
FUNCTIONAL SOFTWARE TEST PLAN

for

Encost Smart Graph Project

Version 1.0

Prepared by: Joey Han
SoftFlux Engineer

SoftFlux

May 3, 2023

Contents

1	Introduction/Purpose	3
1.1	Purpose	3
1.2	Document Conventions	3
1.3	Intended Audience and Reading Suggestions	4
1.4	Testing Scope	4
2	Specialized Requirements Specification	4
3	Black-box Testing	5
3.1	Categorizing Users	5
3.1.1	All Users Decision Table	5
3.2	ESGP Account Login	5
3.2.1	Encost User Decision Table	6
3.3	ESGP Feature Options	6
3.3.1	Community User Decision Table	6
3.3.2	Encost User Decision Table	7
3.4	Loading the Encost Smart Homes Dataset	8
3.4.1	All Users Decision Table	8
3.5	Categorizing Smart Home Devices	11
3.5.1	All Users Decision Table	11
3.6	Building a Graph Data Type	14
3.6.1	All Users Decision Table	14
3.7	Calculating Device Distribution	19
3.7.1	Encost User Decision Table	19
4	White-box testing	25
4.1	Device Distribution Pseudocode	26
4.2	Branch Coverage Testing	26
5	Mutation Testing	27
5.1	Mutant #1	27
5.2	Mutant #2	28
5.3	Mutant #3	28
5.4	Mutant #4	29
5.5	Mutation Score	29

Revision History

Name	Date	Reason for Changes	Version
SoftFlux	03/05/2023	Intro/Purpose, Black box testing, White box testing, Mutation testing	1.0

1 Introduction/Purpose

1.1 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 describe how each of the requirements in the SRS and SDS document will be tested and the recommended testing software tools to perform the automated tests.

1.2 Document Conventions

This document uses the following conventions:

- ESGP: Encost Smart Graph Project
- GSL: GraphStream Library
- SRS: Software Requirement Specification
- SDS: Software Design Specification

1.3 Intended Audience and Reading Suggestions

This document is intended for any developer, tester, and project manager. Here are the potential uses for each of the reader types:

- 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.
- Test Developer: Implement a testing suite for test-driven development.

1.4 Testing 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.

2 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.

3 Black-box Testing

3.1 Categorizing Users

- Level of test: Unit (Recommended to use JUnit to automate test)
- Test technique: Decision table

3.1.1 All Users Decision Table

This decision table is used to test the user categorization.

- Test Case 1: When user enters "1", the system verifies that a community user has logged onto the ESGP and it displays that the user is a Community User and the feature options which the community user has access to.
- Test Case 2: When user enters "2", the system displays the user is a Encost User and prompts the user to enter their username and password.
- Test Case 3: When user enters anything that is not 1 or 2, the system will display an error message which tells the user that the input is wrong(E.g "Please input 1 or 2") and will prompt the user again for the user categorization.

Table 3.1: Categorizing Users Decision Table

Conditions	Input	Output
Case 1	1	Community user output and ESGP feature options displayed
Case 2	2	Encost user output and ESGP account login displayed
Case 3	Anything that is not 1 or 2	Error message displayed for wrong input

3.2 ESGP Account Login

- Level of test: Unit (Recommended to use JUnit to automate test)
- Test technique: Decision table

3.2.1 Encost User Decision Table

This decision table is used to test the account login for the encost users.

- Test Case 1: When user enters the correct username and correct password, the system will display a welcome message to the user and the feature options which the encost user has access to.
- Test Case 2: When user enters the invalid credentials, the system will display an error message to tell the user that their username and password are incorrect(E.g "Username and Password are wrong!") and it will prompt them again for the username and password.
- Test Case 3: When user doesn't enter anything, the system will display an error message to tell the user that they have not inputted anything(E.g "Please enter username and password!") and it will prompt them again for the username and password.

Table 3.2: Account Login Decision Table

Conditions	Input	Output
Case 1	Valid credentials	Welcome message output and ESGP feature options displayed
Case 2	Invalid credentials	Error message displayed for incorrect username and password
Case 3	Blank/NULL	Error message displayed for no input

3.3 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

3.3.1 Community User Decision Table

This decision table is used to test feature options for the community users.

- Test Case 1: When user enters "1", the system opens a graph visualisation UI application.
- Test Case 2: When user enters "X", the system closes the application.

- Test Case 3: When user enters anything that isn't "1" or "X", the system display an error message to tell the user that they inputted a wrong input(E.g "Please input either "1" or "X" only!") and it will prompt them again for what feature option they want.

Table 3.3: Feature Options Decision Table (Community)

Conditions	Input	Output
Case 1	1	Graph visualisation UI displayed
Case 2	X	Exit/Close the application
Case 3	Anything that is not 1 or X	Error message for wrong input

3.3.2 Encost User Decision Table

This decision table is used to test feature options for the encost users.

- Test Case 1: When user enters "1", the system prompts the user for the full file path of the custom dataset they want to use.
- Test Case 2: When user enters "2", the system opens a graph visualisation UI application.
- Test Case 3: When user enters "3", the system will display all the summary statistics in the form of console table outputs.
- Test Case 4: When user enters "X", the system closes the application.
- Test Case 5: When user enters anything that isn't "1" or "2" or "3" or "X", the system display an error message to tell the user that they inputted a wrong input(E.g "Please input either "1" or "2" or "3" or "X" only!") and it will prompt them again for what feature option they want.

Table 3.4: Feature Options Decision Table (Encost)

Conditions	Input	Output
Case 1	1	Custom dataset full file path prompt displayed
Case 2	2	Graph visualisation UI displayed
Case 3	3	Summary statistics displayed
Case 4	X	Exit/Close the application
Case 5	Anything that is not 1,2,3 or X	Error message for wrong input

3.4 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

3.4.1 All Users Decision Table

This decision table is used to test loading the encost smart homes dataset. The testing is based on Figure 3.1.

- Test Case 1: If the CSV format correct, device type is a router and device have the router, The device will be added into the graph.
- Test Case 2: If the CSV format correct, device type is not a router and device's router in the graph, the device will be added into the graph.
- Test Case 3: If the CSV format correct, device type is not a router and device's router not in the graph, the device will be added to the unlinked devices list.
- Test Case 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: If there are incorrect or missing data, the system will display an error message telling the user that there is wrong input in the encost smart homes dataset(E.g "Incorrect or missing input in the dataset").
- Test Case 17: If there are blank or no data, the system will display an error message telling the user that there is no input in the encost smart homes dataset(E.g "No data in the dataset").
- The format for CSV: device ID, date connected, device name, device type, household ID, router ID, can send, can receive
- Correct example 1: EWR-1234, 01/04/22, Encost Router 360,Router, WKO-1234, -, Yes, Yes
- Correct example 2: ELB-4567, 01/04/22, Encost Smart Bulb B22 (multi colour), Light bulb, WKO1234, EWR-1234, No, Yes

Table 3.5: Loading Encost Dataset Decision Table Part 1

Conditions	Case 1	Case 2	Case 3
CSV Format	Correct	Correct	Correct
Device type is a router?	Yes	No	No
Is the device's router in the graph?	NULL	Yes	No
Does the device have the router?	Yes	NULL	NULL
Output	Add device to graph	Add device to graph	Add device to unlinked devices list

Table 3.6: Loading Encost Dataset Decision Table Part 2

Conditions	Case 4	Case 5	Case 6
CSV Format	Correct	Correct	Correct
Device type is a router?	Yes	NULL	NULL
Is the device's router in the graph?	NULL	Yes	No
Does the device have the router?	No	NULL	NULL
Output	Error message for wrong input	Error message for wrong input	Error message for wrong input

Table 3.7: Loading Encost Dataset Decision Table Part 3

Conditions	Case 7	Case 8	Case 9
CSV Format	Correct	Correct	Incorrect
Device type is a router?	NULL	NULL	Yes
Is the device's router in the graph?	NULL	NULL	NULL
Does the device have the router?	Yes	No	Yes
Output	Error message for wrong input	Error message for wrong input	Error message for CSV format

Table 3.8: Loading Encost Dataset Decision Table Part 4

Conditions	Case 10	Case 11	Case 12
CSV Format	Incorrect	Incorrect	Incorrect
Device type is a router?	No	No	NULL
Is the device's router in the graph?	Yes	No	Yes
Does the device have the router?	NULL	NULL	NULL
Output	Error message for CSV format	Error message for CSV format	Error message for CSV format

Table 3.9: Loading Encost Dataset Decision Table Part 5

Conditions	Case 13	Case 14	Case 15
CSV Format	Incorrect	Incorrect	Incorrect
Device type is a router?	NULL	NULL	NULL
Is the device's router in the graph?	No	NULL	NULL
Does the device have the router?	NULL	Yes	No
Output	Error message for CSV format	Error message for CSV format	Error message for CSV format

Table 3.10: Loading Encost Dataset Decision Table Part 6

Conditions	Case 16	Case 17
CSV Format	Incorrect	Blank/NULL
Device type is a router?	NULL	NULL
Is the device's router in the graph?	NULL	NULL
Does the device have the router?	NULL	NULL
Output	Error Message for CSV format	Error Message for no input

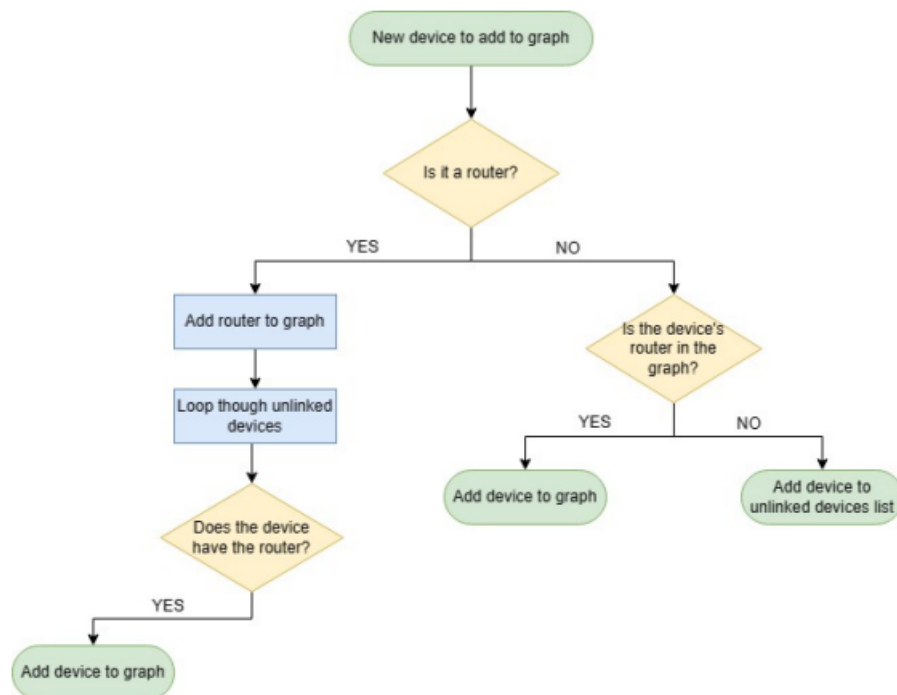


Figure 3.1: Flow diagram showing how devices should be added to the graph

3.5 Categorizing Smart Home Devices

- Level of test: Unit (Recommended to use JUnit to automate test)
- Test technique: Decision table

3.5.1 All Users Decision Table

This decision table is used to test categorization of smart home devices.

- Test Case 1: If the device type is router, the productCategory attribute will have encost wifi routers category.
- Test Case 2: If the device type is extender, the productCategory attribute will have encost wifi routers category.
- Test Case 3: If the device type is hub/controller, the productCategory attribute will have encost hubs/controllers category.

- Test Case 4: If the device type is light bulb, the productCategory attribute will have encost smart lighting category.
- Test Case 5: If the device type is strip lighting, the productCategory attribute will have encost smart lighting category.
- Test Case 6: If the device type is other lighting, the productCategory attribute will have encost smart lighting category.
- Test Case 7: If the device type is kettle, the productCategory attribute will have encost smart appliances category.
- Test Case 8: If the device type is toaster, the productCategory attribute will have encost smart appliances category.
- Test Case 9: If the device type is coffee maker, the productCategory attribute will have encost smart appliances category.
- Test Case 10: If the device type is washing machine/dryer, the productCategory attribute will have encost smart whiteware category.
- Test Case 11: If the device type is refrigerator/freezer, the productCategory attribute will have encost smart whiteware category.
- Test Case 12: If the device type is dishwasher, the productCategory attribute will have encost smart whiteware category.
- Test Case 13: If the device type is incorrect(Wrong Spelling, does not exist), the system will display an error message telling the user that the device type is incorrect(E.g "The device type is incorrect!").
- Test Case 14: If the device type is blank or null, the system will display an error message telling the user that there is not device type input(E.g "There is no device type input!").

Table 3.11: Categorizing Smart Home Devices Decision Table

Conditions	Device Type	Output
Case 1	Router	The productCategory attribute will be Encost Wifi Routers
Case 2	Extender	The productCategory attribute will be Encost Wifi Routers
Case 3	Hub/Controller	The productCategory attribute will be Encost Hubs/Controllers
Case 4	Light Bulb	The productCategory attribute will be Encost Smart Lighting
Case 5	Strip Lighting	The productCategory attribute will be Encost Smart Lighting
Case 6	Other Lighting	The productCategory attribute will be Encost Smart Lighting
Case 7	Kettle	The productCategory attribute will be Encost Smart Appliances
Case 8	Toaster	The productCategory attribute will be Encost Smart Appliances
Case 9	Coffee Maker	The productCategory attribute will be Encost Smart Appliances
Case 10	Washing Machine/Dryer	The productCategory attribute will be Encost Smart Whiteware
Case 11	Refrigerator/Freezer	The productCategory attribute will be Encost Smart Whiteware
Case 12	Dishwasher	The productCategory attribute will be Encost Smart Whiteware
Case 13	Incorrect	Error message for wrong data type
Case 14	Blank/NULL	Error message for no input

Device Category	Device types
Encost Wifi Routers	Router, Extender
Encost Hubs/- Controllers	Hub/Controller
Encost Smart Lighting	Light Bulb, Strip Lighting, Other Lighting
Encost Smart Appliances	Kettle, Toaster, Coffee Maker
Encost Smart Whiteware	Washing Machine/Dryer, Refrigerator/Freezer, Dishwasher

Figure 3.2: Device category and type table

3.6 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

3.6.1 All Users Decision Table

This decision table is used to test building a graph data type and graph visualisation.

- Test Case 1: If the device object router, no router in graph, can send, can receive, the system will add the white colour circle with no connections into the graph.
- Test Case 2: If the device object hubs/controller, no router in graph, can send, can receive, the system will add the orange colour circle with no connections into the graph.
- Test Case 3: If the device object smart lighting, no router in graph, can receive, the system will add the blue colour rectangle with no connections into the graph.

- Test Case 4: If the device object smart appliance, no router in graph, can receive, the system will add the pink colour rectangle with no connections into the graph.
- Test Case 5: If the device object smart whiteware, no router in graph, can Receive, the system will add the green colour rectangle with no connections into the graph.
- Test Case 6: If the device object smart appliance, no router in graph, can send, the system will add the pink colour “X” shape with no connections into the graph.
- Test Case 7: If the device object hubs/controller, router in graph, can send, can receive, the system will add the orange colour circle with a black line connection to white colour circle (Router) into the graph.
- Test Case 8: If the device object smart lighting, router in graph, can receive, the system will add the blue colour rectangle with a black line connection to white colour circle (Router) into the graph.
- Test Case 9: If the device object smart appliance, router in graph, can receive, the system will add the pink colour rectangle with a black line connection to white colour circle (Router) into the graph.
- Test Case 10: If the device object smart whiteware, router in graph, can receive, the system will add the green colour rectangle with a black line connection to white colour circle (Router) into the graph.
- Test Case 11: If the device object smart appliance, router in graph, can send, the system will add the pink colour “X” shape with a black line connection to white colour circle (Router) into the graph.
- Test Case 12: If the device object is incorrect, the system will display an error message telling the user that the device object is incorrect(E.g ”The device object is incorrect!”).
- Test Case 13: If the device object is null or blank, the system will display an error message telling the user that there is not device object input(E.g ”There is no device object input!”).

Table 3.12: Building Graph Data Type Decision Table Part 1

Conditions	Device Object	Router in the graph	Can Send	Can Receive	Output
Case 1	Router	No	Yes	Yes	A circle with white colour will appear in the graph that isn't connected with anything
Case 2	Hubs/Controller	No	Yes	Yes	A circle with orange colour will appear in the graph that isn't connected with anything
Case 3	Smart Lighting	No	No	Yes	A rectangle with blue colour will appear in the graph that isn't connected with anything
Case 4	Smart Appliance	No	No	Yes	A rectangle with pink colour will appear in the graph that isn't connected with anything

Table 3.13: Building Graph Data Type Decision Table Part 2

Conditions	Device Object	Router in the graph	Can Send	Can Receive	Output
Case 5	Smart White-ware	No	No	Yes	A rectangle with green colour will appear in the graph that isn't connected with anything
Case 6	Smart Appliance	No	Yes	No	A "X" shape with pink colour will appear in the graph that isn't connected with anything
Case 7	Hubs/Controller	Yes	Yes	Yes	A circle with orange colour will appear in the graph that is connected using a black line to the white colour circle(SEE FIGURE 3.3)
Case 8	Smart Lighting	Yes	No	Yes	A rectangle with blue colour will appear in the graph that is connected using a black line to the white colour circle(SEE FIGURE 3.3)

Table 3.14: Building Graph Data Type Decision Table Part 3

Conditions	Device Object	Router in the graph	Can Send	Can Receive	Output
Case 9	Smart Appliance	Yes	No	Yes	A rectangle with pink colour will appear in the graph that is connected using a black line to the white colour circle(SEE FIGURE 3.3)
Case 10	Smart White-ware	Yes	No	Yes	A rectangle with green colour will appear in the graph that is connected using a black line to the white colour circle(SEE FIGURE 3.3)
Case 11	Smart Appliance	Yes	Yes	No	A “X” shape with pink colour will appear in the graph that is connected using a black line to the white colour circle(SEE FIGURE 3.3)
Case 12	Incorrect	NULL	NULL	NULL	Error message for wrong device object

Table 3.15: Building Graph Data Type Decision Table Part 4

Conditions	Device Object	Router in the graph	Can Send	Can Receive	Output
Case 13	NULL	NULL	NULL	NULL	Error message for no device object

Graph visualisation with all the different devices objects:

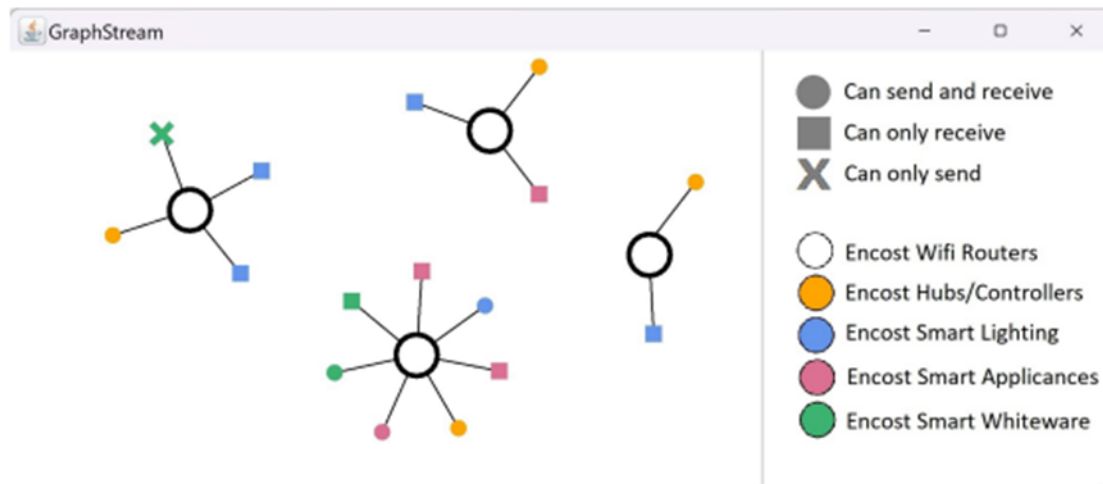


Figure 3.3: Graph visualisation

3.7 Calculating Device Distribution

- Level of test: Integration (Recommended to use JUnit to automate test)
- Test technique: Decision table

3.7.1 Encost User Decision Table

This decision table is used to test calculating device distribution.

getDeviceByCategory Method Testing:

- Incorrect String are strings that are not in the Figure 3.4 or blank/null strings.

- Test Case 1: If the `getDeviceByCategory()` string is correct, the method will return a list of devices where all devices returned are based on what was inputted as the string.
- Test Case 2: If the `getDeviceByCategory()` string is incorrect, the method will return an error message which informs the user that the string inputted is incorrect(E.g "The string inputted is incorrect!").

Table 3.16: `getDeviceByCategory` Method Decision Table

Conditions	Case 1	Case 2
<code>getDeviceByCategory(String)</code>	Correct String(Look at Figure 3.4)	Incorrect String
Output	A list of devices where all devices are returned depending on what string was given.	Error Message for wrong <code>getDeviceByCategory()</code>

`getDeviceByType` Method Testing:

- Incorrect String are strings that are not in the Figure 3.4 or blank/null strings.
- Test Case 1: If the `getDeviceByType()` string is correct, the method will return a list of devices where all devices returned are based on what was inputted as the string.
- Test Case 2: If the `getDeviceByType()` string is incorrect, the method will return an error message which informs the user that the string inputted is incorrect(E.g "The string inputted is incorrect!").

Table 3.17: `getDeviceByType` Method Decision Table

Conditions	Case 1	Case 2
<code>getDeviceByType(String)</code>	Correct String(Look at Figure 3.4)	Incorrect String
Output	A list of devices where all devices are returned depending on what string was given.	Error Message for wrong <code>getDeviceByType()</code>

`getDeviceByProduct` Method Testing:

- Incorrect String are strings that are not in the Figure 3.4 or blank/null strings.

- Test Case 1: If the `getDeviceByProduct()` string is correct, the method will return a list of devices where all devices returned are based on what was inputted as the string.
- Test Case 2: If the `getDeviceByProduct()` string is incorrect, the method will return an error message which informs the user that the string inputted is incorrect(E.g "The string inputted is incorrect!").

Table 3.18: `getDeviceByProduct` Method Decision Table

Conditions	Case 1	Case 2
<code>getDeviceByProduct(String)</code>	Correct String(Look at Figure 3.4)	Incorrect String
Output	A list of devices where all devices are returned depending on what string was given.	Error Message for wrong <code>getDeviceByProduct()</code>

`calculateDeviceDistribution` Method Testing:

- Incorrect String are strings that are not in the Figure 3.4 or blank/null strings.
- Test Case 1: If the string for `getDeviceByCategory` is Encost Wifi Routers, string for `getDeviceByType` is Router and the string for `getDeviceByProduct` is Encost Router 360, the system will output a long table string which has the number of Encost Wifi Routers, Router, Encost Router 360 displayed.
- Test Case 2: If the string for `getDeviceByCategory` is incorrect, string for `getDeviceByType` is incorrect and the string for `getDeviceByProduct` is incorrect, the system will output an error message for wrong `getDeviceByCategory()`, `getDeviceByType()` and `getDeviceByProduct()`.(E.g "The string inputted for all `getDevice` methods are incorrect!")
- Test Case 3: If the string for `getDeviceByCategory` is incorrect, string for `getDeviceByType` is correct and the string for `getDeviceByProduct` is correct, the system will output an error message for wrong `getDeviceByCategory()`.(E.g "The string inputted for `getDeviceByCategory` method is incorrect!")
- Test Case 4: If the string for `getDeviceByCategory` is correct, string for `getDeviceByType` is incorrect and the string for `getDeviceByProduct` is correct, the system will output an error message for wrong `getDeviceByType()`.(E.g "The string inputted for `getDeviceByType` method is incorrect!")

- Test Case 5: If the string for `getDeviceByCategory` is correct, string for `getDeviceByType` is correct and the string for `getDeviceByProduct` is incorrect, the system will output an error message for wrong `getDeviceByProduct()`. (E.g "The string inputted for `getDeviceByProduct` method is incorrect!")
- Test Case 6: If the string for `getDeviceByCategory` is incorrect, string for `getDeviceByType` is incorrect and the string for `getDeviceByProduct` is correct, the system will output an error message for wrong `getDeviceByCategory()` and `getDeviceByType()`. (E.g "The string inputted for `getDeviceByCategory` and `getDeviceByType` methods are incorrect!")
- Test Case 7: If the string for `getDeviceByCategory` is incorrect, string for `getDeviceByType` is correct and the string for `getDeviceByProduct` is incorrect, the system will output an error message for wrong `getDeviceByCategory()` and `getDeviceByProduct()`. (E.g "The string inputted for `getDeviceByCategory` and `getDeviceByProduct` methods are incorrect!")
- Test Case 8: If the string for `getDeviceByCategory` is correct, string for `getDeviceByType` is incorrect and the string for `getDeviceByProduct` is incorrect, the system will output an error message for wrong `getDeviceByType()` and `getDeviceByProduct()`. (E.g "The string inputted for `getDeviceByType` and `getDeviceByProduct` methods are incorrect!")

Table 3.19: calculateDeviceDistribution Method Decision Table Part 1

Conditions	Case 1	Case 2
<code>getDeviceByCategory(String)</code>	Correct String(Encost Wifi Routers)	Incorrect String
<code>getDeviceByType(String)</code>	Correct String(Router)	Incorrect String
<code>getDeviceByProduct(String)</code>	Correct String(Encost Router 360)	Incorrect String
Output	The number of Encost Wifi Routers, Router, Encost Router 360 are displayed in the format in Figure 3.5	Error message for wrong <code>getDeviceByCategory()</code> , <code>getDeviceByType()</code> and <code>getDeviceByProduct()</code>

Table 3.20: calculateDeviceDistribution Method Decision Table Part 2

Conditions	Case 3	Case 4
getDeviceByCategory(String)	Incorrect String	Correct String
getDeviceByType(String)	Correct String	Incorrect String
getDeviceByProduct(String)	Correct String	Correct String
Output	Error message for wrong getDeviceByCategory()	Error message for wrong getDeviceByType()

Table 3.21: calculateDeviceDistribution Method Decision Table Part 3

Conditions	Case 5	Case 6
getDeviceByCategory(String)	Correct String	Incorrect String
getDeviceByType(String)	Correct String	Incorrect String
getDeviceByProduct(String)	Incorrect String	Correct String
Output	Error message for wrong getDeviceByProduct()	Error message for wrong getDeviceByCategory() and getDeviceByType()

Table 3.22: calculateDeviceDistribution Method Decision Table Part 4

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

Device Category	Device types	Device Names
Encost Wifi Routers	Router, Extender	Encost Router 360, Encost Router Plus, Encost Wifi Range Extender 1.0, Encost Wifi Range Extender 2.0
Encost Hubs/-Controllers	Hub/Controller	Encost Smart Hub, Encost Smart Hub 2.0, Encost Smart Hub Mini
Encost Smart Lighting	Light Bulb, Strip Lighting, Other Lighting	Encost Smart Bulb B22 (white), Encost Smart Bulb B22 (multi colour), Encost Smart Bulb E26 (white), Encost Smart Bulb E26 (multi colour), Encost Strip Lighting (white), Encost Strip Lighting (multi colour), Encost Novelty Light (giraffe), Encost Novelty Light (lion), Encost Novelty Light (bear)
Encost Smart Appliances	Kettle, Toaster, Coffee Maker	Encost Smart Jug, Encost Smart Whistling Kettle, Encost Smart Toaster (2 slice), Encost Smart Toaster (4 slice), Encost Smart Coffee Maker, Encost Smart Coffee Maker Mini, Encost Smart Coffee Maker Pro
Encost Smart Whiteware	Washing Machine/Dryer, Refrigerator/Freezer, Dishwasher	Encost Smart Washer, Encost Smart Washer Pro, Encost Smart Dryer, Encost Smart Dryer Pro, Encost Smart Refrigerator, Encost Smart Freezer, Encost Smart Refrigerator/Freezer Combo, Encost Dishwasher, Encost Dishwasher Pro

Figure 3.4: Correct string for category, type and product table

Device Distribution	
=====	
Encost Wifi Routers	20
Router	15
Encost Router 360	2

Figure 3.5: Device Distribution for Test Case 1

Device Distribution	
=====	
Encost Wifi Routers	20
Router	15
Encost Router 360	2
Encost Router Plus	13
Extender	5
Encost Wifi Range Extender 1.0	0
Encost Wifi Range Extender 2.0	5
Encost Hubs/Controllers	6
Hub/Controller	6
Encost Smart Hub	1
Encost Smart Hub 2.0	3
Encost Smart Hub Mini	2
Encost Smart Lighting	34
Light Bulb	12
Encost Smart Bulb B22 (white)	4
Encost Smart Bulb B22 (multi colour)	3
Encost Smart Bulb E26 (white)	2
Encost Smart Bulb E26 (multi colour)	3
Strip Lighting	4
Encost Strip Lighting (white)	1
Encost Strip Lighting (multi colour)	3
Other Lighting	18
Encost Novelty Light (giraffe)	10
Encost Novelty Light (lion)	6
Encost Novelty Light (bear)	2

Figure 3.6: Full Device Distribution Output

4 White-box testing

4.1 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
```

Figure 4.1: getDevicesByCategory() Pseudocode

4.2 Branch Coverage Testing

- Level of test: Unit (Recommended to use JUnit to automate test)
- Test technique: Decision table

Test Case 1: inputtedString is Encost Smart Lighting

- The output will be the list of devices called categoryDevicesList will be returned.

Test Case 2: inputtedString is Blank/NULL

- The output will be an error message displayed telling the user that the inputted string is empty(E.g "Please add a inputted string!").

Test Case 3: inputtedString is 12356

- The output will be the an error message telling the user that the inputted string is wrong(E.g "Inputted string is wrong!").

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

- The output will be the an error message telling the user that the inputted string is wrong(E.g "Inputted string is wrong!").

Test Case 5: inputtedString is Encost Smart Light

- The output will be the an error message telling the user that the inputted string is wrong(E.g "Inputted string is wrong!").

The first test case achieves 100% branch coverage as it is able to go through all the branches of code without missing a single one or having any errors. The other test cases are used for error checking.

5 Mutation Testing

5.1 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
```

Figure 5.1: Mutation one

The mutation is devices size - 2. The original is devices size - 1.

5.2 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
```

Figure 5.2: Mutation two

The mutation is $i = 1$ and $j = 1$. The original is $i = 0$ and $j = 0$.

5.3 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
```

Figure 5.3: Mutation three

The mutation is FOR $i = 0$ to less than or equal to devices size - 1 and FOR $j = 0$ to less than or equal to unLinkedDevices size - 1. The original is FOR $i = 0$ to devices size - 1 and FOR $j = 0$ to unLinkedDevices size - 1.

5.4 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
```

Figure 5.4: Mutation four

The mutation is FOR i = 0 to more than devices size - 1 and FOR j = 0 to more than unLinkedDevices size - 1. The original is FOR i = 0 to devices size - 1 and FOR j = 0 to unLinkedDevices size - 1.

5.5 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.

Table 5.1: getDeviceCategory Testing Table

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

- Mutant 1 will have 7 items for the Encost Smart Lighting input and 5 items for the Encost Smart Whiteware input.
- Mutant 2 will have 7 items for the Encost Smart Lighting input and 5 items for the Encost Smart Whiteware input.

- Mutant 3 will have 8 items for the Encost Smart Lighting input and 6 items for the Encost Smart Whiteware input.
- Mutant 4 will have 9 items for the Encost Smart Lighting input and 7 items for the Encost Smart Whiteware input.

Mutation Score: $3/4$ (75%). Mutation #3 was not caught. The mutations are good as the mutation score is almost 100%.