وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا

Software Engineering Intern

Task 02

Topology API

1. Use Case

Provide the functionality to access, manage and store device topologies.

2. Solution

Write an API library which does the following:

1. Read and write topologies to and from disk.
2. Stores multiple topologies in memory.
3. Execute operations on topologies.

3. Topology Specification

A topology is a set of electronic components that are connected together.
Example JSON file (topology.json):-

4. Functional Requirements

Provide the functionality to:

1. Read a topology from a given JSON file and store it in the memory.
2. Write a given topology from the memory to a JSON file.
3. Query about which topologies are currently in the memory.
4. Delete a given topology from memory.
5. Query about which devices are in a given topology.
6. Query about which devices are connected to a given netlist node in a given topology.

5. Non-Functional Requirements

1. Implementation must be done in an object-oriented manner (encapsulation, inheritance, polymorphism).
2. Choose a suitable programming language (other than Python) and justify your choice.
3. Use managed pointers (depends on programming language).
4. Using a managed build tool is a bonus (Gradle, Maven, ..).
5. Documentation on API level is a must.
6. Documentation on class level is a bonus.
7. Automatic testing on API level is a must.
8. Automatic testing on class level is a bonus.
9. Check your code with a code analysis tool of your choice.
10. Implement the requirements exactly, more is as bad as less.
11. Use version control to publish your code.
12. Make any other design choices as you see fit to the requirements and write them in your solution.

6. Example API

You can implement your API in any way. Here is an example of how it may look like (API.h).

```
Result readJSON(FileName);
Result writeJSON(TopologyID);
TopologyList queryTopologies();
Result deleteTopology(TopologyID);
DeviceList queryDevices(TopologyID);
DeviceList queryDevicesWithNetlistNode(TopologyID, NetlistNodeID);
```