Team J: Dalton Bealer, Jeremy Bell, Brenden Guillen, Joe Panthalani, Jonathan Thornton

1. Project Goals

   a. Primary Objective

      i. Develop a program that, given a user defined graph and other important data, will return the optimal routes for a plane to take while meeting certain criteria (all nodes visited, cannot run out of fuel mid flight, etc.)

   b. Goals

      i. Develop a user interface to easily interact with the server. GUI will ideally allow the user to select locations that need to be visited, then will find the best route when the user submits. The user will also be able to select which algorithm to use.

      ii. Implement multiple different algorithms for the user to use.

      iii. (stretch goal): Create an AI model to generate the path(s)

      iv. Create cost algorithms to use for optimizing routes

      v. (stretch goal): multiple routes/algorithms calculated simultaneously using multithreading

2. Project Boundaries

   a. Included

      i. Project will only handle .csv files

      ii. Find routes for a single plane

      iii. (stretch goal): Add more planes to path finding

      iv. Planes will visit every node

v.   (stretch goal): Planes can be directed to only *need* to visit a set of nodes

b. Excluded

     i.   Will not handle real-time updates to locations (will have to rerun the algorithms to "update" it)

     ii.   Will not check for hazards (ie, other planes or inclement weather)

3. Required Resources

a. Technical

     i.   Python programming language

     ii.   Javascript/HTML/CSS

     iii.   Flask, React, NetworkX/PlantUML

     iv.   Pytest for writing test cases.

b. Data

     i.   Handle input .csv files which we use to populate our own graph with weights and locations.

     ii.   Test data will be a single .csv file generated by either hand or another algorithm.

     iii.   1gb of storage should be the maximum unless graphs given are larger than that (not including AI model if we create one)

c. Computing

     i.   Should be able to be used on small/old CPU's for basic and non-time essential workloads, would be faster with better/modern CPU's, a moderate to good GPU will be needed only if an AI model is used.

ii. Cloud services should *not* be needed, but could be used if the customer would like to

iii. External software licenses should *not* be needed