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

The menu will be printed to the console with the following options:

1. Display information of all stations
2. Display information of stations with wheelchair access
3. Search for a station
4. Add a new station
5. Modify an existing station
6. Remove a station from the CTA System
7. Find the nearest station to a location
8. Find the route between two stations
9. Exit

The user will be continually prompted to choose between the above options. If the user chooses “exit”, the program will end.

1. Describe the programmers' tasks: Basically, I will use similar data structure with my solution of lab6, which includes 3 main classes: GeoLocation, CTAStation(a sub class of GeoLocation), and CTARoute.
2. Describe how you will read the input file.

Use a Scanner to read input from the file.

1. Describe how you will process the data from the input file.

With each line of data, split the line into an array of Strings, then use them to create a new object of the class CTAStation. If this CTAStation belongs to a route (e.g. Green Line), add it to the ArrayList of that route. After finishing reading input from the file, I will sort each CTARoute so that every CTAStation is at its correct order on the route. For example, Howard station will be the first station on the CTARoute “Red Line”. When stations are at their correct order, I can take advantage of the add/remove/insert methods of the ArrayList.

1. Describe how you will store the data (what objects will you store?)

I will store 8 CTARoutes (Green, Red, Blue, Brown, Pink, Purple, Orange, Yellow). Or I may use a ArrayList<CTARoute> to store the data.

1. How will you demonstrate polymorphism?

I can create sub classes of CTAStation, namely RedLine, GreenLine, BlueLine, etc. Each sub class will have a variable to store its position on the route. Each sub class will have its own toString method to show that the station is the x th station on its route. So if the program iterate over an ArrayList of CTAStation and call the toString method, the appropriate method of each route is called.

1. How will you add/delete/modify data?

When adding data, user will be prompted to input information of the station: name, latitude, longitude, location, wheelchair access, the route(s) that station is on, and for each route input the name of the previous and following station (Lets call them A & B). After having all the data, check to see whether those 2 stations are next to each other in a route. If not, tell the user that the input is not correct. If the input is correct, insert the new station to the position of B station using the method of ArrayList. Update position of all stations that come from B to the end of the route.

When deleting a station, user will be prompted to input the name and the route of the station. The program will display the information of the station (if the input is correct) and ask if the user really want to delete this station. If the user chooses “Yes” then the station will be removed. Update position of all stations after the removed station.

When modifying an existing station, user will be prompted to input the name and the route of the station. The program will display the existing information and ask if the user really want to modify this station. If the answer is “Yes” then the user is prompted to enter all the information, similar to the adding data case. A new station will be created and then it replaces the existing station.

1. How will you search the data?

User is prompted to enter name of a station. Loop through all CTARoutes to find if the String entered by the user is the same as the name of stations on CTARoutes.

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

* GeoLocation
* CTAStation (sub class of GeoLocation)
* CTARoute (including an ArrayList of CTAStation)
* Sub-classes of CTAStation (RedLine, GreenLine, BlueLine, BrownLine, PurpleLine, PinkLine, OrangeLine, YellowLine)

1. UML diagram: I ended up with 2 UML diagrams, since I am not sure which one I’m going to use for the final project.
2. Test plan:

|  |  |  |
| --- | --- | --- |
| Condition/Input | Expected result | Notes |
| Run the program | The main menu shows up in the console. |  |
| The entered input is not from “1” to “9” | A message pops up in the console: "Sorry, I do not understand your input, please try again.". The menu shows up again. |  |
| From the menu, enter 9 | The program stops with a message: “Goodbye!” |  |
| From the menu, enter 1 | Information of all stations are printed out to the console. Back to main menu. | Stations on different routes have different toString method -> polymorphism confirmed |
| From the menu, enter 2 | Information of all stations with wheelchair access are printed out. Back to main menu. | Stations on different routes have different toString method -> polymorphism confirmed |
| From the menu, enter 3 | A message prompting for the name of the station |  |
| Enter a random string | Message: “Station not found.” .Back to main menu |  |
| Enter a valid name of a station | Information of station printed out. Back to main menu. |  |
| From the menu, enter 4 | A message prompting for the name of the station |  |
| Enter name of a new station | A message prompting for the latitude of the station |  |
| Enter invalid input/ Enter an input type double | Message: Sorry, I do not understand your input. / A message prompting for the longitude of the station |  |
| Enter invalid input/ Enter an input type double | Message: Sorry, I do not understand your input. / A message prompting for the location of the station |  |
| Enter a string | Message: Does the station has wheelchair access(Y/N)? |  |
| Enter input different from Y or N/ Enter Y or N | Message: Invalid input. Message: Does the station has wheelchair access(Y/N)? / A message prompting for the route the station is on, separated by a comma |  |
| Enter invalid names of routes/ Enter valid names of routes | Message: Invalid input. Please try again. / Message: Station added successfully. |  |
| From the menu, enter 5 | A message prompting for the name of the station |  |
| Enter name of the station | A message prompting for the route the station is on, separated by a comma |  |
| Enter invalid names of routes/ Enter valid names of routes | Message: Invalid input. Please try again. / Information of the station displayed. Message: Are you sure you want to modify this station? (Y/N) |  |
| Enter input different from Y or N/ Enter N / Enter Y | Message: Invalid input. Please try again. / Message: No change was made. Back to main menu. / A message prompting for the name of the station |  |
| Enter name of the station | A message prompting for the latitude of the station |  |
| Enter invalid input/ Enter an input type double | Message: Sorry, I do not understand your input. / A message prompting for the longitude of the station |  |
| Enter invalid input/ Enter an input type double | Message: Sorry, I do not understand your input. / A message prompting for the location of the station |  |
| Enter a string | Message: Does the station has wheelchair access(Y/N)? |  |
| Enter input different from Y or N/ Enter Y or N | Message: Invalid input. Message: Does the station has wheelchair access(Y/N)? / A message prompting for the route the station is on, separated by a comma |  |
| Enter invalid names of routes/ Enter valid names of routes | Message: Invalid input. Please try again. / Message: Station modified successfully. |  |
| From the menu, enter 6 | A message prompting for the name of the station |  |
| Enter name of the station | A message prompting for the route the station is on, separated by a comma |  |
| Enter invalid names of routes/ Enter valid names of routes | Message: Station not found on that route. Back to main menu. / Information of station displayed. Message: Are you sure you want to remove this station? (Y/N) |  |
| Enter input different from Y or N/ Enter N / Enter Y | Message: Invalid input. Please try again. / Message: No change was made. Back to main menu. / Message: Station removed successfully. |  |
| From the menu, enter 7 | A message prompting for the location’s latitude |  |
| Enter invalid input/ Enter an input type double | Message: Sorry, I do not understand your input. / A message prompting for the longitude of the location |  |
| Enter invalid input/ Enter an input type double | Message: Sorry, I do not understand your input. / Information of the nearest station displayed. Back to main menu. |  |
| From the menu, enter 8 | A message prompting for the name of the first station |  |
| Enter name of the first station | A message prompting for the name of the second station |  |
| Enter name of the second station | Route between two stations displayed. Back to main menu. / Message: Cannot find the route between those stations (input not valid). Back to main menu. |  |