DEVELOPMENT PLANNING DOCUMENT

for

Encost Smart Graph Project

Version 1.0

Prepared by: Ben Lee
SoftFlux Engineer

SoftFlux

June 11, 2023

Contents

Revision History

Name	Date	Reason for Changes	Version

1 Introduction/Purpose

1.1 Purpose

This development planning documents purpose is to act as an guide for developers to follow when they are implementing software as this document follows requirements and design philosophies stated in the SRS, SDS and functional software test plan documentation. This document will include Product Backlog and sprint details. This document is written in hopes there is enough information that is clear and concise to avoid any ambiguities, if followed correctly the development cycles will run effectively.

1.2 Document Conventions

- ESGP: Encost Smart Graph Project
- CLI: Command Line Interface
- SRS: Software Requirement Specifications
- SDS: Software Design Specifications

1.3 Intended Audience and Reading Suggestions

Intended Audience

- Product Owner: The product owner will use customer communication to update any requirements.
- Development Team: A team member will require this document to understand sprint details, such as tasks, testing and any backlog items.
- Scrum Master: The team leader tasked with guiding the development team with an understanding of the scrum values, principles and practices.

Reading Suggestions
All team members are expected to have read and studied the SRS, SDS and Test Plan documentation. All functional requirements are used in the software, all non-functional requirements must be considered and utilised when optimising the code. Any diagrams and design principles shown and used in the SDS will be used to design all classes within the software. The Test Plan will be used to design all tests and used to create check functions that validate data.

1.4 Project Scope

Encost uses the ESGP project to better understand how its devices are being used and linked in New Zealand households. The Encost Smart Homes Dataset, a dataset including details about 100 New Zealand homes and the Encost devices being used, will be able to be visualised and summarised through the ESGP project. The 100 New Zealand households that agreed to have their data collected will also receive the project so they may view a visualisation of the Encost Smart households Dataset.

2 Specialized Requirements Specification

2.1 Classes and Functions

Encost Class
The Encost class should run all UI prompts, the functions should be named accordingly, promptCategorisingUser, promptAccountLogin and promptFeatureOptions. There should also be check functions for all these prompts, names should match the test check names.

Device Class
The Device class needs a function that sets the device category, this uses the device type extracted from the dataset to categorise the device.

DeviceGraph Class
The DeviceGraph class is the Graph Data Type, it has just been renamed and slightly adapted for implementation. Functions need to be added to this class, checkDataset function will take the data from each line and validate it. loadDataset function should be added to make sure that any dataset is loaded correctly and has valid data. cleanData function needs to be added, this will remove spacing and punctuation from every piece of data from the dataset. printDeviceDistribution function should be added to accommodate the device distribution calculation, this will take device counters and print them formatted, the formatting must be easy readable. The calculateDeviceDistribution function will be overhauled, it will now use a linked hash map to track type and category counters.

2.2 Tests

Categorising Users Test
Updated CategorisingUsersTest to better illustrate the test name and display name.

Account Login Test
Updated AccountLoginTest to change naming conventions to better illustrate what the purpose of the test is and changing function names to accommodate changes in Encost function names.

Feature Options Test
Updated FeaturesOptionsTest to accommodate any name changes and change test names to better illustrate what the tests does and changes function names to accommodate changes in Encost function names.

Load Dataset Test
Overhauled LoadDatasetTest, this test will check a device is valid using the checkDataset function, this test will validate loading a dataset and building a device graph.

Load Dataset Test
Overhauled LoadDatasetTest, this test will check a device is valid using the checkDataset function, this test will validate loading a dataset and building a device graph.

Categorising Devices Test
Overhaul on CategorisingDevicesTest to use Device class and added setDeviceCategory function.

Device Distribution Test
Overhauled DeviceDistributionTest to match the new implementation of calculateDeviceDistribution function.

3 Product Backlog

3.1 Functional Must Haves:

3.1.1 Categorising Users

Description
Users should be allowed to specify whether they are Community or Encost users. Encost Users will need to login, then they will have access to more features than Community Users.

Use Cases
"Derek Brewer is a community member as he opted to have his information recorded for ESGP, Derek is only interested in viewing the Graph Visualisation of his and others devices he gave permission to record. He doesn't plan on using ESGP very often, so it will be important he can get through the application with ease even after a long period of not using it."
"Harriet Cooper is an Encost member, she is a very busy person and wants to be able to view the summary statistics, she is interested in how popular the devices she owns are. She hopes that Encost's promise to provide Encost member with extra features is upheld. Usernames and passwords will be given to Encost members to make sure only these members will get the extra options."

3.1.2 ESGP Account Login

Description
Employees of Encost should be able to log in to the system and access additional functionalities. For this purpose, a set of login credentials will be issued.

Use Cases
"Ned Rodriguez is an Encost member who knows all his usernames and passwords off by heart and assures us he will only need one attempt."

"Harriet Cooper has informed us she has a lot of usernames and passwords and she is worried she won't get her username and password correct on the first attempt. But as promised by Encost there will be 3 attempts at a login before being sent to the start of ESGP."

3.1.3 ESGP Feature Options

Description
Verified Users should be able to choose between "1" loading a custom dataset, "2" visualising a graph representation of the data, and "3" viewing summary statistics. Community Users should be able to choose between "1" loading a custom dataset.

Use Cases
"Ned Rodriguez works as a programmer and says that he has worked with many command line apps and will have no problem reading and understanding given instructions."
"Katie Ortiz is an encost member but she is an older lady who doesn't have much tech experience and is worried this will prevent her from using the application. To prevent any concern, Encost promises that the process will be intuitive and any incorrect input will result in a detailed response for correct inputs."

3.1.4 Loading the Encost Smart Homes Dataset

Description
The Encost Smart Homes Dataset should be read and processed by the system.

Use Cases
"Natalie Frye is an Encost member involved with the ESGP, she wants to see any information she can on the Encost devices, she is only concerned that the data came from Encost and that she can see the information quickly."
"Isabella Reed is an Encost member, she wants to be updated when the application's system is updating any information, such as if the dataset was successfully loaded or not. To make this happen Encost has promised that the system will update the user whenever the dataset is loaded."

3.1.5 Categorising Smart Home Devices

Description
Each Encost Smart Device should be able to be grouped into one of five categories by the system. These categories will be used to display graphs and summarise statistics.

Use Cases
"Encost would like all devices to be categorised according to the devices type. This is important for calculating the summary statistics for Encost members."

3.1.6 Building a Device Graph

Description
To store all of the Encost Smart Device Objects, the system should develop a graph data structure. This graph will be used to display Encost Smart Devices and provide summary data. A device class will need to be created when creating the graph data type class.

Use Cases
"Encost would like all devices to be stored in Device Graph set, this Graph set must store all the devices from a given dataset and will be used in both graph visualisation and within a set of summary statistics."

3.1.7 Graph Visualisation

Description
The system should provide the user with a visual representation of the graph data structure.

Use Cases
"Derek Brewer has told us that he is only interested in viewing the graph, but it's important to him that the information presented in the graph is easily understood and presentable. He hopes that the promised feature of being able to differentiate devices in the graph is upheld, and the differences are known to the user."

3.1.8 Calculating Device Distribution

Description
Based on the information stored in the graph data structure, Verified Encost Users should be able to visualise the distribution of devices across categories and types.

Use Cases
"Harriet Cooper is a statistician, so she claims any statistics presented in the summary will be easy to comprehend and she has no concern with how the statistics are formatted as long as they are readable."
"Katie Ortiz has brought up a similar worry, however, this time she is concerned with how easily the summary statistics will be to read and understand."

3.2 Functional Should Have:

3.2.1 Calculating Device Location

Description
Based on the information stored in the graph data structure, Verified Encost Users should be able to view the location of devices both geographically and between homes.

3.2.2 Calculating Device Connectivity

Description
Based on the information stored in the graph data structure, Verified Encost Users should be able to view the connection and communication between devices.

3.3 Functional Could Have:

3.3.1 Loading Custom Datasets

Description
Custom datasets should be able to be read and processed by the system as long as their format matches that of the Encost Smart Homes Dataset.

3.4 Non-functional Requirements

See SRS documentation for clarification on all non-functional requirements.
Non-functional requirements include: Response Times, User Authentication, Data anonymity, Reliability, Interoperability and Usability.

4 Sprint Details

4.1 Sprint #1 <10/05 - 10/05>

4.1.1 Product Backlog Items

- Categorising Users
- ESGP Account Login
- ESGP Feature Options

4.1.2 Sprint Tasks

Categorising Users

- Welcome User - UI Design
- Ask for user type - UI Design
- Check an input is valid - System Requirement
- Categorise user type by input as either “encost-unverified” or “community” - System Requirement
- If the user is a community member prompt the ESGP Feature Options - UI Design
- If the user is a encost member prompt ESGP Account Login - UI Design

ESGP Account Login

- Ask user for Username - UI Design
- Check Username is valid, if not then prompt Username again - System Requirement
- Ask user Password - UI Design
- Check Password is valid, if not then prompt Username again - System Requirement
- If 3 failed attempts, prompt Categorising Users - System Requirements/UI Design
- If successful, prompt ESGP Features and set user type to encost-unverified - UI Design

ESGP Feature Options

- Ask for user type - UI Design
- Check an input is valid - System Requirement
- Check input is either “1”, “2” or “3” for “encost-verified” users or Check input is “1” for “Community” users - System Requirement
- If the input is “1” prompt Graph Visualisation - UI Design
- If the input is “2” prompt Custom Dataset - UI Design
- If the input is “3” prompt Statistics Summary - UI Design
- If the input is invalid print to the user what the valid inputs are - UI Design

4.1.3 Software Design

The user experience for these backlog items should be intuitive as our users may have a wide range of competency when it comes to technology. To ensure this visual cues and clarity should be used to illustrate any expected inputs, as this will make progressing through the application simple. When an an incorrect input is made the user should be given feedback and educated to prevent any errors from occurring again. Progression through the should resemble the direction of the flow diagram from the SDS documentation, see Figure 4.1 for clarification.

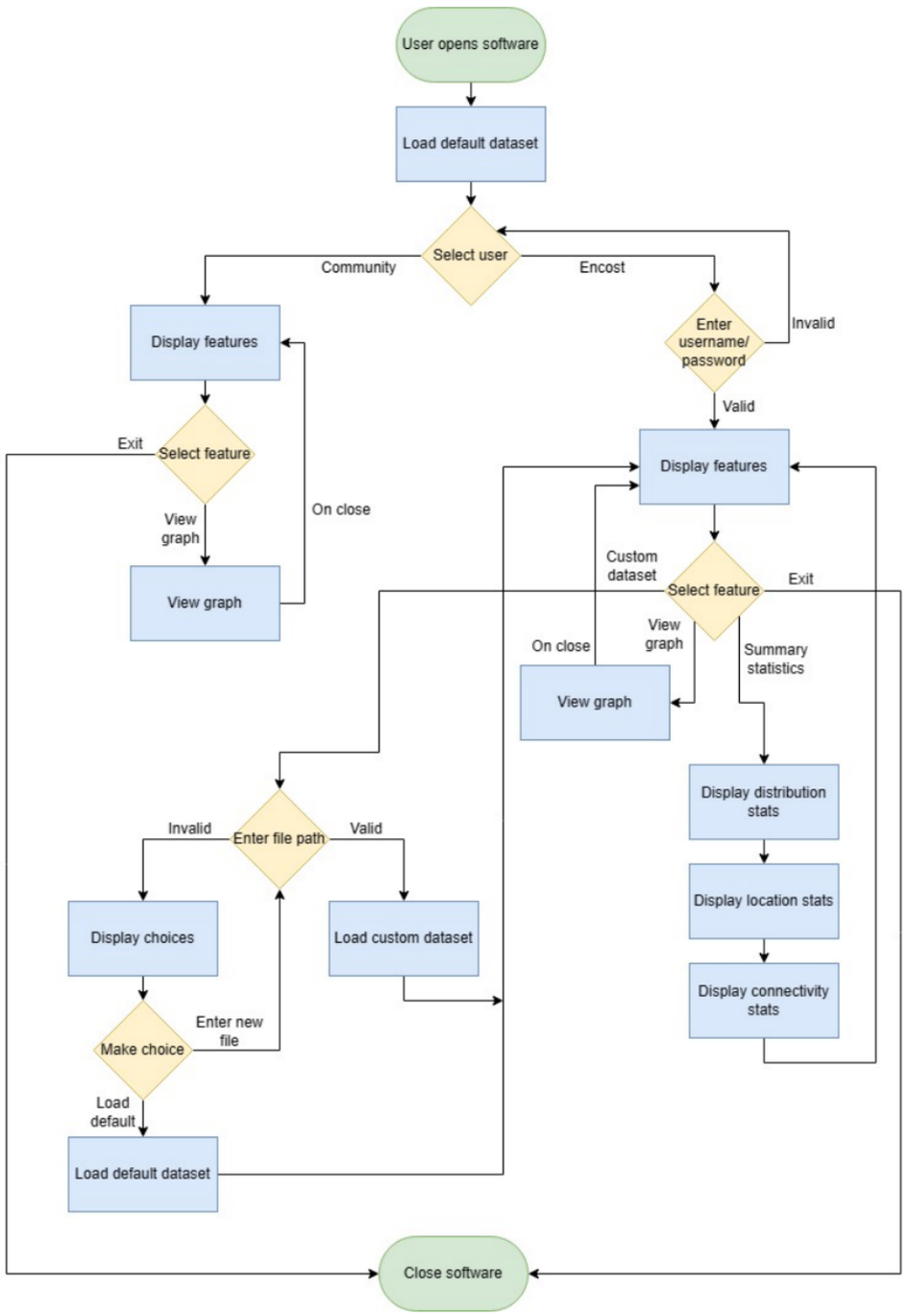

Figure 4.1: Flow Diagram

4.1.4 Software Testing

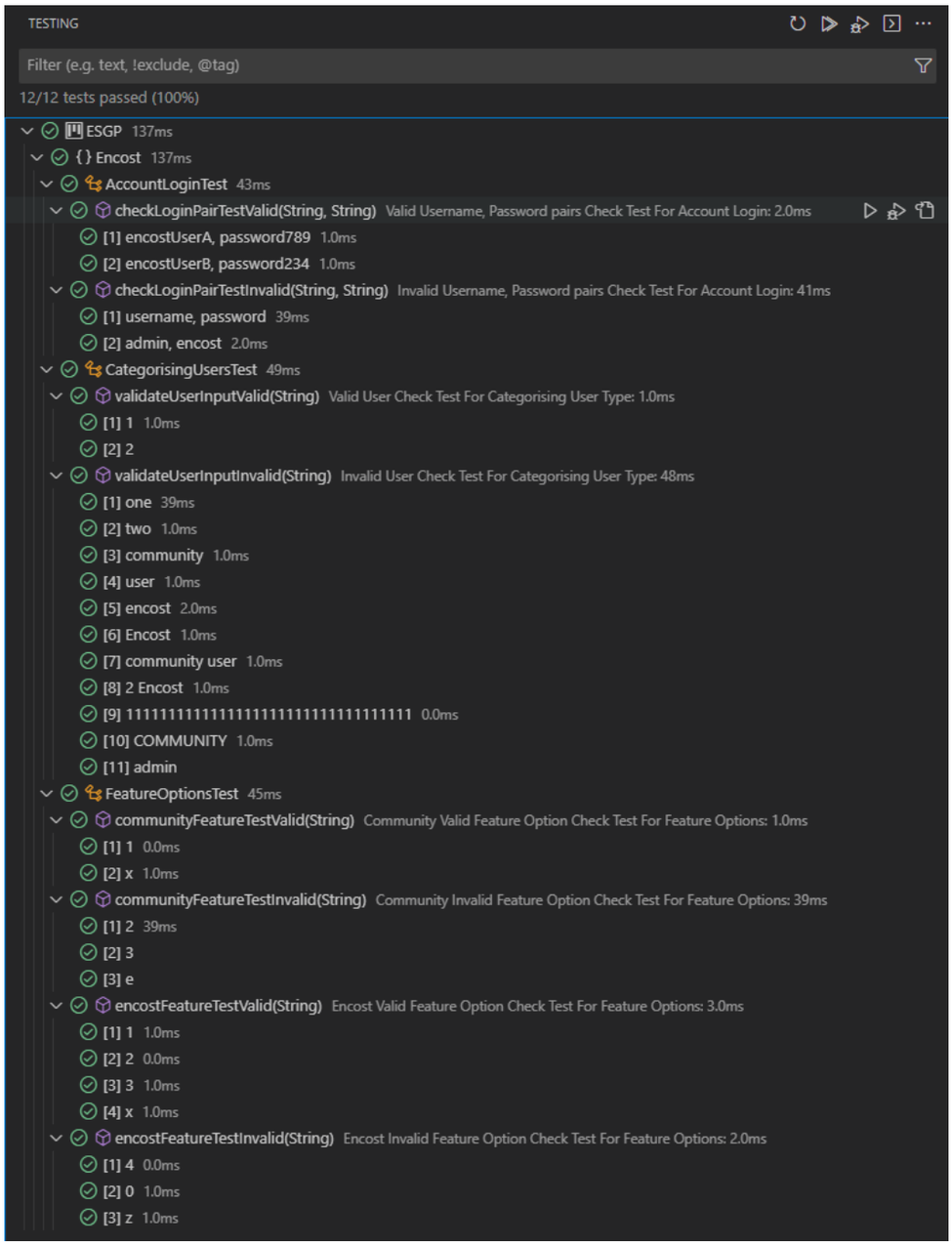

Figure 4.2: Sprint 1 Testing

4.1.5 Sprint Task Completion

Categorising users - Complete
ESGP Account Login - Complete
ESGP Feature Options - Complete

4.2 Sprint #2 <11/05 - 11/05>

4.2.1 Product Backlog Items

- Building a Device Graph

4.2.2 Sprint Tasks

Building a Device Graph

- Set up Device Graph according to the SDS documentation with some altercations - System Requirements
- A cleaning data function must be added to remove all spaces and capitalisation - System Requirements

Create Device class

- Set up Device according to the SDS documentation - System Requirements
- Add an extra function to set device category - System Requirements

4.2.3 Software Design

The Device Graph class will use the Singleton Design pattern. The reason for this is that the Device Graph stores the list of devices extracted from the dataset and we don't want this list to be reinitialised, using the singleton design pattern will prevent this from happening. The Device Graph should be built with Figure 4.3 in mind.

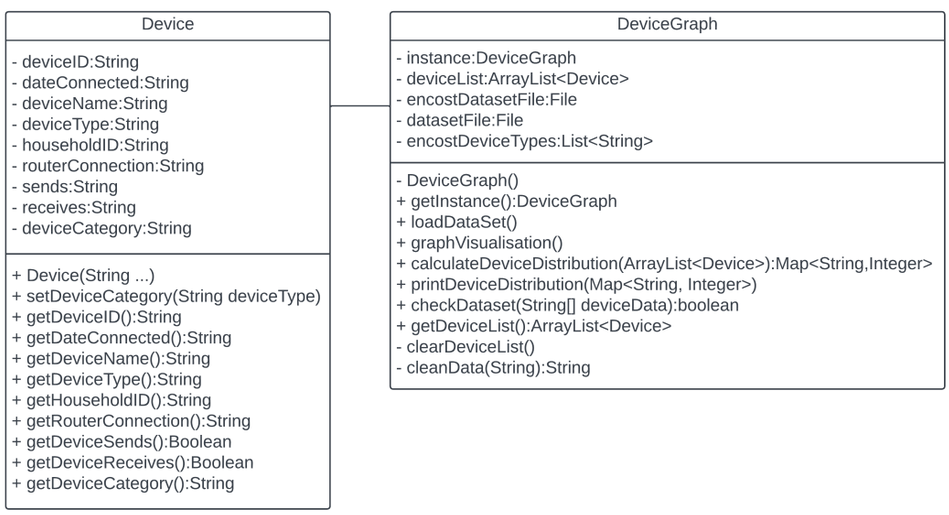

Figure 4.3: Class Diagram

4.2.4 Sprint Task Completion

Building a Device Graph - Next Sprint
Create Device Class - Complete

4.3 Sprint #3 <12/05 - 13/05>

4.3.1 Product Backlog Items

- Loading the Encost Smart Home Dataset
- Categorising Smart Home Devices
- Building a Device Graph

4.3.2 Sprint Tasks

Building a Device Graph

- Add features as needed for current tasks - System Requirements
- Load data set function - System Requirements

Loading the Encost Smart Home Dataset

- Load dataset during Welcome Prompt - System Requirements
- Check the file can be found and is correct file type - System Requirements
- Check line by line that there is enough data - System Requirements
- Check for valid data - System Requirements
- Clean data for entry, by removing spaces and punctuation - System Requirements
- If unsuccessful, prompt "Dataset unsuccessfully loaded" - UI Design
- If successful, prompt "Dataset successfully loaded" - UI Design

Categorising Smart Home Devices

- Add set device category function in device class - System Requirements
- Check the device is a valid device - System Requirements
- Check the device is valid - System Requirements
- Categorise device using Device class - System Requirements

4.3.3 Software Design

All these backlog items should be built with validation in mine, the dataset must be validated before being loaded into the system, this ensures all devices are as expected and means the devices can be categorised properly. All device data including categories should be clear, concise and adaptive, to achieve this all data will be 'cleaned', this mean removing spaces and punctuation as this ensures when data is being passed around the system no errors occur. Scalability should also be considered, to achieve this, encost devices should be stored and can be added to, this allows any new encost devices to be added to the system.

4.3.4 Software Testing

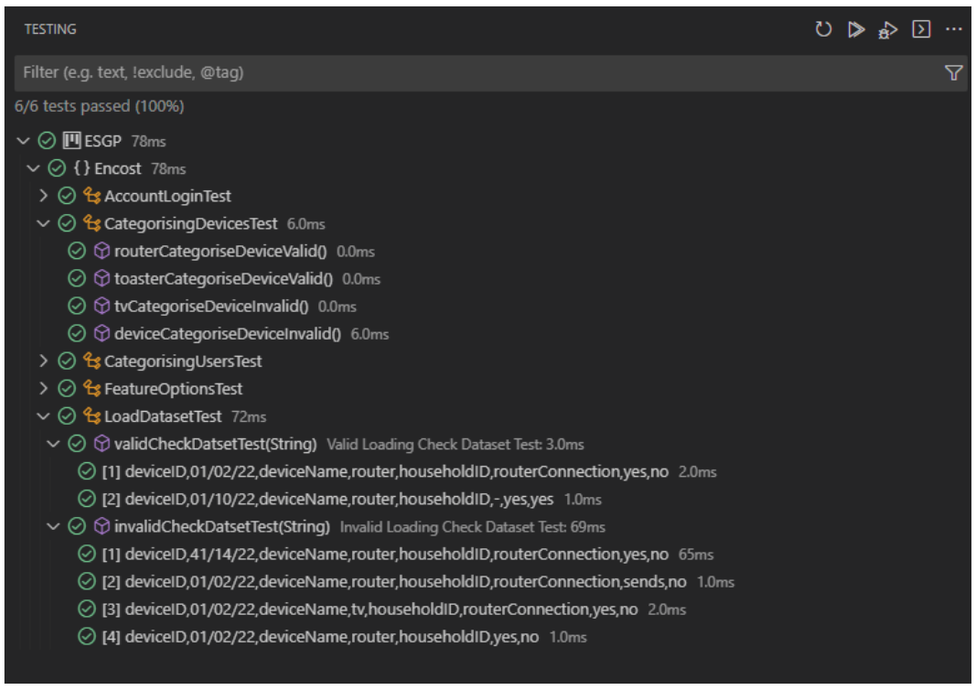

Figure 4.4: Sprint 3 Testing

4.3.5 Sprint Task Completion

Building a Device Graph - Next Sprint
Loading the Encost Smart Home Dataset - Complete
Categorising Smart Home Devices - Complete

4.4 Sprint #4 <14/05 - 14/05>

4.4.1 Product Backlog Items

- Calculating Device Distribution
- Building a Device Graph

4.4.2 Sprint Tasks

Building a Graph Data Type

- Add features as needed for current tasks - System Requirements

Calculating Device Distribution

- Add calculate device distribution function in device graph class - System Requirements
- Set up counters for each device category and type - System Requirements
- Iterate through device list and add to counters - System Requirements
- Add print device distribution function and pass the counters - System Requirements
- The print should be formatted according to SDS documentation - UI Design
- The formatting should be easy to read and understand - UI Design

4.4.3 Software Design

Calculating Device Distribution sends the statistics to a print function, the printed information should be clear and concise making the users experience intuitive, to achieve this formatting should be used. For all summary statistics it's important the calculations are efficient as performance is important for our users.

4.4.4 Software Testing

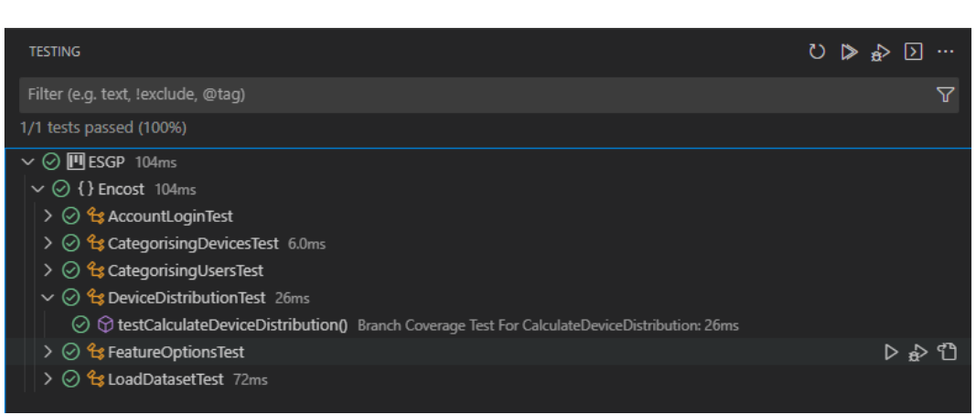

Figure 4.5: Sprint 4 Testing

4.4.5 Sprint Task Completion

Building a Device Graph - In Progress
Calculating Device Distribution - Complete

4.5 Sprint #5 <15/05 - 15/05>

4.5.1 Product Backlog Items

- Graph Visualisation
- Building a Device Graph

4.5.2 Sprint Tasks

Building a Graph Data Type

- Add features as needed for current tasks - System Requirements

Graph Visualisation

- Import GraphStream library - System Requirements
- Set up Devices as Nodes and Connections as Edges - System Requirements
- Display Graph - UI Design
- Graph needs to be easily understandable - UI Design

4.5.3 Software Design

The Graph should be designed with clarity in mind, users should be able to easily decipher the differences between device types, households and device connections. See Figure 4.6 for clarification.

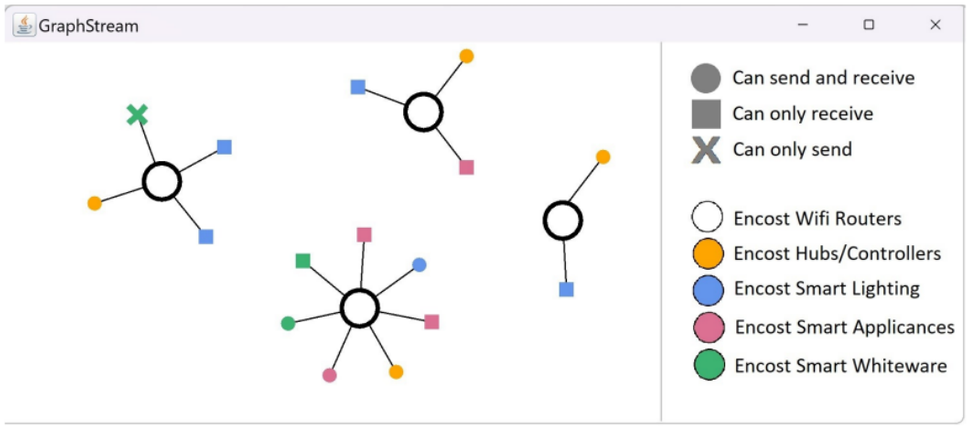

Figure 4.6: Graph Stream

4.5.4 Software Testing

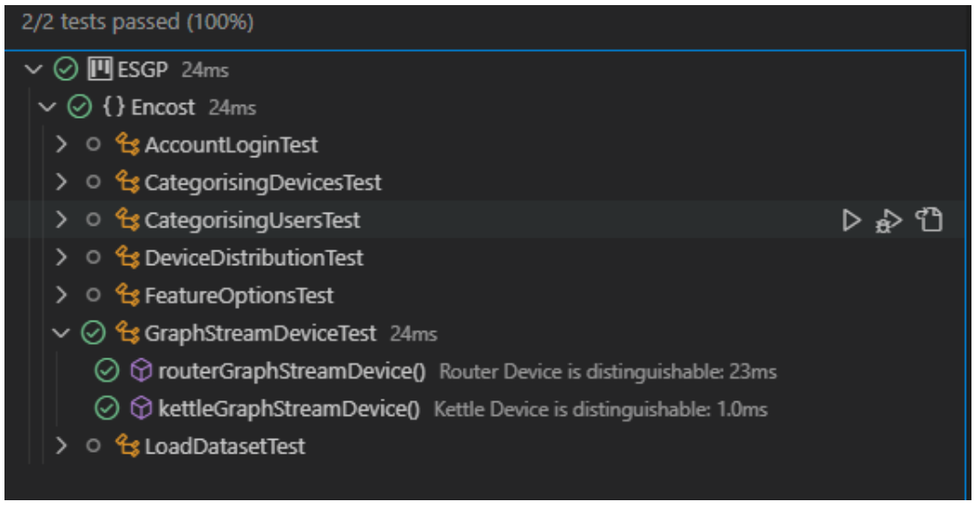

Figure 4.7: Sprint 5 Testing

4.5.5 Sprint Task Completion

Graph Visualisation - Complete
Building a Graph Data Type - Complete

5 Conclusion

Overall the software should have all must have backlog items, this includes all high priority functions stated in the SRS, as well as any design principles included in the SDS document. If the sprints are executed appropriately, all tests will pass and system will run smoothly.