# User manual

1. The programming language used for the 2$^{nd}$ project (shortest path implementation between 2 cities) is Python.

2. The IDE used is PyCharm. The Community edition is free and can be downloaded at this link: https://www.jetbrains.com/pycharm/download/

3. The commented code script with the implementation is in the file *commented.py*.

4. One of the useful Python packages, NetworkX (documentation can be found here: https://networkx.org/ ), was imported in the script for the creation and analysis of the network graph.

5. Excel files, *cityName* and *FromTo*, can also be found in the project folder. Within the script, they are opened for reading and they contain city IDs as nodes of the graph plotted in the script, city names as attributes of city nodes, and edges (links between city nodes).

6. Once the *commented.py* file is opened in the IDE, we can right-click the opened file and select the 'Run' command or use a keyboard shortcut Ctrl + Shift + F10 to run the code.

7. Since the code contains a few tests where the print( ) function was used for checking whether the correct values will be printed, the following is displayed:

- Number of nodes contained in the graph
- Number of edges contained in the graph
- A list of city IDs
- A list of edges (in the form of links between city IDs)
- Fundamental information about the graph (type, number of nodes, number of edges, average degree)
- A list of city IDs and the corresponding city names

8. Finally, there will be a prompt ("Enter source"), where a name of any city from the list displayed above should be entered (with a first letter of the city name written as a capital letter, since the input required is case-sensitive).

9. After that, Enter key should be pressed, displaying another prompt ("Enter destination"), where a name of another city should be entered in the same manner as the first city.

10. After pressing Enter key again, the shortest path between the two entered cities will be displayed, in the form of a sequence of IDs of the cities we go through in order to reach the destination from the source city. Also, a distance/length of this path will be displayed, representing the fewest number of edges required to reach the destination from the source city.