. The admin is asked to enter the flight details and the function then adds the flight to the database.

- `void updateFlight()`: This function is used to update the details of a flight. The admin is asked to enter the flight number and the function then updates the details of the flight.

- `void deleteFlight()`: This function is used to delete a flight. The admin is asked to enter the flight number and the function then deletes the flight from the database.

- `void displayAllFlights()`: This function is used to display all the flights available. The function displays all the flights available in the database.

4.4 Functions for Bandhu–the chatbot

The code for the chatbot is included in the file `Bandhu_chatbot.c` . The main Functions used in the chatbot are as follows:

- `char *lowering(char *str)`: This function is used to convert a string to lowercase. The function takes a string pointer as input and returns the string in lowercase.
- `const char *keyword_match(const char *keywords[], char *response, int size)`: This function is used to match the user's input with the keywords array. The function takes the keywords, the user's input and the size of the array of keywords array as input and returns the matched keyword from the array.
- `int indexOf(const char *str[], const char *substr, int size)`: This function is used to find the index of a string in an array of strings. The function takes the array of strings, the string and the size of the array as input and returns the index of the string in the array.
- `int chat(char *name)`: This function is used to chat with the chatbot. The function takes the user's name as input and then provides responses according to user's input.
  Various arrays of keywords of different categories like booking, cancellation, greeting, etc. are used in the chatbot. The user provides a response and the function finds whether any word from the various keywords array is a substring of the user's input. If a keyword is found, the chatbot provides a response accordingly.

4.5 Structures:

We have used different structures to construct the project. The main structures used are:

- For defining flight

```
typedef struct{
char flightnum[50];
int seatsFree;
char source[100];
char destination [100];
char date[50];
char time[50];
double adult_price;
double child_price;
double infant_price;

} flight;
```

- For defining a person who is booking the flight

```
typedef struct {
char username[50];
char name[50];
int age;
char gender[10];
char type[10];// determines whether infant or child or adult
char meal[20];
char pnr[20];
char seat[20];
char flightnum[20];
double price;
} BOOKER;
```

- For defining the user

```
typedef struct user{
char name[50];
char pwd[50];
}USER;
```

- For defining the admin

```
typedef struct admin{
char name[50];
char pwd[50];
char id[50];
}ADMIN;
```

5 Limitations and Future Prospects

The following are the limitations on which we have to work on:

- Adding flights on the same date(but of different months) is not reflected in the calendar due to some array indexing issues.
- Choosing the 10th row of the seat matrix provides an error because of double digit issues.
- The Chatbox is limited to a few soecific queries and responses containing multiple keywords cant be segregated.

6 Contributions

The project is a collaborative effort of all the members. We have tried to divide our work equally amongst the members Sagnik and Ronit while Aviyank has mainly worked on the chatbot. The contributions from each member are as follows:

- **Ronit Bhuyan** : Worked in the implementation of the admin utilities,like adding , updating and deleting a flight. In the user utilities section, has partial contributions to searching and booking flights. In addition to that, helped in alignment of text and user outputs as well as the documentations.
- **Sagnik Seth** : Has written the code for the signup, login page, partially done the booking tickets for user part, worked on the seat matrix printing, viewing the record of the user, cancelling user tickets, for overall presentation of the interface like printing the calendar and the ASCII arts, partially done the chatbot part, and the documentation.
- **Aviyank Aryan**: Helped in the implementation of the chatbot and completion of the documentations.