Airline Seating and Reservation Management System in Java (100 points)

Project Description:

This project simulates a realistic airline reservation system using Java, combining seat assignment, passenger details tracking, and administrative control. It is designed to provide an interactive console-based experience for customers and admins. The system allows passengers to book seats from a dynamically generated seating layout, validates and stores customer data, and enables administrators to manage bookings, cancel reservations, etc.

Key Components:

1. Customer Interface: ***(10 points)***

- Passengers can view the aircraft seat layout and choose from available seats.
- Collects passenger details: **Name, Age, Address, SSN, Contact Number, and Desired Seat**.
- Displays seat map:
    - Occupied seats are marked with **X**, and the map updates after each booking.
    - If a selected seat is taken, the system prompts the user to pick another seat.
- Input validation prevents invalid seat selections and infinite loops.

Assign Passenger seating. Assume 7 rows 4 wide with seats assigned in this manner

Row No.	**---Seats---**
1	A B D C
2	A B D C
3	A B D C
4	A B D C
.	
.	
N	A B D C
.	
.	
last	A B D C

Program requests number of rows. The following display displays the seating pattern with an X for occupied rows:

Row No.	**---Seats---**
1	X B C D
2	A X C D

3	A B	C D		
4	A B	X D		
.				
.				
.				
k	A B	C D		
.				
.				
.				
n-1		A B	C D	
n		A B	C D	

2. Admin Interface: ***(10 points)***

- Admin can:
    - View the complete seating chart with **passenger names** shown in place of seat letters.
    - Cancel a customer's reservation.
    - View and manage customer data saved in CustomerDetails.txt.
- Admin keeps track of each user in a file named: "CustomerDetails.txt" in the following format (All customer records follow the same format):

NAME	SSN	SEAT NUMBER	ADDRESS	AGE	Contact
John Doe	12345678	1A	NewYork	24	3451627900
Jane Doe	……2C…….				
And so on.					

Whenever Admin looks for available seats, he should see the following display:

Row No.	---Seats---			
1	John Doe	B	C	D
2	A	X	Jane Doe	D
3	A	B	C	D
.				
.				
.				
k	A	B	C	D
.				
.				
.				

n-1	A	B	C	D
n	A	B	C	D

If a user cancels their trip, the user's seat should be made available and their details should be deleted from the **CustomerDetails** file. The **CustomerDetails** file cannot hold more 28 customers' details.

Use the following Java concepts in your code:

Concept	Description
Class & Object	Classes for Passenger, Flight, SeatMap, and Admin. ***(10 points)***
Constructor	Parameterized constructors to initialize passengers, flights, and seats. ***(10 points)***
Array & ArrayList	2D arrays for seat layout (seats[row][seat]) and dynamic lists for customers. ***(10 points)***
Inheritance	Admin and Passenger inherit from base class Person. ***(10 points)***
Interfaces	Reservable, Cancellable, and Viewable interfaces for modular design. ***(10 points)***
Exception Handling	Custom exceptions for invalid inputs and overbooking (should include listed and more) - SeatNotAvailableException, InvalidInputException, FlightNotFoundException, InvalidAddressException, InvalidPhoneExcpetion, etc. ***(10 points)***
File Handling	Read/write/update passenger details in CustomerDetails.txt. ***(10 points)***

- ✓ This is an individual project.
- ✓ *You must upload your own original code. Submitting work copied from online sources or others will be considered academic dishonesty and may lead to serious consequences in accordance with the institution's policies.*
- ✓ *Upload all your .java files and a Word document that includes your code and screenshots of your project output* ***(10 points)*** *to Blackboard.*