This java program contains 9 classes which solve the given problem using the Dijkstra algorithm.

Firstly, an ArrayList was created in which all of the constraints were stored by calling the getConstraints() function. The function simply extracts all of the constraints from a txt file and sends them back to the mentioned ArrayList while also storing them into the Constraints class. This class represents constraints between two places (destination and source) and includes the probability that they might occur.

Afterwards we extract all the places and store them in a class called Places which helps in managing them and extracting the needed information.

6 graphs were created using the Graph class. This class represents a graph data structure with vertices and edges. It includes methods for managing vertices, adding edges and checking for constraints. This class helps in building and managing graphs needed for the program.

To support our class Graph, classes Vertex and Distance were made. Vertex includes information about the vertex's name, its neighboring vertices and the corresponding travel times to those same neighbors.

The class Distance represents a combination of a Vertex and an associated distance. Two getters are present which are used to return the needed object.

After creating our graphs in the main program, we call the output() function which will write out the results of applying Dijkstra's algorithm on a graph to a file. It iterates through all pairs of source and destination vertices in the graph, calculates the shortest path and writes the results to the specifies graph.

The Dijkstra's algorithm was implemented in a class called Dijkstra, which finds the shortest path in a graph. It utilizes a priority queue which is implemented as an ArrayList of Distance objects to efficiently go through vertices based on their current distances from the starting vertex. An additional helping function was added, getMinIndex, that finds the index of the Distance object with the minimum distance.

To make sure that the program works as it is needed, we implemented various tests to validate the functionality of the Dijkstra algorithm implementation.

Writing this program was challenging. I searched the internet for hours trying to gather insight for solving the problem at hand, looking at examples, graph theory and similar. I asked countless questions and asked for help from my colleagues and with their support, tips and logic the code was finally written.