ELEN4002 - Lab Project Plan Preliminary Report on the Design and Creation of a Wits Analytics and Visualization of Energy Systems

Marion Heimann - 788579
Tristan Kuisis - 812587
Supervisor: Prof Ken Nixon

Abstract

The main purpose of this document is to outline a project plan for the data analytics and visualization of energy systems across the multiple Wits campuses. The multiple energy meters installed across the properties have been gathering data for a number of years. A web server is constructed that is capable of autonomously and repeadly drawing the data from the database. The server will then host a portal that is capable of displaying the data to a user. The web portal will also be used to generate unique visualizations of the data from the sensors.

Index Terms

Selenium,

I. Introduction

THIS report details the project plan for the data analytics and visualization of energy systems that make up the Wits campuses.

There are a number of large energy requirements that Wits has-electricity, water, natural gas, petrol, diesel, as well as a number of other resources which make up smaller fractions of the universities energy usage.

This project, firstly makes use of the electricity meters installed throughout the buildings on the properties. These meters have been providing data for varying amounts of time, as well as data outages occurring occasionally (this will have to be dealt with). The wealth of information that these data loggers provide allow for the creation of a web server which is capable of drawing this data from the database, and making use of this data to visualize the energy usage across Wits. The web server will be instrumental in allowing a user to visualize this energy in a multitude of ways. The timelines, methods, tools, and a number of other details are discussed throughout this report.

II. Project Specifications

A. The Data

As discussed above, there are a number of energy meters placed throughout the Wits properties. There are approximately 310 data loggers that are connected to the current web portal used by the university. The system in place, run by IST [1], is based off-campus. This means that their web service and data is housed on their side. All of the data retrieval (from the data loggers) is done through the Wits network and through in some cases through mobile data. This element of the project is discussed further down in sections XIV, XIII, XV.

There are two other highly relevant data sets that will be used for the core part of the project. These are: the energy generated by the solar panels placed on top of a number of buildings throughout campus, and the weather of Wits. These two data sets will allow for a unique picture to be painted which further enhances the visualization of the system as a whole.

Currently, this is the chosen data sets that will be used for the core of the project, however, if time permits, the university holds a wealth of information regarding the location of individuals throughout the university, this is provided with the use of the Integrated Campus Management (ICAM) system which is used for the access control for the university. This system is likely to unlock further insights on the universities energy usage and how the movement of individuals affects the energy usage of specific areas.

B. Back-End

In order to run the system proposed, a number of operations are required to take place in a selection of programming languages, and all of these should be able to communicate with one another such that the process described by .... can take place.

The back-end can be described by a local web server that is run on the users machine (in this case, the server is run on each students' personal computer), and it is important as it allows the simulation of a server that will (ideally) eventually be placed onto a standalone system such that users with access to the internet will be able to use the system anywhere. This section is discussed in further detail in section XII.

C. Front-End

In order to allow for the visualization of energy, the use of standard front-end web tools will be employed in order to provide the user with interaction. The back-end will provide all of the processing requirements, and will manage the data. the front-end communicates with the back-end (server) in order to illustrate the required information. The three main tools (commonly used in any website), are: Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript. These three tools communicate with each other and the server in order to provide the user with the required information when viewed from a web browser.

It is this part of the system where a multitude of visualization tools are used to gain insights into the data. These tools will be discussed in section VIII

This completes the overall description of how the system is constructed and how it functions in order to get the required functionality.

III. Scope Statement

There are four sections to the scope for this project, these are: data gathering, back-end system, front-end system, and the ability to use the front-end system to gain insights. All of these sections for part of the scope for the project.

The data gathering system is required to be capable of autonomously and periodically drawing data from the IST server for all of the data loggers and all of the data that they have attained since installation. There are a few details for this section which is left for section XIII as the system does not always need to gather in the same way.

The back-end system takes part in the data gathering process, this is why it is important that if functions as required. The back-end deals with all of the data storage, as well as processing of the data in order to send it through to the front-end such that a user can interact with the system. This is used as the system which manages the storage and manipulation of the data which is highly important. In order to reduce the work required by the client (front-end), the majority of the presentation work will be done on the server, then once it is ready, sent through to the client. There may be cases where client side processing will be used, this will be in cases where the visualization tools require it, an example of this will be Dygraphs which required the clients browser to interact.

The front-end is important for any web application, as this is the interface which allows user interaction, which is where the focus and importance rests. The front-end makes use of HTML, CSS, and JavaScript to generate the standard format of the system. This configuration is then used with a number of other tools in order to have multiple types of data visualization. The details of the front-end is discussed in section VI-C.

Finally, the fourth key part of the scope is that of the data visualization. This will be done with the use of many tools and methods such that insights can be gained from the data. The most important decision for this part of the scope is that the system should be designed such that a greater understanding of the system can be gained. Two examples of this are: verify that the billing from City power matches that which the system measures, and another interesting, possibly engineering test, is to verify how much energy has been saved with the exchange of the old lights to all LED lights throughout campus.

IV. Timeline

It is important to illustrate a basic timeline for any project. In this case, the timeline forms a similar order to how the scope is laid out. The following list illustrates the important dates for the project, this indicates when certain parts of the project need to be completed.

- 16 July - Project Plan Due
- 16 July - Lab Project Officially Begins
- 17 August - Deadline to change project name
- 27 August - Staff Inspection Day
- 28 August - Open Day
- 3 September - Project electronic submission deadline
- 13 September - Laboratory project conference - presentations and interviews

This indicates that there are approximately six weeks in which the project takes place. There is a specific method in which the project will take, and this is such that minimal functionality can be gained from the system from the earliest point possible. This means that each of the four major sections laid out in the scope will be put together such that the system can be up and running to test out simple functionality. This method can be compared to that described in The Pragmatic Programmer, where they illustrate the method of using tracer rounds [2], where the use of a full paper design will lead to a successful product, however, in some cases, it may be more beneficial to make use of *tracer rounds* in order to get feedback within a shorter time period and to see the effects of the system quickly. The nature of this project is such that the requirements from the *client* can be relatively vague, and this type of project is unique as it is quite in it's dataset and use case. This implies that many of the tools and methods that are used throughout the project will change over time as there are many unknowns throughout the process.

These unknowns can be reduced with the use of prior research and comparing similar systems, however, there will always be a non-zero amount of uncertainty for parts of the project.

This is why a lot of prior research and testing has taken place before the project begins.

The following figure, Fig. 1 illustrates a first estimate on how the project timeline will be structured. The gantt chart is set out in week days, as there are six weeks.

Where:

- A - Data Retrieval
- B - Data Storage and Manipulation
- C - Back-End Setup
- D - Back-End Link with Data Retrieval System
- E - Front-End Design
- F - Integration with visualization tools
- G - Reassess System
- H - Iterative Methodology
- I - Documentation
- J - System Testing

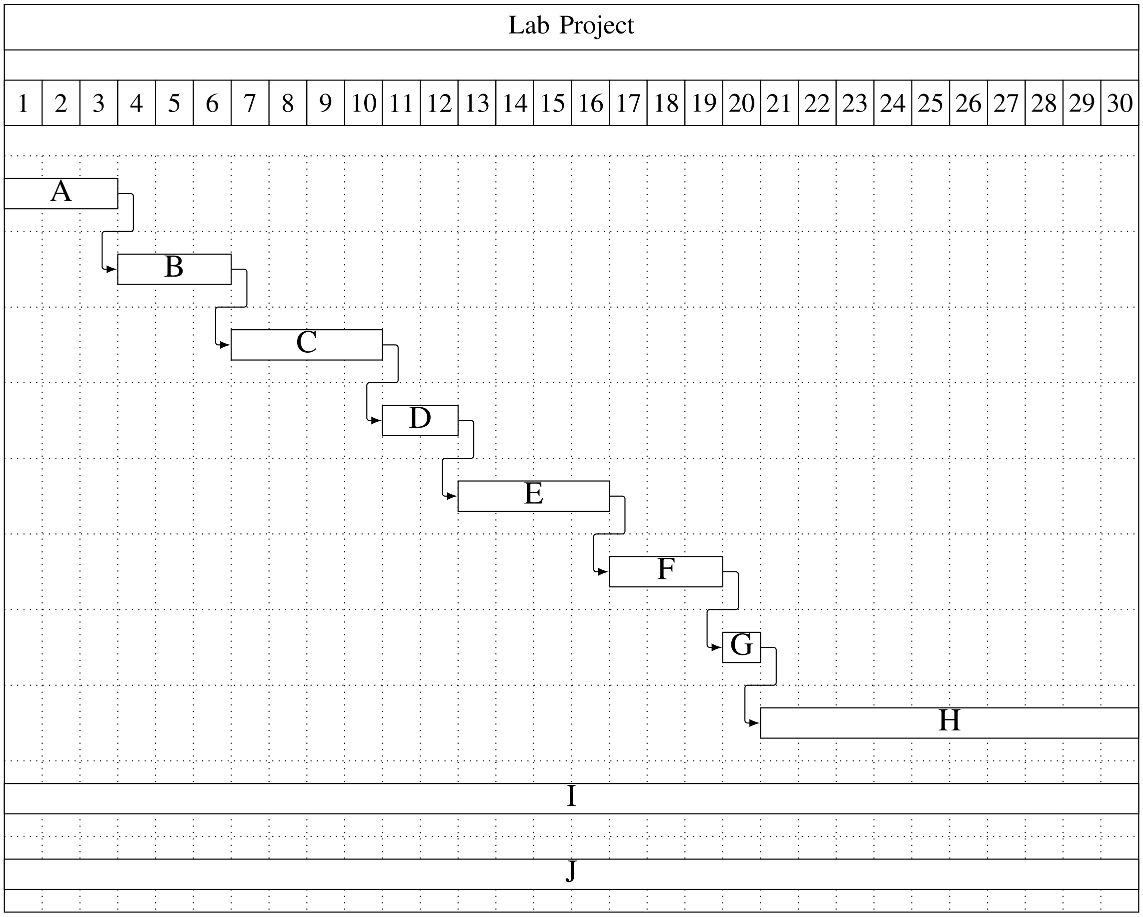

Fig. 1: Estimate Gantt Chart of Project

As it commonly works with gantt charts, there will be tasks which take less time than is scheduled, and they can also run over time, this shifts sections of the chart and effects tasks further down the line. There are a number of tasks which run throughout the entire project, these are documentation and system testing, this is done as it helps with the development process to run smoothly and any changes that happen to the system are always tested so that different components in the system are not affected with changes to any other system. This testing helps the interfacing of the different components run smoothly, this is discussed in section X.

Finally, there is a task called *Iterative Methodology*, this is where the *tracer bullets* system comes into play, all of the tasks before this are set out to get the system up and running so that vulnerabilities and loopholes can be found in the system, and further visualizations can be added to the system. Once the data sets and the system as a whole is set up, it becomes a matter of working with the data to provide multiple visualizations with that data.

A. Working Times

It has been recommended that the working times for this project should be kept to a standard 08:00 to 17:00 working time, this is done this way so that the project can simulate a *real* project. There may be cases where these times will have to be altered such that certain tasks can be completed within scheduled times.

Throughout the project, meetings with the project supervisor, as well as other individuals will take place, these will be used for consulting purposes and follow up meetings with progress on the project.

B. Milestones and Deliverables

There are a number of milestones that are used for the duration of the project, these are mostly made up of the major sections in the scope, and are illustrated in the list below:

- Retrieval and Storage of Entire Database
- Periodic Retrieval and Storage of Database
- Back-End Setup
- Front-End Skeletal Design
- Visualizations
- Refinement of each stage

These are set as the major milestones and deliverables that are currently set out for the project. The nature of the data that is presented by this system affords a highly important resource for major stakeholders in the Wits community. This is why a number of stakeholders have been contacted and asked for their input on the system. The input from these stakeholders can guide the later stages of the project as it will be their ideas which determine which are the major visualizations and information that should be inferable from the data.

V. Risks

Risks take part in any project, these risks need to be: assessed and analysed, evaluated, and in some cases, treated and responded to. The risk management process and methods will be applied to these risks such that the success of the project is not compromised. Some of the risks posed by this project are unavoidable and will need to be dealt with on a case by case basis. Other risks can have contingency plans in place. A list of possible risks are discussed in the subsections to follow.

A. Data Outages

There are always cases where data loggers are damaged, data corruption occurs, or data is lost. This can happen for a number of reasons. This results in outages in sections of the data. There are a number of ways in which the system can be made to estimate the missing data points, however, this will never represent the actual data that was present in the first case.

The way in which the data loggers have been installed throughout the university is such that many were installed at different times. This means that, as one looks further back into the past, the picture painted by the data becomes less indicative of the energy usage across most of the university.

In some cases, it is can be useful to visualize just where the data is missing, rather than estimating what the data was in that outage. This implies that the project can make use of this missing data to its advantage. There are a number of tools that allow one to make use of this fact [3], [4].

This fact about the system can end up requiring a large amount of work, which can mean manipulating the data to suit the users needs.

A second important point to note about the data outages is where the data is coming from, this is introduced in the following subsection.

B. Web Scraping Dependencies

It must be known that a major portion of the project relies on the data relayed by IST. IST are in control of the database, and currently, the only method to access this data is through their web portal. Gathering the data from this system has posed some potential challenges which have currently been overcome with the use of web scraping tools. These tools simulate user interaction on the website in order to download the data from the website in chunks. The web scraping tool is manipulated such that it enters the information and selects the buttons that a user would do when they are interacting with the website. This poses a risk for the gathering of the data in the case that the website is changed and the scripts used no longer function as they did when created. The use of this web scraping tool is employed for this first section of the project as direct access to the database has not yet been granted to the project. The risk that this poses is unlikely to heavily impact the project as the chances of the website being altered within the allotted time for the project is low.

There solar panels provide a non-negligible amount of energy to the campus, and this system provides users with energy generation data. The project will also make use of web scraping in order to draw the relevant data. This method of drawing the data will also have the same risk posed as it can stop functioning with changes to the site structure.

Included in this section is that of the security and validity of the data provided from this system. It is assumed that the data housed by IST will be safe and secure for the duration of the project and for the continued use of this system to be created. The access to this system has been granted to the student group for the duration of the project and it is assumed that access to the site will be granted for the duration. During the research of the web scraping tools during the July holidays, there were periods where the site was down for maintenance, and there were also periods where it had reduced performance, this resulted in slow page load times. Fortunately this happened before the official start of the project, however, if it does occur during the project, it can have an impact on the projects timeline.

The University is a public research institution, as such much of the information that it generates is available to the public. The information gathered from this system has not been released to the public before this point. As the energy costs are publicly available, this project has a large responsibility for how the data is handled. The system currently in place is used as an accounts verifier among other uses. This means that the data on the system should not be tampered with, corrupted, or lost. The relevant stakeholders will assist in the correct way to acquire this data safely. During the web scraping testing, it appears as if the security of the system is not as it should be as the users have direct read and write access to the data. This implies that the handling of this system should be handled with care.

C. Licensing

This project is likely to make use of a large number of programming languages and tools in order to get the functionality required. Consequently, the licenses of these tools need to be considered for use in the project. At this stage of the project, it is not planned for the program to be sold as a product or make any revenue. However, in such a case that the project is successful, it makes will be easier for future work to take place if the tools used have GPL or MIT licenses. This reduces the future work to change out the tools which require licenses to be purchased. Some simple guidelines are used for the choice of tools throughout the project, the licenses that are chosen will be along the lines of the GPL, MIT, or similar [5].

VI. System Architecture

This section makes use of a number of figures in which to illustrate the structure of the system. Illustrated in Fig. 2 represents the data loggers and their connection to the relevant buildings, transformers, and systems- each of which is then connected to the IST system.

In the figure, the devices which are being measured can be from multiple different sources. These can be connected to main transformers, or to individual buildings. The managing of all of these buildings and how the systems are interconnected will require some work as there will be cases where one data logger will be measuring a main incomer, and several other data loggers will make up the buildings that are supplied by that incomer. Once the data has been drawn from the data loggers, it will be a matter of piecing together the different sensors so that the system can be represented. The line connecting the data loggers to their relevant systems will be direct connections via the sensors. The data loggers are then connected to the communications channel via different methods. Some of the data loggers are connected to the server via the Wits network, and some via 3G or other communications media. The methods by

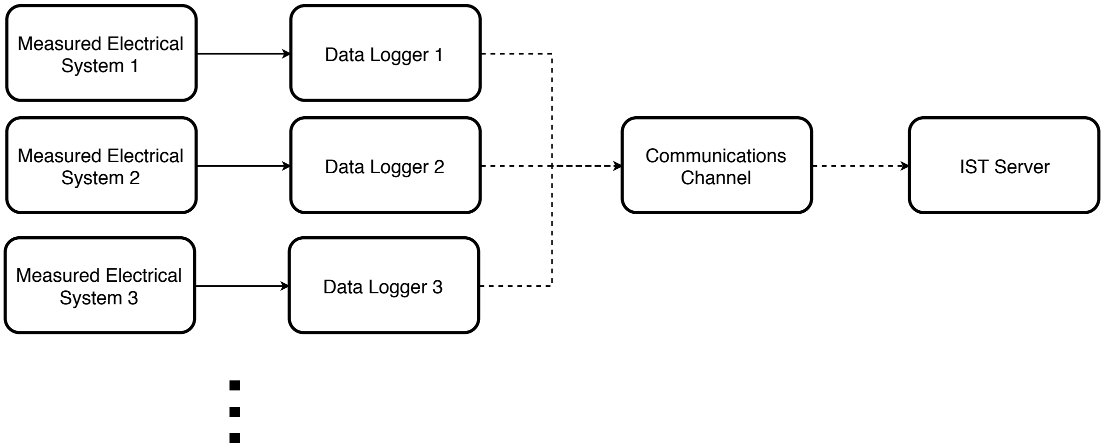
Measured Electrical System 1
Measured Electrical System 2
Measured Electrical System 3
Data Logger 1
Data Logger 2
Data Logger 3
Communications Channel
IST Server

Fig. 2: Data Loggers and their Connection to the IST Server

which the data loggers are connected to the system are not of importance to this project and will not be investigated, unless it proves to be beneficial to get an idea of which method of communication results in higher data outages.

The IST server, as discussed above draws data from the meters every thirty minutes. This is then stored on the IST database for use on the ecWin web portal.

The next section is that of the back-end, it becomes pertinent to illustrate the two together as they work together to create the system.

- Back-End
  - Node.js
  - Python
  - Python Flask
  - Python Beautiful Soup
  - Python Requests
  - Selenium
  - Tableau
  - mySQL
- Front-End
  - HTML
  - CSS
  - JavaScript
  - D3.js
  - Dygraphs
  - Google Maps

A. *The Back-End*

The back-end will be run on local machines such that the system can be hosted locally while development is under way. The local server will be run by making use of Python's Flask package, this will work hand in hand with Node.js in order to manage all of the different languages and tasks that are required to take place within the system.

Python will be handling the majority of the calculations and data manipulation for the system. There are multiple Python packages which are used these are listed below: Python Beautiful Soup: Web scraping tool that draws data from the HTML pages required. Python Requests: Another web scraping tool which allows the system to make HTTP requests to specific web pages and gather data from those sites. Python

Selenium: Selenium is a package that is available from the Python database, it is as a higher level of web scraping, such that a script is capable of further and more advanced web scraping capabilities. This tool is capable of interacting with the JavaScript layer of the web page, where the other tools fall short. The use of this package is highly useful for the ecWin website as well as the solar panel web portals as they make use of AJAX web development techniques.

The next three tools: D3.js, Dygraphs, and Tableau are used for the visualizing of specific parts of the data. This not a final list of tools to be used, however, it is believed that the first two tools will feature heavily in the final system.

It is important to consider in what format the data will be stored. Commonly, standard relational databases are used for managing some kinds of datasets. However, it must be noted that Dygraphs is designed to perform well when given comma separated value (CSV) files. The database that is used should be chosen based on a number of factors: performance, size, ease of use, visualization tool requirements, etc. The database is important as this is the interface between the data, and the tools used by the system. It may be the case that one database type does not have to be the only implemented; there may be multiple database formats used for the different tools.

B. The Front-End

The front-end has been discussed in some detail above, however it is important to note that in most cases the front-end will be designed such that the majority of the work will be done by the server side. A number of tools used do, however, require some processing on the client side. The processing on the client side will come about because of the data visualization tools such as Dygraphs and D3.js.

Google Maps is an interesting addition to the front-end, as an attempt to show energy usage overlaid on a map can provide a different perspective on the system.

This is likely to be a small selection of the tools that the project will eventually use, however, these will be capable of undertaking the majority of the visualization needs.

C. Website/Portal Architecture

1) Prepare for the activity of sitemapping
2) Brainstorm the types of content
3) Define primary navigation
4) Flesh out second and third level structure and content
5) Dont forget about utility pages
6) Create notes and high-level specifications for each page
7) Designate the type of design template
8) Iterate. Iterate. Iterate.

Maps which illustrate how one navigates through the website.

VII. Resources

VIII. Applications and Tools

IX. Development Approach

agile, waterfall, scrum

Testing methods

X. Dependencies and Interfaces

XI. Documentation

documentation - each and every part of the system makes use of markdown files which enable users to interact with the system and test it out on their machines.

XII. Server

XIII. Data Gathering

weather data

The system has a collection schedule set at thirty minutes, this means that each of the data loggers is sent a request, and the data is the sent to the IST servers, where it is stored in their database.

IST host a web service which allows customers to view the relevant information from these data loggers. The web portal allows the user to view a range of details about the system, the homepage of the system is illustrated in Fig. 3.

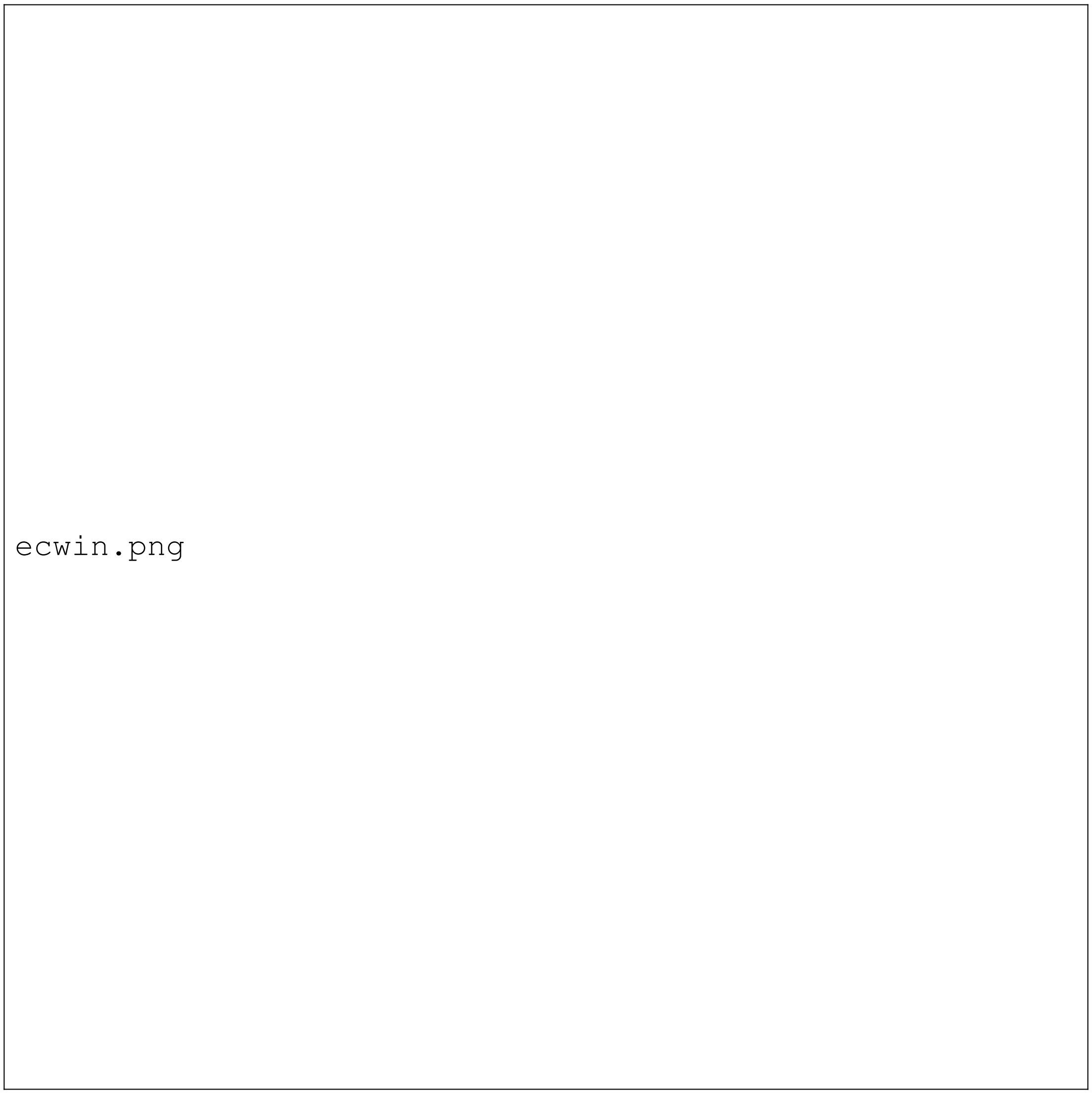

Fig. 3: ecWIN Web Portal

This is the web portal that is to be used to gather the data from, this can be done by navigating to two different sections of the system: through the data editor section, and through the reports section. Both of these are, however, limited in their functionality. It must be noted that access to the IST database directly has not been granted as of writing this report, this is why the following methods have been utilised.

The data editor has a major shortfall in that one can only view the data loggers' data in small intervals, this means that when gathering the data for the specific meters, the system will have to download multiple files and then stitch them together to get complete representation of the data from that meter.

The reports section allows the user to view the data with no limits on the date range, however, when one selects the option to export the data, an error occurs (illustrated in Fig. 4).

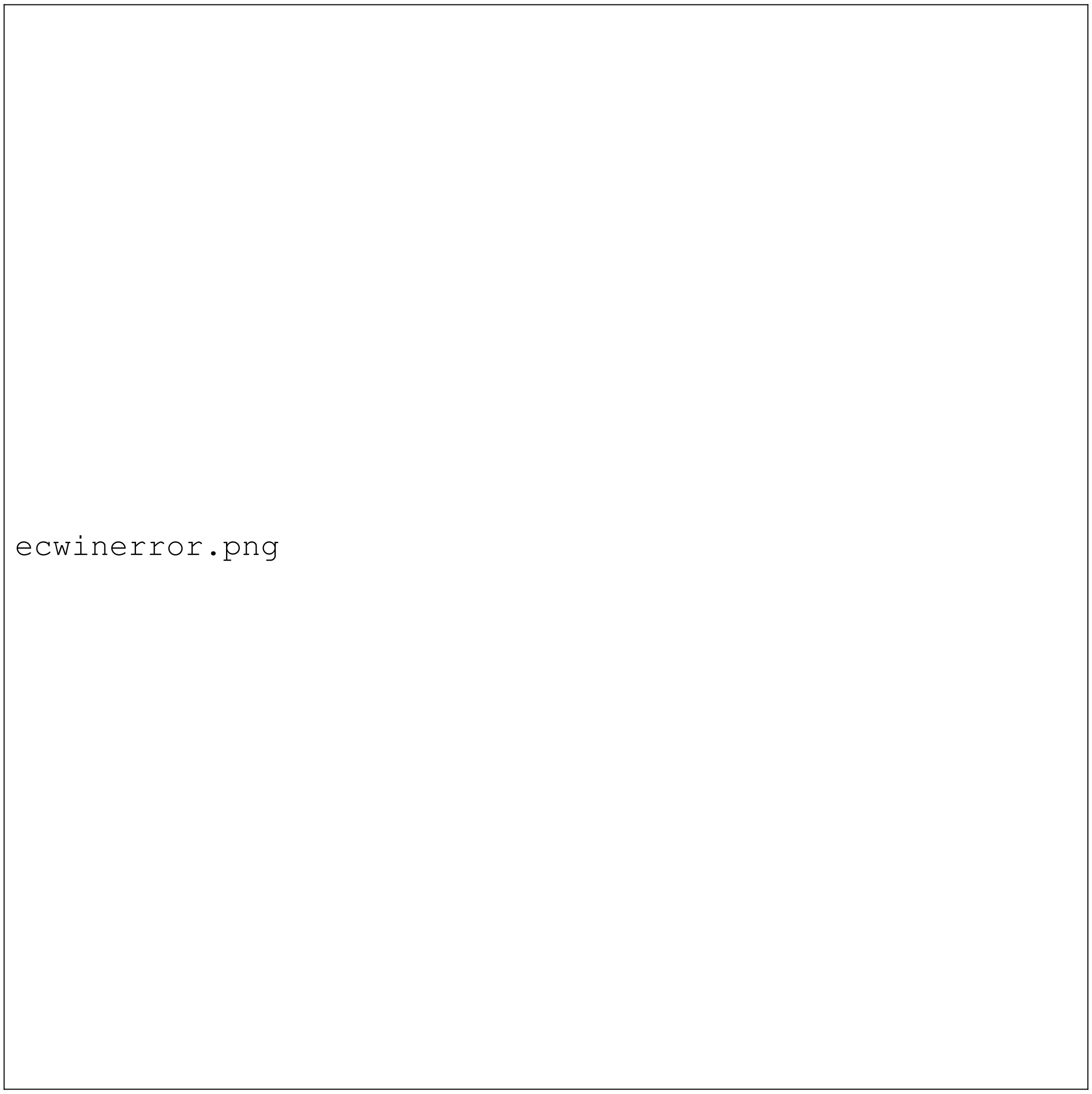

Fig. 4: ecWIN Data Export Error

Thus, the choice of gathering the data from the data editor is chosen, and the method of gathering this data is further discussed in section XIII.

XIV. Data Conversion

XV. Data Manipulation

XVI. Accessibility

XVII. Insights

[6]

References

[1] Integrate Solve Together (IST). [Online]. Available: https://ist.co.za

[2] A. Hunt and D. Thomas, *The Pragmatic Programmer: From Journeyman to Master*. Pearson Education, 1999. [Online]. Available: https://books.google.co.za/books?id=5wBQEp6ruIAC

[3] Nanair. [Online]. Available: https://cran.r-project.org/web/packages/naniar/vignettes/naniar-visualisation.html

[4] N. Yau, Flowing Data. [Online]. Available: https://flowingdata.com/2018/01/30/visualizing-incomplete-and-missing-data/

[5] Y. Bugayenko, yegor256, May 2018. [Online]. Available: https://www.yegor256.com/2018/05/08/open-source-attributes.html

[6] IST. [Online]. Available: https://www.ecwin.co.za/ecWIN/wits/