Problem 01
NO ACCIDENT PLEASE

Solution:

To ensure that flights path should not intersect we can use the method name “Graph Coloring algorithm” which we can combined with “Path finding algorithm”.

- Where coordinates of each airport will be represented as nodes
-And paths of flights will be represented as edges.

Use of graph coloring algorithm in this approach:

We use graph coloring to color different paths to ensure not path intersect each other, it means each color represent each flight path which will not intersect different flight path.

This can we done using greedy graph coloring algorithm to ensure that no to nodes share the same color , otherwise there would be intersection of edges.

Steps :

01) Create graph G= (V,E) where V are vertices and E are edges.
02) Graph coloring Use a greedy algorithm to color the graph:
    1. Initialize all vertices with no color.
    2. For each vertex v in the graph, assign the smallest available color that has not been used by its adjacent vertices.
    3. Ensure that no two adjacent vertices (connected directly by a flight segment) share the same color.

03) Path defining with constraints:

1. For each flight, use a path finding algorithm to find a path from the starting airport to the destination.
2. Modify the path finding algorithm to avoid edges that are already used by another flight (based on colors).

Pseudo code of given solution is attached in folder named with file: NoAccidentPlease.java

Given solution can be more optimized… but some what near soultion is provided

References:

https://www.geeksforgeeks.org/graph-coloring-set-2-greedy-algorithm/

https://neo4j.com/docs/graph-data-science/current/algorithms/pathfinding/