Jenny Lam, Bella Zhan
COMP SCI 393
Professor Dimoulas
30 September 2019

Assignment 1 Part 3 - Interface

The following interface design describes a simple software system with a front-end service that passes user inputted information to a back-end service for processing. Once processing is complete, the front-end service provides the user with output returned from the back-end service.

Interfacing Operations Offered:
Front-end needs to be able to check that the back-end isn't working on processing something.
Front-end needs to be able to send one list of JSON objects to back-end.
Back-end needs to be able to receive one list of JSON objects.
Back-end needs to be able to send one list of JSON objects back to front-end.

Front Component
Internal Data:

- **objArr**: Array to store inputted JSON objects
- **nsq**: Queue for storing non sorted arrays
- **sortedArr:** Array for storing sorted arrays

Methods:

- readIn()
  - Accepts inputted JSON objects from STDIN and stores them into **objArr.**
- partition()
  - Divides the inputted JSON objects into lists of 10 by looping through **objArr** and adding every 10 objects into a new list before pushing the list of JSON objects to **nsq.**
- send()
  - Iterates through **nsq** and pops off a list of 10 objects each time to send to the back-end component, if the back-end component is not working on a process as determined by the back-end **busy** field. Waits for response.
  - Once the back-end component returns a sorted list of JSON objects, the new sorted list will be pushed to the front-end internally stored array, **sortedArr**.
  - Send will terminate when **nsq** is empty and will proceed to call print().
- print()
  - Prints **sortedArr** to STDOUT as an array of sorted arrays of JSON objects.

Back Component

Internal Data:

- **lst**: list for storing the accepted list of JSON objects from the front-end component
- **busy**: boolean indicating whether the back-end service is currently processing a list

Methods:

- receive()
  - Checks that **busy** is False before accepting any new lists and checks that the list being received is in the correct format (list of 10 JSON objects or less).
  - Marks **busy** as True, stores the correctly formatted list, and calls sort().
  - If **busy** is already True, it will send a status message to the front end to wait before sending the next list.
- sort()
  - Sorts the received list of JSON objects from the front-end component by some predetermined algorithm and returns the sorted list back to the front-end.
  - Marks **busy** as False after sending sorted list

Other Relevant Information:

We are assuming that if the number of JSON objects is not divisible by 10, the services will be able to send and sort lists that have length less than 10. The front-end component will control this case where the number of JSON objects is not divisible by 10.