

First Prototype Design and Execution of a Smart Home Power Management System

2016/04/11

Ari Croock - 718005
Kanaka Babshet - 678851
Alice Yang - 597609
Daniel Weinberg - 547937

ELEN4009 - Software Engineering

School of Electrical & Information Engineering, University of the Witwatersrand, Johannesburg

Abstract: This paper details the design and implementation of the first prototype of a smart home power management system. The final project submission is composed of an application which can add a household device, and control its state, as well as a compilation of all the documentation written in the process of designing this application. A critical analysis of the issues experienced during the design, implementation, and testing, has been provided. All constraints and specifications are successfully met, and additional improvements (as stated herein) can still be made in the future.

Key words: Design, Smart Home, Power Management, Internet of Things

1. INTRODUCTION

A smart home power management system allows users to control and supervise various electrical devices within the household in order to manage the overall power consumption. The system also automatically manages various devices without human interaction depending on the settings enabled in advance by the system user. This interaction and status display is done via a webpage accessible to all predefined household users.

The aim of this project is to design the complete aforementioned system and implement some aspect of the system which is defined as the first prototype. The prototype implemented is made up of a scrollable webpage, with a page header and tabs, the addition and removal of a device, with corresponding specifications, viewing the devices connected, and toggling between the states of devices.

Section 2 details the software requirement specification, and is followed by section 3 which contains an extensive design document split into the relevant front and back end sections. Section 4 introduces the sprint planning and illustrates the work breakdown over the time designated for this project, while section 5 provides a sprint retrospective. Section 6...

2. SOFTWARE REQUIREMENT SPECIFICATION

2.1 Purpose

This section details the system requirements specification (SRS) for the Smart Home Power Management System. The system design documentation will be developed from this information.

With the rapidly growing interest in new Internet

of Things (IoT) technologies, networks consisting of these devices will become increasingly difficult to manage and control. Additionally, power consumption and monitoring will become a greater concern, especially in emerging markets such as South Africa.

This project aims to provide a flexible software system which is able to remotely control and monitor IoT devices, as well as perform detailed power consumption diagnostics.

2.2 Project Scope

The project is a system that will allow for remote control, monitoring and automation of IoT devices on a Local Area Network (LAN). A client-server architecture will be used since this allows a back-end server to continuously manage devices while allowing for a client to connect on-demand. The front-end will initially be implemented as a web page for simplicity and compatibility with many existing devices (such as cell-phones and personal computers).

The back-end provides functionality to control and monitor devices, and to log device power consumption. Additionally, the back-end will contain the web server used for interfacing with users.

The front-end will consist of a web-based user interface which will provide secure access to a device dashboard. The dashboard will provide remote control and configuration of devices, as well as access to power consumption data. Addition and removal of IoT devices will also be performed through the dashboard.

2.3 Project Overview

Figure 1 shows a data flow diagram of the system. Data collected from IoT devices is stored in a database

and used to control other devices as defined by the user. A web interface allows the user to configure devices and to view the stored data.

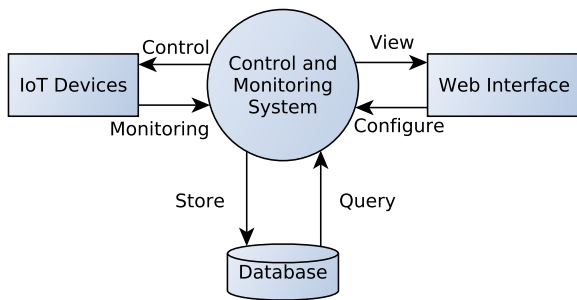


Figure 1 : Data flow diagram (DFD) of the system.
All data flows are bidirectional

2.3.1 Assumptions:

- All of the IoT devices will be connected to the same LAN
- Only a small number of simultaneous users and devices will be connected at once

2.3.2 Constraints:

- The user interface must be accessible through a web browser
- All devices in the system require wireless connectivity

2.3.3 User Classes and Characteristics:

- Basic users who can configure basic devices and view power usage statistics
- Advanced users who can configure custom devices and export power usage statistics for further processing

2.3.4 Operating Environment: The following technologies will be used to implement the system:

- Django Web Framework (Python 3.5.1)
- PostgreSQL Database Management System (DBMS)
- Docker 1.8.2

Assuming that few users and devices will be connected to the system simultaneously (such as in a typical household), the system requirements are fairly relaxed:

- any modern Microsoft Windows or GNU/Linux system that supports the technologies used
- a connection to the LAN, such as Ethernet or WiFi

- low-cost hardware such as a Raspberry Pi

However, depending on the system configuration, an external storage solution may be needed for storing a large amount of power usage data.

2.3.5 Software Development Life-cycle: The system will be developed using an Agile software development methodology. In particular, the popular SCRUM framework will be used so that the system's requirements and design details can change quickly if need be.

2.4 System Features

The smart home power management system is made up of various features which enhance the usability of the product, along with some additional features which simplify all related functionality.

2.4.1 System Feature 1 - Remote On/Off Switching:

Switch any of your selected appliances on or off from any personal device connected to the home network.

Description: The user can control the power status of various appliances in the household. The appliances are connected independently, for which reason, multiple appliances can be controlled through the same interface, simultaneously.

Stimulus/Response Sequences: The user has to select the "Power Remote" tab on the home page to display the on and off options corresponding to each connection.

2.4.2 System Feature 2 - User Configurable Triggers: Personalise various situations which will trigger an action.

Trigger 1 - Motion Detection: Set appliances to switch on or off depending on surrounding motion.

Description: The user can control whether specific appliances must be configured with motion detection or not. For example, if motion detection is set for a certain set of lights, they will detect movement in the surrounding area and switch on, or switch off when there is no motion detected for a set period of time. If the user does not want an appliance or set of lights to be triggered by motion, the respective switch is "off" on the webpage.

Stimulus/Response Sequences: The user has to select the "Triggers" tab from the home page, and thereafter select the "Motion Detection" tab at the top left of the newly loaded page to display the corresponding settings. Figure 3 in Section 4 below illustrates this page.

Trigger 2 - Temperature Detection: Set appliance statuses to vary depending on the surrounding temperature.

Description: The user can control whether specific appliances must be configured with heat triggers or

not. For example, if temperature detection is set for a specific appliance, it will detect a certain temperature in the surrounding area and switch on, switch off, or maintain state if it is too hot, too cold, or within the correct temperature range depending on the user choices. If the user does not want an appliance or set of lights to be triggered by the ambient temperature, the respective switch is "off" on the webpage.

Stimulus/Response Sequences: The user has to select the "Triggers" tab from the home page and thereafter select the "Temperature Detection" tab at the top of the newly loaded page, (second from the left), to display the corresponding settings.

Trigger 3 - Light Detection: Set appliance statuses to vary depending on the surrounding light.

Description: The user can control whether specific appliances must be configured with light triggers or not. For example, if light detection is set for a specific appliance, it will detect a certain ambient light level in the surrounding area and correspondingly vary, or maintain state depending on the user choices. If the user does not want an appliance or set of lights to be triggered by the ambient light levels, the respective switch is "off" on the webpage.

Stimulus/Response Sequences: The user has to select the "Triggers" tab from the home page and thereafter select the "Light Detection" tab at the top of the newly loaded page, (third from the left), to display the corresponding settings.

Trigger 4 - Time Scheduling: Select and adjust times for the different appliances to be on, off, or running on defined settings.

Description: The user can activate time scheduling for each device/set of lights independently connected to the smart home power management system. For example, the user can set the geysers to switch on from 6 am to 8 am and from 7 pm to 9 pm everyday. Note that the selection of time scheduling for appliances does not affect the user's ability to manually control the power via the system, or directly in person.

Stimulus/Response Sequences: The user has to select the "Triggers" tab from the home page and thereafter select the "Time Scheduling" tab at the top right of the newly loaded page to display the corresponding settings.

2.4.3 System Feature 3 - Power Consumption Monitoring: Monitor the power consumption by various appliances and sets of lights in the household.

Description: Given that this system is based on power management, the user can easily view detailed layouts of the total household power consumption data and analytics. The user will be able to view graphical statistics of the power consumption over the course of the day, as well as detailed statistics over a longer period of time (previous month, or year). This allows the user to compare the household power consumption with that of the usage if this system was not implemented.

Stimulus/Response Sequences: The webpage automatically starts on the "Home Status" tab upon loading. The user can view this page for a daily consumption summary, and can then click on the "Consumption Monitoring" tab seen on the left of the same page to view these analytics in greater detail, over a longer period of time.

2.4.4 System Feature 4 - Energy Source Management: Control and manage the different energy sources in the household such as varying the usage between electricity and solar power.

Description: This system encourages the use of alternative, renewable energy sources. If the user has catered for these facilities, the system allows additional power volume management through the energy source management configuration.

Stimulus/Response Sequences: The user has to select the "Energy Source Management" tab on the left of the homepage to be directed to this page.

2.4.5 System Feature 6 - User Configurable Alerts: Choose which events should result in an immediate alert/notification on your system.

Description: Along with all the personalised settings configured on the system, the user is able to define situations in which alerts appear on the system and possibly directly to the user's smart device.

Stimulus/Response Sequences: The user has to select the "Alerts" tab on the left of the homepage to then configure these settings.

2.4.6 System Feature 7 - User Configurable Webpage Settings Define webpage settings in order to enhance the power management system experience.

Description: The user can add or remove devices to be controlled by the smart home power management system, alter the webpage appearance, manage users on the network, and edit security settings.

Stimulus/Response Sequences: The user must click on the "Settings" link to the right of the title on the home page to be able to configure all the required settings.

2.5 External Interface Requirements

The external interface is the interface which allows the user to interact with the system through a hardware device using software. The external interfaces consists of a user interface, hardware interface, software interface and communication interface.

2.5.1 User Interface - GUI: The user interface is how the user will interact with the system. The requirements for the design of user interface must be simple and easy to navigate. A simple format of representing all the devices connected to the Smart Home

Power Management System is to display the features available for all devices on a dashboard. This can be seen in Figure 2.

Figure 3 is a simple illustration of the interface between the user the light detection triggers on the system. The layout of the tab is further divided to clearly illustrate the sub-systems which the user has access to. As seen from the figure, the switches on the triggers tab are designed in a graphical format that is simple for users to use.

Figure 4 is the settings page, in which it designed for the user to navigate through each sub-system. The *Add/Remove Device* feature is an important feature, as it is expected that the client may have to add or remove devices for future use. The settings page is set up similarly to the home page on the trigger tab, as user’s familiarity to the website is of importance.

- Home Page Dashboard

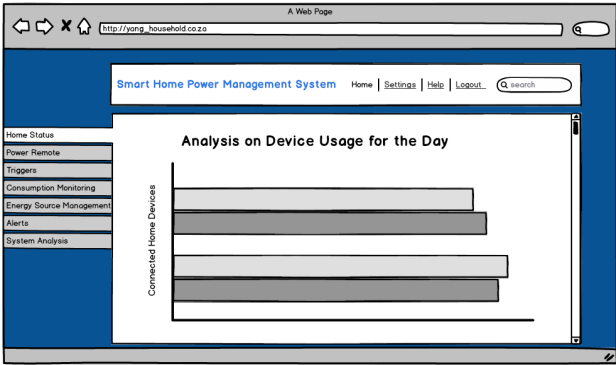


Figure 2 : Figure illustrating the home page the user will see once they have logged on to access the system.

- Trigger Tab on Dashboard

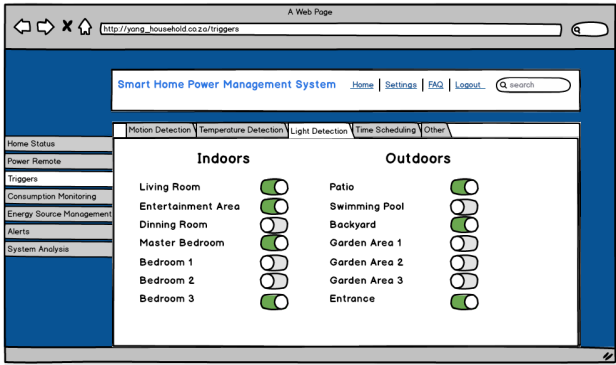


Figure 3 : Figure illustrating the triggers tab, to allow users to control switching of sensors in their home through the system.

- Settings Page

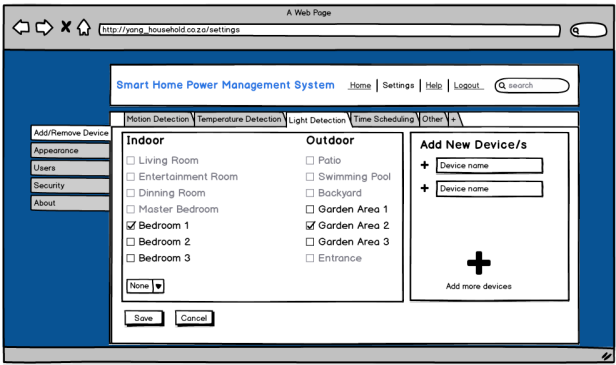


Figure 4 : Figure illustrating the settings page.

2.5.2 Hardware Interface: The user interface is implemented on a web based platform that is accessible through a browser. The hardware required to access the system can be either of the listed devices found in the list below:

- Large Desktops (1200 - pixels)
- Medium Desktops (992 - 1199 pixels)
- Small Devices (768 - 991 pixels)
- Extra Small Devices (max 767 pixels)

2.5.3 Communication Interface: The Smart Home Power Management System is a web-based application that requires a internet connection for the communication between the hardware devices, system and IoT devices. The main connection between the system and the IoT devices is through the user’s home wireless connect (WiFi). For the hardware device to communicate with the system any type of internet connection will do.

2.6 Other Non-Functional Requirements

2.6.1 Performance Requirements: The smart home power system is an application and system that is designed for efficiency. This means that the application as well as the connecting household components need to respond quickly and in real time.

The application will most likely undergo several updates and changes for the user's benefit. The system is required to update on user command. Updates will include further improvements to the application as well as fixes for issues that arise in operation.

2.6.2 Safety Requirements: The application as well as the system components need to take certain safety concerns into account.

Due to the fact that the application is essentially controlling most of a home's appliances and electricity usage, it is important that it monitors everything it is controlling to prevent problems that may arise. The application needs to ensure that any device connected in the system does not reach a dangerous level of usage. In this case, the application should either switch off the device in question or notify the user that something is not functioning correctly and needs to be addressed.

2.6.3 Security Requirements: Due to the fact that the system controls a user's home, it therefore requires security considerations to be taken into account.

The system needs to ensure that only the user has access to the application to prevent external parties gaining control of the connected components within a house. This can be done with a signup, login and authentication process. The components within the house that the application connects with, need to be protected from external parties and therefore they must be explicitly authenticated.

2.6.4 Software Quality Attributes: The application is web based and needs to be user friendly. The functioning of the application needs to be simple so that no additional documentation or prior knowledge or experience is required.

3. DESIGN DOCUMENT

The purpose of this section is to provide details regarding the software development of the project.

3.1 Scope

The scope of the project is for a basic prototype of the proposed power management system. The focus of the system is to manage connected devices and their

corresponding power usage data.

The scope of this prototype, however, is limited to the management and analytics portion of the complete system, not the inter-device communication aspect.

I. FRONT-END

3.2 Overview

The front-end of the power home management system is an integral component of the system as a whole. Without the presence of the front-end, the back-end becomes essentially irrelevant. The front-end allows the user to interact with the smart home system through a web-page platform. Within the front-end are graphical and functional features that makes the connected devices easily controlled. The front-end consists of a login page, an administration page, a settings page, a device details page, both individual and collective, as well as a dashboard page containing system and device diagnostics.

3.2.1 Purpose The purpose of the front-end is to provide a user friendly graphical user interface which allows the user to interact with the smart home power management system. It provides a means of communication between the front-end and the back-end of the system in order to display the relevant information or changes to the user. In addition it allows the user to control the devices remotely through the platform at any location.

3.2.2 Scope The implementation of the front-end is for a smart home power management system which monitors the usage and power consumption. It therefore allows user the ability to alter power usage as well as configure alerts and manage the settings of the front-end. The front-end also allows the user to initiate remote triggering of connected devices such as lights and kitchen appliances. It is also designed to display the outcomes of triggering motion detectors and temperature sensors.

3.2.3 Definitions, Acronyms and Abbreviations

- **CSS** - Cascading Style Sheets which is a stylesheet language that sets up the appearance of the HTML [1].
- **HTML** - Hypertext Markup Language which is used to generate web pages and its objects form the basis of all websites [2].
- **Django** - A framework used to compile and render the HTML pages.
- **Bootstrap** - provides a html and css template for web development
- **Javascript** - A high level and dynamic programming language that is used with HTML and the

web [3].

3.3 Use Cases

The use cases for the front-end are as given in the tables below.

Table 1 : Table illustrating the details of the use case for trigger switches

Use Case	Trigger Switches
Description	The goal of the trigger switching is to allow the users to manually control the connected devices' state through a web-page platform.
Actors	Users within a household
System Under Design	Web-page Platform for trigger switching
Preconditions	The user has successfully passed the authentication process via the login system.
Assumptions	It is assumed that the user has a basic understanding of how to access a web-page platform using technological devices.
Non-Functional Requirements	The switches are required to be instantaneously responsive when triggered. In addition the triggers must be able to tolerate system malfunctions. .
Frequency	The usage frequency of the trigger switches are on a hourly to daily bases
Priority	It is of priority, as it is the most accessed functionality in the Smart Home Power Management system. It is also of importance since the functionality allows the user to override the time scheduling settings.

Table 2 : Table illustrating the details of the use case for the addition and removal of household devices

Use Case	Addition and Removal of Devices
Description	The purpose of the feature is to enable the user to add and remove devices on the system.
Actors	Administration user
System Under Design	Addition and Removal web-page platform
Preconditions	The user has passed the authentication procedure and they have administration rights.
Typical Flow of Events	<ol style="list-style-type: none"> 1. User logs into the system 2. Accesses the administration web-page 3. Selects the <i>Add Device</i> option 4. Provides device details in the allocated input boxes 5. Once satisfied with the details of the devices, then select the <i>save</i> button to save.
Assumptions	There is at least one allocated administration user in the household.
Non-Functional Requirements	The addition and removal of the devices in the system is expected to instant, to prevent any confusion to any of the users.
Frequency	The frequency usage of the addition and removal feature is low, since the devices will be added during the installation of the systems. If there are any device removals, this will only occur on occasion when the device/s are defective.
Priority	This is a high priority feature in the system, as this system is reliant on the ability of the administration user to add and remove devices. Without this feature, the system becomes redundant.

Table 3 : Table illustrating the details of the use case for the authentication procedure

Use Case	Authentication Procedure
Description	The aim of the authentication procedure is to ensure security on the system which results in preventing unwanted user access.
Actors	User that wishes to access the system
System Under Design	Authentication procedure
Preconditions	The user must have previously created an account on the system.
Typical Flow of Events	<ol style="list-style-type: none"> 1. User accesses login page. 2. User provides valid username in the allocated input box. 3. User provides valid password in the allocated password input box. 4. Select <i>Login</i> button to submit details for validation.
Assumptions	It is assumed that the user has an already existing account. In addition it is assumed that once the user has logged into their account, they remain logged in.
Non-Functional Requirements	The procedure is required to hide the password provided by the user and allow immediate access to the user once the provided information is verified.
Frequency	Moderate usage due to the assumption that once the user has logged in, the user remains logged unless they are accessing the system from a new mobile or desktop device.
Priority	The authentication procedure has high priority, due to the fact that the user is required to log in to access the system. This is important in terms of security, which prevents unwanted users from accessing it.

Table 4 : Table illustrating the use case of settings procedure

Use Case	Settings Procedure
Description	The focus of the settings page is to enable the user to alter both the settings of the connected devices as well as the appearance of the web-page to the user's needs.
Actors	Household users
System Under Design	Settings Procedure
Preconditions	The user is required to have passed the authentication procedure by the logging into the system.
Typical Flow of Events	<ol style="list-style-type: none"> 1. Accessing the settings page 2. Selecting either the settings for the connected devices or the appearance of the web-page. 3. The settings for the connected devices is performed by selecting the desired device that the user wishes to alter. 4. The appearance settings is opened by selecting the <i>Appearance</i> tab, in which this takes the user to the web-page that enables them to alter the settings to their customized needs. 5. For both the settings the user is required to select the <i>save</i> button to ensure that the alterations made are saved.
Assumptions	There are default settings implemented for each device during the system installation.
Non-Functional Requirements	The settings page is required to be easily accessible and understood by the user to ensure that it is user friendly.
Frequency	Moderate usage, since the settings will be performed during installation. It will only be accessed by the user should the user wish to adjust the settings due to changes done in the system.
Priorities	High priority, since this establishes the operations of each device. This includes timing, automating devices and power monitoring.

Table 5 : Table illustrating the use case of the device details page

Use Case	Device Details Page
Description	The goal of the device details page is to inform the user of the details of each individual connected devices. In addition it displays the power consumption of each connected devices, as well as the total power consumption in a graphical format.
System Under Design	Device Details Page
Preconditions	The devices are required to be connected to the system and the user must have passed the authentication procedure to have access the device details.
Assumptions	It is assumed that the devices are connected to the system, and that the user has an account.
Non-Functional Requirements	It is required to instantly display the device details in a method that is easily understood by the user.
Frequency	High usage, since the main reason for the Smart Home system is to manage the power consumption and save, the details device page is predicted to be accessed often to enable the user to analyse the power consumption in their home.
Priority	High priority, because this is a crucial component of the system, which entails the details of the each device and its power consumption.

3.4 Design Overview

The Design Overview is a section in which it illustrates the architecture of the front-end implementation in the system. It further places it into context with external systems in which is enables the reader of this document to understand the overall details of the front-end design.

3.4.1 Overall Architecture

3.4.2 System Interfaces An interface is a shared boundary across which two separate components exchange information

External User Interface Requirements:

1. *User interfaces:* The user interface enables the user to interact with they system using a graphi-

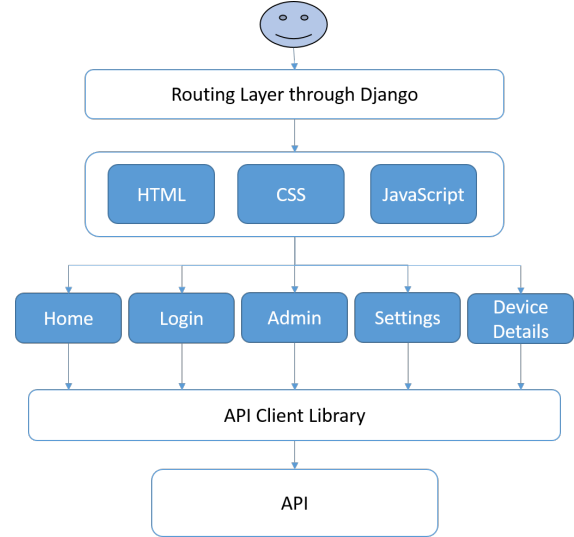


Figure 5 : Figure showing the front-end system architecture

cal user interface (GUI). This can occur without any knowledge of the back-end functionality. The GUI is in the form of a web page which provides a platform for the user to communicate with multiple connected devices. The interface requires a digital device such as a mobile or desktop device which consists of a screen and keyboard. The GUI is designed in such a way as to create an easily navigable display to ensure it is user friendly.

2. *Software Interfaces:* The front-end is required to interact with the management system in order to obtain data from the database. This is done by using the supplied Django library namely, *django.shortcuts*, in which the functions `render()` and `get_object_or_404()` are used to link the back-end with the front-end HTML.
3. *Communication Interfaces:* The communication between the management system back-end and the front-end is performed using JavaScript. The connection enables the user to interact with the database by updating it with specific actions.

3.4.3 Constraints and Assumptions **Constraints:**

The main constraint that is experienced in terms of design is time which is a significant limiting factor. The user must be able to access the interface through a web browser which requires internet connectivity.

List of Assumptions: It is assumed that the interface is designed for users that have previous knowledge and experience in navigating through a web page platform. The experience in which the user has consists of knowing how to create an account as well as logging into the account for future use. Individual assumptions regarding each use case has been discussed previously.

List of Dependencies: The front-end of the system is dependant on a single database management system

in the back-end. Without the information that can be pulled from the database, the front-end becomes essentially useless.

3.5 Non-functional Requirements

3.5.1 Performance The front-end web page needs to be able to respond quickly and in real time. It is also likely that it will need to undergo several updates triggered by user command.

3.5.2 Safety In terms of safety, the front-end needs to be able to notify the user in the event that a connected device malfunctions. This allows the user to take action depending on the notification.

3.5.3 Security The front-end reinforces security of the system by having login and authentication procedures to prevent unwanted external interference from non-household members.

3.5.4 Software Quality Attributes The front-end needs to be user friendly in a way that the user does not require any supporting documentation or prior knowledge of the system to navigate through the web page.

3.5.5 Design Constraints The front-end should be built while taking full advantage of open source libraries and supporting software to avoid the need for licensing and additional costs.

II. BACK-END

3.6 Design Overview

This will give a brief introduction to the smart home power management system back-end design.

3.6.1 System Architecture: The system architecture will give an overall view of the entire back-end system designed.

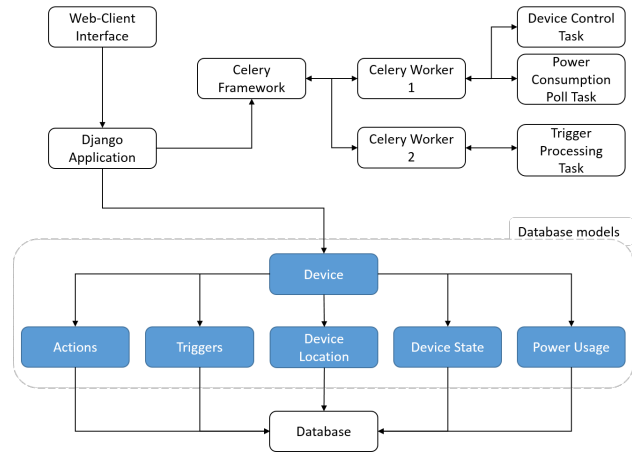


Figure 6 : System architecture diagram

3.6.2 Constraints:

- The user interface must be accessible through a web browser
- All devices in the system require wireless connectivity

3.6.3 Assumptions:

- All of the IoT devices will be connected to the same LAN
- Only a small number of simultaneous users and devices will be connected at once

3.7 Object Descriptions

- **Django Application:** This section is the main application responsible for serving the front end and controlling the database interactions. It uses the Django framework in order to implement this functionality
- **Database Models:**
 - *Device Model:* This model represents a single device. It keeps track of relevant device information, including name, IP address, location, and state. It is the core model of the database system. Two devices cannot have the same name and location attributes. Additionally, no two devices can have the same IP address.
 - *Device Location:* This model represents the location in the household where a device can be found. A location can store many devices, but a device can only be in one location.
 - *Device State:* This model represents the various power states that a device can be in. It is initially populated with the values "on" and "off," however the user still has the ability to add more depending on device capability.
 - *Triggers:* This model represents the various triggers that the system can incorporate, namely,

motion, temperature, and light. It stores an input device ID, an output device ID, and a string of Python code which will provide the conditions for the trigger to fire as well as the action to apply.

- *Actions*: This model represents the various actions that can be taken by the system when a trigger is fired.
- *Power Usage*: This model represents the cumulated power usage of the household devices. It accepts the time frame provided by the user in the front end and returns the total power consumption corresponding to that time specification.
- **Celery**: This is an asynchronous task queuing framework that supports scheduling and real-time operation. The Django application uses Celery in order to manage long running tasks such as trigger monitoring as well as controlling and monitoring devices. Using the default RabbitMQ message broker, and the MQTT messaging protocol, Celery communicates with Internet of Things devices over the network.
- *Celery Worker 1*: This worker will handle tasks requested by the user, such as changing the device state, or polling a device for power usage data.
- *Celery Worker 2*: This worker will periodically execute the trigger processing task by fetching all the triggers from the database and executing those actions which correspond to the trigger that was flagged.

3.8 Non-Functional Requirements

3.8.1 Performance Requirements: The smart home power system is an application and system that is designed for efficiency. This means that the application as well as the connecting household components need to respond quickly and in real time.

The application will most likely undergo several updates and changes for the user's benefit. The system is required to update on user command. Updates will include further improvements to the application as well as fixes for issues that arise in operation.

3.8.2 Safety Requirements: The application as well as the system components need to take certain safety concerns into account.

Due to the fact that the application is essentially controlling most of a home's appliances and electricity usage, it is important that it monitors everything it is controlling to prevent problems that may arise. The application needs to ensure that any device connected in the system does not reach a dangerous level of usage. In this case, the application should either switch off the device in question or notify the user that something is not functioning correctly and needs to be addressed.

3.8.3 Security Requirements: Due to the fact that the system controls a user's home, it therefore requires security considerations to be taken into account.

The system needs to ensure that only the user has access to the application to prevent external parties gaining control of the connected components within a house. This can be done with a signup, login and authentication process. The components within the house that the application connects with, need to be protected from external parties and therefore they must be explicitly authenticated.

3.8.4 Software Quality Attributes: The application is web based and needs to be user friendly. The functioning of the application needs to be simple so that no additional documentation or prior knowledge or experience is required.

3.9 Supplementary Documentation

3.9.1 Tools used to create diagrams:

- UML diagrams: ArgoUML, <http://argouml.tigris.org>

4. SPRINT PLANNING

The product backlog drawn up on 24 March 2016 is as shown below (ordered by priority):

- As a user I would like to be able to add/remove a device
 - Create form to add new device
 - Create list of devices which can be disabled/removed
- As a user, I would like to be able to manually switch on and off appliances remotely
 - Create page header menu
 - Create sidebar and container
 - Create page tabs
- As a user I would like to view which devices are using power, and how much they are using
 - Create dummy data on backend
 - Create basic power usage graph for each device
- As a user I would like to personalise various situations which will trigger an action
 - Create list of different triggers
 - Create trigger settings page
 - Create form to choose triggers
- As a user I need to choose which events should result in an immediate alert or notification on my system
 - Logic to monitor status of different subsystems
 - Configure how alert will be sent to user and connect accordingly
- As a user it is convenient for me to be able to define webpage settings in order to enhance the power management system experience
- As a developer I want to be able to easily run the server using Docker

- Set up docker
- Configure docker to run server and database

On this day, with two weeks remaining till project submission, a sprint planning meeting was held in order to discuss and plan the work to be performed in the first sprint which would have a timebox of one week. The screenshot below illustrates the items from the product backlog that were moved to the sprint backlog on this day.

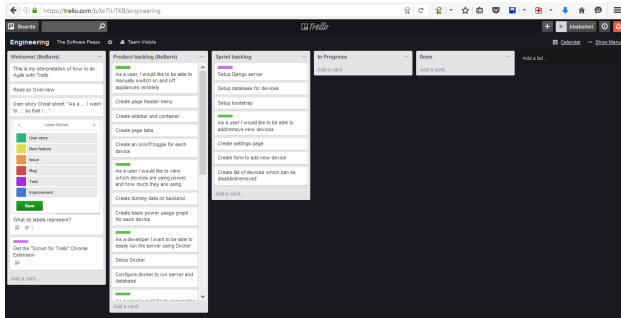


Figure 7 : Screenshot of the Trello page drawn on the first day of the first sprint for the initial prototype implementation

After the first week, a second sprint planning was held for the following week-long sprint. The screenshot below illustrates the new items that were now moved to the next sprint backlog.

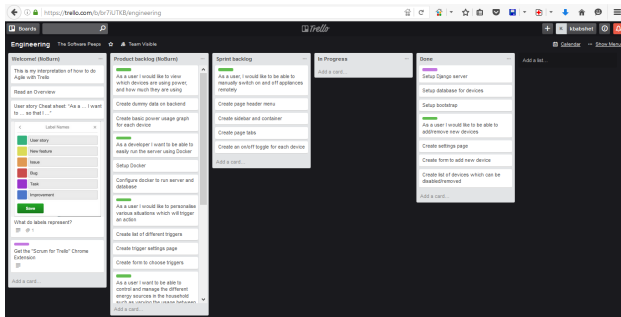


Figure 8 : Screenshot of the Trello page drawn on the first day of the second sprint of the for the initial prototype implementation

5. DESCRIPTION OF RELEVANT MODULES

Refer to the "Object Descriptions" (Section 3.4) within the Design Document for a presentation of the relevant modules implemented in the smart home power management system.

6. SPRINT RETROSPECTIVE

REFERENCES

- [1] w3schools, "Css tutorial," May 2016. [Online]. Available: www.w3schools.com/css/
- [2] —, "Html tutorial," May 2016. [Online]. Available: www.w3schools.com/html/
- [3] —, "Javascript tutorial," May 2016. [Online]. Available: www.w3schools.com/js/