Littlepay Coding Exercise

Tuan Luong

Assumptions Made during this exercise

1. I’ve assumed that this problem will have a big json file. In public transport, thousands of users use the public transport system each day, which requires a big input file. So I used the Jackson streaming library to parse the JSON file into variables.
2. The second assumption I’ve made is that the customer will always Tap On and Tap off when cancelling their trip. An edge case that might occur is that a customer cancels their trip, but forgots to touch off, which can lead to an incomplete trip.
3. The Third assumption that I’ve made during this coding exercise is that the JSON output is in an unordered format in the trips.json, due to the JSON object being based on a hashmap.
4. The Fourth assumption that is made in this exercise is that the taps.json will be from real time data, as each user taps on or off.

Running the program.

When this program is run, it will take input from the taps.json as the test harness and output in the trips.json file. The main method is in the main.java file and the functionality is implemented in the TapSystem.java file. The solution is a eclipse java archieve which can be imported using the instructions below.

1. Open Eclipse
2. Click File->Import->Projects from Folder or Archieve

Graphical user interface, application

Description automatically generated

1. Select Archive near the archive source input box

Graphical user interface, application, table

Description automatically generated

1. Click the littlepayexercise.zip

Graphical user interface, text, application

Description automatically generated

1. Select the littlepay folders

Graphical user interface, text

Description automatically generated with medium confidence

1. Click Finish to import project into eclipse workspace.
2. In the TapSystem.java file on line 141, ensure the jsonparser source is set to “src/taps.json” (Test harness)

And on Line 210, where the filewriter is ensure it is set to “src/trips.json” (Output file)

Text, letter

Description automatically generated

Text

Description automatically generated with low confidence

1. Run the program from main.java in eclipse.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

1. The output will be in the trips.json file.

A picture containing graphical user interface

Description automatically generated