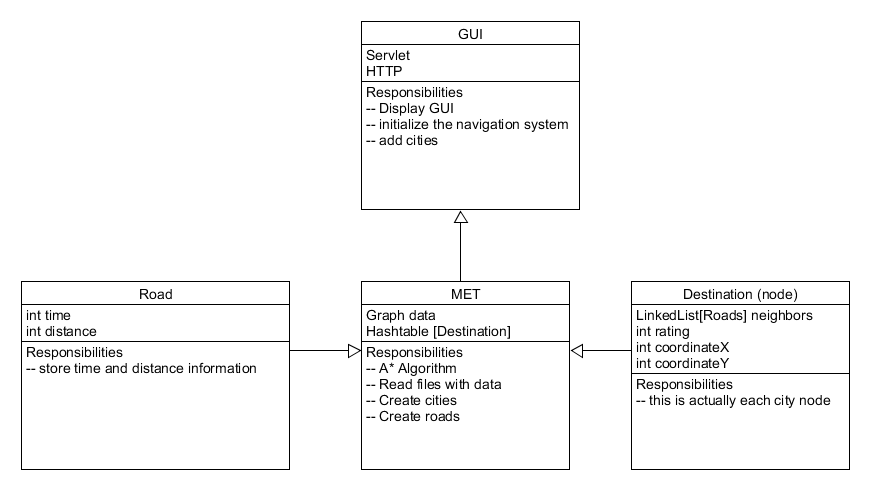
**UML Diagram**



**User Stories**

* A user would want to find how long it takes to get from point A to destination B.
* A user would also want to get the shortest distance from point A to destination B.
* The user will be given the option to choose up to two other routes by distance or times.
* Show all the cities and destinations by ratings
  + EXTRA: Show all the cities and destinations near the destination
* A user could use the trip planner to specify the miles or times that he/she is willing to spend on the road. Based on that information, the navigation system will provide several trip options.
* The user would be able to add destination that does not exist already to the map.
  + The user would need to add the city name, rating, connections to other cities or destinations, and coordinates (x and y).

**Data Structures used and their justification:**

We will use a Hash Table to store all the “nodes” or cities, and then access our “beginning node” for where the user is with O(1) access. We will use a graph as a general data structure that contains all destination nodes (this is also a requirement of the project). We will use a linked list inside the Destination classes that contains other destinations (or roads to other destinations) that (this) destination is connected to. A linked list is the best option because in our A\* algorithm, we will be looking at every node in the linked list so we don’t need an O(1) access that would be available in a data structure such as an Array-List, however we do want to be able to add on connections to other cities if the user decides to, which is an O(1) operation in a linked list.

GUI DISPLAY SKETCH

