Group DO Project Three Proposal

GROUP: DO	TA NAME: Rahul
FLIPGRID: https://flip.com/2e92f352	TA EMAIL: rahul.choudhary@wisc.edu

Name	Email	Team	P1 Role	P2 Role	P3 Role
Rohan Ramachandran	RNRAMACHANDR@WISC.EDU	Red	DW	FD	BD
Raihan Tanvir	RTANVIR@WISC.EDU	Red	AE	DW	FD
Aditya Roy	ROY35@WISC.EDU	Red	BD	AE	DW
Ruilin Yan	RYAN56@WISC.EDU	Red	FD	BD	AE
		Blue	DW	FD	BD
		Blue	AE	DW	FD
SHANTANU Chaudhuri	SCHAUDHURI5@WISC.EDU	Blue	BD	AE	DW
Shyama Subrahmanya Nikhil Kruthiventi	SKRUTHIVENTI@WISC.EDU	Blue	FD	BD	AE

Project Title: Bus Route Calculator

Brief Project Description:

The application allows users to generate shortest and most efficient bus routes between two bus stops. The application allows users to enter their current Bus stop location and their destination location, in order to generate the shortest route between them. The application will also generate the minimum spanning tree of the graph and the total distance when a set of paths are inputted by the user. The application is based on a graph data structure where the vertex is equivalent to a bus stop and the edge is equivalent to the path between both the stops.

Representative Tasks Performed Using this Application:

1. <description of first task> (demo by <assign role>)
2. To generate the shortest paths between bus stops using for minimum spanning tree and checking if it is generated correctly(demo by Algorithm Engineer)
3. To demonstrate UI that can be used to travel between bus stops and find the distance traveled between the two stops. (demo by Frontend Engineer)
4. To demonstrate that when a user requests a bus route with a stop, it will be the correct shortest path. (demo by Backend Developer)

Data Wrangler (DW) Role: **Aditya Roy, SHANTANU Chaudhuri**

Ensure that the .gv files that we use are converted into a form that is usable by the rest of the team

Data Description:

We are taking a .gv file and then converting the data to something that our Backend can use.

Development Responsibilities:

We are going to implement the interface IDataLoader.java, containing the method float[][] loadData(File gvFile) and a helper method int getRowCol(File gvFile) in order to get the amount of stops. Our backend will use this method to load the data from a .gv file.

Presentation Responsibilities:

Demonstrate selecting a file for our calculations.

Algorithm Engineer (AE) Role: **Ruilin Yan, Shyama Subrahmanya Nikhil Kruthiventi**

The Algorithm Engineer has two main responsibilities. One responsibility is to extend the GraphADT Interface to use the functionality of the Dijkstra's algorithm and other methods of its implementation. The Second responsibility will be to develop a method to find the Minimum spanning tree for the graph to find the shortest path between all the given nodes and a method to find the total cost of all edges in the input list.

Capabilities Added to Required Data Structure:

Since everyone in the group will be implementing the Dijkstra’s Shortest Path Algorithm, in addition we are also implementing : -

- A method to calculate the Minimum spanning tree which will provide the shortest path between each Bus Stop.
- A method to find the total cost of all the edges in the input list.

Development Responsibilities:

The Algorithm Engineer will focus on extending the GraphADT.java interface by implementing the BusStopGraphInterface.java. This extension will allow the Backend Developer to make use of the methods of GraphADT in a project specific way. In addition to extending, the Algorithm Engineer will also develop the methods MST() and getEdgeWeightsSum() while implementing the BusStopGraphInterface.java interface.

Presentation Responsibilities:

After integration, the Algorithm Engineer will record a video demonstration of using the Bus Route Calculator Application to generate the shortest paths between bus stops using the MST method in Algorithm Engineer code and show that the minimum spanning tree is correctly generated.

Backend Developer (BD) Role: **Rohan Ramachandran,**

The Backend developer's responsibilities include implementing the algorithms provided by the algorithm engineer to provide functionality to the app. This will include implementing methods like finding the shortest bus path possible. They will also be responsible

Backend Functionality Description:

Depending on what other implementation details we decide upon, the Backend Developer will also be responsible for creating that functionality in the app. Backend Developer will also pass its methods to Frontend to allow the user to access the methods.

Development Responsibilities:

Backend Developer will develop a class for the IBusRouteCalculatorBackend interface. This class will take the algorithms with extended functionality provided by the AE and will add their functionality to the app. This class will be passed to the frontend so users are able to access the full functionality of the app.

Presentation Responsibilities:

Demonstrate that when a user requests a bus route with a stop, it will be the correct shortest path.

Frontend Developer (FD) Role: **Raihan Tanvir,**

To create an interface that ther user and use to access and utilize the program. The interface will have find the beginning bus stop and let the user choose where they wish to travel to. The interface will then tell them they have traveled to the bus stop and will tell them the distance that was traveled.

Log of a Sample Execution of the App (user input is colored **red**):

```
Welcome to a map of bus routes!
x+x+x+x+x+x+x+x+x+x+x+x+x+x+x+x+x+x+x+x

You are currently located in:
          1) Location 1
          2) Location 2
          3) Location 3
          4) Location 4
1

Where would you like to travel to?:
          1) Location 2
          2) Location 3
          3) Location 4
          4) I am done traveling!
2
You have traveled from Location 1 to Location 2, you traveled (distance)

Where would you like to travel to?:
          1) Location 1
          2) Location 3
          3) Location 4
          4) I am done traveling!
3

You have traveled from Location 2 to Location 4, you traveled (distance)

Where would you like to travel to?:
          1) Location 1
          2) Location 3
          3) Location 4
          4) I am done traveling!
4

Thank you for traveling with us! (End program)
```

Development Responsibilities:

Frontend Developer will implement a class for the IBusRouteCalculatorFrontend interface. This implementation will be used to accept user input. The constructor of the class will accept the backend class and will provide access to the backend methods of the program. Then, as the final functionality of the program, the constructor will also accept a scanner to allow it to read inputs/choices from the user. This will allow the user to be able to fully utilize the application.

Presentation Responsibilities:

Frontend will record and present a functional UI that takes input and returns an output.

Scope and Signatures:

Ideas for Scoping Up:

One option for scoping up the project is to allow the frontend to receive input to add a new bus stop(vertex) into the graph. In addition, we can also extend the project to allow user to delete a path(edge) in the graph if multiple path exists to the destination stop(vertex)

Ideas for Scoping Down:

One option for scaling down the project are 1) omit the Minimum Spanning tree method 2) omit removeVertex() method

Outside Libraries and Other Tools:

.dot or .gv file with DOT(graph description language) to import the graph(Dataset).

Team Signatures:

Name	Email	Team	Type Name As Signature
Rohan Ramachandran	RNRAMACHANDR@WISC.EDU	Red	Rohan Ramachandran
Raihan Tanvir	RTANVIR@WISC.EDU	Red	Raihan Tanvir
Aditya Roy	ROY35@WISC.EDU	Red	Aditya Roy
Ruilin Yan	RYAN56@WISC.EDU	Red	Ruilin Yan
		Blue	
		Blue	
SHANTANU Chaudhuri	SCHAUDHURI5@WISC.EDU	Blue	Shantanu Chaudhuri
Shyama Subrahmanya Nikhil Kruthiventi	SKRUTHIVENTI@WISC.EDU	Blue	Shyama Subrahmanya Nikhil Kruthiventi

TA Feedback:

<During grading, your TA will share any questions or concerns that they have about this group proposal here.>

Team Response:

<After grading, if the TA Feedback above describes and required clarifications or changes to this proposal, please discuss as a group before acknowledging and addressing those concerns here.>

Proposal Amendments:

If your group needs to make any changes to what is described above after the proposal deadline, then 1) make sure everyone in your group agrees with those changes, 2) describe those changes in the first empty row below, and then 3) notify your group's TA about those changes and whey are being made. Your TA will then review your request and indicate whether they approve of such changes by adding their initials to the end of that amendment's row below.

Number	Description	TA Approval
1	Prim's Algorithm implementation of MST for this project is not possible because the edges are directed, so a modified version is implemented instead where the MST consists of the shortest route to all other nodes, so depending on the starting node the algorithm will return different MSTs.	Approved.
2		
3		
4		
5		
6		
7		
8		