Air Console Reservation System Planning

Basic Requirements

- Create an Airplane Seat Reservation System
- There is only one route going from Madrid to Barcelona
- There is only one airplane.
- The airplane has 40 seats
- There are 8 Rows, with 5 seats on each row
- Seats are Denominated A-E
- The seat number is a combination of the row number and the seating letter (1-A)(3-E)etc
- There are 2 seat classes (business and economy)
- Business-class spans row 1-5
- Economy-class spans the remaining rows
- All user input must be validated
- Only seat reservation must be considered but upgrades may come later
- All passenger details are saved in a flight manifest file loaded on application start.
- This will be a single terminal application

General Flow

- User start application on the console
- The user gets prompted between (Reservation, Seat Verification, Exit the System)
- User selects R
    - The user gets prompted between (Business Class, Economy Class)
    - The user selects an option
        - The user gets prompted shown grid of seats in their class to choose
        - The user enters row number
        - The user enters seat letter
        - Verify Seat is available
        - User Enters passenger information(first, last, passport number)
        - Shows seat was successfully booked(Adds the booking to the manifest)
        - Goes back to the main menu when the user clicks any key

- User Selects S
    - The user enters row number
    - The user enters seat letter
    - Print passenger details for that seat
    - Goes back to the main menu when the user clicks any key
- User Selects X
    - Exit Application

Additional Questions / Requirements

Can the same passenger register multiple seats with the same information?
(first name, last name, passport number) **No, as different people will be flying**

Full Requirements

1. Console Booking Application
2. Business-class spans row 1-5
3. Economy-class spans row 6-8
4. Allow the user to select a seat based on the row number and seat letter
5. Show the user seats in their selected class
6. Allow the user to enter the passengers (first name, last name, passport number) to complete the booking
7. Goes back to the main menu when actions are completed
8. Allows lookup by seat number to get the registered information
9. Exit application if the user hits x on the main menu

UI Design

```
Welcome Message
Ask the user to select between (R: Reservation, S: Seat Verification, X: Exit the System)
Verify input
User makes a selection
Clear
R
    Show Seat Class Message
    Ask the user to select between (B: Business Class, E: Economy Class)
    Verify input
    Show Selected Classes Message
    Show Selected Classes Grid of seats
    Ask the user to enter a row number
    Verify input
    Ask the user to enter a seat letter
    Verify input
    Verify seat is available and tell the user
    Ask the user to enter the passenger's first name
    Ask the user to enter the passenger's  last name
    Ask the user to enter the passenger's passport number
    Store booking in manifest
   Tell user booking was successful
    Go back to the main menu when the user presses a key
    clear
S
    Ask the user for the row number
    Ask the user for the seating letter
    Print the passenger details for that seat
    Go back to the main menu when the user presses a key
    clear
X
    Exit the application
```

Logic Design

Method: Print out the prompts
Method: Create the grid for each class
Method: Print out the seating class grid
Method: Determine if the seat entered is available
Method: Method: Verify input is a letter A-E
Method: Verify input is a number on the grid
Method: Load Manifest on initial load
Method: Save booking to manifest by overriding initial save location
Method: Check if Seat is Business or Economy Class

Data Design

PassengerModel
Passenger’s FirstName - string
Passenger’s LastName - string
Passenger’s FullName - string
Passenger’s Passport Number - string

GridSpotModel
SpotLetter - string
SpotNumber - int

ManifestModel
SeatsAvailable - List<GridSpotModel>
SeatsTaken - Dictionary<GridSpotModel, PassengerModel>