**Joshua Gizaw**

**Uml Diagram**

**Assignment 6**

**Graph**

* **Attributes:**
  + vertices: HashSet<Town>
  + edges: HashSet<Road>
  + shortestPath: ArrayList<String>
* **Methods:**
  + Graph()
  + boolean addVertex(Town v)
  + boolean removeVertex(Town v)
  + Road addEdge(Town sourceVertex, Town destinationVertex, int weight, String description)
  + Road removeEdge(Town sourceVertex, Town destinationVertex, int weight, String description)
  + boolean containsVertex(Town v)
  + boolean containsEdge(Town sourceVertex, Town destinationVertex)
  + Set<Town> vertexSet()
  + Set<Road> edgeSet()
  + Set<Road> edgesOf(Town vertex)
  + Road getEdge(Town sourceVertex, Town destinationVertex)
  + ArrayList<String> shortestPath(Town sourceVertex, Town destinationVertex)
  + void dijkstraShortestPath(Town sourceVertex)
  + int getTotalWeight(String str, Town sourceVertex)
  + int getInteger(String str)

 **Town**

* **Attributes:**
  + name: String
  + neighbors: HashSet<Town>
* **Methods:**
  + Town(String name)
  + String getName()
  + void addNeighbor(Town t)
  + void removeNeighbor(Town t)
  + Set<Town> getNeighbors()

 **Road**

* **Attributes:**
  + source: Town
  + destination: Town
  + weight: int
  + description: String
* **Methods:**
  + Road(Town source, Town destination, int weight, String description)
  + Town getSource()
  + Town getDestination()
  + int getWeight()
  + String getName()
  + boolean contains(Town t)