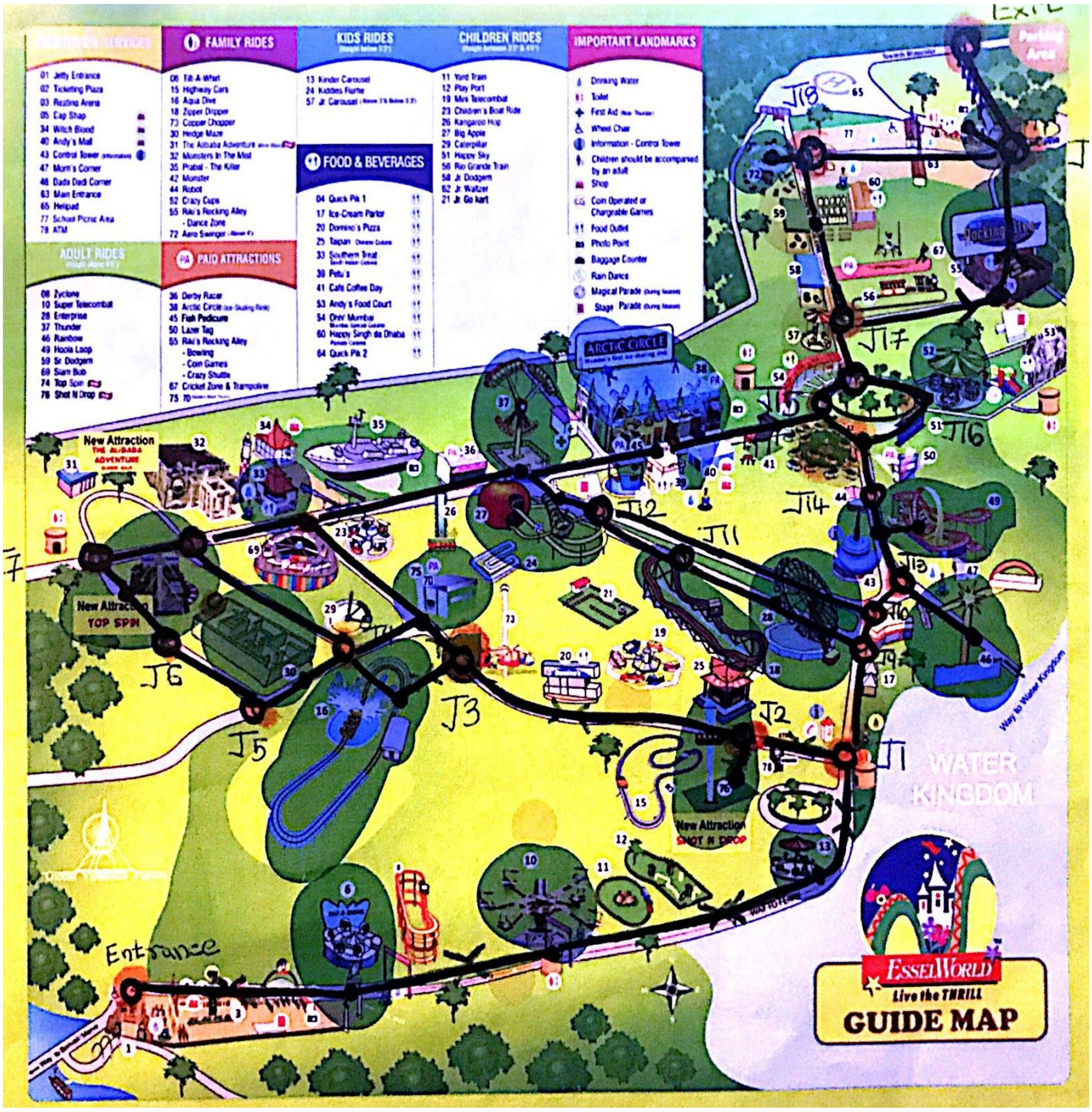
ASSIGNMENT 7

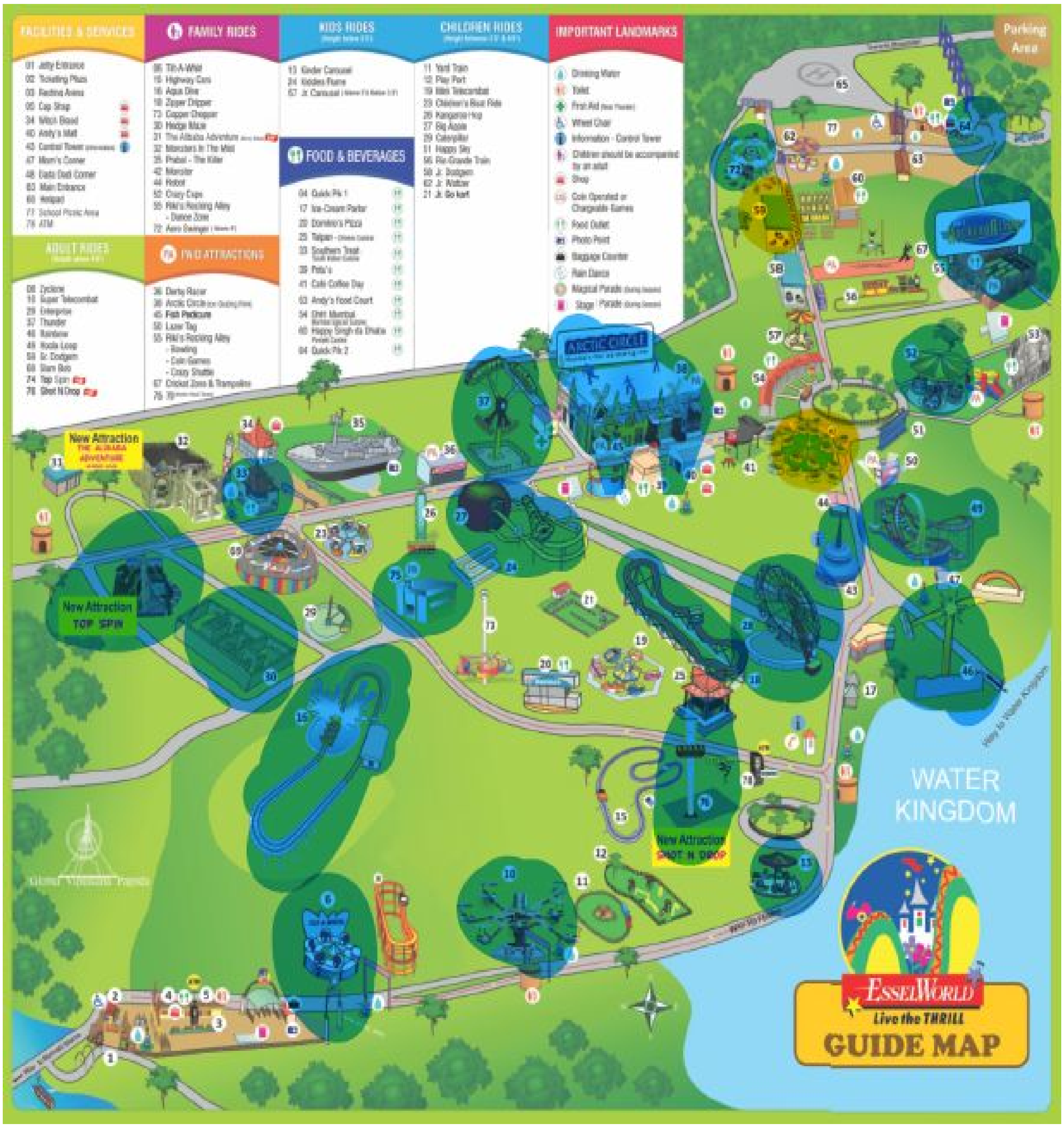
TEAM 5: Saloni Agrawal, Varun Kasbekar, Xiang Chen, Yujia Wang

- We have chosen the map of **ESSEL WORLD**, MUMBAI, INDIA
- We have selected 23 attractions from the map.
- We've also created 2 extra points named entrance and exit.
- Also, we tried to focus on the entire map rather than choosing a specific part.

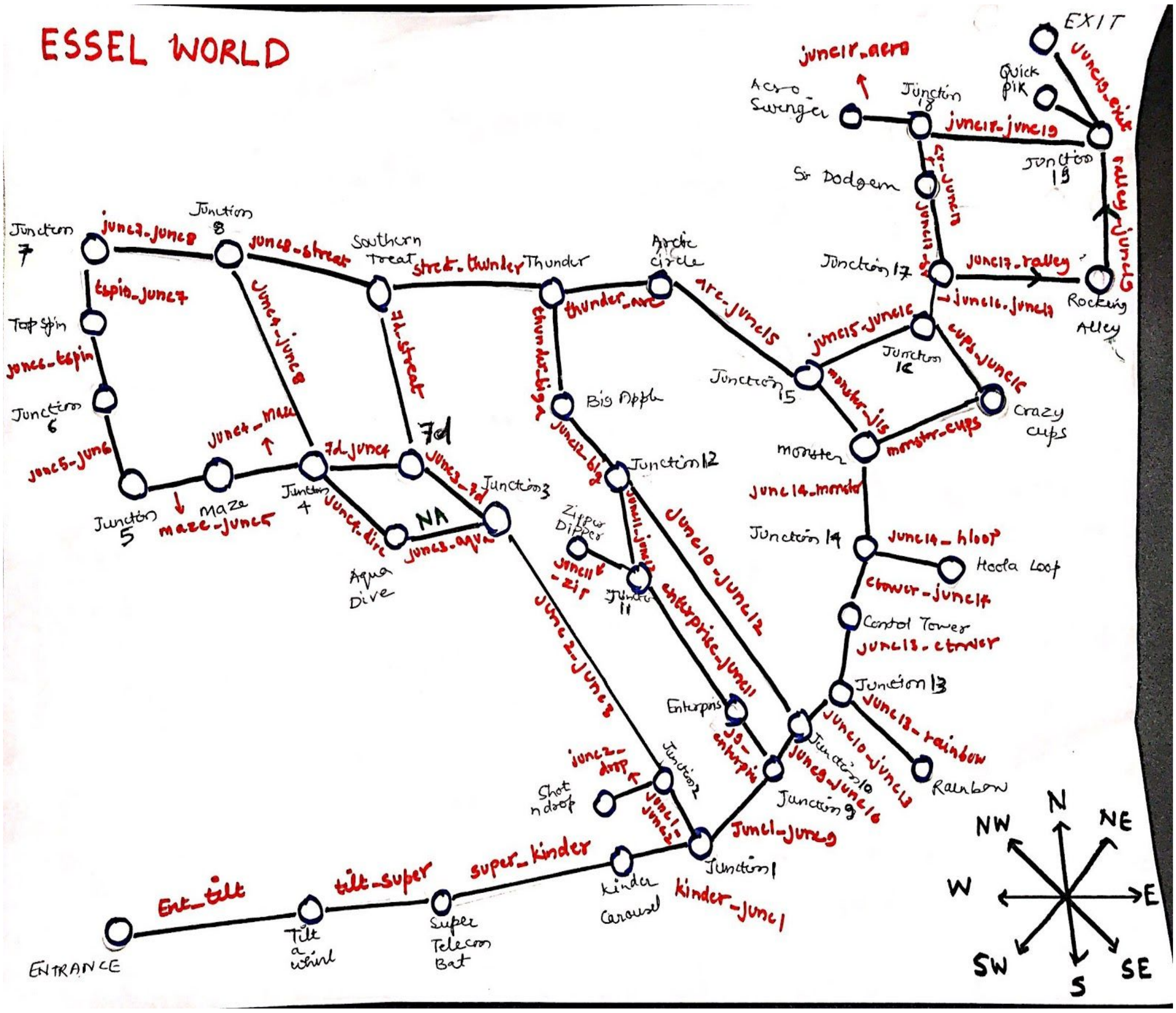
The below image shows the map with all the pathways.



This image shows highlighted attractions.



The below image shows the network graph with all the nodes and the edges.



- We started with choosing the Amusement Park. Then we firstly determined which attractions to cover such that the whole map is traversed. We selected all the possible types of attractions including Adult rides, Family rides, Children rides, Food and beverages corner, etc.
- Then we made the edges that is the path from each selected attraction and then we created nodes and edges for the map.
- All this work made our data in the form of a CSV file containing a From node which will be the starting point of the user, a To node which corresponds to the destination of the user, an edge column specifying the road user will take, a weight column for the distance between start point and the destination, a direction column for the direction and bi-directional column specifying

whether the road is unidirectional or bidirectional and at last an accessible column to say whether it is accessible for handicapped or not.

- We also made an additional file named attractions, which includes the name of the attraction, its type that whether it is adult or children or other type of attraction, the minimum height required to visit that attraction and the average weight time at that attraction.
- We then assigned weight and distance to the graph according to the distance between the attractions and the directions based on Earth's direction system.
- A number of extra junctions were needed so as to determine the path between various attractions.
- There were a number of slides which were not on the main road so we have created additional junctions that takes a direct path to reach those slides.
- The program is created such that even handicapped people can access it.
- The path from **Junction 3 to Aqua Dive** is **not accessible** for handicapped because it has stairs so the handicapped people can go to that slide from other way which goes from **Junction 4 towards Aqua Dive**.
- Also the path from **Junction 17 to Rocking Alley** and **Rocking Alley to Junction 19** is **unidirectional.**
- The program is coded such that even people with vision disability can access it by the **SOUND** feature we have added.
- The program is designed so as to compute the shortest distance along with the route to reach one attraction from the other.
- If the user enters a node which is not given in the list displayed at the start of the program, then the user gets an error message saying that the node is not present in the graph.
- We have used Djikstra's algorithm to determine the shortest path between 2 attractions.
- The program also satisfies the concept of object oriented classes and doctests are implemented for all the functions we have created.
- At the end of the program an automated test displays whether all the other nodes in the graph are reachable from the specified node.
- The automated test is incorporated in such a manner that it does not affect the user so once when the user enters his/her destination and when the program displays all the required things for the user, then the user is prompted with a message- Do you wish to travel to another destination or Quit?
- So when the user says Quit after displaying Thank you, after that the automated test can be checked by entering the node to be checked.
- The most challenging part of this assignment was making the program object oriented and then incorporating the doctests for those functions.

We made a slack group to communicate and we conducted bi-weekly meetings in which everyone updated the group with their assigned work and then we discussed the further work to be done.
We also use to discuss the updates after the class lecture.

Work done by Team members

- ❏ Saloni, Varun, Yujia and Xiang have created the data. All the naming of the nodes and edges have been done by the entire team. Further we created an Excel file esselworld.csv to store all the data.
- ❏ Xiang created the Git repository.
- ❏ Yujia and Varun created the attractions.csv where information about the type, average waiting time and minimum height required.
- ❏ Graphs were created by Saloni. All the nodes and paths were highlighted by her and Xiang.
- ❏ The functionality of the sound for disabled people was incorporated by Varun.
- ❏ Coding was done by the entire team in parts. We segmented complex problems into small parts and implemented the solutions.
- ❏ Varun and Saloni created the first draft of the code to find shortest path between the start and the destination. We ensured that all nodes were connected and the distance between them was accurate.
- ❏ Yujia created the functions for the code.
- ❏ Yujia and Xiang converted the code to object oriented. Yujia implemented the unidirectional part.
- ❏ Varun and Saloni implemented accessibility part in the code.
- ❏ Varun updated the functions and added additional functions.
- ❏ Saloni added the functionality to check if a node is present in the graph or not.
- ❏ Xiang wrote the initial doctests along with documentation.
- ❏ Varun converted the doctests into object oriented format and updated the documentations for doctests.
- ❏ Varun created the automated function to check reachability of every node. This function asked user to input a specific attraction and then checked whether there was a valid path from that attraction to every other node.
- ❏ Finally the report was created by Varun and Saloni.
- ❏ All the team members have periodically committed changes to github.