**FUNCTIONAL**

**SOFTWARE**

**TEST PLAN**

**for**

**Encost Smart Graph Project**

**Version 1.0**

|  |  |  |
| --- | --- | --- |
| **Stu** | **dent** | **Name** |

**Prepared by:**

**SoftFlux Engineer**

**SoftFlux**

**April 4, 2023**

Contents

[1 Introduction/Purpose 3](#_Toc4219)

[1.1 Purpose 3](#_Toc4220)

[1.2 Document Conventions 3](#_Toc4221)

[1.3 Intended Audience and Reading Suggestions 3](#_Toc4222)

[1.4 Project Scope 3](#_Toc4223)

[2 Specialized Requirements Specification 3](#_Toc4224)

[3 Black-box Testing 4](#_Toc4225)

[3.1 Requirement #1 4](#_Toc4226)

[3.2 Requirement #2 4](#_Toc4227)

[3.3 Requirement #3 4](#_Toc4228)

[3.4 Requirement #n 4](#_Toc4229)

[4 White-box testing 4](#_Toc4230)

[4.1 *<*Requirement/Feature Name*>* Pseudocode 5](#_Toc4231)

[4.2 Branch Coverage Testing 5](#_Toc4232)

[5 Mutation Testing 5](#_Toc4233)

[5.1 Mutant #1 5](#_Toc4234)

[5.2 Mutant #2 5](#_Toc4235)

[5.3 Mutant #3 5](#_Toc4236)

[5.4 Mutant #4 5](#_Toc4237)

[5.5 Mutation Score 5](#_Toc4238)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason for Changes | Version |
|  | 15/04/2023 | Intro/Purpose, Black box Testing | 1.0 |
|  | 17/04/2023 | Black box testing | 1.1 |
|  | 18/04/2023 | White box testing, Mutation testing | 1.2 |
|  | 27/04/2023 | Black box testing | 1.3 |
|  | 29/04/2023 | Black box testing | 1.4 |

# Introduction/Purpose

## Purpose

This document is the Functional Software Test Plan for the Encost Smart Graph Project (ESGP) for the company Encost which is made based on the SRS and SDS 1 documents. The purpose of this document is to aid in the implementation and testing.

## Document Conventions

This document uses the following conventions:

ESGP: Encost Smart Graph Project

GSL: GraphStream Library

SRS: Software Requirement Specification

SDS: Software Design Specification

## Intended Audience and Reading Suggestions

* Developer/Project Manager: Uses this document to help implement the software.
* Tester: Uses this document to understand what the program should do, and what should be tested.

## Project Scope

SoftFlux is only responsible for the base version and maintenance of the software. This version is only in English and includes the functionalities:

* Handling input and output
* User login
* Graph visualisation
* Data processing and statistics

The ESGP does not integrate with any hardware.

# Specialized Requirements Specification

* The black box testing is testing both UI and functionality tests.
* The building a graph data type test also tests for the graph visualisation test.
* The file path for the Encost Smart Homes Dataset is stored within the backend of the application which is located locally.

# Black-box Testing

## Categorizing Users

Level of test: Unit (Recommended to use JUnit to automate test)

Test technique: Decision table

Test inputs:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Conditions | Case 1 | Case 2 | Case 3 | Case 4 |
| Input | 1 | 2 | Any number that is not 1 or 2 | Blank/NULL |
| Output | Community User Output and ESGP Feature Options Displayed | Encost User Output and ESGP Account Login Displayed | Error Message Displayed for wrong input | Error Message Displayed for no input |

Test Case 1: Number 1 = Community User Output and ESGP Feature Options Displayed

Test Case 2: Number 2 = Encost User Output and ESGP Account Login Displayed

Test Case 3: Any number that is not 1 or 2 = Error Message

Test Case 4: Blank/Null = Error Message

## ESGP Account Login

Level of test: Unit (Recommended to use JUnit to automate test)

Test technique: Decision table

Test inputs:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Conditions | Case 1 | Case 2 | Case 3 | Case 4 |
| Username | Incorrect | Incorrect | Correct | Correct |
| Password | Incorrect | Correct | Incorrect | Correct |
| Output | Error Message Displayed for Incorrect Username and Password | Error Message Displayed for Incorrect Username and Password | Error Message Displayed for Incorrect Username and Password | Welcome Message Output and ESGP Feature Options Displayed |

Test Case 1: Incorrect Username and Password = Error Message

Test Case 2: Incorrect Username and Correct Password = Error Message

Test Case 3: Correct Username and Incorrect Password = Error Message

Test Case 4: Correct Username and Password = Welcome Message Output and ESGP Feature Options Displayed

## ESGP Feature Options

Level of test: Integration with GSL for the Graph Visualisation UI (Recommended to use SmartBear TestComplete to automate test)

Test technique: Decision table

Test inputs:

**Community User**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Conditions | Case 1 | Case 2 | Case 3 | Case 4 |
| Input | 1 | X | Anything that is not 1 or X | Blank/NULL |
| Output | Graph Visualisation UI Displayed | Exit/Close the Application | Error Message for wrong input | Error Message for no input |

Test Case 1: Number 1 = Graph Visualisation UI Displayed

Test Case 2: String X = Exit/Close Application

Test Case 3: Anything that is not 1 or X = Error Message

Test Case 4: Blank/Null = Error Message

**Encost User**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Conditions | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 | Case 6 |
| Input | 1 | 2 | 3 | X | Anything that is not 1,2,3 or X | Blank/NULL |
| Output | Custom Dataset Full File Path Prompt Displayed | Graph Visualisation UI Displayed | Summary Statistics Displayed | Exit/Close the Application | Error Message for wrong input | Error Message for no input |

Test Case 1: Number 1 = Custom Dataset Full File Path Prompt Displayed

Test Case 2: Number 2 = Graph Visualisation UI Displayed

Test Case 3: Number 3 = Summary Statistics Displayed

Test Case 4: String X = Exit/Close Application

Test Case 5: Anything that is not 1, 2, 3 or X = Error Message

Test Case 6: Blank/Null = Error Message

## Loading the Encost Smart Homes Dataset

Level of test: Integration with GSL to add device to graph (Recommended to use SmartBear TestComplete to automate test)

Test technique: Decision table

Test inputs:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Conditions | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 | Case 6 | Case 7 | Case 8 |
| CSV Format | Correct | Correct | Correct | Correct | Correct | Correct | Correct | Correct |
| Device type is a router? | Yes | No | No | Yes | NULL | NULL | NULL | NULL |
| Is the device’s router in the graph? | NULL | Yes | No | NULL | Yes | No | NULL | NULL |
| Does the device have the router? | Yes | NULL | NULL | No | NULL | NULL | Yes | No |
| Output | Add device to graph | Add device to graph | Add device to unlinked devices list | Error message for wrong input | Error message for wrong input | Error message for wrong input | Error message for wrong input | Error message for wrong input |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Conditions | Case 9 | Case 10 | Case 11 | Case 12 | Case 13 | Case 14 | Case 15 | Case 16 |
| CSV Format | Incorrect | Incorrect | Incorrect | Incorrect | Incorrect | Incorrect | Incorrect | Incorrect |
| Device type is a router? | Yes | No | No | NULL | NULL | NULL | NULL | NULL |
| Is the device’s router in the graph? | NULL | Yes | No | Yes | No | NULL | NULL | NULL |
| Does the device have the router? | Yes | NULL | NULL | NULL | NULL | Yes | No | NULL |
| Output | Error message for CSV format | Error message for CSV format | Error message for CSV format | Error message for CSV format | Error message for CSV format | Error message for CSV format | Error message for CSV format | Error Message for CSV format |

|  |  |
| --- | --- |
| Conditions | Case 17 |
| CSV Format | Blank/NULL |
| Device type is a router? | NULL |
| Is the device’s router in the graph? | NULL |
| Does the device have the router? | NULL |
| Output | Error Message for no input |

The format for CSV: device ID, date connected, device name, device type, household ID, router ID, can send, can receive

Example 1: EWR-1234,01/04/22,Encost Router 360,Router,WKO-1234,-,Yes,Yes

Example 2: ELB-4567,01/04/22,Encost Smart Bulb B22 (multi colour),Light bulb,WKO1234,EWR-1234,No,Yes

Test Case 1: CSV format correct, device type is a router and device have the router = Add device to graph

Test Case 2: CSV format correct, device type is not a router and device’s router in the graph = Add device to graph

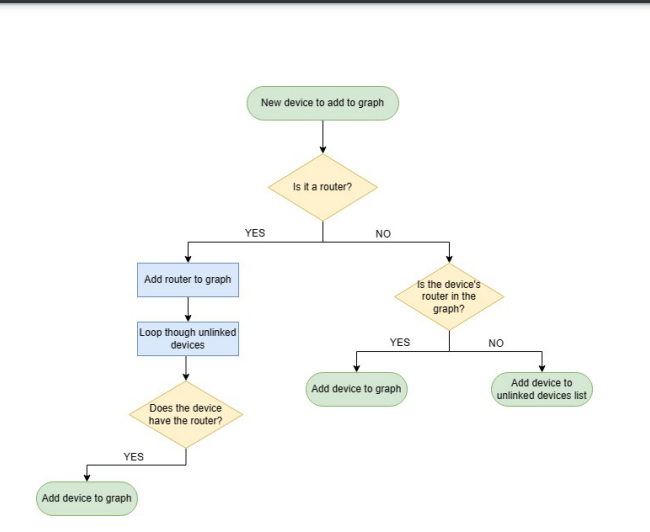
Test Case 3: CSV format correct, device type is not a router and device’s router not in the graph = Add device to unlinked devices list

Test Cases 4-15: This test cases are impossible to check because we need to use the CSV format condition to be correct in order to start checking the other conditions

Test Case 16: Incorrect/Missing data = Error Message

Test Case 17: Blank/Null = Error Message

The testing for loading the encost smart homes dataset is following the flow diagram below.



## Categorizing Smart Home Devices

Level of test: Unit (Recommended to use JUnit to automate test)

Test technique: Decision table

Test inputs:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Conditions | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
| Device Type | Router | Extender | Hub/Controller | Light Bulb | Strip Lighting |
| Output | The productCategory attribute will be Encost Wifi Routers | The productCategory attribute will be Encost Wifi Routers | The productCategory attribute will be Encost Hubs/Controllers | The productCategory attribute will be Encost Smart Lighting | The productCategory attribute will be Encost Smart Lighting |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Conditions | Case 6 | Case 7 | Case 8 | Case 9 |
| Device Type | Other Lighting | Kettle | Toaster | Coffee Maker |
| Output | The productCategory attribute will be Encost Smart Lighting | The productCategory attribute will be Encost Smart Appliances | The productCategory attribute will be Encost Smart Appliances | The productCategory attribute will be Encost Smart Appliances |

|  |  |  |  |
| --- | --- | --- | --- |
| Conditions | Case 10 | Case 11 | Case 12 |
| Device Type | Washing Machine/Dryer | Refrigerator/Freezer | Dishwasher |
| Output | The productCategory attribute will be Encost Smart Whiteware | The productCategory attribute will be Encost Smart Whiteware | The productCategory attribute will be Encost Smart Whiteware |

|  |  |  |
| --- | --- | --- |
| Conditions | Case 13 | Case 14 |
| Device Type | Incorrect | Blank/NULL |
| Output | Error Message for wrong data type | Error Message for no input |

Correct Device Category and Device Types:

A picture containing text, screenshot, font, number

Description automatically generated

Test Case 1: Correct Device Type = productCategory attribute will have the above screenshot device category depending on what device type it is

Test Case 2: Incorrect Device Type (Wrong Spelling, does not exist) = Error Message

Test Case 3: Blank/Null = Error Message

## Building a Graph Data Type

Level of test: Integration with GSL for interaction with graph (Recommended to use SmartBear TestComplete to automate test)

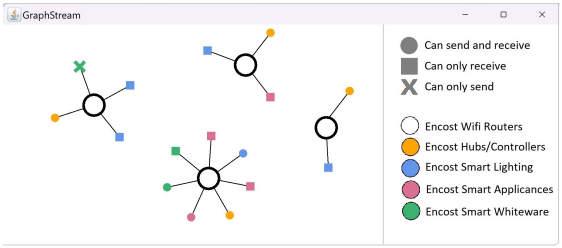
Test technique: Decision table

Test inputs:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Conditions | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 | Case 6 |
| Device Object | Router | Hubs/Controller | Smart Lighting | Smart Appliance | Smart Whiteware | Smart Appliance |
| Router in the graph | No | No | No | No | No | No |
| Can Send | Yes | Yes | No | No | No | Yes |
| Can Receive | Yes | Yes | Yes | Yes | Yes | No |
| Output | A circle with white colour will appear in the graph that isn’t connected with anything | A circle with orange colour will appear in the graph that isn’t connected with anything | A rectangle with blue colour will appear in the graph that isn’t connected with anything | A rectangle with pink colour will appear in the graph that isn’t connected with anything | A rectangle with green colour will appear in the graph that isn’t connected with anything | A “X” shape with pink colour will appear in the graph that isn’t connected with anything |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Conditions | Case 7 | Case 8 | Case 9 | Case 10 | Case 11 | Case 12 | Case 13 |
| Device Object | Hubs/Controller | Smart Lighting | Smart Appliance | Smart Whiteware | Smart Appliance | Incorrect | NULL |
| Router in the graph | Yes | Yes | Yes | Yes | Yes | NULL | NULL |
| Can Send | Yes | No | No | No | Yes | NULL | NULL |
| Can Receive | Yes | Yes | Yes | Yes | No | NULL | NULL |
| Output | A circle with orange colour will appear in the graph that is connected using a black line to the white colour circle(SEE GRAPH VISUALISATION FIGURE) | A rectangle with blue colour will appear in the graph that is connected using a black line to the white colour circle(SEE GRAPH VISUALISATION FIGURE) | A rectangle with pink colour will appear in the graph that is connected using a black line to the white colour circle(SEE GRAPH VISUALISATION FIGURE) | A rectangle with green colour will appear in the graph that is connected using a black line to the white colour circle(SEE GRAPH VISUALISATION FIGURE) | A “X” shape with pink colour will appear in the graph that is connected using a black line to the white colour circle(SEE GRAPH VISUALISATION FIGURE) | Error Message for wrong device object | Error Message for no device object |

Graph Visualisation with all the different devices objects. (REMEMBER TO ADD FIGURE INTO Decision table)



Test Case 1: Device Object Router, No Router in graph, Can Send, Can Receive = White colour circle with no connections in graph

Test Case 2: Device Object Hubs/Controller, No Router in graph, Can Send, Can Receive = Orange colour circle with no connections in graph

Test Case 3: Device Object Smart Lighting, No Router in graph, Can Receive = Blue colour rectangle with no connections in graph

Test Case 4: Device Object Smart Appliance, No Router in graph, Can Receive = Pink colour rectangle with no connections in graph

Test Case 5: Device Object Smart Whiteware, No Router in graph, Can Receive = Green colour rectangle with no connections in graph

Test Case 6: Device Object Smart Appliance, No Router in graph, Can Send = Pink colour “X” shape with no connections in graph

Test Case 7: Device Object Smart Lighting, Router in graph, Can Receive = Blue colour rectangle with a black line connection to white colour circle (Router)

Test Case 8: Incorrect Device Object = Error Message

## Calculating Device Distribution

Level of test: Unit (Recommended to use JUnit to automate test)

Test technique: Decision table

Test inputs:

getDeviceByCategory method testing:

|  |  |  |
| --- | --- | --- |
| Conditions | Case 1 | Case 2 |
| getDeviceByCategory(String) | Correct String(Look at Figure) | Incorrect String |
| Output | A list of devices where all devices are returned depending on what string was given. | Error Message for wrong getDeviceByCategory() |

getDeviceByType method testing:

|  |  |  |
| --- | --- | --- |
| Conditions | Case 1 | Case 2 |
| getDeviceByType(String) | Correct String(Look at Figure) | Incorrect String |
| Output | A list of devices where all devices are returned depending on what string was given. | Error Message for wrong getDeviceByType() |

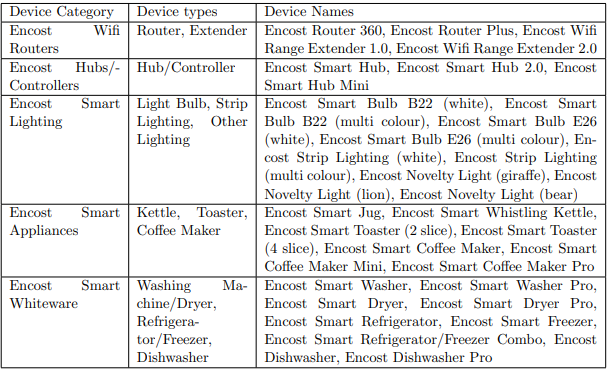
getDeviceByProduct method testing:

|  |  |  |
| --- | --- | --- |
| Conditions | Case 1 | Case 2 |
| getDeviceByProduct(String) | Correct String(Look at Figure) | Incorrect String |
| Output | A list of devices where all devices are returned depending on what string was given. | Error Message for wrong getDeviceByProduct() |

calculateDeviceDistribution method testing:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Conditions | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
| getDeviceByCategory(String) | Correct String(Encost Wifi Routers) | Incorrect String | Incorrect String | Correct String | Correct String |
| getDeviceByType(String) | Correct String(Router) | Incorrect String | Correct String | Incorrect String | Correct String |
| getDeviceByProduct(String) | Correct String(Encost Router 360) | Incorrect String | Correct String | Correct String | Incorrect String |
| Output | The number of Encost Wifi Routers, Router, Encost Router 360 are displayed in the format in Figure | Error message for wrong getDeviceByCategory(), getDeviceByType() and getDeviceByProduct() | Error message for wrong getDeviceByCategory() | Error message for wrong getDeviceByType() | Error message for wrong getDeviceByProduct() |

|  |  |  |  |
| --- | --- | --- | --- |
| Conditions | Case 6 | Case 7 | Case 8 |
| getDeviceByCategory(String) | Incorrect String | Incorrect String | Correct String |
| getDeviceByType(String) | Incorrect String | Correct String | Incorrect String |
| getDeviceByProduct(String) | Correct String | Incorrect String | Incorrect String |
| Output | Error message for wrong getDeviceByCategory() and getDeviceByType() | Error message for wrong getDeviceByCategory() and getDeviceByProduct() | Error message for wrong getDeviceByType() and getDeviceByProduct() |



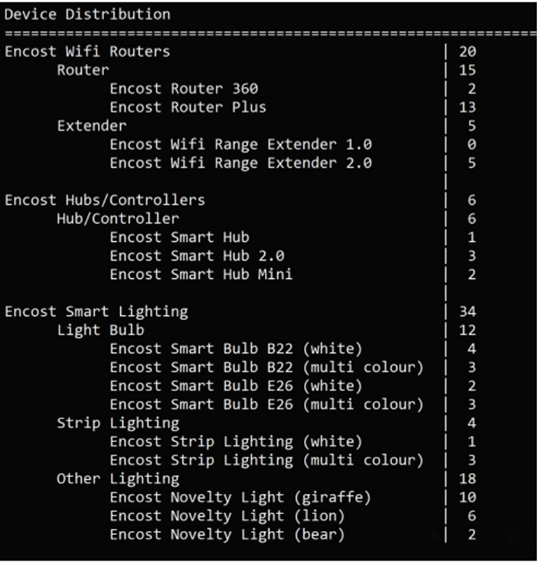
Device Distribution Output Example for test case 1:

A picture containing text, screenshot, font, line

Description automatically generated

Device Distribution Full Output Example:

This is the device distribution output if there are more device categories, types and products.



Incorrect String are string that are not in the figure and blank/null strings.

Test Case 1: getDeviceByCategory(Encost Wifi Router), getDeviceByType(Router), getDeviceByProduct(Encost Router 360) = The number of Encost Wifi Routers, Router, Encost Router 360 is displayed in the format in the example above

Test Case 2: Incorrect getDeviceByCategory(), getDeviceByType(Router), getDeviceByProduct(Encost Router 360) = Error Message

Test Case 3: getDeviceByCategory(Encost Wifi Router), Incorrect getDeviceByType(), getDeviceByProduct(Encost Router 360) = Error Message

Test Case 4: getDeviceByCategory(Encost Wifi Router), getDeviceByType(Router), Incorrect getDeviceByProduct() = Error Message

Test Case 5: Blank/Null = Error Message

# White-box testing

## Device Distribution Pseudocode

*Requirement 1: The system should use the information stored in the graph data structure to calculate the number of devices that exist in each device category*.

getDevicesByCategory(inputtedString) Pseudocode

INITIALISE list of devices called categoryDevicesList

FOR i = 0 to devices size - 1

IF devices productCategory attribute is same as inputtedString

ADD devices[i] into categoryDevicesList

END IF

ENDFOR

FOR j = 0 to unLinkedDevices size - 1

IF unLinkedDevices productCategory attribute is same as inputtedString

ADD unLinkedDevices[i] into categoryDevicesList

END IF

ENDFOR

RETURN categoryDevicesList

## Branch Coverage Testing

Test Case 1: inputtedString is Encost Smart Lighting

Test Case 2: inputtedString is Blank/NULL

Test Case 3: inputtedString is 12356

Test Case 4: inputtedString is $#$%^(\*^&

Test Case 5: inputtedString is Encost Smart Light

# Mutation Testing

## Mutant #1

INITIALISE list of devices called categoryDevicesList

FOR i = 0 to devices size - 2

IF devices productCategory attribute is same as inputtedString

ADD devices[i] into categoryDevicesList

END IF

ENDFOR

FOR j = 0 to unLinkedDevices size - 2

IF unLinkedDevices productCategory attribute is same as inputtedString

ADD unLinkedDevices[j] into categoryDevicesList

END IF

ENDFOR

RETURN categoryDevicesList

## Mutant #2

INITIALISE list of devices called categoryDevicesList

FOR i = 1 to devices size - 1

IF devices productCategory attribute is same as inputtedString

ADD devices[i] into categoryDevicesList

END IF

ENDFOR

FOR j = 1 to unLinkedDevices size - 1

IF unLinkedDevices productCategory attribute is same as inputtedString

ADD unLinkedDevices[i] into categoryDevicesList

END IF

ENDFOR

RETURN categoryDevicesList

## Mutant #3

INITIALISE list of devices called categoryDevicesList

FOR i = 0 to less than or equal to devices size - 1

IF devices productCategory attribute is same as inputtedString

ADD devices[i] into categoryDevicesList

END IF

ENDFOR

FOR j = 0 to less than or equal to unLinkedDevices size - 1

IF unLinkedDevices productCategory attribute is same as inputtedString

ADD unLinkedDevices[i] into categoryDevicesList

END IF

ENDFOR

RETURN categoryDevicesList

## Mutant #4

INITIALISE list of devices called categoryDevicesList

FOR i = 0 to more than devices size - 1

IF devices productCategory attribute is same as inputtedString

ADD devices[i] into categoryDevicesList

END IF

ENDFOR

FOR j = 0 to more than unLinkedDevices size - 1

IF unLinkedDevices productCategory attribute is same as inputtedString

ADD unLinkedDevices[i] into categoryDevicesList

END IF

ENDFOR

RETURN categoryDevicesList

## Mutation Score

devices list contains 3 Encost Wifi Routers, 2 Encost Hubs/Controllers, 4 Encost Smart Lighting, 1 Encost Smart Appliances and 5 Encost Smart Whiteware.

unLinkedDevices list contains 2 Encost Wifi Routers, 4 Encost Hubs/Controllers, 4 Encost Smart Lighting, 1 Encost Smart Appliances and 1 Encost Smart Whiteware.

|  |  |
| --- | --- |
| Input (inputtedString) | Output (categoryDevicesList) |
| Encost Smart Lighting | List of devices containing 8 items |
| Encost Smart Whiteware | List of devices containing 6 items |

Mutation Score: 3/4 (75%). Mutation #3 was not caught.