IoT to Improve the Modern Day Life

A Project Report
Presented to
The Faculty of the Computer Engineering Department
San Jose State University
In Partial Fulfillment
Of the Requirements for the Degree
Bachelor of Science in Computer Engineering

By
Gary Lai, Phi Lam, Thinh Phan, & Allardyce Suba
05/2016

APPROVED FOR THE COLLEGE OF ENGINEERING

Donald Hung

Digitally signed by Donald Hung
DN: cn=Donald Hung, o=San Jose State University, ou=Computer Engineering Department, email=donald.hung@sjsu.edu, c=US
Date: 2016.05.07 01:39:00 -07'00'

Donald Hung, Project Advisor

Professor Keith Perry, Instructor

Dr. Xiao Su, Computer Engineering Department Chair

ABSTRACT

IoT to Improve the Modern Day Life

By Gary Lai, Phi Lam, Thinh Phan, & Allardyce Suba

Internet of Things (IoT) is a fast growing trend in technology. IoT provide devices access to networks where they could be controlled and monitored from a local or remote location. Home devices, doors, and lights are some of the things that can utilize IoT.

There are limited tools and resources for users to monitor and control their homes. Lights are frequently left turned on when not needed. This form of idle energy consumption is a waste of electricity. The home should also be a place of safety and privacy, but there are cases of break-ins by intruders. This type of dangerous circumstance demands a measure of basic security in the home.

This project configures the Raspberry Pi development board as a local controller that connects to a server that utilized IoT. The local controller with sensor peripherals manages the behavior of the home system. Users were able to access the home network locally or remotely to monitor and control their devices. To reduce power consumption, automated features were implemented to minimize the necessary operation time of each device. A two way communication system allowed the local controller and server to interact. Sensors were installed onto doors that alerted the users in cases of intrusion. The home monitoring system provided log entries and notify the users on the status of their home. With the utilization of IoT, Raspberry Pi, and a server, users were provided with more control over their homes.

Table of Contents

List of Figures

List of Tables

Chapter 1 Project Overview

1.1 Project Goals and Objectives

Internet of Things (IoT) is the interconnection of physical objects in a network. The objects are able to communicate with each other, whether through machine-to-machine interaction or human-to-computer interaction. Architecture, sensors, programming, and transmissions are some of the most important technologies needed for IoT (Gubbi, 2013). Architecture, for instance, is the foundation for almost all technologies. If a solid IoT architecture is not present, it will be difficult to find solutions to many of the important concepts of IoT.

Implementing IoT into the system is the most important task. Without it, users are unable to communicate with objects remotely. Gathering information on all the different technologies needed for IoT will take some time, but they should not be difficult to find. The most difficult part is putting all the technologies together and creating a working home system that is able to utilize IoT. Many homes today lack the method to monitor and control their home devices and appliances. The proposed project will aim to implement a home system that will be able to control and monitor a home environment. This project will utilize IoT, which will follow a server-client design to allow communication between devices using Amazon Web Services (AWS) as the server and the Raspberry Pi as the client. These two systems can be used to monitor, store, receive, and send information interchangeably.

There are two main focuses for the project: to give users the ability to monitor and control selected devices, and to notify users when someone enters or leaves their home. Security is one of the most important features that a home should have for their users and by implementing a notification system, users will know when someone enters or leaves their home either locally or remotely. Idle device usage is also a big factor in many homes since it leads to higher energy bills for users because it consumes non-renewable energy resource. By giving the user the power to control their devices remotely and locally, users will be more aware of their energy usage.

1.2 Problem and Motivation

1.2.1 Problem

It is common for people to forget to turn off their devices and leave the house. This simple mistake can lead to many problems. One problem is the increase in energy bill. When electricity is wasted, users will still have to cover the execess cost. This means that users are paying more for electricity than they need to. Another problem is that forgetting to turn off devices causes an increase in Carbon Dioxide (CO_2) emissions. Increase in CO_2 is one of the major causes of climate change and therefore is important to find ways to offset or slow down its effects by reducing energy consumption when possible.

The home should always be a safe environment protected from intruders. Home burglary can cost people hundreds to thousands of dollars in stolen property. Therefore, home owners need to be alerted when an intruder attempts to access their home so they can prevent events such as that from happening.

1.2.2 Motivation

IoT is still a developing technology which is not widely known yet. One of the ways this project contributes to the development of this technology is by spreading awareness. By promoting IoT technology, it will likely create more interest such that the growing support for the technology will contribute to the development of it.

1.3 Project Application and Impact

This project will be integrated into homes, that will allow users to control and monitor different devices in their homes. Sensors will be installed on doors and home appliances to monitor the devices. Users will be able to access these devices locally or remotely. With a two way communication, the local controller and the server will be able to interact and the ability to control home appliances remotely will give users the ability to turn off unwanted appliances in the home, therefore reducing the power consumption in the home. Sensors on doors will be able to notify the users of any intrusion. With the completed home system, users will be able to control and know the status of their homes.

1.4 Project Results and Deliverables

1.4.1 Project Results

A web-server was used to control appliances inside the home. Sensors are attached to lights to determine whether the lights are on and they were also attached to doors to determine whether the door was open. A server-client python script was used to connect the local controller to the server such that they can send and receive data. A database was interfaced with the server and was used to store the data from the sensors. The database will allow users to monitor the status of their home. The overall completed project will allow a capable environment to turn devices and applications into smart technology.

1.4.2 Project Deliverables

An implemented server will allow communication for both the devices and users. The server was the center of the home system that allowed devices to be controlled remotely or locally. This server included a deliverable of a notification system for notifying the users the status of their homes and were received from an eMail or text. The prototype of this project used a room as the IoT environment to demonstrate how certain connected devices to the system will be controlled and monitored. This project report is also included as a deliverable since it provides the technical details of the project, such as the requirements of the project, its system design, and its implementation process.

Another deliverable for this project is a list of features that users desire. Other users aside from the group may request additional features to be implemented onto the home system. The current features to be implemented allows users to monitor and control their home devices locally or remotely. Suggested extra features may include a timer to lock all doors in homes or a video camera to view the peephole of a door.

1.5 Market Research

The International Data Corporation (IDC) has estimated the compound annual growth rate (CAGR) of the IoT market to be 17%. This means that the IoT market would increase from its $656 billion worth in 2014 to $1.7 trillion in 2020 (International Data Corporation, 2015). Companies such as Cisco and General Electric (GE) are some contributors to the IoT market. For example, an executive from Cisco, John Chambers, estimated the available net profit of the IoT market to be $19 trillion from 2014 to 2020 (Chambers, 2014). Chambers argue that this net profit will originate from the increased profits of businesses and the increased revenue of the government. Additionally, an estimated reduction of energy costs by 70% to 80% would occur if the ongoing usage of IoT continues since cities have the potential to utilize sensors in street lamps (Chambers, 2014). Similarly, GE estimated an additional $10 trillion to $15 trillion to the global GDP by 2020. The continued growth of the IoT market arises from the emerging interest of the technology in 2014.

As IoT continues to grow, more devices are connecting to the internet and as shown in Fig. 1, the estimated amount of devices connected to the internet is 35 billion by 2019 with a CAGR of 35% (Greenough, 2015). A press release from an information technology research and advisory company, Gartner, studied an estimated amount of consumers and business spending from 2014 to 2020 as shown in Table 1. This data reflects the revenue generated in the IoT market since it displays the rapid growth of spending on IoT products by consumers and the steady increase of spending from businesses (Gartner, 2015). The interest of IoT technology is once again emphasized since IoT services are becoming more available to consumers as the vast amount of devices connected increase.

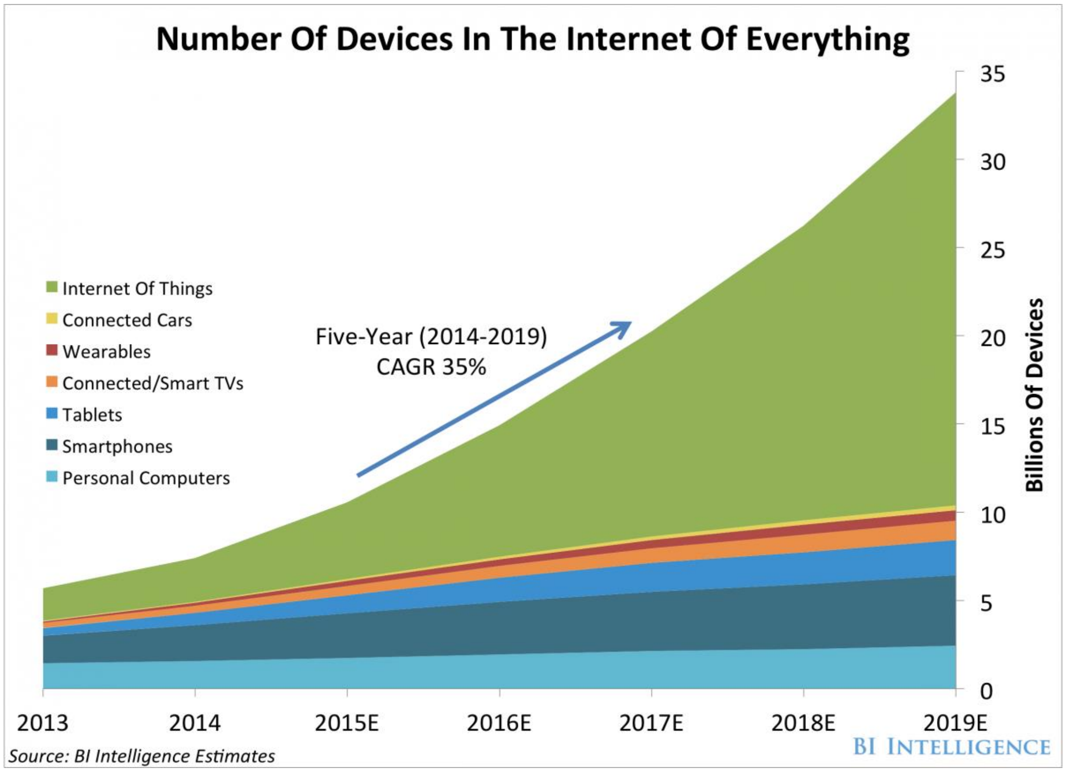

Figure 1. Number of Connected Devices from 2014-2019 Source: www.businessinsider.com

Table 1. Internet of Things Endpoint Spending by Category (Billions of Dollars)

Category	**2014**	**2015**	**2016**	**2020**
Consumer	257	416	546	1,534
Business: Cross-Industry	115	155	201	566
Business: Vertical-Specific	567	612	667	911
Grand Total	**939**	**1,183**	**1,414**	**3,010**

Source: www.gartner.com

Chapter 2 Background and Related Work

2.1 Background and Used Technologies

The project required a good background of many different concepts and knowledge on a wide range of categories. Software engineering, technical writing, hardware, database, and programming concepts needed to be studied in order to implement a successful project. Knowing software engineering concepts allowed the group to design test cases, use cases, and produce requirements for the project. The skill to find good and useful journal articles allowed the group to better understand the topic and the technical writing skill learned from an Engineering Report course helped provide a better explanation of the project to the audience. Hardware concepts were used for the design of different circuits in the project that required the knowledge of knowing how current and voltage flows through a circuit. Database concepts were required since almost all of the information required for the project were stored in a database and knowing how to insert, delete, add, and query information through MySQL was vital. The group came from a C/C++ programming language background, but the concepts and knowledge learned from those computer languages made it easier to learn new computer languages such as HTML, PHP, and Python. Table 2 below lists the relevant courses taken and how they were applied to the project.

Table 2. Relevant courses and their application to the project

<table>
<tr><th>Relevant Courses</th><th>Application to Project</th></tr>
<tr><td>CMPE 30 - Programming Concepts and Methodology</td><td rowspan="3">● Basic Programming, C
● Object Oriented Programming, C++
● These three courses helped the group provide understanding of basic programming languages, which allowed quicker understanding to different computer languages such as HTML, PHP, and Python</td></tr>
<tr><td>CMPE 50 - Object-Oriented Concepts and Methodology</td></tr>
<tr><td>CMPE 126 - Algorithms and Data Structure Design</td></tr>
</table>

CMPE 125 - Digital Design	● Helped design architecture systems ● Understand the lower level design of a system
CMPE 140 - Computer Architecture and Design	
CMPE 127 - Microprocessor Design	● Help solder components to the necessary hardware ● Organize the wiring on devices connected to the Raspberry Pi
CMPE 110 - Electronics for Embedded Systems	● Current and Voltage Flow Through Circuits
CMPE 138 - Database Systems	● MySQL ● Entity Relationship Diagram to help create the database required for the group's home system
CMPE 131 - Software Engineering	● How to write a SRS Document ● Types of software developments ● Requirements, Use Cases, Test Cases
ENGR 100W - Engineering Reports	● Skills for research and technical writing

2.2 State-of-the-Art

There are currently three different kind of concepts that utilize IoT: universal controller application, sensor based, and self-learning (Dongyu, Lo, Bhimani, & Sugiura, 2015). Universal controller application uses signals, either through infrared (IR) or WiFi, to communicate with appliances. Some examples of universal controller applications are Beacon and Broadlink e-Remote. Beacon uses IR signals to communicate with appliances, while Broadlink e-Remote requires a WiFi connection. Sensor based IoT uses sensors to determine when to communicate with an appliance through a WiFi connection. WeMo Switch + Motion is an example sensor based IoT, which uses sensors to detect movement to turn on or off an appliance connected to a switch. Similar to the sensor based, self-learning will have automated features that will turn on appliances when the user is present, but will also allow customization for users. There are currently two existing products that utilize self-learning: Nest and Tado, which are portable thermostats that allow monitoring of appliances through a smartphone application and user customization.

Anycontrol is a product that improves a user's experience by eliminating the concern about the appliances inside their homes (Dongyu et al., 2015). Anycontrol is easily installed inside homes by using add-on modules that are controlled by IR signals, to change the home's appliances. Users are able to customize task that will signal different appliances depending on the task. A Raspberry Pi was used as the universal controller to send signals to the sensors, and a smartphone application was used to monitor, control, and start task.

Samsung has released the SmartThing hub that is placed inside homes and will allow communication between devices from Samsung and other accessories like Bose, Philips, Honeywell, and others (Gibbs, 2015). For example, lights are able to communicate with the heating system. The SmartThing hub allowed users to control and monitor their devices through a smartphone application. Currently SmartThing hubs do not allow communication between all devices, but Samsung has pledged that it will be able to within five years.

2.3 Literature Survey

The ability to create smart devices by utilizing IoT opens new opportunities in the market. According to a study of the IoT market, many services and applications can be developed using wireless network sensors (WSN) (Bohli, Sorge, & Westhoff 2009). WSN is anticipated to play a big role in the IoT market because it is used by large companies and organizations. One of the services mentioned in the study was an information service, which aim to provide consumers a more understanding of the physical and digital world since sensor peripherals collect data to communicate with the internet. For example, sensors in vehicles are used to measure road conditions to improve safety. This information service could create businesses for companies and organization by renting their WSN to consumers who have use of the large collected data.

The study of the IoT market also suggests its future economy based on the consumer's willingness to pay. This willingness is determined by the quality of the service that is provided. That is, a high quality information service would provide accurate sensor readings while a low quality information service provide lesser accuracy. Moreover, the cost-effective factor of producing new services in the IoT market is low since there are already IoT network infrastructure available (Bohli et al., 2009). The only expense required would be the intermediaries for providing the services. On the other hand, since information service is new in the market, it is argued to prove its usefulness before allowing consumers to subscribe to the service by providing the service for free to closed groups.

There are five technologies that have been used to create successful IoT–based products: radio frequency identification (RFID), WSB, middleware, cloud computing, and IoT application software (Lee & Kyoochun, 2015). RFID uses radio waves, tags, and a reader to identify objects and collect data. There are three different kinds of tags used: passive, active, and semi-passive and each are able to store more data than barcode. Wireless sensor networks are uncontrolled devices equipped with sensors that are able to monitor the conditions around the device. When working with RFID systems, they are able to better track an object's status. Middleware is the bridge between software applications and makes it easier for communication, sending inputs, and receiving output. It is able to the hide the details of other technologies and is needed to hide software services that are not needed for the IoT application. Cloud computing is a network of shared resources and can be used to store large about of data and process them in real time. IoT applications are on certain devices, which allow these devices to communicate with another device or human. For a good IoT application, it has to reliably receive data and act in a timely manner.

Chapter 3 Project Requirements

3.1 Domain and Business Requirements

3.1.1 The system shall give user more control over their home.

3.1.2 The system shall alert users of unauthorized intruders.

3.1.3 The system should reduce energy consumption in the home.

3.1.4 The system shall allow users to monitor lights that are turned-on in their homes.

3.2 System (or Component) Functional Requirements

3.2.1 System

Table 3. System Functional Requirements

Req ID	Function	Description
1	Monitor Data	The system should be able to monitor data provided by the sensors.
2	Notification	The system shall be able to notify users regarding the status of their home.
3	Control	The system shall be able to control devices connected to it.

3.2.2 Server

Table 4. Server Functional Requirements

Req ID	Function	Description
4	Access Internet	The server shall be able to be accessed through the internet.
5	Control Local Controller	The server shall be able to control the local controller.
6	Simultaneous Connection	The server should be able to handle multiple and simultaneous connections from different users.

7	Communicate with Local Controller	The server shall be able to communicate with the local controller.
8	Email/Text notification	The serve shall be able to provide users with email and text notification.
9	Read and Write to database	The server shall be able to read and write to the database.
10	Access Control	The server shall have a login page where users are required to login to access other pages.

3.2.3 Database

Table 5. Database Functional Requirements

Req ID	**Function**	**Description**
11	Add	The database shall be able to add data to the database.
12	Delete	The database shall be able to delete data to the database.
13	Modify	The database shall be able to add data to the database.
14	Connection with Server	The database shall be connected to the server.
15	Log and Store Data	The database shall be able to log and store data provided by the server.

3.2.4 Local Controller

Table 6. Local Controller Functional Requirements

Req ID	**Function**	**Description**
16	Control Sensors	The local controller shall be able to control the sensors.
17	Receive Data	The local controller shall be able to receive data from the sensors.
18	Send Data	The local controller shall be able to send data to the server.

3.2.5 Peripheral Sensors

Table 7. Peripheral Sensors Functional Requirements

Req ID	Function	Description
19	Control Devices	The peripheral sensor shall be able to control the device.
20	Monitor and Log Data	The peripheral sensor shall be able to monitor and log data.
21	Send Data	The peripheral sensor shall be able to send data to the local controller.

3.3 Non-functional Requirements

Table 8. Non-functional Requirements

Req ID	Function	Description
22	Reliability	The system should not be down for more than 15 minutes a week
23	Response Time	The system should respond to the user inputs within 1 second.
24	User Interface	The system should have user-friendly interface such that 80% of the users will be able to navigate around the website.

3.4 Context and Interface Requirements

Table 9. Context and Interface Requirements

Req ID	Context and Interface	Description
25	Local controller and server	The local controller shall be connected to the server via the internet.
26	Server and Database	The server shall be the host of the database.

27	Photocell and Local Controller	The photocell circuit shall be connected to the local controller’s GPIO ports.
28	Relay and Local Controller	The relay circuit shall be connected to the local controller’s GPIO ports.
29	Magnetic Contact Switch and Local Controller	The magnetic Contact Switch shall be connected to the Local Controller’s GPIO ports.
30	Local Controller and Database	Python scripts shall be used to interface between the Local Controller and the database.
31	Database And the Web-Server	PHP scripts shall be used to interface between the database and the web-server
32	Local Controller and Sensors	Python scripts shall be used to communicate between sensor and the Local Controller.

Chapter 4 Technology Descriptions

4.1 Technology and Resource Requirements

4.1.1 Technology Requirements

4.1.1.1 Wireless Relay Circuit

Table 10. Wireless Relay Circuit Technology Requirements

Requirement	Description
Connection with Local Controller	The wireless relay circuit shall be connected to the local controller.
Connection with Device	The wireless shall have a device connected to it
Power Source	The wireless relay circuit shall be connected to the grid through an outlet.

4.1.1.2 Contact Switch

Table 11. Contact Switch Technology Requirements

Requirement	Description
Connection	The contact shall be connected to a door or window.
Power Source	The contact switch shall be connected to at least a 3.3V power source.

4.1.1.3 Photocell

Table 12. Photocell Technology Requirements

Requirement	Description
Power Source	The photocell shall be connected to at least a 3.3V power source
Connection	The photocell shall be connected to the local controller.

4.1.1.4 Raspberry Pi

Table 13. Raspberry Pi Technology Requirements

Connection	Description
Power source	The Raspberry Pi shall be connected to an external power source.

4.1.1.5 RF Transceiver

Table 14. RF Transreceiver Technology Requirements

Requirement	Description
Connection with Local Controller	The RF transceiver shall be connected to the local controller.
Communication with RF transceivers	The RF transceiver shall transmit or receive data from other RF transceiver.
Power Source	The RF transceiver shall connect to at least a 3.3V power source

4.1.1.6 Arduino Wireless Node

Table 15. Wireless Node Technology Requirements

Requirement	Description
Wireless capability	The wireless node shall communicate wireless to a base node using RF signals.
Connection with Device	The wireless node shall have sensors and devices connected to it.
Power Source	The wireless node shall be connected to a 3.3V power source.

4.1.2 Resource Requirements

4.1.2.1 Internet access

4.1.2.1.1 Access to the server and database

4.1.2.1.1 Monitoring and controlling devices through the website

4.1.2.1.1 Update database and server

4.1.2.2 Power Source

4.1.2.2.1 Each unit requires a power source to operate.

4.1.2.3 Equipment & devices

4.1.2.3.1 Computer is needed to access the terminal of the local controller, server, and to write and edit code

4.1.2.3.2 Sensors and local controllers are needed for each unit.

4.2 Environment and Interface Requirements

4.2.1 Environment Requirements

4.2.1.1 Wireless Relay Circuit

4.2.1.1.1 The wireless relay circuit shall output status reports.

4.2.1.1.2 The wireless relay circuit shall take inputs from the local controller.

4.2.1.2 Magnetic Contact Switch

4.2.1.2.1 The contact switch shall take at most 100mA current.

4.2.1.2.2 The contact switch shall signal “off” when the distance between the two parts are less than or equal to 15mm.

4.2.1.2.3 The contact switch shall be connected to a power source that does not exceed 200 V.

4.2.1.3 Photocell

4.2.1.3.1 The photocell shall be located in an area that shall be able to capture light intensity.

4.2.1.3.2 The photocell shall not exceed 150V.

4.2.1.3.3 The photocell ambient temperature range shall fall between -30°C to 70°C.

4.2.1.4 Raspberry Pi

4.2.1.4.1 The raspberry pi shall be located in the same room of the sensors.

4.2.4.4.2 The raspberry pi operating temperature shall range from 0°C to 70°C.

4.2.1.5 RF Transceiver

4.2.1.5.1 The RF transceiver shall operate under 14mA current.

4.2.4.5.2 The RF transceiver shall send RF signals on 2.4GHz ISM band.

4.2.4.5.3 The RF transceiver shall send/receive data with at least 1Mbps on-air data-rate.

4.2.1.6 Arduino Wireless Node

4.2.1.6.1 The wireless node shall communicate with home controller within at least 30 feet distance apart.

4.2.4.6.2 The wireless node shall enter power-down mode whenever possible when not transmitting or receiving data.

4.2.4.6.3 The wireless node shall wake up from power-down mode when triggered by external interrupt from a sensor.

4.2.2 Interface Requirement

4.2.2.1 Wireless Relay Circuit to Raspberry Pi

The wireless relay circuit shall have a two way communication with the Raspberry Pi to allow inputs from the Raspberry Pi to turn on or off devices, or output status reports from the wireless relay circuit to the Raspberry Pi.

4.2.2.2 Magnetic Contact Switch to Arduino Wireless Node

The magnetic contact switch shall have a one way communication with the wireless node, in which the magnetic contact switch shall trigger interrupts to the wireless node.

4.2.2.3 Photocell to Raspberry Pi

The Photocell shall send a value of resistance to the Raspberry Pi, depending on the amount of light present in the environment.

4.2.2.4 Raspberry Pi to Server

The Raspberry Pi shall have a two way communication with the server that shall allow inputs from the server to trigger GPIO pins and shall also send information about the sensors to the server.

4.2.2.5 RF transceivers to local controller

The RF transceivers shall connect with their local controller through SPI pins for master-slave two way communications.

4.3 Interface and Components Design

In Fig. 2 below, the diagram shows how components are deployed in hardware. The Presentation Layer component that is part of the web server was used to communicate with the user’s web browser. Also part of the web server is the Database Interface component that communicates between the Presentation Layer component and the Database Server through the MySQL Database component. The Web Server and the Database Server were enclosed by Amazon Web Service since they were implemented and ran on AWS infrastructure.

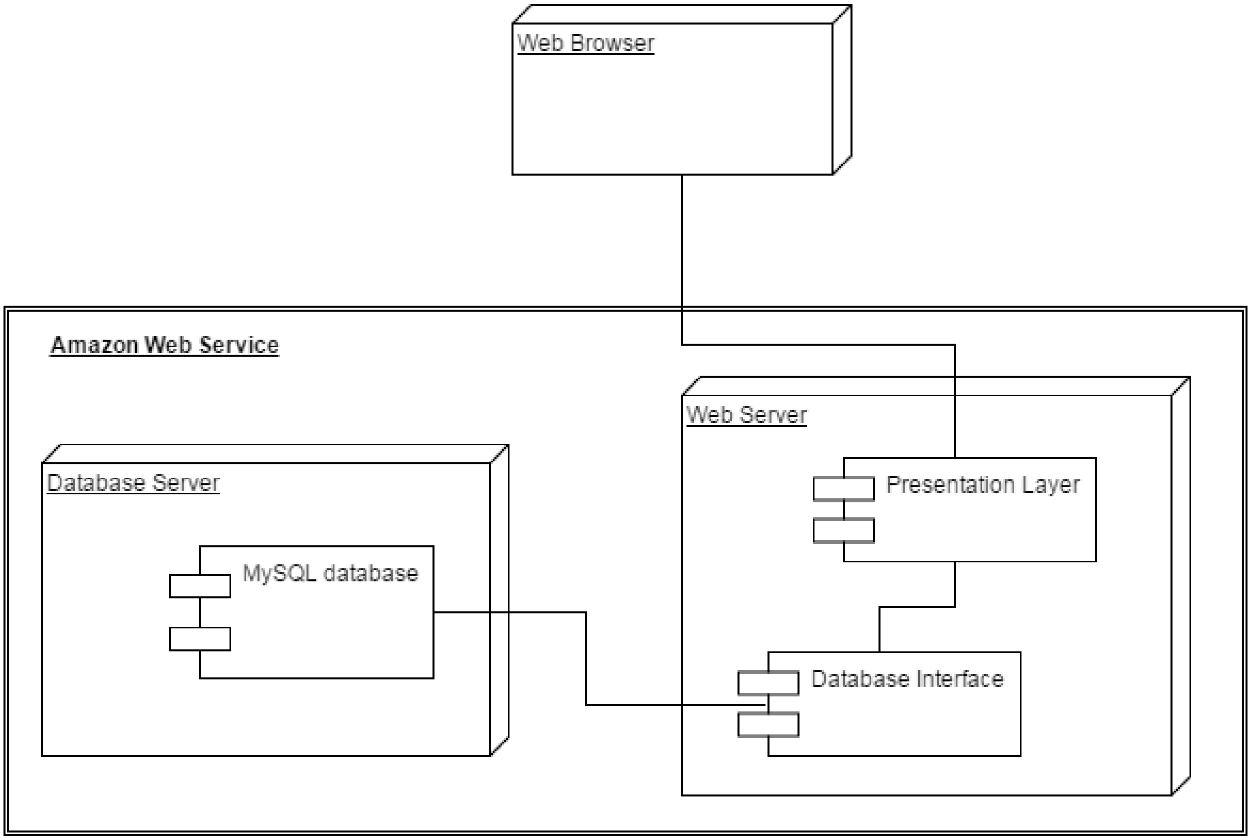

Figure 2. Components Deployment Diagram

Chapter 5 Project Design

5.1 Architecture Design

The project's architecture layout consisted of three main components: a server module, a microprocessor, and peripheral sensors. The microprocessor is considered the brain of the project's setup that handled and performed tasks based on inputs received from the server or sensors. The sensors were tools utilized to extend the microprocessor's functionalities by giving it controlling and monitoring capabilities of the home environment. The server was the front-end part of the project that allowed remote access of the home system to the user. This section includes details pertaining to the project's architecture layout and its subcomponents.

In Fig. 3, the top-level block diagram shows the system prototype layout for the project. The Door Sensor, Photocell Sensor, and Relay Switch Controller acted as peripheral sensors connecting to the Raspberry Pi system. The Raspberry Pi system was the intermediate communication between the peripheral sensors and the cloud web-base server. This project utilized Amazon Web Service (AWS) to host the web server and database for the project.

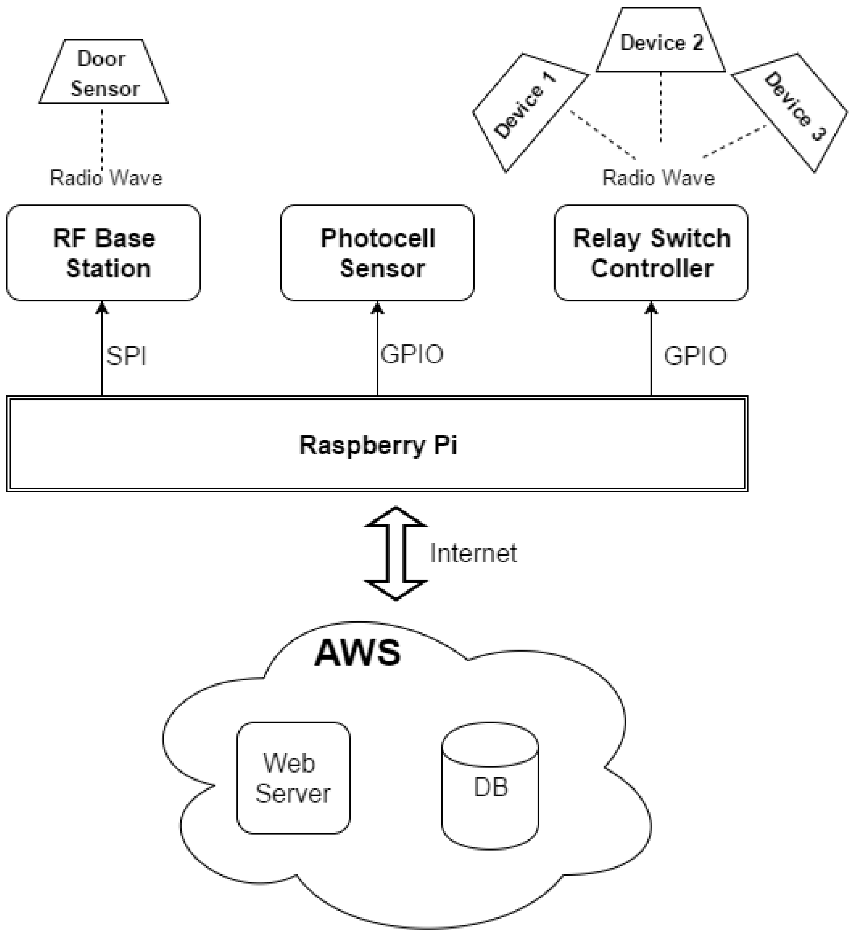

Figure 3. Top-level project design

The photocell circuit, as shown in Fig 4, was used for measuring light intensity of a room. This was a simple circuit where the photoresistor R_L increased or decreased based on light intensity. Additionally, the group used inputs provided by the R_L to detect if there were too much illumination in a room. That is, the group's system was able to monitor and detect whether there were significant light in the room. This circuit was integrated into the system such that it allows users to decide whether they want the system to automatically control the lights depending on the intensity of light in the room. The photocell consisted of the following parts:

- GL5528- photoresistor
- 1 µF capacitor

○ VCC and GND are supplied by Raspberry Pi pins.

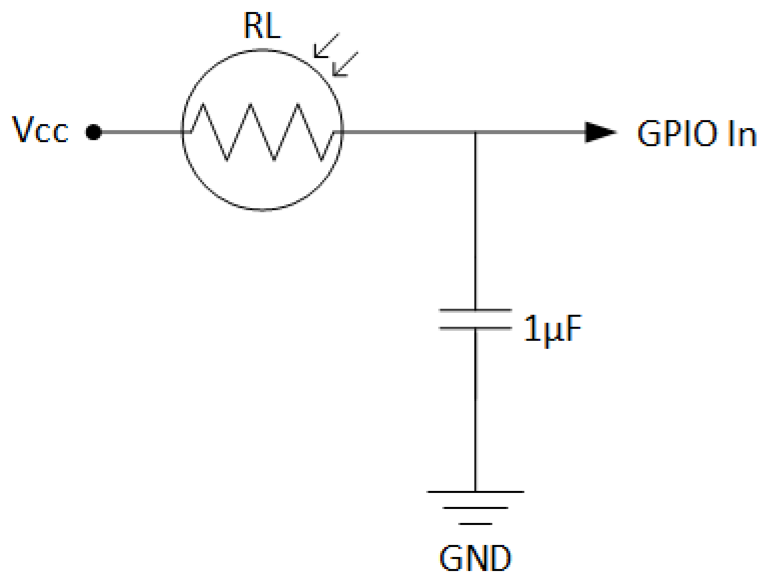

Figure 4. Photocell Circuit Diagram

In Fig. 5, the interaction between the Raspberry Pi and a wireless relay is shown. This circuit was created by soldering button switches to each Collector of the NPN transistors. The NPN transistors in this circuit acted as switches to control the wireless relay by connecting a GPIO Pin of the Raspberry Pi at each Base of transistor. For example, GPIO_1 was able to turn on the wireless relay if a signal is received from the server by allowing current to flow from the Base to the Collector. Similarly, if an off signal was received from the server, GPIO_2 turned off the wireless relay. If there were no signals received, the emitter of the transistor sinks the current, which acted as an open circuit. The communication between the transmitter and receiver was a radio frequency operating at 315MHz.

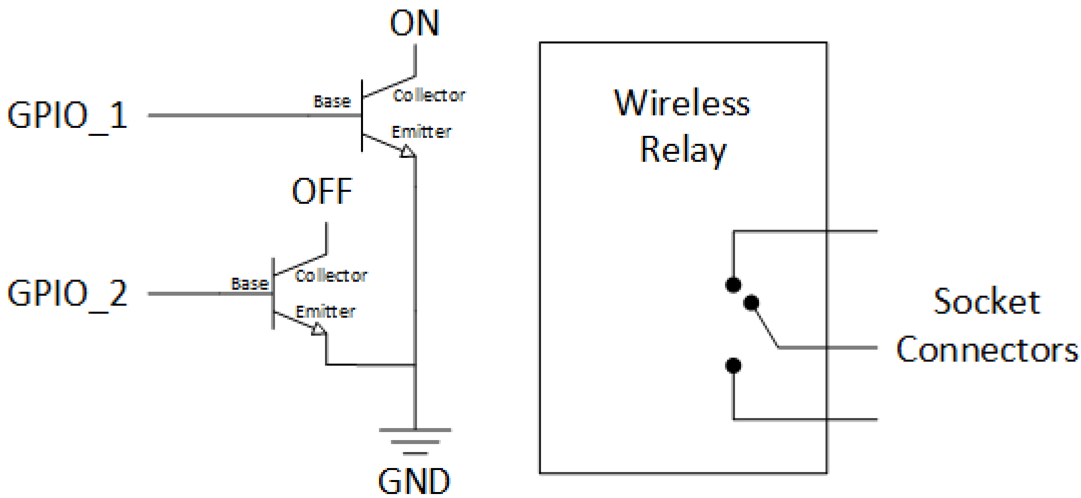

Figure 5. Raspberry Pi and Wireless Relay Circuit

In Fig. 6, the interface connections between the radio frequency (RF) base station and Raspberry Pi is shown. The RF base station is a Nordic nRF24L01+ that is highly integrated, ultra-low power, 2Mbps RF transceiver IC for the 2.4GHz ISM (industrial, Scientific and Medical) band. This RF transceiver communicate with the Raspberry Pi microcontroller through SPI protocols using pins such as Master Input Slave Output (MISO), Master Output Slave Input (MISO), and Serial Clock (SCLK). Since the nRF23L01+ operates at 3.3 V, the 3.3V output pin from the Raspberry Pi connects to the VCC input of the RF base module. The IRQ pin of the RF transceiver is unused, so it was left unconnected. The purpose of this RF transceiver was to act as the RF base station for the Raspberry Pi to receive and transmit data with other peripheral RF transceiver modules residing with wireless door sensor.

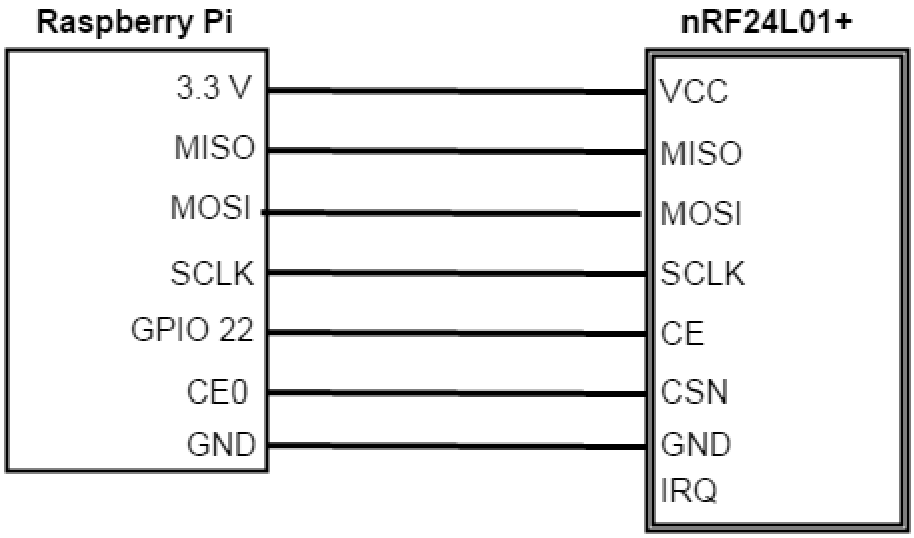

Figure 6. RF Base Interface with Raspberry Pi

In Fig. 7, the wireless door sensor components and its interface connections are shown. The Arduino Mini Pro microcontroller connection to the RF transceiver forms a wireless node that can communicate to a RF base station. The 3.3V 8 MHz version of Arduino Mini Pro was selected in implementing the wireless sensor node because it can output a regulated 3.3V, which is applicable for connecting and powering low powered devices such as the magnetic contact switch and the nRF24L01+ transceiver module. The magnetic contact switch is a normally open reed switch enclosed in an ABS plastic shell. Typically, open means that there is no connection between the two lead wires when there is no magnet close to the reed switch. But when there is a magnet approximately less than 13mm away, the reed switch closes. Base on the connection diagram, when the reed switch opens, the INT0 input pin of Arduino Mini Pro will read HIGH level voltage, triggering an external interrupt that prompt the Arduino Mini pro to send data to the RF base station using RF transceivers.

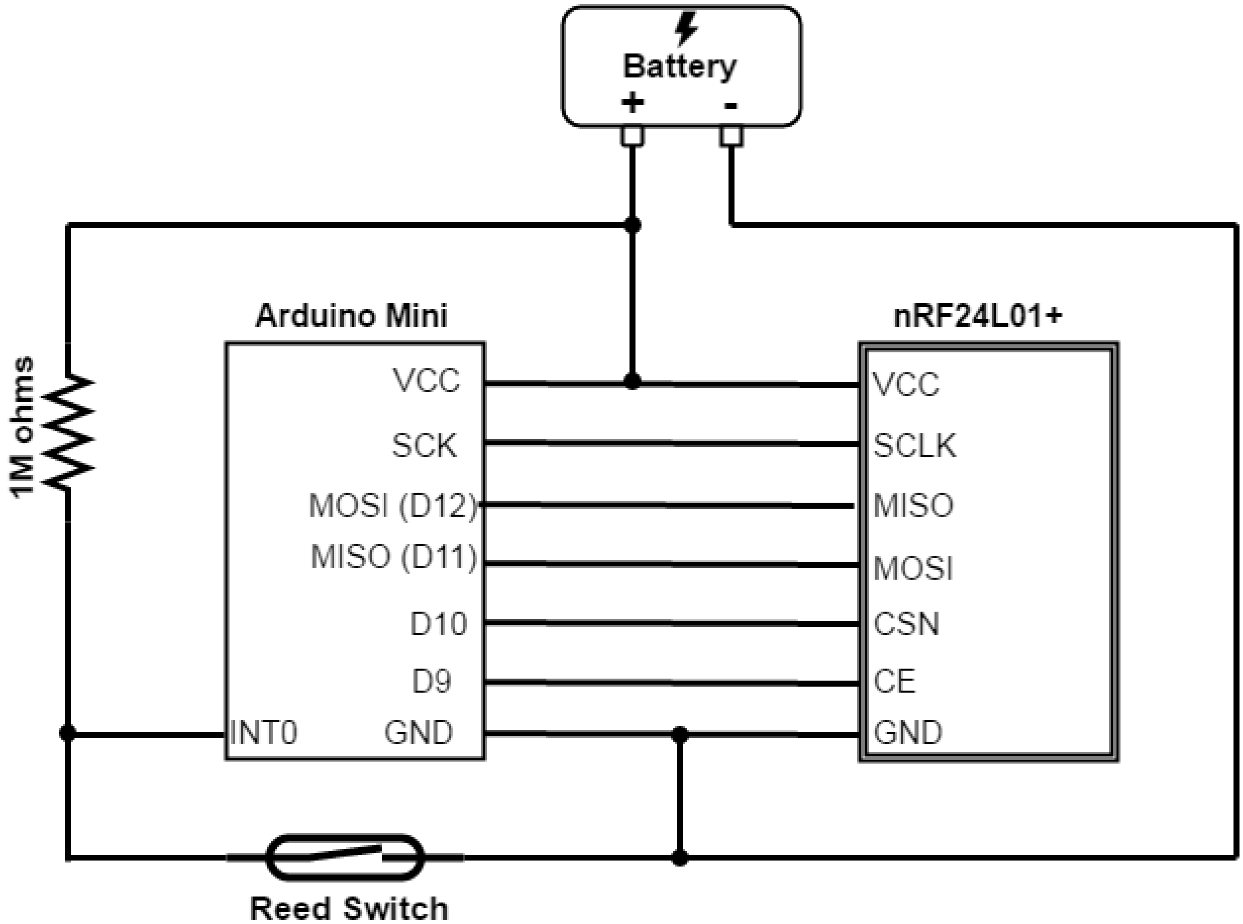

Figure 7. Wireless Door Sensor Connections

In Fig. 8, the ER diagram for the project is shown. The ER diagram represents the database used in the server and how each entity related to each other. The ER diagram consisted of six entities that were used. Three entities represented the devices that communicated with the Raspberry Pi: the contact switch, photocell, and the relay circuit. The Raspberry Pi was the most important entity because it is used to send communication from the devices and to the server. For example, the Raspberry Pi needs to communicate with at most one server, but the server could communicate with many Raspberry Pis.

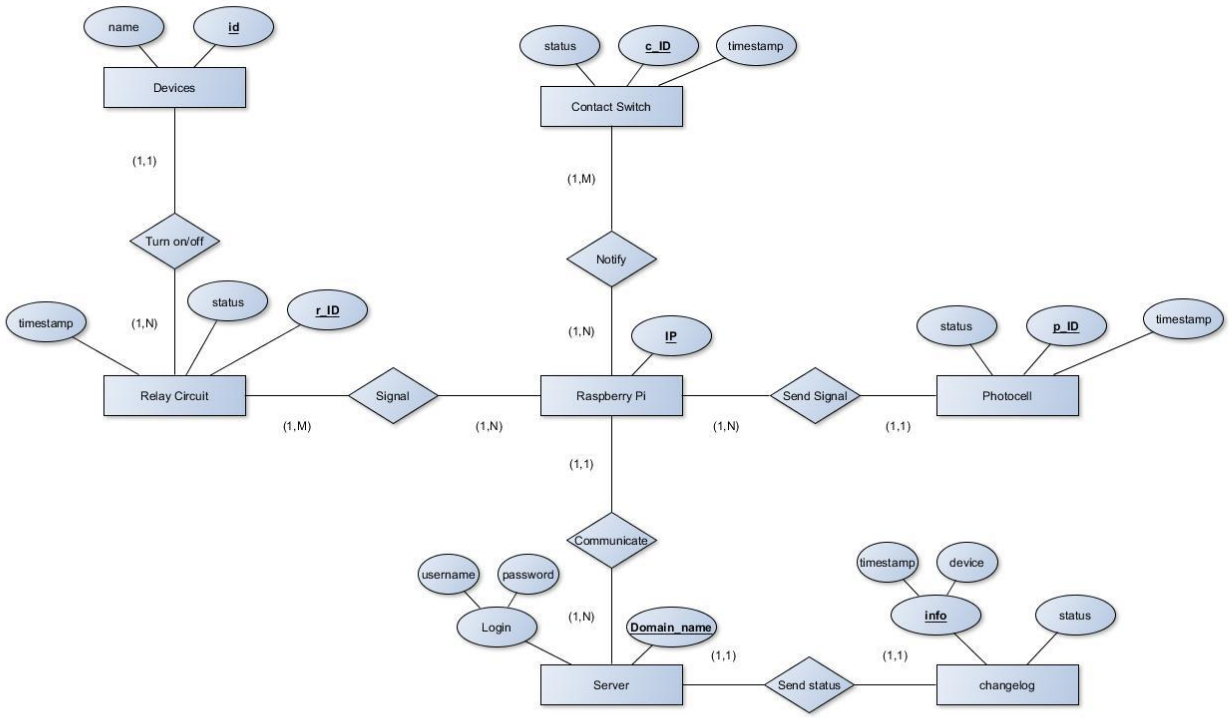

Figure 8. ER Diagram of Project's database

In Fig. 9, the generalization of the database is shown. Device 1, device 2, and device 3 can be generalized as relay, while username and password can be generalized as login. All other entities can be generalized as the web server.

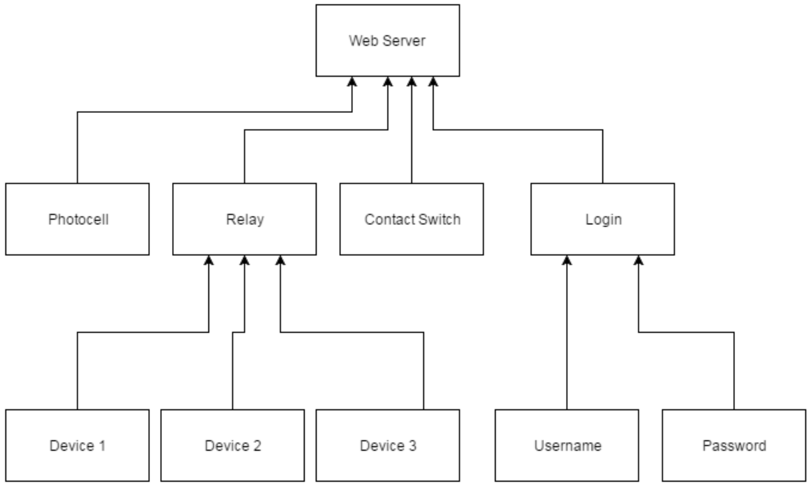

Figure 9. Generalization of Database

5.2 Structure and Logic Design

5.2.1 Structure Design

In Fig. 10, the communication relationship between each major component of the product is shown. Wireless relay circuit and contact switch will send boolean values, 0 or 1, while the photocell will send double values as status reports to the Raspberry Pi. The Raspberry Pi will write these data values to the server/database. The server will be able to write control signals to the database, and from the database, the Raspberry Pi will be able to send signals to the wireless relay circuit to turn on or off the device. Status reports will be constantly outputted from the server to be displayed on the website.

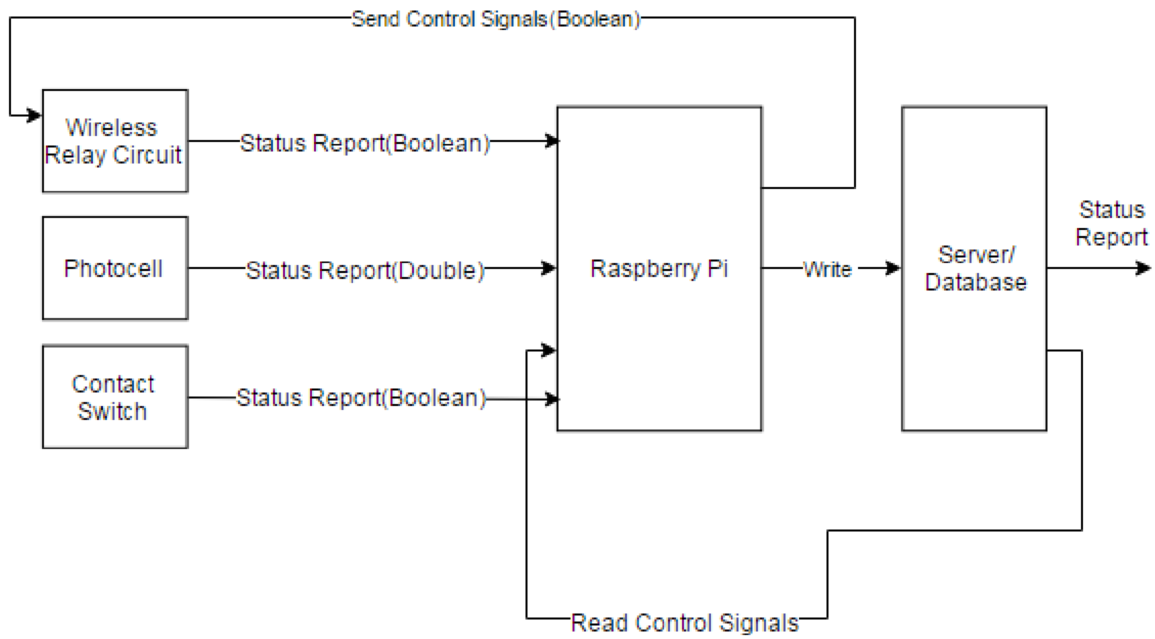

Figure 10. Communication relationship between each major part of the project

5.2.2 Logic Design

In Fig. 11, the state machine diagram for the relay circuit is shown. In this design, the initial state is S0 where the device is idle and waiting for a signal. When the signal received is "on," then the state is moved to S1 where the GPIO output connected to the on switch of the relay transmitter is turned on. On the other hand, when the signal received is "off," the state is moved to S2 where the GPIO output connected to the off switch of the relay transmitter is turned off. After performing the statements in S1 or in S2, the relay circuit is back in idle state to wait for a new signal.

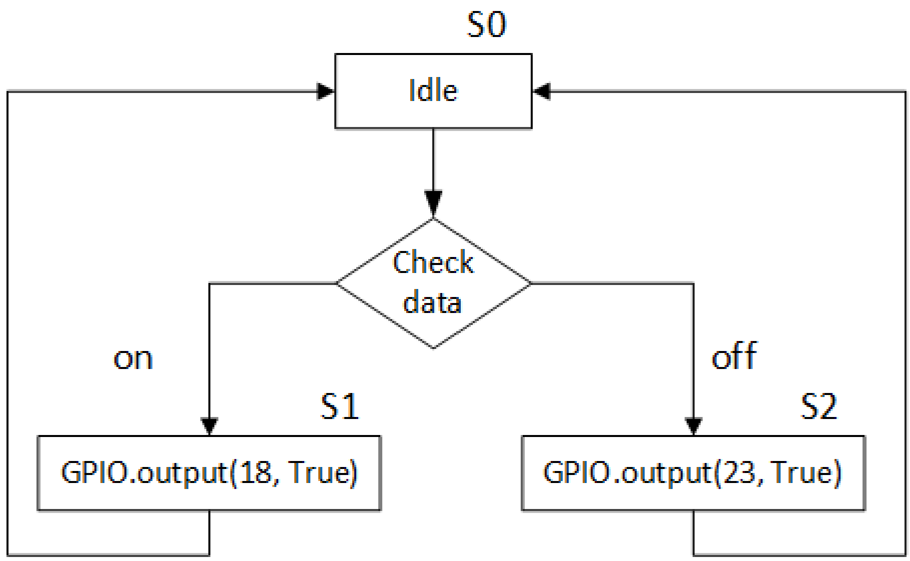

Figure 11. Relay Circuit State Machine Diagram

In Fig. 12, the flow logic diagram of the wireless door sensor system is shown. At state 0, the sensor is powered on and transition into state 1 to enter sleep mode. At state 1, the system remains idle during power-saving mode until an external interrupt event occurs. In the event of an external interrupt (voltage level changes), the system enters state 2 where the door status is checked by reading the voltage level of the external interrupt pin. If the door status read a LOW (0) value, the door sensor sends the payload message “door close” to the base station at state 3 and if the door status read a HIGH (1) value, the payload message “door open” is sent at state 4. The system then transition to state 1 to resume in sleep mode and wait for another external interrupt event.

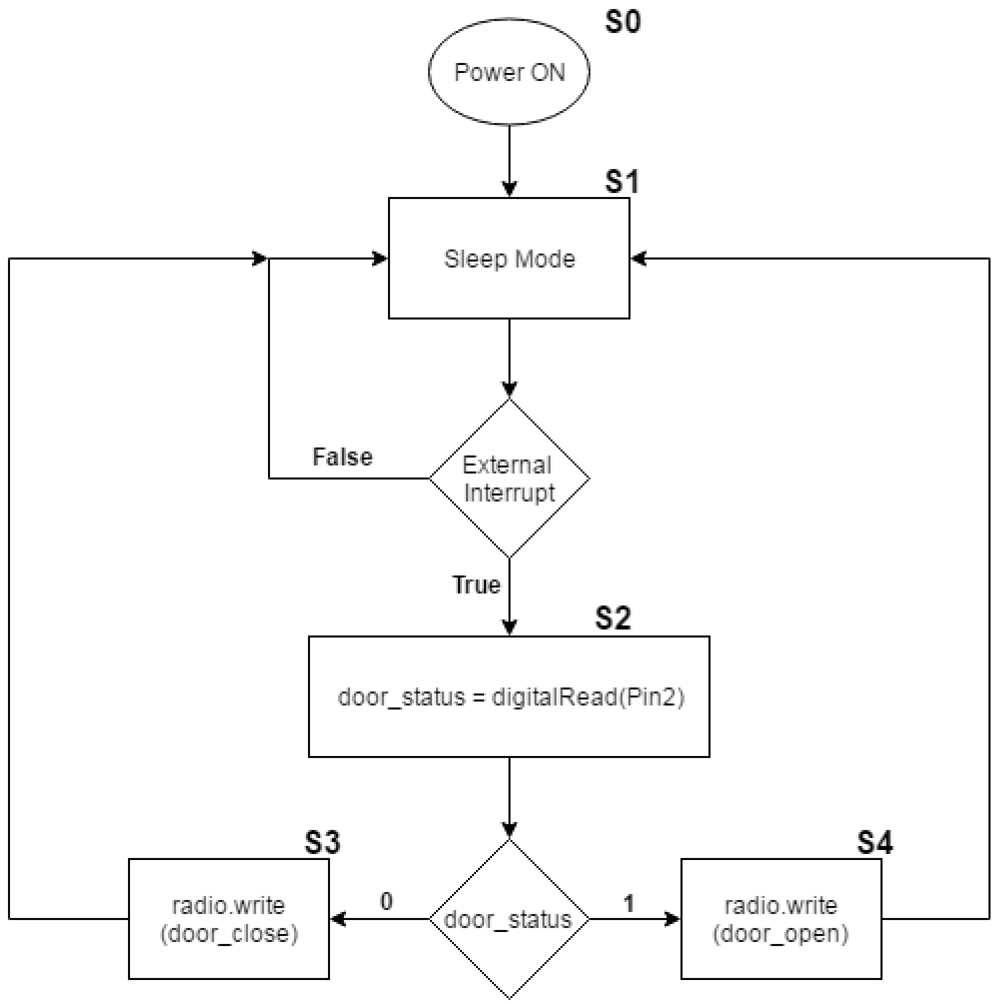

Figure 12. Wireless Door Sensor System Logic Flow

In Fig. 13, the sequence diagram is shown and shows how the system will work depending on the user's input. There are four different inputs that are shown on the sequence diagram, which are login, turn on device, turn on light control, and turn on door notification. For example, the user will have to enter username and password on the login page, which will be checked in the database. If the username and password are correct, the user will be able to view the control and monitor page, otherwise they would get an error message. Another example, would be to turn on a device, the user will input turn on device on the control page. The status of the device would be checked in the database, if it is the same, nothing happens, otherwise it will send the instruction to the local controller. From the local controller, instruction to turn on the GPIO will be sent to the device, then the device will be turned on. Once the device is turned on, the status will be sent back to the database and updated, which will be able to be seen on the control page.

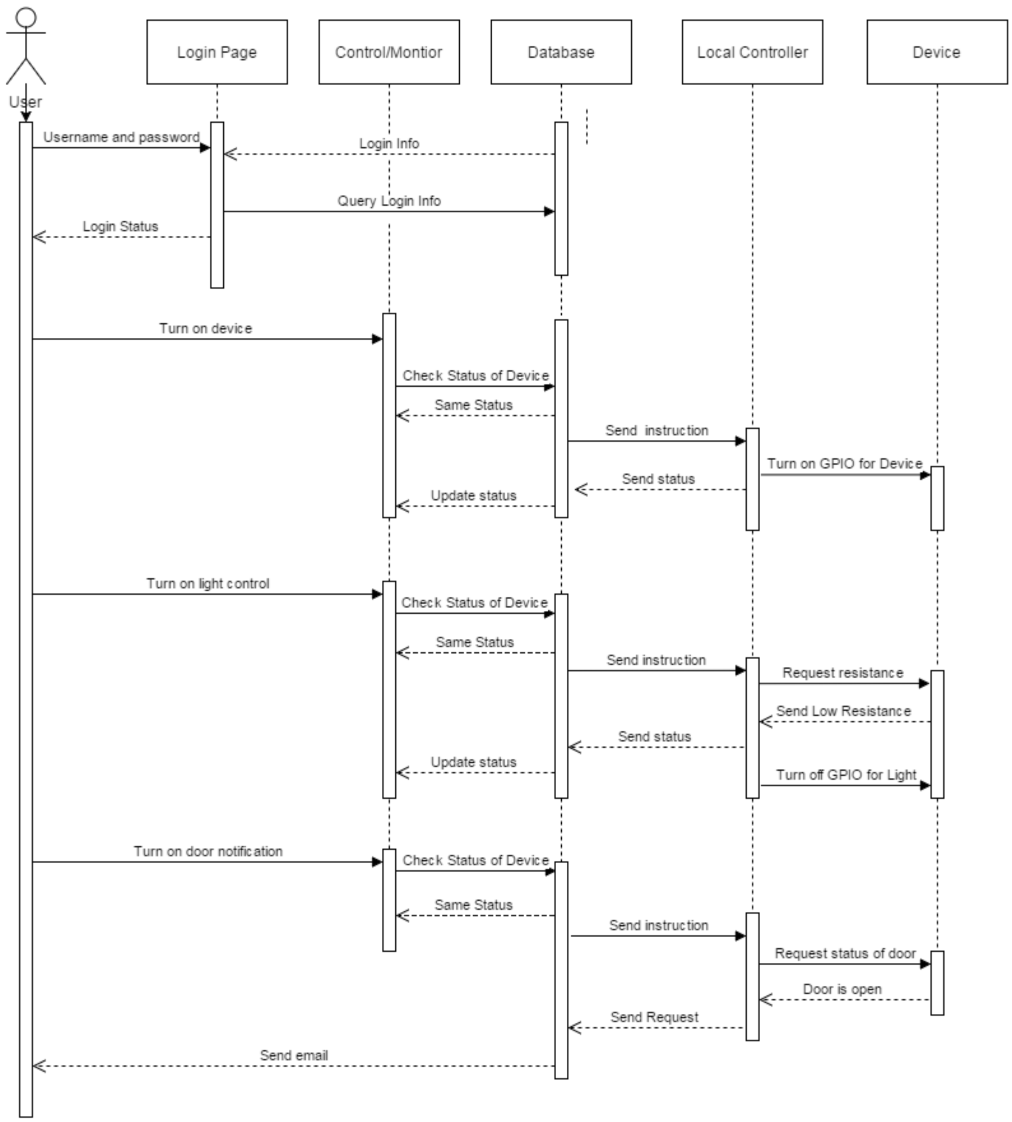

Figure 13. Sequence Diagram for the system

In Fig. 14, the client server diagram is shown. In this diagram, the client is the local controller and the server is the AWS server. The local controller will need to constantly request information about the status of the devices from the server, which is stored in the database. The server is able to query information to and from the database. The results of the query will be sent as a response from the server to the local controller.

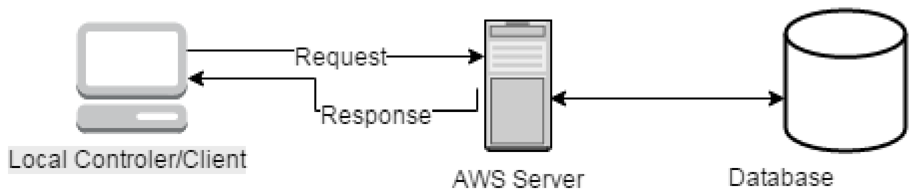

Figure 14. Client-Server Diagram

5.3 Design Constraints, Problems, Solutions, and Tradeoffs

5.3.1 Design Constraints

The hardware constraints that were present in the project's design were its wired connections, 5V limitation of the relay circuit, required WiFi dongle for connecting to the internet, and the required connected power source of the Raspberry Pi. Another constraint was the voltage required to power the transmitter of the relay circuit.

5.3.2 Design Problems and Challenges

The wired connections of each sensor peripheral required the Raspberry Pi to be nearby each sensor. The problem that was encountered from this constraint was the distance between each sensor since some were not able to reach a physical connection with the Raspberry Pi. For example, the outlet used for the relay circuit and the magnetic contact switch for the door were in opposite sections of the room. Additionally, the problem with the 5V limitation of the relay circuit prevented certain devices to be connected to the home system. The Raspberry Pi using WiFi to connect to the internet was also a problem because the connection strength relied on the distance between the Raspberry Pi and the router. For example, if the Raspberry Pi was located in a room 100 ft. from the router, the internet connection would be weaker than if it was right next to the router.

Rather than using a battery to power the transmitter of the relay circuit, the group wanted to use the 5V source available from the Raspberry Pi. However, the transmitter required 12V to be functional. If the transmitter were powered by the 5V source of the Raspberry Pi, the communication would not be complete between the transmitter and the receiver since the signals will not be sent.

5.3.3 Design Solutions and Trade-offs

Wireless sensors were used to address the constraint of wired connections, but using wireless sensors proposed a design tradeoff since wireless sensors have limited reach. This trade-off prevented the Raspberry Pi's location to surpass the reachable distance of the wireless sensors. This problem was addressed by relocating the Raspberry Pi to be in a designated spot within reach of the wireless sensors. The 5V limitation of the relay circuit was addressed using a voltage regulator, which allowed certain devices to regulate within the limits of the relay circuit. Moreover, the tradeoff of using a WiFi dongle is that it eliminated the use of wire connection, but resulted in slower response time as well as a distance constraint. This problem was solved by placing the Raspberry Pi as close as possible to the router.

The solution to continue using the 5V source from the Raspberry Pi to power the transmitter of the relay circuit was to use a step-up voltage. This step-up voltage takes the 5V output source of the Raspberry Pi as an input and amplify the output voltage of 12V. This 12V output was then used as the source to power the transmitter.

Chapter 6 Project Implementation

6.1 Project Team

The project team consists of 5th year Computer Engineering undergraduate students at San Jose State University. The team members are Gary Lai, Phi Lam, Thinh Phan, and Allardyce Suba. The team was created in Summer 2015 and each member have been associated with each other for many years within the Computer Engineering program. In this project, each member of the team was assigned tasks and were responsible for completing the assigned tasks within its specified completion date. The tasks that were distributed in this project consisted of hardware and software units that later needed to be integrated into a system. This distribution of tasks allowed the group to work efficiently when not in collaboration since the separate units were capable of being completed alone. On the other hand, if a team member was unable to complete a specified tasks within its given time, another member of the team is capable of providing help to complete the task on time. The group aims to graduate in May 2016 with a working prototype to present towards the end of the semester.

6.2 Project Tasks and Schedule

Table 16. Project Tasks and Schedule

ID	Task Name	Lead	Start	Finish	Duration
1	Contact Switch (CS) Integration	Phi Lam	2/5/2016	2/12/2016	8d
2	CS Integration testing	Phi Lam	2/12/2016	2/13/2016	2d
3	Relay Circuit (RC) Integration	Allardyce Suba	2/5/2016	2/19/2016	15d
4	RC Integration testing	Allardyce Suba	2/19/2016	2/22/2016	4d
5	Photocell (PC)	Thinh Phan	2/12/2016	2/19/2016	8d

	Integration				
6	PC Integration testing	Thinh Phan	2/19/2016	2/22/2016	4d
7	Milestone 1	Group	2/22/2016	2/22/2016	0d
8	Database Creation	Gary Lai & Thinh Phan	2/22/2016	2/26/2016	5d
9	Database Integration	Gary Lai & Thinh Phan	2/26/2016	3/4/2016	8d
10	Integration testing and debugging	Group	3/4/2016	3/18/2016	15d
11	Milestone 2	Group	3/18/2016	3/18/2016	0d
12	Notification System	Phi Lam & Allardyce Suba	3/18/2016	4/4/2016	18d
13	UI Web Server	Gary Lai	3/18/2016	4/4/2016	18d
14	System Testing	Group	4/4/2016	5/6/2016	33d

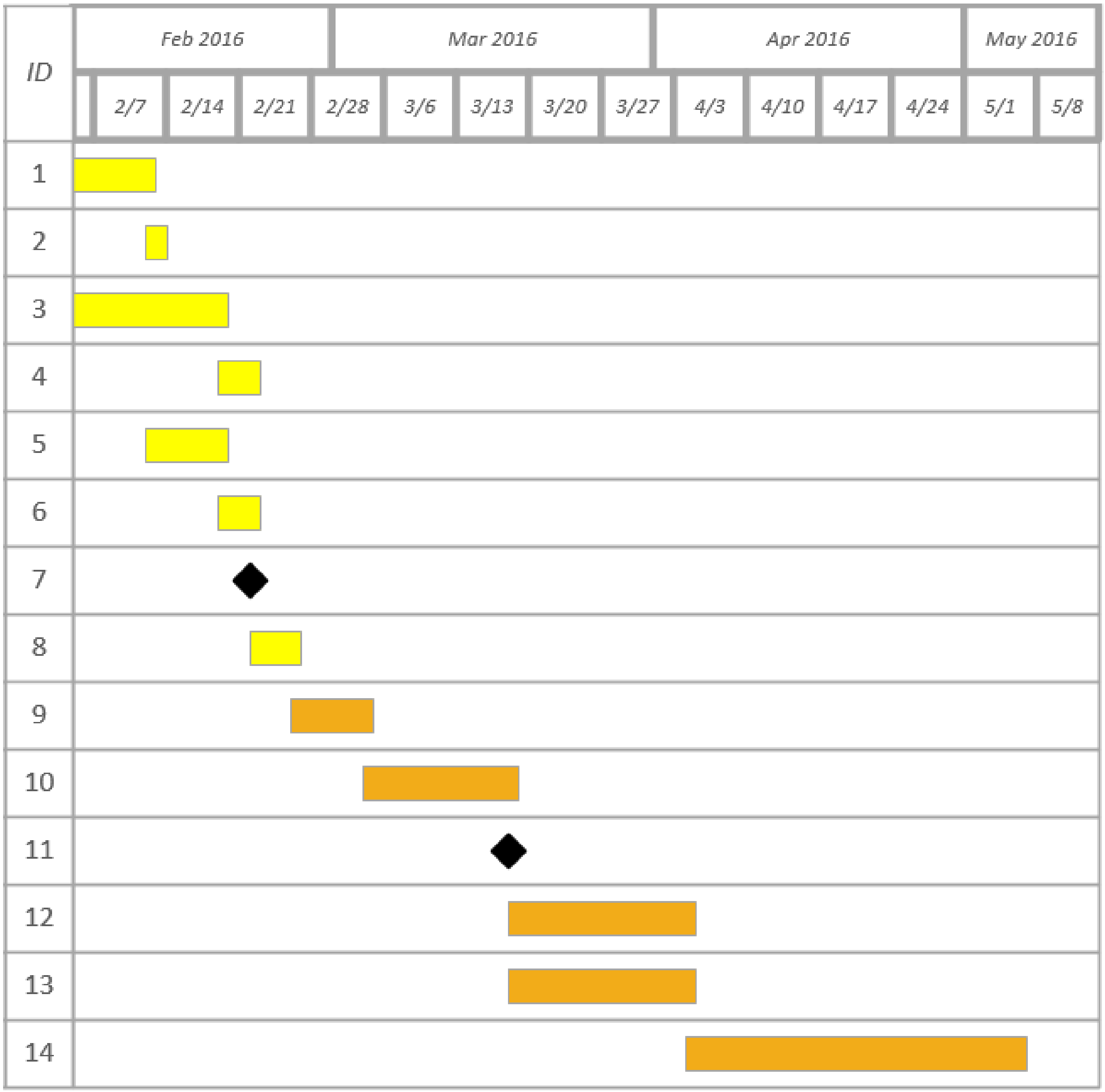

Figure 15. Project GANTT Chart

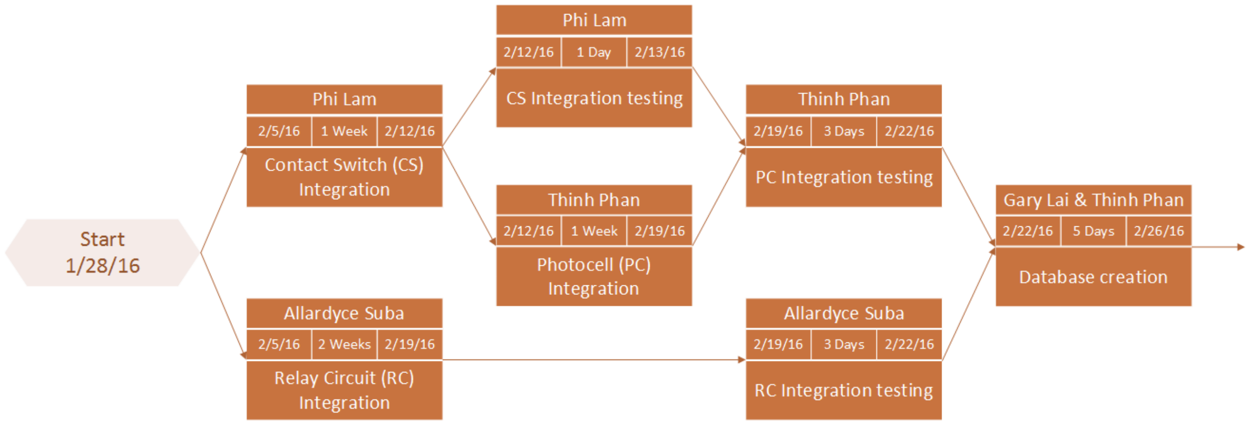

Figure 16. Project PERT Chart (1)

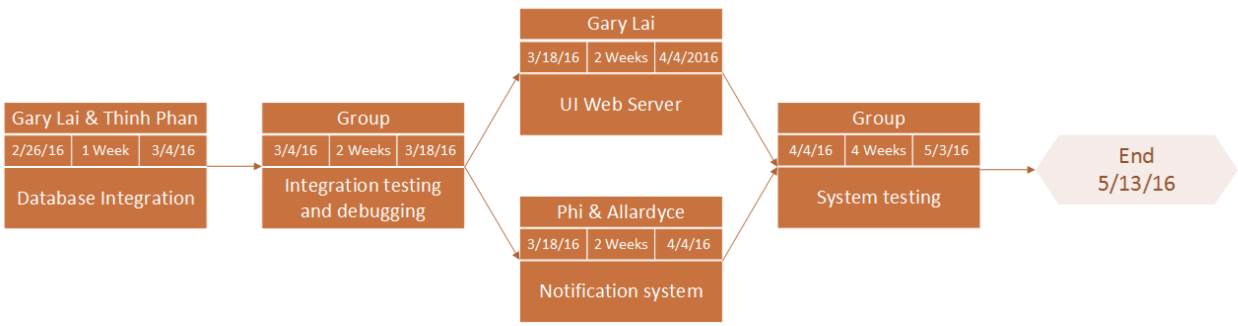

Figure 17. Project PERT Chart (2)

6.3 Tools Used

One of the modern engineering tools that the group used was Github, which provided a repository for the project's programs and has contributed to the better work efficiency of the group. Github allowed each member of the group to make direct changes to the programs in the repository with a visible revision history that helped keep track of the changes. This revision history was helpful since it made debugging easier by allowing the group to identify which lines of code contributed to the solution or the problem to a certain program. Additionally, Github allowed the group to work remotely when not collaborating in person since the tool allowed files to be easily pushed or pulled from devices.

Another tool that was used in this project was Google Drive, which was used as a repository for the project's documentation. Google Drive is similar to Github since it allowed the group to document the project remotely and provided a revision history on the changes that each member has committed. Additionally, Google Drive allowed the group to communicate with each other through its built-in chat messenger or by leaving a comments on certain line of the document. Usually, the comments are for proofreading purposes and each member could respond by either replying to the comment or by fixing the suggested problem.

Wampserver is a local based server that already includes Apache, PHP, and MySQL database, which were used to test the different programming languages needed for the website that includes HTML, CSS, PHP, and MySQL. Since the project was split up into separate units, the project's server was not available for use of testing the unit. Without a server, PHP codes were not able to be tested, since it is a server-side programming language, but Wampserver made it possible. Additionally, Wampserver allowed the group to test how each of the programming languages worked with each other and the resulting web page.

Amazon Web Service (AWS) was a tool that was used to create the server for the project. Amazon Web Service has a wide range of different resources available and the EC2 resource was used for the project. To access and manage the server, two other tools were used, Putty and WinSCP. Putty was used to access the server, which allowed the use of the terminal to install new resources to be used on the server, such as Apache, PHP, and MySQL. WinSCP allowed files to be managed, created, and edited through the graphical user interface (GUI) rather than the terminal. One of the biggest advantage of using WinSCP rather than the terminal, is that it made it much easier to edit files through the computer's text editor than on the terminal.

Chapter 7 Testing and Verification

The general purpose of performing testing is for verification and validation of the project's specification requirements. Verification is performed to test whether the project is implemented correctly and validation is performed to test whether the project meets its intended use. The testing that were performed in this project consisted of unit testing, integration testing, system testing, functional testing, non-functional testing, reliability testing, and acceptance testing. In each of these tests, white-box testing and black-box testing were performed. White-box testing was performed for code review and black-box testing was performed to verify the expected output of each unit or system. To better track the progress of testing, a traceability matrix was created that listed all test cases of the project and whether each test case were successful. This matrix was useful since it was used whenever regression testing was performed. From each test performed, the group receives an analysis of the flow of progress in the project. Testing for this project is a reiterative process for producing more successful results for each test.

Black-Box testing

Black-box testing is performed under each test to verify the actual output and expected output of each unit, integration, and system. The results received from performing this test provided an analysis to whether a certain test case had to be performed again.

Photocell Black Box Testing

Black box testing was done on the units. For the photocell circuit, the code was executed and the output was compared with the inputs. In a dark room, the photocell resistor would have a high resistance and that would be the input, and therefore our output would specify that the room is dark. The group then turn on the light, and the photocell resistor would have a lower resistance, and therefore our output would specify that the room is bright.

Contact Switch Black Box Testing

Black box testing was also done on the contact switch circuit. The code was executed and the output was compared to the inputs. The inputs is whether the contact switch is touching or not, and the output would be “on” if they are touching and “off” if they are not touching. The group tested it by touching the switches together and the result was that it printed “on” and when the switches are not touched, the results printed was “off”.

Relay Circuit Black Box Testing

For the relay circuit, the group created a local server which served the purpose of a “driver” because the web-server was not ready at the time. The code was executed for the relay switch circuit and through a local server, which had buttons for “on” and “off”. When “on” was pressed, the buttons would turn on and when the button “off” was pressed, the lights would turn off.

Arduino Wireless Node Black Box Testing

Black box testing was also done on the Arduino Wireless Node’s capability to detect interrupts in changing voltage level. To test interrupt functionality, the wireless node is initially put to sleep by running on powered down mode because in order for the system to wake up, from sleep, an interrupt must occur. The group tested it by changing the voltage level from LOW to HIGH and from LOW to HIGH on an interrupt pin and see if a variable’s value is incremented in the Interrupt Service Routine function.

Database Black Box Testing

For the database, the group used a test stub that created dummy values for our data to insert and update the database because the database was not yet integrated with the photocell circuit, relay circuit, or the contact switch at the time. The code was executed to insert data from the test stub into tables in the database. The results indicated that the values were inserted successfully into the database.

White Box Testing

The purpose of white-box testing is for code review without executing the code. This form of testing is used to verify the project by having a different member inspect the code. For example, Allardyce wrote the code for the relay circuit and Thinh review the code by inspecting its control flow and its data flow to determine whether each statement or path are feasible. Additionally, the requirements specification document was analyzed for ambiguity, contradictions, illogical test cases, and understandability. White-box testing was also used in the requirement specification document of the project for analyzing possible unit tests. Performing white-box testing is important before execution of the code to analyze possible bugs or errors in the code rather than the unexpected output of each unit.

Unit Testing

Unit testing is performed on each unit of the system, where each unit will have its own code and hardware. This project will have the photocell circuit, relay switch circuit, magnetic contact switch circuit, and the database as units. Test cases will be designed for each system,which will have inputs and expected output. Each unit would be then tested with the test cases to see if the actual output equals the expected output. If any bugs were to happen, changes would be made in the system and all test cases will be tested again using regression testing to make sure the bug was fixed as well as no new bugs will be introduced.

Photocell Circuit

To test the photocell circuit, the group had to write a code such that it would convert the measurements for the resistance and categorize it as either “Dark” or “Bright”. The group then tested the photocell circuit by running the code and seeing if measurements are printed out. Furthermore, the group then manipulated the light around the photocell resistor, which either makes it darker or lighter, and compare it with the measurements. For example, if the light around the photocell resistor is dark, then the printed results will be “Dark” and by increasing the light around the photocell resistor, the printed results will be “Bright”.

Relay Switch Circuit

To test the relay switch, a code was written to turn on or turn off the relay circuit through a local server by using a smartphone or computer to access the server. The inputs are either an on or off signal from the server, which should result in the relay circuit turning on or off .For example, if a lamp is plugged into the relay switch, the lamp should turn on when a user clicks the “on” button on the local server and turn off when the “off” button is clicked. This unit test for the relay switch circuit will provide an analysis of how well the relay switch could be controlled remotely using a server.

Magnetic Contact Switch Circuit

To test the magnetic contact switch circuit, the group created a code that will print ‘on’ or ‘off’ depending if the magnetic contacts were touching or not. The code will take input values from the magnetic contact switch. The expected value for if the magnetic contact switches were connected to each other would be “off”, while detached from each other would be “on”. The group would then compare these values to the actual result, and fix again bugs that would occur.

RF Transceiver

To test the RF transceiver, the group used the RF24 library to print the RF module’s system details after powering on. In the program code, the Tx/Rx addresses used for transmitter and receiver pipes were initialized to a known value. After booting up the RF transceiver, it prints the system details that include the pipe addresses for receiving and transmitting, power settings, data-rate, model number, and CRC length. The group then verified that the Tx/Rx addresses, power settings, data-rate, and CRC length were correct.

Database

To test the database, test cases that included queries for insert, update, and create was created for each table. Each test case had a query that will act as the input and an expected output. These test cases were executed through the database to check if the expected value was equal to the actual value. If the values were not equal to each other, changes to the queries would have to be done and regression testing would have to be done on database again.

Integration Testing

As each unit is done with their section of unit testing, each unit will be integrated into the system. This project will follow a sandwich approach for integration testing by using stubs and drivers.

Photocell and Database Integration

The group integrated the units to form an integrated system. For example, the group integrated the photocell and the database and performed integrated testing.The goal is to send data given from the photocell circuit which is on the local controller to the database, which will update the values in the table for the photocell. Before integration was done, unit testing was done on the database, which made sure the data could be inserted, updated, and removed. Unit tested was also done on the photocell to make sure it was outputting the correct measurements. The group integrated the photocell and the database by making a tcp/ip connection between the server and the local controller. The group was able to send the data over to the database from the photocell. Then, the results was verified by creating a simple query which selects all the data and displays the values in the photocell table. The values were then verified to determine if those values were inserted and updated by checking the timestamp of when it was dark or light to the timestamp and data in the database. Base on the data, the group was able to come to a conclusion that the photocell and the database were integrated successfully.

Contact Switch/Relay Circuit Database Integration

A similar approach will be taken when integrating the database with the contact switch and relay switch. Values will be sent from the local controller to the server via tcp/ip connection and the server will update the values into the database. Timestamps will be used to verify if the tables for the relay switch and contact switch was updated properly and in a timely matter.

Integrated System with Web-server Integration

After integrating the database with relay switch circuit, photocell circuit , and the contact switch circuit, the group will integrate the integrated system with the web-server to create our final system. The group will perform integration testing by verifying that the web-server can monitor and control the devices via the internet. Then, the group will verify if the server is still inserting and updating the correct values into the database.

System Testing

The group will test the system by testing to see if all the functionalities such as the automated power saving features, notification system, and the ability to monitor and control devices connected to the system. A stress test will also be done on the server to test how long the server can operate under load. Also, the functional and nonfunctional requirements from the specified requirement specification will be compared with the final system to see if the final system meets the requirements.

Regression Testing

Regression testing was used throughout the project. The purpose of regression testing was to verify the functionality of each unit and to verify whether new bugs were introduced when integrating each unit. For example, when the magnetic contact switch unit was integrated into the current system of the database and photocell, the test from the database and photocell units would need to be tested again to verify they remain functional and no new bugs were introduced. Additionally, when the relay circuit was integrated into the server, integration testing and regression testing was performed to verify that the unit test for the relay switch circuit was still valid, and that integrating the circuit into the system did not introduce new bugs. That is, the new functionalities of the relay circuit is functional with the server along with the, magnetic contact switch, database, and photocell units.

Table 17. Functional Requirements Traceability Matrix

Requirement Information			**Testing Results**			
Req. ID	**Test Case ID**	**Description**	**Input**	**Expected Output**	**Actual Output**	**Pass/Fail**
1,3,5,7,9 10,11,13, 14, 15,17,18 19,20, 21	PC_1	The system will measure amount of light in a room. If there is high intensity light, will turn off the lights	In a bright environment, with light (Device 1) turned on, change status of Light Control to on.	Light will turn off.	Light will turn off.	Pass
1,3,5,7,9 10,11,13, 14, 15,17,18 19,20, 21	PC_2	The system will measure amount of light in a room. If there is high intensity light, will turn off the lights	In a dark environment, with light (Device 1) turned on, change status of Light Control to off.	Light will stay on.	Light will stay on.	Pass
3,5,7,9, 10,11,12, 13,14,16, 17,18,19, 21	RC_1	The system will control devices connected to it.	Pressing "On" on the control webpage for Device 1	Device 1 turn on	Device 1 turn on	Pass
3,5,7,9,	RC_2	The system will	Pressing	Device 1 turn	Device 1 turn	Pass

10,11,12, 13,14,16, 17,18,19, 21		control devices connected to it	“Off” on the control webpage for Device 1	off	off	
3,5,7,9, 10,11,12, 13,14,16, 17,18,19, 21	RC_3	The system will control devices connected to it.	Pressing “On” on the control webpage for Device 2	Device 2 turn on	Device 2 turn on	Pass
3,5,7,9, 10,11,12, 13,14,16, 17,18,19, 21	RC_4	The system will control devices connected to it.	Pressing “Off” on the control webpage for Device 2	Device 2 turn off	Device 2 turn off	Pass
3,5,7,9, 10,11,12, 13,14,16, 17,18,19, 21	RC_5	The system will control devices connected to it.	Pressing “On” on the control webpage for Device 3	Device 3 turn on	Device 3 turn on	Pass
3,5,7,9, 10,11,12, 13,14,16, 17,18,19, 21	RC_6	The system will control devices connected to it.	Pressing “Off” on the control webpage for Device 3	Device 3 turn off	Device 3 turn off	Pass
1,2,4,5,7, 8,9,10,11 ,13,14,15 ,17,18, 20,21	CS_1	The system will be able to detect whether the door is opened or closed.	With “Door Control” status on, open the door.	An email will be sent to the user with date and time of when the switch was triggered.	An email will be sent to the user with date and time of when the switch was triggered.	Pass
1,2,4,5,7, 8,9,10,11 ,13,14,15 ,17,18, 20,21	CS_2	The system will be able to detect whether the door is opened or closed.	With “Door Control” status of, open the door.	There will be no emails sent to the user.	There will be no emails sent to the user.	Pass
4,9,10	WS_1	The web server will restrict access to unauthorized	Putting in the incorrect information as a login.	Display Incorrect information and Restrict	Display Incorrect information and Restrict	Pass

		users.		access to monitor and control pages.	access to monitor and control pages	
4,9,10,13	WS_2	The web server will allow access to authorized users.	Putting in the Correct information as a login	Provide access to the monitor and control page. Redirects to control page	Provide access to the monitor and control page. Redirects to control page	Pass
4,10,13	WS_3	The web server will allow users to log off.	After logging in, Choose “log off”	Redirects to home page. Restrict access to subsequent pages.	Redirects to home page. Restrict access to subsequent pages.	Pass
4,10,13	WS_4	The server will restrict access to certain pages if the user is not logged in.	Access monitor pages when not logged in.	Provide an error message and redirects to login page	Provide an error message and redirects to login page	Pass
10	WS_5	The server will restrict access to certain pages if the user is not logged in.	Access control pages when not logged in.	Provide an error message and redirects to login page	Provide an error message and redirects to login page	Pass
1,4,9,14, 15	WS_6	The user will be able to filter change logs	On the monitor page, choose “Device 1” under the “View Selected Device” section	Only display changelog for Device 1	Only display changelog for Device 1	Pass
1,4,9,14, 15	WS_7	The user will be able to filter change logs	On the monitor page, choose “Device 2” under the “View Selected	Only display changelog for Device 2	Only display changelog for Device 2	Pass

			Device" section			
1,4,9,14, 15	WS_8	The user will be able to filter change logs	On the monitor page, choose "Device 3" under the "View Selected Device" section	Only display changelog for Device 3	Only display changelog for Device 3	Pass
1,4,9,14, 15	WS_9	The user will be able to filter change logs	On the monitor page, choose "All Devices" under the "View Selected Device" section	Changelog for Device 1, Device 2, and Device 3 are displayed	Changelog for Device 1, Device 2, and Device 3 are displayed	Pass
1,4,9,14, 15	WS_10	The user will be able to filter change logs	On the monitor page, choose "Device Changes" under the "View Selected Device" section	Display changes to the connection to the Devices	Display changes to the connection to the Devices	Pass
1,4,9,14, 15	WS_11	The user will be able to filter change logs	On the monitor page, choose "Door Notification" under the "View Selected Device" section	Display changelog for "Door Control"	Display changelog for "Door Control"	Pass
1,4,9,14,	WS_12	The user will be	On the	Display	Display	Pass

15		able to filter change logs	monitor page, choose “Light Controls” under the “View Selected Device” section	changelog for “Light Control”	changelog for “Door Control”	
1,4,9,14, 15	WS_13	The user will be able to filter change logs	On the monitor page, choose “login”	Display changelog for “login”	Display changelog for “login”	Pass
1,4,9,14, 15	WS_14	The user will be able to filter change logs	On the monitor page, choose “View All” under the “View Selected Device” section	Display everything in the change log	Display everything in the change log	Pass
1,4,9,14, 15	WS_15	The user will be able to clear changelog of Device 1.	On the monitor page, choose “Device 1 ” under the “Clear Selected Device” section	Clear changelog of “Device 1”	Clear changelog of “Device 1”	Pass
1,4,9,14, 15	WS_16	The user will be able to clear changelog of Device 2.	On the monitor page, choose “Device 2” under the “Clear Selected Device” section	Clear changelog of “Device 2”	Clear changelog of “Device 2”	Pass
1,4,9,14, 15	WS_17	The user will be able to clear	On the monitor	Clear changelog of	Clear changelog of	Pass

		changelog of Device 3.	page, choose "Device 3 " under the "Clear Selected Device" section	"Device 3"	"Device 3"	
1,4,9,14, 15	WS_18	The user will be able to clear changelog of "All Device "	On the monitor page, choose "All Device " under the "Clear Selected Device" section	Clear changelog of "All Devices"	Clear changelog of "All Devices"	Pass
1,4,9,14, 15	WS_19	The user will be able to clear changelog of "Device Changes "	On the monitor page, choose "Device Changes " under the "Clear Selected Device" section	Clear changelog of "Device changes"	Clear changelog of "Device changes"	Pass
1,4,9,14, 15	WS_20	The user will be able to clear changelog of "Device Controls"	On the monitor page, choose "Door Control" under the "Clear Selected Device" section	Clear changelog of "Door Control"	Clear changelog of "Door Control"	Pass
1,4,9,14, 15	WS_21	The user will be able to clear changelog of "Light Control"	On the monitor page, choose "Light Controls " under the	Clear changelog of "Light Controls"	Clear changelog of "Light Controls"	Pass

			“Clear Selected Device” section			
1,4,14	WS_22	The user will be able to order the devices alphabetically.	On the monitor page click “Device” on the change log	The changelog is order by devices alphabetically	The changelog is order by devices alphabetically	Pass
1,4,14	WS_23	The user will be able to order the status alphabetically.	On the monitor page click “Status” on the change log	The changelog is order by status alphabetically	The changelog is order by status alphabetically	Pass
1,4,14	WS_24	The user will be able to order the changelog by time.	On the monitor page click “Date/Time” on the change log	The changelog is order by time.	The changelog is order by time.	Pass
1,4,10,14	WS_25	User will be able to navigate to the monitor page.	After logging in, click on the monitor tab.	Go to the monitor page	Go to the monitor page	Pass
1,4,10,14	WS_26	User will be able to navigate to the control page.	After logging in, click on the control tab.	Go to the control page	Go to the control page	Pass
1,4,10,14	WS_27	User will be able to navigate to the about page.	After logging in, click on the about tab.	Go to the about page	Go to the about page	Pass
1,4,10,14	WS_28	User will be able to navigate to the info page.	After logging in, click on the info tab.	Go to the info page	Go to the info page	Pass
1,4,10,13 ,14,15	WS_29	Users will be able to edit the name of the devices.	After logging in, in the control page, edit text box	Device 1 will named as fan	Device 1 is named as fan	Pass

			for device 1 to “Fan”			
1,4,10,13 ,14,15	WS_30	Users will be able to edit the name of the devices.	After logging in, in the control page, edit text box for device 2 to “Fan”	Device 2 is named as fan	Device 2 is named as fan	Pass
1,4,10,13 ,14,15	WS_31	Users will be able to edit the name of the devices.	After logging in, in the control page, edit text box for device 3 to “Fan”	Device 3 is named as fan	Device 3 will named as fan	Pass
1,4,10,13 ,14.15	WS_32	The web-server will be able to communicate with the database.	After logging in, in the control page, Press “on” for Device 1	The status of Device 1 is “on” regardless of the previous status and the status of Device 1 in the database is on	The status of Device 1 is “on” regardless of the previous status and the status of Device 1 in the database is on	Pass
4,10,13, 14.15	WS_33	The web-server will be able to communicate with the database.	After logging in, in the control page, Press “on” for Device 2	The status of Device 1 is “on” regardless of the previous status and the status of Device 2 in the database is on	The status of Device 1 is “on” regardless of the previous status and the status of Device 2 in the database is on	Pass
4,10,13, 14.15	WS_34	The web-server will be able to communicate with the database.	After logging in, in the control page, Press “on” for Device 3	The status of Device 1 is “on” regardless of the previous status and the status of Device 3 in	The status of Device 1 is “on” regardless of the previous status and the status of Device 3 in	Pass

				the database is on	the database is on	
4,10,13, 14.15	WS_35	The web-server will be able to communicate with the database.	After logging in, in the control page, Press “on” for Door Control	The status of Device 1 is “on” regardless of the previous status and the status of Door Control in the database is “on”	The status of Device 1 is “on” regardless of the previous status and the status of Door Control in the database is “on”	Pass
4,10,13, 14.15	WS_36	The web-server will be able to communicate with the database.	After logging in, in the control page, Press “on” for Light Control	The status of Light Control is “on” regardless of the previous status and the status of Device 3 in the database is “off”	The status of Light Control is “on” regardless of the previous status and the status of Device 3 in the database is “off”	Pass
4,10,13, 14.15	WS_37	The web-server will be able to communicate with the database.	After logging in, in the control page, Press “off” for Device 1	The status of Device 1 is “off” regardless of the previous status and the status of Device 1 in the database is “off”	The status of Device 1 is “off” regardless of the previous status and the status of Device 1 in the database is “off”	Pass
4,10,13, 14.15	WS_38	The web-server will be able to communicate with the database.	After logging in, in the control page, Press “off” for Device 2	The status of Device 2 is “off” regardless of the previous status and the status of Device 1 in the database is “off”	The status of Device 2 is “off” regardless of the previous status and the status of Device 1 in the database is “off”	Pass

4,10,13, 14.15	WS_39	The web-server will be able to communicate with the database.	After logging in, in the control page, Press “off” for Device 3	The status of Device 3 is “off” regardless of the previous status and the status of Device 1 in the database is “off”	The status of Device 3 is “off” regardless of the previous status and the status of Device 1 in the database is “off”	Pass
4,10,13, 14.15	WS_40	The web-server will be able to communicate with the database.	After logging in, in the control page, Press “off” for Door Control	The status of Door Control is “off” regardless of the previous status and the status of Door Control in the database is “off”	The status of Door Control is “off” regardless of the previous status and the status of Door Control in the database is “off”	Pass
4,10,13, 14.15	WS_41	The web-server will be able to communicate with the database.	After logging in, in the control page, Press “off” for Light Control	The status of Light Control is “off” regardless of the previous status and the status of Light Control in the database is “off”	The status of Light Control is “off” regardless of the previous status and the status of Light Control in the database is “off”	Pass
3,5,6, 7,9, 10,11, 12,13,14, 16,17,18, 19,21	WS_42	The server will be able to handle multiple connection simultaneously.	User 1 and User 2 will log into the website. User 1 will turn on Device 1 and User 2 will turn on Device 2.	Device 1 and Device 2 will be turned on.	Device 1 and Device 2 will be turned on.	Pass

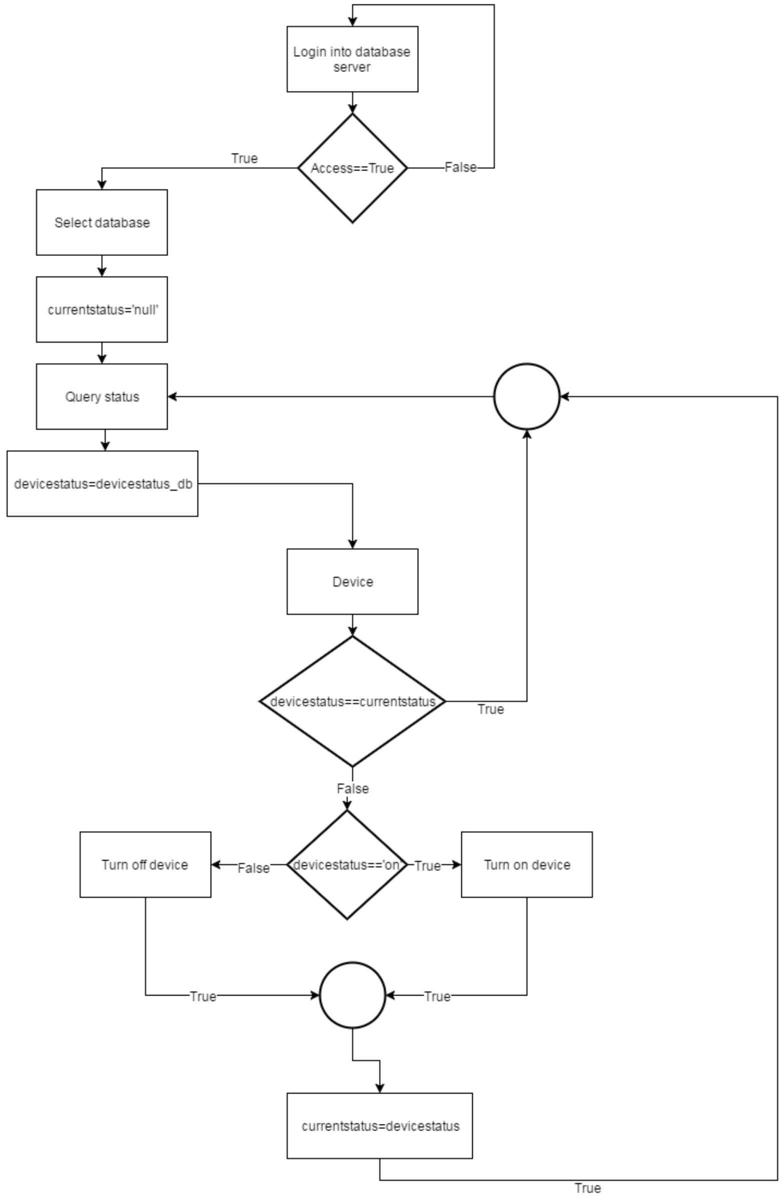

Figure 18. Control flow of the relay circuit

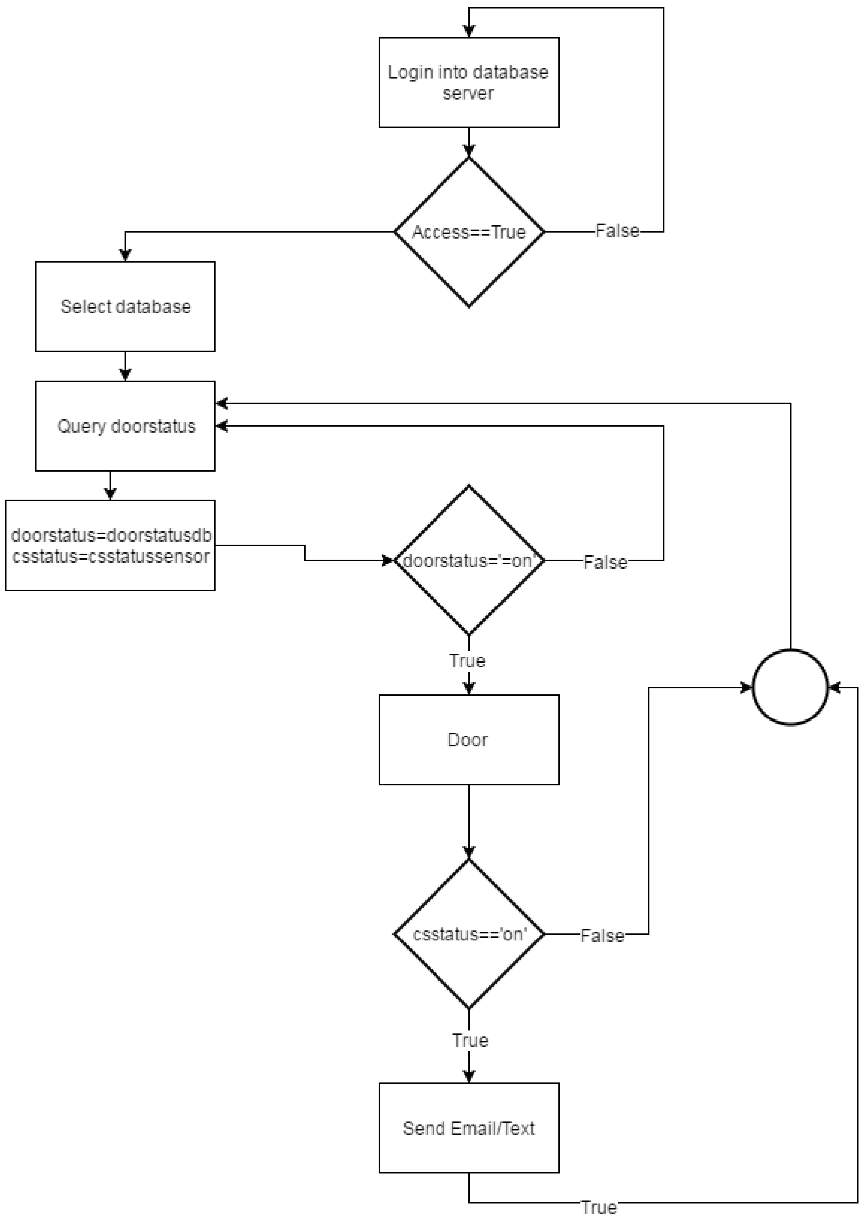

Figure 19. Control flow of the contact switch

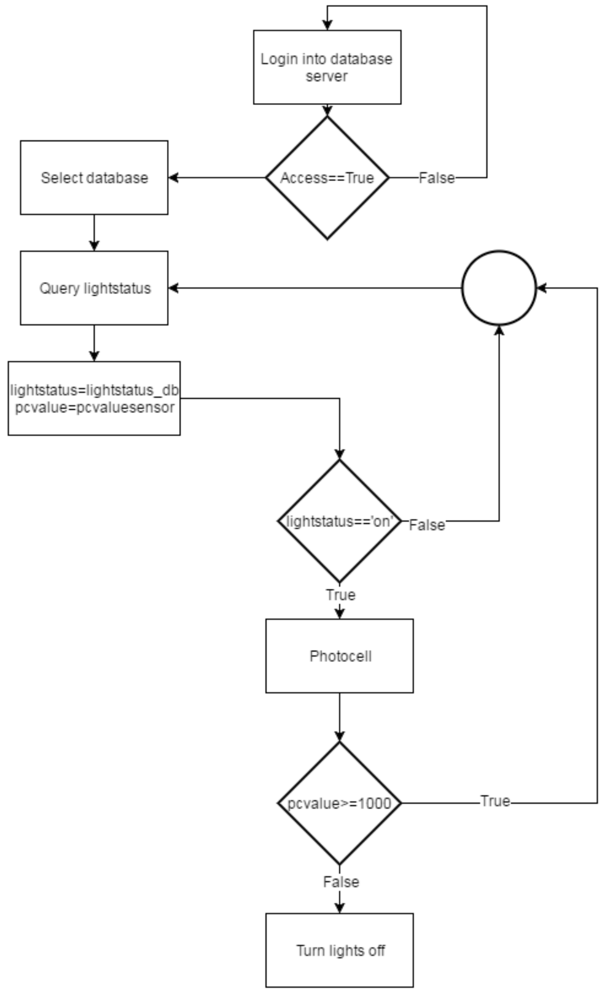

Figure 20. Control flow of photocell

7.1 Functional Testing

Table 18. PC_1 Test Case

ID	PC_1
Title	Monitor light intensity
Steps	1. Turn on light control in the website 2. Turn on light or use natural sunlight pointed towards photocell
Expected Result	Device 1 turns off.
Pass/Fail	Pass

Table 19. PC_2 Test Case

ID	PC_2
Title	Monitor light intensity
Steps	1. Turn off light control in the website 2. Turn on light or use natural sunlight pointed towards photocell
Expected Result	Device 1 remains on.
Pass/Fail	Pass

Table 20. RC_1 Test Case

ID	RC_1
Title	Turn on Device 1
Steps	1. Plug in Receiver #1 in outlet. 2. Go to Control section of the website using Phone or Computer. 3. Press the “Device 1” on button.
Expected Result	Device 1 turns on.
Pass/Fail	Pass

Table 21. RC_2 Test Case

ID	RC_2
Title	Turn off Device 1
Steps	1. Plug in Receiver #1 in outlet. 2. Go to Control section of the website using Phone or Computer. 3. Press the “Device 1” off button.
Expected Result	Device 1 turns off.
Pass/Fail	Pass

Table 22. RC_3 Test Case

ID	RC_3
Title	Turn on Device 2
Steps	4. Plug in Receiver #2 in outlet. 5. Go to Control section of the website using Phone or Computer. 6. Press the “Device 2” on button.
Expected Result	Device 1 turns on.
Pass/Fail	Pass

Table 23. RC_4 Test Case

ID	RC_4
Title	Turn off Device 2
Steps	1. Plug in Receiver #2 in outlet. 2. Go to Control section of the website using Phone or Computer. 3. Press the “Device 2” off button.
Expected Result	Device 1 turns off.
Pass/Fail	Pass

Table 24. RC_5 Test Case

ID	RC_5
Title	Turn on Device 3
Steps	1. Plug in Receiver #3 in outlet. 2. Go to Control section of the website using Phone or Computer. 3. Press the “Device 3” on button.
Expected Result	Device 3 turns on.
Pass/Fail	Pass

Table 25. RC_6 Test Case

ID	RC_6
Title	Turn off Device 3
Steps	1. Plug in Receiver #3 in outlet. 2. Go to Control section of the website using Phone or Computer. 3. Press the “Device 3” off button.
Expected Result	Device 3 turns off.
Pass/Fail	Pass

Table 26. CS_1 Test Case

ID	CS_1
Title	Check to see if door has been opened and if users will be notified.
Steps	1. Connect contact switch to door 2. Go to control section of website using Phone or Computer. 3. Set status of Door Notification to “on” 4. Open Door
Expected Result	Email will be sent notifying users that the door is opened.
Pass/Fail	Pass

Table 27. CS_2 Test Case

ID	CS_2
Title	Check to see if door has been opened and if users will be notified.
Steps	1. Connect contact switch to door 2. Go to control section of website using Phone or Computer. 3. Set status of Door Notification to “off” 4. Open Door
Expected Result	There will be no notifications for the users.
Pass/Fail	Pass

Table 28. WS_1 Test Case

ID	WS_1
Title	Check correct login info
Steps	1. Go to website 2. Log in into the website with correct username and password.
Expected Result	Session created and moved to control page
Pass/Fail	Pass

Table 29. WS_2 Test Case

ID	WS_2
Title	Check incorrect login info
Steps	1. Go to website 2. Log in into the website with incorrect username and password.
Expected Result	Error message appears and will stay on login page
Pass/Fail	Pass

Table 30. WS_3 Test Case

ID	WS_3
Title	Logoff
Steps	1. Login into website 2. Click “Log Off” button
Expected Result	Session will close and will be moved to login page
Pass/Fail	Pass

Table 31. WS_4 Test Case

ID	WS_4
Title	Deny access to control page for users not logged in
Steps	1. Go to website 2. Go to control page
Expected Result	Error message and will be moved to login page
Pass/Fail	Pass

Table 32. WS_5 Test Case

ID	WS_5
Title	Deny access to monitor page for users not logged in
Steps	1. Go to website 2. Go to monitoring page.
Expected Result	Error message and will be moved to login page.
Pass/Fail	Pass

Table 33. WS_6 Test Case

ID	WS_6
Title	Filter only Device 1 changelog
Steps	1. Go to Monitor section on the website 2. Press the “Device 1” button in view selected section
Expected Result	Only Device 1 changelog is shown
Pass/Fail	Pass

Table 34. WS_7 Test Case

ID	WS_7
Title	Filter only Device 2 changelog
Steps	1. Go to Monitor section on the website 2. Press the “Device 2” button in view selected section
Expected Result	Only Device 2 changelog is shown
Pass/Fail	Pass

Table 35. WS_8 Test Case

ID	WS_8
Title	Filter only Device 3 changelog
Steps	1. Go to Monitor section on the website 2. Press the “Device 3” button in view selected section
Expected Result	Only Device 3 changelog is shown
Pass/Fail	Pass

Table 36. WS_9 Test Case

ID	WS_9
Title	Filter all devices in the changelog
Steps	1. Go to Monitor section on the website 2. Press the “All Devices” button in view selected section
Expected Result	Changelog for Device 1, 2, and 3 changelog is shown
Pass/Fail	Pass

Table 37. WS_10 Test Case

ID	WS_10
Title	Filter device changes in changelog
Steps	1. Go to Monitor section on the website 2. Press the “Device Changes” button in view selected section
Expected Result	Only changes to the connection of devices is shown
Pass/Fail	Pass

Table 38. WS_11 Test Case

ID	WS_11
Title	Filter only door changes in the changelog
Steps	1. Go to Monitor section on the website 2. Press the “Door Notification” button in view selected section
Expected Result	Only changes to the door notification is shown
Pass/Fail	Pass

Table 39. WS_12 Test Case

ID	WS_12
Title	Filter only light controls in the changelog
Steps	1. Go to Monitor section on the website 2. Press the “Light Controls” button in view selected section
Expected Result	Only changes to the light controls is shown
Pass/Fail	Pass

Table 40. WS_13 Test Case

ID	WS_13
Title	Filter only the login in the changelog
Steps	1. Go to Monitor section on the website 2. Press the “Login” button in view selected section
Expected Result	Only login info is shown.
Pass/Fail	Pass

Table 41. WS_14 Test Case

ID	WS_14
Title	Show everything in the changelog
Steps	1. Go to Monitor section on the website 2. Press the “View all” button in view selected section
Expected Result	Every change is shown in changelog
Pass/Fail	Pass

Table 42. WS_15 Test Case

ID	WS_15
Title	Delete rows about device 1 in changelog
Steps	1. Go to Monitor section on the website 2. Press the “Device 1” button in clear selected device
Expected Result	Delete all rows with Device 1
Pass/Fail	Pass

Table 43. WS_16 Test Case

ID	WS_16
Title	Delete rows about device 2 in changelog
Steps	1. Go to Monitor section on the website 2. Press the “Device 2” button in clear selected device
Expected Result	Delete all rows with Device 2
Pass/Fail	Pass

Table 44. WS_17 Test Case

ID	WS_17
Title	Delete rows about device 3 in changelog
Steps	1. Go to Monitor section on the website 2. Press the “Device 3” button in clear selected device
Expected Result	Delete all rows with Device 3
Pass/Fail	Pass

Table 45. WS_18 Test Case

ID	WS_18
Title	Delete rows about all devices in changelog
Steps	1. Go to Monitor section on the website 2. Press the “All Devices” button in clear selected device
Expected Result	Delete rows associated with each device.
Pass/Fail	Pass

Table 46. WS_19 Test Case

ID	WS_19
Title	Delete rows about device connection changes in changelog
Steps	1. Go to Monitor section on the website 2. Press the “Device Changes” button in clear selected device
Expected Result	Delete rows with changes to devices
Pass/Fail	Pass

Table 47. WS_20 Test Case

ID	WS_20
Title	Delete rows about door notification in changelog
Steps	1. Go to Monitor section on the website 2. Press the “Door Notification” button in clear selected device
Expected Result	Delete rows about door notification
Pass/Fail	Pass

Table 48. WS_21 Test Case

ID	WS_21
Title	Delete rows about light controls in changelog
Steps	1. Go to Monitor section on the website 2. Press the “Light Controls” button in clear selected device
Expected Result	Delete rows about Light controls
Pass/Fail	Pass

Table 49. WS_22 Test Case

ID	WS_22
Title	Order by device
Steps	1. Go to Monitor section on the website 2. Click “Device” on the changelog
Expected Result	Device will be ordered alphabetically
Pass/Fail	Pass

Table 50. WS_23 Test Case

ID	WS_23
Title	Order by status
Steps	1. Go to Monitor section on the website 2. Click “Status” on the changelog
Expected Result	Status will be ordered alphabetically
Pass/Fail	Pass

Table 51. WS_24 Test Case

ID	WS_24
Title	Order by date/time
Steps	1. Go to Monitor section on the website 2. Click “Date/Time” on the changelog
Expected Result	Device will be ordered by time.
Pass/Fail	Pass

Table 52. WS_25 Test Case

ID	WS_25
Title	Monitor Page
Steps	1. Log into website 2. Click on “Monitor” on navigation page
Expected Result	Will go to Monitor page
Pass/Fail	Pass

Table 53. WS_26 Test Case

ID	WS_26
Title	Control Page
Steps	1. Log into website 2. Click on “Control” on navigation page
Expected Result	Will go to Control page
Pass/Fail	Pass

Table 54. WS_27 Test Case

ID	WS_27
Title	About Page
Steps	1. Log into website 2. Click on “About” on navigation page
Expected Result	Will go to About page
Pass/Fail	Pass

Table 55. WS_28 Test Case

ID	WS_28
Title	Info Page
Steps	1. Log into website 2. Click on “Info” on navigation page
Expected Result	Will go to Info page
Pass/Fail	Pass

Table 56. WS_29 Test Case

ID	WS_29
Title	Update Device 1
Steps	1. Log into website 2. Go to Control Page 3. Edit Textbox for device 1 4. Click Update
Expected Result	Title Updated to new value from textbox
Pass/Fail	Pass

Table 57. WS_30 Test Case

ID	WS_30
Title	Update Device 2
Steps	1. Log into website 2. Go to Control Page 3. Edit Textbox for device 2 4. Click Update
Expected Result	Title Updated to new value from textbox
Pass/Fail	Pass

Table 58. WS_31 Test Case

ID	WS_31
Title	Update Device 3
Steps	1. Log into website 2. Go to Control Page 3. Edit Textbox for device 3 4. Click Update
Expected Result	Title Updated to new value from textbox
Pass/Fail	Pass

Table 59. WS_32 Test Case

ID	WS_32
Title	Update Device 1 to on
Steps	1. Log into website 2. Go to Control Page 3. Click on “on” for Device 1
Expected Result	If device 1 was off , database will update device 1 to on
Pass/Fail	Pass

Table 60. WS_33 Test Case

ID	WS_33
Title	Update Device 2 to on
Steps	1. Log into website 2. Go to Control Page 3. Click on “on” for Device 2
Expected Result	If device 2 was off , database will update device 2 to on
Pass/Fail	Pass

Table 61. WS_34 Test Case

ID	WS_34
Title	Update Device 3 to on
Steps	1. Log into website 2. Go to Control Page 3. Click on “on” for Device 3
Expected Result	If device 3 was off , database will update device 3 to on
Pass/Fail	Pass

Table 62. WS_35 Test Case

ID	WS_35
Title	Update Door control to on
Steps	1. Log into website 2. Go to Control Page 3. Click on “on” for Door Control
Expected Result	If Door control was off , database will update Door control to on
Pass/Fail	Pass

Table 63. WS_36 Test Case

ID	WS_36
Title	Update light control to on
Steps	1. Log into website 2. Go to Control Page 3. Click on “on” for Light Controls
Expected Result	If Light control was off , database will update light control to on
Pass/Fail	Pass

Table 64. WS_37 Test Case

ID	WS_37
Title	Update Device 1 to off
Steps	1. Log into website 2. Go to Control Page 3. Click on “off” for Device 1
Expected Result	If device 1 was on , database will update device 1 to off
Pass/Fail	Pass

Table 65. WS_38 Test Case

ID	WS_38
Title	Update Device 2 to off
Steps	1. Log into website 2. Go to Control Page 3. Click on “off” for Device 2
Expected Result	If device 2 was on, database will update device 2 to off
Pass/Fail	Pass

Table 66. WS_39 Test Case

ID	WS_39
Title	Update Device 3 to off
Steps	1. Log into website 2. Go to Control Page 3. Click on “off” for Device 3
Expected Result	If device 3 was on, database will update device 3 to off
Pass/Fail	Pass

Table 67. WS_40 Test Case

ID	WS_40
Title	Update Door control to off
Steps	1. Log into website 2. Go to Control Page 3. Click on “off” for Door Notification
Expected Result	If Door control was on , database will update Door control to off
Pass/Fail	Pass

Table 68. WS_41 Test Case

ID	WS_41
Title	Update light control to off
Steps	1. Log into website 2. Go to Control Page 3. Click on “off” for Light Controls
Expected Result	If Light control was on , database will update light control to off
Pass/Fail	Pass

Table 69. WS_42 Test Case

ID	WS_42
Title	Simultaneous Connection
Steps	1. User 1 logs in website 2. User 2 logs in website 3. User 1 turns on Device 1 4. User 2 turns off Device 1 5. User 1 turns on Device 2 6. User 2 turns on Device 3 7. User 1 turns off Device 3 8. User 2 turns off Device 2

Expected Result	User 1 and User 2 should be able to simultaneously control the devices in the control page.
Pass/Fail	Pass

7.2 Non-Functional Testing

Table 70. Non-functional testing traceability matrix

Req_ID	Description	Testing	Results	Pass/Fail
22	The system shall be reliable.	The system will be left on for a week. In order to pass, the system must not be down for more than 15 minutes.	The system was down for more than 15 minutes. The local controller crashed after 4 days.	Fail
23	The system will have a quick response to user inputs.	Inputs will be given to the system and response time will be measured in milliseconds. Average response time must be less than 1 second to pass.	The average response time for all 3 devices is less than 1 second. Results are displayed in the Table 66 and Table 67 .	Pass
24	The system will have a friendly and easy to use user interface.	Random people will be chosen to test the interface. Users will be asked to navigate to specific sections of the website. At Least 80% of users will be able to navigate	90% of the people chosen were able to navigate correctly around the website.	Pass

		correctly around the website for it to pass.		
23	The system will be able to perform despite being under load.	AB will be run on the web server to check the performance of the server underload. The result given from AB will decide whether the system pass.	Have not been performed yet.	Unknown
22	The system will be able to recover after a crash.	The power to the local controller will be removed and then plugged back in to simulate a power outage. If the Pi can recover from this, it will determine if it passed the test.	The local controller was able to reconnect to the database and server successfull y and continue to perform it’s regular functions.	Pass

Chapter 8 Performance and Benchmarks

8.1 Performance Characteristics

The performance characteristics that were measured and observed in this project were accuracy, repeatability, and response time. The group used accuracy as a characteristic to observe the accuracy of the values received from the system. This observation was used to determine whether the expected output matched the actual output of the system. Additionally, response time was another characteristic used since its measurements provided an analysis on how well the system responded to a user input.

8.1.1 Accuracy

In this project, the accuracy of the website, photocell, and contact switch were observed. The accuracy of these units were important since the project had a non-functional requirement of reliability. The accuracy of the website was observed by comparing the expected output with the actual output. For example, when a user clicks the on button on Device 1 of the website, the database should read device 1 turn on and the local controller should communicate with the radio frequency of transmitter with the receiver for device 1 to turn on. For the photocell, accuracy for the resistance reading was important since the group did not want the light to turn off when the room is actually dark. This required consistent calibration with the photocell. Lastly, for the contact switch, its accuracy was observed by opening and closing a door. This accuracy was important since the group did not want to inconvenience the user by notifying them if there was a false reading in the system. By using accuracy as a performance characteristic, the repeatability of the system was also observed.

8.1.2 Response time

The response time characteristic was used to observe the response time of the website and the relay circuit with given inputs. This observation was used to analyze the communication between the units of the system and possible problems that needed to be addressed. The response time for both the website and the relay circuit were tested under five samples with multiple inputs to measure an average response time.

The response time of the website was measured using a browser plugin on Google Chrome, "Page load time." This plugin measured the loading time of the pages in the website in milliseconds. In Table 71, the response times of the website are shown along with the average response time of each page, which consists of navigating the website and sending request information from the control page. After calculating the average response time of each of the pages and requests, the response time of the pages that contained more information took the longest was observed. For example, the monitoring page of the website took the longest to respond since it contained many SQL queries to display on the screen.

Furthermore, the response time of the relay circuit was measured using a stopwatch timer. It was noted that there were inaccuracy in measuring the response time since the human response of pressing the start and stop button on the stopwatch contributed to partial error. The relay circuit of the project used an radio frequency transmitter and a receiver, but only the time of the transmitter could be measured from software side. Since the receiver was wireless, the team could not determine a definite way of measuring its response time. The relay circuit response time are shown in Table 72 with the calculated average response time. This average response time of the relay circuit provided information on the distance the radio frequency could travel and whether it was lost. When measuring the response time of the relay circuit, device 1 was closest to the transmitter and device 2 was furthest from the transmitter. In one of the test samples, the response time of device 2 measured to be 1.5 seconds and the group determined that obstacles and distance away from the transmitter was a factor. In order to get a better and a much improve response time, the transmitter and receiver should be placed without obstacles for a free communication.

Table 71. Website Response Time

	Response Time (ms)					
	Run 1	**Run 2**	**Run 3**	**Run 4**	**Run 5**	**Average Time**
Accessing Website	300	550	330	460	540	436
Log-in	290	430	410	610	430	434
Accessing Control Page	150	190	200	180	230	190
Button-click	160	160	130	200	170	164
Accessing Monitor Page	320	460	490	1500	500	654
Accessing About Page	110	110	110	110	110	110
Accessing Info Page	100	110	120	100	100	106
Log-off	320	260	420	180	200	276

Table 72. Relay Circuit Response Time

	Response Time (ms)					
	Run 1	**Run 2**	**Run 3**	**Run 4**	**Run 5**	**Average Time**
Device 1 On	640	320	320	440	480	440
Device 1 Off	750	480	610	480	880	640
*Device 2 On	1810	810	950	870	2730	1434
*Device 2 Off	1200.00	1020	880	690	980	954
Device 3 On	1200	670	980	540	480	774
Device 3 Off	540	1640	630	870	320	800

* Device 2 was furthest from the transmitter

8.2 Issues and Potential Solutions

The relay circuit requires communication between the transmitter and receiver by using radio frequency signals. Obstacles like doors and walls can interfere with the communication between the transmitter and receiver. This can result in loss of signals or a slow response time as analyzed from the performance characteristic of response time. The loss of signal and response time of the relay circuit is important since the radio frequency signals are required to communicate between the transmitter and the receiver. If the radio frequency signals are lost, then the user input will not match the expected output, which will be unreliable for controlling the devices in the system. This problem can be fixed by limiting the number of obstacles between the transmitter and receivers. Additionally, the transmitter could be positioned at the center of the receivers to provide a better response time and a shorter distance between the receivers. In the condition that radio frequency signals are lost, the relay circuit code could also be modified to always send the requested signal from the transmitter to the receiver. The only trade-off from this solution is the response time since signals can be lost, but still received at a longer time.

The photocell resistance readings are potential issues to the project since all photocells do not have the same calibration and each can measure different resistances under the same environment. This photocell circuit in this system is used as a power saver to control and monitor the relay controlling the lights. The current calibration on the photocell turns off the lights when the photocell measures resistance below 1000Ω. However, this is still problematic since when the light is turned on in the evening, the photocell could measure low resistance, which can turn off the light and prove to be inconvenient. However, this problem could be solved by turning off the light control in the website to prevent the photocell from turning off the light. The problem can also be solved by calibrating the photocell to match the value measured in the operating environment. This calibration could take several days to obtain the optimum calibration since the measurements of the resistance differ for each day and each night.

Chapter 9 Applied Knowledge from Related Coursework

9.1 Math

Generally, engineers should have a strong background in math. Many different math courses were taken while proceeding towards a Computer Engineering degree. These math courses taught the group how to calculate values from given information. For example, when designing and testing circuits, the group would have information on the voltage and resistances used in the circuits. With this information the group would be able to calculate the expected current of the circuit and compare it to the actual current. Additionally, Math 42 or Discrete Mathematic, has provided the group the necessary knowledge of Boolean Algebra that was applied to the project. The group's project applied Boolean Algebra to coding in python, which utilized true and false statements. Boolean Algebra was also applied to the I/O of the circuits that utilized high and low as logical 1's and 0's respectively.

9.2 Science

To better identify the potential problems that the group's project may solve, science-related courses were examined. One of these courses was Engineering 100W - Engineering Reports, that informed the group the dangers in the environment, such as carbon-dioxide (CO_2) emissions. CO_2 is emitted by consuming fossil fuels for energy. For example, electricity is generated from the use of fossil fuels and electricity is one of the main contributor to CO_2 emission in the Earth's atmosphere. This rapid increase of emission results in climate change. In this course, the group was also informed on methods to reduce CO_2 emission by simply using less or by using renewable energy, such as solar, wind, or water. This project aim to address the issue of the high electricity consumption in homes by implementing a system that monitors and controls devices and appliances in the home remotely. Through the ability to monitor and control devices remotely, idle energy consumption at home could be reduced. That is, homeowners will be able to identify what appliances are consuming energy and turn off and the unwanted appliances while absent from their homes. This project contributes to the environment by simply utilizing renewable energy and reducing energy consumption.

9.3 Engineering

The majority of the concepts in the Computer Engineering courses that the group have taken were applied in this project. For example, programming in C/C++ was learned in CmpE 30, CmpE 50, and CmpE 126. Additionally, assembly programming was learned in CmpE 102, CmpE 140, and CmpE 152. These courses provided the group the foundation of programming. From the knowledge learned in these courses, it was easier to understand newer languages required to implement the project, such as Python, HTML, CSS, and PHP. The sensors in this project were programmed using Python and the server is designed using HTML, CSS, and PHP. Furthermore, other courses that were applied to this project were CmpE 110, CmpE 127, and CmpE 138. In CmpE 110, the group learned the concepts of electronics in embedded system, which allowed better understanding of circuits, sensors, and transistors. Understanding how circuits worked allowed the group to design a relay circuit using transistors to act as switches for the devices. In CmpE 127, the group learned how to apply the knowledge learned in CmpE 110 to develop a microprocessor, which helped understand the Raspberry Pi development board. The course also taught the group how to use the GPIO of a microprocessor to control sensors needed for the project. Furthermore, in CmpE 138, the group learned how to create a database using SQL. The database was essential to the project since the notification system of the project was dependent on whether the status of the devices were concurrent with the status on the database.

The database was also used to send the signals to relay circuit from the server. In CMPE 131, Software Engineering I, the group learned about the software development process and the techniques to manage the project. In Cmpe 187, Software Quality Engineering, the group learned concepts of testing such as unit testing, integration testing, regression testing, and system testing. These techniques were widely applied to the project since units were developed and integrated. Integration testing was used to verify and validate the functionalities of each unit as an integrated system. Regression testing and system testing was required to make sure that integration and further developing did not change any of the intended functionalities.

Chapter 10 Summary, Conclusions, and Recommendations

10.1 Summary

The group implemented a smart home system that utilized the IoT technology by using a web server, database, and a local controller with sensor peripherals. The web server was implemented using Amazon Web Server and the database was implemented using SQL. The local controller was a microprocessor, Raspberry Pi, which communicated with sensor peripherals. The group utilized the GPIO ports of the Raspberry Pi for the sensors to allow communication with the website and database. These sensors consisted of a photocell to measure light intensity in a room, a door sensor to determine whether a window or door was opened, and a relay circuit which was used to control specified devices.

The website, database, local controller, and sensors communicated with each other to obtain requests from users to perform the functionalities of the project. These functionalities were controlling and monitoring devices in a home environment remotely through the computer or a smartphone.

The photocell circuit was used to measure light intensity in a room. It was used to implement the automated power saving functions. The photocell would measure the light intensity. If the light intensity is high, it would communicate with the database and change the status of the "lights" to off. If the measurement of the light intensity is low, it would do nothing but continue measuring light intensity.

The wireless door sensor was used to detect openings and closing of a door. The wireless feature was implemented by creating a wireless node using a low powered Arduino Mini Pro and a RF transceiver to talk with a RF base station at the local controller side. Attaching a magnetic contact switch to an interrupt pin of the Arduino Mini Pro provides the wireless door sensor the capability to wake itself up and alert the local controller of a door opening event. In an event of a door opening, the door sensor alerted the local controller by sending a payload, “open”, over a RF channel. The local controller process the payload and if the door notification feature was active, it sends an email containing a timestamp of the door event to the home user.

The relay circuit was the unit used to control the devices remotely by using radio frequency transmitter and receivers. This circuit also utilized transistors as switches to control the flow of current to the receiver. If a signal to turn on a device was received from the website, the current would flow from the base of the transistor to the collector. On the other hand, if a signal to turn off a device was received, the current would sink from the base of the transistor to the emitter. In this project, three receivers were used to control three devices.

The group performed testing for verification and validation of the project. Theses test consisted of unit testing, integration testing, system testing, and regression testing. White-box testing was used to review a code that another member has written to analyze possible improvements, better efficiency to a code, and possible bugs to the code. Once a code was analyzed, the team performed black-box testing to run the code to compare the expected output with the actual output. A traceability matrix was created to keep track of the test cases used during testing. This also allowed ease of use for tracking which test cases needed to be performed again during regression testing. Passing each of the test was important since it was a measure to how well the group validated the verified the project’s specification requirements.

10.2 Conclusions

Internet of Things is an emerging technology that is becoming more relevant to the future of many devices today. The group was able to implement and complete a system which utilized the IoT technology to control and monitor devices at home. The completed prototype of the projected was able to remotely control and monitor devices. The functionalities consisted of controlling three selected devices using the three receivers of the relay circuit, a magnetic contact switch to detect whether a door was opened, and a photocell to monitor whether a light intensity in the room. A notification system was also implemented to email the user when a door is opened while away from home. The project's website allowed the user to control and monitor their devices remotely through authentication of logging in with their given username and given password. If the user is unable to authenticate himself, the page would reload. The website included a login screen, a monitor page, a control page, an about page, and an information page. In the monitor page, the activities a user has performed are shown, such as turning on a device or turning off a device. In the control page, the user can control the three devices, control the light power saving mode, as well as control notification from door activity. The about page provided a detailed information about the group's project and its motivation and in the information page, detailed information about the group are provided. The completed hardware of the project was finished in a breadboard which displayed the necessary components of the device, such as the Raspberry Pi, relay circuit transmitter, magnetic contact switch, and the photocell.

The group aimed to address the issue of idle energy as well as safety in the home. By implementing this project that utilized IoT, users are able to control and monitor devices in their home. The ability to control and monitor devices remotely using IoT is a contribution to reducing idle energy since users are able to control idle devices away from home. Additionally, if intrusion was present in the home, the users are notified instantly in order to notify the authorities. Furthermore, the market research of the IoT technology revealed its potential global market contribution in the future. By 2020, over 30 billion devices are estimated to be utilizing this technology. This estimation displays the importance of IoT in the developing world since more devices are actively utilizing it annually.

10.3 Recommendations for Further Research and Revisions

The Internet of Things is very relevant in making human lives more simple and efficient. IoT technology enables devices to be connected to the internet which allows them to be controlled and monitored. For that purpose, users can monitor and control the devices locally or remotely. This provides users with more flexibility and control. Although the most typical usage for IoT technology is for home devices and appliances, IoT technology has the capability to expand outside the home. An example is IoT in healthcare. Medical devices can be connected to the internet which allows doctors to monitor and keep track of their patient's health and provide better healthcare for their patients. Because of the potential for growth and the benefits IoT can have on society, the group believes that further research in this technology will have a positive impact on the future.

The group's completed project is far from a viable market product since there are many improvements and revisions that can be implemented. One of these revisions is choosing an alternate way of controlling the devices remotely. Radio frequency signals are reliable, but there is a chance of losing the signal during transmission if obstacles are present between the radio frequency transmitter and receiver. Additionally, the UI of the website could be improved by making it mobile friendly and a phone application could be implemented to make website more user friendly. The prototype design of the project could also be improved by using printed circuit board (PCB) for the circuits of the project. The current prototype of the project has all the visible wires and connections, which may be vulnerable to damages from misuage overtime. If the prototype was finished in a PCB, the project would be safer, mobile, and portable. In short, IoT is a versatile technology that is a valuable contribution to the improvement of the quality of life.

References

D. Bandyopadhyay & S. Jaydip 'Internet Of Things: Applications And Challenges In Technology And Standardization'. Wireless Pers Commun 58.1 (2011): 49-69. http://dx.doi.org/10.1007/s11277-011-0288-5

J. Bohli, C. Sorge, & D. Westhoff. "Initial Observations on Economics, Pricing, and Penetration of the Internet of Things Market." ACM SIGCOMM Computer Communication Review, vol. 39, pp. 50-55, 2 April 2009. https://dx.doi.org/10.1145/1517480.1517491

J. Chambers. (2014). Are you ready for the Internet of everything? World Economic Forum. Retrieved from https://www.weforum.org/agenda/2014/01/are-you-ready-for-the-internet-of-everything/

D. Wang, D. Lo, J. Bhimani, & K. Sugiura. (2015). AnyControl -- IoT based home appliances monitoring and controlling. Paper presented at the Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual, , 3 487-492. doi:10.1109/COMPSAC.2015.259

Gartner. (2015). Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015 [Press release]. Retrieved from http://www.gartner.com/newsroom/id/3165317.

S. Gibbs (2015). Samsung launches SmartThings internet of things hub. The Guardian. Retrieved from http://www.theguardian.com/technology/2015/sep/03/samsung-launches-smartthings-internet-of-things-hub.

Buyya, R. & J. Gubbi. 'Internet Of Things (IoT): A Vision, Architectural Elements, And Future Directions'. Future Generation Computer Systems 29.7 (2013): 1645-1660. http://dx.doi.org/10.1016/j.future.2013.01.010

J. Greenough. (2015). The Internet of Everything: 2015. Business Insider. Retrieved from http://www.businessinsider.com/internet-of-everything-2015-bi-2014-12.

International Data Corporation. (2015). Explosive Internet of Things Spending to Reach $1.7 Trillion in 2020, According to IDC [Press release]. Retrieved from http://www.idc.com/getdoc.jsp?containerId=prUS25658015.

R. Khan. S. Khan, & R. Zaheer. 'Future Internet: The Internet Of Things Architecture, Possible Applications And Key Challenges'. 2012 10th International Conference on Frontiers of Information Technology (2012). http://dx.doi.org/10.1109/FIT.2012.53

I. Lee & L. Kyoochun. 'The Internet Of Things (IoT): Applications, Investments, And Challenges For Enterprises'. Business Horizons 58.4 (2015): 431-440. http://dx.doi.org/10.1016/j.bushor.2015.03.008

N. Huansheng, & W. Ziou. 'Future Internet Of Things Architecture: Like Mankind Neural System Or Social Organization Framework?'. IEEE Communications Letters 15.4 (2011): 461-463. doi: 10.1109/LC10.1109/LCOMM.2011.022411.110120OMM.2011.022411.110120

Appendix

Appendix A – Prototype and UI of the Project

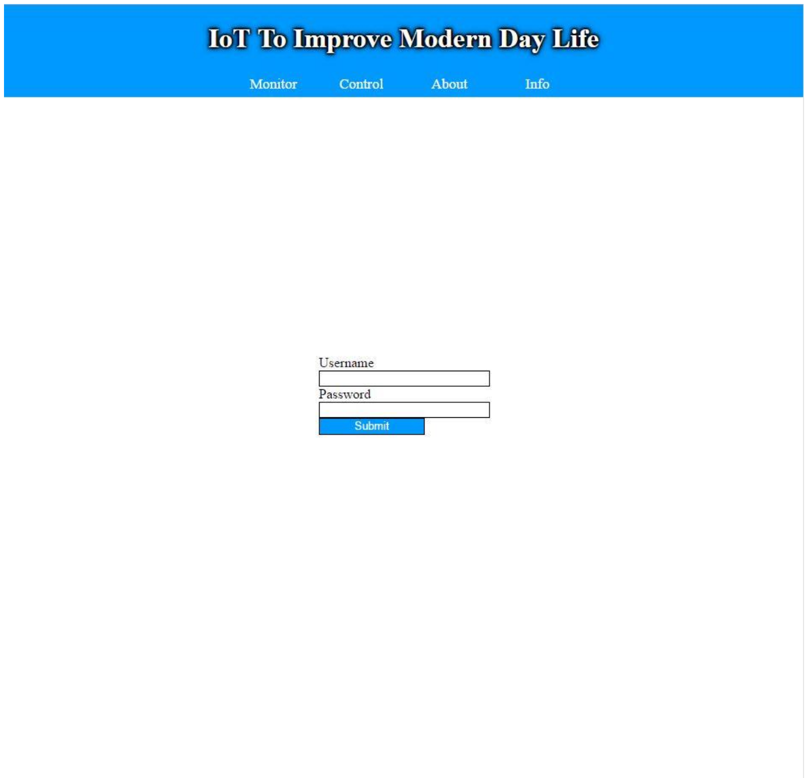

Figure 21. Login of the website

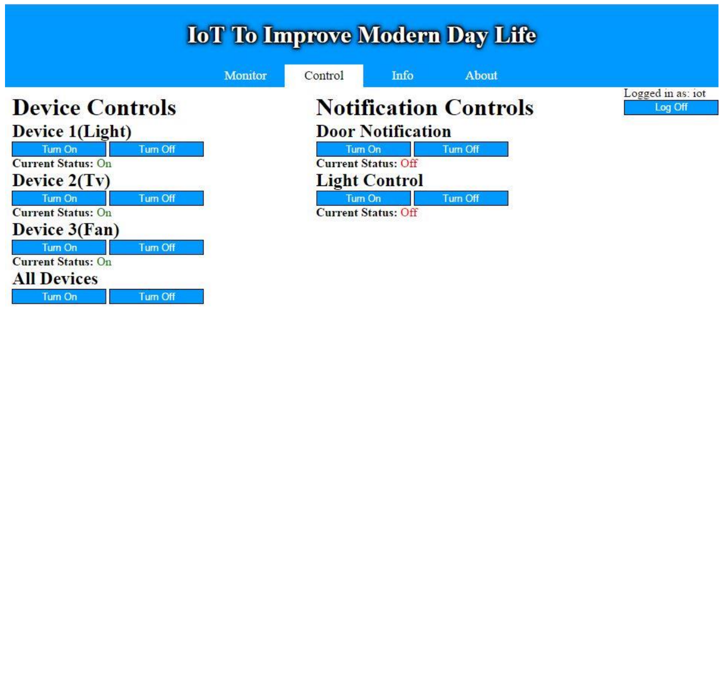

Figure 22. Control page for the website

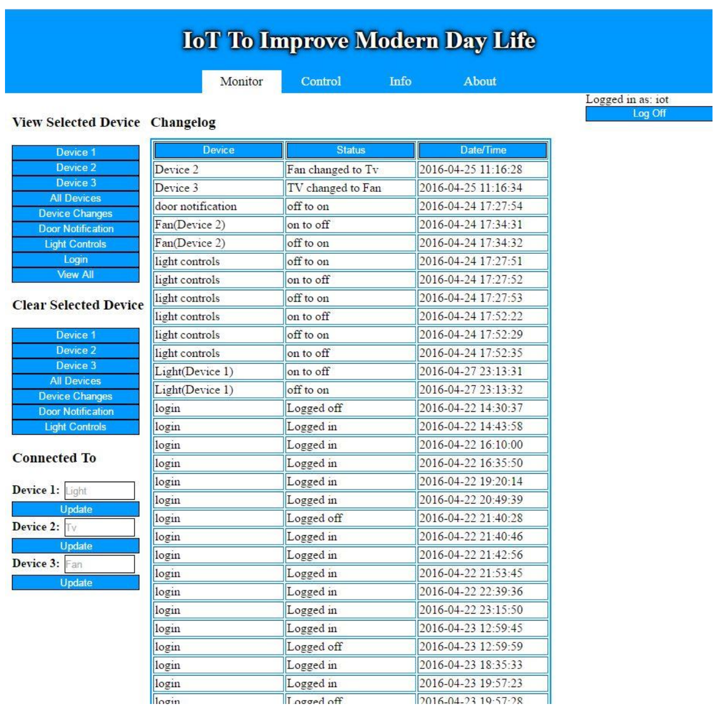

Figure 23. Monitor page for the website

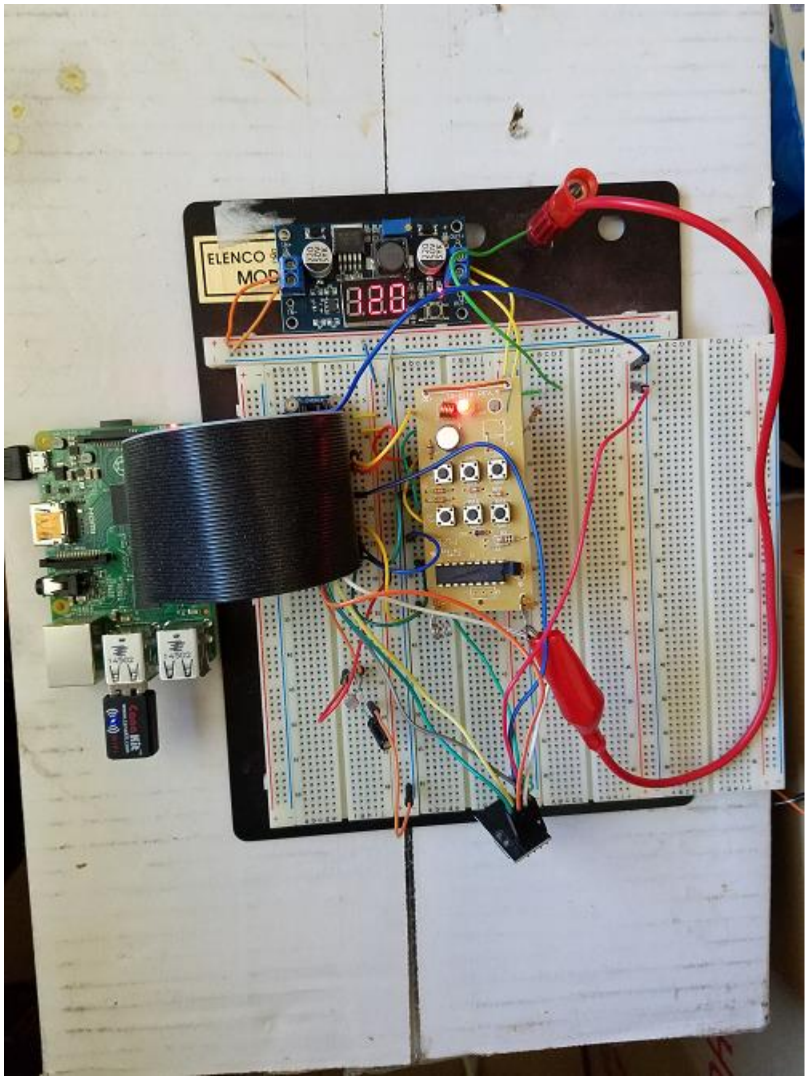

Figure 24. Project Prototype


Figure 25. Relay Receiver

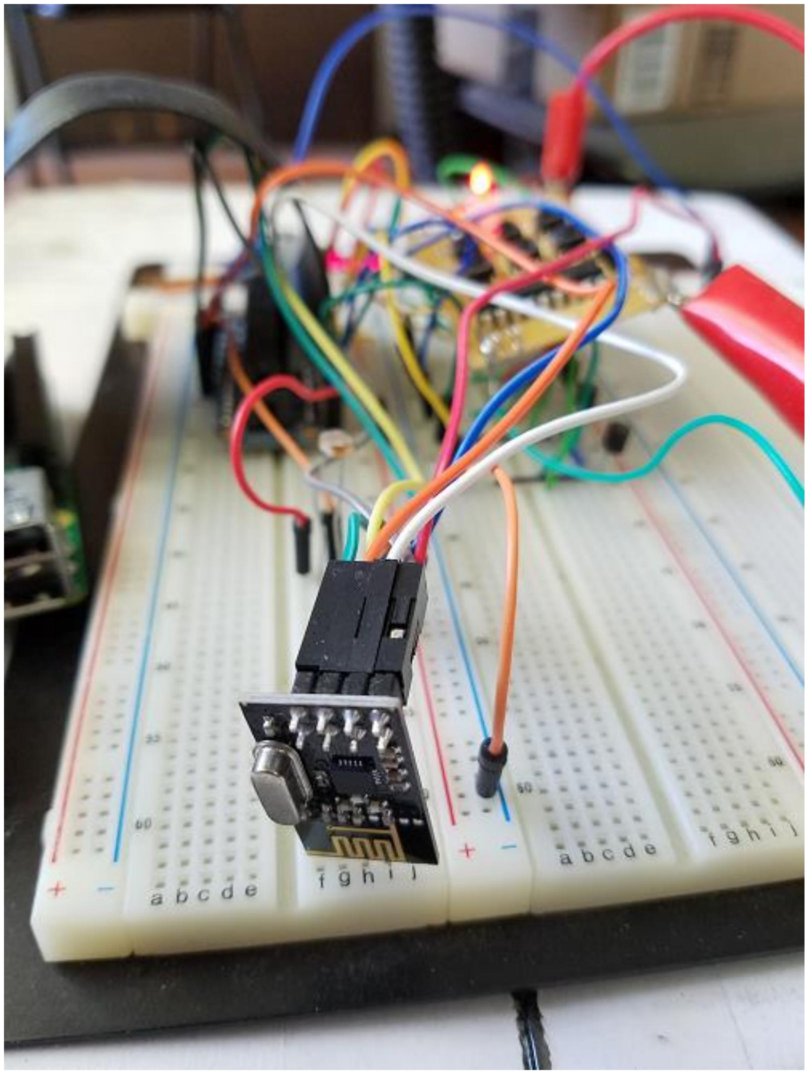

Figure 26. Door Switch Base Station

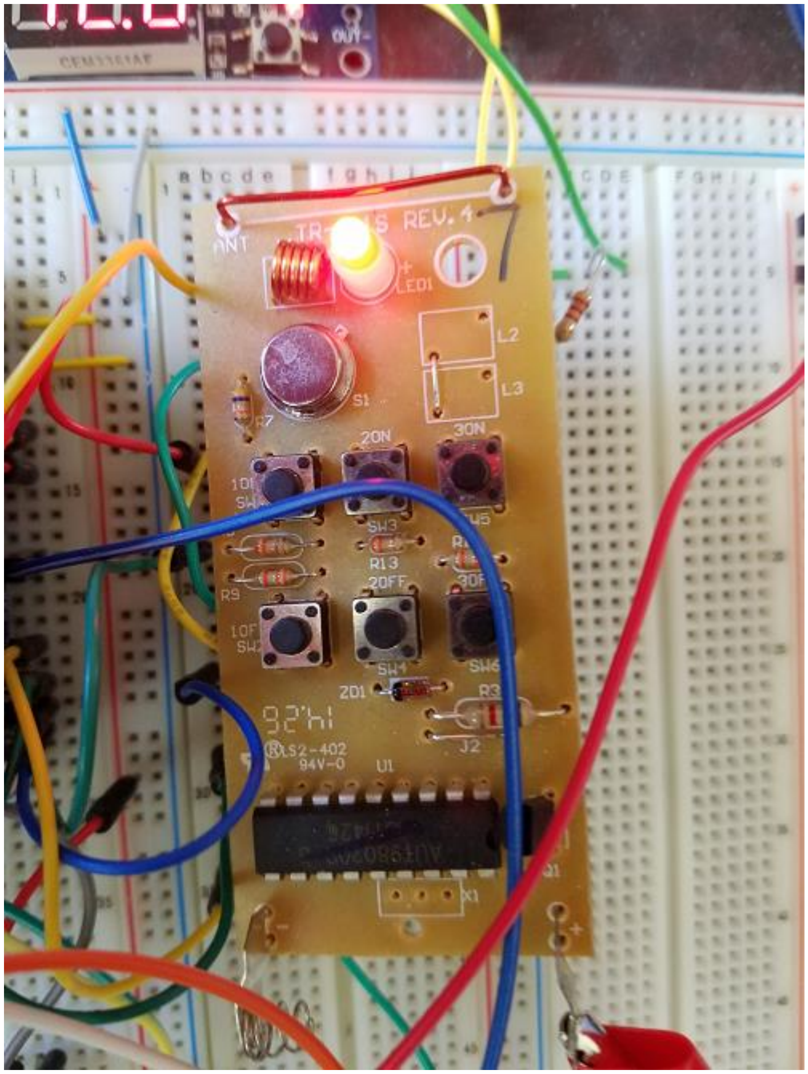

Figure 27. Relay Transmitter

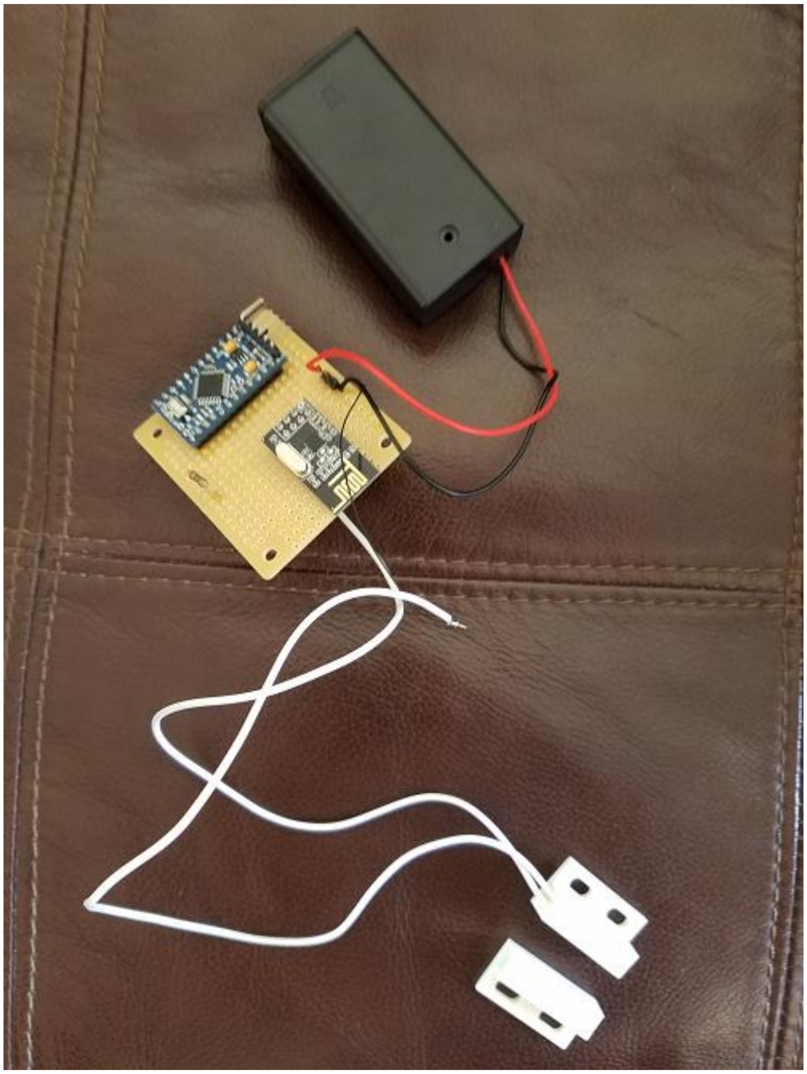

Figure 28. Door Switch

Appendix B - Contributions

Phi Lam

- 1.1 - Project Goals and Objectives
- 1.2 - Problem and Motivation
- 4.1 - Technology and Resource Requirements
- 4.2 - Environment and Interface Requirements
- 4.3 - Interface and Components Design
- 5.1 - Architectural Design
- 5.2 - Structure and Logic Design
- 7.0 - Testing and Verification
- 7.1 - Functional Testing

- 9.3 - Engineering
- 10.1 - Summary

Gary Lai

- 1.1 - Project Goals and Objectives
- 1.3 - Project Application and Impact
- 2.1 - Background and Used Technologies
- 2.2 - State-of-the-Art
- 4.1 - Technology and Resource Requirements
- 4.2 - Environment and Interface Requirements
- 5.1 - Architectural Design
- 5.2 - Structure and Logical Design
- 5.3 - Design Constraints, Problems, Solutions, and Trade-offs
- 6.3 - Tools Used
- 7.0 - Testing and Verification
- 7.1 - Functional Testing
- 8.2 - Issues and Potential Solutions
- 9.1 - Math

Thinh Phan

- 1.1 - Project Goals and Objectives
- 1.2 - Problem and Motivation
- 1.4 - Project Results and Deliverables
- 3.1 - Domain and Business Requirements
- 3.2 - System (or Component) Functional Requirements
- 3.3 - Non-functional Requirements
- 3.4 - Context and Interface Requirements
- 4.1 - Technology and Resource Requirements
- 5.1 - Architectural Design
- 5.2 - Structure and Logic Design
- 7.0 - Testing and Verification
- 7.1 - Functional Testing
- 7.2 - Non-Functional Testing
- 10.1 - Summary
- 10.3 - Recommendations for Further Research and Revisions

Allardyce Suba

- 1.1 - Project Goals and Objectives
- 1.4 - Project Results and Deliverables
- 1.5 - Market Research
- 2.1 - Background and Used Technologies

- 2.3 - Literature Survey
- 5.1 - Architectural Design
- 5.3 - Design Constraints, Problems, Solutions, and Trade-offs
- 6.1 - Project Team
- 6.2 - Project Tasks and Schedule
- 6.3 - Tools Used
- 7.0 - Testing and Verification
- 7.1 - Functional Testing
- 8.1 - Performance Characteristics
- 8.2 - Issues and Potential Solutions
- 9.2 - Science
- 9.3 - Engineering
- 10.1 - Summary
- 10.2 - Conclusions
- 10.3 - Recommendations for Further Research and Revisions