Airline reservation system

Imagine that you are writing software for an airline reservation system. Implement a program that will read two input files and output to a different file with the format mentioned.

Input file 1 – This input file has information about flights. The data in this file is comma separated with columns Flight number, Number of seats in flight, Price per seat, Origin, and Destination

Input file 2 – This file has list of transactions your application has to perform with comma separated values

Output file – display of your output should be in output.<extension> file in the format mentioned below

Input assumptions:

1. All input will be comma delimited
2. Consider the price amount to be $ and output display summary should have $ in front of the amount. Ex.: output as $45 while input was given as 45 for the ease of coding
3. Passenger name is a sequence of letters with no spaces
4. Origin & Destination are 3 letter airport/city codes
5. Flight number is a letter followed by a set of numbers

Requirements:

1. Read input file #1 and make flight details
2. Write an application to do operations listed below
    a. Book a passenger on a flight – input file has name of the passenger followed by origin & destination
        i. If a flight doesn't exist when booking a passenger, ignore that command
    b. Change the price of a seat on a flight – input file has the flight number followed by the new price
        i. The future bookings made on this flight should be with new price and the earlier bookings remain with older price
    c. Cancel a booking on a flight – input file has passenger name followed by origin & destination

        i. Should refund the booked price to the customer but not the current price since the price may change any time for that flight
    d. Display summary information of a flight – Summarize the below information for each flight listed in input file 1 after all the operations listed in input file2 are over.
        i. Number of seats currently booked, number of seats remaining on that flight, revenue, the passenger information like name, seat number of that passenger and price they paid for that seat
    e. Display EOD summary
        i. Summary of all the flights in total like number of seats and total revenue generated for all the flights together
3. Read input file #2 for list of transactions to perform
4. Process all the operations mentioned in transactions file and display the output in a third file named output
5. If there are multiple flights for a given origin & destination, pick the cheapest flight to book a passenger

Output format (example output):

Flight#: A124 Number of seats available: 49
Total seats sold: 5
Total revenue on this flight: $700

Passenger Name	Seat #	Price
GeorgeWashington	4	$150
MikeSmith	1	$150
AlfredoHatch	6	$130
KenNygard	2	$110
LindaHenry	3	$160

Flight#: B435 Number of seats available: 72
…
Followed by rest of the flights in the similar manner

System's summary:
Total seats sold: 54
Total revenue: $565,891

Additional details:

- You must submit a .zip as an email attachment to orbitzcodingtest@gmail.com
- Email subject line and .zip folder name should both be FirstName_LastName_SchoolName_CodingLanguageUsed
- The .zip archive must contain
    - A README file that includes:
        - Instructions on how to compile/run your program
        - A brief description of your implementation
        - Your name, email, phone number, and school
    - All source files in an src/ directory
    - Input files are maintained in in/ directory
    - Output file is in out/ directory
- Therefore the zip directory structure should look like:
    - FirstName_LastName_SchoolName_CodingLanguageUsed.zip
        - README
        - src/
            - <source files>
        - in/
            - <input files>
        - out/
            - <output file>
- Your program will be evaluated for correctness, performance (space and time efficiency), and code quality.