Contents

[1. Introduction: 2](#_Toc426636280)

[2. Menu: (Common responsibility) 2](#_Toc426636281)

[3. Search a Flight (Trainee 1) 3](#_Toc426636282)

[4. Book a Flight (Trainee 2): 4](#_Toc426636283)

[5. Search a Hotel (Trainee 3): 5](#_Toc426636284)

[6. Book a Hotel (Trainee 4) 7](#_Toc426636285)

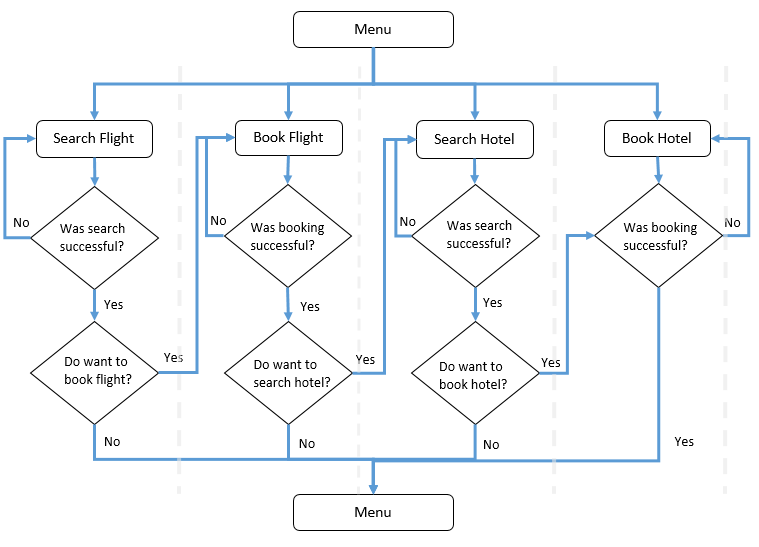
[7. Advanced Search (Common Module): 9](#_Toc426636286)

[Appendix: 9](#_Toc426636287)

# Introduction:

Integrated Flight services (IFS) is a system which allows user to book flights and hotels. The user can book only flights or only hotels or both. Assume the flights and hotels are available always 24x7.

The overall flow of the system is as follows:

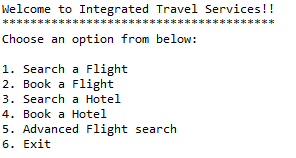


IFS is a menu based system where the user can choose and option by entering an appropriate menu number

The various modules associated with the system are as follows:

# Menu:

The user should be shown a menu as below:



If the user enters an invalid option, display an appropriate error message and display the menu again.

If the user enters a valid option, invoke the appropriate module.

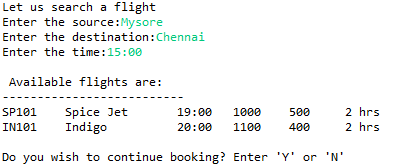
On selecting exit display a thank you message and quit the system.

# Search a Flight (Trainee 1)

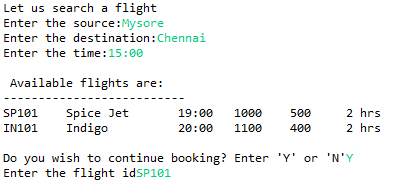
This module allows the user to search for a flight.

This module should accept the source, destination and time of departure from the user and display the details of all flights which have the source and destination as entered by the user and the departure time is greater than or equal to the one entered by the user.

If flights are found, display the flight details as shown below:

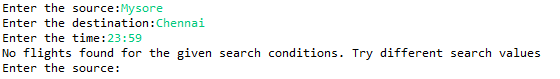


Also ask the user if the user wants to book a flight as well. If the user enters Y, get the flight Id from the user and invoke the Book Flight Module as shown below:



If the user enters ‘N’ display the main menu.

If the validations fail or if no flights are found display an appropriate error message and ask the user to enter the details again. For example:



**Note**:

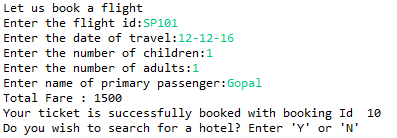
* Source and destination must contain only alphabets.
* The time must be in HH:MM format.
* The source must be a valid source.
* The destination must be a valid destination.

# Book a Flight (Trainee 2):

This module allows the user to book a flight.

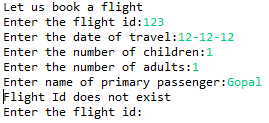
The user has two options. The user can book a flight as a continuation of searching a flight or he can directly choose the option from the menu. If the user is booking a flight as a continuation of searching a flight, the flightId will be supplied from Trainee1 module. Else, Trainee2 has to get the flightid from the user.

The module should get the below inputs from the user:



If the booking was successful, calculate and display the total fare and the auto generated booking Id. Also ask the user if the user wants to search a hotel. If ‘Y’ then, invoke the ‘Search Hotel’ module with necessary data. Else, display the main menu.

In case of any errors, display an appropriate error message and ask the user to reenter the values. For example:

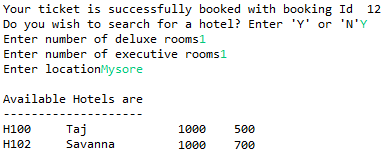


Note:

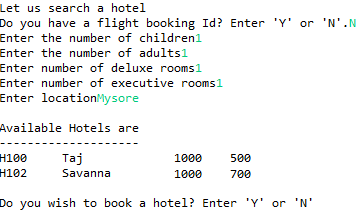
* FlightId must be a valid flight Id
* Date of travel must be dd-mm-yy format.
* Number of children must be 0-4( inclusive)
* Number of adults must be 1-4 (inclusive)
* Every flight will have an different adult fare and child fare
* If the booking was successful, the new booking id should be one greater than the last booking Id. If no booking If exists, it should start with 1.

# Search a Hotel (Trainee 3):

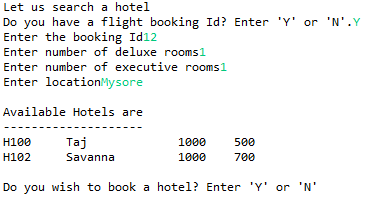
This module allows a user to search for a hotel. The user can search a hotel as a continuation of booking a flight or directly from the menu. If the user is continuing from booking a flight, the menu will be as below:



If the user is searching for a hotel from the menu directly the output will be as below:

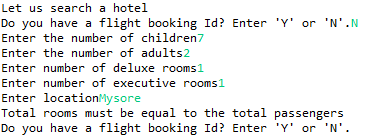


If the user has a flight booking Id, then the output should be as:



Once the searching is done, ask the user if the user want to continue booking a hotel. If yes, invoke the Hotel Booking Module with necessary data. Else display the main menu.

If the searching failed due to any reason, display appropriate error message and ask the user to renter the data. For example:

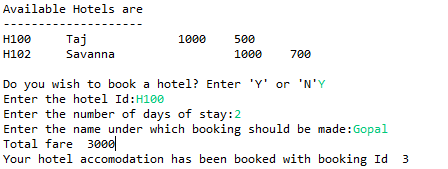


Note:

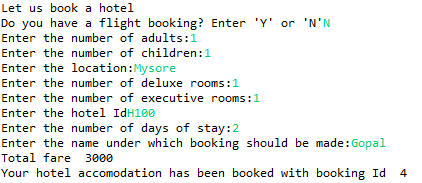
* If flight booking id is used, it must be a valid booking id.
* Number of adults should not be less than the number of executive rooms.
* The number of rooms should be equal to number of people.
* The location entered must be a valid location.
* There must be at least one hotel which match the searching criteria.

# Book a Hotel (Trainee 4)

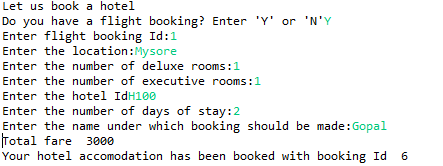
This module allows a user to book a hotel. The user can book a hotel as a continuation of searching a hotel or directly from the menu. If the user is booking a hotel as a continuation of searching a hotel, then the output will be:



If the user is booking a hotel from the main menu, then the output will be:

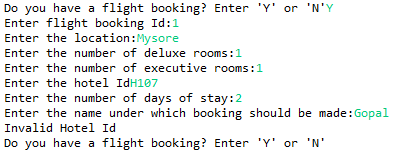


If the user has a flight booking, then the output will be,



Once the booking is done, display the main menu.

If there are any errors, display an appropriate error message and ask the user to reenter the values. For example:



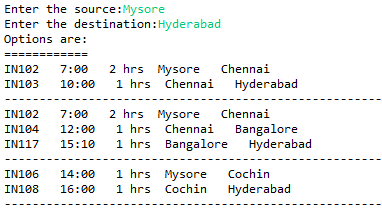
Note:

* If the user is booking a hotel from the main menu, invoke the Trainee3 module and validate the data entered by the user.
* Check if the hotelId entered by the user is valid.
* Include number of days in the calculation of total fare.
* If the booking was successful, the new booking id should be one greater than the last booking Id. If no booking If exists, it should start with 1.

# Advanced Search :

In this module, the user should enter the source and destination. The output should be all the possible routes between the source and destination, with a **maximum of two stops** in between.

A sample output is:



**Note**:

* The departure time of next flight should be after the previous flights duration in the route.
* If there is a direct flight without any stops, then that direct flight should not be repeated

# Appendix:

A user searching a flight, booking a flight, searching a hotel, booking a hotel in a continuous way would have the following flow:

