California State University, San Bernardino

School of Computer Science and Engineering

Internet Enabled Garage Door Opener

Submitted by

Luis Garcia

Ileana Lopez

Anwar Ibrhaim

Executive Summary

The purpose of this report is to understand how our internet enabled garage door opener was developed. We first started with understanding the customers needs which was creating a garage door opener that can be accessed from anywhere through a website that has some security. We had some constraints that we needed to overcome like delivering this design within a 9 month time period and many more which are listed in our table of contents under constraints. Once we understood the problem and the constraints, my team and I documented the principle of operation, and thoroughly understood how a microcontroller will send and receive signals. One of the important things in this report is the user manual which we created for users to set up by themselves. Furthermore, we came up with 3 system alternatives that cover the constraints and satisfy the customers needs which were: NodeMCU, Arduino, and Raspberry pi. There is more detail in our system alternatives and alternative selection from this document but,we came to a conclusion that the microcontroller we would use would be the NodeMCU. Not only is it cost-efficient but satisfies the customer needs. In addition, we did research on the microcontroller that was chosen and created a block diagram. We documented how exactly each block diagram was going to work with one another. Finally, once everything was ready, we started creating the user interface and database. Once those were finished we started to code the NodeMCU and tested it with an LED. Once we had all the code ready, the components we integrated together were: power source, web application, database server, NodeMCU, magnetic sensor, relay module, and garage door. When we noticed everything was working properly, we created a circuit design with the garage and software diagram which is per detail in our table of contents. In order to know if our design was fully functioning properly, we created a system test plan and results under system test plan and results. An example of a test plan would be to make sure that our index page or in other words our login page would not let a user in if the user was not in the database. All the system tests passed and our design was functioning as intended. If we were to put this on the market, our total cost for our design was $26.00 without labor. At the beginning of creating the project, we came up with a timeline that had tasks with due dates which is what we wanted to follow to accomplish some constraints that the project came with. As we looked at our timeline and our actual timeline, we were close to following the timeline as per document.

Some things we would like to work on for future projects is making our website more secure and fixing a delay within our status changes.

Problem Description

My team is manufacturing a garage door opener using an internet enabled node MCU for anyone who owns a home with a garage. There are multiple reasons we are implementing this design such as: forgetting the remote for the garage, forgetting to switch the remote to the car they are using for the day, or someone else might want to try to open the garage, but there aren't enough remotes. My team proposes to open the garage door from where you're at using your mobile phone. Anyone will be able to open the garage door as long as they have the username and password. My team will also need to ensure that the web interface is user friendly in order for the product/prototype to be useful anywhere.

Constraints

- This design must be completed and the prototype delivered in about 9 months.
- The marketing plan depends on the manufacturing being about $20 with the garage door and motor already installed.
- The design will be used only with homes with garage doors already installed.
- The web interface will be accessible using a phone and the user will be able to unlock the garage door with their username and password.
- The nodeMCU will be powered by an outlet using a 5v power adapter.
- The user interface should be simple, it should indicate whether the door is closed or open.
- Garage should not close if something is in the way. Ie. a person walking or an object blocking.
- Garage door opener should be internet enabled, meaning it is accessible from anywhere
- Databases should be used for storing username and password for security purposes.
- In order to use a webpage to control the output of the node MCU remotely, it must act as a web server.

Requirements Specification

The Deliverables

- One working prototype of the internet enabled garage door opener using nodeMCU with a web application.
- Requirements specification document.
- Video representation of acceptance testing that verifies the working garage door opener.
- User's Manual.
- The system specification document that includes a design concept, block diagrams, functional description of the blocks, a system description, and any analysis done during system design.
- Schematic diagrams and circuit descriptions.

Special Restrictions

All components in our design are internet based, meaning internet connection is required.
The user interface can only be changed with written permission of the owner.

Principle of Operation

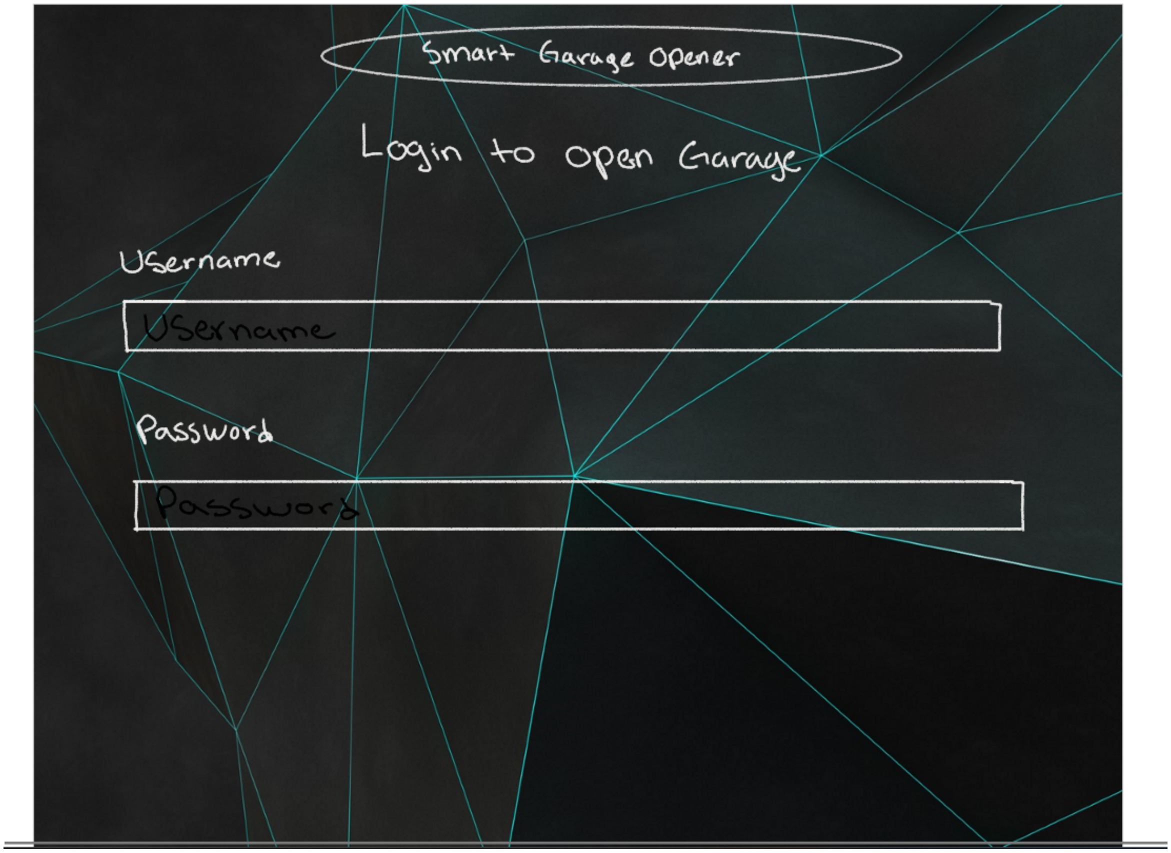
The nodeMCU will be connected to the garage door opener and controlled using a website that the guest will have to login to control. Usually, the garage is connected to some wires and controlled by a controller that is already programmed. The nodeMCU would need to be plugged into the wall as it was determined it would remain stationary and not move. There will also be a status indicator on the web application that lets the user know if the garage is open or closed.

User Interface

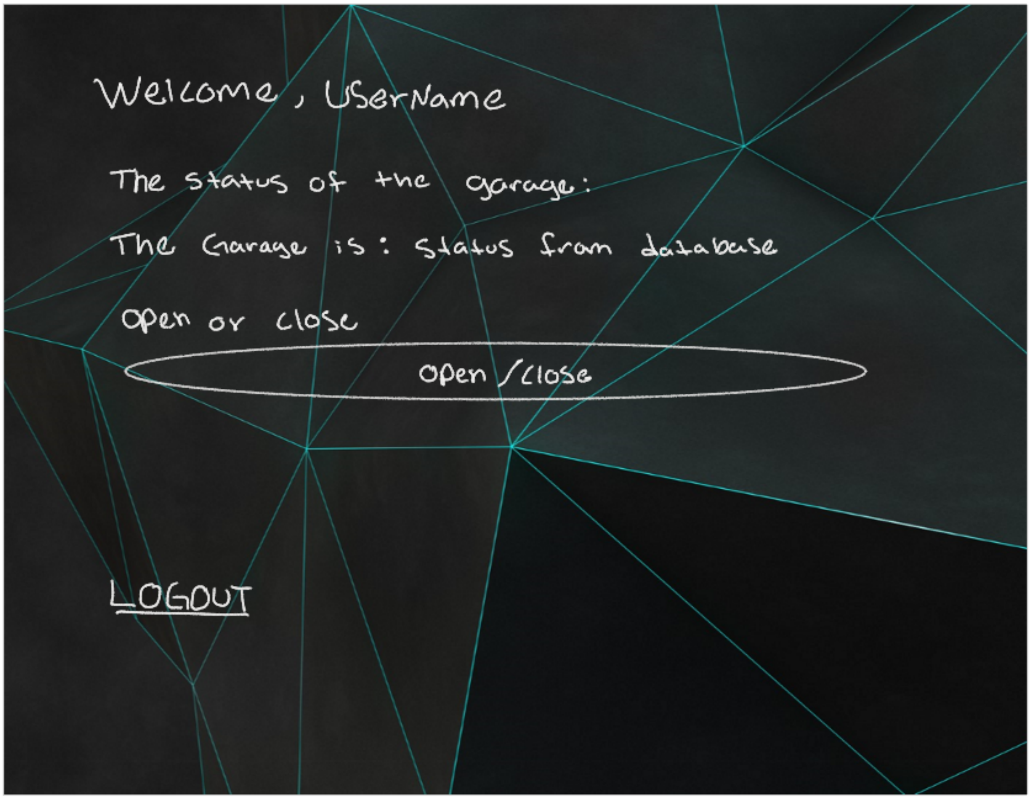
The user interface will be a website that prompts the user to enter username and password. Once they are logged in, they will have access to a button which will open and close their garage door. The user interface will display on the website whether their garage door is open or closed.

index.php

Friday, May 5, 2023 8:23 PM



home.php


Welcome, UserName
The status of the garage:
The Garage is: Status from database
open or close
open /close
LOGOUT

Input

The inputs in this design more or less will be all the signals passed from and to the nodeMCU. We will have a website that prompts the user for their username and password. Once the user logs in successfully, there will be a button to press which would be considered one of the inputs. If the user enters the wrong username or password, the website will not let the user enter the website until they enter the correct login. Once the user presses the button, the nodeMCU will receive a signal and based on that signal, the garage door will be triggered to open or close. In order for all this to function, the nodeMCU must be connected to a wifi network.

For security purposes, it would be best for the nodeMCU to communicate with the MySQL server indirectly using http or https since direct access could lead to security issues. The complexity of the codes for nodeMCU and the MySQL script would also be increased if we were to directly connect to the MySQL server. This would also lead to the nodeMCU demanding more resources, aka, memory and cpu usage. There is also the risk that NodeMCU will run out of memory if the server returns a large amount of data. Lastly, the MySQL libraries do not support SSL(Secure Sockets Layer) and TLS (Transport Layer Security) which encrypts the data and allows us to specify the security state of the connection. The username and password would be sent in plain text which is a big security gap. The indirect way of connecting to the MySQL server would solve these issues. This is an example of how the nodeMCU would communicate based on the input with a database and should work similarly with other solutions.

Output

- The outputs will be the garage door opening or closing as well as an indicator on the web application status. When it comes to connecting the hardware on the nodeMCU, we will need a relay board which will connect the garage to it. The relay board will connect to pin GPIO15 which is D8 on the nodeMCU board.
- If the user is logged in and does not input anything, the website should show the last status of the garage door.
- If the open/close button is pressed, the garage door will either open or close based on the position of the garage door.

- If the garage door is open, website will have indicator showing that it is open
- If the garage door is closed, the website will have an indicator showing that it is closed.
- If information entered incorrectly, the website will prompt the user “incorrect”.
- If anything is in the way of the garage door, the garage door will not close due to safety reasons and status of the website reflects if it hasn’t been closed.
- The way that it would work is that a http request would be sent to the web server, the web server runs the PHP script, then it would process the data and interact with the MySQL database and would lastly return the result to nodeMCU via a http response. This output process with MySQL should be how it works even if a different database software is used.

The User’s Manual

This internet enabled garage door opener that uses NodeMCU ESP8266 Wi-Fi and a web application that will allow you to open your garage door from anywhere.

The first thing to do is to make an account using 000webhost and create a web domain that will be used, the hosting service can be changed but for this version of the project, webhost is used as the hosting service for the web application

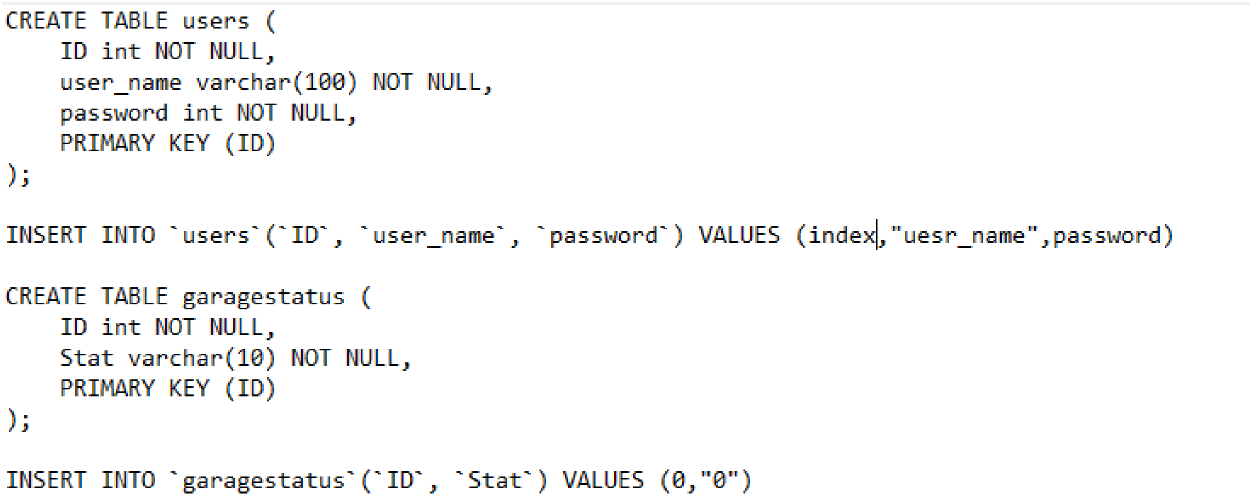
Download the necessary php files that will be used for the web application in terms of the UI and backend when communicating with the database and NodeMCU.

Make sure to download the arduino IDE from arduino’s website directly, https://www.arduino.cc/en/software

Download the arduino file that is necessary for the nodeMCU to connect directly to the garage

Create the database used with the name of garage_db which is under tools and database manager, then click manage and phpMyAdmin

Use the commands below once redirected to phpMyAdmin, for users, use the insert into command to insert any number of users you want in the format seen below



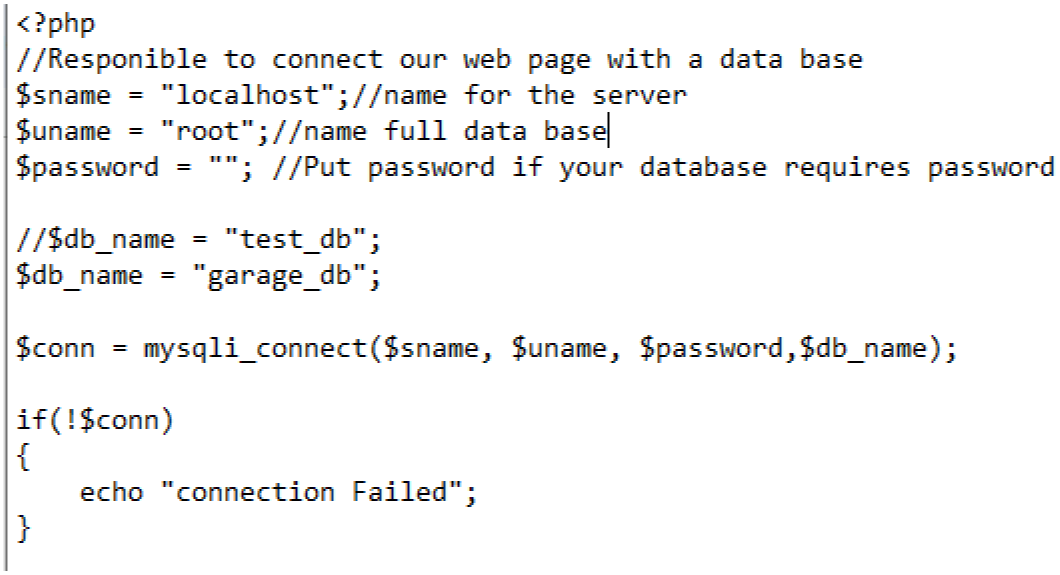
```
CREATE TABLE users (
    ID int NOT NULL,
    user_name varchar(100) NOT NULL,
    password int NOT NULL,
    PRIMARY KEY (ID)
);

INSERT INTO `users`(`ID`, `user_name`, `password`) VALUES (index,"uesr_name",password)

CREATE TABLE garagestatus (
    ID int NOT NULL,
    Stat varchar(10) NOT NULL,
    PRIMARY KEY (ID)
);

INSERT INTO `garagestatus`(`ID`, `Stat`) VALUES (0,"0")
```

Before putting the php files on the website, make sure to adjust parameters in the db_conn.php file to the names in your database manager



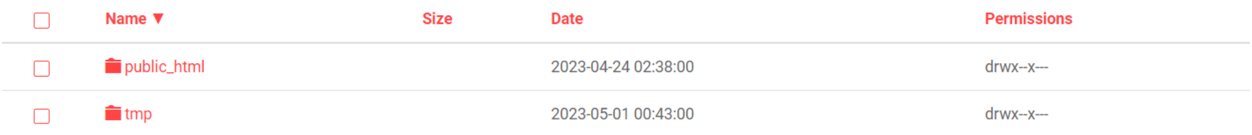
```
<?php
//Responible to connect our web page with a data base
$sname = "localhost";//name for the server
$uname = "root";//name full data base
$password = ""; //Put password if your database requires password

//$db_name = "test_db";
$db_name = "garage_db";

$conn = mysqli_connect($sname, $uname, $password,$db_name);

if(!$conn)
{
    echo "connection Failed";
}
```

Going back to the tools for the website, click file manager and upload all the php files under the public_html folder



□	Name ▼	Size	Date	Permissions
□	public_html		2023-04-24 02:38:00	drwx--x---
□	tmp		2023-05-01 00:43:00	drwx--x---

Now, it's time to set up the circuit for the nodeMCU.

We will set up the node mcu on a breadboard. We will connect the ground to the negative on the breadboard and the 3.3V from the NodeMCU to the positive on the breadboard. Now the bread board has power and ground so we can start adding our components. The first thing we will connect is the relay module. The relay module has 3 connections on one end which are in(input), VCC, and GND. We are going to connect in(input) on pin D8 on the NodeMCU, the VCC will be connected to power and the gnd will be connected to ground from the breadboard. On the other side of the relay module, we have 3 connections which are comm, NO(normally open), and NC(normally closed). We will be connected comm to 3 on the garage door and NC(normally closed) to number 4 on the garage door. We will not be using NO since you only need one configuration. Our next component is the hall effect sensor, you will need to place the hall effect sensor on the board with the pentagon shape facing you. The hall effect sensor has three legs which if you have it facing you, the one on the left is for voltage, the middle is for gnd, and the one on the right is for our input which is connected to pin D7. We will need a 2k ohm resistor from the positive terminal of the hall effect sensor to the input on the hall effect sensor. Lastly, this part is optional, if you would like to add an LED for testing purposes to ensure the button is working properly, connect the LED to pin D6 on the board with a 2k ohm resistor. The shorter side of the LED needs to be connected to ground. The other side will be connected to the 2k ohm resistor that leads to D6 on the NodeMCU.

Make sure to have the appropriate libraries installed in the arduino IDE
Before connecting the nodeMCU and relay module to the garage lifter, connect it to your computer and upload the code provided to it with your parameters which includes Network SSID, password, and the web host.
Connect the pins from the relay module to pins 3 and 4 on the garage lifter and place magnet on belt when garage is closed right next to where you position your hall effect sensor The COM pin needs to be connected to pin 3 on the garage lifter, and NC pin was connected to pin 4 on the garage lifter
Now you are able to use the website by logging in with the usernames and passwords you created in the database, and just click the button to open/close the garage with garage status being updated based upon whether it is open or closed.

Acceptance Tests

The performance testing of the garage door opener will be done through a video that will be shown in class that verifies that the web application will open the garage after user input.

Acceptance test for Password Accuracy:

- The Engineer will save 5 passwords that the user would like and save it on the database.
- The user will attempt to type passwords really similar to the ones they created but not exactly saved on the system. The system will prove whether or not the password accuracy is good or not.
- The user will attempt to type a username and password that are not saved to the system for security purposes.

Acceptance test for NodeMCU connected to garage:

- Users will be connected to the same wifi that the nodeMCU is connected to and see if the garage door responds to user input.
- Users will be connected to mobile data through their phone plan to ensure distance is not an issue as long as there is an internet connection.
- User will determine response time and compare that to what they had before with the standard button for garage doorAfter user input, garage should also open within the appropriate time and should not be slower for any reason
- The user will have an object in the way of the garage and see if that status is updated on the web application as the garage will not close.

Product Cost

The end-product cost is approximately $30.00, this includes: NodeMCU, power supply adapter cord for NodeMCU,jumping cables,breadboard, hall effect sensors and relay module.

Dispute Resolution Mechanism

All disputes will be settled by binding arbitration, with the arbitrator being Dr. Amir Ghasemkhani. The arbitrator will only be brought in if the team has failed to resolve a problem.

If my team is not able to finish the prototype, there is a risk that we will fail the course potentially.

System Alternatives and Alternative Selection

The way my team came up with the below design alternatives was by looking at our problem statement seeing what the design needs to do and what constraints it should have. We also factored in cost when coming up with the design, our main design had to do with what microcontroller would fit our constraints as well as cost. The three microcontrollers we came up with as a team were NodeMCU, Arduino, and Raspberry Pi.

Concept 1 Node MCU(wifi built in)

Specification:

Processor: Tensilica L106 32-bit

Processor speed: 80~160MHZ

Price : $7.00(Node MCU Development Board)

Hardware components:

Node mcu

Sparkfun single relay

Jumper wires

Bread Board

Strengths:

Built in wifi.

The price compared to other boards is adequate.

16 digital pins with PWM(Pulse Width Modulation)

NCMu comes with 128KB Ram and could store more which is more compared to uno.

Node MCU is compatible with MySQL which is a widely used relational database management system that uses structured query language.

Weakness:

Only one analog input pin

Concept 2 Arduino with wifi chip

Specification:

Processor: ATmega328P

Processor speed: 16MHZ

Price:$18.00(Arduino Development Board)

14 digital pins with 6 PWM(Pulse Width Modulation)

6 analog

Hardware components:

Arduino Uno

Sparkfun single relay

Jumper wires

Wifi chip(ESP8266)

Strengths:

It is compatible with MySQL which is a requirement as a database is needed for security and the board will be wifi enabled.

The price is reasonable and competitive.

There is a reasonable amount of documentation and resources that can be found online.

Weakness:

The wifi is not builtin and a separate wifi chip is required

Concept 3 Raspberry Pi 3 (wifi built in)

Specification:

Uses Broadcom BCM2711 SoC with 1.5 GHz processor

64-bit quad-core ARM Cortex-A72 processor

Price: ranges from $35 to $75 depend on the memory needed

Hardware Components:

Runs in Linux

Memory capacity from 2GB RAM to 8GB RAM

Raspberry Pi

5v Relay board

Bread board

Strengths:

Supports multiple sensors

Supports all types of codes

Faster Processor(1.6 GHz Processor in the 4B variant of Raspberry Pi)

Can be used as a portable computer

Raspberry pi is better at talking to the web.

Weaknesses:

Overheat if used for multiple hours

Will need an SD card inserted into the provided space

At this point in time, raspberry pi is more expensive than both arduino and node mcu as there is higher demand for this part in particular.

Concept 1 is the most promising concept because node MCU is the most cost effective and it is similar in the way that it works compared to the arduino. Working with the arduino is also a strong option and may still be considered in the future, but in terms of cost to performance, it is not as cost effective as the node MCU. Concept 3 also works, but raspberry pi is higher in demand meaning it is more expensive than arduino and the node MCU. In short, concept 1 is the most cost effective and makes the most sense for the project as we are trying to make a competitive product for the current market.

System Design

Block Diagram for the Chosen Concept

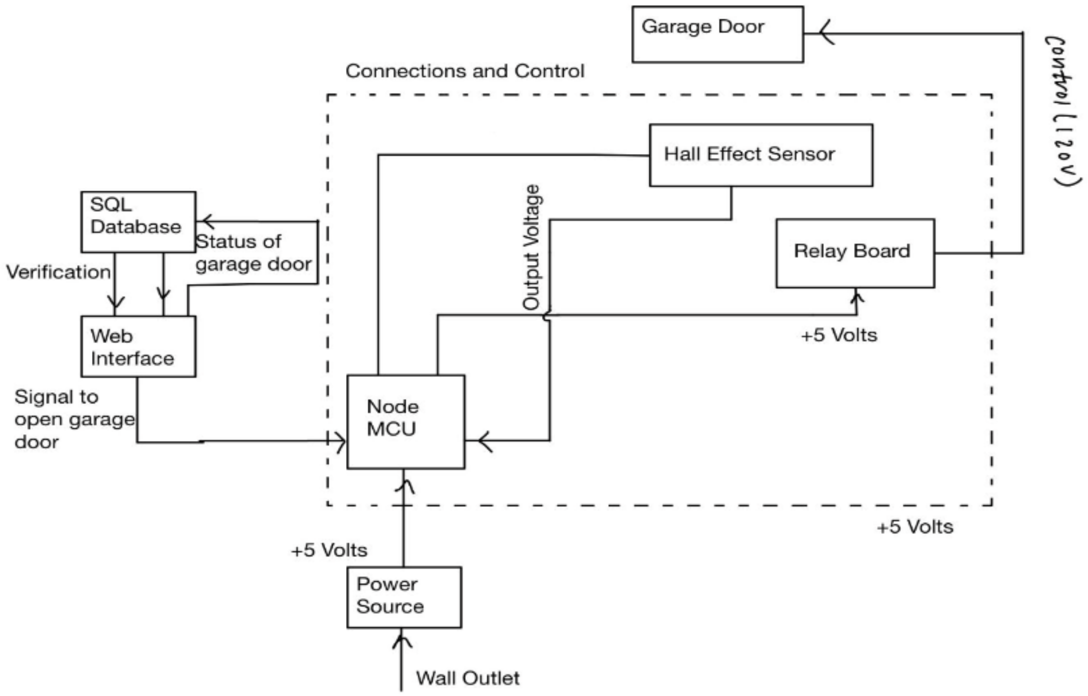


Figure 1. Block diagram for Garage Door opener using an internet enabled with Node MCU

Principle of Operation

The block diagram shown above is how an internet enabled garage door opener would work with Node MCU specifically which works with arduino IDE. This block diagram includes the following: a power source for the Node MCU, a web application and database for user input, hall effect sensors to sense whether the garage is open or closed, and a relay board is required to control the garage door itself since that is what will be connected directly to the garage opener. The intent for this design is to make the most cost effective internet enabled garage door opener with our current knowledge and since Node MCU is similar to arduino in terms of the programming, we thought it would be the best choice. It also has a wifi module built-in already so we do not have to worry about adding that into the cost. We believe the Node MCU is the best development board for this project due to the research my team has done.

When everything is all connected properly, the user logs in and presses the button on the web application to open or close the garage door which is connected to a database to verify the user’s login information is correct. After the verification process is complete, the user has access to control the garage door and the garage door status will either display open or close due to the hall effect sensor. The status of the garage door will be displayed on the web application as shown in the figures above. The garage door will open and close with the help of a relay board which will act as the switch for the garage door lifting up or closing down.

The SQL database will be used since there is an abundance of resources on how it works and is compatible with the Node MCU which works since it was the original plan if the arduino was used. It works similarly in which a user or users are created that will be made specifically for one garage door. The database will store any information required such as the status of the garage door and can be accessed via the web application.

Specification of the Blocks:

There are 7 blocks in the block diagram in figure 1 which is based around a Node MCU with a built-in wifi module. The boxes in the figure will indicate the action taken. The dotted line in the block diagram indicates all the components that use 5V. The regular lines indicate how the blocks are connected and how each of the blocks will interact with one another in terms of inputs and outputs. Ultimately, we want to be able to open a garage door via a web browser that anyone who knows the username and password can control through their phone. In the block diagram, we have all signal inputs that are sent to the node MCU and the signal outputs from the node MCU. The functions of the 7 blocks in Fig. 1 are described below.

Power Source:

The power source that will be used and supplied is 5V through a micro usb cable that will be connected to a wall outlet as portability is not a concern. In terms of making sure the temperature of node MCU is not too hot, we thought it would be best that the power supplied to the node MCU should not exceed 5V. For the node MCU, 5V should be enough power as it allows us to use up to 500mA, which should be enough for the sensors that we need to add as well as having to use wifi. My team has learned that the node MCU draws 200mA power for boot and Wifi operation, it draws up to 200 mA peak current. There are two other ways to supply power to the node MCU in which one should not be necessary and the other does not meet our requirements. The first way would be to use an external power supply using the VIN pin which allows us to power the node MCU anywhere from 5V to 12V ,i.e a 9V battery. The second way would be via the 3.3V pin in which a steady 3.3V is supplied to the node MCU. The second way does not meet our requirements as it has to be used with caution since if the voltage gets higher than 3.6V, this could cause the board to be damaged or overheat.

Web Application:

We believe that PHP would be the best option to develop the web application as it works hand in hand with mySQL which is used for databases. PHP is also open source and free, meaning there is an abundance of documentation available online when it comes down to the implementation. PHP is a general purpose server side scripting language that can be used for websites and applications. PHP will be used to create the web application as it is an acronym for hypertext processing and is suited for web development and can be embedded into html. It is also appropriate for dynamic web pages and database applications which is exactly what we need. My team learned a little bit about networking and security which helps understand the process of this. PHP will be installed on the web server that will host the application and client applications can access the server. It is important to also gain an understanding of how HTML works as that is mainly what goes into designing and making a website. PHP allows us to expand upon that knowledge to connect to a database server to retrieve and store information. The web application will consist of a header stating what the website is intended for such as garage door opener, it will also have a username and password for the user to promptly enter in which communication with the database is necessary. After the verification and successful sign in attempt of the user, the website will redirect you to another page with a simple button to click to open or close the garage door with the status of the garage door being opened or closed will also be shown. Lastly, we will use cascading style sheets to give our website that nice look.

Database server:

Structured Query Language is the most common language for extracting and organizing data that is stored in a relational database. For security purposes, it would be best for the node MCU to communicate with the MySQL server indirectly using http or https since direct access could lead to security issues. The complexity of the codes for arduino and the MySQL script would also be increased if we were to directly connect to the MySQL server. This would also lead to the node MCU demanding more resources, aka, memory and cpu usage. There is also the risk that node MCU will run out of memory if the server returns a large amount of data. Lastly, the MySQL libraries do not support SSL, TLS which encrypts that data and allows us to specify the security state of the connection. The username and password would be sent in plain text which is a big security gap. The indirect way of connecting to the MySQL server would solve these issues. In addition, we will be using a php script to insert data into the MySQL database. This will be used when sending username and password to the database to check whether or not it's the same. This is our vision of how the node MCU would communicate based on the input with a database and should work similarly with other solutions. My team has learned that a database can use tables to store information as well as retrieving information from those tables when necessary. In our case, this will be things like the username and password as well as the status of the garage door which will be retrieved from the magnetic sensors. It is important to get that information from the node MCU and store that in the database to update the web application with.

Node MCU:

Node MCU is a low cost development board that we will be using for this project. My team is well oriented with arduino, but don't have much experience with node MCU, but we believe we should be able to accomplish the given task with this board. Node MCU works with the arduino IDE as well which we are familiar with and it is more of a matter of making sure we understand the difference between node MCU and arduino. According to our block diagram, the Node MCU will have one output and three inputs. One input will be from the power source which we agree that it will be 5V through a micro usb cable that will be connected to a wall outlet. Most homes that have garages have that motor near a wall outlet usually in the ceiling which should be accessible to connect the NodeMCU. Another input that is going into the Node MCU is the username and password which is entered in the website, and retrieved through the database. The output will be the signal sent to the relay board to trigger the garage to either open or close.

Magnetic Sensors:

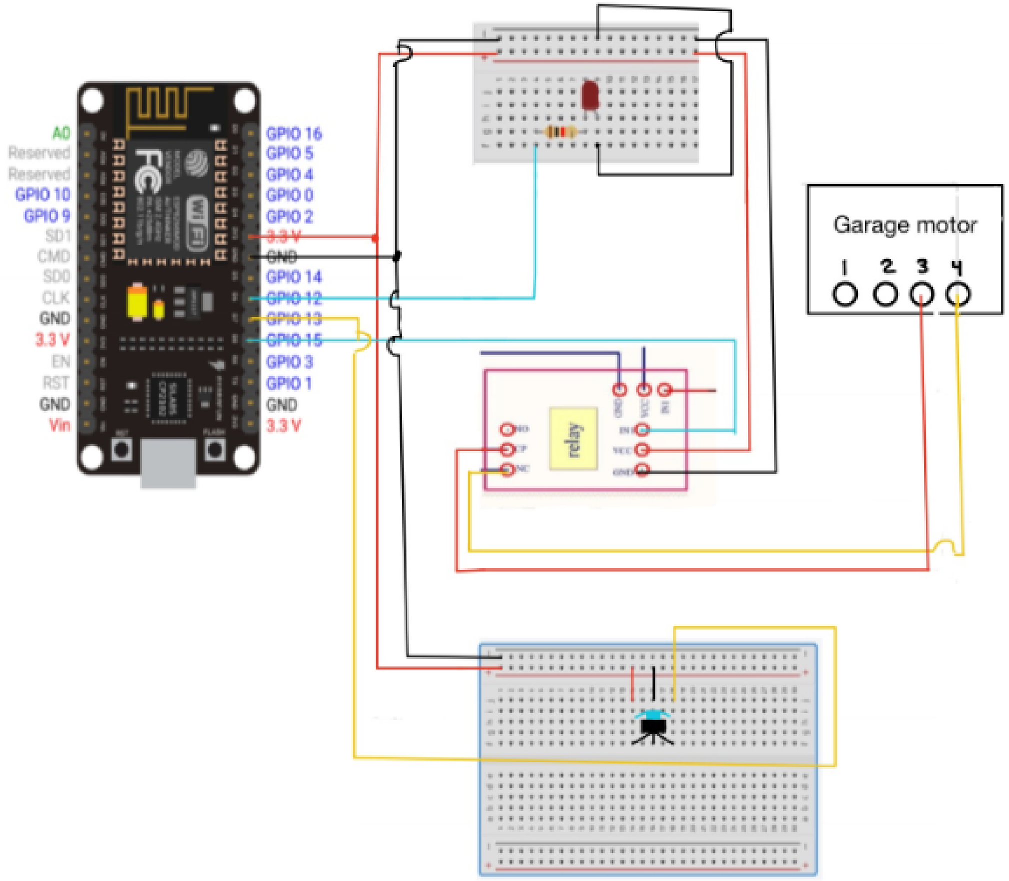
Based on research done, it seems that it would be best to use a hall effect sensor to determine whether the garage door will be open or closed. The hall effect is essentially the production of a voltage difference across an electrical conductor, transverse to an electric current in the conductor and to an applied magnetic field perpendicular to the current. These sensors work by measuring the changing voltage when the device is placed in a magnetic field. We will need to install a magnetic sensor onto the belt of the garage. The hall effect sensor will be placed near the magnetic sensor letting us know the garage is closed, when the garage opens, the magnet will move letting us know the garage is open.

Relay Board And Garage Door:

The output connection on the Node MCU is the connection with the relay board and the signals sent to the garage. A relay board is an array of relays and switches that have the output of terminals that are designed to control the voltage supply. In other words, this relay module is operated by a relatively small electric current and the relays turn on or off in our case, close or open a much larger electric current. We are visually going to see an output once the garage opens and closes. A relay board also tends to be safe as there would not be physical contact between the node MCU and the garage door or any device in general. A relay module also has two configurations which are NO (Normally Open) and NC (Normally Closed). There is also a COM pin which stands for common pin which is used in both configurations. The NC configuration is when you want the current relay to be closed by default, meaning the current is flowing unless a signal is sent from the Node MCU to open the circuit and stop the current. The NO configuration works the other way around in which the relay is always open, meaning the circuit is open unless a signal is sent from the Node MCU to close the circuit. It has a coil that is energized by 5V which is supplied through our node MCU which will also be powered by 5V and when the coil energized switching takes place, The relay module is what allows us to essentially control the garage door opener to lift the garage to either open or close it as the node MCU cannot do it standalone. The relay module is essential to a web enabled garage door opener since the garage door opener is already installed and would have a much larger electrical current and can be controlled with the much smaller electrical current safely.

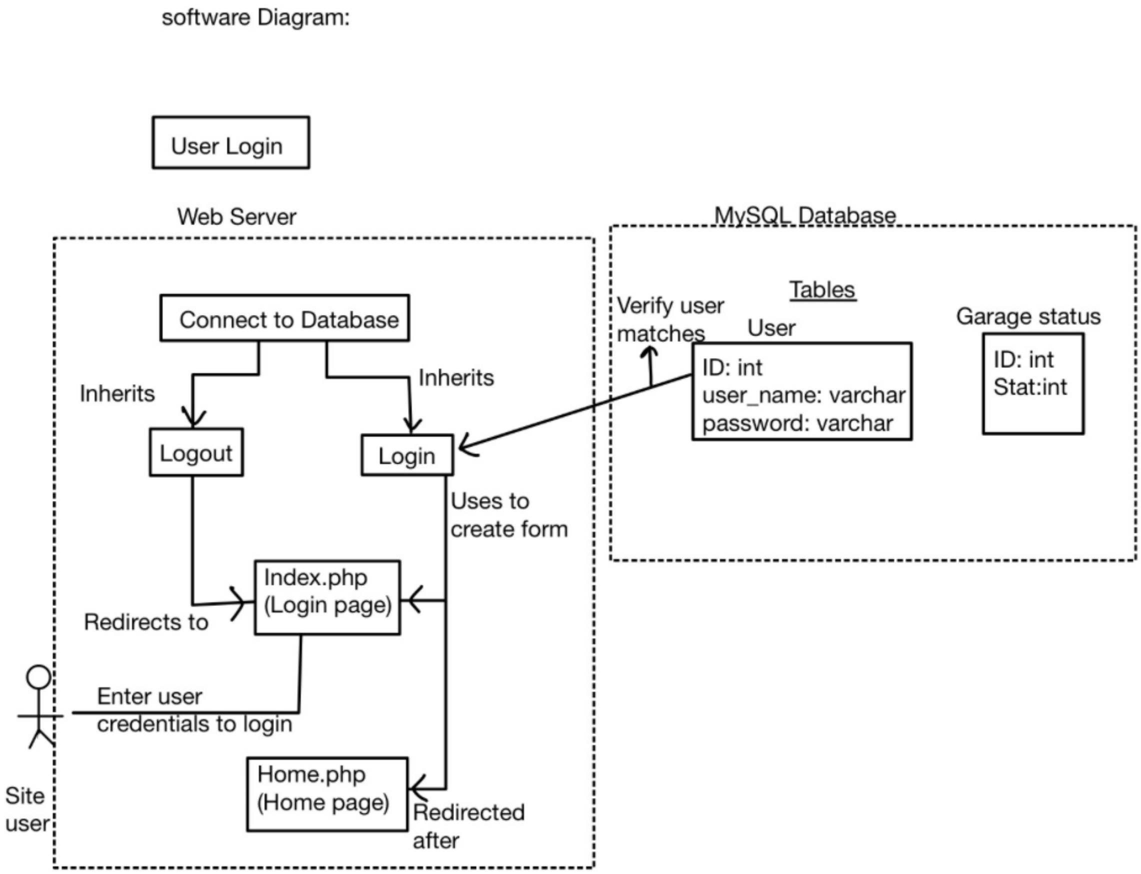
Detailed Design

Circuit Design with Garage



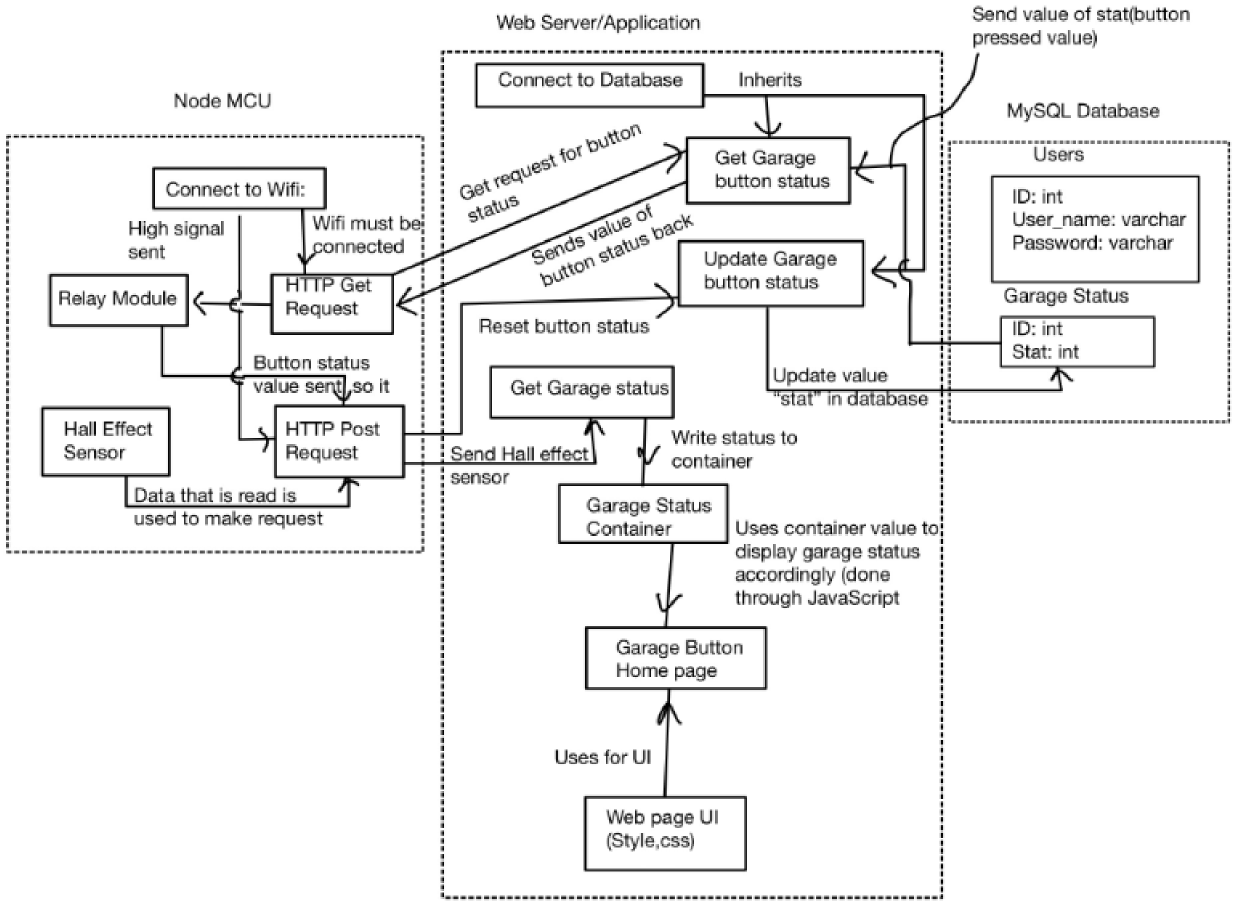
Software Diagrams

User Login



The first thing the web server does is connect to the database and both the login and logout inherits the file that does so. The site user enters their credentials through the login page, index.php, which has a form that uses login.php. Once the user has entered their credentials, the login file will handle parsing the user credentials and run a SQL query to compare to all the values “user_name” and “password” stored in the database. Login will then redirect the user to the homepage for the garage door. Logout button on the home page will redirect the user back to the login page.

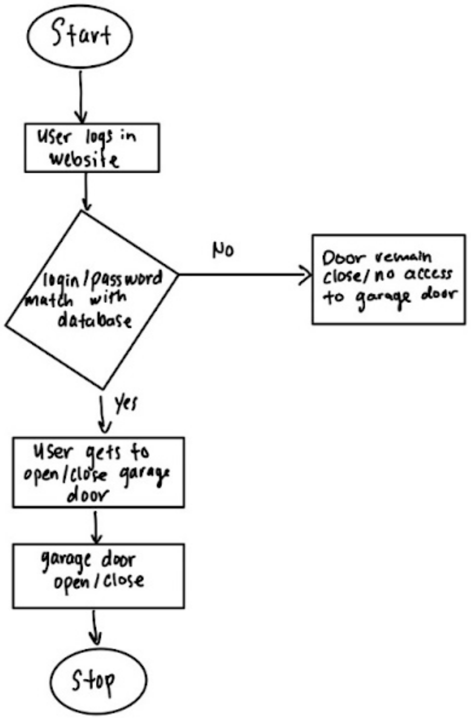
Interaction with Garage Door



The nodeMCU connects to wifi and sends http get and post requests to the web server. The web server connects to the database and button status is set to 1 when the button is pressed which the nodeMCU retrieves via httpGet Request to trigger the relay module which the get garage button status handles. Button status is reset after the garage opens/closes via httpPost Request, and the database value "Stat" is updated. The web server retrieves hall effect sensor data via http Post Request and stores it in the garage status container to determine what status of the garage is shown in which the file involving home page handles through javascript.

Flow chart Design:

Garage Door Opener:


Start
User logs in website
login/password match with database
No
Door remain close/no access to garage door
Yes
User gets to open/close garage door
garage door open/close
Stop

System Test Plan and Results

System Test Plan

Equipment List	Procedure	Description(Results)
For Website Only: Electronic device connected to wifi or mobile network Link to website after created	Use web address created with web hosting service and see if login page is loaded Login to website with at least 3 different users saved in the database Login to website with invalid user_name or password, need	The login page should load successfully when web address is typed in and entered Upon successful login, user should be redirected to page where the garage can be open and status of garage is shown Upon invalid user_name or

	both inputted in textbox Login to website with no user_name entered Login to website with no password entered	password input, message should pop up, "Incorrect User name or password" Upon no user_name entered, message should pop up, "User Name is required" Upon no password entered, message should pop up, "Password is required"
For Garage Door interaction: NodeMCU connected to wifi network that will be used for garage opener Garage lifter Hall effect sensor Wires Relay Module 2 Resistors (2k Ohms) Magnet Electronic device connected to separate network Link to website that will be used	Login with existing user saved in database Once redirected to welcome page, press button to open to garage door After garage door is open, press the button again to see if the garage closes Check that garage status is correct after opening or closing garage via button press	Upon user input and login is successful, should be redirected home page Garage door should respond accordingly meaning if it's closed, it should open after button is pressed, if it is closed, it should open after button press After the garage is opened, garage status should be open. If the garage is closed, the garage status should be closed

Results

Procedure	Results
Use web address created with web hosting service and see if login page is loaded Login to website with at least 3 different users saved in the database	The login page does load correctly when address is typed and entered Website logs in successfully with at least 3 users and the user is redirected to the home page

Login to website with invalid user_name or password, need both inputted in textbox Login to website with no user_name entered Login to website with no password entered	User is not redirected to home page, message pops up that says “Incorrect User name or Password” User is not redirected to home page, message pops up that says “User name is required” User is not redirect to the home page, message pops up that say “Password is required”
Login with existing user saved in database Once redirected to welcome page, press button to open to garage door After garage door is open, press the button again to see if the garage closes Check that garage status is correct after opening or closing garage via button press	When the user is saved in the database, our website lets the user in and redirects them to the home page. Once in the home page, when the button is pressed, the garage does open or close regarding what state it is in. When the garage was open, the status was corrected as well as when it was closed. We did have a small delay within our software code, but the status did change after 3-5 seconds.

Economic Analysis

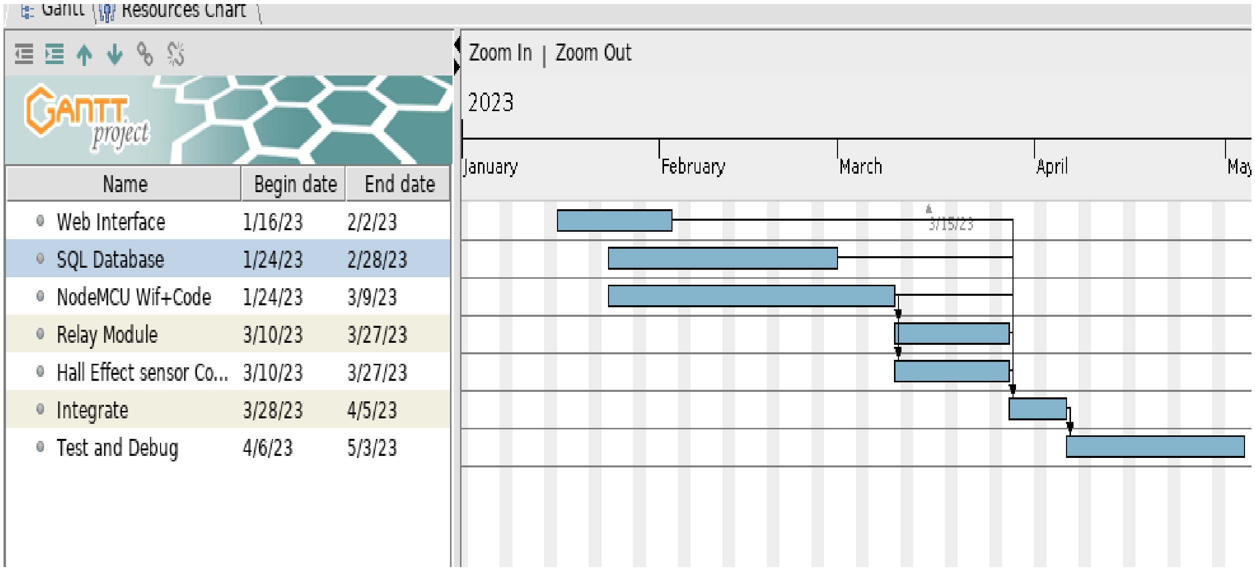
Materials	Description	Cost
NodeMCU	Open source platform that connects objects and let data transfer using WIFI	$8.00
Hall Effect Sensor	Sensor which detects the presence and magnitude of a magnetic field using hall effect	$6.00
Relay Module	An electrical switch that is operated by an electromagnet	$4.00

Jumper Wires	Electric wire that connects remote electric circuits used for printed circuit boards	$3.00
Breadboard	Plastic block holding a matrix of electrical sockets of a size suitable for gripping thin connecting wire, component wires or the pins of transistors and integrated circuits (ICs)	$5.00
Charger	Power brick with micro usb cable to power the NodeMCU, relay module, and hall effect sensor	$7.00
1 Magnet	Magnet will be placed on garage belt for Hall Effect Sensor to detect	$.50
Total Cost		$33.50

To produce this design at scale, other factors would have to be considered such as labor costs which are not here as we are not being paid to create this. In terms of the pure cost of materials when produced at scale, for roughly 100,000 units for example, the cost would be about 3.35 million dollars. If we want to make a profit, a reasonable number would be about 40 dollars for the actual price which would net 4,000,000 dollars in revenue. Some of the advantages for this product in terms of economic feasibility is that since the consumer would be the ones themselves setting up and connecting the circuit to the garage lifter, it would cut down on labor costs. In terms of this product actually being competitive in the market and being able to make a profit, it is possible since there are products that are above this price range and below it as well. The initial estimated budget for this design was around 26 dollars as we were considering using we did not include a charging cable, but overall our cost for everything else remained relatively the same. Ultimately, in terms of economic feasibility, it can be improved with a more cost effective design as proven by similar products in the market.

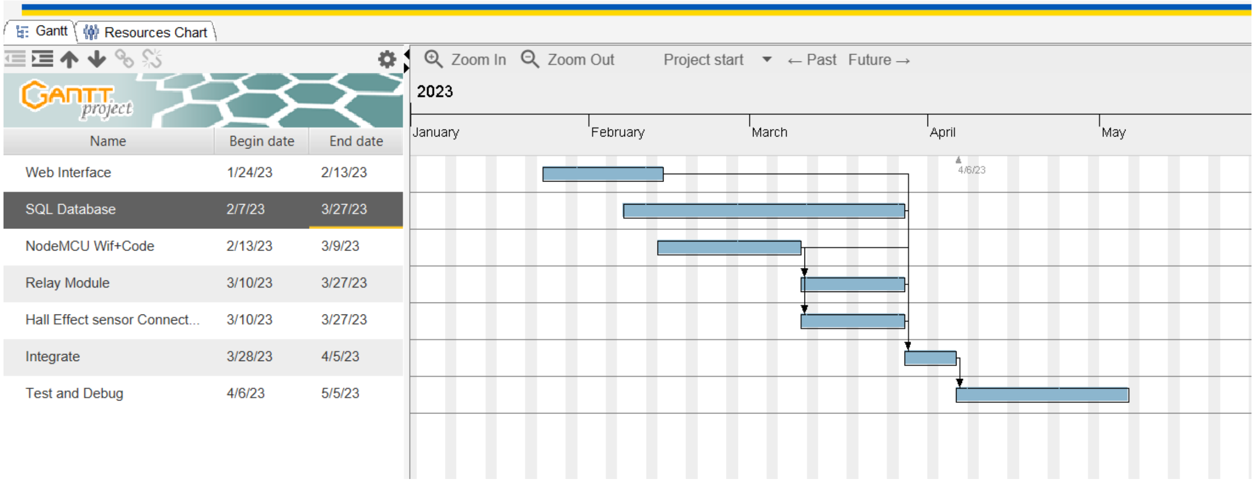
Project Management

Expected Dates:



Name	Begin date	End date
Web Interface	1/16/23	2/2/23
SQL Database	1/24/23	2/28/23
NodeMCU Wif+Code	1/24/23	3/9/23
Relay Module	3/10/23	3/27/23
Hall Effect sensor Co...	3/10/23	3/27/23
Integrate	3/28/23	4/5/23
Test and Debug	4/6/23	5/3/23

Actual Dates:



Name	Begin date	End date
Web Interface	1/24/23	2/13/23
SQL Database	2/7/23	3/27/23
NodeMCU Wif+Code	2/13/23	3/9/23
Relay Module	3/10/23	3/27/23
Hall Effect sensor Connect...	3/10/23	3/27/23
Integrate	3/28/23	4/5/23
Test and Debug	4/6/23	5/5/23

Expected and Actual Timelines

Website

The website was planned to start on January 16, 2023 and finish on February 2nd of 2023. We had anticipated starting a week before the semester began, but some things came up so we started on the first day of the semester. We started on January 24th, and it took us about three weeks to complete which was until February 13.

SQL Database

The SQL database was expected to be done in 3 weeks from January 24 ,2023 to February 28th, 2023. It was actually started on February 7, 2023 to April 2, 2023 to get it to work properly.

Node MCU

Node MCU Wifi + Code was expected to begin on January 24, 2023 and finish on March 9, 2023. We started working on the NodeMCU February 12, 2023 and got it finished March 27, 2023.

Relay Module

The relay module was expected to begin on March 10th, 2023 and be finished on March 27th, 2023. It was actually started on February 12, 2023 and was finished by March 27, 2023.

Hall Effect Sensor

The hall effect sensor code and circuit setup was expected to begin on March 10, 2023 and finished by March 27, 2023. The actual timeline it took was the same as the expected timeline.

Integrate

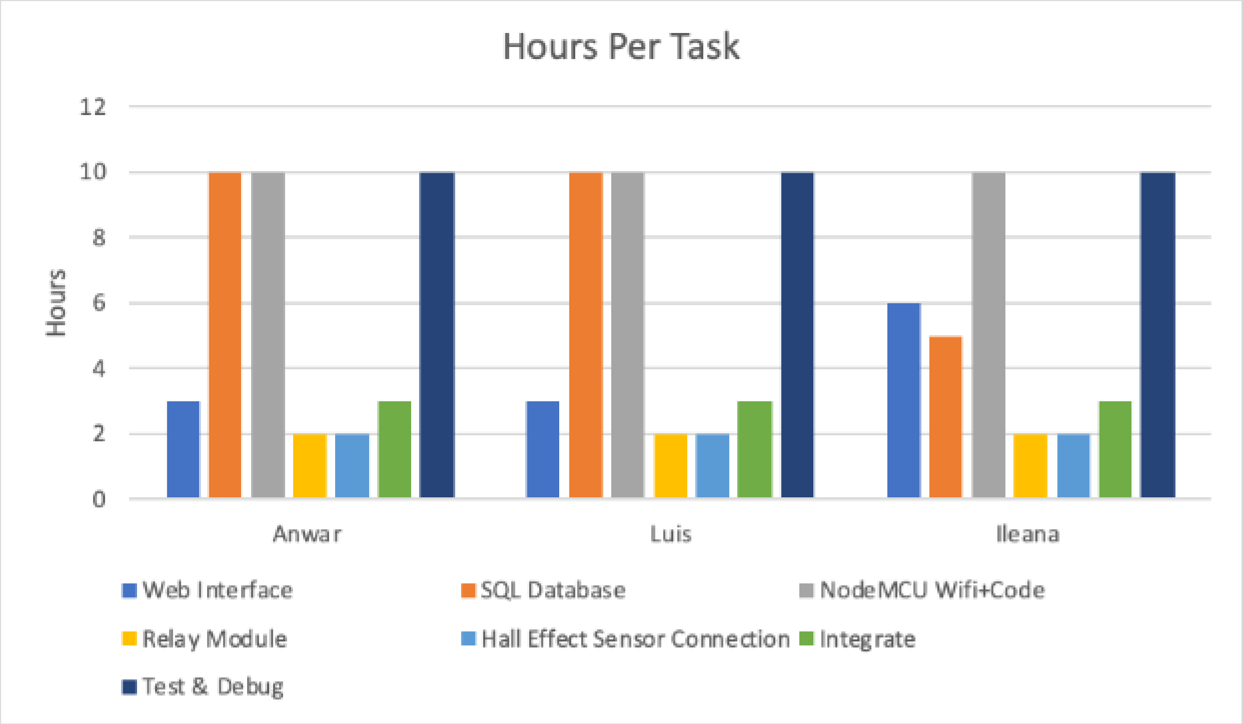
The Integrate part was expected to begin on March 28, 2023 and finished by April 5, 2023. The actual timeline was the same as the expected timeline for the integration.

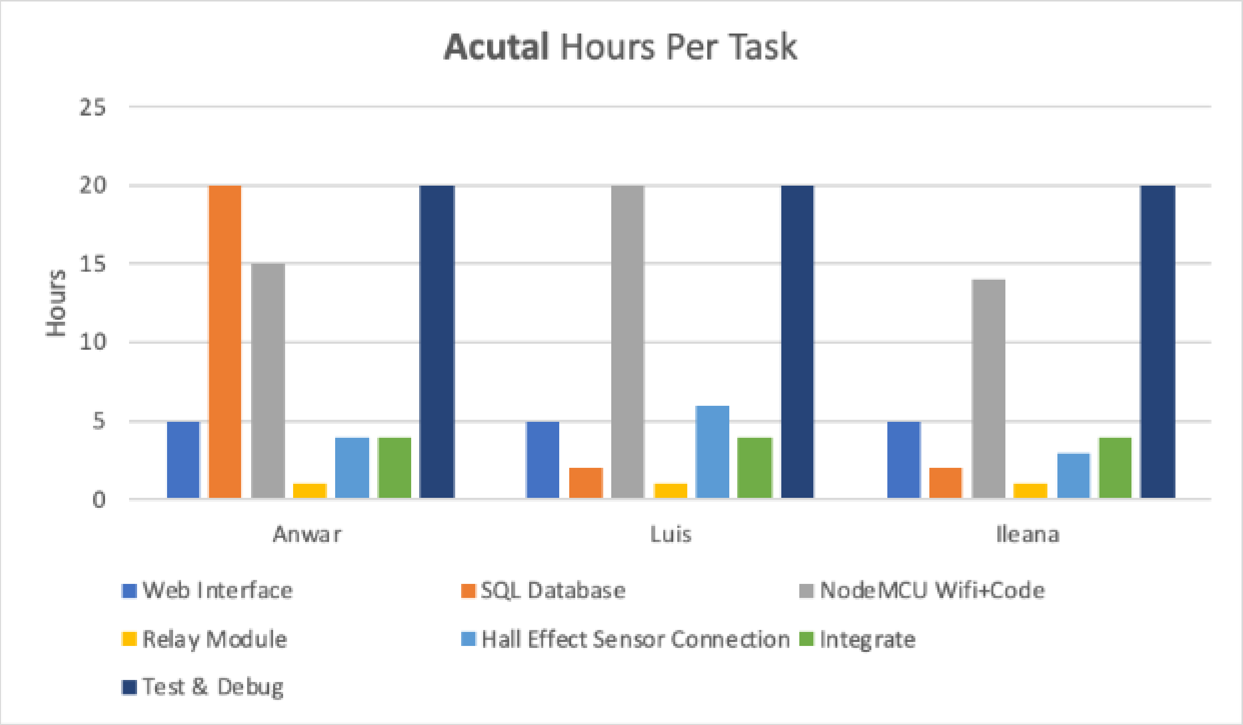
Test and Debug

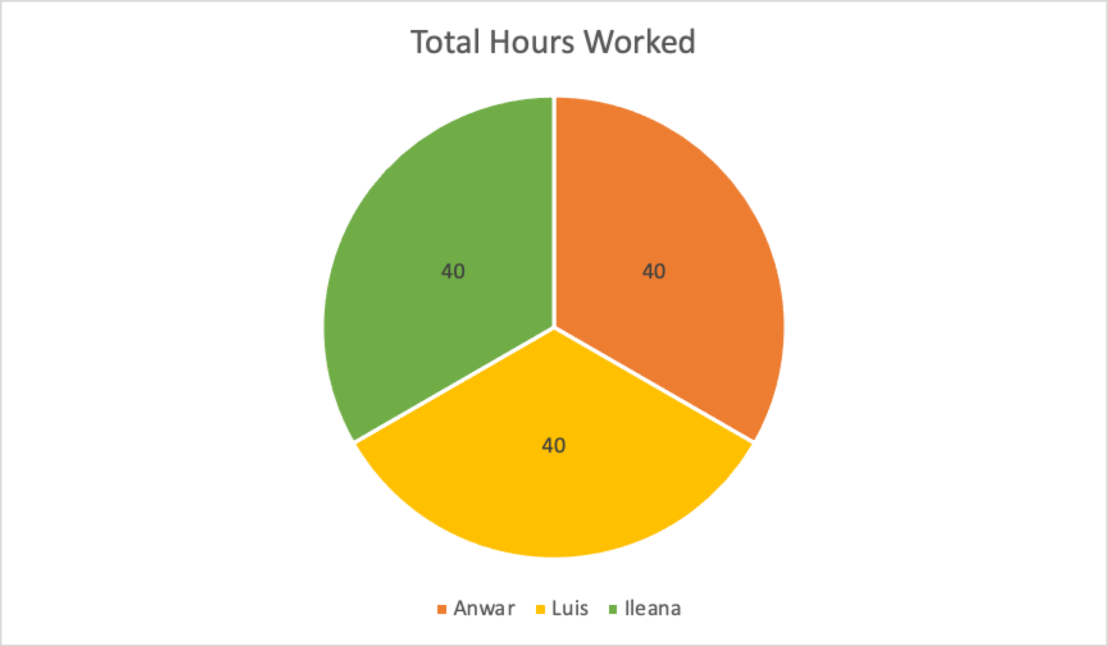
The test and debug of the project was expected to begin on April 6, 2023 and finished by May 3, 2023. The actual timeline of when we started was March 29th and finished on May 7.

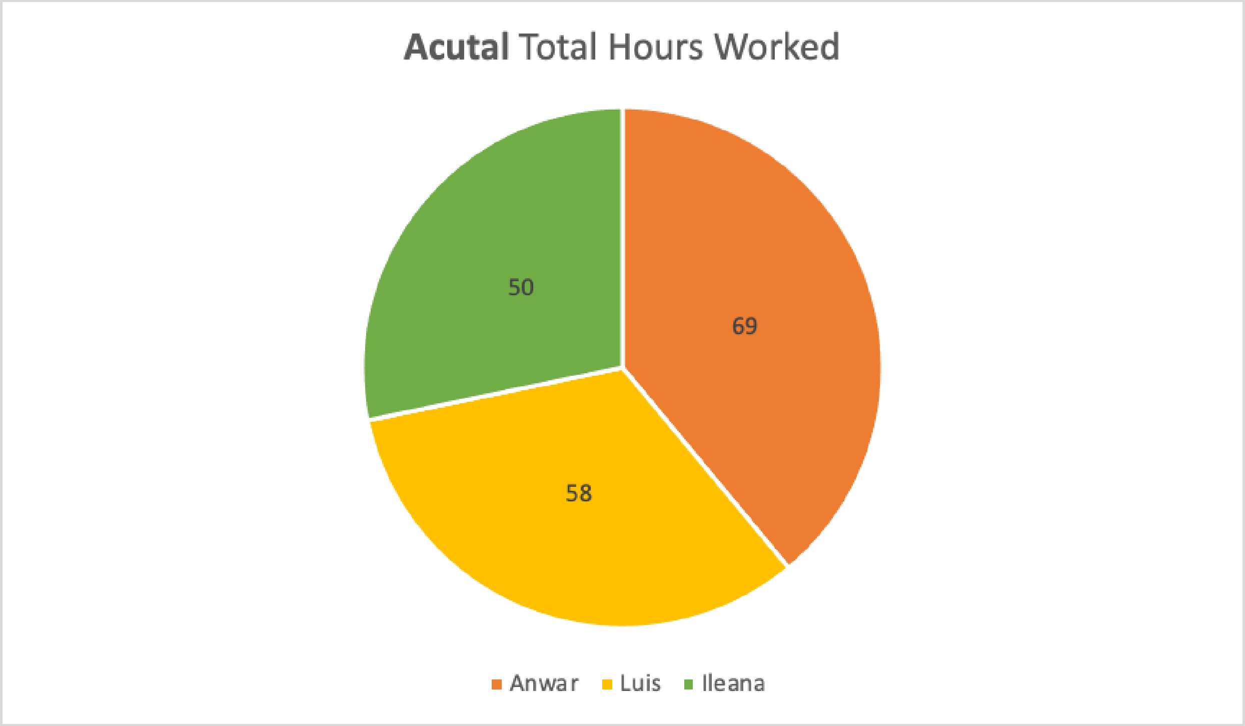
Tables

Task	Anwar Ibrahim	Luis	Ileana
Website	Expected: 3 hrs Actual: 5 hrs	Expected: 3 hrs **Actual:** 5 hrs	Expected: 6 hrs **Actual:** 5 hrs
SQL Database	Expected: 10 hrs Actual: 20 hrs	Expected: 10 hrs **Actual:** 2 hrs	Expected: 5 hrs **Actual:** 2 hrs
Node MCU	Expected: 10 hrs Actual: 15 hrs	Expected: 10 hrs **Actual:** 20 hrs	Expected: 10 hrs **Actual:** 14 hrs
Relay Module	Expected: 2 hr Actual: 1 hr	Expected: 2 hr **Actual:** 1 hr	Expected: 2 hr **Actual:** 1 hrs
Hall Effect Sensor	Expected: 2 hr Actual: 4 hrs	Expected: 2 hr **Actual:** 6 hrs	Expected: 2 hr **Actual:** 3 hrs
Integrate	Expected: 3 hrs Actual: 4 hrs	Expected: 3 hrs **Actual:** 4 hrs	Expected: 3 hrs **Actual:** 4 hrs
Test and debug	Expected: 10 hrs Actual: 20 hrs	Expected: 10 hrs **Actual:** 20 hrs	Expected: 10 hrs **Actual:** 20 hrs


Hours Per Task
12
10
8
6
4
2
0
Hours
Anwar
Luis
Ileana
Web Interface
SQL Database
NodeMCU Wifi+Code
Relay Module
Hall Effect Sensor Connection
Integrate
Test & Debug


Acutal Hours Per Task
25
20
15
10
5
0
Hours
Anwar
Luis
Ileana
Web Interface
SQL Database
NodeMCU Wifi+Code
Relay Module
Hall Effect Sensor Connection
Integrate
Test & Debug


Total Hours Worked
40
40
40
Anwar
Luis
Ileana


Acutal Total Hours Worked
50
69
58
Anwar
Luis
Ileana

Comments

The discrepancies we had for the project was that the web interface took an additional week, the SQL database took about an additional 3 weeks, the nodeMCU was started later and took about roughly the same time as expected timeline, and the test and debug was started earlier and took two more weeks to finish. The way we tried to manage our time during the semester was that we were aiming to finish the project as early as possible. It was what allowed us to actually align with our expected timelines pretty closely for the most part but in terms of actual hours put in, it was more hours than we had initially expected. We made sure to communicate regularly in order to mitigate discrepancies between our expected timelines and actual timelines.

Summary and Future work

Some of the strengths in our design include how cost efficient it is provided the potential materials that could be used, and how it can be used with different web hosting services since we have used php. Some weaknesses of this design include the UI and number of features that could be implemented to make the user experience better in terms of ease of use. It is also how the website can be accessed using http meaning the website is unsecure which was due to the hosting service itself. An improvement that could be made on this design is that the website could be made secure, and there would just need to be minor adjustments made in terms of the implementation as it would be similar with http or https. Something else that could be improved with this project is the delay in which the garage door opens or closes. Another aspect of the design that could be improved upon to better meet the customer needs would be to add ease of use features for the website so the customer would not have to do as much setup.

Reference

PHP and MySQL **Dates:**

https://websitebeaver.com/php-pdo-vs-mysqli 2/13/23

https://www.youtube.com/watch?v=yNoIUEBE3Wc 1/24/22

https://www.php.net/manual/en/intro-whatis.php 2/13/23

https://www.geeksforgeeks.org/php-vs-html/ 11/10/22

https://stackoverflow.com/questions/47043662/php-variable-inside-a-javascript-text-change-onclick 3/5/23

https://www.geeksforgeeks.org/how-to-pass-variables-and-data-from-php-to-javascript/ 3/5/23

HTML

https://www.w3schools.com/tags/att_button_type.asp 4/14/23

https://www.w3schools.com/html/html_examples.asp 2/20/23

CSS

https://www.w3schools.com/css/css3_buttons.asp 4/15/23

NodeMCU

https://diyi0t.com/esp8266-nodemcu-tutorial/#:~:text=The%20voltage%20must%20be%20between,on%20the%203.3V%20pin 11/15/22

https://iotprojectsideas.com/power-supply-board-for-nodemcu-esp8266/ 11/12/22

https://arduinogetstarted.com/tutorials/arduino-mysql 10/25/22

Relay Module

https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/ 1/24/23

Hall Effect Sensor

https://se.rs-online.com/web/generalDisplay.html?id=ideas-and-advice/hall-effect-sensors-guide#:~:text=So%2C%20how%20does%20a%20Hall,sense%20the%20position%20of%20objects. 11/5/22

WebHost

https://www.000webhost.com/ 4/28/23