Project Design

Eunjin Kwon

1. Describe the user interface. What are the menu options and how will the user use the application?

   Menu options

   1. added a new station on the CTA
   2. modify an existing station on the CTA
   3. remove a station from the CTA
   4. search for a station within the CTA
   5. find the nearest station within the CTA
   6. show a path between two stations
   7. exit

   the user will use this application, adding, removing, and searching the station on the CTA. With the latitude and longitude, the user typed, calculate the distance and find the nearest station. Lastly, the user would know the best path to go from the current station to the destination station.

2. Describe the programmers' tasks:

- Describe how you will read the input file.
  For reading data line by line, construct by passing the file object of provided CSV file within a try/catch.

- Describe how you will process the data from the input file
  Using string.split() to split the line by the comma can facilitate reading data from the file.

- Describe how you will store the data (what objects will you store?)
  Create a new instance to set each information from the CSV file using while loop

- How will you add/delete/modify data?
  To add a station on the CTA, create the method of addstation with scanner and ArrayList. Prompt the station name to a user and create new station information to add the station to the array. To remove a station on the CTA, establish the method of removestation with the scanner and ArrayList parameter as well. Ask the user which station would like to remove and create the new ArrayList to find that station from the file. If the station is matched, remove the station using remove( ). Lastly, to modify data, find the station asking for the station name. And look up all stations with the given name. If the station had that name, use

- How will you search for the data?

  Create the method for search the station and returns all the stations which are a match with a given name.

- List the classes you will need to implement your application.

  Geolocation: To manage the latitude, longitude and find the nearest station
  CTAstation: To manage the information of each station
  CTARoute: To manage the modification of the stations.

2. Draw a UML diagram that shows all classes in the program and their relationships.

Geolocation
-latitude: double -longitude: double

CTAstation
-name: String -location: String -opened: boolean -location: String -line: int

CTARoute
-name: String -stops: ArrayList<CTAstation>
+addstation(station:CTAstation) : void +removestation(station:CTAstation) : void +modifystation(station:CTAstation) : void +lookupstation(position: int, station:CTAstation) : CTAstaion +neareststation(latitude: double, longitude: double): CTAstation +bestpath(name: String, name: String): CTARoute

3. Think how you will determine if your code functions are expected. Develop a test plan based on the above description; how will you test that the expected outputs have been achieved? **Be sure this test plan is complete.**

   The input will depend on the option selected by the user. Finding the best path between two stations shows the route which is connected to two stations the user entered. If the user wants to add or modify a station, the input will be the information of the station, and the output will be reflected when the user prints the list of stations. After I create the addstation, removestation, modifystation methods from the menu options, I will test the method using system.out.println and scanner input to see the result of them. From the repetitive process, I will notice where I made an error and how to fix my code. Plus, I can confirm that my application actually adds and removes the station on the CTA.