Institiúid Teicneolaíochta Cheatharlach

INSTITUTE *of* TECHNOLOGY CARLOW

PiLYNK

Home Automation System

Functional Specification

Project Name:	PiLYNK Home Automation
Student Name:	Keith Byrne
Student ID:	C00170460
Class Group:	Software Development
Supervisor:	Joseph Kehoe
Document:	Functional Specification

Table of Contents

1. Project Overview

1.1 Introduction

PiLYNK will be an all in one self-contained micro computing home operating system acting as privately hosted web server in tandem. The system can be installed on any premise with ease and little to no expert training is required for initial setup and base operation. This will be of significant consideration in the physical design of the system in order to reduce installation efforts as much as possible.

Users will be able to access the system using a routine switch and link IP service such as NO-IP. This is due to the fact that most broadband providers don't offer static IP services to typical consumers and a cloud hosting solution would add more dependencies to the system. Thus, access will still be possible from anywhere in the world with a public network link and a web browser and potentially an Android application if feasible.

1.2 System Architecture Strategy Brief

The system will be divided into three separate layers to allow for all functional needs to be catered for. Design will be focused from a top down perspective with the user being the first point of contact and control being the last. A C3 (command, control and communicate) strategy will be adopted in dealing with the key concerns of controlling devices from integrated command protocols from users via the systems communication layer (front end).

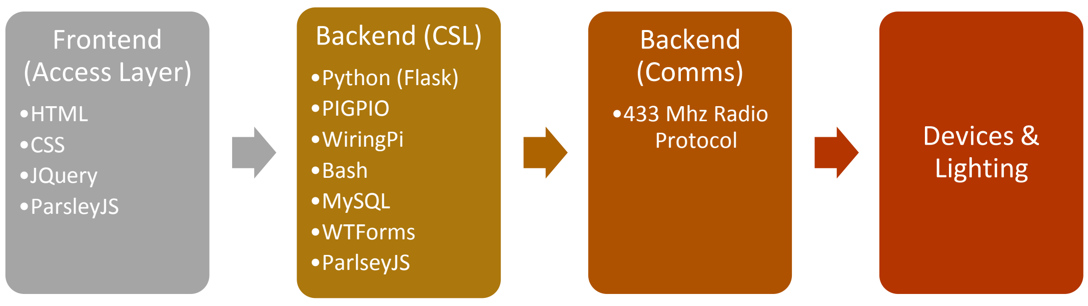


Figure 1.a - Architectural Overview (Brief)

1.2.1 Communication Access Layer (Frontend)

Present at this layer will be the principal GUI's, either in the form of a web browser application or Android application. This will provide the primary system stakeholders a method of interaction with the system. The bulk of the work regarding the frontend is carried out in iteration two and iteration three.

1.2.2 Control Service Layer (Backend)

The Control Service Layer (CSL), is where the Raspberry Pi and all integrated developments will become operational. The Raspberry Pi will be used to host a lightweight web server to complement the Pi's lower power operating model using Apache and Flask WSGI.

1.2.3 Command Layer (Backend Extension)

This layer represents the lowest level component structure of the system. All radio communication devices will be present here. It has been decided that this layer be separated from the initial backend layer because of the nature of this portion of the system. Certain aspects can't be abstracted and require a particular level of detail. Mainly due to the fact that a schematic is required for its construction and the Raspberry Pi's GPIO sits at a lower layer than the control service layer itself. However, no software requirements exist at this layer, it is purely physical and also returns no data to layer two.

1.3 System Architecture Expanded

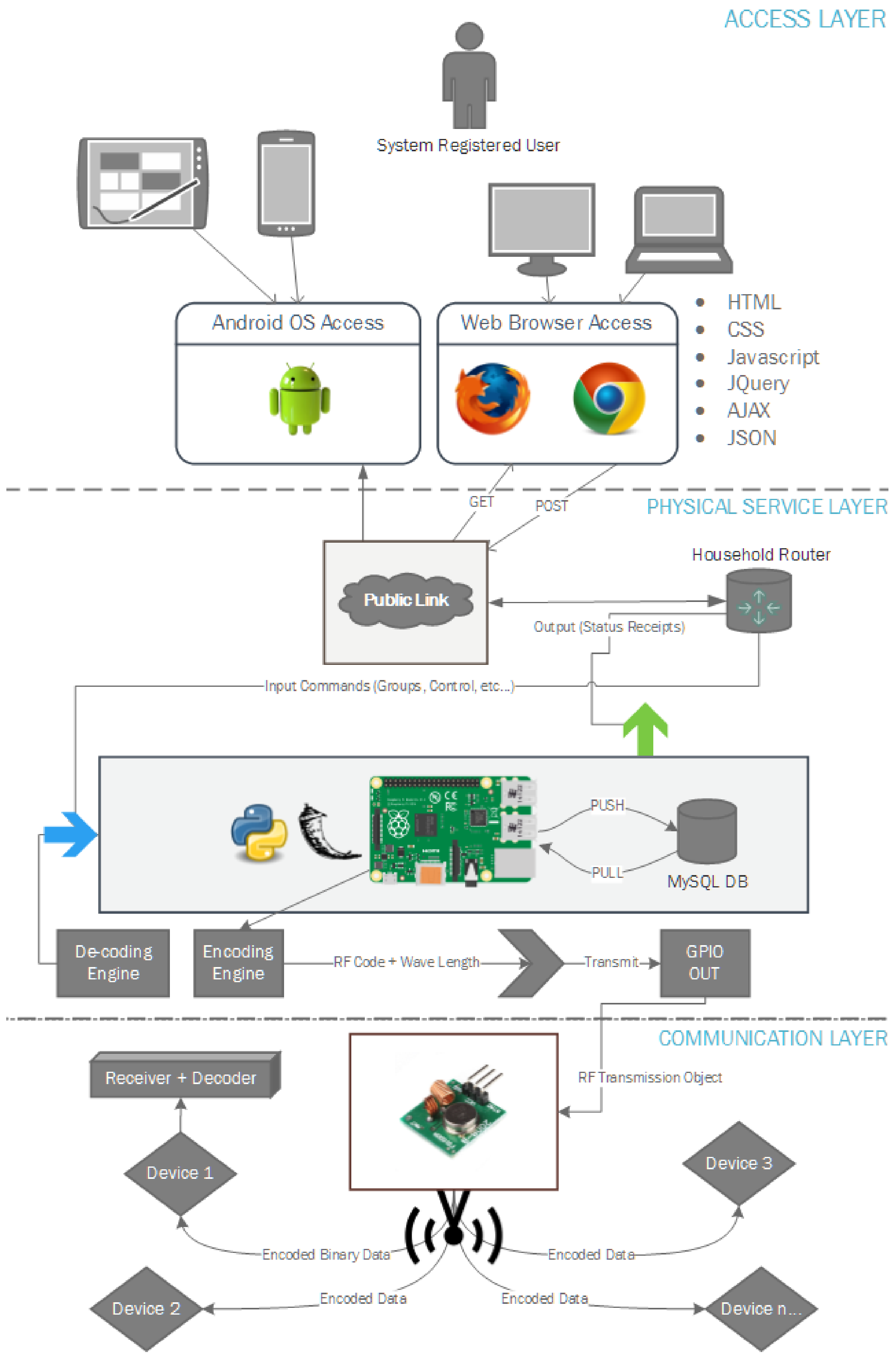


Figure 1.b - System Architecture (Detailed)

2. Functional Requirements

2.1 Key Functional Point Analysis

2.1.1 Control Device Power

This is, at its core, the most crucial functional point of the project. All other function points extend from this core feature. This will require all layers to operate effectively. In essence, it is the ability to control the flow of electrical current to a device such as a phone charger, television set, desk lamp, floor lamp, ceiling lights, etc. Users should be able to interact with this functionality via a simplistic user interface.

2.1.1.1 Outlet Device Control

This introduces the notion of controlling the flow between the household mains socket and the device in question. This can be a somewhat dangerous measure, thus, the safest method is being considered as not all interested stakeholders of the system may own their own property. Interference with the outlets on a property is forbidden in most letting agreements.

Lightwave RF offer pre built solutions to this using inline radio controlled relays integrated into a standard socket wiring solution. Deconstruction of the gang sockets is necessary in this case. Belkin WeMo offer the safer, external option of inline socket interrupts. Which is the basis of this project's control method also, at a much lower cost, however.

2.1.1.2 Lighting Control

Controlling ceiling lights offers a more difficult set of problems than standard plug and control devices. Very little third party solutions exist to allow for this. Nevertheless, during extensive research it was discovered that a set of integrated lighting socket interrupts similar to the wall socket interrupts are available.

Philips Hue specialises in this area of home automation, although most of their solutions are controlled via a locked down control bridge.

2.1.2 Device Grouping

Users can request that several devices be bound together into a node group. This will allow macro control of a set of devices without individual interaction. This has uses from parental control to total household switch off.

2.1.3 Cost Estimation

Deployed with the system should be an industry researched estimation list. This will allow users to assign estimate metric control to a device, for example, the typical wattage of a household floor lamp. These metrics can be expanded manually and the internal calculation processor will output cost estimations.

2.1.4 Accounts

Since the system will be publically accessible via the web, it is crucial that account security be held in the highest regard. For this, this set of key functionality points will be flagged as a priority one development consideration.

2.1.5 Expansion

It would be ideal to allow for the system to be expanded post-installation by a typical user. Consideration will be given to this however, deploying a test set of devices is a bigger priority. The greatest complication with this is the varying sets of radio standards available. Some devices operate on completely different protocols than the given set with this project.

2.1.6 Sensor Network Monitoring

2.1.6.1 Temperature

After extensive research into sensors compatible with the Raspberry Pi’s GPIO pins, it was concluded that integrating sensors would be relatively simplistic and consideration has been given for a temperature monitor relay system. This offers the user the ability to obverse current and past temperature readings using a simple module such as the DHT22 Digital Temperature & Humidity Sensor. Compact devices such as this use a capacitive humidity sensor and a thermistor to measure the surrounding air, and outputs a digital signal on the data pin to the Raspberry Pi’s GPIO input pin.

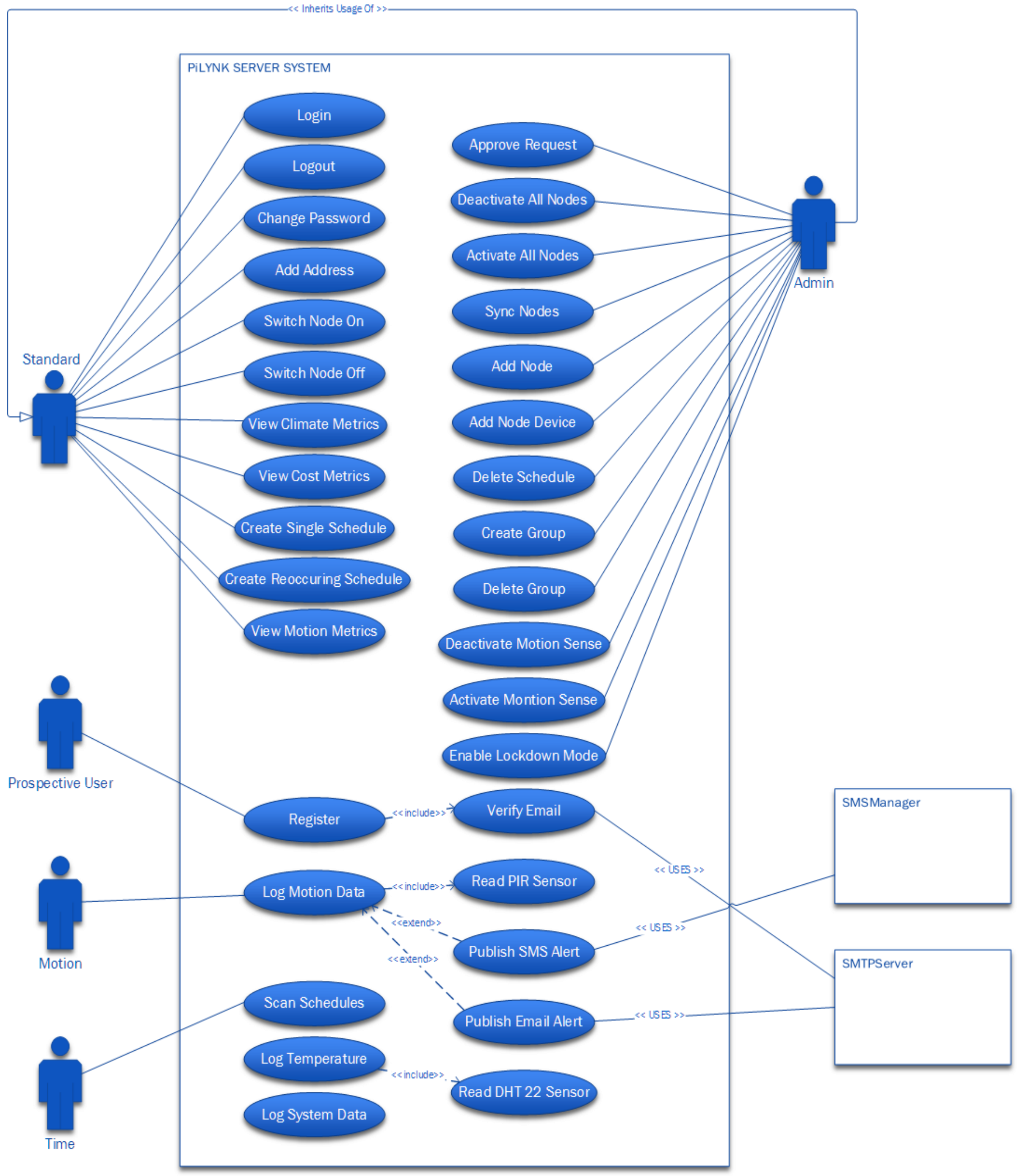
2.1.6.2 Movement

Monitoring how much movement occurs in the home or even on a business premises can be a useful situation. This can offer homeowners or premises occupiers an extended branch of safety in addition to commercial security products. As this is a highly sensitive and safety critical feature it will be in the additional functionality category and focus will not be given to this feature branch until the core components are handled. This branch can be handled will very inexpensive PIR modules, averaging at a price of two euro per unit.

3. Use Cases

3.1 Use Case Diagram

The following diagram illustrates the principal use case or user story specification for the PiLYNK system. External dependencies are noted as SMS Manager (Text Local SMS service) and External SMTP server for external communications.



3.2 Brief Use Cases

3.2.1 Login

CASE CATEGORY:	SECURITY
NAME:	Login
ACTORS:	Standard User, Administrative User, System
PRECONDITIONS:	The web application has successfully loaded from the web server and the initial login screen has been successfully loaded.
ACTIVITY:	This use begins when the user opens the PiLYNK web page. The system listens for the user login username and password. Validations carried out on the client.
CONSEQUENCE:	Once confirmed on the back end, the system displays the home page. System logs activity.

3.2.2 Logout

CASE CATEGORY:	SECURITY
NAME:	Logout
ACTORS:	Standard User, Administrative User, System
PRECONDITIONS:	The user has been successfully logged in.
ACTIVITY:	This use case begins when a user has completed all intended actions on the front end of the system. The user clicks the logout button and asked for confirmation of this action.
CONSEQUENCE:	The user session is destroyed and the user must log in again in order to return to the system dashboard. System logs activity.

3.2.3 Change Password

CASE CATEGORY:	SECURITY
NAME:	Change Password
ACTORS:	Standard User, Administrative User, System
PRECONDITIONS:	The user has been successfully logged in.
ACTIVITY:	This use case begins when a user wishes to change the login password for PiLYNK. A prompt will be enabled from the User Information section and user will be asked for previous, new and confirmation password.
CONSEQUENCE:	Previous password is deprecated and user will be required to enter new password on next login.

3.2.4 Add Address

CASE CATEGORY:	SECURITY
NAME:	Add Address
ACTORS:	Standard User, Administrative User, System
PRECONDITIONS:	The user has been successfully logged in.
ACTIVITY:	This use case begins when a user wishes to add a new address to their account. User will be prompted with an address entry screen from the User Information section.
CONSEQUENCE:	Many address will be linked to the active user on user display screens.

3.2.5 Switch Node (Device) On

CASE CATEGORY:	CONTROL
NAME:	Switch Device
ACTORS:	Standard User, Administrative User, System, RF Controller
PRECONDITIONS:	1. User has been successfully logged in and directed to the dashboard. 2. A device has been setup with the system.
ACTIVITY:	Use case begins when a user wishes to switch a linked device on or off indiscriminately. The system will feedback the current status and signal the RF Controller to activate the current to the device relay.
CONSEQUENCE:	The device will switch. Time rules will still remain active on the device. System logs activity.

3.2.6 Switch Node (Device) Off

CASE CATEGORY:	CONTROL
NAME:	Switch Device
ACTORS:	Standard User, Administrative User, System, RF Controller
PRECONDITIONS:	3. User has been successfully logged in and directed to the dashboard. 4. A device has been setup with the system.
ACTIVITY:	Use case begins when a user wishes to switch a linked device on or off indiscriminately. The system will feedback the current status and signal the RF Controller to disable the current to the device.
CONSEQUENCE:	The device will switch. Time rules will still remain active on the device. System logs activity and clips node session for node session log.

3.2.7 View Usage History

CASE CATEGORY:	METRICS
NAME:	View Usage History
ACTORS:	Standard User, Administrative User, System
PRECONDITIONS:	1. User has been successfully logged in and directed to the dashboard. 2. Historical records exists. Reports will not be permitted with no data.
ACTIVITY:	This use begins when the admin user wishes to view the activity log of the system. This log will contain historical user-system communication transactions.
CONSEQUENCE:	None

3.2.8 View Motion Metrics

CASE CATEGORY:	METRICS/SECURITY
NAME:	View Motion Metrics
ACTORS:	Standard User, Administrative User, System
PRECONDITIONS:	1. User has been successfully logged in and directed to the dashboard. 2. Historical records exists. Reports will not be permitted with no data.
ACTIVITY:	This use begins when the admin user wishes to view the motion activity log of the PIR movement records. System will report lockdown and standard capture records distinctly.
CONSEQUENCE:	None

3.2.9 Activate Motion Sense

CASE CATEGORY:	SECURITY
NAME:	Enable Motion Sense
ACTORS:	Administrative User
PRECONDITIONS:	1. User has been successfully logged in and directed to the dashboard and is admin. 2. PIR Motion service is disabled.
ACTIVITY:	This use case begins when a user wishes to enable the motion sense background listening service for the PIR motion sensor.
CONSEQUENCE:	All motion activity will be logged.

3.2.10 Deactivate Motion Sense

CASE CATEGORY:	SECURITY
NAME:	Disable Motion Sense
ACTORS:	Administrative User
PRECONDITIONS:	1. User has been successfully logged in and directed to the dashboard and is admin. 2. PIR Motion service is enabled. 3. Lockdown mode is not enabled.
ACTIVITY:	This use case begins when a user wishes to disable the motion sense background listening service for the PIR motion sensor.
CONSEQUENCE:	No further motion event records will be logged.

3.2.11 Enable Lockdown Mode

CASE CATEGORY:	SECURITY
NAME:	Enable Lockdown Mode
ACTORS:	Administrative User
PRECONDITIONS:	1. User has been successfully logged in and directed to the dashboard and is admin.
ACTIVITY:	This use case handles to enabling of the lockdown functionality. User selects enable, the system registers data exchanges and enables alert broadcast functionality.
CONSEQUENCE:	Registered admins will receive email notifications of motion alerts. If motion sense was not active, it will be activated after the request is complete.

3.2.12 View Users

CASE CATEGORY:	SECURITY
NAME:	View Users
ACTORS:	Administrative User
PRECONDITIONS:	1. User has been successfully logged in and directed to the dashboard and is admin.
ACTIVITY:	This use case handles the action of displaying all active user information for administrative purposes.
CONSEQUENCE:	None.

3.2.13 View Climate Metrics

CASE CATEGORY:	METRICS
NAME:	View Climate Metrics
ACTORS:	Standard User, Administrative User.
PRECONDITIONS:	1. User has been successfully logged in and directed to the dashboard. 2. System has registered temperature records.
ACTIVITY:	This use case relates to the reporting of temperature sensor relay information. A user selects daily, weekly or monthly to view the *general* report for that duration.
CONSEQUENCE:	System logs activity.

3.2.14 View Cost Metrics

CASE CATEGORY:	METRICS
NAME:	View Cost Metrics
ACTORS:	Standard User, Administrative User.
PRECONDITIONS:	1. User has been successfully logged in and directed to the dashboard. 2. System has registered temperature records.
ACTIVITY:	This use case relates to the reporting of temperature sensor relay information. A user selects daily, weekly or monthly to view the *general* report for that duration.
CONSEQUENCE:	System logs activity.

3.2.15 View Motion Metrics

CASE CATEGORY:	METRICS
NAME:	View Motion Metrics
ACTORS:	Standard User, Administrative User.
PRECONDITIONS:	1. User has been successfully logged in and directed to the dashboard. 2. System has registered temperature records.
ACTIVITY:	This use case relates to the reporting of temperature sensor relay information. A user selects daily, weekly or monthly to view the *general* report for that duration.
CONSEQUENCE:	System logs activity.

3.2.16 Create Single Schedule

CASE CATEGORY:	CONTROL
NAME:	Create Single Schedule
ACTORS:	Standard User, Administrative User
PRECONDITIONS:	A single linked node exists
ACTIVITY:	User wishes to initiate a single schedule trigger (Date/time to Date/time no repetition). Form details are returned to client and user fills in relevant details and submits the request for data acceptance.
CONSEQUENCE:	Set nodes will fire only once.

3.2.17 Create Multi Schedule

CASE CATEGORY:	CONTROL
NAME:	Create Multi Schedule
ACTORS:	Administrative User, Standard User
PRECONDITIONS:	A single linked node exists
ACTIVITY:	User wishes to initiate a multi schedule trigger (Date/time to Date/time with repetition of selected weekdays). Form details are returned to client and user fills in relevant details and submits for database acceptance.
CONSEQUENCE:	Set nodes will fire each set day until schedule is disabled.

3.2.18 Register

CASE CATEGORY:	SECURITY
NAME:	Register
ACTORS:	Prospective User
PRECONDITIONS:	Standard user credentials do not exist
ACTIVITY:	This use case begins when a non-registered user wishes to access the system. The user enters their details via application form. The form is submitted to the system and then the decision is forwarded for approval or rejection.
CONSEQUENCE:	The user must wait until they have been approved before proceeding to any future stages of the system. This period is purely at the discretion of the principal administrator.

3.2.19 Approve Request

CASE CATEGORY:	SECURITY
NAME:	Approve Request
ACTORS:	Administrative User
PRECONDITIONS:	1. Admin user has been successfully logged in. 2. A user has requested system access (Register use case dependency)
ACTIVITY:	This use begins when an admin user receives an account access request from a non-registered user. The admin user has the right to review and deny or approve the request be it from a family member or other member of the same household.
CONSEQUENCE:	The user requesting access gets accepted or denied with the system and receives email notification of the process.

3.2.20 Verify Email

CASE CATEGORY:	CONTROL
NAME:	Verify Email
ACTORS:	Unregistered
PRECONDITIONS:	Registration has been successful
ACTIVITY:	This use case handles the approval process of new users. Registered users will be sent an approval mail to verify email. Users are required to follow the link and update verification status
CONSEQUENCE:	User email verified status is updated.

3.2.21 Delete Schedule

CASE CATEGORY:	CONTROL
NAME:	Delete Schedule
ACTORS:	Standard User, System
PRECONDITIONS:	A single schedule exists
ACTIVITY:	This use begins when a user wishes to remove an existing node schedule
CONSEQUENCE:	Database will be cleared of aforementioned schedule

3.2.22 Edit Schedule

CASE CATEGORY:	CONTROL
NAME:	Edit Schedule
ACTORS:	Standard User, System
PRECONDITIONS:	A single schedule exists
ACTIVITY:	This use begins when a user wishes to edit an existing node schedule. User selects alterations to the schedule much like creating a new one, once reviewed, changes are submitted and the updated schedule is registered.
CONSEQUENCE:	N/A

3.2.23 Sync Nodes

CASE CATEGORY:	CONTROL
NAME:	Sync Nodes
ACTORS:	Administrative User
PRECONDITIONS:	At least one node has been linked to the system
ACTIVITY:	This use case begins when an admin user wishes to synchronise all linked nodes to the current system status. This handles out of sync device statuses.
CONSEQUENCE:	Devices will turned on or off depending on their previous status.

3.2.24 Activate All Nodes

CASE CATEGORY:	CONTROL
NAME:	Activate All Nodes
ACTORS:	Administrative User
PRECONDITIONS:	At least one node has been linked to the system
ACTIVITY:	This use case begins when an admin user wishes to activate all linked nodes and update system status.
CONSEQUENCE:	***ALL*** devices will be turned on.

3.2.25 Deactivate All Nodes

CASE CATEGORY:	CONTROL
NAME:	Deactivate All Nodes
ACTORS:	Administrative User
PRECONDITIONS:	At least one node has been linked to the system
ACTIVITY:	This use case begins when an admin user wishes to deactivate all linked nodes and update system status.
CONSEQUENCE:	***ALL*** devices will be turned off.

3.2.26 Create Device Group

CASE CATEGORY:	CONTROL
NAME:	Create Device Group
ACTORS:	Administrative User, System
PRECONDITIONS:	A single linked node exists
ACTIVITY:	This use begins when a user wishes to create a control group of nodes. The users selects the nodes which to include and additional details for the group and submits the request.
CONSEQUENCE:	Included nodes will be listed under control group until remove manually.

3.2.27 Delete Device Group

CASE CATEGORY:	CONTROL
NAME:	Delete Device Group
ACTORS:	Administrative User, System
PRECONDITIONS:	A single control group exists
ACTIVITY:	This use begins when a user wishes to delete an existing control group. User selects the group and submits a deletion request for that group.
CONSEQUENCE:	Group will no longer exist in the system.

3.2.28 Add Node

CASE CATEGORY:	CONTROL
NAME:	Add Device
ACTORS:	Administrative User, System
PRECONDITIONS:	None
ACTIVITY:	This use deals with the process of adding a new device node to the system network. Users will post relevant form details and the system will determine the next available bit sequence to control the device, map it to the node data and test the node.
CONSEQUENCE:	Node will be linked to the programmed bit sequence

3.2.29 Create Single Schedule

CASE CATEGORY:	CONTROL
NAME:	Create Single Schedule
ACTORS:	Standard User, System
PRECONDITIONS:	A single linked node exists
ACTIVITY:	User wishes to initiate a single schedule trigger (Date/time to Date/time no repetition). Form details are returned to client and user fills in relevant details
CONSEQUENCE:	Set nodes will fire only once.

3.2.30 Create Repetitive Schedule

CASE CATEGORY:	CONTROL
NAME:	Create Repetitive Schedule
ACTORS:	Standard User, System
PRECONDITIONS:	A single linked node exists
ACTIVITY:	User wishes to initiate a multiple time schedule trigger (Date/time to Date/time with repetition). Form details are returned to client and user fills in relevant details.
CONSEQUENCE:	Set nodes will fire at user set intervals for each day assigned

3.2.31 Add Node Device

CASE CATEGORY:	CONTROL
NAME:	Add Node Device
ACTORS:	Administrative User, System
PRECONDITIONS:	A single linked node exists
ACTIVITY:	The use case involves the action of linking a new device for cost metrics to an existing node.
CONSEQUENCE:	None

3.2.32 Log Motion Data

CASE CATEGORY:	SECURITY
NAME:	Log Motion Data
ACTORS:	Motion, System
PRECONDITIONS:	Motion Sense is active
ACTIVITY:	This use case is a motion triggered event. The system logs the date and time of the event and triggers any motion triggered nodes for thirty seconds. If lock down mode is enabled, the system will also distribute alerts to any admins on the broadcast list.
CONSEQUENCE:	None

3.2.33 Scan Schedules

CASE CATEGORY:	CONTROL
NAME:	Scan Schedules
ACTORS:	Time, System
PRECONDITIONS:	At least a single schedule has been setup.
ACTIVITY:	This use case involves the system and time based interval triggering to activate and deactivate nodes based on start and end date time signatures from a user setup schedule entry.
CONSEQUENCE:	Node will be switched on or off.

3.2.34 Log Temperature

CASE CATEGORY:	METRICS
NAME:	Log Motion Data
ACTORS:	Motion, System
PRECONDITIONS:	Motion Sense is active
ACTIVITY:	This use case is a motion triggered event. The system logs the date and time of the event and triggers any motion triggered nodes for thirty seconds. If lock down mode is enabled, the system will also distribute alerts to any admins on the broadcast list.
CONSEQUENCE:	None

3.2.35 Read DHT 22 Sensor

CASE CATEGORY:	CONTROL
NAME:	Read DHT22 Sensor
ACTORS:	System
PRECONDITIONS:	None
ACTIVITY:	This use case begins when the system triggers a sensor read of the DHT22 climate sensor or when a user first logs in. The system reads the values from the sensor data returned and parses the result into temperature, humidity and meta values.
CONSEQUENCE:	None

3.3 Detailed Use Cases

3.3.1 Login

Actors: Administrator; Guest.

Description: The use begins when either user type wishes to login into the system.

Preconditions: None

Main Success Scenario:

1. The system renders the login view for user input.
2. The user enters username or email and password and submits the request.
3. The system validates the user credentials.
4. The system returns a response stating that the login attempt was successful.
5. The system returns a redirect for the client to navigate to home page.

Alternatives:

3a. User validation has been unsuccessful due to invalid details.

   1. The system returns a negative validation attempt and prompts the user accordingly.

3.3.2 Logout

Actors: Administrator; Guest.

Description: This use begins when an admin user or guest wishes to log out of the application.

Preconditions: User must be logged in and hold current session information.

Main Success Scenario:

1. The user selects the Logout option from the main system dashboard.
2. The system clears user session data
3. The system then returns the user to the initial login view.

3.3.3 Change Password

Actors: Administrator; Guest.

Description: This use begins when an admin user or guest user wishes to change their current login password information.

Preconditions: User must be logged in and hold current session information.

Main Success Scenario:

1. The user selects the Change Password dialog from the main dashboard.
2. The system renders the Change Password view and returns the view to the user.
3. The user inputs old password, new password and a confirmation of new password.
4. The user submits the password information to the system.
5. The system verifies the password is a valid password string.
6. The system registers the new password.
7. The system returns a confirmation response to the user.

Alternatives:

5a. Input password string is not a valid format or length.

   1. The system returns a negative action response to the user.

3.3.4 Add Address

Actors: Administrator; Guest.

Description: This use begins when an admin user or guest wishes to add a new address to their user account.

Preconditions: User must be logged in and hold current session information.

Main Success Scenario:

1. The user selects the Add Address dialog from the main dashboard.
2. The system renders the Add Address view and returns the view to the user.
3. The user inputs new address details including: house, street, town, county, country and a current address flag and then submits the data.
4. The system validates the integrity of the address data.
5. The system inserts the new data into the database.
6. The system returns a confirmation response that the address was registered.

3.3.5 Switch Node On

Actors: Administrator; Guest.

Description: This use begins when an admin user or guest wishes to switch a device on.

Preconditions: User must be logged in and hold current session information. At least one node must be linked to the system.

Main Success Scenario:

1. The user views the list of linked devices from the main dashboard [login redirect].
2. The user selects switch 'on' node from the node interface panel.
3. The system receives the request.
4. The system then retrieves node radio data for that node.
5. The system processes the node radio code and creates a transmitter (TX) object.
6. The system offers the code to the TX object with relevant waveform construct data.
7. The TX control object then parses the base 10 string to bit code form.
8. The TX outputs the code using a GPIO control library.
9. The system returns any clock or waveform errors.
10. The system updates the node status after successful switch.
11. The system adds the node to the session context manager (for live analysis).
12. The system then notifies the user that the device has been successfully switched.

Alternatives:

5a. TX object is already actively transmitting from another request.

   1. System waits for control of GPIO object.

8a. Error in transmission of radio code.

   1. System receives an interrupt from the TX object that the waveform was not handled.
   2. System notifies user that there was an error handling the request and prompts to try again soon after.

11a. Node unique identifier already present in session context manager.

   1. System bypasses the session entry.
   2. System then notifies the user that the device has been successfully switched.

3.3.6 Switch Node Off

Actors: Administrator; Guest.

Description: This use begins when an admin user or guest wishes switch a node off.

Preconditions: User must be logged in and hold current session information. At least one node must be linked to the system.

Main Success Scenario:

1. The user views the list of linked devices from the main dashboard [login redirect].
2. The user selects switch on control from the node interface panel.
3. The system receives the request.
4. The system then retrieves node radio data for that node.
5. The system processes the node radio code and creates a transmitter (TX) object.
6. The system offers the code to the TX object with relevant waveform construct data.
7. The TX control object then parses the base 10 string to base 2 code form.
8. The TX outputs the code using a GPIO control library.
9. The system returns any clock or waveform errors.
10. The system updates the node status after successful switch.
11. The system ends the node session in the session context manager.
12. The system removes the node from the session.
13. The system adds the node session data to the database.
14. The system then notifies the user that the device has been successfully switched.

Alternatives:

5a. TX object is already actively transmitting from another request.

1. System waits for control of GPIO object.

8a. Error in transmission of radio code.

1. System receives an interrupt from the TX object that the waveform was not handled.
2. System notifies user that there was an error handling the request and prompts to try again soon after.

11a. Node unique identifier is not present in session context manager.

1. System bypasses the process of ending the node session.
2. System then notifies the user that the device has been successfully switched.

3.3.7 View Climate Metrics

Actors: Administrator; Guest.

Description: This use begins when an admin user or guest wishes to view the climate reports.

Preconditions: User must be logged in and hold current session information. At least one element of climate data must be captured.

Main Success Scenario:

1. User wishes to view the captured temperature and humidity records by selecting the option from the main dashboard.
2. The system receives the request.
3. The system generates a report for daily, weekly and monthly data sets.
4. The system renders the view and returns the report data.
5. User is prompted with the view screen and metric data.

3.3.8 View Cost Metrics

Actors: Administrator; Guest.

Description: This use begins when an admin or guest wishes to view cost analysis report.

Preconditions: User must be logged in and hold current session information. At least one node session must be recorded.

Main Success Scenario:

1. User wishes to view the captured temperature and humidity records by selecting the option from the main dashboard.
2. The system receives the request.
3. The system generates a report for daily, weekly and monthly data sets.
4. The system renders the view and returns the report data.
5. User is prompted with the view screen and metric data.

3.3.9 Create Single Schedule

Actors: Administrator; Guest.

Description: This use begins when an admin or guest wishes to create a one off schedule.

Preconditions: User must be logged in and hold current session information. At least one node must be linked to the system.

Main Success Scenario:

1. User selects the create schedule option from the main dashboard.
2. System renders and returns the create schedule view and form data.
3. User then selects create single schedule option.
4. User enters relevant schedule data and submits the form.
5. System validates the request.
6. System insert schedule data into the database.
7. System notifies the user that the data has been successfully registered.

Alternatives:

5a. Submitted request data is invalid

1. System returns an error message to the user indicating data is invalid.
2. System waits for further input from the user.

3.3.10 View Motion Metrics

Actors: Administrator; Guest.

Description: This use begins when an admin or guest wishes to view the motion metric data.

Preconditions: User must be logged in and hold current session information. At least one motion trigger event must be recorded by the system.

Main Success Scenario:

1. User selects view motion metric data from the main dashboard.
2. System retrieves motion data from the database and processes report.
3. System returns view and relevant motion data to the user.

3.3.11 Register

Actors: Prospective User

Description: This use case begins when an un-registered user wishes to submit an access request for the PiLYNK system.

Preconditions: None

Main Success Scenario:

1. The prospective user selects the “Register” section on the login screen.
2. The system displays the registration view.
3. The user enters their details, including username, password, address and extended info.
4. The system validates the data.
5. The system automatically assigns permission ‘N’ to the user submission.
6. The system registers the data into persistent storage.
7. The system transmits an email verification notice to the registered email address.
8. The system notifies the user that the registration was successful.

Alternatives:

4a. Input username is not alpha only.

1. System returns a response to the user to amend their username
2. System awaits further input.

4b. Input username is already registered.

1. System returns a response to the user to amend their username.
2. System awaits further input.

4c. Input email is already registered.

1. System returns a response to the user to amend their email.
2. System awaits further input.

4d. Form input data is invalid.

1. System returns a response with form errors to the user.
2. System awaits further input.

3.3.12 Approve Request

Actors: Administrator

Description: This use begins when an admin user wishes to approve a registration request to the system.

Preconditions:

1. User must be logged in and hold current session information.
2. User must be administrative user.

Main Success Scenario:

1. The admin user selects the Approval Requests section from the main dashboard.
2. The system retrieves all request pending user data.
3. The system renders the view with the data and returns it to the user.
4. The user reviews the user request data.
5. The user selects a request approval from the dashboard.
6. The system updates the approval user entry in the database.
7. The system returns a confirmation response to the user.

Alternatives:

4a. Admin select deny request option for candidate request.

1. The system invalidates the candidate request from the database.
2. The system return a confirmation response to the user.

3.3.13 Deactivate All Nodes

Actors: Administrator.

Description: This use begins when an admin wishes to deactivate all nodes (master).

Preconditions: User must be logged in and hold current session information. At least one node must be linked to the system.

Main Success Scenario:

1. User selects deactivate all nodes control from the main dashboard.
2. System retrieves request.
3. System retrieves all node and radio transmission data from the database.
4. System creates a transmitter (TX) control object.
5. System iterates each node in the master collection.
6. System provides each code from each node to the TX object.
7. The system then outputs each TX code using the GPIO output.
8. The TX object parses the base 10 code strings to base 2 code form.
9. The TX object returns any subsequent waveform errors.
10. The system updates the status for all nodes linked to the system.
11. The system ends any currently active node sessions.
12. The system then registers any closed node sessions and removes them accordingly.
13. The system then notifies the user that all nodes has been successfully switched off.

Alternatives:

5a. TX object is unable to acquire GPIO control.

1. System waits for completion of current process.

3.3.14 Activate All Nodes

Actors: Administrator.

Description: This use begins when an admin wishes to activate all nodes (master).

Preconditions: User must be logged in and hold current session information. At least one node must be linked to the system.

Main Success Scenario:

1. User selects activate all nodes control from the main dashboard.
2. System retrieves request.
3. System retrieves all node and radio transmission data from the database.
4. System creates a transmitter (TX) control object.
5. System iterates each node in the master collection.
6. System provides each code from each node to the TX object.
7. The system then outputs each TX code using the GPIO output.
8. The TX object parses the base 10 code strings to base 2 code form.
9. The system then outputs each TX on code using the GPIO output.
10. The system updates the status for all nodes linked to the system.
11. The system then registers all nodes to the session context manager.
12. The system then notifies the user that all nodes has been successfully switched off.

Alternatives:

5a. TX object is unable to acquire GPIO control.

1. System waits for completion of current process.

11a. Node n is currently in the node session control collection.

1. System ignores activation time and bypasses addition.

3.3.15 Sync Nodes

Actors: Administrator.

Description: This use begins when an admin wishes to synchronise all nodes (master).

Preconditions: User must be logged in and hold current session information. At least one node must be linked to the system. Request has not been made twice in the timeout period.

Notes: This request will override any pending or current requests made to the transmission object as it is consider critical admin control.

Main Success Scenario:

1. User selects synchronise nodes from the main dashboard.
2. The system receives the request.
3. The gets all node data and transmission data from the database.
4. The system creates a transmitter (TX).
5. System iterates each node in the master collection.
6. System provides each code from each node to the TX object.
7. The system then outputs each TX code using the GPIO output.
8. The TX object parses the base 10 code strings to base 2 code form.
9. The system then signals the TX object to broadcast all off codes (respective of status).
10. The system checks the node session manager to ensure node n is not present and clears accordingly.
11. The system then signals the TX object to broadcast all on codes (respective of status).
12. The system checks the node session manager to ensure node n is present.
13. The system then notifies the user that all nodes has been successfully synchronised.

4. Supplementary Specification

4.1 Objectives

The purpose of this section of the document is to define supplementary requirements of the PiLYNK system. This Supplementary Specification lists the requirements that are not readily captured in the use cases of the use-case model. The Supplementary Specifications and the use-case model together capture a complete set of requirements on the system.

4.2 Functionality

This application places a heavy focus on responsive user interaction with the system and the precise, reliable and reportable control of devices around the home. After all, the system is responsible for reducing the cost of living but also to be safe.

Supplementing the base functionality covered in the Use Cases, this section contains a list of functional considerations that are common to more than one use case.

4.2.1 System Activity Logging

All core system activity will be logged. This is in order to monitor trends in activity so the system may suggest improvements. This is however a post project submission consideration.

The system activity messages will include a text description of the error, date time of the activity event, user details and the event sub system description. All system activity reports will be stored in a separate Audit Control Database with access granted only to the principal system administrator.

4.2.2 System Error Logging

All exceptional system behaviour will be logged. This is to allow the system administrator to understand the cause of critical system failure in order to diagnose the problem.

The system error messages will include a detailed text description of the error, date time of the event, which user triggered the sequence to cause the event and which section of the server was responsible for throwing the exception.

4.3 Usability

It has been heavily considered that not all system users will be computer literate adults. Most of these uX interaction related issues will be teased out during the core design phase of the project. This section lists all of those requirements that relate to, or affect, the usability of the system.

4.3.1 Browser Compliance

Development consideration cannot be given to all currently used browsers available. Google Chrome, mobile browsers and Mozilla Firefox will be the focus point of the design.

4.3.2 Design for Ease of Use

The user interface of the PiLYNK System will be designed for ease-of-use and shall be appropriate for both computer-literate and limited literacy users. No training will be required to use the system.

4.4 Reliability

This section lists the reliability attributes of the system.

The system should be available 24 hours a day, 7 days a week. Downtime should only occur when external forces are involved, principally regarding power outages and maintenance on the premises. Nevertheless, the system should be able to automatically recover and restart server operations once power has resumed. A maximum of 10% downtime per year should be the achieved.

4.5 Performance

The performance characteristics of the system are outlined in this section.

4.5.1 Simultaneous Users

The system should be able to support a number of users at one time. However, due to the typical bandwidth of home broadband, this should not be strained extensively. Subsequently, a maximum count of ten users will be enforced in the system.

4.5.2 Database Access Response Time

Due to certain CPU limitations on the Raspberry Pi, database read and writes may be slow if the buffer is overloaded. Due to this, certain capacity thresholds may be implemented to allow for successful albeit slow right over many unsuccessful writes. One possible consideration for this is a timeout counter on device control.

4.6 Supportability

The system will not initially be heavily concerned with supporting minor changes. Because of the number of integrated technologies ranging from Python, Javascript, HTML, CSS, etc. Any updates may cause dependency related compatibility failures. Hence, major system upgrades will be handled on a board by board basis rather than exclusively any particular module or sub section of the system. Upgrades will be carried out on a dormant system board rather than a live system board at any time.

Newly developed, tested and stable iterations will be available via GitHub for product owners. This will be a simple pull, update, restart server process with written assistance and troubleshoots for each revision.

4.7 Security

4.7.1 Storage

Due to the nature of the system having the ability to control devices in the home and potentially create fire hazards, the security specific features and various other threats found through poor data protection, a strong and secure user presence and persistent storage is required. To tackle this issue, the system will be developed with encrypted username and password storage using the Python Werkzeug library, specifically the SHA1 hashing algorithms.

4.8 Design Constraints

This section lists any design constraints on the system being built.

4.8.1 Platform Requirements

As the system will be running Python/Flask micro web framework as the server technology. Development will be focused around Python version 2.7 and continuously the latest Flask releases. C++ version will be the default version included on the Raspberry Pi B+ model.

The server operating system will be the Debian based Raspbian:

- **Wheezy**: Kernel 3.18
- **Released**: May 2015

5. Project Plan

The project will be split into three development iterations or sprints and sub phases within each stage. In each of these iterations, certain deliverables are expected to be produced. This plan will detail the necessary timeframe to complete all objectives, research, reviews, testing and production integration.

5.1 Conceptual Design Philosophy

The ordering of function priority is first to allow for control, then use control functionality to provide metrics for cost analysis and passive metrics without control, then this is all maintained by a secure foundation of safe data. Illustrated below the top down representation of functional importance and lower development branches being of data importance. This admits a contrasting division of security and control, which ensures that control is not possible without security and there is no requirement for security if there is no control.

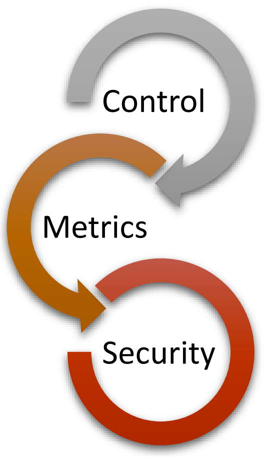


Figure 3 - Critical Function Model

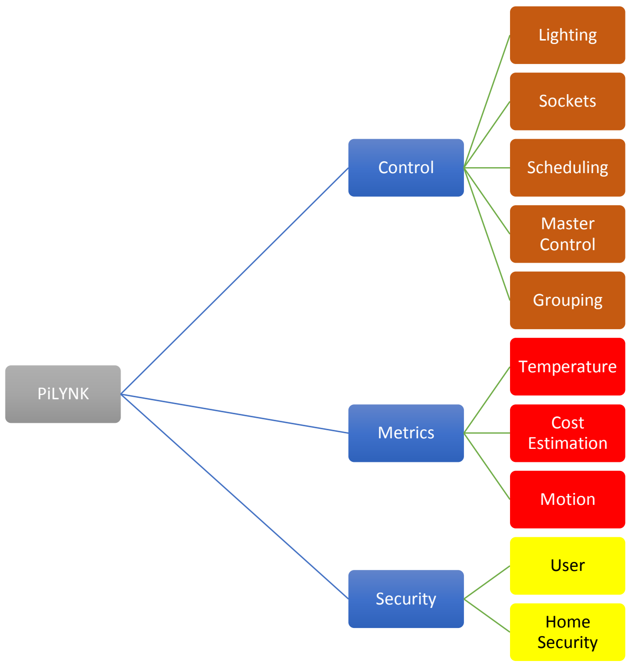


Figure 4 - Concept Branch Diagram

5.2 Iteration One

5.2.1 Infrastructure Setup

The core focus of iteration one should be to enable communications between devices and the raspberry pi itself. For this, a prototype web server with minimal functionality should be developed to test networking access and load handling on the Flask server. The raspberry pi operating system must be installed to achieve this, all Python dependency libraries must be downloaded and installed. MySQL must be installed and LYNK schema created.

5.2.2 Device Setup

For this stage of the first iteration, test devices are required. The first phase device range consists simple of RF controlled inline relay switches adapted for 240v mains interrupt. These devices are available in several outlets and rank at an average recommended retail price of seven euro per unit. Most packages include device remotes also.

5.2.3 RF Controller

To control any RF devices, a module must be implemented to transmit binary codes encoded for use with receiving devices. To accomplish this, waveform analysis of the basic remotes provided with the RF relays is necessary. This can be accomplished using an adapted 1/8 inch inline microphone jack wired from a basic 5v receiver unit and routed through to any modern digital audio workstation or logic analyser on desktop or laptop. This phase of the first iteration is expected to be the most difficult in terms of logical problems and system sensitive issues such a CPU interrupts, among other expected issues.

5.2.4 Networking and Public Domain Linking

In order to allow users to control the devices remotely, the Raspberry Pi web server must be accessible from outside the home local area network. The cheapest and most practical solution to this is to a dynamic fixing service to allow for the changing IP address of the web server to be looked up using a dynamic domain name server (DDNS). For this, NO-IP is a highly promising candidate as it is free subscription and can provide services for a proof of concept and also long term scalability.

This phase also includes network setup of the home router, allowing for relevant network ports to be opened and packets forward to the Raspberry Pi. Initial development will be handled using Flask default port 5000 and later port 80 under Apache WSGI routing through to Flask.

5.3 Iteration Two

The focus of iteration two is presentation, data and extended communications building on and functionally integrating iteration one deliverables. This is to be the most critical phase of the entire project where the most important foundations for later developments will be based. The graphical user interface of the system will also be developed heavily during this iteration.

5.3.1 Database Modelling

In order to ensure that the system scales with integrity over a long period of time. A heavily constrained and properly normalised database is required. The system will capture aspects of data relating to objects such as user, nodes, devices, temperature logs, motion logs, node activity session logs, radio broadcast information and meta-data on nodes such as device wattage, voltages and various other measurable data elements.

5.3.2 Graphical User Interface

As specified in the initial vision, users should be allowed to interface with their home and setup automation and control devices using a fluid user interface that is developed to a modern standard using W3C web development standards, allow for some cross browser support and allows for a simplistic control interface for non-technical users. An initial design will the foundation of this development and will be changed as the project progresses and needs assessed. The following image is a sample of a basic design foundation:

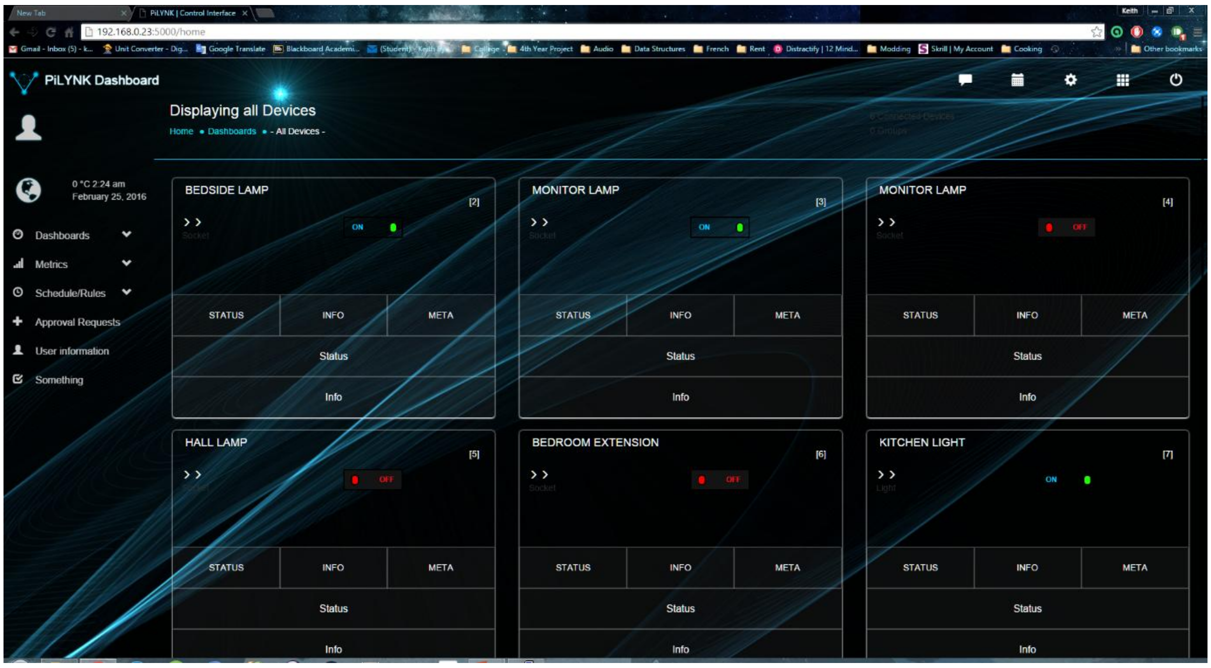


Figure 5 - Prototype Dashboard Interface - Iteration Two

5.3.3 Scheduled Services and Background Services

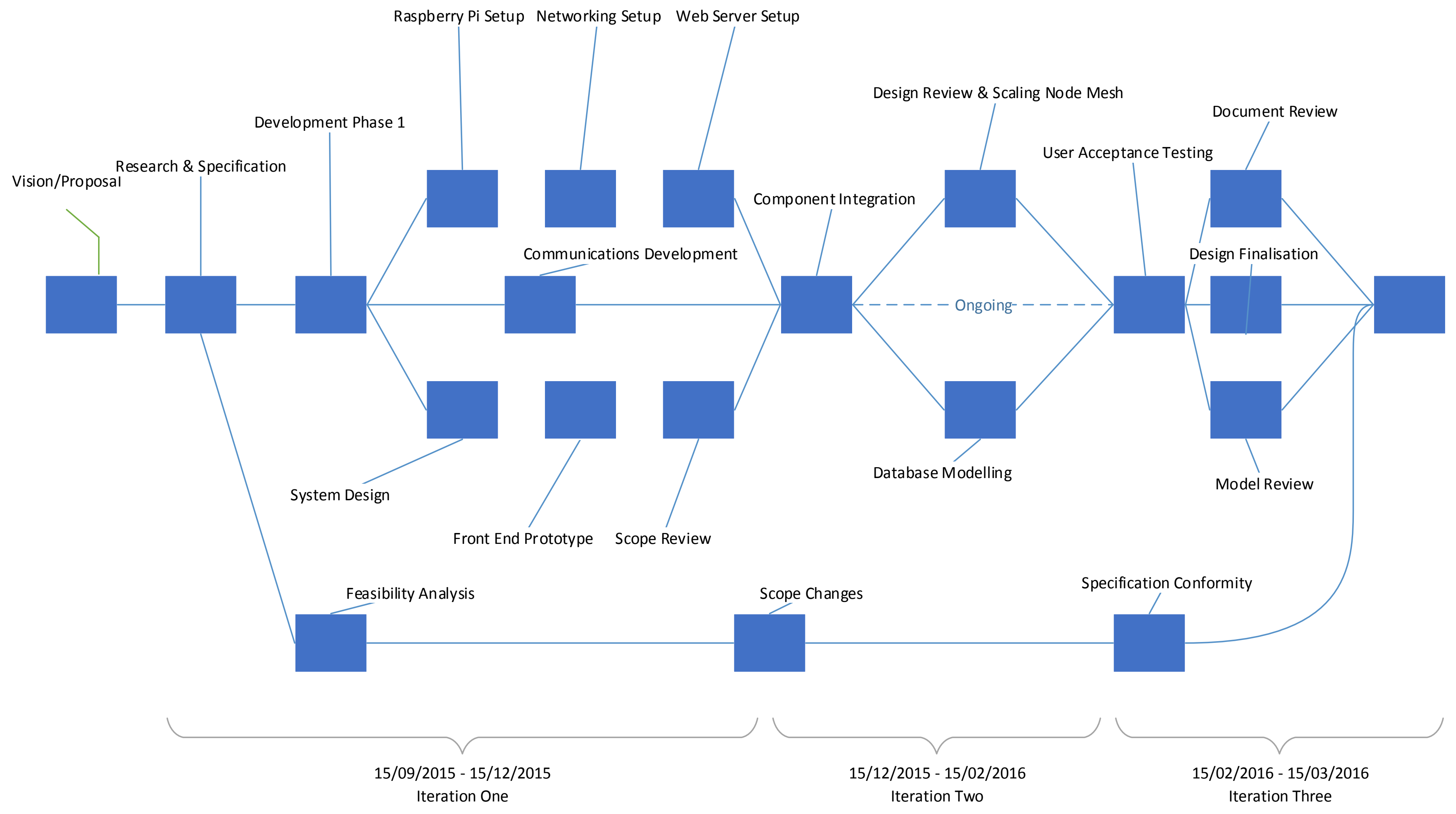
In order for the system to run and maintain a true sense of automation. Regular background operations must run constantly to ensure schedules are maintained, logs are successfully written, errors are handled, restarts and unscheduled downtime periods are recovered from for without hang up issues or downtime. Debian provides many services for this. However, it would be preferred for an all in one Python solution using cron like scheduling for interval and trigger based service operations. These services will handle temperature logs, motion sensing, user defined activation/deactivation schedules and system logging features.

5.3.4 Documentation and Showcase Webpage

Documentation is a key part of any project, big or small. During iteration two, a full functional specification, research document and design document are required. Additionally, a significant amount of a technical user document should be written by the end of this cycle to ensure usability of the system.

Also, in order to showcase the project, a demonstration website is required to give users a brief view of what the system can do and the target audience should be identified.

5.4 Project Workflow Map

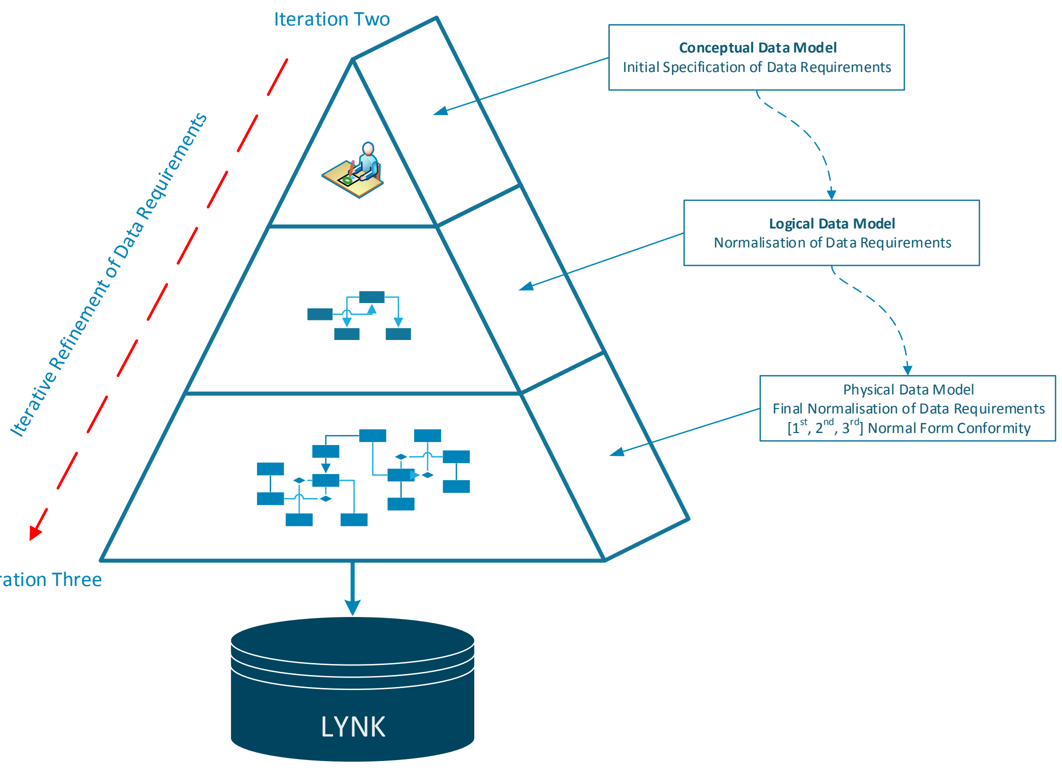


6. Data Requirements Specification

The following sections detail the design details of the underlying LYNK logical entity relationship model. Before any data modelling can commence, the initial functional data requirements have to be addressed. This data modelling processing is split into three layers and inherently require three modelling revisions. The structure to this process is broken into: conceptual modelling, logical modelling and finally the physical data modelling phase. As illustrated in the workflow map in functional specification, the key areas are present in all iterations and require attention at all stages.

- Iteration One [Development Phase 1] should involve the sourcing of data requirements. Users, Nodes, essentially all the objects the system will keep track of.
- Iteration Two [Data Modelling should focus heavily on the normalisation of the data model by further expanding previously identified data requirements into sub categories to obey first normal form.
- Iteration Three should focus heavily on finally refine the data requirements into 2nd and 3rd normal form for production usage.

This iterative data refinement process is illustrated below.


Iteration Two
Iterative Refinement of Data Requirements
Iteration Three
Conceptual Data Model
Initial Specification of Data Requirements
Logical Data Model
Normalisation of Data Requirements
Physical Data Model
Final Normalisation of Data Requirements
[1st, 2nd, 3rd] Normal Form Conformity
LYNK

7. References

[1] WeMo, “WeMo,” 1 October 2015. [Online]. Available: http://www.wemo.com/.

[2] Belkin, “Belkin WeMo Products,” 20 September 2015. [Online]. Available: http://www.belkin.com/us/Products/home-automation/c/wemo-home-automation/.

[3] L. W. R. PLC, “Light Wave RF,” Light Wave RF, 5 May 2015. [Online]. Available: http://lightwaverf.com/. [Accessed 2 October 2015].

[4] ADT, “ADT Security Specialists,” 2015 ADT LLC DBA ADT Security Services, 2015. [Online]. Available: http://www.adt.com/. [Accessed 1 October 2015].

[5] Philips, “Meet Hue,” Philips, 2015. [Online]. Available: http://www2.meethue.com/en-gb/about-hue/what-hue-does/. [Accessed 2 October 2015].

[6] Philips, “Philips Apple HomeKit,” Philips, 2015. [Online]. Available: http://www2.meethue.com/en-gb/friends-of-hue/apple-homekit/. [Accessed 2 October 2015].

[7] A. Fruit, “Raspberry Pi B+ Model,” Ada Fruit, 2015. [Online]. Available: https://www.adafruit.com/products/1914. [Accessed 5 September 2015].