

booking.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
```

Go to the source code of this file.

Data Structures

```
struct ticket
    Structure to store the bookings done. More...
```

Enumerations

```
enum bool { false, true }
```

Functions

```
void increaseCapacity ()
    Increases the capacity of the structure. More...

void rewriteFile ()
    Rewrites file containing bookings. More...

void readFile ()
    Reads file containing the bookings. More...

void startBooking ()
    Initiates booking. More...

void gen_reference (char *s, int len)
    Generates booking reference number. More...

bool displayFlight (char dep[], char des[])
    Display flights for the route. More...

void feedDetails (int choice)
    Extract details of the flight selected. More...

int count (int n)
    Counts the number of digits in a number. More...

void remove_newline (char *a)
    Remove trailing new line character. More...

void updateSeats (char *fn, int new_seatAvail)
    Modifies the seat availability. More...

bool isSeatAvailable (int n)
    Checks for seat availability. More...

void displayBooking (int x)
    Displays all the details of a booking. More...

void updateFare (char *fn, int new_fare)
    Modifies the fare. More...

void newBooking ()
    Records a new booking. More...

void change_fare ()
    Queries about flight fare change. More...

int getIndex (char *br)
    Finds the index of the booking. More...

void findBooking ()
    Queries about the booking. More...

void deleteBooking ()
    Deletes a booking. More...

void addFlight ()
    Adds a new flight. More...

void change_seatAvail ()
    Queries about the flight seat availability change. More...

char * encrypt (char *array)
    Encrypts a string. More...

bool checkPass ()
    Queries for password. More...

void change_password ()
    Changes administrator password. More...

int nTicket =-1
    Number of tickets booked. More...

int capacity =1
    Number of bookings that the structure array can store. More...

ticket * booking
    Pointer to the structure array containing bookings. More...

FILE * fo
    File variable for flights.txt. More...
```

FILE * **fi**
File variable for availableFlights.txt. More...

Detailed Description

Note
DSC Project

Author
Anirudh Singh & Apeksha K.

Version
1.1

Date
10/11/2014

Definition in file **booking.h**.

Enumeration Type Documentation

enum bool

Enumerator

false

true

Definition at line **29** of file **booking.h**.

Function Documentation

void addFlight ()

Adds a new flight.

Adds a new flight to the database, entering all the details and storing it in the file flights.txt

Definition at line **510** of file **booking.h**.

References **fi**, and **remove_newline()**.

Referenced by **main()**.

void change_fare ()

Queries about flight fare change.

Takes input of flight number and the new fare from the administrator

Definition at line **436** of file **booking.h**.

References **updateFare()**.

Referenced by **main()**.

void change_password ()

Changes administrator password.

Asks the administrator to enter a new password, encrypts it and stores it in the file password.txt

Definition at line **603** of file **booking.h**.

References **encrypt()**.

Referenced by **main()**.

void change_seatAvail ()

Queries about the flight seat availability change.

Takes input of flight number and the new seat availability from the administrator

Definition at line **546** of file **booking.h**.

References **updateSeats()**.

Referenced by **main()**.

bool checkPass ()

Queries for password.

Matches the password entered with the password stored in file

Returns

True if passwords match, false otherwise

Definition at line **579** of file **booking.h**.

References **encrypt()**.

Referenced by **main()**.

int count (int n)

Counts the number of digits in a number.

Parameters

n Number whose digits need to be counted

Returns

Number of digits in the number passed as argument

Definition at line **232** of file **booking.h**.

void deleteBooking ()

Deletes a booking.

Changes the active attribute to false, rendering the booking deleted

Definition at line **492** of file **booking.h**.

References **ticket::active**, **getIndex()**, and **rewriteFile()**.

Referenced by **main()**.

void displayBooking (int x)

Displays all the details of a booking.

Displays the details of the booking at the specified index in the structure array

Parameters

x Index of the booking present in the structure array

Definition at line **315** of file **booking.h**.

Referenced by **findBooking()**, and **newBooking()**.

```
bool displayFlight ( char dep[],
                    char des[]
                  )
```

Display flights for the route.

Displays all the flights available flying from the entered departure city to the destination city

Parameters

dep Name of Departure City

des Name of Destination City

Returns

True if some flights are found False if no flights are found

Definition at line **165** of file **booking.h**.

References **fi**, and **fo**.

Referenced by **newBooking()**.

```
char* encrypt ( char * array )
```

Encrypts a string.

Encrypts the string using XOR operations on its characters

Parameters

array String to be encrypted

Returns

Encrypted string

Definition at line **563** of file **booking.h**.

Referenced by **change_password()**, and **checkPass()**.

```
void feedDetails ( int choice )
```

Extract details of the flight selected.

Copies details of the booking to the structure variable according to the choice of the flight entered by the user

Parameters

choice Index representing the flight selected by the user

Definition at line **201** of file **booking.h**.

References **ticket::arr**, **ticket::dep**, **ticket::fare**, **fo**, and **nTicket**.

Referenced by **newBooking()**.

```
void findBooking ( )
```

Queries about the booking.

Takes input of the booking reference number from the user, if found displays the booking otherwise informs the user that the booking was not found

Definition at line **471** of file **booking.h**.

References **displayBooking()**, and **getIndex()**.

Referenced by **main()**.

```
void gen_reference ( char * s,
                    int   len
                  )
```

Generates booking reference number.

Generates a random alphanumeric reference number associated with each booking the last digit representing the number of tickets booked till now

Parameters

- s** String to store the reference number
- len** Length of the required reference number

Definition at line 141 of file **booking.h**.

References **nTicket**.

Referenced by **newBooking()**.

```
int getIndex ( char * br )
```

Finds the index of the booking.

Traverses through the structure array and locates the index of the booking

Parameters

- br** Booking reference number to search for

Returns

Index of the booking in the array corresponding to the booking reference number

Definition at line 453 of file **booking.h**.

References **nTicket**.

Referenced by **deleteBooking()**, and **findBooking()**.

```
void increaseCapacity ( )
```

Increases the capacity of the structure.

Increases capacity of the structure if the array of structures is completely filled.

Definition at line 80 of file **booking.h**.

References **capacity**, and **nTicket**.

Referenced by **newBooking()**, and **readFile()**.

```
bool isSeatAvailable ( int n )
```

Checks for seat availability.

Checks if the number of seats available is 0 for the flight selected by the user

Parameters

- n** Index of flight selected by the user

Returns

True if seats available false otherwise

Definition at line 291 of file **booking.h**.

References **fo**.

```
void newBooking ( )
```

Records a new booking.

Makes a new booking by taking inputs from the user and adds it to the structure array

Definition at line 378 of file **booking.h**.

References **ticket::active**, **ticket::arr**, **ticket::dep**, **displayBooking()**, **displayFlight()**, **feedDetails()**, **gen_reference()**, **increaseCapacity()**, **nTicket**, **remove_newline()**, and **updateSeats()**.

Referenced by **main()**.

void readFile ()

Reads file containing the bookings.

Opens bookings.dat and copies the structure variables present into structure array

Definition at line **109** of file **booking.h**.

References **increaseCapacity()**, and **nTicket**.

Referenced by **startBooking()**.

void remove_newline (char * a)

Remove trailing new line character.

Replaces the trailing new line character in the string (added by fgets()) with a null character

Definition at line **247** of file **booking.h**.

Referenced by **addFlight()**, and **newBooking()**.

void rewriteFile ()

Rewrites file containing bookings.

Rewrites the file containing the bookings after deleting a booking (active changed to false)

Definition at line **94** of file **booking.h**.

References **nTicket**.

Referenced by **deleteBooking()**.

void startBooking ()

Initiates booking.

Called in the beginning to open the bookings.dat file and allocate memory for a structure object

Definition at line **128** of file **booking.h**.

References **capacity**, and **readFile()**.

Referenced by **main()**.

**void updateFare (char * fn,
 int new_fare
)**

Modifies the fare.

Modifies the fare of the entered flight, in the file flights.txt

Parameters

fn Flight Number of the flight whose fare needs to be changed

new_fare New fare for the selected flight

Definition at line **350** of file **booking.h**.

References **fi**.

Referenced by **change_fare()**.

```
void updateSeats ( char * fn,
                  int   new_seatAvail
                  )
```

Modifies the seat availability.

Modifies the number of seats of the entered flight, in the file flights.txt

Parameters

fn Flight Number of the flight whose seat availability needs to be changed

new_seatAvail New number of seats for the selected flight

Definition at line 259 of file **booking.h**.

References **fi**.

Referenced by **change_seatAvail()**, and **newBooking()**.

Variable Documentation

ticket* booking

Pointer to the structure array containing bookings.

Definition at line 69 of file **booking.h**.

int capacity =1

Number of bookings that the structure array can store.

Definition at line 33 of file **booking.h**.

Referenced by **increaseCapacity()**, and **startBooking()**.

FILE* fi

File variable for availableFlights.txt.

Definition at line 73 of file **booking.h**.

Referenced by **addFlight()**, **displayFlight()**, **updateFare()**, and **updateSeats()**.

FILE* fo

File variable for flights.txt.

Definition at line 71 of file **booking.h**.

Referenced by **displayFlight()**, **feedDetails()**, and **isSeatAvailable()**.

int nTicket =-1

Number of tickets booked.

Definition at line 31 of file **booking.h**.

Referenced by **feedDetails()**, **gen_reference()**, **getIndex()**, **increaseCapacity()**, **newBooking()**, **readFile()**, and **rewriteFile()**.