SOFTWARE DESIGN SPECIFICATION

for

Encost Smart Graph Project

Version 1.5

Prepared by: Student # 4
SoftFlux Engineer

SoftFlux

April 7, 2023

Contents

Revision History

Name	Date	Reason for Changes	Version
	01/04/2023	Initial Creation and intro written	1.0
	02/04/2023	Software Architecture section written	1.1
	03/04/2023	Program Framework section written	1.2
	04/04/2023	Console and Graph Interface section written	1.3
	06/04/2023	Conclusion and Addition Specification written	1.4
	07/04/2023	Polish and Some diagrams added	1.5

1 Introduction/Purpose

1.1 Purpose

The purpose of this document is to provide the details and design specifications that meet the requirements of of Encost for the Encost Smart Graph Project. This Software Design Specification document explains the software architecture and component design for the project in detail through the use of diagrams and discussion. It is expected that this document will be used to develop the base version of the Encost Smart Graph Project (Referred to as ESGP throughout the document) and will also be used to test against the final product to ensure the requirements set out by Encost have been met.

1.2 Intended Audience

- Developer: The developer who is creating the ESGP will need this document to ensure that the final program created conforms to the design specifications as set out.
- User: The user of this program may want to review the functionality of the final ESGP result to determine if the program adheres to the specifications as set out by this document or to further understand how the program works at a deeper level.
- Tester: The tester of the ESGP program will require this document to ensure that the final program adheres to the design specifications as set out.

1.3 Product Scope

The ESGP program is used to help Encost understand how their devices are being used and how they are connected within households across New Zealand. The ESGP program will enable the visualisation and summarisation of the Encost Smart Homes Dataset which is a dataset that contains information regarding 100 New Zealand homes and the Encost devices being used. The program will also be sent to the 100 New Zealand homes that consented to their data being collected so that they can see a visualisation of the Encost Smart Homes Dataset.

1.4 References

SHA-256 algorithm: https://www.movable-type.co.uk/scripts/sha256.html
GraphStream: https://graphstream-project.org/

2 Specialized Requirements Specification

Within the Software Requirements Specification document, it is required that the ten passwords are encrypted to maintain security. However, it was not indicated how the passwords are encrypted. We at SoftFlux recommend the use of the SHA-256 hashing algorithm to maintain security. SHA-256 is a widely used and secure hashing algorithm that has no known vulnerabilities that make it insecure. Other popular hashing algorithms have known vulnerabilities that may compromise the security of the credentials.

It was also not clear what file format should be accepted by the program when loading custom datasets. It was clarified by Encost that the Encost Smart Home Dataset was in the .txt filetype. We have decided that all custom datasets must also be of the .txt file format to ensure reliability of the application.

It was not specified how loading a custom dataset would affect being able to load the Encost Smart Home Dataset again. Encost clarified that it was a design choice that we, SoftFlux, could make. We have decided that there will be an extra option at the feature options select prompt for the user to unload the custom dataset they currently have loaded and reload the Encost Smart Home Dataset. This option will only be available to the user if such a custom dataset is loaded.

It was unclear what information would need to be displayed on the graph, apart from the fact that the graph needs to include a way to distinguish the category a node is in and the connection capability any device might have. We have decided that the graph will also include labels alongside a node to determine the name of a device and the household it belongs to. We recommend this method as these 2 pieces of information are important for determining what a device is and where it is geographically. The other information in a node may not be as useful for the user and could clutter the graph.

3 Software Architecture

3.1 Component Architecture

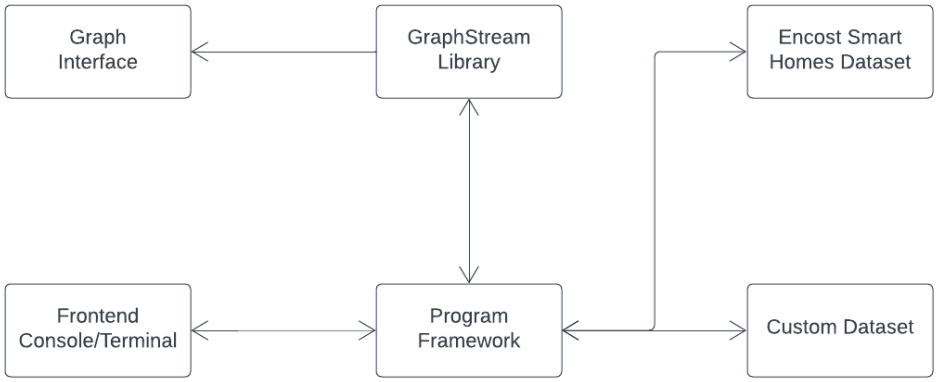

Figure 3.1: Component Diagram

This diagram demonstrates how the separate components of the ESGP program interact with each other. Users will interact with the program directly via the console. The console is the only way the user will be able to interact directly with the program to influence how it behaves. The console speaks directly to the Program Framework which handles all input from the console such as, logging in, selecting features, loading custom datasets etc. The framework stores the Encost Smart Homes Dataset and also information relating to a Custom Dataset if the user supplies one (and has entered the correct login information). The Program Framework processes data from either dataset and can output summary information to the console for the user to view. The Framework can also process the datasets and speak with the GraphStream Library in order to generate a visualisation of the data provided. This graph visualisation is then shown to the user via a Graph Interface. However, the Graph Interface does not send data back to the GraphStream Library as it will not be designed to accept input as seen in figure 3.1.

3.2 User Process Flow

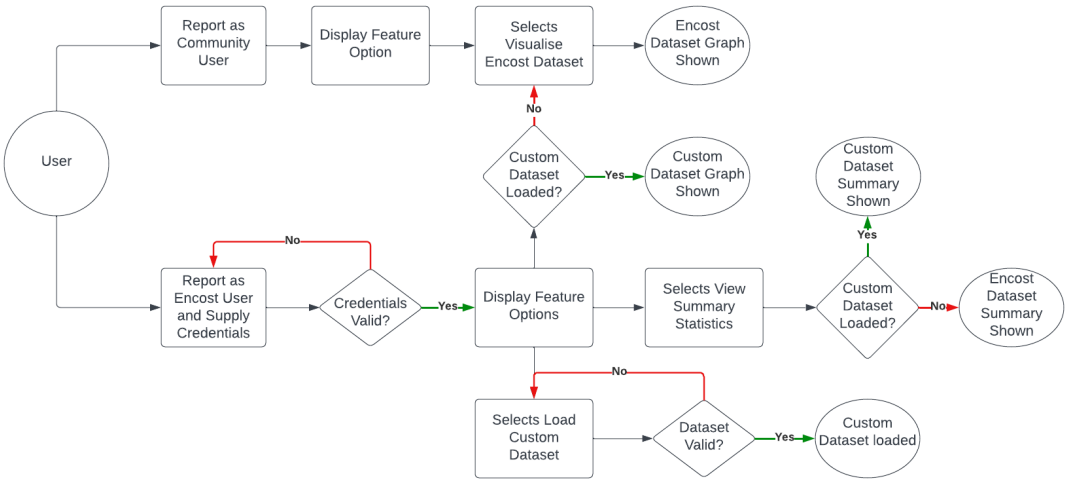

Figure 3.2: Process Diagram

This process diagram details how a user can interact with the program and the steps the system takes to display information to the user. As shown in Figure 3.2, the user can report to the program as either a community user or an Encost user. If the user specifies they are an Encost user, the program will check their credentials. Community users will only be able to visualise the Encost Smart Homes Dataset and thus it will be the only feature option displayed to them. Encost users however, will also be able to load a custom dataset and see summary information about either the Encost Smart Homes Dataset or the custom dataset depending on what is loaded. They will also be able to visualise the Encost dataset as a graph, as well as a custom dataset. When the user attempts to load the custom dataset, it must be in the valid format as required by Encost. If not, the user will be prompted to load a different dataset and the dataset supplied will not be loaded. Once a user has reported as either a community user or an Encost user, they will be able to return to the "Display Feature Options" step (that their user status is allowed to see) in the flowchart at anytime.

3.3 Data Flow

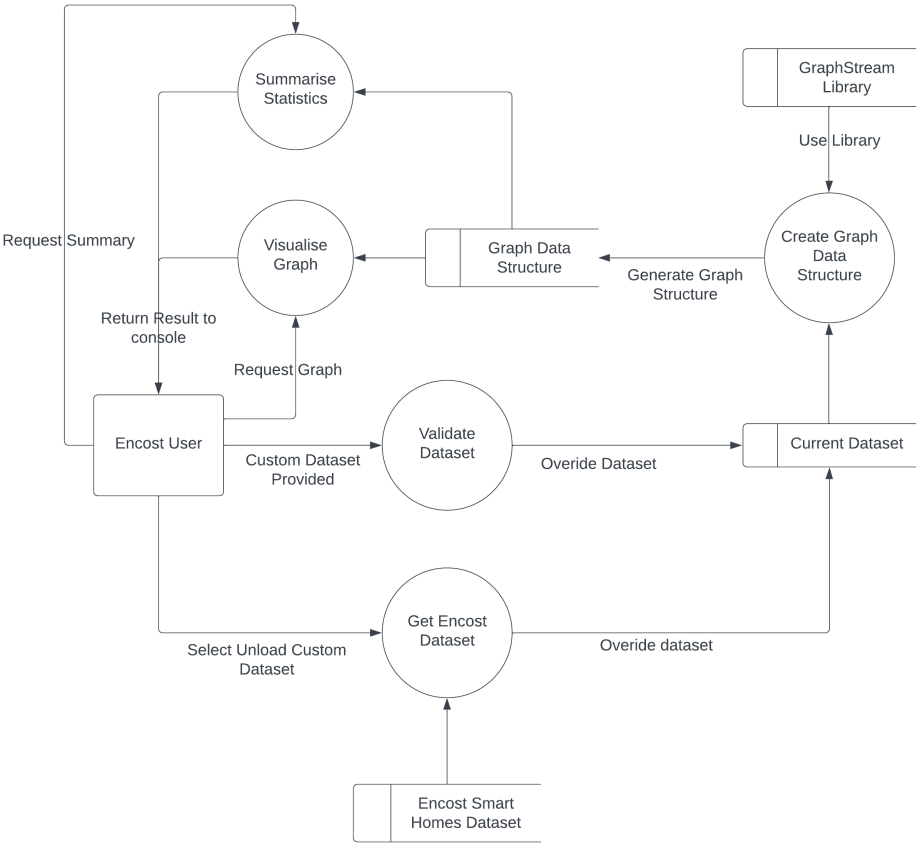

Figure 3.3: Data Flow Diagram

Figure 3.3 describes how data flows through the system and what processes take place within the program. A user can supply a custom dataset which is then validated then overrides current dataset if it is valid. The user can also select to unload the custom dataset in which the Encost Smart Homes Dataset is loaded and overrides the currently loaded dataset. The loaded dataset is transformed into a graph data structure using the GraphStream library. This graph data structure can then be displayed to the user as a summary of the statistics of the dataset or as a graph. The user is always notified that a process has taken place.

4 Component Design

4.1 Program Framework

The Program Framework will handle input coming from the user via the console and will manipulate the Encost Smart Homes Dataset, along with any custom dataset provided. The datasets will be processed according to the requirements of Encost and then be displayed to the user in the following ways:

- Visualising Data: Data will be translated into a Graph Structure and then displayed to the user via the GraphStream Library. Devices that are in the dataset provided will be displayed as nodes and edges will signify their connections. Further information regarding information displayed on the graph is described in detail later in this document.
- Summarising Data: All information regarding devices provided by a given dataset will be summarised into various forms of information. This information will be displayed to the user via the console. Information regarding the information summarised is described in detail later in this document.

4.1.1 Encost User Credentials

When a user runs the ESGP program, they will be prompted to select whether they are a Community user or an Encost user. In the event that the used selects Encost user, they will also be prompted to supply their credentials in order to access the Encost user feature options. The credentials that allow a user to login as an Encost user will be stored within the Program Framework. Encost will supply SoftFlux with ten username and password pairs. It is vital that the passwords are encrypted before being stored within the application. It is SoftFlux's recommendation that SHA-256 be used to encrypt the passwords. The framework will be able to verify that the correct password has been entered for a given username by hashing the inputted password and validating it against the stored, encrypted password.

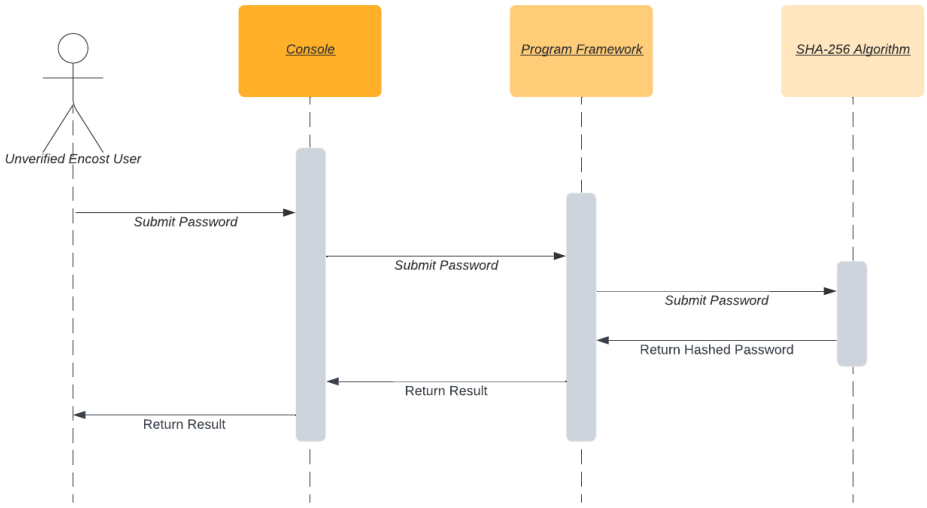

Figure 4.1: Password Sequence Diagram

As seen in Figure 4.1, the unverified Encost user submits a password to the console, the console passes the password to the Program Framework which encrypts the password using the SHA-256 algorithm. The encrypted inputted password is compared to the corresponding stored hashed password and then the result is sent back to the console which lets the user know the result of the credential input.

The Program Framework will store the indicated user-type in order to restrict the features a user may select. If a user selects the community user option, the framework will store the fact that they belong to that group. If the user selects the Encost user option, their user-type will be selected to be encost-unverified. Only once the encost-unverified user verifies that it is in-fact an Encost user by supplying correct credentials, will the users user-type then be assigned as encost-verified which will allow the user to see the full feature options available. Only if the user-type is encost-verified will the user be able to see additional feature options.

4.1.2 Feature Options

Depending on what the Framework has stored as the users user-type, the user will be presented options to select via the console. The features that are available to a user depending on their user-type are depicted in the table below:

Roles	Custom datasets	Graph visualisation	Summary statistics
Community user	No	Yes	No
Encost user	Yes	Yes	Yes

Only the options that apply to the user will be available to select in the console. The

Framework will decide what can be shown to the user and will send the options to the console for selection. The user can then enter which feature they would like in the console which will then be read by the Framework. Depending on which option the user selected, the Framework will then take the required steps to satisfy the input. If the option to visualise a graph is selected, the program will display the graph of the current dataset to the user. If the user selects loading a custom dataset, another prompt will be supplied to load a dataset. Finally, if summary statistics is selected, the Framework will calculate the required information to be sent to the console and then will display it to the user. Details regarding how the graph will be visualised can be found at 4.3. Details regarding how a custom dataset is loaded can be found at 4.1.3. Details regarding how a summary statistics are generated can be found at 4.1.4.

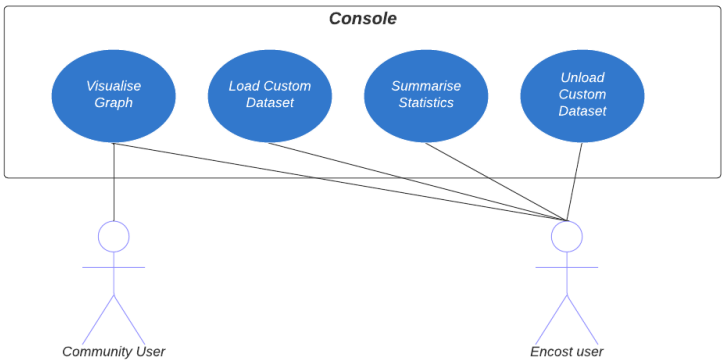

Figure 4.2: Feature Option Use Case Diagram

Figure 4.2 demonstrates the use cases for each type of user. It is to be noted that an extra, 4th option can appear to an Encost user so long as they have a valid custom dataset loaded. This option will allow the user to unload the currently loaded custom dataset and to reload the Encost Smart Homes Dataset. Any time the user does not have a custom dataset loaded, or it is unloaded, the option will not appear to the user.

4.1.3 Loading Datasets

The Program Framework will be responsible for loading the data within datasets provided into objects for each device provided. Encost will supply a dataset called the Encost Smart Homes Dataset which will be stored alongside the Program Framework. This is the default dataset to load and is the only dataset available to community users. That is, if no custom dataset is loaded, the Encost Smart Homes Dataset should be loaded always. This data set will be stored as a hidden file alongside the program and the system will always know the location of the dataset.

Valid Dataset Form

When loading a dataset, it is important that the dataset is in a form readable by the Framework. The only readable filetype will be a .txt file. The Program Framework will enforce that when datasets are loaded, they will adhere to the format that matches the Encost Smart Homes Dataset format. If the dataset being loaded does not adhere to the format required, the a prompt will be sent to the console indicating to the user that dataset that was provided was not in the correct format and thus, will not be loaded. The user will still be able to supply a new dataset in which the validation check will occur again or the user can select the Encost Smart Homes dataset instead. If at any point, a line is not in the correct format, the whole dataset will be denied.

Storage of Loaded Devices

When an option is selected to either visualise a dataset or summarise a dataset, the dataset will be processed into separate objects for every device listed. Each device has information about the device within the dataset that needs to be processed and stored within the device object. Figure 4.3 is a class diagram detailing how these devices will be created.

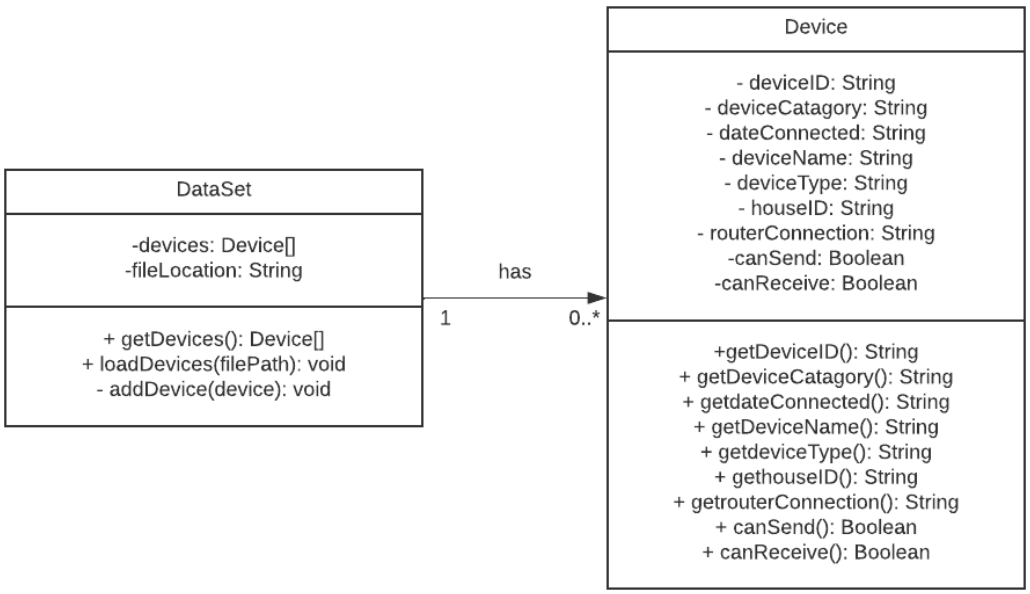

Figure 4.3: Device and Dataset Class Diagram

Devices are stored as individual objects and store all possible information about that device. It will only be possible to get and attributes value from a device and not set

it with a method as it is not intended for a dataset to change once it has been loaded. Devices will be loaded into a single DataSet object that will store all of the devices for that device within it. The DataSet object will be able to load all devices from a dataset file via the loadDevices method. This method will utilise the addDevice method to add devices to its list of devices. The loadDevices method will parse data into Device objects, ensuring to include all information about the devices within the objects.

The DataSet object will be accessed when either visualising the devices as a graph or when summarising the devices. In either case, the DataSet object will be passed to the method handling either operation. The loadDevices method will also categorise each device and assign it a category as specified by REQ-1 at 4.6.3 in the supplied SRS document.

It is important to note that only a single DataSet should ever be loaded at a time, and that whenever a new dataset is introduced, the DataSet object should run loadDevices again with the new dataset to replace the old dataset. Inititally, the Encost Smart Home Dataset should be loaded and only overridden when a valid custom dataset is loaded. The user will have the option to reload the Encost Smart Home Dataset via the feature options prompt if they have a custom dataset loaded.

4.1.4 GraphStream Data Structure

The Program Framework will use the GraphStream library to create a graph data structure. Each device that was loaded via the corresponding dataset will be stored as a node within the GraphStream data structure. Within the Device objects, the state of their connections are stored. Each connection will be represented within the graph data strucure in the form of edges between nodes. This data strucure will be used to generate both the visualisation of the dataset and the summary of the dataset.

It is possible to store attributes within GraphStream nodes. Each node within the graph structure will have the same attributes as its corresponding device. This will allow the program to display information about the nodes and edges to the user when visualising the graph and to generate summarised data about the graph data structure.

Visualising Graphs

Within the GraphStream library exists functions designed to display the graph data structure as a visualisation to the user. These functions will be called when the user selects the option to visualise the selected dataset. The Framework will handle the calling of these methods from the GraphStream library along with setting the style sheet as to which will dictate the style of the graph. The style of the graph is detailed later in the document.

Summarising Data

When the user selects the option to summarise the dataset provided, the Program Framework will summarise the data into different sets of information using the graph data structure. The summary will then be sent to the console for the user to analyse. The following types of information will be summarised and displayed to the user:

- The total number of devices that belong to each category
- The total number of devices that belong to each device type for each category
- The total number of households for each region within New Zealand (Names and codes are listed as in the ISO 3166-2 standard published by the ISO 3166 Maintenance Agency (ISO 3166/MA))
- The total number of devices for each region within New Zealand
- The average number of devices for all households in each region of New Zealand
- The average number of devices in each category, for each household, within each region of New Zealand
- The total number of devices in each category, within each region of New Zealand
- The average number of devices that are connected to a Encost Wifi Router
- The minimum and maximum number of devices that a Encost Wifi Router is connected to
- The average number of Encost Hubs/Controllers that an Encost device is receiving commands from
- The minimum and maximum number of Encost Hubs/Controllers that an Encost Device is receiving commands from
- The average number of Encost Devices that an Encost Hub/Controller is sending commands to
- The minimum and maximum number of Encost Devices that an Encost Hubs/-Controller is sending commands to

The Program Framework will be responsible for calculating each of these sets of summaries. It will calculate this information based on the nodes and edges stored within the graph data structure. This information can be determined by the attributes of each node.

Once the summary has been calculated, it will be sent to the console within 10 seconds. If the process takes longer than 1 second, the user will be notified via the console. Information regarding the formatting of the output to the user via the console is detailed later in this document. It is important that the calculation method is efficient so that the Framework does not take more than ten seconds to load.

4.2 User Console

The user console will use the command prompt console as supplied by Windows 10. All user input will be inputted via the console and all information except for the graph visualisation will be outputted to the user via the console. The console acts essentially as the controller of the program as it interacts directly with the Program Framework to do all the required specifications for the ESGP program.

4.2.1 User-type Selection and Login

When the user first runs the program, a welcome message will be displayed and they will be prompted to select if they are either a community user or an Encost user. The option the user selects will be handled by the Program Framework. If the user selects the community user option, the Framework will advance directly to the feature option selection stage of the program. However, if the user selects the Encost user option, the console will prompt the user for credentials. The user will be instructed to either type '1' if they are a community user or '2' if they are a Encost user. Any invalid input will be denied and the user will be asked again to supply a number corresponding to the user type they are.

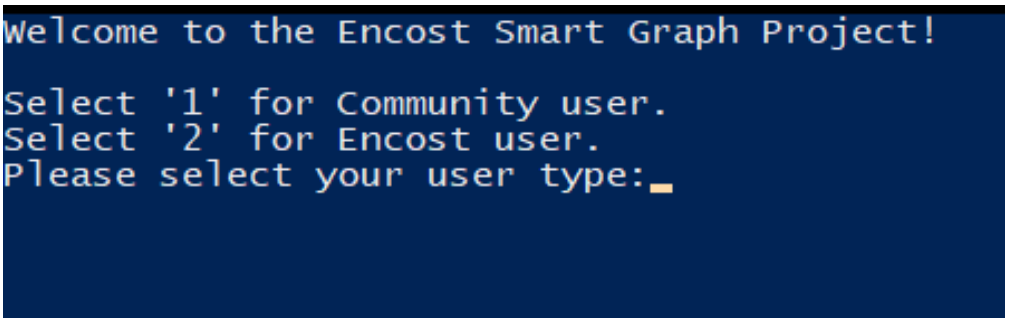

Figure 4.4: Welcome Message

When the Encost user option is selected, the first prompt to the user will be to enter a username. Upon entering a username, the framework will check against the credentials to validate that the username supplied is valid. If the username is valid, the user will be prompted to enter a password for that username. If the username is not valid, the user will be notified that the username does not exist and to enter a different username. When the user is prompted for a password, the password provided will be hashed using SHA-256 and compared against the stored and hashed password corresponding to the username supplied. If the hashes do not match, the user will be prompted to enter the password again to try again. Upon 3 unsuccessful attempts, the user will be returned to the initial welcome prompt. If the hashes do match, the user will be successfully logged in as a verified Encost user.

4.2.2 Feature options

Upon reaching the feature options prompt, the Program Framework will determine what options the user should be able to see. All users will be able to see the visualise graph option, but Encost users will also be able to see the option to load a custom dataset and also to summarise the dataset currently loaded. When any option is selected, it will be sent to the Program Framework and the corresponding option will then be handled then displayed to the user via the console, or in the case of selecting to visualise the graph of the current dataset, via the Graph Interface.

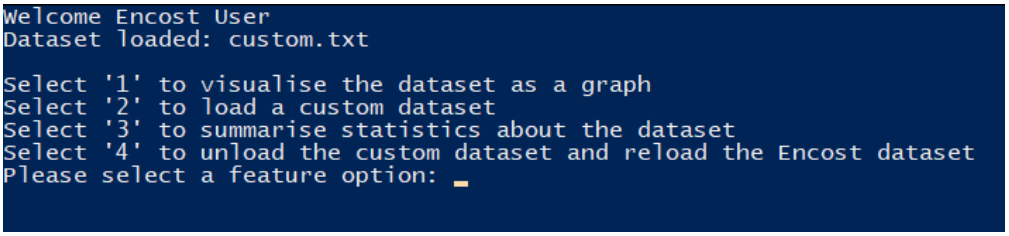

Figure 4.5: Feature Option Selection for an Encost User

The user will be always be instructed to at least type the value '1' into the console to select the visualise graph option. If the user is an Encost user, they will also be prompted to type '2' if they would like to load a custom dataset. Furthermore, they will be instructed to type '3' if they would like a summary of the selected dataset. If the user has a custom dataset loaded, they will be prompted that they can also type '4' to unload the custom dataset and re-load the Encost Smart Home Dataset. This is demonstrated in Figure 4.5 When option 4 is selected, it will then disappear and the Program Framework will load the default dataset over the old custom dataset effectively replacing it.

4.2.3 Graph Visualisation Option

When the user selects the visualise graph option in the feature options prompt screen, a graph will be displayed to the user via the Graph Interface. The Program Framework handles the creation of this interface along with the GraphStream library. The graph displayed will depend on the dataset loaded at that moment. The graph displayed could either represent the Encost Smart Home Dataset or a custom dataset loaded by the user. If displaying the graph takes longer than 1 second to process, the user will be notified that the graph is loading. When the user selects this option and the graph is displayed.

When the graph is displayed, the console will output a legend which will allow the user to distinguish which nodes in the graph corresepond to what category by indicating what colour corresponds the a category. The user will also be prompted to press any key to close the Graph Interface and return to the feature options prompt screen.

4.2.4 Custom Dataset Option

When the user selects the option to load a custom dataset, they will be prompted to enter the full filepath for that dataset and for the file to be of filetype .txt. The console will first determine if the file exists. If the file does not exist, then the user will be prompted to enter another filepath, or they can type 'exit' to go back to the feature options screen. The console will also deny the loading the file if the file is not a .txt file. If the file exists and is a txt file, the Program Framework will determine if the file supplied is in the valid format to be entered into the graph data structure. If it is not, the user will be notified that the format is incompatible and will be prompted to enter a different file path or to type 'exit' to go back to the feature options prompt screen.

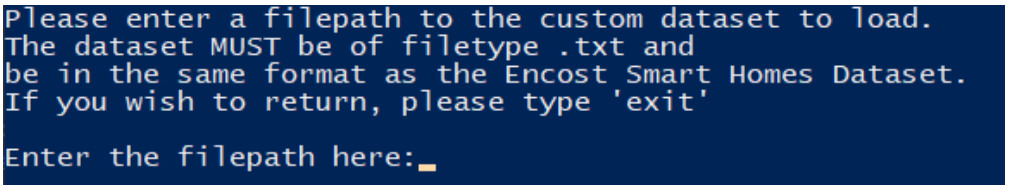

Figure 4.6: Custom Dataset File Input

If the dataset provided is valid, it will then be loaded by the Program Framework and the user will be returned to the feature options prompt screen. If loading the dataset takes longer than 1 second, the user will be notifed that the dataset is loading.

4.2.5 Summarise Statistics Option

If the user selects the option to summarise the statistics of the currently loaded dataset, the console will notify the Program Framework to summarise the dataset using the graph data structure and will return the summarised data to the console. The summarised data will be grouped in the console by:

- Device Distribution
- Device Location
- Device Connectivity

Information relating to the specific types of summaries that will be shown can be found in 4.1.4 Summarising data.

If summarising the data takes longer than 1 second, the user will be notified that the summary is loading. Once the summary has been displayed, the user can press any button to return to the feature options prompt screen.

4.3 Graph Interface

When the user selects the visualise graph options from the feature options prompt via the console, the currently loaded dataset will be displayed in a graph data structure format using the GraphStream library. All devices within the dataset will be represented by nodes within the graph, and all connections will be indicated with the use of edges between nodes. Nodes will be coloured depending on their category and will have a label which informs the user the device's name and its connection capability. The Program Framework will assign attributes to each node corresponding to the datapoints a device may have. These attributes will be used to colour code the nodes and label them.

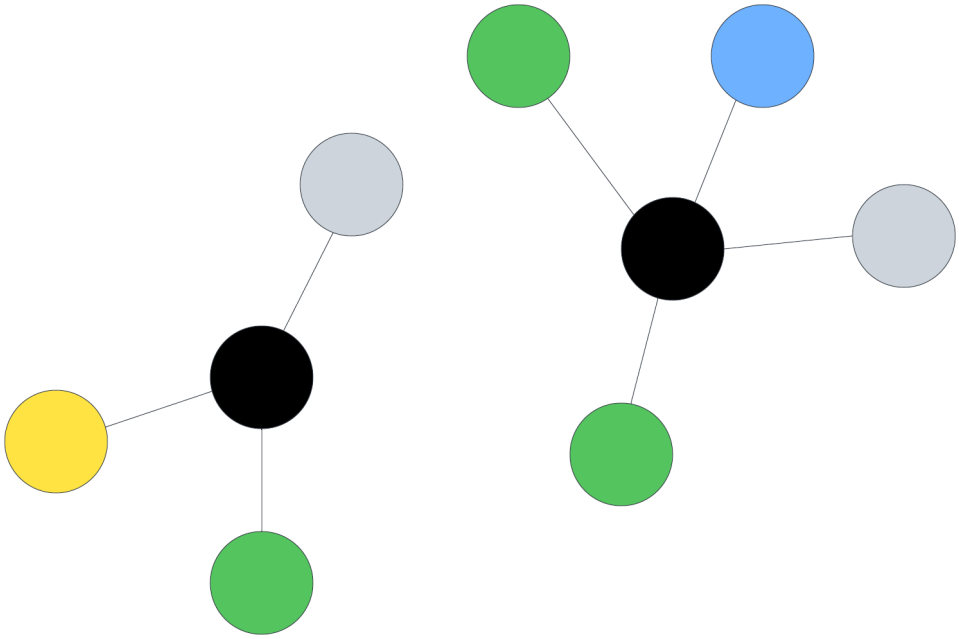

Figure 4.7: Example of Nodes and Edges

The Graph Interface can be closed at anytime by the user via either entering any key into the console or by closing the Graph Interface window. In both cases, the user will be prompted to select a new feature option again.

4.3.1 Distinguishing Category

The graph visualisation of the loaded dataset will allow the user to distinguish the category any one device belongs to, as well as its capability to send command or recieve them. Distinguishing the difference between the categories of each device can be determined by accessing the nodes attributes and finding the category it was assigned, this is handled via the Program Framework. The category a node is in will be able to be distinguished based on the colour of the node. The colouring scheme is as follows:

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

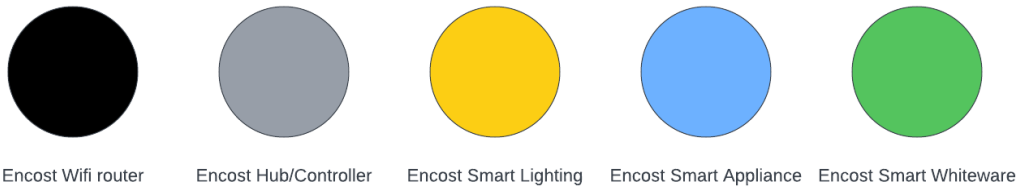

Figure 4.8: Node Colour Representations

Refer to Figure 4.8 for examples of the colouring scheme. Users will be able to determine what colour corresponds to what category via the console, which will display a legend indicated which colour means what category.

4.3.2 Distinguishing Connection Capability

The connection capabilities of a devices will be indicated via a text label, along with the device's name. The format in which the label should follow is:
"<Device Name> - Household: <householdID> - Can Send: <true/false> - Can Receive: <true/false>"
This label should not cover the node itself or other nodes and each type of information should be placed on a new line. Refer to the Figure 4.9 for an example.

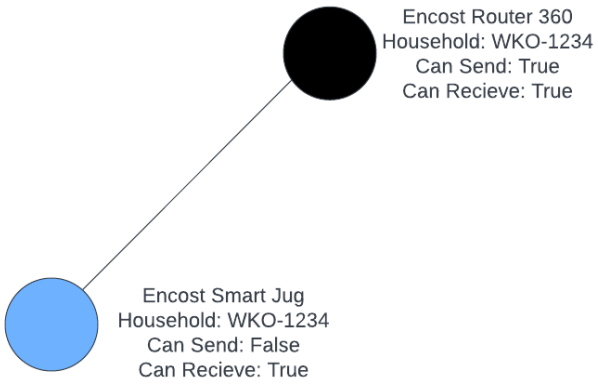

Figure 4.9: Node Details

5 Conclusion

In conclusion, this Software Design Specifications document has provided an overview of the Encost Smart Graph Project's design specification. The software has been designed to meet the requirements as set out by Encost in the Software Requirements Specification document. We are confident that the Encost Smart Graph Project will fulfill the needs of its users in order to visualise and summarise the Encost Smart Home Dataset and any other custom dataset provided.