**SOFTWARE**

**MAINTENANCE**

**DOCUMENT**

**for**

**Encost Smart Graph Project**

**Version 1.0**

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

**Prepared by:**

**SoftFlux Engineer**

**SoftFlux**

**May 18, 2023**

Contents

[1 Introduction/Purpose 3](#_Toc6590)

[1.1 Purpose 3](#_Toc6591)

[1.2 Document Conventions 3](#_Toc6592)

[1.3 Intended Audience and Reading Suggestions 3](#_Toc6593)

[1.4 Project Scope 3](#_Toc6594)

[2 Specialized Requirements Specification 3](#_Toc6595)

[3 Maintenance 4](#_Toc6596)

[3.1 Maintenance Task #1 4](#_Toc6597)

[3.1.1 User Story 4](#_Toc6598)

[3.1.2 Problem/modification request 4](#_Toc6599)

[3.1.3 Problem/modification analysis 4](#_Toc6600)

[3.1.4 Acceptance/rejection 5](#_Toc6601)

[3.1.5 Modification implementation 5](#_Toc6602)

[3.1.6 Maintenance review/acceptance 5](#_Toc6603)

[3.2 Maintenance Task #2 5](#_Toc6604)

[3.2.1 User Story 5](#_Toc6605)

[3.2.2 Problem/modification request 5](#_Toc6606)

[3.2.3 Problem/modification analysis 5](#_Toc6607)

[3.2.4 Acceptance/rejection 5](#_Toc6608)

[3.2.5 Modification implementation 6](#_Toc6609)

[3.2.6 Maintenance review/acceptance 6](#_Toc6610)

[3.3 Maintenance Task #3 6](#_Toc6611)

[3.3.1 User Story 6](#_Toc6612)

[3.3.2 Problem/modification request 6](#_Toc6613)

[3.3.3 Problem/modification analysis 6](#_Toc6614)

[3.3.4 Acceptance/rejection 6](#_Toc6615)

[3.3.5 Modification implementation 7](#_Toc6616)

[3.3.6 Maintenance review/acceptance 7](#_Toc6617)

[4 CI/CD Pipeline 7](#_Toc6618)

[5 Conclusion 7](#_Toc6619)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason for Changes | Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction/Purpose

## Purpose

This document is Software Maintenance Document for the software Encost Smart Graph Project (ESGP) for the company Encost. The purpose of this document is to provide a way to show software maintenance of the ESGP, CI/CD pipeline and to track what maintenance have been done already.

## Document Conventions

This document uses the following conventions:

* ESGP: Encost Smart Graph Project
* SDPD: Software Development Planning Document
* FSTP: Functional Software Test Plan
* SDS: Software Design Specification

## Intended Audience and Reading Suggestions

* Software Maintainer and DevOps Engineer: Uses this document to track the progress and to see what has been newly implemented.

## Project Scope

The ESGP is a software system with aims to enable the Encost smart devices to be visualised in a graph data structure from the command line. With both accessibility to Encost verified users and community users but with restricted access to the software capability. There will be no hardware integration required for the ESGP nor for the testing purposed of the software. SoftFlux is responsible to the software development and maintenance of the ESGP and the insurance that the software meets the requirements listed in the SDPD 2, FSTP 4 and the SDS 3.

# Specialized Requirements Specification

# Maintenance

## Updating GraphStream to latest version

### User Story

* The client (Encost) has decided that the system should use the most up to-date version of GraphStream.

### Problem/modification request

* Update GraphStream to the latest version (V2.0).

### Problem/modification analysis

**Maintenance Category:** Perfective

**Impact Analysis**

* **Verify the problem:** The system is currently using GraphStream 1.3 which is not the latest version.
* **Implementation options:** Update the GraphStream to version 2.0 and change some of the methods used for GraphStream 1.3.
* **Effort:** 10-15 lines of code including testing code, need to learn some new methods created in the new version which throws away some of the old methods used.
* **Resources:** 2 hours which includes making changes and running all the tests it previously had.

### Acceptance/rejection

* Accepted

### Modification implementation

Before in GraphVisualiser java file:



After in GraphVisualiser java file:



* Changed addAttribute method to setAttribute method in the GraphVisualiser java file.

### Maintenance review/acceptance

Compiling all the java files including GraphVisualiser java file:



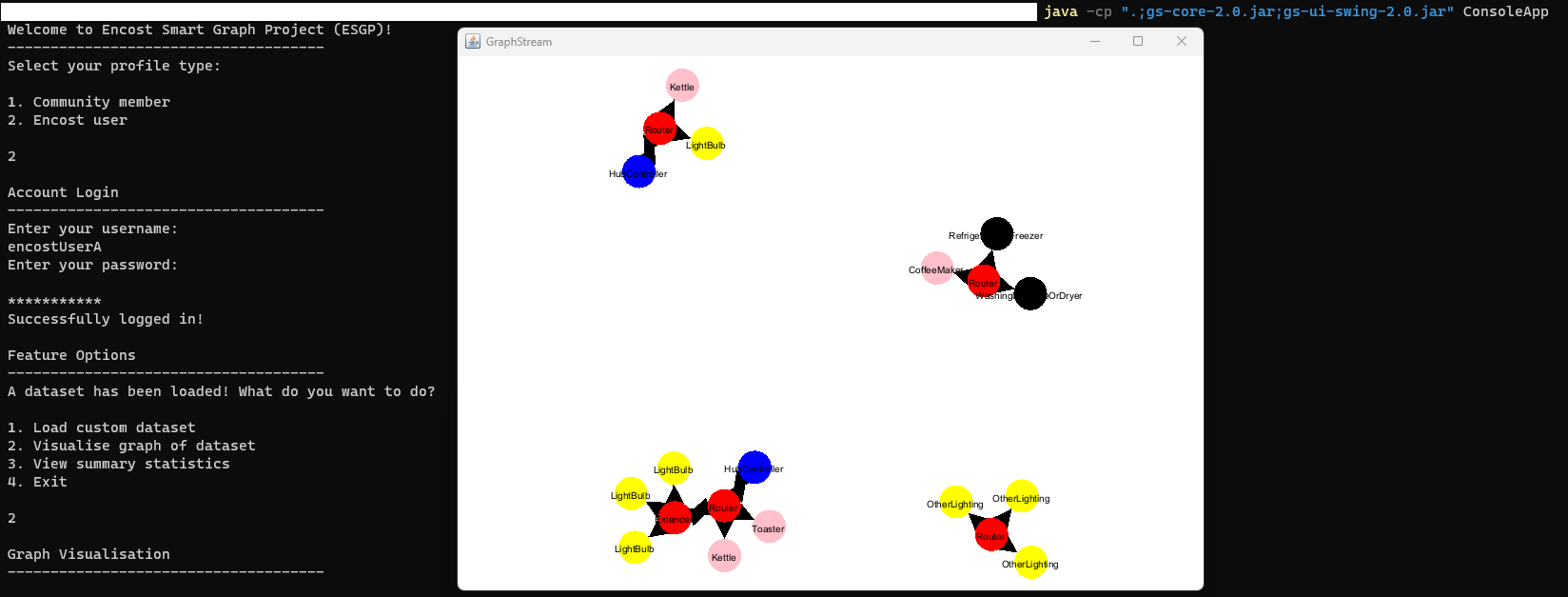
* It shows no errors so that means all the methods used for graph stream is compatible with the latest version.

Community user graph visualisation:

A screenshot of a computer

Description automatically generated

Encost user graph visualisation:



## Calculating and displaying device distribution in string table format

### User Story

* The Encost users have noticed that the summary statistics aren’t working. They would like to be able to see a textual summary of the device distribution (SRS 4.9 Calculating Device Distribution).

### Problem/modification request

* Investigate the summary statistics and add a textual summary of device distribution.

### Problem/modification analysis

**Maintenance Category:** Corrective

**Impact Analysis**

* **Verify the problem:** The summary statistics is not implemented, and we must also implement textual summary of device distribution.
* **Implementation options:** Make a DataDistributionStatistics class java file which have a method to calculate the device distribution and another method to display it in a textual summary format.
* **Effort:** 200-400 lines of code including testing code, need to learn how to display device distribution in a textual summary format and how to calculate the device distribution.
* **Resources:** 12 hours to 24 hours which includes making changes and making new tests.

### Acceptance/rejection

* Rejected, takes too much time and resources to do.

## One household graph visualisation

### User Story

* The Community users are having trouble telling which household is theirs in the graph visualisation. They would like the additional feature of being able to see a graph visualisation for their household only.

### Problem/modification request

* Add a new feature for community users so that they can see their own household.

### Problem/modification analysis

**Maintenance Category:** Perfective

**Impact Analysis**

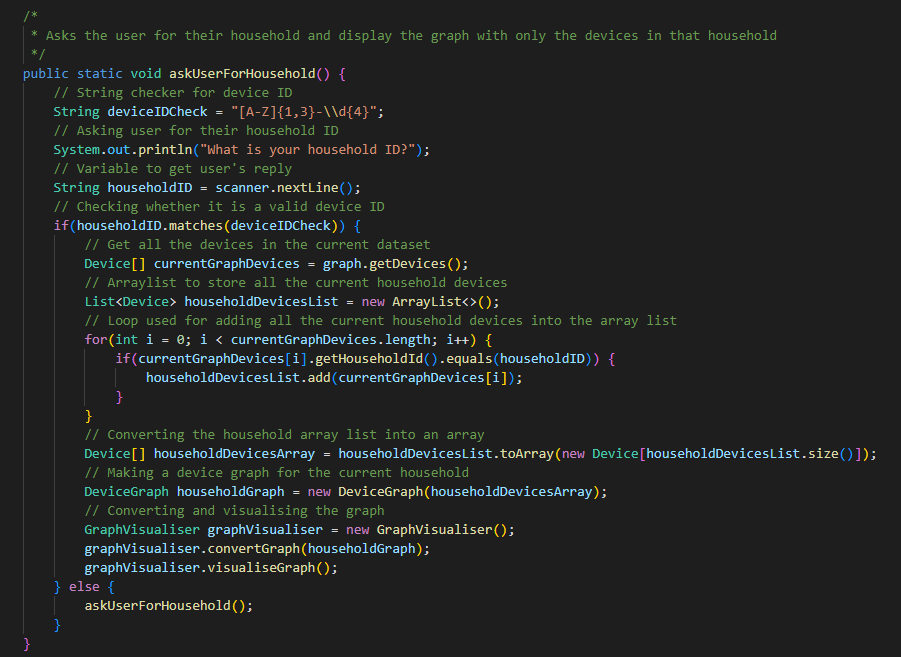
* **Verify the problem:** The system currently only visualises all graphs made using the dataset. The new feature will let the user tell the system their household and it will display a graph with only that household devices.
* **Implementation options:** Create a new method called askUserForHousehold that will ask the user for their household, and it will get the current household devices from the current dataset and display it all in the graph.
* **Effort:** 100-250 lines of code, no new knowledge needed.
* **Resources:** 4-8 hours which includes making changes, running all the tests it previously had and doing manual tests.

### Acceptance/rejection

* Accepted

### Modification implementation

askUserForHousehold method:



Before in ConsoleApp java file selectFeature method:

A screen shot of a computer

Description automatically generated with medium confidence

After in ConsoleApp java file selectFeature method:

A screen shot of a computer

Description automatically generated with low confidence

* Changed the input “2” to be used for the household graph and made input “3” the exit instead.

Before in ConsoleApp java file getOptions method:

A screen shot of a computer program

Description automatically generated with low confidence

After in ConsoleApp java file getOptions method:

A screen shot of a computer code

Description automatically generated with low confidence

* Added the option to visualise the current household and made the exit option number 3 instead of 2 for the optionsCommunity array.

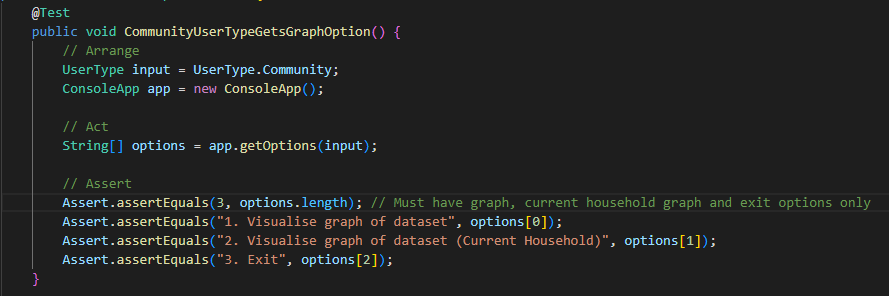
### Maintenance review/acceptance

Before in FeatureOptionsTest java file:

A screen shot of a computer program

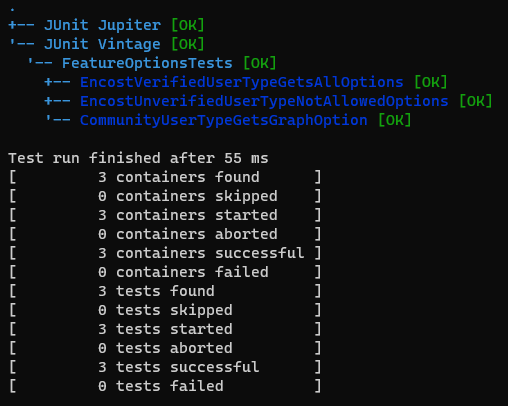
Description automatically generated with low confidence

After in FeatureOptionsTest java file:



* Changed the assertEquals for option 2 to visualise for current household and made the exit option number 3.

FeatureOptionsTest JUnit test:



Existing household graph visualisation:

A screenshot of a computer

Description automatically generated

Non-existing household graph visualisation:

A screenshot of a computer

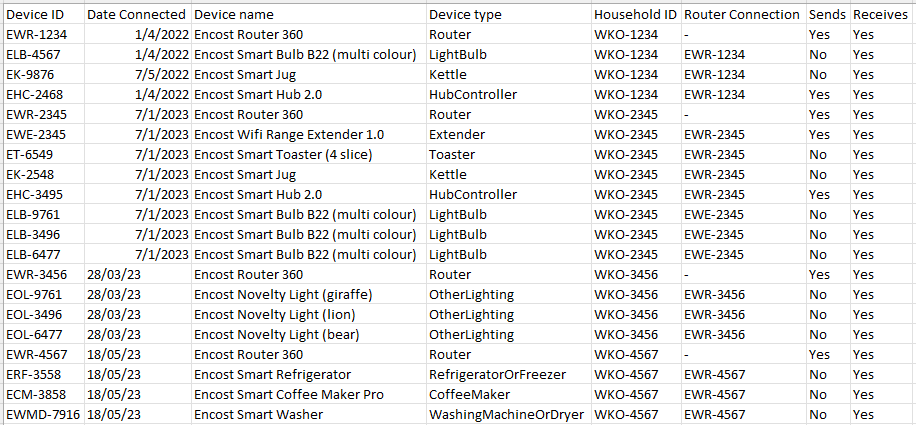
Description automatically generated

If the user inputs anything that isn’t a device ID, the system will keep asking for their household ID like in the figure below. Device ID must have 1-3 upper case letters first and a dash then followed by 4 digits.

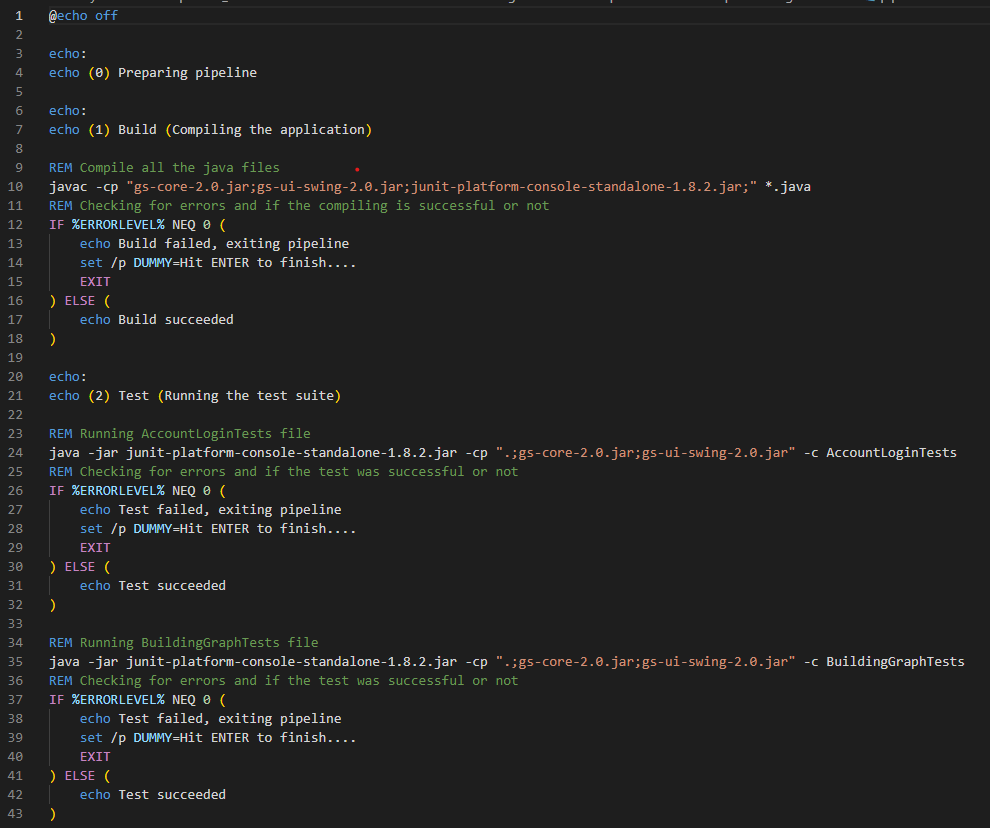
A screenshot of a computer screen

Description automatically generated with medium confidence

Dataset used for testing:



# CI/CD Pipeline



A screen shot of a computer program

Description automatically generated with low confidence

A screen shot of a computer program

Description automatically generated with low confidence

A screen shot of a computer program

Description automatically generated with medium confidence

My CI/CD pipeline has 5 portions. The preparing in the pipeline is just used to start the pipeline. The building in the pipeline is used to compile all the java files and check whether it succeeds or not. The test in the pipeline is used to run all tests and checks whether they all succeed or not. The release in the pipeline is used to commit to the repository. The pipeline first checks whether any files need to be added. If there is any files needed to be added, it will add it. After it adds it, it will ask the user for the commit message. Once the user enters the commit message, the pipeline will commit and push the files into the repository. The last portion is the deploying. The pipeline just runs the application and let the user use the application. It will show a message if it fails or succeed after the user exits the application.

# Conclusion

With the maintenance, the GraphStream for the application is now the latest version and there is a new feature which allows community user to give their household ID and it will display a graph visualisation with only the devices from the household ID stated. The CI/CD pipeline is now able to prepare, build, test, release and deploy.