Senior Programming Problem

The year is 2019. You and your friends are running a small software company and you have heard that the massive international food courier service BreadEx is looking for developers to create a digital asset management system for the people at their warehouse in Calmonton, Alberta. 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.

BreadEx is open to any kind of user-friendly application that is able to plan the shipments for all the parcels that need to be delivered. The application needs to be able to decide which parcels go on which trucks, and when they should depart, the goal being to clear the warehouse. For example, every parcel has a time received, expiry date, weight, and some have special requirements such as delicate handling or refrigeration. Additionally, each truck has certain attributes such as speed, capacity, fuel economy, and services such as delicate handling and refrigeration. Each truck always departs with a load and returns empty. From the beginning, some trucks are on their way back to the warehouse. You must include this fuel consumption as well.

The initial state of the BreadEx assets/system will be provided in a CSV file, containing parcel arrival, truck availability, and destination data 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 live/random/spontaneously generated judging data as well.

To simulate a more accurate real-life situation, the sample data will be released incrementally throughout the development time.

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

All times will be specified in terms of "hours from now"; thus negative times signify the past. Times may also be decimal values.

Note that some times are in the past and some are in the future. This means that some parcels have yet to arrive,soyou need to plan accordingly. Sometimes the most optimal solution is to wait for a parcel to arrive. Parcels do not arrive by truck, so they need not be considered during arrival. Also keep in mind that each truck must return back to the warehouse by the end, but each truck may be used more than once.

BreadEx requires the following; however any additional features will be looked upon favourably.

- Be able to load the CSV files at runtime (user-friendliness would be nice)
- A detailed, visual schedule of all truck arrivals and departures
    - Include a visual packaging representation for how the parcels are packaged in each truck for each delivery,
    - Include all the relevant numbers for each load (truck capacity, load weight, fuel consumption)
- If it is not possible to deliver a parcel before it expires, report it as such and do not deliver it
- Report the following values. Use your judgement and explain in your presentation which values you prioritized
    - Number of truck trips (including return trips)
    - Total distance travelled
    - Total fuel consumption (the ultimate goal is to minimize this)
    - Sum of all the "leftover capacities" for each load (capacity minus load weight)
    - Total sum of expiry date minus arrival time for each parcel (the ultimate goal is also to minimize this)
    - Arrival time of the last truck
    - Number of expired parcels
- Output a comprehensive log file of every truck load and all of its contents

Data Format

Data will be delivered in the following order and with the following headings:

Trucks

ID number	Initial distance (km)	Speed (km/h)	Capacity (kg)	Fuel economy (L/km)	Delicate (Y/N)	Refrigerated (Y/N)

Destinations

ID number	Name (display this)	Distance (km)	Fuel economy (L/km)

Parcels

ID number	Name (display this)	Arrival time (hours from now)	Destination (ID number)	Expiry (hours from now)	Weight (kg)	Fuel economy (L/km)	Delicate (Y/N)	Requires Refrigeration (Y/N)

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**