# GROUP 3



# DCIT 204 END OF SEMESTER PRESENTATION

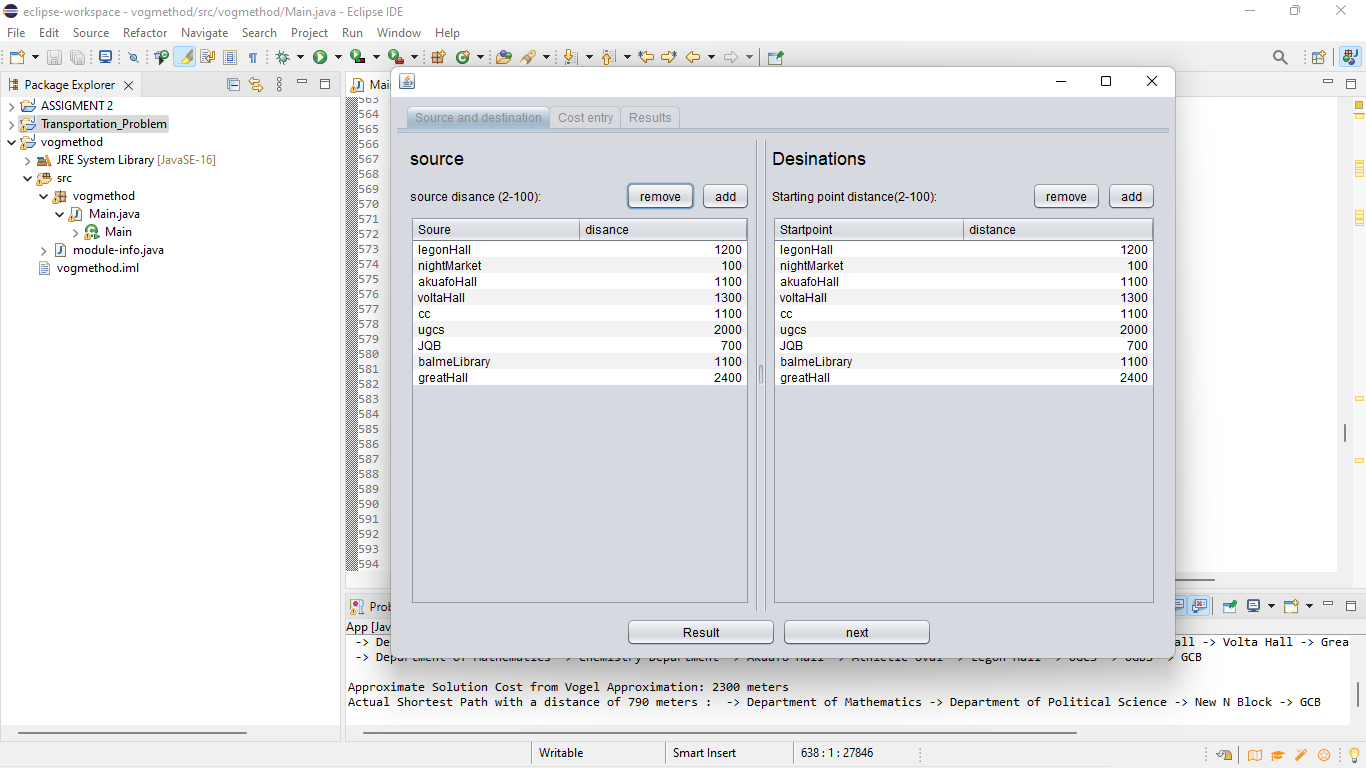
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | ID | ROLE | ACTIVITIES  PERFORMED | PERCENTAGE CONTRIBUTION |
| James Mensah | 10847265 |  | User interface design, part of the vogel algorithm | 33.33 |
| Edward Kuagbenu | 10840142 |  | Wrote the algorithm, calls for group meetings, did the implementation of the various codes. | 33.33 |
| Ofori Eric Young | 10824156 |  | Found the distances between venues on campus, worked on the graph and vertex | 33.33 |
| Amoakohene | 10837866 |  |  | 0 |
|  |  |  |  |  |
|  |  |  |  |  |

This program was written as an assignment of the our Data Structure and Algorithm 1 end of second semester project to solve transportation problem on campus ,thus find the efficient and shortest possible path between two locations.

Group Objective:

1. Find shortest paths in weighted graphs
2. Java Class Design
3. Write classes in Java to implement graphs
4. Create graphs in Java using an adjacency matrix representation and an adjacency list representation
5. Critique aspects of code design Key algorithms used in the project:
6. Breadth First Search
7. Vogel’s algorithm
8. A\* Search

User interface:



**ALGORITHM:**

* The node/location to be searched for is empty
* Assign distance values on the various locations that can be used
* Input your locations i.e., source/starting point and destination
* When the locations are passed, the nodes of the vertices nearby the location given is returned
* The nodes of the various vertices are compared to find the shortest path
* If a particular node/location is the shortest distance to the destination without hinderances, select that path. The various paths gotten is stored which will serve as a reference to find other paths
* You can even remove or add your locations and their various distances.
* If the nodes don’t give you the shortest path continue comparing till the path is found
* When shortest path to the destination is found, end the program

**PROJECT DEVELPOMENT:**

**Due to insufficient time, we could not finalize the whole project but with time, we hope to make more it better and make a more nicer user interface, and to even initiate the map into for the shortest possible distance to be shown.**

**LESSONS:**

1. **The shortest route is not only implemented using the Djikstra algorithms.**
2. **We also learned that vogel’s algorithm one of the best algorithms for transportation problems.**
3. **Lastly, we learned team work is very important, with some activities only a person can’t achieve everything inside.**
4. **We had to learn some assumption methods and implement them into our codes.**

**CHALLENGES:**

**We had challenges organizing group meetings.**

**We could finally and properly implement the vogel’s algorithm with the references on sakai and some few researches of understating.**

**Time was so fast against us.**

**We had issues with the time aspect, and the method that would be used to**

**implement it.**

**Lastly, the group actually gave us a tough time, some even left, others too never talked or shared a single idea. We had to use our studies time to do almost everything.**

**SIGNATURES:**

