SOFTWARE DESIGN SPECIFICATION

for

Encost Smart Graph Project

Version 1.0

Prepared by: Joey Han
SoftFlux Engineer

SoftFlux

April 4, 2023

Contents

Revision History

Name	Date	Reason for Changes	Version
Joey Han	04/04/2023	Initial Version	1.0

1 Introduction/Purpose

1.1 Purpose

The purpose of this document is to specify the software design specification of a software system called Encost Smart Graph Project (ESGP).

1.2 Document Conventions

This document uses the following conventions:

- ESGP: Encost Smart Graph Project
- GSL: Graph Stream Library

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: The developer who wants to use this document to be able to structure and build this software.
- Tester: The tester needs this document to validate the extra requirements specification and have a general idea of how this software will be used.

1.4 Project Scope

Encost is a new and emerging Smart Home development company. They manufacture a series of Smart Home and IoT solutions, including Wifi Routers, Smart Hubs and Controllers, Smart Light Bulbs, Smart Appliances, and Smart Whiteware. Encost is interested in investigating how their smart devices are being used and connected within households across New Zealand. They have worked, in partnership with energy companies and their users, to gather information about the smart devices that have been in use in 100 New Zealand homes between April 2020 and April 2022 (called the Encost Smart Homes Dataset).

The Encost Smart Graph Project (ESGP) is a software system that enables the visualisation of Encost's devices using a graph data structure. When provided with the Encost Smart Homes Dataset, ESGP enables users to view all of the devices in the dataset, along with their connection to one another. It also provides verified users with summary statistics on device distribution, location, and connectivity.

2 Specialized Requirements Specification

2.1 Extra product requirements

Loading the Encost Smart Homes Dataset:

- The system will keep a backup of the Encost Smart Homes Dataset file in another location (backup folder) in case it is not found in its default location.

Loading Custom Datasets:

- If the custom dataset has any items that isn't made from Encost, it will prompt the user to remove any items that isn't made from Encost and allow them to (a) use the Encost Smart Homes Dataset or (b) enter a different file path.
- If the system read and processes the dataset, inform the user whether they want to (a) visualising a graph representation of the data or (b) viewing summary statistics.

Calculating Device Distribution

- The system should output their figures to the console in a table format to be clear and concise.

Calculating Device Location:

- The system should use the information stored in the graph data structure to calculate the average number of devices that exist in each household, in each region in New Zealand.

- The system should use the information stored in the graph data structure to calculate the average number of devices in each category, for each household, in each region in New Zealand.
- The system should output their figures to the console in a table format to be clear and concise.

2.2 Non-Functional Requirement

2.2.1 Safety Requirements

The application will need to ensure it does not crash while the user is using it. The system will automatically ask the user whether they want to report the reason of crash to the developer via email.

3 Software Architecture

3.1 Component Diagram

Figure 3.1: The component diagram of the ESGP

The ESGP has two components which is the Graph Stream Library(GSL) and Application. ESGP is run locally with Graph Stream Library(GSL) as it's only software interface. The Graph Stream Library(GSL) is used to create graph visualisations by receiving data from the application and sending it back.

3.2 Process Diagram

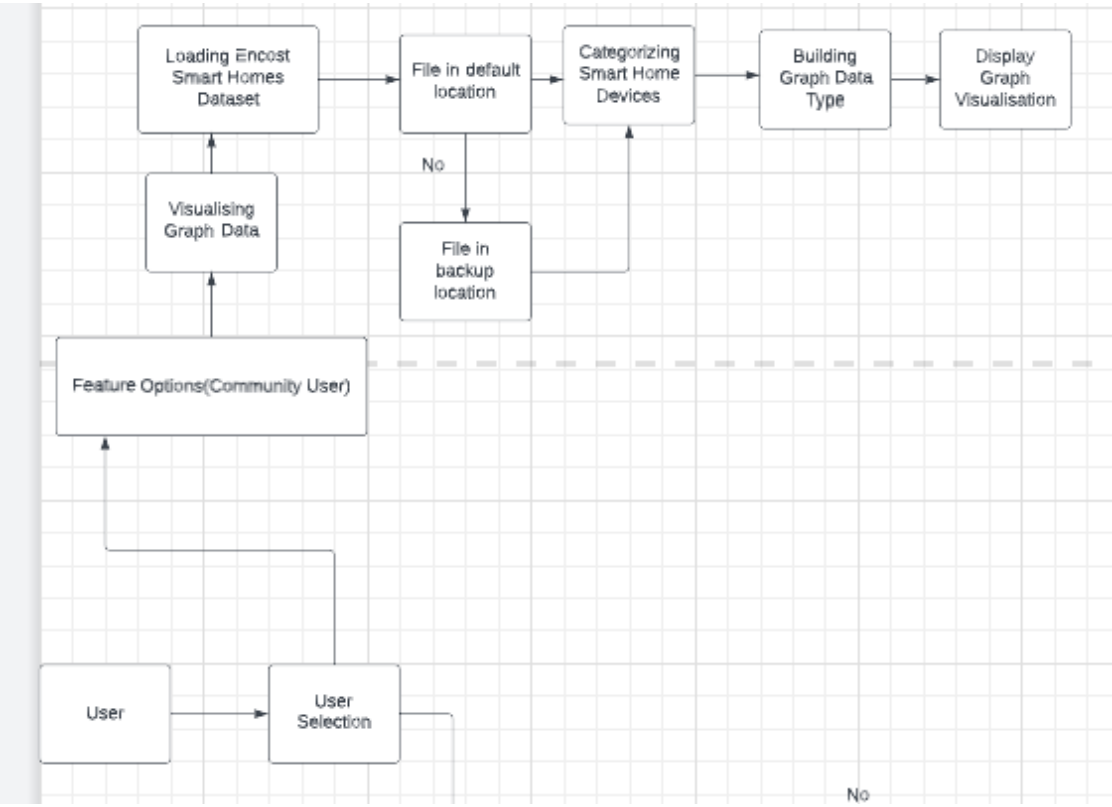

Figure 3.2: The process when user is a Community User

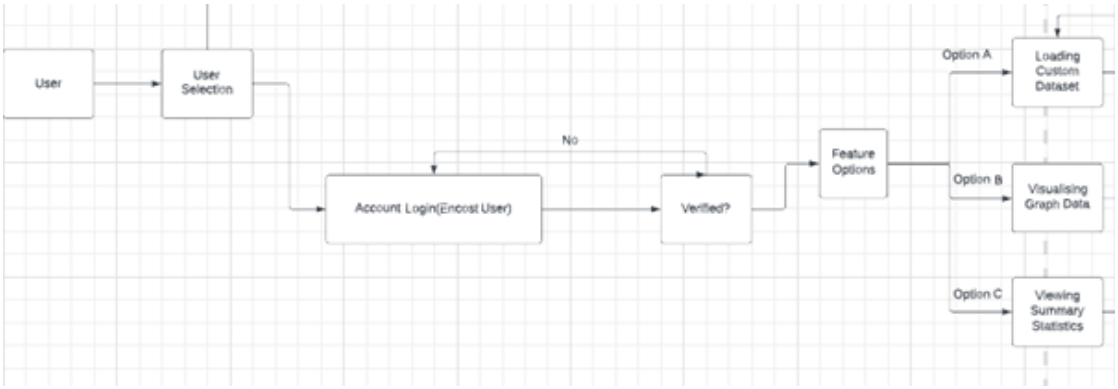

Figure 3.3: The process when user login as a Encost User

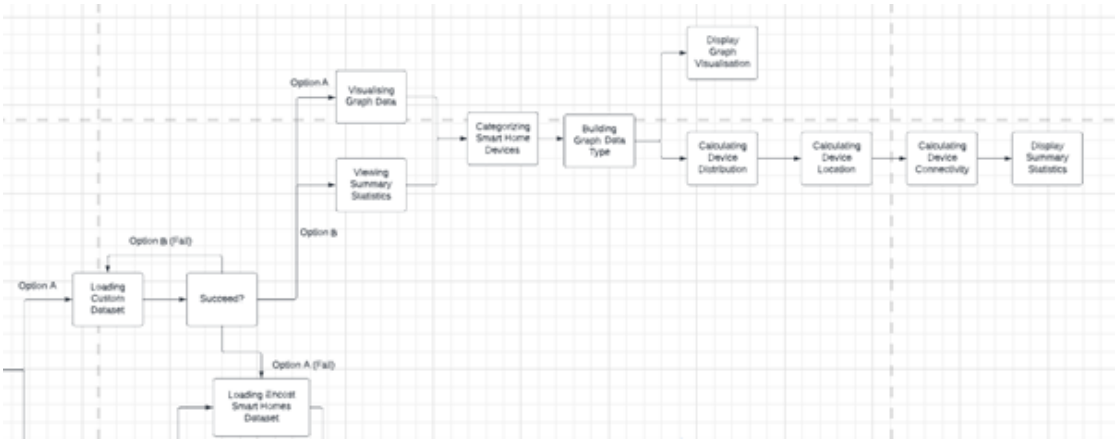

Figure 3.4: The process where user login as a Encost User and loads custom dataset.

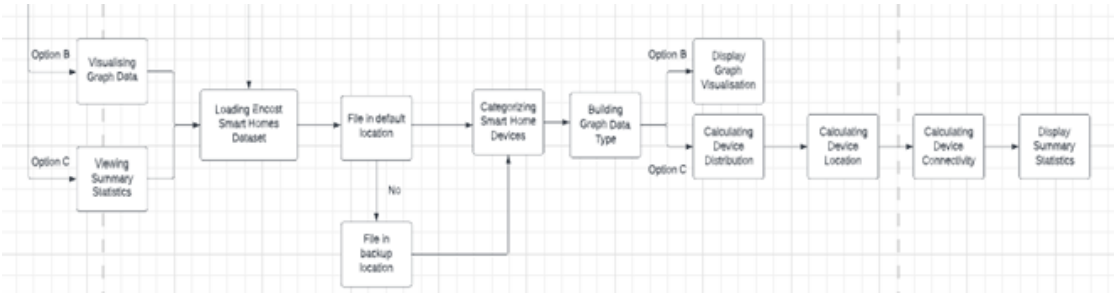

Figure 3.5: The process where user login as a Encost User and visualise graph data or view summary statistics.

3.3 Individual Classes

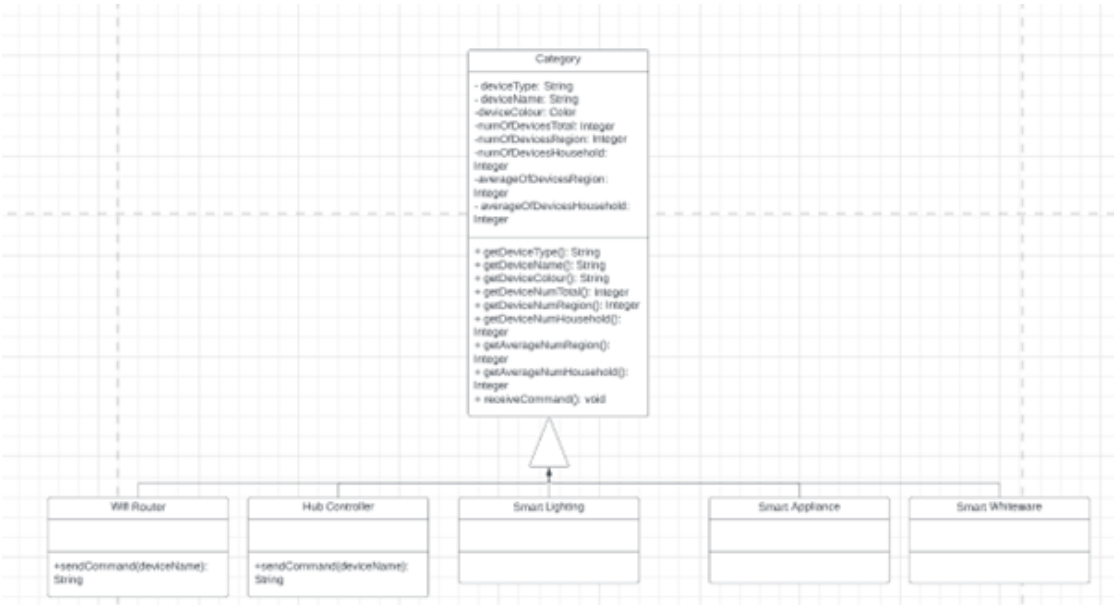

Figure 3.6: This is the UML diagram of the graph data type.

The Wifi Router, Hub Controller, Smart Lighting, Smart Appliance and Smart Whiteware classes inherits from the Category Class.

3.3.1 Category Class:

Attributes:

- deviceType: Stores as a String. It is used to tell what type of device it is.
- deviceName: Stores as a String. It Is used to tell the name of the device. E.g Encost Router 360, Encost Smart Hub.

- deviceColour: Stores as a Color. It is used to differentiate the different categories.
- numOfDeviceTotal: Stores as an Integer. It is used to see the total number of devices.
- numOfDevicesRegion: Stores as an Integer. It is used to see the total number of devices in a region.
- numOfDevicesHousehold: Stores as an Integer. It is used to see the total number of devices in a household.
- averageOfDevicesRegion: Stores as an Integer. It is used to see the average number of devices in a region.
- averageOfDevicesHousehold: Stores as an Integer. It is used to see the average number of devices in a household.

Methods:

- getDeviceType(): This methods finds out what type of device it is and returns it as a String.
- getDeviceName(): This method finds out the name of the device and returns it as a String.
- getDeviceColour(): This method finds out the colour of the device by checking the type of device and returns it as a Color.
- getDeviceNumTotal(): This method gets the total number of devices and returns it as an Integer.
- getDeviceNumRegion(): This method gets the total number of devices in a region and returns it as an Integer.
- getDeviceNumHousehold(): This method gets the total number of devices in a household and returns it as an integer.
- getAverageNumRegion(): This method gets the average number of devices in a region and returns it as an Integer.
- getAverageNumHousehold(): This method gets the average number of devices in a household and returns it as an Integer.
- receiveCommand(): This method will be used to receive commands from devices and won't return anything.

3.3.2 Wifi Router and Hub Controller Class

Methods:

- sendCommand(deviceName): This method is used to send commands to other devices and will return it as a String.

4 Component Design

4.1 User Interfaces and wireframes

4.1.1 Graph Visualisation

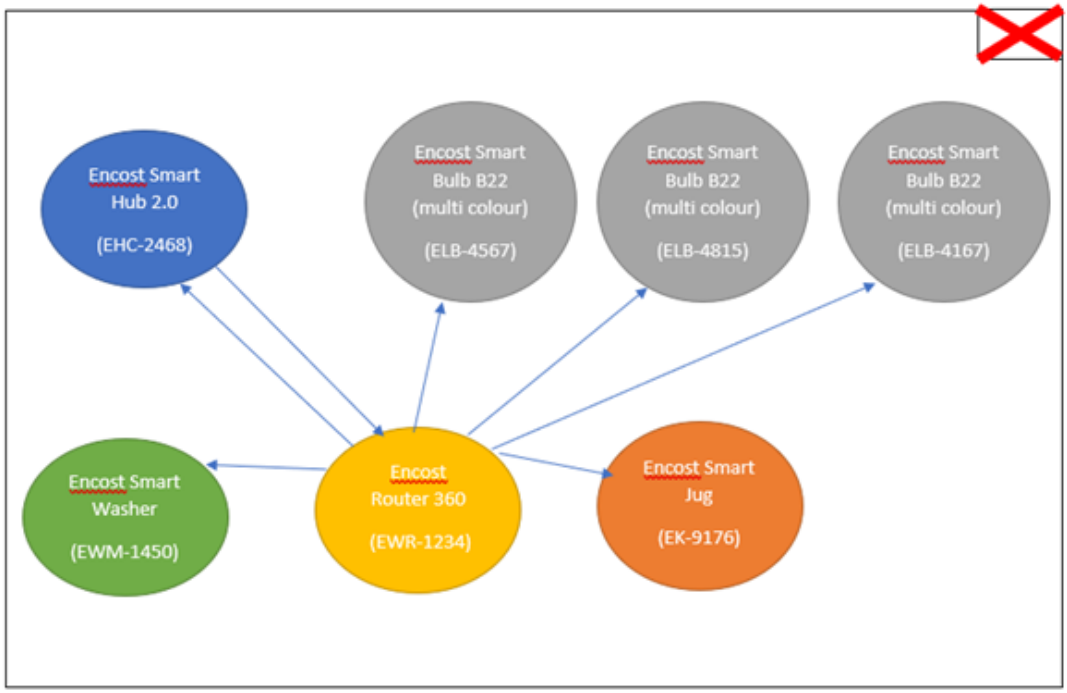

Figure 4.1: Visualisation of the graph

This graph visualisation is what the user would see after picking the option of visualising a graph representation of the data. The user can only look at the user interface and close it by clicking the "X" button.

- Yellow: Encost Wifi Routers
- Blue: Encost Hubs/- Controllers
- Grey: Encost Smart Lighting
- Orange: Encost Smart Appliances
- Green: Encost Smart Whiteware

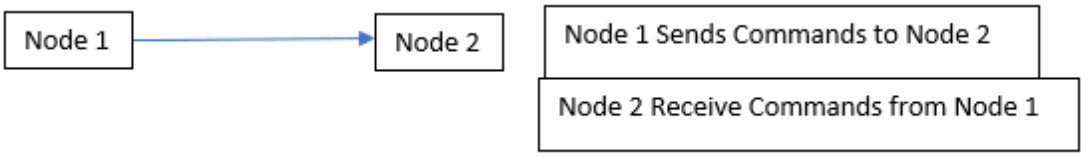

Figure 4.2: Sending and Receiving Commands

4.1.2 Calculating Device Distribution

Table 4.1: Device Distribution Output

Region	Household	Number of Devices
Northland(NZ-NTL)	350	1000
Auckland(NZ-AUK)	300	800
Waikato(NZ-WKO)	250	750
Bay of Plenty(NZ-BOP)	100	300
Gisborne(NZ-GIS)	120	400
Hawke's Bay(NZ-HKB)	150	450
Taranaki(NZ-TKI)	100	320
Whanganui - Manawatu(NZ-MWT)	125	380
Wellington(NZ-WGN)	500	1250
Marlborough(NZ-MBH)	100	200
Nelson(NZ-NSN)	200	486
Tasman(NZ-TAS)	150	300
West Coast(NZ-WTC)	127	215
Canterbury(NZ-CAN)	215	600
Otago(NZ-OTA)	175	500
Southland(NZ-STL)	300	565

This is to show an example of a clear and concise table output.

4.1.3 Calculating Device Location

Table 4.2: Average Number Of Devices Per Household In Each Region

Region	**Household**	**Number of Devices**	**Average Number of Devices per household**
Northland(NZ-NTL)	350	1000	2
Auckland(NZ-AUK)	300	800	3.5
Waikato(NZ-WKO)	250	750	1.5
Bay of Plenty(NZ-BOP)	100	300	3
Gisborne(NZ-GIS)	120	400	2
Hawke's Bay(NZ-HKB)	150	450	2
Taranaki(NZ-TKI)	100	320	4
Whanganui - Manawatu(NZ-MWT)	125	380	2.2
Wellington(NZ-WGN)	500	1250	5
Marlborough(NZ-MBH)	100	200	1.2
Nelson(NZ-NSN)	200	486	1.4
Tasman(NZ-TAS)	150	300	1.6
West Coast(NZ-WTC)	127	215	1
Canterbury(NZ-CAN)	215	600	2.14
Otago(NZ-OTA)	175	500	2.35
Southland(NZ-STL)	300	565	1.8

This is to show an example of a clear and concise table output.

5 Conclusion

This document has defined the design of the Encost Smart Graph Project (ESGP) requested by Encost. It has covered the extra requirements, software architecture and the component design.