Junior Programming Problem

The year is 2019. You and your friends are running a small software company and you have heard that the massive local airline Airberta is looking for developers to create a digital booking system for the people at their airport in Calmonton, Alberta (airport code YEY). They have put out a call for the most impressive proof-of-concept and will hire their favourite group. You really need this job because all the work you've been getting so far is far too easy for you and you require a challenge.

Airberta is open to any kind of user-friendly application that has a front-end for customers to purchase tickets. The initial state of the application will be provided in a CSV file and should be able to be loaded at start time. Your group will be given a file to test with during development, but your application must be able to load the judging data as well.

Presentation

Once you have completed your application, each group will give the judges a 15-minute presentation as a pitch for your product. It should outline how your product meets or exceeds the expectations and why it is the best choice. This will be worth 30% of your overall score and will be judged based on clarity, confidence, organization, visual aesthetic, and your ability to answer questions.
Order of presentations will be randomly chosen.

Submission

All code along with any user documents must be emailed to robotoshi@gmail.com by **1:30pm**.
Please include SEC in the subject line and list your group member names in the email.

Design Specifications

Airberta has two different types of planes, the **BigLad9000** and the **SmolBoi220**.
The SmolBoi220 has 24 seats, with 6 rows of 2 seats on either side of the centre aisle. There is no first class.
The BigLad9000 has a first class section, which has 8 seats, with 4 rows of 2 seats on either side of the centre aisle, as well as a coach section which has 36 seats, with 6 rows of 3 seats on either side of the centre aisle.

The CSV will contain a schedule of flight departures with the departure time in unix time format.

Airberta requires the following, however any additional features will be looked upon favourably.

- Must be able to load a CSV file at start time
- Allow the customer to find and reserve a flight based on:
  - Destination
  - Date and time
  - Price
  - Seats together available
- When purchasing a ticket, store the customer's information and log the transaction
- Customer will be given a confirmation and ticket number for each seat purchased
- Allow the customer to request up to three seats together for no extra charge
- Allow the customer to visually pick a specific seat for an extra fee of $50
  - If the customer does not elect to pick their own seat, they should be automatically placed in such a way that maximizes the available space on the plane
- Automatically reduces the price of every remaining empty seat by $70 once there are only individual seats available in that section
  - Inform the customer that this is a discount
- Output a log file of every action that has occured

Judging:

Will be judged based on efficiency and user usability.

Criteria	Points
Design & Performance	**/70**
• Does the design work? • How well does the design meet the requirements of the project? • Does the solution include relevant extra components on top of those requested? • Did the solution come with appropriate user documents? • Does the program use a simple interface? Does the program utilize mouse and keyboard?	
Presentation	**/30**
Presentation • Did the members of the team appear to work well as a team? • Did all members contribute to the problem solving process? • Did the team appear professional? Were all members available? • Was time used appropriately? • Did all team members participate equally in the presentation? • Was the team communication clear? Such as: Proper use of jargon? Not extremely technical?	/10
Code Methodology • Was the code architecture explained? • Were the benefits and principles of the design clearly explained? • Was the program code made in a systematic method? • Were any open source components made apparent?	/10
Code Scalability and Maintainability • Did the team properly consider Scalability and Maintainability? How was this implemented? Were this properly demonstrated/presented? • Did the team follow industry level coding architecture to facilitate future use, Scalability, and Maintainability?	/10
Total	**/100**