# Use Case Design Specification

## **360-degree scan for a <node name> with depth limit <1/2/3>**

Step 1: Use the following cypher query to do the scan with max level value inputted by user as 1.

#Spinning Tree Algorithm

**match (n:Process {name:'CSM'})**

**CALL apoc.path.spanningTree(n, {**

**minLevel: 1,**

**maxLevel: 1 (#value taken from user input)**

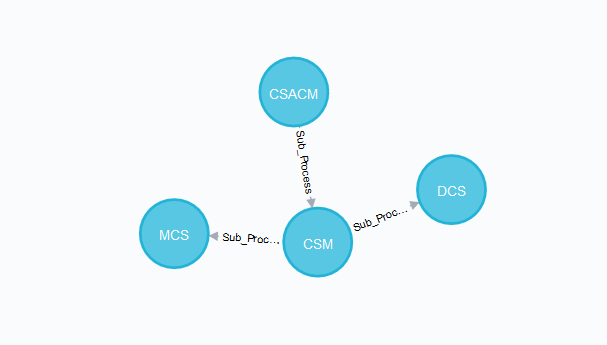
**})**

**YIELD path**

**RETURN path;**

Step 2: Display the following Graph result in the Web UI with one summary statement:

**360-Degree Scan Result**: Please refer to the graph for the 360-degree scan for node CSM (Customer Self Management) with depth limit 1, if you wish to view more information about this node, please query again using a higher depth limit ranging from 1 to 3



Step3: Use the same cypher query to to do the scan with max level value inputted by user as 2 or 3

#Spinning Tree Algorithm

**match (n:Process {name:'CSM'})**

**CALL apoc.path.spanningTree(n, {**

**minLevel: 1,**

**maxLevel: 2 (#value taken from user input)**

**})**

**YIELD path**

**RETURN path;**

Step 4: Display the following Graph result in the Web UI with one summary statement and a table that illustrate the step cost details

**360-Degree Scan Result**: Please refer to the following graph for the 360-degree scan for node CSM (Customer Self Management) with depth limit 1, if you wish to view more information about this node, please query again using a higher depth limit ranging from 1 to 3

