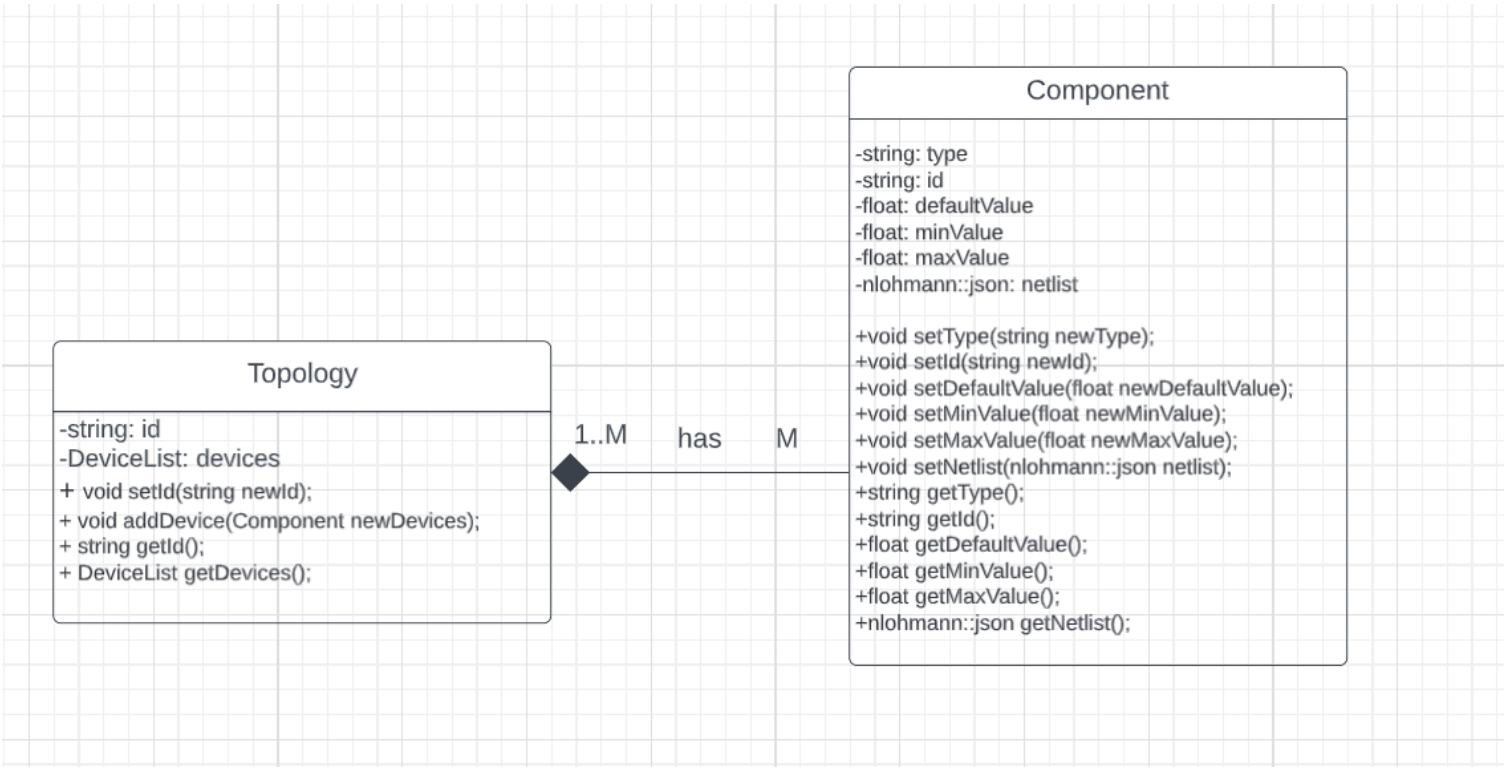
1- Problem Description

Write an API library that reads and writes topologies to and from disk, stores multiple topologies in memory and executes operations on topologies.

2- Functional Requirements

- Read a topology from a given JSON file and store it in the memory.
- Write a given topology from the memory to a JSON file.
- Query about which topologies are currently in the memory.
- Delete a given topology from memory.
- Query about which devices are in a given topology.
- Query about which devices are connected to a given netlist node in
- a given topology.

3- Class diagram



4- Tools used

- nlohmann/JSON (JSON for modern C++) used to read/write from/to JSON file JSON C++
- C++
- Build Tool: CMake
- Version Control: GitHub
- Static Code Analysis: CppCheck
  I used C++ because it supports OOP, faster than other popular languages and I can code smoothly by C++

5- API Documentation

```
bool Api::readJSON(string FileName)
```

It takes file name and its path. It reads JSON file and stores it in Topology object in memory

Steps:

1- first, we are trying to read file then check if it read successfully or not

2- if not read successfully, it will return false

3- if it is read successfully, parsing JSON file into JSON object called jsonFile

4- make new Topology object called topology

5- set topology id equals id in jsonFile

6- iterate through all devices in jsonFile then in every iteration we make Component object called device

7- setting type and id of device

8- checking type and set default, min and max value of device

9- set netlist of device then add device object to topology

10- I made netlist as JSON object so we can iterate through every key-value in it so we can check if netlist id in their or not

11- after adding all devices we will add this topology to list contains all topologies

```
bool Api::writeJSON(string topologyID)
```

It takes id of topology that we want to write it in JSON file in disk

1- we iterate through all topologies to find if topology with that id is exist or not

2- if we found that topology we will store it in topology object

3- if not found we will return false

4- after founding topology we will create JSON object that store in it id of topology

5- we will iterate through all devices then set type and id of device in JSON object

6- we will set default, min and max values and netlist of JSON object

7- we will write this JSON object in JSON file then return true

```
TopologyList Api::queryTopologies()
```

Used to return list contains all topologies in memory

```
bool Api::deleteTopology(string topologyID)
```

Used to delete specific topology from list of topologies

1- we iterate through all topologies in list then check every topology's id if it equals given id

2- if found we delete it then return true

```
DeviceList Api::queryDevices(string topologyID)
```

Used to return all devices in given topology

1- we iterate through all topologies and if topology's id equals given id then we will return devices list

2- if this id not found we will return empty vector

```
DeviceList Api::queryDevicesWithNetlistNode(string topologyID, json NetlistNodeID)
```

It returns all devices that have given netlist id

1- we iterate through all topologies and check if given id equals topology's id

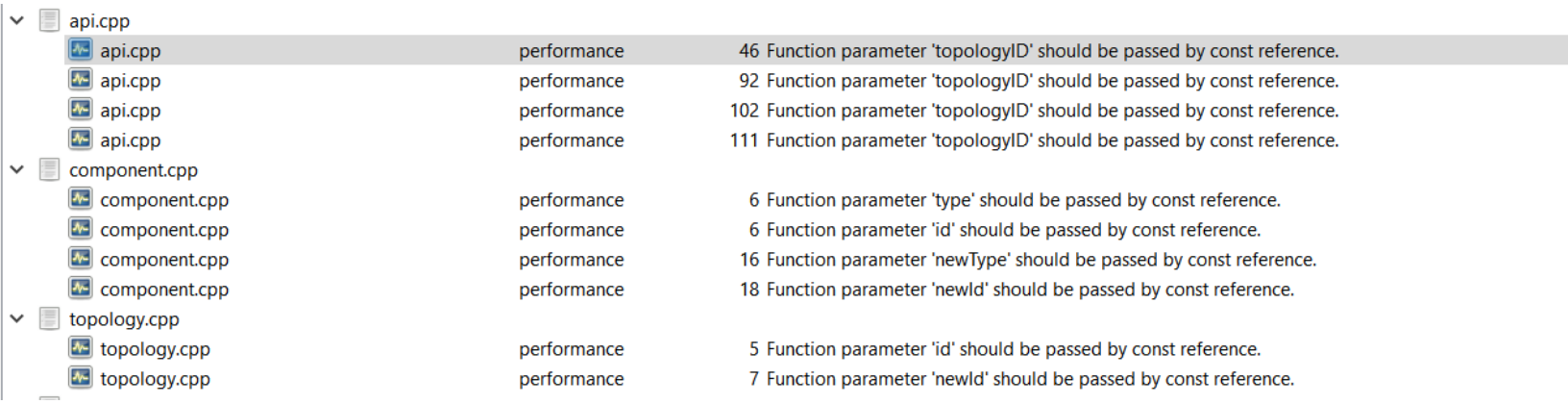
2- if equals then store that topology in new Topology object

3- we will iterate through all devices in that topology and store its netlist in JSON netlist

4- we will iterate through all key-value in topology's netlist

5- if key-value equals given JSON will add this device and continue iterating through remaining devices

Code Analysis:



api.cpp			
api.cpp	performance	46	Function parameter 'topologyID' should be passed by const reference.
api.cpp	performance	92	Function parameter 'topologyID' should be passed by const reference.
api.cpp	performance	102	Function parameter 'topologyID' should be passed by const reference.
api.cpp	performance	111	Function parameter 'topologyID' should be passed by const reference.
component.cpp			
component.cpp	performance	6	Function parameter 'type' should be passed by const reference.
component.cpp	performance	6	Function parameter 'id' should be passed by const reference.
component.cpp	performance	16	Function parameter 'newType' should be passed by const reference.
component.cpp	performance	18	Function parameter 'newId' should be passed by const reference.
topology.cpp			
topology.cpp	performance	5	Function parameter 'id' should be passed by const reference.
topology.cpp	performance	7	Function parameter 'newId' should be passed by const reference.

We found that when we made analysis on code

I prefer not to edit these things because if we made these parameters passed by reference that if we pass it in function when returning from function that variable will be invalid