

Institiúid Teicneolaíochta Cheatharlach



At the Heart of South Leinster

Utility Watch

Functional Specification

Student Name: Ray Shannon

Student ID: C00079959

Student e-mail: c00079959@itcarlow.ie

Supervisor: Paul Barry

BSc(Honours) in Software Development (4th Year)

Institute Of Technology Carlow,

Kilkenny Road,

Carlow.

Date: 17/04/2015

Department of Computing and Networking,

Institute Of Technology Carlow

Abstract

The purpose of this document is to describe the objectives and specifications of the project. This document will contain, in detail, the purpose and goals of the project, the project's functionality flow, the scope of the project and detail the interfaces and procedures used to achieve this functionality.

This document will also include the project plan, which will cover the main problem statement, reference to the expected deliverables and a predicted timeline of the major project milestones.

Table of Contents

1.	Problem Statement.....	5
2.	Project Description.....	6
2.1.	Introduction	6
2.2.	System Description	6
2.2.1.	Backend components.....	6
2.2.2.	Frontend component	6
2.2.2.1.	Graphical User Interface	7
2.2.2.2.	System Overview Diagram	7
3.	Functional Requirements.....	8
3.1.	Functionalities.....	8
3.1.1.	Monitor Usage	8
3.1.2.	Display Data	8
3.1.3.	Calculate Cost.....	8
3.1.4.	Control Heating System	8
3.1.5.	Control Lighting System	8
3.1.6.	Account login	8
3.2.	Use Cases	9
3.2.1.	Use Case Diagram	9
3.2.2.	Brief Use Case 1 – Login.....	10
3.2.3.	Brief Use Case 2 – View Usage	10
3.2.4.	Brief Use Case 3 – Control Heat	11
3.2.5.	Brief Use Case 4 – Control Light.....	11
3.2.6.	Brief Use Case 5 – Logout	12
4.	Project Plan	13
4.1.	Deliverables.....	13
4.1.1.	1 st Sprint	13
4.1.2.	2 nd Sprint	14

4.1.3. 3rd Sprint..... 14

4.2. Timeline..... 15

5. Conclusion..... 16

1. Problem Statement

There is a proverb which states, “If it is not measured, it is not managed”. Ok, utilities are measured, but only in the form of a meter, which is either buried underground in your drive or stuck in a box on the external wall of your property, displaying spinning numbered dials. These dials inform customers of very little, if anything at all.

Most consumers of electricity are unaware that in their homes at the small hours of the night, they are consuming power unnecessarily. Electrical appliances in standby mode, kitchen appliances unnecessarily active or lighting left on, are increasing our utility bills.

This application will provide a visual and tabulated representation of the energy usage and cost while identifying which periods of the day, week or month are heaviest on energy consumption. These measures and indicators will highlight to the user the trend of their property’s energy usage and enable them to make informative and decisive decisions helping them to reduce their utility bill. This will not only impact their pocket, but will have a positive effect on our environment also.

The heating control functionality can also help with energy efficiency as it gives instant and remote access to the consumer, who can, at the touch of a button, override previous settings which may be no longer applicable.

The lighting control functionality may assist with property security, enabling property owners to turn on internal or external lighting if they are absent from their property.

I feel there is a need and demand for residential and commercial property owners to have instant, remote access to their data, and based on that information, be able to control devices or systems which will improve energy consumption or improve property security. Utility Watch will provide such requirements.

2. Project Description

2.1.Introduction

Utility Watch will be a standalone unit which can be installed on any property containing an electrical energy supply. The primary function of the system will be to monitor and display the energy consumed by the property on which the unit is installed. The product will also provide remote control functionality for a heating and lighting system in operation on the property.

The client will be able to access historical information of their energy usage and control heating and lighting systems from any web browser on any platform or device. This service will be provided through a secure login process on the applications homepage.

2.2.System Description

The system comprises of both back-end and front-end components.

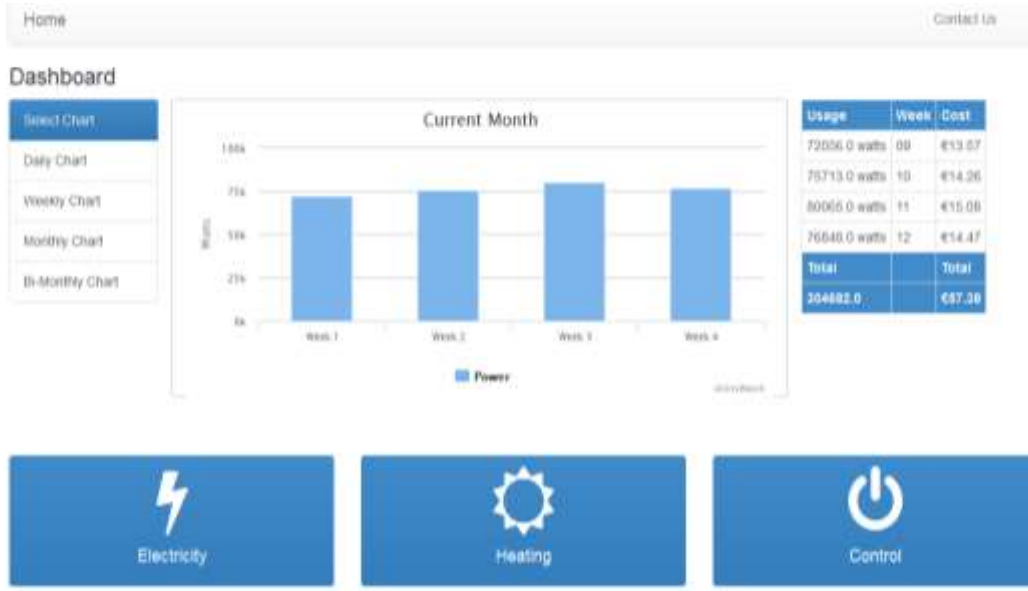
2.2.1. Backend components

- Intel Galileo Board – The Galileo is a microcontroller board based on the Intel Quark X1000 Application Processor and is the major component of the project. The Galileo will be used as a server to host a Flask application. It will also store the databases required to house the data generated by the application. The background processes required to ensure the application runs will be executed by the Galileo board through the Python language.
- Energy Gateway and RF readers – A 3rd party hardware is required to interrogate the electricity usage. An amp reader, clamped around the property mains supply, will send signals back to a gateway informing it of the current watt reading. The gateway will also listen for requests on the network to execute various instructions providing remote control functionalities. These instructions will be generated by the Python program, through the User Interface.

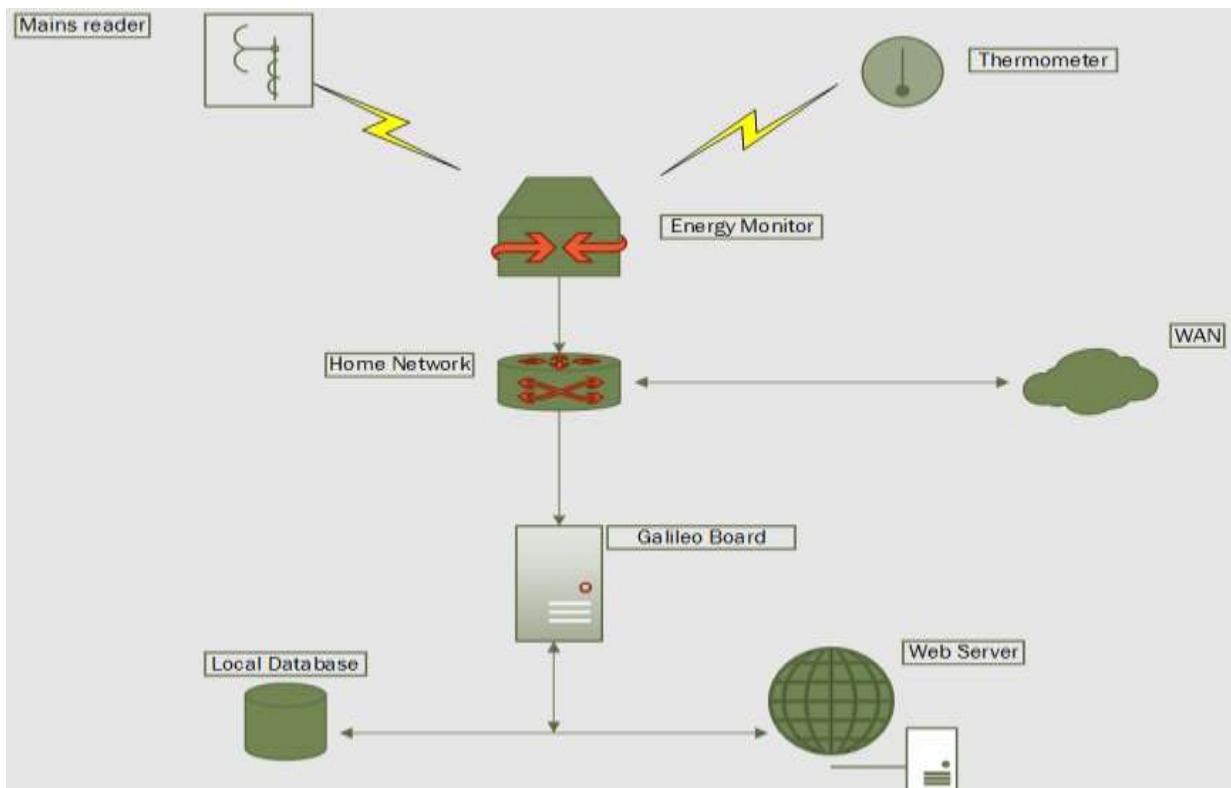
2.2.2. Frontend component

- The frontend component consists of a graphical user interface, (GUI), in the form of a web application. The GUI facilitates interaction between the user and the system. It will provide a means for the user to view historical data, monitor and control room temperature and control electric light circuits. A typical GUI is shown below.

2.2.2.1. Graphical User Interface



2.2.2.2. System Overview Diagram



3. Functional Requirements

3.1. Functionalities

The following is a description of each of the functionalities the Utility Watch system will provide.

3.1.1. Monitor Usage

One of the main functionalities of the project is to monitor the electricity consumption of the premises where the system is installed. This data will be stored locally and used to populate graphs and tables for the end user.

3.1.2. Display Data

Another important functionality involves displaying the gathered data in graph and table form. The graphs will assist the consumer in visually identifying the peaks and troughs of energy usage for the selected period of time, while an accompanying table will give the data in numerical values.

3.1.3. Calculate Cost

Using a unit rate and standing charge applicable to the property, the cost is calculated by analysing the units used. This data will also be displayed in the accompanying tables and gives the customer feedback in monetary terms.

3.1.4. Control Heating System

The Utility Watch system will provide a live reading of the room temperature and the option to turn on or off the heating system remotely. The live reading will show the expected increase in temperature while the heating system is switched on.

3.1.5. Control Lighting System

The Utility Watch system will provide remote control functionality of a lighting system on the property. A live reading of the meter will display an increase in units used when lights are switched on.

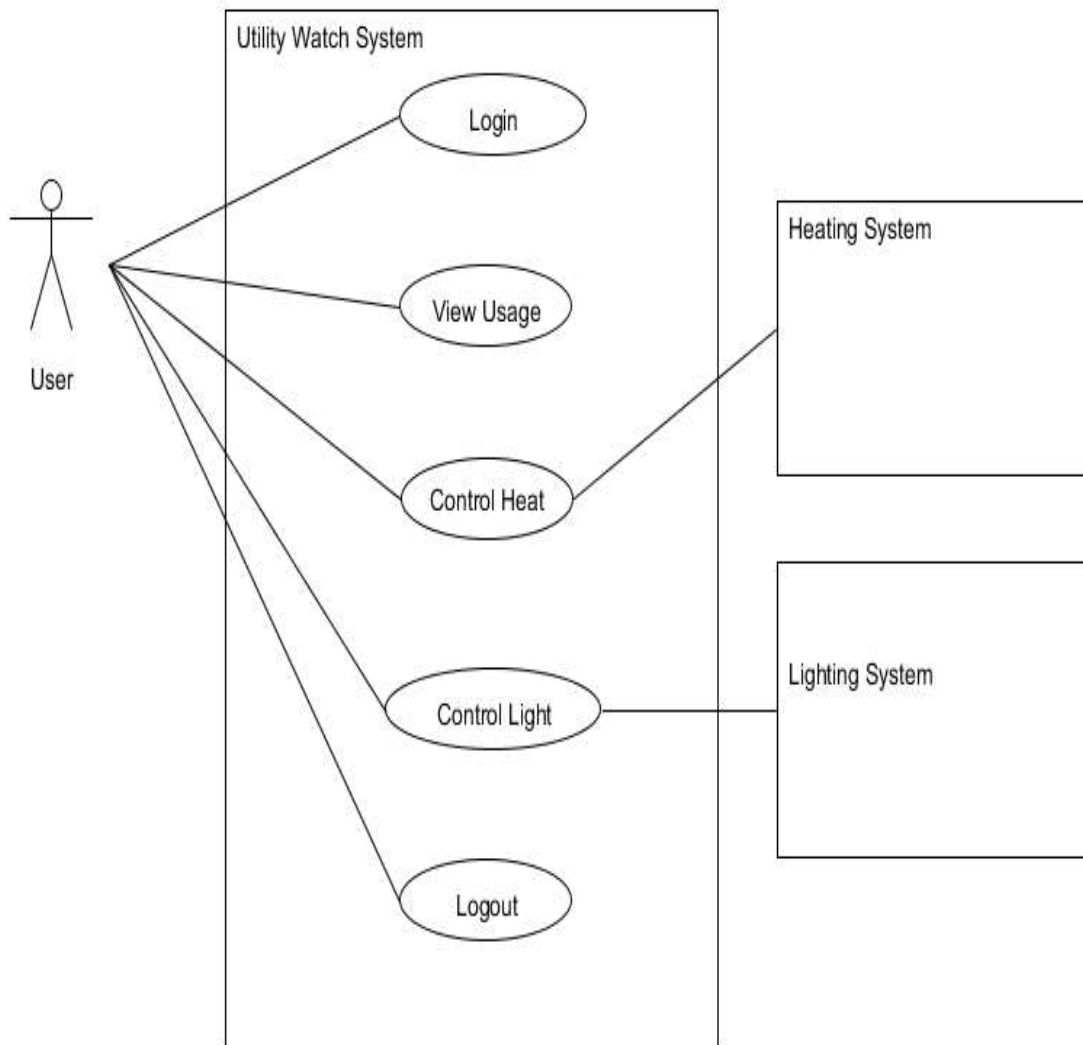
3.1.6. Account login

The property owner will be able to login to their application from any web browser and have total control of all the above mentioned functionalities.

3.2. Use Cases

The Use Cases set out below are an initial draft, while some of the brief Use Cases are not fully defined; this has been done purposefully to allow for changes which will occur during further research of the project. This will ensure the software will be thorough in its final outcome.

3.2.1. Use Case Diagram



3.2.2. Brief Use Case 1 – Login

<i>ID</i>	Login
<i>Actors</i>	User, System
<i>Preconditions</i>	The web application starts up displaying the login screen, asking the user for valid login details.
<i>Description</i>	The use case begins when the user opens the home page of the Utility Watch application. The user enters a valid username and password and presses the login button.
<i>Post condition</i>	Upon pressing the enter button, the dashboard is displayed, offering the various options available.

3.2.3. Brief Use Case 2 – View Usage

<i>ID</i>	View Usage
<i>Actors</i>	User, System
<i>Preconditions</i>	The user has selected 'Electricity' from the dashboard.
<i>Description</i>	The application displays the electricity usage for the current day. The option of selecting a week, month or bi-month view is also available. The corresponding table of data is also displayed
<i>Post condition</i>	Depending on action of user, other views will be displayed.

3.2.4. Brief Use Case 3 – Control Heat

<i>ID</i>	Control Heating
<i>Actors</i>	User, System
<i>Preconditions</i>	The user has selected the 'Heating' option from the dashboard
<i>Description</i>	The use begins with the user shown the current room temperature and the option of turning on or off the heating system by means of a button
<i>Post condition</i>	When the user turns on the heating, the live temperature gauge will increase its value to correspond with room temperature

3.2.5. Brief Use Case 4 – Control Light

<i>ID</i>	Control Light
<i>Actors</i>	User, System
<i>Preconditions</i>	The user has selected the 'Control' option from the dashboard
<i>Description</i>	The use case begins when the user wishes to control the lighting system. The user, using the switch, can turn on or off the lights. The live gauge will display the increase or decrease in power usage to reflect the new condition.
<i>Post condition</i>	The lights will be turned on or off depending on the user action.

3.2.6. Brief Use Case 5 – Logout

<i>ID</i>	Logout
<i>Actors</i>	User, System
<i>Preconditions</i>	The user is finished and wished to logout.
<i>Description</i>	The use case begins when the user selects the logout option from any screen.
<i>Post condition</i>	Upon selection of 'logout', the user is returned to the login screen.

4. Project Plan

The project plan will discuss the project deliverables and the timeframe in which they are expected to be completed. As the project is divided into 3 sprints, the project plan will aim to correspond to this format.

4.1. Deliverables

4.1.1. 1st Sprint

- **Galileo Board**

The Galileo board will require an initial setup to allow external communication. This will be done by executing an Arduino sketch through the mini usb serial port on the board from the Arduino IDE on a local pc. The setup will also perform initial testing of the board using pre –built sketches from the Arduino IDE Library.

Once communication is established and the board is functioning correctly, an Ethernet sketch will be executed enabling the board to be installed on the local network. This will give the board an IP configuration complying with the local network settings. All future communication with the board can be done on the local area network.

The board requires a light weight linux operating system to be installed enabling required packages and libraries to be downloaded. Python is available with the installation of the linux operating system (OS) which will give us access to certain database functionality at a later stage. It will also provide us with a web server to host our application on.

- **Energy Gateway**

3rd party hardware device will be required to interrogate the electricity meter and determine how many watts of energy are being used at any one time. The initial research will be to investigate the operation of this device, and determine the best solution to incorporate it into the project setup. It will also require network configuration to allow access onto the local network.

- **RF Sensors**

The radio frequency (RF) sensors are the wireless devices that will talk to the gateway and enable remote control functionality for heating and lighting systems. Again, the initial research will be to investigate the required commands to activate and control these devices and incorporate them into the projects setup.

- **Remote Access**

A critical part of the project is the accessibility from external networks, namely the Wide Area Network (WAN). The Galileo must be contactable from outside the LAN by the user. This will be made possible with the use of dynamic domain name server (DDNS), which is a service provided by the LAN router once an account is setup, and creating firewall rules to allow access on certain ports to the application.

4.1.2. 2nd Sprint

- **Web Application**

A web app, sitting on top of a Flask application, will be the main running program. This will provide the user the interface to access and control the application as described in the functionality requirements.

- **Database**

As mentioned previously, there will be a lot of information generated. Approximately 240 new lines of data will be written to the main database per hour. Each subsequent database then uses that data to generate its own data. A structured database will be needed to facilitate this requirement.

- **Background Services**

This project will require services to be working in the background to assist with the running of the application. Background activity will include requesting live readings from the electricity meter, updating daily, weekly and monthly tables at the appropriate times, and ensuring the web server is active 24 hours a day.

4.1.3. 3rd Sprint

- **Web Application**

Applying the finishing touches to the GUI and ensuring ease of use for the user. Monitoring the application as the data builds up, ensuring the graphs are behaving as programmed and tables are populating correctly.

- **Documentation**

At this stage the documentation will be the biggest workload, being important to show the progress of the project. It is an opportunity to display all the unseen effort like research, testing and unused material.

4.2. Timeline

- Diagram goes on this page

5. Conclusion

This specification document has highlighted the main functionalities that will assist with determining the project. It states the proposed features of the project and the necessary processes required to produce those features. It describes, at a high level, the functionalities which will be provided by the application without determining how those functionalities will be achieved.