CSE Senior Design

System Requirements Specification

Department of Computer Science and Engineering
The University of Texas at Arlington

Team: BetaHomes

Project: Z-Wave Smart Home

Team Members:
Nicholas Earwood
Benjamin Frank
Andrei Patapau
Aerina Shrestha
Santosh Upadhyay

Last Updated: 4/22/12 20:55

Table of Contents

Document Revision History

Revision Number	**Revision Date**	**Description**	**Rationale**
0.1	3/17/12	First Rough Draft	A rough draft is required.
0.2	3/18/12	First Rough Draft	A rough draft.
0.3	3/18/12	First Rough Draft	A rough draft.
0.4	3/18/12	First Rough Draft	A rough draft.
0.5	3/18/12	Final Rough Draft	A complete rough draft is required.
0.6	3/28/12	Professor Revision	Changes made based on Professor Comments
0.7	4/2/12	GTA Revision	Changes made based on GTA Comments
0.8	4/9/12	Sponsor/Peer Revision	Changes made based on Sponsor/Peer Comments
1.0	4/22/12	SRS Baseline	Passed Gate Review

List of Figures

List of Tables

1. Product Concept

This section gives the reader an upper-level understanding of the Z-wave Smart Home (ZSH) product. First, Z-wave is a next generation wireless ecosystem that lets home electronic devices communicate with each other and the user. It uses reliable low-power radio waves (908.42 MHz, U.S. standard) that travel through walls and floors to create a mesh network of communication. Z-wave control can be added to almost any electronic device such as kitchen appliances, window shades, thermostats, and home lighting. The ZSH product creates this communication network to allow the user control over their electronic devices without being near the device. The ZSH product is specifically designed to control a Blinds Control Unit but will also control other Z-wave enabled devices. Further description of the ZSH product will be provided in the rest of the document with the first two sections detailing the Purpose and Use along with the Intended Audience.

1.1 Purpose and Use

The purpose of the Z-wave Smart Home (ZSH) product is to control electronic devices in the home remotely and wirelessly. Using the ZSH product should make controlling electronic devices in the home more convenient for the user. The ZSH product is not a replacement for appliance controllers such as Television and DVD players but instead is a method for controlling the power on and off of such devices. Typical use of the ZSH product would be using a smart phone with Internet connection to logon to a web server. The web server then controls the Z-wave enabled devices in the House. The user is expected to use this method of access to directly control appliances or create schedules/rules for automated control.

1.2 Intended Audience

The Z-wave Smart Home (ZSH) product is designed to be a practical method of control for people who would enjoy greater control and ease of use of household appliances. This includes persons with appliances in normally hard to reach places. The ZSH product would also accommodate persons with disabilities as they can control frequently used appliances without being physically near said appliances. These appliances could include but are not limited to coffee makers and toasters. The ZSH device is web enabled for easy access through a smart phone or personal computer both of which the intended audience are required own for access to all the features of the ZSH product.

2. Product Description and Functional Overview

This section provides an overview of the Z-wave Smart Home (ZSH) product's physical specifications and functions. Features and Functions will describe the components used in the ZSH system and their function within the system. External Inputs and Outputs will define what commands the user can put into the ZSH system and how the system responds to the user. The Product Interface determines how the user interacts with the ZSH system and gives examples of what this interaction would look like.

2.1 Features and Functions

The Z-wave Smart Home (ZSH) product is designed to operate Z-wave enabled devices by allowing the user to control these devices remotely or via schedules/rules. ZSH is not a decision making product, the system cannot decide what it will and will not do. Control of the ZSH system is always in the hands of the user and the system will only operate within the parameters defined by the user. The ZSH design is a step toward the future of home independent automation but is still far from an Artificially Intelligent home. For product visual concept see figure 2-1.

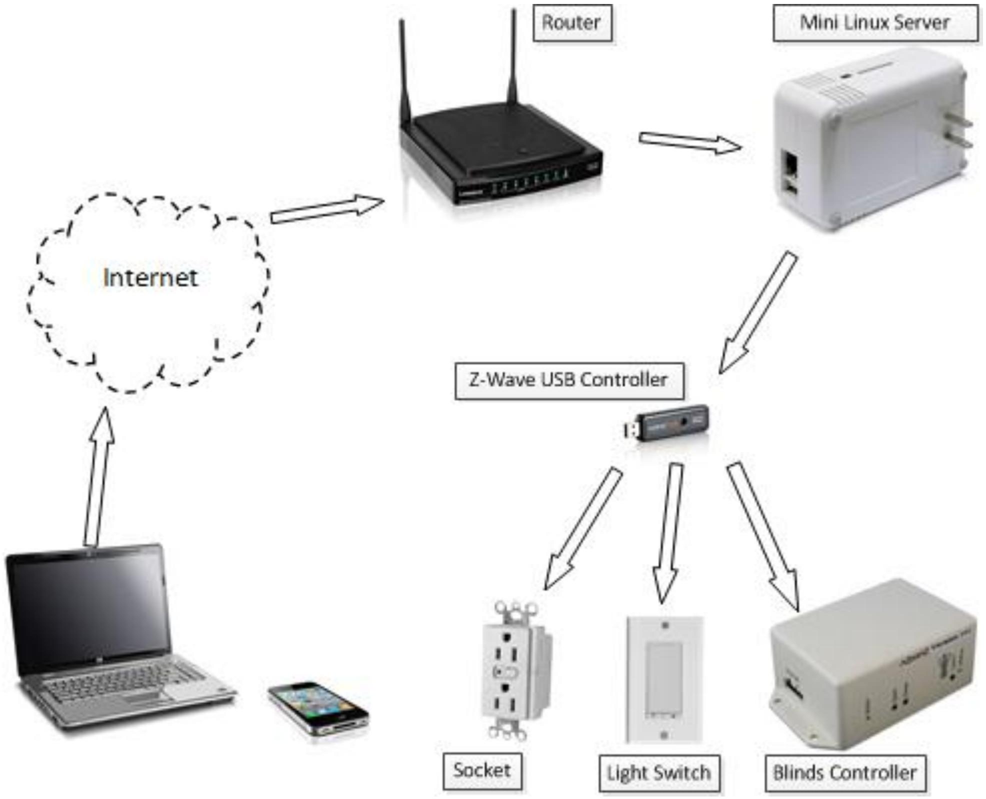


Figure 2-1: Product visual concept

The ZSH will feature/use the following components:

1) Internet – The Internet will be required at all times to operate the ZSH product remotely. It is required for communication between the smart phone controller and ZSH server.

2) Mini-Linux Server PC – The main component of the ZSH product. The server will communicate with Internet browsers as well as store schedules/rules as defined by the user. The server will command the Z-wave USB adapter. The server will host the web-interface.

3) Z-Wave USB adapter – Translates communications with all Z-wave enabled devices and the server within a minimum 17 meter radius. The adapter plugs into the server via USB port. The adapter is controlled by the server.

4) Web-Interface – The server will host the web-interface. The web-interface will be the main form of user interaction with the ZSH product. The web-interface will be database driven, allowing the user to store schedules and rules for the system to execute.

5) Router/Modem – Communication transition between the Internet and the ZSH server.

6) Z-wave modules – Controlled through the server/adapter. The modules in this ZSH product will consist of light switches, electrical sockets, and a Blinds Control Unit. These modules will communicate with each other and the adapter.

7) Blinds Control Unit (BCU) – Communicates with the adapter and other Z-wave modules. Used to open and close household blinds via changes in blind’s slat angles. BCU uses rechargeable batteries to operate. BCU uses solar panels to recharge batteries. BCU is wall mounted and designed to retrofit many blind types.

8) Database – The database is maintained on the server. The database stores rules and schedules created via user interaction with the web-interface.

9) User Interface (UI) – The UI will consist of Personal Computer and smart phone devices. The UI will require Internet access and the ability to communicate via Hyper Text Transfer Protocol. The UI will interact with the Web-Interface. This will be the main form of user control over the ZSH system.

Refer back to figure 2-1 for a visual product concept.

The ZSH product will consist of the server, adapter, Z-wave modules, and Blinds Control Unit (BCU). The server is a small wireless box that plugs directly into a wall outlet, preferably close to Z-wave enabled devices and the router. The adapter is a similar in look to a USB data stick, and uses a USB port on the server. For most intents and purposes the server and adapter can be thought of as a single component. The Z-wave modules can be any number of Z-wave enabled devices but the ZSH product will provide only a light switch, electrical socket, and BCU. The light switch and electrical socket would replace a standard light switch or electrical socket and should be installed according to the directions found with the purchase of those items. The BCU is a device that is no larger than 30cm x 30cm x 15cm and weigh no more than 1kg. The BCU is a wall mounted unit that the user would need to screw into the wall near the rod that controls the angle of the blind's slats. The rest of the components listed above are separate from the ZSH product but are still required for its use.

2.2 External Inputs and Outputs

Controller		
Name	**Description**	**Use**
Power	Constant power source	Required to operate at all times
Intranet Connection	The product must be connected to a home intranet	Required for configuration and management
Internet Connection	A consistent internet connection	Required for remote control
Z-Wave Connection	Reliable Z-Wave communications	Required for proper operation
Smart Phone	A device with a web browser	Required for remote control
Personal Computer	A device with a web browser	Required for configuration and management

Table 2-1: Controller External Inputs and Outputs

Blinds Control Unit		
Name	**Description**	**Use**
Power	Rechargeable D-cell batteries	Required to operate at all times
Compatible Blinds	Blinds unit that fits the ZSH product	Required for blinds manipulation
Z-Wave Connection	Reliable Z-Wave communications	Required for proper operation

Table 2-2: Blinds Control Unit External Inputs and Outputs

2.3 Product Interfaces

Most of the Z-wave Smart Home (ZSH) system is plug and play with one notable exception. The ZSH Web-Interface is the main way for the user to interact with the ZSH system. Access is obtained through either a Personal Computer or a smart phone with Internet access. The following pictures show how a user would add devices to their home using the ZSH Web-Interface. A written description will follow.

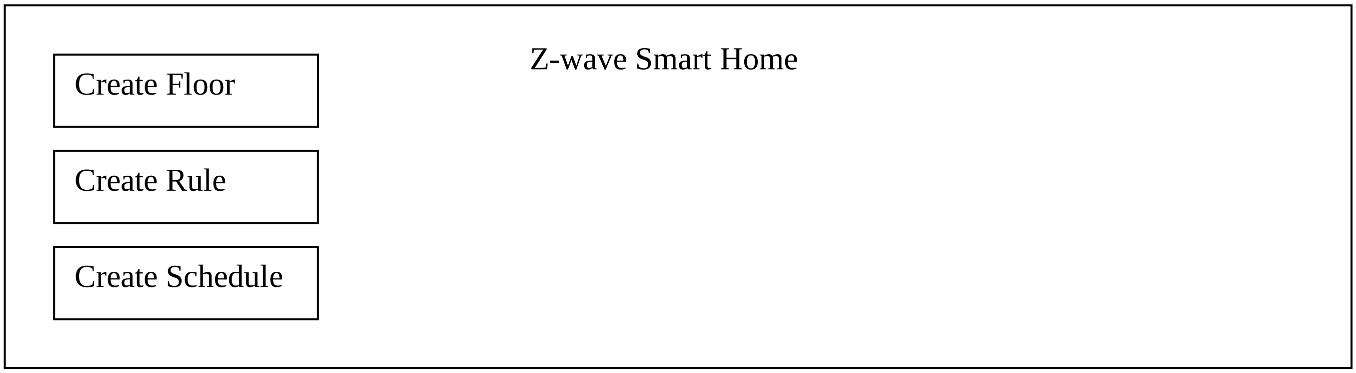


Figure 2-2: Main Screen

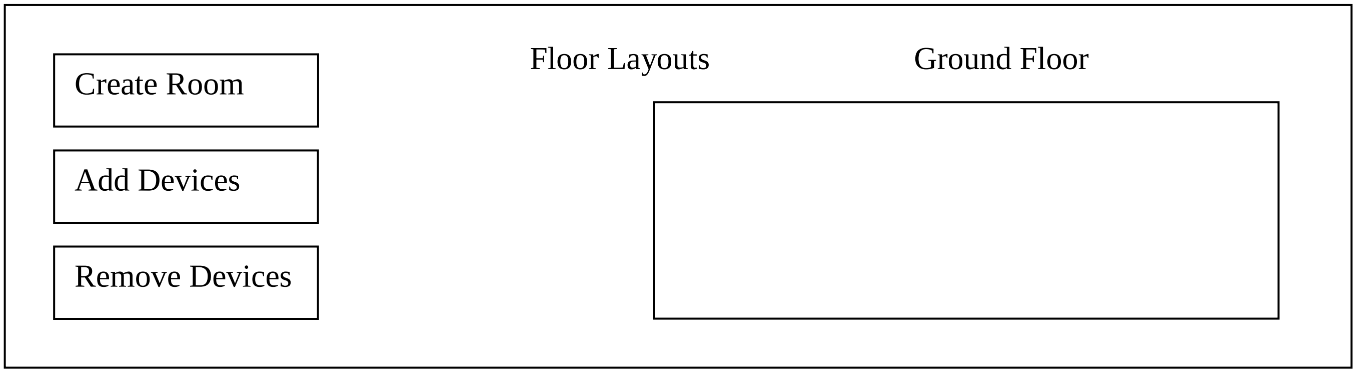


Figure 2-3: Floor layout

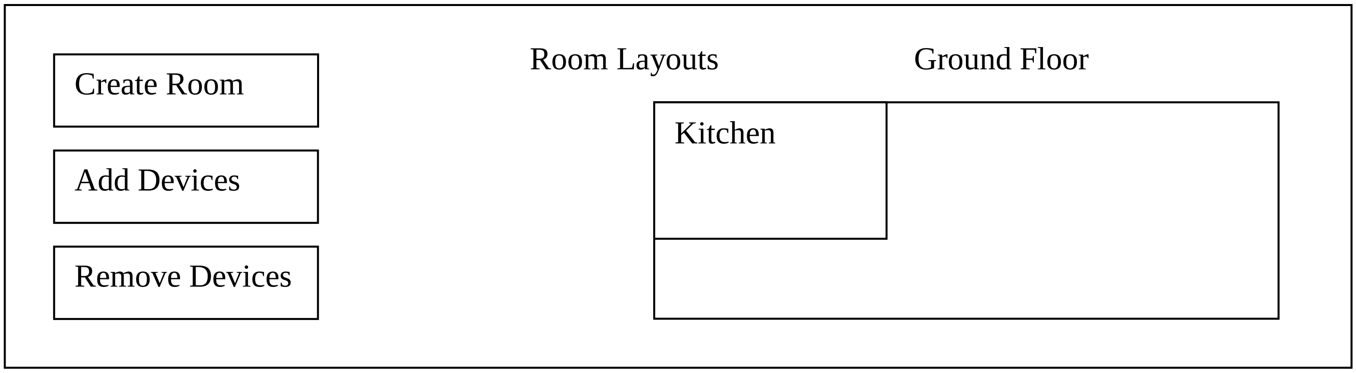


Figure 2-4: Room Layout

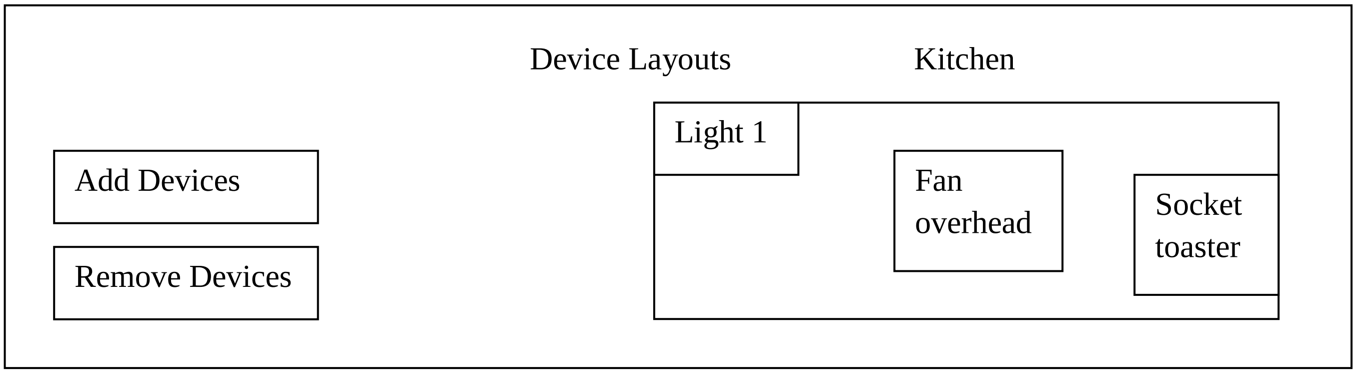


Figure 2-5: Device Addition

When the user accesses the Web-Interface they would see a screen much like Figure 2-2. Clicking "Create Floor" would add a floor layout to the screen and allow the user to name the floor. Then clicking on the floor layout will display the "Create Room" button that will allow the user to place a room within the floor by using a pointing device. This room can be clicked and named as well. Once the room is clicked the user can add or remove devices. Clicking "Add Devices" would provide a list of common Z-wave appliances to choose from. Once a device is selected it can be moved and placed within the room using a pointing device. These appliances can be named by the user for quick referencing. The user can also remove devices by clicking the "Remove Devices" button and selecting a device they wish to remove. The following figure 2-6 is a concept visualization of a floor plan when the previously described steps are taken.

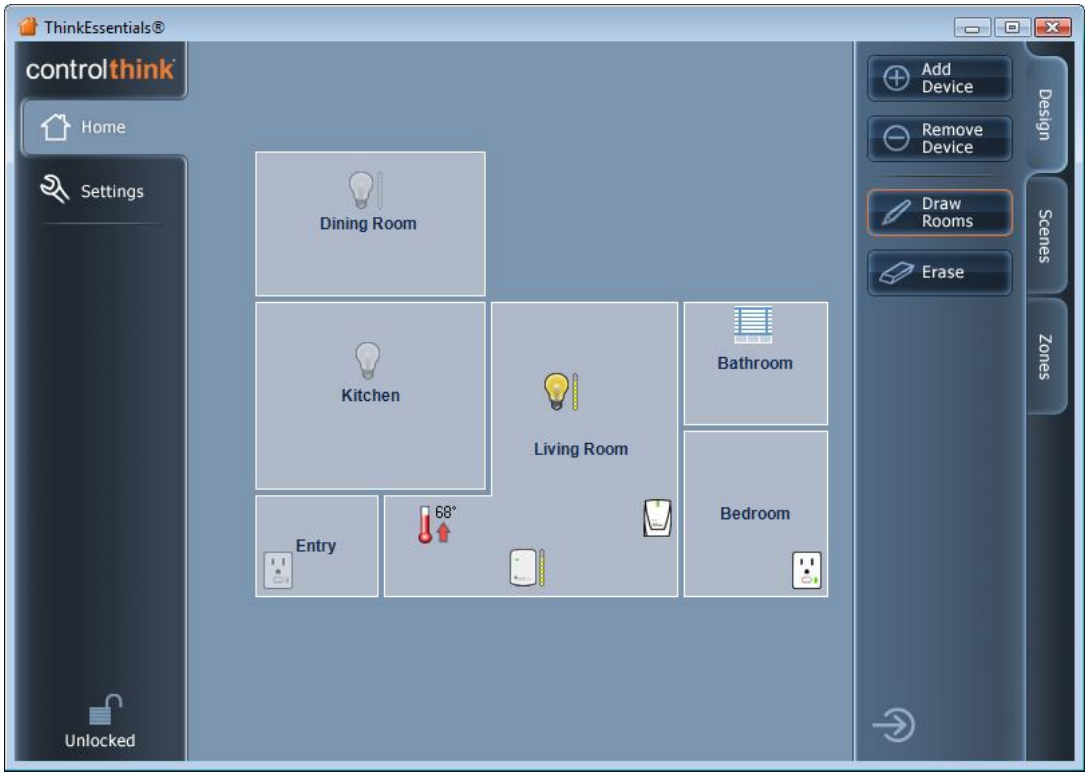
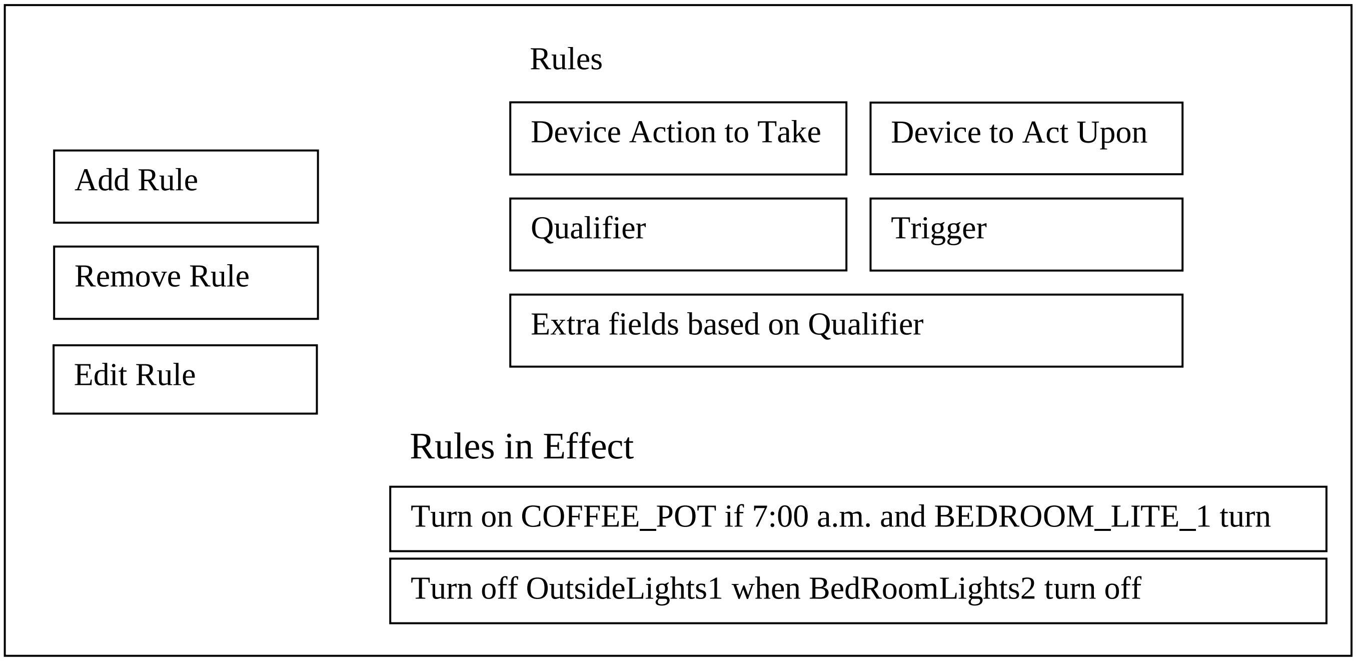


Figure 2-6: Visual concept of Web Interface

This next segment will describe the “Create Rule” interface. Pictures will be provided followed by a written description.



Turn on COFFEE_POT if 7:00 a.m. and BEDROOM_LITE_1 turn

Turn off OutsideLights1 when BedRoomLights2 turn off

Figure 2-7: Create Rules

When the user clicks “Add Rule” the rules window opens that looks much like Figure 2-7. The user can then select what they want the device to do and which device to perform the action on. The qualifier will be a dropdown list of options like “at”, “when”, and “if” which will change the triggering options and extra fields when selected. Based on the qualifier the trigger could be a time or the status of another device which causes the rule to execute. The Rules in Effect portion would show the rules already created by the user. These rules can be edited as required by the user by clicking on the “Edit Rule” option. “Remove Rule”, when clicked, will allow the user to select a rule to delete.

The “Create Schedule” window looks much like the Rules window and so only a written description will be given. When the user selects “Add Schedule” fields will appear allowing the user to select a device to perform an action on and what action to take. There will also be a field to allow the user to enter a time for the action to take place. The “Remove Schedule” button, when clicked, will allow the user to delete a schedule in the Schedules in Effect area. This Schedule in Effect portion will show the schedules already created by the user and can be edited as required by the user by clicking on the “Edit Schedule” option.

The following figures are visual concepts for the Web Interface lights and blinds schedule creation.

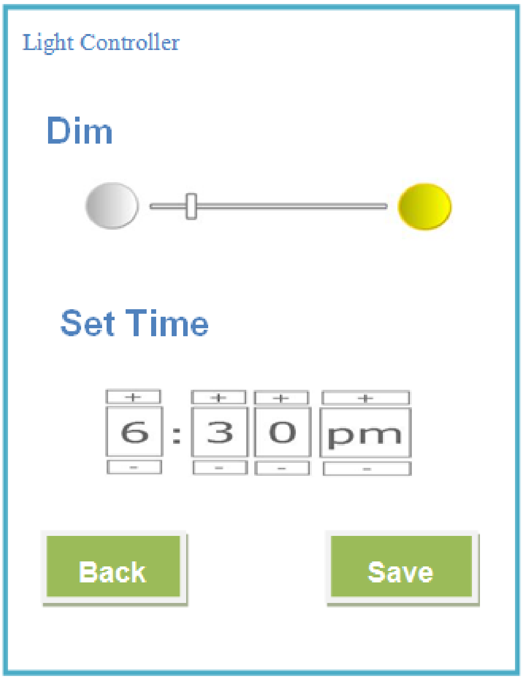


Figure 2-8: Lights schedule creator

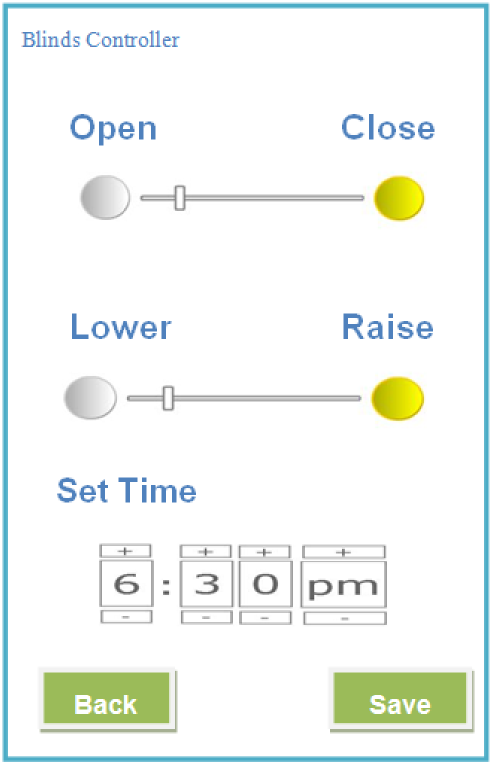


Figure 2-9: Blinds schedule creator

3. Customer Requirements

This section provides the basic functions, features and appearance of the product as requested by customer. The following requirements were gathered by team BetaHomes from sponsor of the project – Dr. Gergely Zaruba. This section specifies what the intended audience can/cannot do.

3.1 Web Interface

3.1.1 Description: The product shall be controlled using a “web interface”. The web interface shall work on any device that can process web pages and has access to the internet. Using the web interface, the user shall be able to setup, configure, and control all paired Z-Wave devices.

3.1.2 Source: Sponsor and Team BetaHomes.

3.1.3 Constraints: Smart phones will be limited in both screen space and technology use. Main setup and configuration may have to be solely through a PC web browser.

3.1.4 Standards: Must work with any of the late versions of MSIE, Firefox, Chrome, Opera, etc. Basically, any standards compliant browser, the actual standards are TBD.

3.1.5 Priority: High

3.2 Automated Device Control

3.2.1 Description: The controller shall be able to automatically control any Z-Wave enabled devices based on schedules and rules defined by the user.

3.2.2 Source: Sponsor

3.2.3 Constraints: Access to manufacturer device specifications.

3.2.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview. Individual device protocols as defined by their manufacturer.

3.2.5 Priority: High

3.3 Scheduling

3.3.1 Description: The server shall control individual and groups of devices using hourly, daily, weekly, or monthly schedules. These schedules can be created, modified, or deleted as per the user’s requirement.

3.3.2 Source: Sponsor

3.3.3 Constraints: Limit scheduling to the mentioned categories.

3.3.4 Standards: None

3.3.5 Priority: Moderate

3.4 Rules

3.4.1 Description: The server shall allow creation, modification, and deletion of rules for the operation of individual and groups of devices. For example, a rule could be setup so that an interior lamp becomes lit whenever the main overhead lights are off, so that a room is always lit. Another example could be that the exterior lights always turn on 10 minutes after dusk.

3.4.2 Source: Sponsor

3.4.3 Constraints: Rule-based scheduling/activation require unknown knowledge of computer rule systems. Advanced rule system may be relatively time-consuming to implement.

3.4.4 Standards: None

3.4.5 Priority: Moderate

3.5 Manual Device Control

3.5.1 Description: The user shall have the ability to manually control a Z-wave device such as turning on and off. This will override the devices schedule and/or rules until the next scheduled change or rule event triggers the device.

3.5.2 Source: Sponsor

3.5.3 Constraints: New devices may not supported these features.

3.5.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview.

3.5.5 Priority: High

3.6 Modularity

3.6.1 Description: The product shall be able to add and remove devices without causing failure to other devices.

3.6.2 Source: Sponsor and Team BetaHomes

3.6.3 Constraints: All devices can still be reached within the network.

3.6.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview.

3.6.5 Priority: High

3.7 Voice Control

3.7.1 Description: The product shall be able to recognize speech, translate spoken words into text, parse the text request and compare it with predefined action requests. If spoken request belongs to database of available actions – execute corresponding action.

3.7.2 Source: Sponsor

3.7.3 Constraints: The user shall have the ability to manually control a Z-wave device such as turning on and off. This will override the devices schedule and/or rules until the next scheduled change or rule event triggers the device.

3.7.4 Standards: N/A

3.7.5 Priority: Moderate

3.8 2D Map

3.8.1 Description: The Web Interface shall display a 2D map view of working range of Z-Wave modules.

3.8.2 Source: Sponsor

3.8.3 Constraints: Size of the screen displaying the map. The 2D map willl display a maximum of 25 rooms, 10 devices per room. A maximum of 255 devices may be configured, whether assigned to a room or not (5.5.3).

3.8.4 Standards: N/A

3.8.5 Priority: Moderate

3.9 Z-Wave Device Communication

3.9.1 Description: The system shall use Z-Wave standards for communication between devices.

3.9.2 Source: Sponsor and industry standard

3.9.3 Constraints: N/A

3.9.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview.

3.9.5 Priority: High

3.10 Z-Wave Controller Communication

3.10.1 Description: The system shall use Z-wave standards for communication between the controller and devices. The controller will use an internal or external Z-Wave adapter.

3.10.2 Source: Sponsor and industry standard

3.10.3 Constraints: The Z-Wave adapter must be compatible with our software and hardware and any 3rd party software we use.

3.10.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview.

3.10.5 Priority: High

3.11 Blinds Control Unit Slat Elevation

3.11.1 Description: The Blinds Control Unit (BCU) shall be able to lower and raise blind’s slats.

3.11.2 Source: Sponsor

3.11.3 Constraints: Force required for lowering or raising slats.

3.11.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview.

3.11.5 Priority: High

3.12 Blinds Control Unit Slat Rotation

3.12.1 Description: The Blinds Control Unit (BCU) shall be able to rotate blind’s slats.

3.12.2 Source: Sponsor

3.12.3 Constraints: N/A

3.12.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview.

3.12.5 Priority: High

3.13 Blinds Control Unit Solar Energy

3.12.1 Description: The Blinds Control Unit shall use solar energy to recharge its batteries.

3.12.2 Source: Sponsor

3.12.3 Constraints: Sponsor's definition of acceptable solar panel size is 6" x 6".

3.12.4 Standards: N/A

3.12.5 Priority: Moderate

3.14 Z-wave Light Switch

3.14.1 Description: The Z-wave Light Switch shall control lights with the commands on, off, and dim.

3.14.2 Source: Sponsor

3.14.3 Constraints: Access to manufacturer's device specifications.

3.14.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview.

3.14.5 Priority: Moderate

3.15 Blinds Control Unit Dimensions

3.15.1 Description: The Blinds Control Unit shall be no larger than 30cm x 30cm x 15cm.

3.15.2 Source: Sponsor

3.15.3 Constraints: Components' size and placement.

3.15.4 Standards: None

3.15.5 Priority: Moderate

3.16 Blinds Control Unit Weight

3.16.1 Description: The Blinds Control Unit shall weight no more than one kilogram.

3.16.2 Source: Sponsor

3.16.3 Constraints: Components' weight.

3.16.4 Standards: None

3.16.5 Priority: Moderate

3.17 Graphical User Interface (GUI)

3.17.1. Description: The product GUI shall have simple and meaningful instructions, icons and symbols.

3.17.2. Source: Sponsor

3.17.3. Constraints: None

3.17.4. Standards: None

3.17.5. Priority: High

3.18 Device Status Feedback

3.18.1. Description: The user interface shall provide a meaningful feedback on device change of status.

3.18.2. Source: BetaHomes

3.18.3. Constraints: Any form of signal interference while sending the feedback.

3.18.4. Standards: None

3.18.5. Priority: High

3.19 Blind Control Unit Manual Control

3.18.1. Description: The user shall be able to control the blinds manually, and the feedback will be sent to the server about the change in the position and elevation of the blind’s slat.

3.18.2. Source: BetaHomes, and Project Manager

3.18.3. Constraints: Use of excessive force while controlling the blinds manually may not send the right feedback.

3.18.4. Standards: None

3.18.5. Priority: Moderate

4. Packaging Requirements

This section defines what the product will consist of. When the user obtains the Z-wave Smart Home (ZSH) product the following requirements describe what parts the user will receive.

4.1 Z-wave Smart Home Server

4.1.1 Description: The product shall consist of one server.

4.1.2 Source: Sponsor

4.1.3 Constraints: N/A

4.1.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview.

4.1.5 Priority: Critical

4.2 Blinds Control Unit

4.2.1 Description: The product shall consist of one Blinds Control Unit.

4.2.2 Source: Sponsor

4.2.3 Constraints: None

4.2.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview.

4.2.5 Priority: Critical

4.3 Z-wave Smart Home Software

4.3.1 Description: The product shall have operational software installed on the server.

4.3.2 Source: Sponsor

4.3.3 Constraints: None

4.3.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview.

4.3.5 Priority: Critical

4.4 Z-wave Smart Home Backup

4.4.1 Description: BetaHomes shall provide a CD with backup software for mini Linux PC.

4.4.2 Source: Sponsor

4.4.3 Constraints: None

4.4.4 Standards: None

4.4.5 Priority: High

4.5 Z-wave Light Switch Module

4.5.1 Description: The product shall consist of one Z-wave Light Switch Module.

4.5.2 Source: Sponsor

4.5.3 Constraints: None

4.5.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview.

4.5.5 Priority: Critical

4.6 Z-wave Electrical Socket Module

4.6.1 Description: The product shall consist of one Z-wave Electrical Socket Module.

4.6.2 Source: Sponsor

4.6.3 Constraints: None

4.6.4 Standards: Z-Wave protocol as defined in Zensys Software Design Specification: Z-wave Protocol Overview.

4.6.5 Priority: Critical

5. Performance Requirements

The quality of any product is determined by its performance. The follow requirements will describe the performance characteristics of the Z-wave Smart Home (ZSH) product.

5.1 Remote Server Response Time

5.1.1 Description: The server shall respond to remote user input within two seconds. The user will receive either an acknowledgement or response that their input has been received when using the device's web interface.

5.1.2 Source: Team BetaHomes

5.1.3 Constraints: Reliable internet connection. Network latency not included in response time calculations, as they are impossible to predict.

5.1.4 Standards: Internet protocols as defined by IEEE.

5.1.5 Priority: Moderate

5.2 Blinds Control Unit Battery Replacement

5.2.1 Description: The Blinds Control Unit shall operate for at least one year without requiring battery replacement.

5.2.2 Source: BetaHomes

5.2.3 Constraints: Excessive use.

5.2.4 Standards: None

5.2.5 Priority: High

5.3 Blinds Control Unit Battery Recharging

5.3.1 Description: The Blinds Control Unit shall recharge its batteries to full capacity with four hours of direct sunlight.

5.3.2 Source: BetaHomes

5.3.3 Constraints: Sufficient Sunlight. Excessive use.

5.3.4 Standards: None

5.3.5 Priority: Moderate

5.4 Web Interface Portability

5.4.1 Description: The Z-wave Smart Home (ZSH) Web Interface shall operate on Internet Explorer and Firefox web browsers.

5.4.2 Source: Team BetaHomes

5.4.3 Constraints: Reliable Internet connection. Future browser versions remain backwards compatible.

5.4.4 Standards: None

5.4.5 Priority: High

5.5 Adding New Devices

5.5.1 Description: The Z-wave Smart Home server shall add a new device within one minute upon starting the association process.

5.5.2 Source: Team BetaHomes

5.5.3 Constraints: The devices may not be detected if it is out of range. If not detected, it won’t be able to relay with other Z-wave enable devices either. The ZSH controller will allow up to 255 devices to be added as a maximum.

5.5.4 Standards: None

5.5.5 Priority: High

5.6 Blinds Control Unit Motor Usage

5.6.1 Description: The Blinds Control Unit motor shall not draw more than half of the batteries’ current capacity per use.

5.6.2 Source: Team BetaHomes

5.6.3 Constraints: Battery capacity.

5.6.4 Standards: None

5.6.5 Priority: Moderate

5.7 System Range

5.7.1 Description: All Z-wave enabled devices shall be within 15 meter radius of another Z-wave enabled device or the Z-wave Smart Home server.

5.7.2 **Source:** Team BetaHomes

5.7.3 **Constraints:** Any form of signal interference.

5.7.4 **Standards:** None

5.7.5 **Priority:** Moderate

6. Safety Requirements

Safety requirements are of very high priority because this product is targeted to ease the use of household appliances for everyday use. These requirements are critical to protect the user against any injury. Thus, they need to be addressed to the user by providing a clear and easily understandable user manual. Also, the product should include warning labels that can be referred to at any time.

6.1 Insulated Wires

6.1.1 Description: The product shall not have exposed electric wires.

6.1.2 Source: Team BetaHomes

6.1.3 Constraints: N/A

6.1.4 Standards: N/A

6.1.5 Priority: High

6.2 Hanging Wires

6.2.1 Description: The product shall not have hanging wires as it can lead to strangulation or electric shock.

6.2.2 Source: Team BetaHomes

6.2.3 Constraints: The blind unit's solar powered charger may require wires to be routed from the unit to the solar cells.

6.2.4 Standards: None

6.2.5 Priority: High

6.3 Sharp Edges

6.3.1 Description: The product shall not have any sharp edges. Sharp edges are defined as an edge that can cut skin when applying less than one psi.

6.3.2 Source: Team BetaHomes

6.3.3 Constraints: None

6.3.4 Standards: None

6.3.5 Priority: Moderate

7. Maintenance and Support Requirements

Maintenance requirements are those that facilitate maintenance of the system and assist in repairs upon fault detection. This section includes requirements for future expansion of the system and adding additional functionality.
Team BetaHomes shall not be responsible for supporting the Smart Home Controller package after August 7, 2012. The requirements listed below will be used to indicate the level of support provided by Team BetaHomes.

7.1 Product shall be designed to be modular and extensible.

7.1.1 Description: The team shall architect, design, and build the product on a modular basis. Both software and hardware components must be replaceable to enable future maintainers to add functionality or maintain update individual parts.

7.1.2 Source: Team BetaHomes

7.1.3 Constraints: N/A

7.1.4 Standards: N/A

7.1.5 Priority: High

7.2 Product shall come with design documentation and User's Manual.

7.2.1 Description: As a part of development process, the team shall provide detailed design documentation and user's manual. These documents will enable future maintainers to provide a better service and understanding of a product concept.

7.2.2 Source: Team BetaHomes

7.2.3 Constraints: N/A

7.2.4 Standards: N/A

7.2.5 Priority: High

7.3 Software maintenance shall be performed remotely.

7.3.1 Description: Software maintenance shall be done remotely, however team BetaHomes reserves the right to perform software maintenance in the field if necessary.

7.3.2 Source: Team BetaHomes

7.3.3 **Constraints:** N/A

7.3.4 **Standards:** N/A

7.3.5 **Priority:** Moderate

7.4 Hardware maintenance shall be done remotely.

7.4.1 Description: Hardware maintenance shall be done remotely due to product modularity and independence of each component, however team BetaHomes reserves the right to perform hardware maintenance in the field if necessary.

7.4.2 Source: Team BetaHomes

7.4.3 Constraints: N/A

7.4.4 Standards: N/A

7.4.5 Priority: Moderate

7.5 System shall provide error log.

7.5.1 Description: To enable ease of maintenance the software shall be able to produce an error log with detailed description of each error encountered. Allowable size of the log may vary according to user specification.

7.5.2 Source: Team BetaHomes

7.5.3 Constraints: Error log memory size. Excessive error logging.

7.5.4 Standards: N/A

7.5.5 Priority: Moderate.

7.6 The team shall not provide any support or maintenance after project completion date.

7.6.1 Description: Team BetaHomes will not be responsible for product support/maintenance after project completion date (August 2012).

7.6.2 Source: Team BetaHomes

7.6.3 Constraints: Sufficient documentation.

7.6.4 Standards: N/A

7.6.5 Priority: High

8. Other Requirements

These requirements do not fit into other areas based on the source and nature of their criteria. This section is designed to be a catch all for such requirements and as such does not have an overall theme. Individual requirements often are not related to each other but to aid greater comprehension the requirements contained within this section will be grouped into related areas as much as possible.

8.1 Server Power On: Networking

8.1.1 Description: The server shall, upon powering on, automatically attempt to obtain a network address, and broadcast it's hostname to the network.

8.1.2 Source: Team BetaHomes

8.1.3 Constraints: N/A

8.1.4 Standards: N/A

8.1.5 Priority: High

8.2 Server Power On: Software

8.2.1 Description: The server shall, upon powering on, automatically start the Z-wave Smart Home software package. The web server, and all application software will start automatically and be ready for use within 5 minutes.

8.2.2 Source: Team BetaHomes

8.2.3 Constraints: Any errors during device boot up may hinder or prevent system start up.

8.2.4 Standards: N/A

8.2.5 Priority: High

8.3 Server Backup

8.3.1 Description: The server shall allow the user to backup the system. This includes rules, schedules, devices, and all configuration data.

8.3.2 Source: Sponsor

8.3.3 Constraints: Space on backup medium.

8.3.4 Priority: Low

8.4 Server First Start

8.4.1 Description: If the server does not have any users defined, it shall create a administrator user and password and display the credentials to the user. The user may then log in using those credentials. The user, after logging in, should be able to change their password.

8.4.2 Source: Sponsor

8.4.3 Constraints: None

8.4.4 Priority: High

9. Acceptance Criteria

Acceptance requirements are the requirements that must be completed for the project to be accepted as complete. These requirements include the top priorities of the project which are derived from the critical requirements and would affect the functionality of the product if not taken into consideration.

9.1 Verify the product is web based.

9.1.1 Requirement(s) addressed: Requirement 3.1 Web Interface: The product shall be controlled using a “web interface”.

9.1.2 Verification Procedure: We will verify by interacting with the system using web based application.

9.2 Verify the controller is Z-wave enabled

9.2.1 Requirement(s) addressed: Requirement 3.2 Z-Wave Device Controls: The controller shall be able to add, remove, control and configure any Z-Wave enabled device.

9.2.2 Verification Procedure: We will verify that the controller detects and displays all Z-Wave devices under a certain range when it scans for the compatible devices. We will also verify that the devices respond to the commands correctly by visual inspection.

9.3 Verify the system is modular

9.3.1 Requirement(s) addressed: Requirement 3.6: The product shall be able to add and remove devices without causing failure to other devices.

9.3.2 Verification Procedure: We will verify the system is modular by making sure that adding or deleting a z-wave module does not affect the functionality of any other modules and devices that are already in the system.

9.4 Verify the product has no sharp edges

9.4.1 Requirement(s) addressed: Requirement 6.3: the product shall have no Sharp Edges.

9.4.2 Verification Procedure: We will inspect to make sure that the product has no sharp edges that can cause potential harm to the user.

9.5 Verify the system is properly insulated

9.5.1 Requirement(s) addressed: Requirement 6.1: The product shall have properly Insulated Wires.

9.5.2 Verification Procedure: We will verify that the wires are not exposed and properly insulated.

9.6 Verify the system has no hanging wires

9.6.1 Requirement(s) addressed: Requirement 6.2: The product shall not have any wires that are hanging.

9.6.2 Verification Procedure: We will inspect to make sure that the product has no hanging wires that can cause potential harm to the user.

9.7 Verify that the modules are z-wave enabled

9.7.1 Requirement(s) addressed: Requirement 3.11: The Blinds module shall be controlled by z-wave. Requirement 3.12: The Light module shall be controlled by z-wave.

9.7.2 Verification Procedure: We will inspect to make sure that the modules are controlled by the z-wave controller by making sure they respond correctly to the commands of the controller.

9.8 Verify the application has a meaningful GUI

9.8.1 Requirement(s) addressed: Requirement 3.14: The application shall have simple and meaningful GUI.

9.8.2 Verification Procedure: The sponsor shall approve of the GUI after a visual inspection. He will also verify that the instructions, symbols and icons used in the GUI are easy to follow and meaningful.

10. Use Cases

Use case is a list of steps that defines interactions between an actor and a system, to achieve a goal. Thus, use case is a business process that starts and ends with a user. Actor in our case is the user and the system can be the Z-Wave system, the web/ mobile application or the blinds. TUCBW stands for the use case begins with and TUCEW stands for the use case ends with.

10.1 Log in

10.1.1 Scenario: The user logs into the web application.

10.1.2 Actor(s): User.

10.1.3 TUCBW: The user enters the user id and password.

10.1.4 TUCEW: The user sees the welcome page or an error message saying that the log in failed.

10.2 Search Devices

10.2.1 Scenario: The user searches for Z-wave devices.

10.2.2 Actor(s): User.

10.2.3 TUCBW: The user hits on the search button.

10.2.4 TUCEW: The user sees the list of available z-wave devices.

10.3 Add Devices

10.3.1 Scenario: The user adds a Z-wave device.

10.3.2 Actor(s): User.

10.3.3 TUCBW: The user selects the z-wave device he/she wants to add.

10.3.4 TUCEW: The user sees “successfully added” message.

10.4 Remove Devices

10.4.1 Scenario: The user removes a z-wave device from the system.

10.4.2 Actor(s): User.

10.4.3 TUCBW: The user selects remove z-wave device option.

10.4.4 TUCEW: The user sees the "successfully removed" message.

10.5 Add Schedule

10.5.1 Scenario: The user adds schedule as per his requirement.

10.5.2 Actor(s): User.

10.5.3 TUCBW: The user clicks on the add schedules option.

10.5.4 TUCEW: The user sees "added successfully" message.

10.6 Remove Schedule

10.6.1 Scenario: The user removes schedule as per his requirement.

10.6.2 Actor(s): User.

10.6.3 TUCBW: The user clicks on the remove schedules option.

10.6.4 TUCEW: The user sees "removed successfully" message.

10.7 Edit Schedule

10.7.1 Scenario: The user edits schedule as per his requirement.

10.7.2 Actor(s): User.

10.7.3 TUCBW: The user clicks on the edit schedules option.

10.7.4 TUCEW: The user sees "edited successfully" message.

10.8 Add Rule

10.8.1 Scenario: The user adds rule as per his requirement.

10.8.2 Actor(s): User.

10.8.3 TUCBW: The user clicks on the add rules option.

10.8.4 TUCEW: The user sees "added successfully" message.

10.9 Remove Rule

10.9.1 Scenario: The user removes a rule as per his requirement.

10.9.2 Actor(s): User.

10.9.3 TUCBW: The user clicks on the remove rules option.

10.9.4 TUCEW: The user sees “removed successfully” message.

10.10 Edit Rule

10.10.1 Scenario: The user edits rule as per his requirement.

10.10.2 Actor(s): User.

10.10.3 TUCBW: The user clicks on the edit rules option.

10.10.4 TUCEW: The user sees “edited successfully” message.

10.11 Log out

10.11.1 Scenario: The user logs out from the system.

10.11.2 Actor(s): User.

10.11.3 TUCBW: The user clicks on the log off button.

10.11.4 TUCEW: The user sees the “log off successful” message.

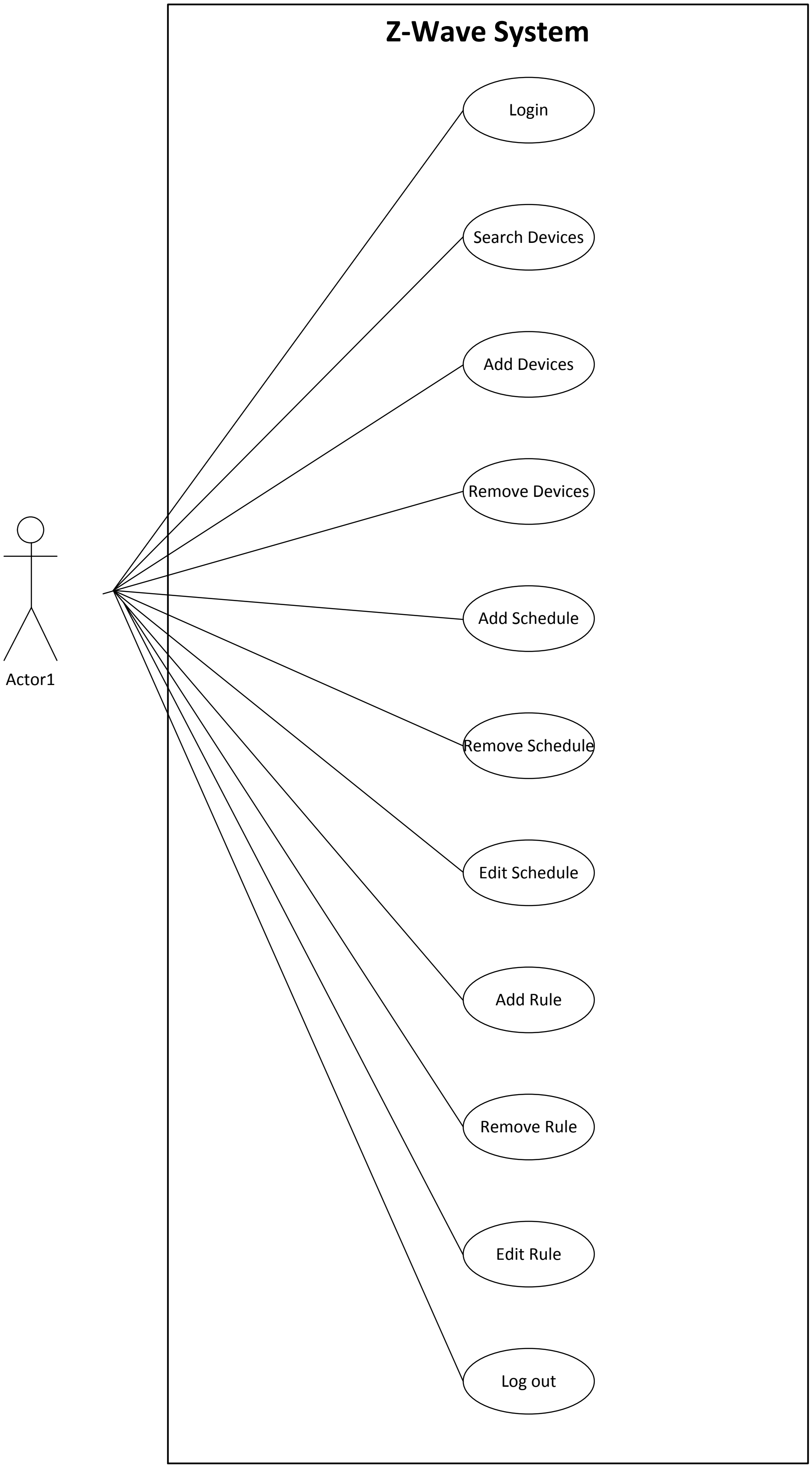

Z-Wave System
Login
Search Devices
Add Devices
Remove Devices
Add Schedule
Remove Schedule
Edit Schedule
Add Rule
Remove Rule
Edit Rule
Log out
Actor1

Figure 10-1: General use cases

10.12 Rotate Blinds

10.12.1 Scenario: The user rotates the blinds.

10.12.2 Actor(s): User.

10.12.3 TUCBW: The user sends in commands to rotate the blinds.

10.12.4 TUCEW: The user sees the blind rotate.

10.13 Lower/Raise Blinds

10.13.1 Scenario: The user lowers/raises the blinds.

10.13.2 Actor(s): User.

10.13.3 TUCBW: The user sends in commands to lower/raise the blinds.

10.13.4 TUCEW: The user sees the blind lower or raise.

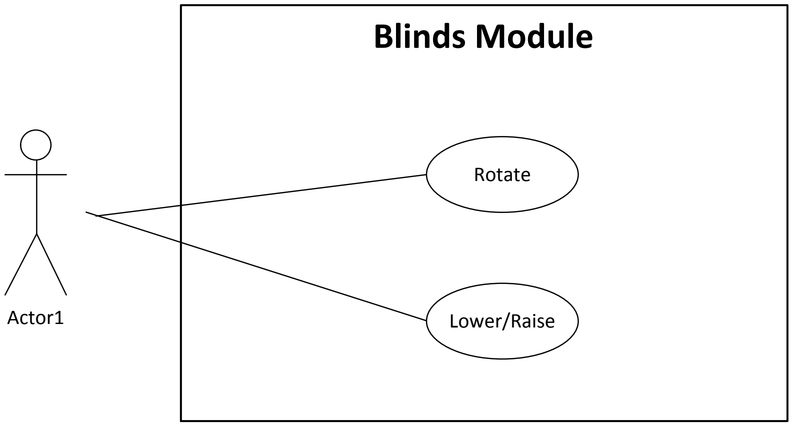


Figure 10-2 Use Case: Blinds Module

10.14 Dim Lights

10.14.1 Scenario: The user dims the lights.

10.14.2 Actor(s): User.

10.14.3 TUCBW: The user sends in commands to dim the lights.

10.14.4 TUCEW: The user sees the lights dim.

10.15 Turn Lights on

10.15.1 Scenario: The user turns the lights on.

10.15.2 Actor(s): User.

10.15.3 TUCBW: The user sends in commands to turn on the lights.

10.15.4 TUCEW: The user sees the lights turn on.

10.16 Turn Lights off

10.16.1 Scenario: The user turns the lights off.

10.16.2 Actor(s): User.

10.16.3 TUCBW: The user sends in commands to turn off the lights.

10.16.4 TUCEW: The user sees the lights off.

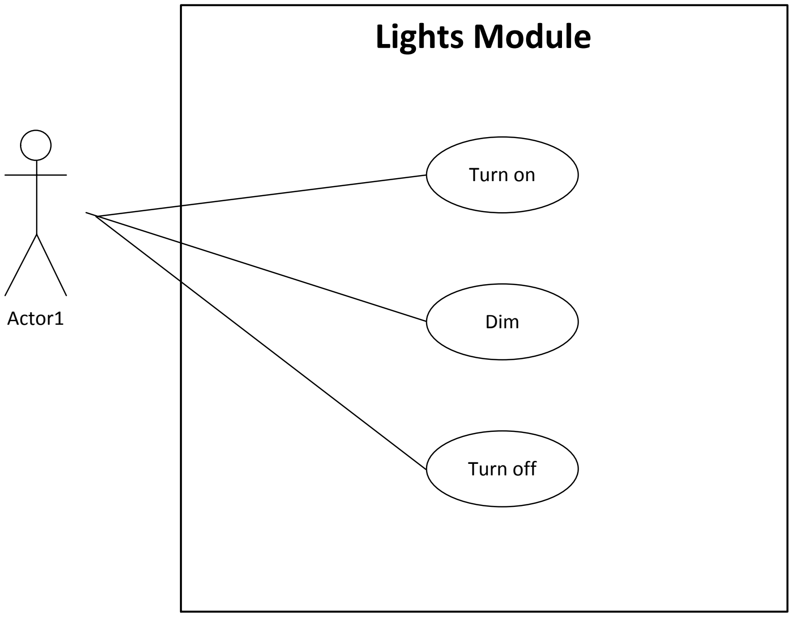


Figure 10-3 Use Case: Light Module

11. Feasibility Assessment

This section analyzes the feasibility of Z-wave Smart Home project, based on the following six components: scope analysis, research completed/remaining; technical analysis; cost analysis, resource analysis; and schedule analysis. The detail design of the project has not begun yet, and this assessment is solely based on team's knowledge, research, and critical analysis of each section herein. We are aware that it is a really critical section of the project as it would help us to determine the completion of this project on time and within the allotted resources.

11.1 Scope Analysis

There are many high-priority requirements, but meeting each one is feasible. Our team has a strong knowledge of programming and web development. Additionally, the blinds module is a separate unit that can be developed without being dependant on the project's z-wave controller. That is, if it is a Z-Wave device we can test it with off-the-shelf controller software. This will allow simultaneous development of both parts of the system. There are examples of Z-Wave controller software APIs and each discrete task (web server, web server and custom application interaction (e.g. CGI, etc.), FPGA programming, rule-based systems, mobile development, and so on) have strong representations in the market and online. The main bulk of the work to be done is integrating all these individual systems and protocols into a self-contained system.

Based on the current scope of the project the conclusion is that the Z-wave Smart Home product is feasible within the time and budget constraints.

11.2 Research

For this project, our first research job was to be familiar with Z-Wave technology. It was a new term for us, so we researched on what Z-wave actually is, and how it used. Then, we looked for the prices of the off-shelf Z-wave devices that were related to our project. There are a few products that are almost identical to what our project will do. This gives Team BetaHomes a resource to utilize during design considerations.

Since one of our requirements is to build Z-wave enabled blinds, we researched the mechanism of simple blinds, the usage of servo motors to control the slats, and transforming regular blinds to electronically controlled, Z-Wave enabled units. We found online communities where other people have modified devices for wireless or electronic control. There are also many online communities dedicated to just Z-Wave technology. Some are just dedicated to DIY Z-Wave homes, and they have a lot of information that pertains to our project.

One of the draw-backs we learned is that the Z-Wave protocol is not an open protocol. It is proprietary and the open-source and DIY communities have reverse engineered the protocol. This may hinder development of our controller and blinds.

11.3 Technical Analysis

The Smart Home Controller system will consist of 5 main parts: Z-Wave adapter, GNU/Linux server, Z-Wave modules, and the web application. The LAN and internet connection will be provided by the customer. Z-Wave adapter shall be used to communicate with Z-Wave enabled devices and provide network physical communication layer.

The Linux server is a miniature, inexpensive version of Linux PC which will communicate with mobile/PC applications accessible from the web and Z-Wave application installed on the server. The Z-Wave server software shall process web requests from the physical network layer and pass those to the Z-Wave controller software which will interact with the Z-Wave adapter, which in turn shall communicate with modules.

All of the above technical aspects of our project require quite a sophisticated knowledge of signal processing, FPGA programming, web development and circuitry design (blinds module implementation). Fortunately, the technology that we are dealing with been around for a decent amount of time and we have enough learning material available on the web to develop the product. However, the inventors of Z-Wave (ZenSys) do not support open-source development, and also, the manufactures of the various Z-Wave controllers do not provide details of their protocols. This means that development of open source software currently problematic, but yet possible because of broad variety of enthusiastic Z-Wave developers who share their achievements with public.

11.4 Cost Analysis

The Z-wave Smart Home (ZSH) system was purposefully designed to be low cost so that it would be affordable to a wider audience. Using this design decision the component selection process aimed for the lowest cost parts that are commercially available while still delivering a quality product. Based on a preliminary assessment of the parts required to complete the ZSH product the conclusion is that this project will be within an $800 budget.

These results are based on the following assumptions:

All manual labor is provided for free.
A Linux server costs approximately $80
A Z-wave USB adapter costs approximately $40
A solar cell costs approximately $20
Two rechargeable D batteries cost approximately $20

Two Servos cost approximately	$40
A 10 pack of gears costs approximately	$30
A light switch costs approximately	$60
An electrical socket costs approximately	$40
Blinds cost approximately	$50

The total cost of the project is approximately $400. Based on this cost approximation, the project is feasible from a cost perspective.

11.5 Resource Analysis

Team Beta Homes consists of two Computer Engineering students, two Computer Science students, and one Software Engineering student. The team has really good programming skills, especially in Java, C, and C++. Two members of the team also have knowledge of microcontroller operation and programming. However, the team lacks the required real life knowledge in the Design of the Project, but since it is being covered in our lectures, we will have more knowledge on it as the semester proceeds. The team is well balanced depending upon the amount of hardware and software requirements of this product. The product also demands lot of interfacing between hardware and software, and good networking knowledge. Since one of team members is taking network class, it will take care of our networking issues. And our team is going to do more research on how to interface hardware and software to make it a complete product. We will have to build our own Z-wave enabled blinds, integrate it with other hardware devices like Z-wave enabled light switches, and dimmable lights, and run it on the Smart Phones, or iPhone platform. Our computer engineers are researching on required hardware knowledge like basic mechanism of blinds, making it Z-wave enabled, and integrating it with other Z-wave devices. The computer engineers will also address the mechanical design considerations required to implement the Blinds Control Unit via research and consulting with outside sources such as the project sponsor or other teams. Similarly, other members of the team are also researching on the open source software and it’s utilization to control these Z-wave devices. Thus all the members of the Team BetaHomes know what they are doing, and what they have to do to complete this project. Hence, in terms of resources, this project is feasible.

11.6 Schedule Analysis

We are using function point estimation to estimate the duration for the project. The function point estimation is based on the number of Inputs, outputs, inquires, logical internal files and external logical files.

These function points are by multiplied with respective function-point multipliers as discussed in Table 8-2 of Steve McConnell's Rapid Development and added together to calculate the unadjusted function-point.

Program Characteristics	Low Complexity	Mid Complexity	High Complexity
Number of Input	3 * 3 = 9	1*4 = 4	0*6=0
Number of Output	4*4 =16	0*5=0	0*7=0
Inquiries	4*3 = 12	2*4=8	0*6=0
Logical Internal File	3 *7 = 21	0*10=0	1*15=15
External Interface File	2 *5 = 10	2*7=14	3*10=30
			UAF=139

Table 11-1: Unadjusted Function Point Calculation

The next step is calculating Jones' First-Order Estimate influence multipliers as shown in Table 11-2.

Data Communication	1
Distributed Data Processing	2
Performance	2
Heavily Used Configuration	1
Transaction Rate	2
Online Data Entry	2
End-User Efficiency	2
Online Update	2
Complex Processing	3
Re-usability	1
Installation Ease	2
Operational Ease	1
Multiple Site	1
Facilitate Change	1
Total	23

Table 11-2: General System Characteristics

Here, Value adjustment factor is calculated as follows:

VAF=(0.65+(0.01×23))=.88

The function point is calculated as follows:

FP=UAF×VAF=139×.88=122.32

We then use McConnell's Table 8-7 to calculate the best, average, and worst-case scenarios as shown in Table 11-3. Here, we use the exponents related to Systems development organization taking into regards the nature of our project and calculate an estimate of calendar months.

Worst Case	Average Case	Best Case
$122.32^{.48}$	$122.32^{.45}$	$122.32^{.43}$
10.046 calendar month	8.69 calendar month	7.899 calendar month
$(10.046/3)^3$=37.55 man months	$(8.69/3)^3$=24.30 man months	$(7.899\ /3)^3$=18.25 man months
37.55/5=7.51	24.30 /5=4.86	18.25 /5=3.65

Table 11-4: Man month calculation

Since we have 5 members in the team it could take us up to 7.5(worst case) to complete the project. But if we consider the average case, which is 24.3 man months it will take us about 5 months to complete the project. Thus, it is feasible for us to complete the project August 2012.

12. Future Items

This section enlists the Customer Requirements that team Beta Homes talked about and would like to make it available in the final product. But due to lots of feasibility issues/constraints, these requirements would not be available in first prototype of this product.

12. 1 Customer Requirement 3.2: The unit shall be designed to function without Internet connection

12.1.1 Requirement Description: The unit shall function properly within the household without Internet connection. Only a wireless home network will be required. The Z-wave enabled devices would be able to pick up the signal from Z-wave microcontroller, which in turn receives the signal straight from interface (IPhone/Smartphones). The microcontroller shall customize the signals received from interface to the band of infrared frequency without losing any valuable bits. And that band of frequency shall be within the range (band) of Z-wave signals. Any remote use of the Web Interface will be impossible without an Internet connection.

12.1.2 Constraint: Such design will change the implementation of cloud requirements such as Voice control, which shall be provided using a voice management system. In addition to that, it would also require lot of programming in microcontroller with possibility of hardware manipulation in order to customize the signals. Not only would it become time consuming, it would not be feasible budget wise as well because it might require adding lots of band pass filters, which are really expensive in the market.

12. 2 Customer Requirement 3.3: 3D Map

12.2.1 Requirement Description: The system's interface shall provide 3D map view of working range of Z-Wave modules. It would also allow the user to create 3D map view depending upon the house floor plan he/she has. 3D Map view shall be broken down into the functions of number of floors the house has, then number of rooms in each floor, the location of the different kind of rooms (kitchen, living room, bathroom, bedroom, and so on), and the type of Z-wave devices that has been activated in each room.

12.2.2 Constraint: The following requirement is a whole new project which requires a lot of extra time and sophisticated knowledge of 3D modeling. As per our understanding 3D modeling requires lot of programming in flash which will affect our project in two ways. Since our module is web-based, enabling flash programs in it would complicate the performance of the website in which our Z-wave devices would be operating on. And since none of our team members are that good in doing complicated design in flash programs, it would require first learning, and then writing a program to make it work, which would be really time consuming.

12. 3 Customer Requirement 3.13: Thermostat

12.3.1 Requirement Description: The system shall also include Z-wave enabled thermostat that can be controlled from iPhone/Smart Phone interface. The thermostat setting can both be operated real-time or schedule based. The real time would mean if the user is feeling too hot, or too cold, his voice expression would be entered as an input, then the thermostat would change the temperature until user is satisfied with it. The schedule based setting shall allow the user to set various temperatures for various schedules. For morning, the temperature would automatically become little higher, and during day time, it would be little colder. Similarly, during winter temperature could be set for higher 70s, whereas for summer, it can be set to mid 60s and so on. Nonetheless, the user can set temperature anytime he wants, while he is in his house or outside.

12.3.2 Constraint: The first constraint would be budget feasibility as Z-wave enabled thermostat costs little higher. Then the complexity of the project, realizing that we are already building one Z-wave enabled device (blinds), and controlling other two, becomes really high. And then there is the time constraint, as dealing with thermostat requires whole new set of rules, description, and programming, which does not seem feasible for this project.