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

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 University of the Witwatersrand (Wits) campuses. The multiple energy meters installed across the properties have been gathering data for a number of years. A web server is to be constructed that is capable of autonomously and repeatedly drawing 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. This report details the project management aspect of the project to be undertaken.

I. Introduction

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

The operation of Wits requires large amounts of energy for various uses. The major resources that are used and can be monitored are- electricity, water, natural gas, petrol, and diesel usage.

This project, firstly makes use of the electricity meters installed throughout the many buildings on the properties. These meters have been providing data for varying amounts of time. Data outages occur occasionally (this will 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. It will make use of this data to visualize the energy consumption/generation across Wits. The web server will be instrumental in allowing a user to visualize this energy in a multitude of ways.

II. Project Specifications

A. The Data

As discussed above, there are a number of energy meters placed throughout the Wits properties. There are over 300 data loggers that are connected to the current web portal used by the university. The web service and data system in place, run by IST [1], is based off-campus. All of the data retrieval (from the data loggers) is done through the Wits network and in some cases through a mobile data connection. This element of the project is discussed further down in Sections A.

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 installed on top of a number of buildings on the main campus, and the weather at this campus. 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, these are the chosen data sets that will be used for the core of the project. If time permits, information regarding the location of individuals throughout the university will be used. This is provided with the use of the Integrated Campus Management (ICAM) system which is used for the access control for the university [2]. The help of individuals from Wits' Business Intelligence Systems (BIS) this information integration will be made possible. 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 Section VI 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 internet access will be able to use the system anywhere. This section is discussed in further detail in Section VI-A.

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 and visualize the data in multiple ways.

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 left for Section A 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 it 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 an example of this is Dygraphs. This requires the clients' browser to interact and process information and data[3].

The front-end is important for any web application, as this is the interface which allows user interaction. The front-end makes use of HTML, CSS, and JavaScript to generate the standard format of the user interface (UI). This configuration is 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.

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 lighting systems 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 which the project uses, this is such that minimal functionality can be gained out of the system as soon as 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 [4], 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 *tracer rounds* in this project mean a system that has simplistic functionality in each of the components. The nature of this project is such that the requirements from the *client* can be relatively vague, and this type of project is unique 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 are 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.

The following figure, Fig. 1 illustrates a first estimate on how the project timeline will be structured. The gantt chart is set out in working days.

Where:

- A - Data Retrieval
- B - Data Storage and Manipulation
- C - Back-End Set-up
- 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

There are a number of tasks which run throughout the entire project (documentation and system testing), this is done as it helps the development process 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 component. This testing helps the interfacing of the different components run smoothly and is discussed further in Section VI-C.

Finally, there is a task called *Iterative Methodology*, this is where the *tracer bullets* methodology is used. 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

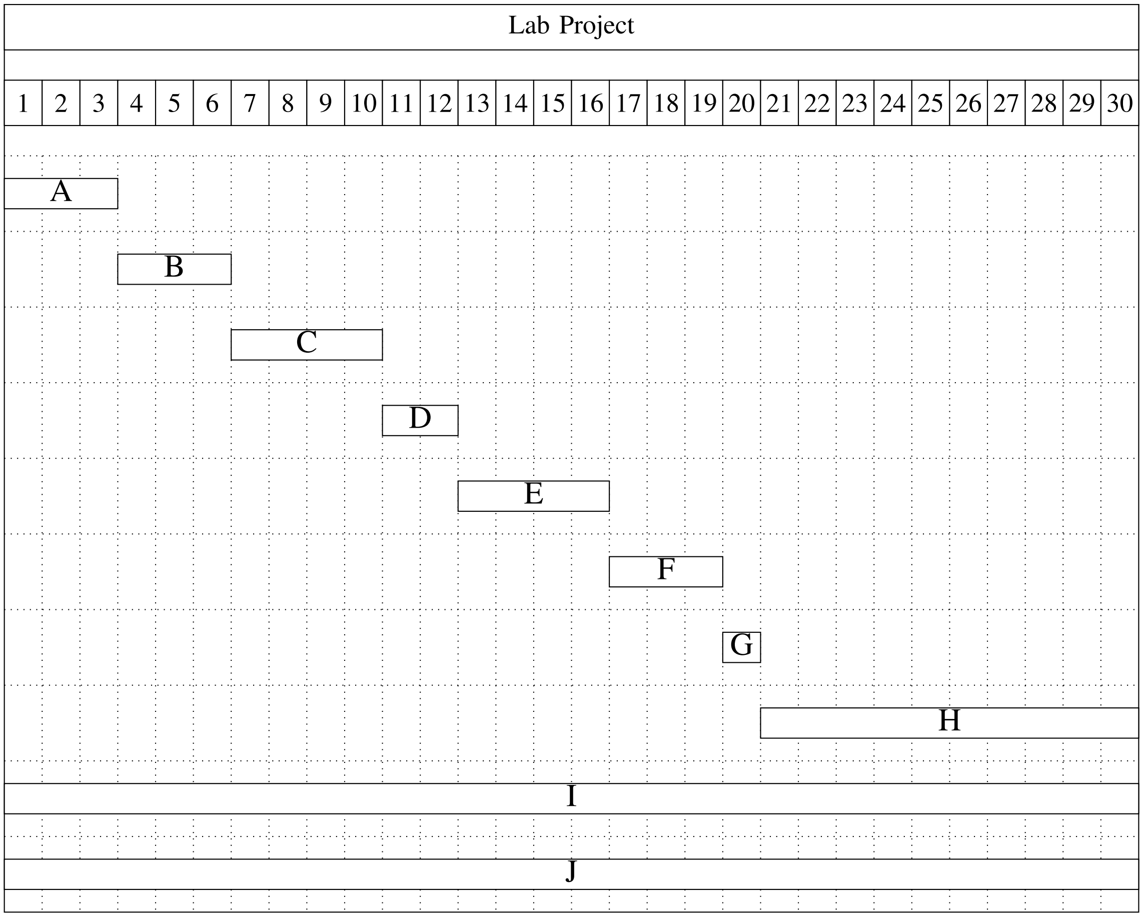


Fig. 1: Estimate Gantt Chart of Project

the data sets and the system as a whole is functioning, it becomes a matter of working with the data to provide multiple visualizations.

It should be noted that the tasks described in this section have been planned such that they all follow a consecutive order, however, as discussed further down in the report, there are components which can have clearly defined inputs and outputs, this allows for earlier development of components. This means that in some cases, modules can be developed in parallel if required or becomes more productive.

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, done such that the project can simulate a *project* in industry. 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 below:

- Retrieval and Storage of Entire Database
- Periodic Retrieval and Storage of Database
- Back-End Set-up
- 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. The input from these stakeholders can guide the later stages of the project as it will be their ideas/questions which determine what the major visualizations and information should be.This input allows for these later milestones and deliverables to be more defined.

V. Risks

Risks appear in all projects, 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 be dealt with on a case by case basis. A list of possible risks are discussed in the subsections to follow.

A. Data Outages

Data loggers can be damaged, data corruption occurs, or data is lost. This can happen for a number of reasons, and results in outages 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 are 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 [5], [6].

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

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 the 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 when interacting with the website. This poses a risk for gathering data if the website is changed, resulting in the scripts no longer functioning as they required. 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. 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.

The solar panels provide a non-negligible amount of energy to the campus, and this system provides users with energy generation data. The web scraping tools will also be used 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

TABLE I: Risks

Risk	Risk Level Likelihood (1-10)	Risk Impact (1-10)	Comments
IST System Down	4	8	No access to system means no data access
IST System Crash	2	9	Loss of data implies loss of historic data
IST System Change	6	5	Lose time because it means editing of web scraping system
Project Data Loss (damaged computer)	3	1	The codebase and documentation is stored on the github repository
Team's inability	2	7	If required, assistance is available
Intellectual Property/Plagiarism	2	9	
Internet Access Loss	5	2	Wits internet can be intermittent which reduces productivity
Lack of Weather Data	3	5	Decreased ability to visualize impact of weather due to reduced sensor fusion
Budget Risk	1	3	Currently there is no forseen expenses
Miscommunication	4	6	Time wasted
Changes in Tools (leading to malfunctioning system)	3	5	Loss of time in getting the system working again
Missing Data	8	5	Inventive ways will need to be used to manage this quirk of the data

continued use of the system. During the research of the web scraping tools in 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.

C. *Intellectual Property and Ethics*

The university is a public research institution, which means that the information generated is available to the public. The information gathered from this system has not been released to the public before this point. As 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.

D. *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 are considered for use. 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 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 required licenses. Some simple guidelines are used for choosing tools throughout the project; GPL, MIT, or similar take preference [7].

VI. System Architecture

This section discusses the general structure of the components and how they interact. 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.

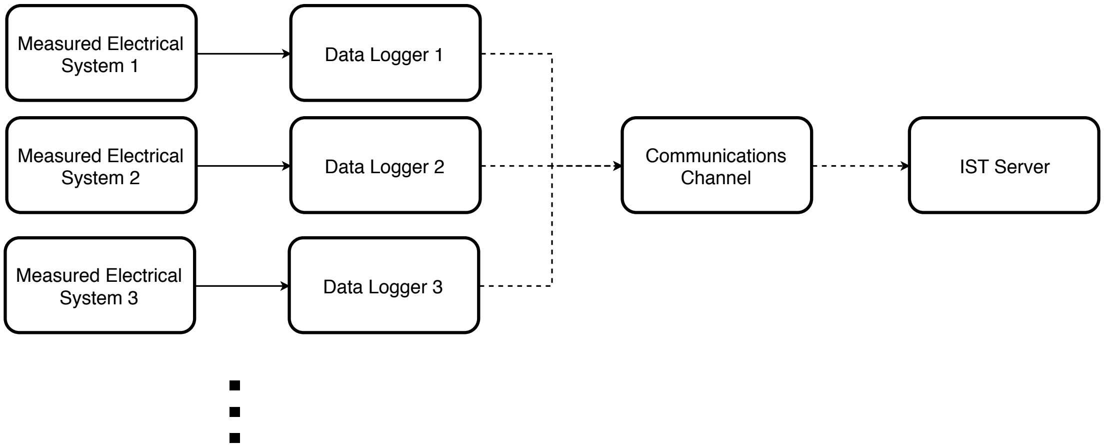

Measured Electrical System 1
Measured Electrical System 2
Measured Electrical System 3
Data Logger 1
Data Logger 2
Data Logger 3
Communications Channel
IST Server

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

In the figure, the devices measured can be from multiple different sources. They 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 sensors so the system can be represented correctly. The line connecting data loggers to their relevant systems are direct connections via the sensors. The data loggers are then connected to the communications channel via differing 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 which the data loggers are connected to the system are not of importance to this project and are not investigated.

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
  - SlashDB
- Front-End
  - HTML
  - CSS
  - JavaScript
  - D3.js
  - Dygraphs
  - Google Maps

A. *The Back-End*

The back-end will be run on local machines so the system can be hosted locally while development is under way. The local server is run by making use of Python's Flask package, working hand in hand with Node.js to manage the different languages and tasks required.

Python will handle the majority of the calculations and data manipulation for the system. There are multiple Python packages which are used, and 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 used 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 this system where no information is found within the HTML page.

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 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 work with a number of database types [8]. The database that is used should be chosen based on a number of factors: performance, size, ease of use, visualization tool requirements, etc [9]. SlashDB is seen to be a commonly used database with D3.js, however, the implementation of databases is not limited to this. 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 one 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. *Dependencies and Interfaces*

The aspect of interfaces was discussed above, however, this forms an integral part of the project as there are many components which are required to work together for the system to function. Every component should be designed such that its inputs and outputs are clearly defined such that it can work seamlessly with the surrounding components. An example of this is that of the web scraping tool- it is required to gather csv files from a website, it expects these csv files to be of a specific format, it is then expected that this system place these files in a folder with a name that is standardised with the next component in the system. This allows for efficient modularity within the system and if there are cases where specific components need to be changed, it does not affect other components within the system. The dependencies and interface requirements of all of these components is specified in the documentation and comments of the code of the component. This allows for easy modification down the line.

The other important consideration for the system is dependencies of the tools required for the system to function. There will be a multitude of tools used throughout all with different requirements and settings so that they can be used effectively. These dependencies will also form part of the documentation of the system for future use.

VII. Resources

The nature of this project is such that open source and freely available tools will be used to create the system. This implies that a budget will not be required for the project. There is a set budget which is allocated for each project and it is unlikely that any of this budget will be used for the project. The tools which potentially require licenses (eg. Matlab) will be provided by the school, however, it is unlikely that these tools will be used at this point as they do not currently have a use in the project.

The project is heavily based on the use of software tools and machines on which to develop the system. The students will make use of their personal machines to do this. There may be cases where a more powerful machine may need to be used, in this case the machines in the D-lab will be utilised.

The major cost in this system has already been covered, this is the cost of the data loggers and the servers provided by IST. This has been put in place over the last few years by the university.

VIII. Applications and Tools

This section details the auxiliary tools used during the development of the system. The first of these tools, the operating system, is Windows 10, chosen as this operating system is already installed on the students machines before the project started. Ideally, the system should eventually be placed onto a server, which will likely be running a distribution of Linux, so considerations should be made for the structure of the system such that porting the system will be a simple affair. The tools that are discussed in this report are useful in that they can easily be used on both operating systems with minimal change to the overarching system. During development, however, the design should be such that the system can run on as many different machines while requiring little to no editing of the code base.

The system will require a large amount of software writing, as such the use of Visual Studio Code will be used for the majority of the editing. This is chosen as it is open source, and has a large third party extensions database which makes the development process easier.

A version control software is used throughout the development of the project. Since the bidding stage, a git repository has been used that the two students can work with. The git repository is currently hosted on gitHub using a private repository.

This system allows for convenient and safe work on the project to take place. The repository allows the two users to work remotely while developing different aspects of the project at the same time.

IX. Development Approach

The programming methodology discussed in *The Pragmatic Programmer* is made use of throughout the project, this is: work iteratively, make small changes to the system and re-evaluate.

In any software project, documentation plays a role. It needs to be present in some form. There are many different ways of documenting a project. One commonly used way is to create and manage a Wiki; this means that all of the documentation for the project can be found in one place [10]. A second popular method is to make use of markdown files throughout the system. This forms part of the repository and can be found alongside the code.

Make sure that as code is written, it is commented and documentation is made for that section. Documentation is highly important for the success of the project because it should be able to be used by future teams. Up to this point, markdown files have been used to document meetings, notes, ideas, etc. and will continue to be used for the time being. The use of a Wiki, or other similar documentation structure may be considered at a later stage during the project. It might be worth investigating the use of GitHub Wiki which will work hand in hand with the currently used GitHub repository [11]. These markdown files form an instruction manual for the different parts of the system.

Testing of the system is done throughout the development process, this is such that each and every component can be automatically verified that it does what it should do. The interface of each component in the project will be tested using this testing procedure.

There are multiple types of testing which may become useful for the system to implement: unit testing, integration testing, component interface testing, system testing, and operational acceptance testing [12]. The package, Selenium, becomes highly relevant when testing out the front-end as it emulates the process that would happen for when a user is interacting with the system. This tool has been used extensively by Dropbox for their product line, indicating a reliability in this type of testing for websites [13].

References

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

[2] G. Watermever, University of the Witwatersrand. [Online]. Available: http://www.icam.wits.ac.za

[3] "Dygraphs." [Online]. Available: http://dygraphs.com

[4] 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

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

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

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

[8] "dygraphs data format," dygraphs, 2018. [Online]. Available: http://dygraphs.com/data.html

[9] E. Bricker, SlashDB, February 2018. [Online]. Available: https://www.slashdb.com/2018/02/15/howto-d3js/

[10] D. Mytton, "How do you document your ops infrastructure?" April 2016. [Online]. Available: https://blog.serverdensity.com/how-do-you-document-your-ops-infrastructure/

[11] "About github wikis," GitHub, 2018. [Online]. Available: https://help.github.com/articles/about-github-wikis/

[12] P. Bourque, F. Robert, J. M. Lavoie, A. Lee, S. Trudel, and T. C. Lethbridge, "Guide to the software engineering body of knowledge (swebok) and the software engineering education knowledge (seek) - a preliminary mapping," in *10th International Workshop on Software Technology and Engineering Practice*, Oct 2002, pp. 8–23.

[13] R. Tene, Dropbox, May 2018. [Online]. Available: https://blogs.dropbox.com/tech/2018/05/how-were-winning-the-battle-against-flaky-tests/

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

Appendix

Data Gathering

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 [14].

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.

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).

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 A.

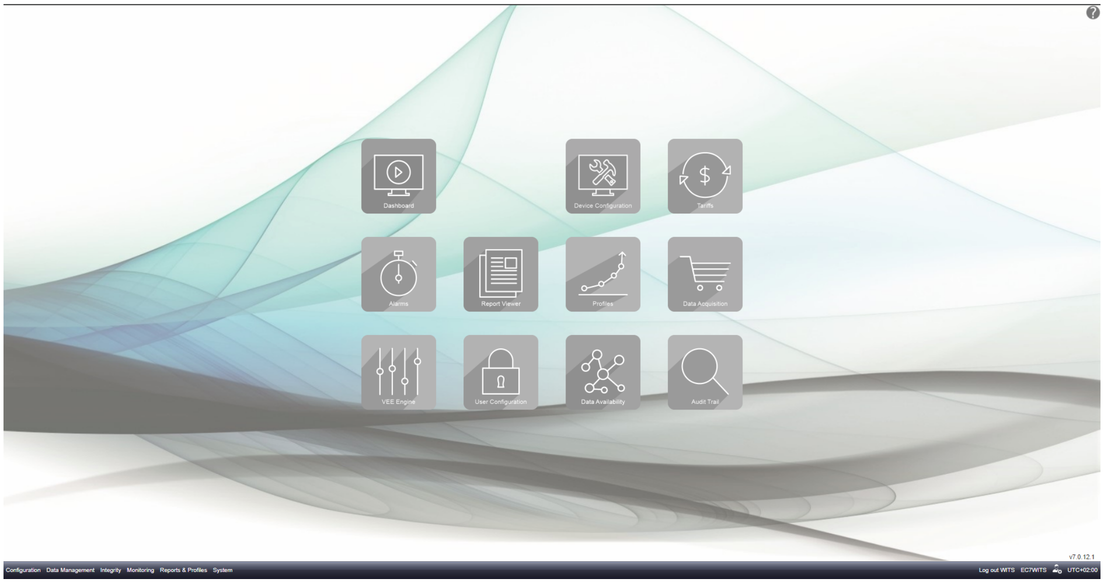
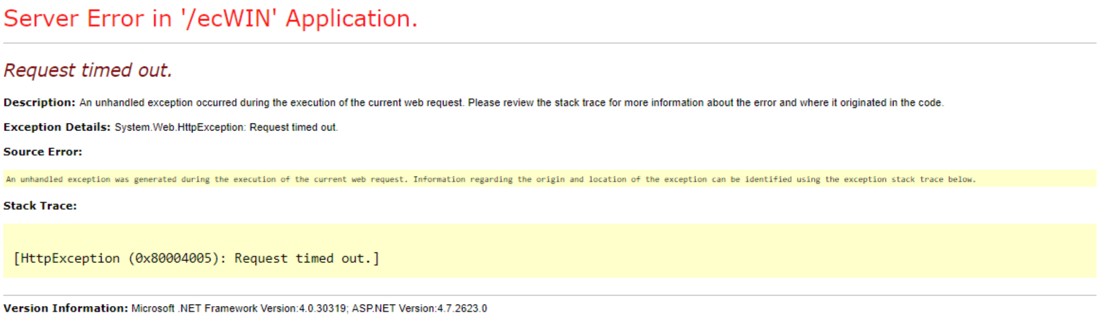


Fig. 3: ecWIN Web Portal



Server Error in '/ecWIN' Application.

Request timed out.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Web.HttpException: Request timed out.

Source Error:

```
An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.
```

Stack Trace:

```
[HttpException (0x80004005): Request timed out.]
```

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.7.2623.0

Fig. 4: ecWIN Data Export Error

As alluded to earlier in the report, the system will not always draw data going back to the installation of each datalogger. Once the system has stored the entire dataset, it only needs to download the latest entries in order to keep updated, this reduces the load on the server and the time taken to update to the latest entries. The system should also have the ability to verify the data that it has within its database, as there can be changes to the IST database once the data has already been extracted.