Uriel Gonzalez

Daniel Delgado Acosta

Pablo Vazquez

CSE 5408

Web Door Lock Final Report

Executive Summary

In this report we're going to discuss the journey of planning and creating a web door lock. We're going to talk about the problem description as to why a web door lock would be beneficial to any residential house. The requirement specification will explain more in detail why those components are crucial to the creation of the web door lock. As in any other project there's always a backup plan if our initial design wasn't going to work out and that'll be explained later in system alternative and alternative selection. In the system and detailed design section it'll go more in depth of the whole layout of the web door lock and the function of how the final product looks. As well as the subsystems that were used in order for the other features of the web door lock to work. To make sure the prototype was ready to go, we had it go through some tests and make sure it passed each and every test in order for the user not to have any problems in the future. We had to consider which parts we had to order for it to be economic for us. If we had more time then we could've spent a lot less but due to the timing we couldn't reduce the price anymore then we did. Managing our times and making sure we hit the deadlines was a big thing and we had to be on top of that. In the project management section we'll talk about how we thought we would meet the initial deadlines but at the end it wouldn't work out like that.

Problem Description

There are times when a person wants to get into their house, but they forgot their keys inside the house. That person would then have to wait for someone who has spare keys or call someone to open the door for them. Furthermore, some people might just find it inconvenient to have to get up to unlock the door in order to let a visitor inside. One way that this problem could be solved is by using a web controlled door lock that can be opened using a web browser from a smartphone or a laptop, removing the need of using a key. The lock will have the controls on a simple website to lock or unlock the door. The door lock will be powered using a rechargeable battery so that the user won't have to worry about wiring the device to a power source. This project aims to have a working prototype by the end of this semester. Only one prototype will be created and tested.

Functional Description of Design:

- The prototype should be ready by May 2023.
- This design focuses on doors found in residential homes.
- Phones or laptops with a web browser should be able to lock or unlock the door through a website.
- The lock should use a rechargeable battery.
- The battery should last at least a day on a full charge
- The lock should have an LED or display to tell the user when the battery is low.
- Lock should be able to withstand a shove.

- The lock should have a backup key option in case the battery runs out, preventing the lock from working

Requirements Specifications

Deliverables:

This project will have four deliverables:

1. A working prototype that will implement all the features specified.
2. A user manual to demonstrate how to connect and use the device to the user.
3. A system specification, which should include block diagrams, describe the functions of each block, and a description of the system.
4. Schematics and circuits needed for the product.

Special Restrictions:

1. All components should be compatible with the ESP32 development board.

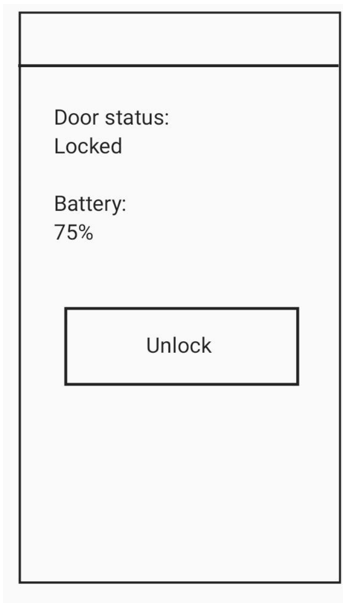
Principle of Operation:

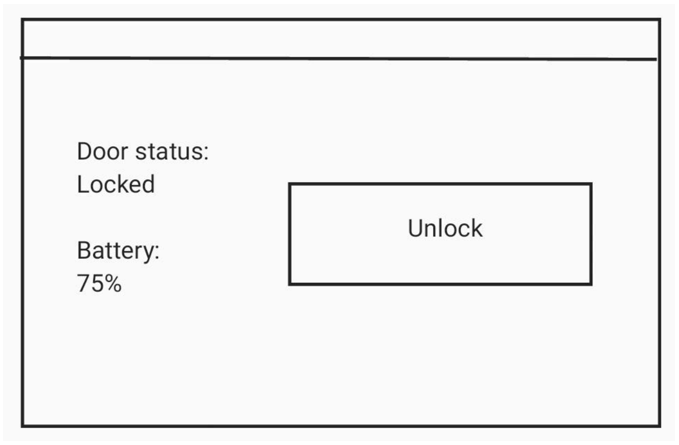
The web server door lock is meant to be used in residential homes. The user will initially set up the door lock to have access to their WiFi network. Once the device is set up, the user can then type the IP address of the door lock to access a website that allows them to lock or unlock the door with the press of a button. The website will be updated showing the status of the door, which is whether the door is locked or unlocked, and will also display the battery percentage.

User Interface:

The lock itself will have one LCD on the front of it. The LCD will have two functions, which is to tell the user how much battery is left at a glance and to briefly show the user the IP address they need to type into the web browser in order to access the door lock controls. The lock will also have a web interface that shows the user the status of the door lock, the battery percentage, and a button that the user can press to engage or disengage the lock and it will also update the door status.

Initial Concept Sketches of Website UI:





User’s Manual:

The web door lock is a way of locking or unlocking a door by simply using any device that can connect to a WiFi network. There is an LCD screen built in that shows how much battery life is left in the device and shows the IP address you need to type into your web browser. The door lock comes with a rechargeable 3.7V battery built in. To charge the device, plug in a USB-C cable to the slot on top of the device.

The following are instructions on how to properly set up and use the device for the first time:

1. Turn the screws on the device until it is properly secured to the door

2. Plug in the battery
3. Connect to the local WiFi network created by the device
4. Input the network credentials and wait for the device to connect
5. Once the device is connected, type the IP address shown on the display into a web browser to access the controls. Simply press the button on the web site to lock or unlock the door.

Description of Inputs:

The inputs of the device are the WiFi credentials in the initial setup, the button found in the web site, and the charging port. A web server will be hosted using an ESP32 and when the ESP32 isn't connected to an existing WiFi network, it will set up its own to allow the user to connect it to their home network by inputting the WiFi credentials. The other input is the lock or unlock button found on the web site, which sends a command to the ESP32 to engage or disengage the and update the door status. The final input is the charging port, which lets the user recharge the battery on their device.

Description of Outputs:

The outputs of the device are the LCD showing the battery level and the IP address of the ESP32, the lock status and battery level on the web site, and the lock physically opening or closing. When the button on the web site is pressed, the status is updated on the website to show the user if the lock is engaged or not. The ESP32 will read the battery level of the LiPo battery and send the percentage to the LCD and web site so the user knows when they need to charge the battery. The other output is the IP address on the LCD, which is shown for a brief moment once the user connects the device to a WiFi network. The final output is the lock physically opening or closing. When the user presses the button on the web site, the ESP32 changes the angle of the servo motor to spin gears in order to lock or unlock the door.

Dispute Resolution:

If the project is not completed by the deadline, we will fail the course. The final decisions of completion will be decided by the professor.

System Alternatives and Alternative Selection

When coming up with the system alternatives, we looked at different locks that were being used to prevent doors from opening and we examined how they operated and how secure they are. The two system concepts that were being discussed were two different types of locks, which were a sliding door latch and a deadbolt lock.

Design 1 - Sliding Door Latch

A sliding door latch works by sliding a metal rod or bolt into a slot in front of a door to prevent it from being opened. This type of lock is mainly used when someone is already inside of a home. This design uses a servo motor controlled by an ESP32 to slide the latch in front of the door to prevent it from being opened. This type of design is lighter, which uses less battery life, since the metal rod is very light and can easily be moved by a lighter servo motor. Sliding door latches tend to be cost effective and should be able to meet most of the requirements. The downside of this design is that it’s not very strong since they tend to be small and light.

Design 2 - Deadbolt lock

Deadbolt locks are one of the most commonly used locks in homes. It works by having a bolt that can slide into a door frame by turning the lock. It can also be opened from the outside by using a key. This design also uses a servo motor to engage the bolt controlled by an ESP32. The advantages of using this type of lock is that it's pretty strong since the bolt is larger, so it's more secure. The downside of this type of lock is that the bolt is heavy, so it might require a stronger servo to move it, which also leads to higher power consumption.

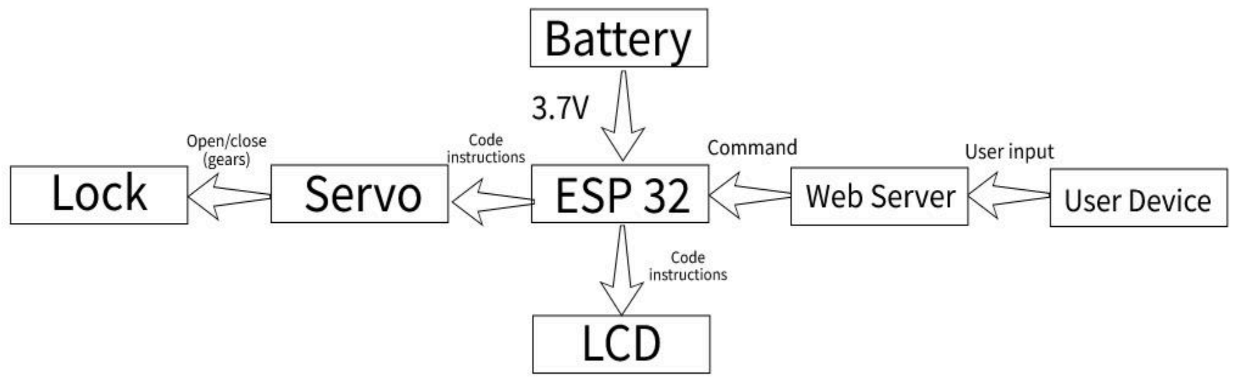
Priority List of Concepts:

- Most promising - Design 2
- Second most promising - Design 1

Design 2 was chosen over the other design because it would provide the most security out of the two. Deadbolt locks are more secure because the bolt is larger while the bolt in the sliding door latch is thinner. Design 2 would most likely cost more battery life to operate but the durability of the lock should help with keeping others out.

System Design

Block Diagram of Chosen Design



Description of Blocks:

In the block diagram for the chosen design, there are a total of seven blocks and each arrow shows how each block interacts with each other.

User Device Block:

This block describes the device that the user is using to access the door lock's controls. The user's device connects to the web server in order to access the web site, and using the button of the server the device can send their WiFi credentials and any lock or unlock commands to the server.

Web Server Block:

The web server is how the user is able to communicate with the ESP32 from a distance. It receives the lock or unlock commands and forwards them to the ESP32 to control the servo.

ESP32 Block:

The ESP32 is the main development board being used to set up the web server and control the LCD and servo. The ESP32 receives the commands from the web server and changes the angle of the servo motor in order to move the bolt. The board controls what is displayed on the LCD, it reads the battery level and IP address in order to send to the LCD.

Battery Block:

A 3.7V 10000mAh Lithium Polymer battery is used to power the device. The battery is rechargeable and powers the entire project.

LCD Block:

A 16x2 LCD is used to display the current battery level of the device so the user can tell when their battery needs to be recharged. The display also shows the local IP address of the ESP32 for a short period of time in order to tell the user what they need to type in their web browser to access the website controls of the lock.

Servo Block:

The servo motor is primarily responsible for turning a gear in order to turn a gear placed on the bolt of the lock. It turns to a specific angle to make sure the door is able to lock or unlock.

Lock Block:

The lock chosen for this device is a deadbolt lock that's placed in line with the door. When the door is locked, the servo spins with a gear to move the bolt into the wall, preventing the door from being opened. The servo spins the opposite direction to retract the bolt. The entire deadbolt lock and all components in the project are encased in a plastic case placed on the inside of the door to keep them in place and protect them.

Detailed Design

The web server was implemented using the ESP32's WiFi library and an external WiFimanager library. The WiFiManager library helps set up a local WiFi network and automatically sets up a website for the user to input their WiFi credentials to connect to their own WiFi. The ESP 32 is powered by a 3.7v 10,000mAh LiPo battery connected using the JST 2.0 interface and is rechargeable through the USB C port on the ESP 32. On initial power up, an LCD will first tell the user to check the WiFi settings on an internet enabled device. The ESP 32 communicates to the LCD using UART communication protocol through the arduino software serial library. When the user checks the WiFi settings on a device, the network with ssid "WiFiMangerAP" should appear. This wireless access point is the ESP 32 hosting a temporary WiFi signal. When connected to the ESP 32 WiFi signal, the user is able to input their home WiFi credentials once they are automatically directed onto the temporary WiFi manager server. After successfully connecting the ESP 32 to their home WiFi, the ESP32 will send its local IP address to the LCD using UART in order to briefly show the user the IP address needed to access the web server through a browser. The ESP 32 will then on remember the WiFi credentials and connect to the same network if ever powered off. Once the web server is set up on the home wifi network, both the LCD and the website will display the battery life of the LiPo battery. The door lock controls site was coded using HTML and CSS. In an effort to conserve battery life, the LCD will dim and continue displaying the battery life. The battery life is obtained using the fuel gauge integrated on the ESP 32 and by using the sparkfun MAX1704X fuel gauge library. The default positional degree of the servo is 110 which is the unlocked state of the deadbolt. When the user clicks the button on the web server the ESP 32 will send instructions to the servo and rotate to the positional degree 60 which is the locked state of the deadbolt. The status of the lock states is updated on the web server each time the user clicks on the button. The servo is mechanically able to lock and unlock the deadbolt with the use of gears. The servo has a fitted gear that rotates and causes a second gear on the deadbolt key shaft to rotate. Two other free rotating gears are placed around the shaft gear causing an even distribution of force to smoothly rotate the shaft of the deadbolt.

System Test Plan and Results

Tests

Test-1: Locking, having the web server functioning with the ESP32 and the servo was needed in order for this test to happen. As well as having the gears connected to the deadbolt and servo. In order to actually start the test the user needs to connect the ESP32 to the WiFi and from there input the ip address on a web server, from there the website will pop up and a simple button will be provided on the website in which you press the button that sends a signal to the servo motor to rotate the gears for the deadbolt to go outwards, locking the door.

Test-2: Unlocking, Same as test-1 but the button tells the servo to spin the opposite direction to retract the deadbolt, unlocking the door.

Test-3: Key functioning, with the prototype fully in place we needed the user to have another way to enter if the ip address was forgotten. Just like any other deadbolt lock, we needed the key to be functionally like a regular deadbolt lock. So the user will use the key to unlock the door, which will move the gears and servo motor without destroying anything internally.

Test-4: Durability, the door lock will be tested on how strong the deadbolt is by shoving the door. The door should be able to survive a shove in order to make sure that the lock is strong enough to be able to prevent others from destroying it to enter the door. If the door lock still works fine after being shoved then this test is considered a pass.

Results

Test 1: Locking

The web door lock was able to pass the initial locking test, it was able to fully engage the lock using the controls we created in the web site. The gears were able to rotate to turn the deadbolt in order to extend it into a door frame to prevent the door from opening.

Test 2: Unlocking

The door lock passed this test, since it was able to fully retract the bolt back into the door in order to allow the door to open. The status of the door was also updated successfully to show the door is unlocked.

Test 3: Key function

The door lock struggled with this test. Although the key mechanism is in place, the key struggles to move the bolt when turned. This is due to the servo being forced to stay at a position when assigned a specific angle to turn, which prevented the gears from turning. When the device runs out of power, the key does function properly.

Test 4: Durability

This test ended up passing. The door lock was shoved in order to see if the internal components were able to stay in place. The device was tested and all the components were able to stay in place and the bolt was still able to extend and retract just fine.

Economic Analysis

Predicted costs:

Materials	**Cost**
Arduino Uno	$18.00
Node MCU module	$8.00
Battery clip	$6.00
Batteries	$13.00
Servo module	$11.00
Jumper Wire	$7.00
Breadboard	$7.00
Lock	$16.00
Led	$1.00
Resistors	$2.00
Total	$89.00

Final cost:

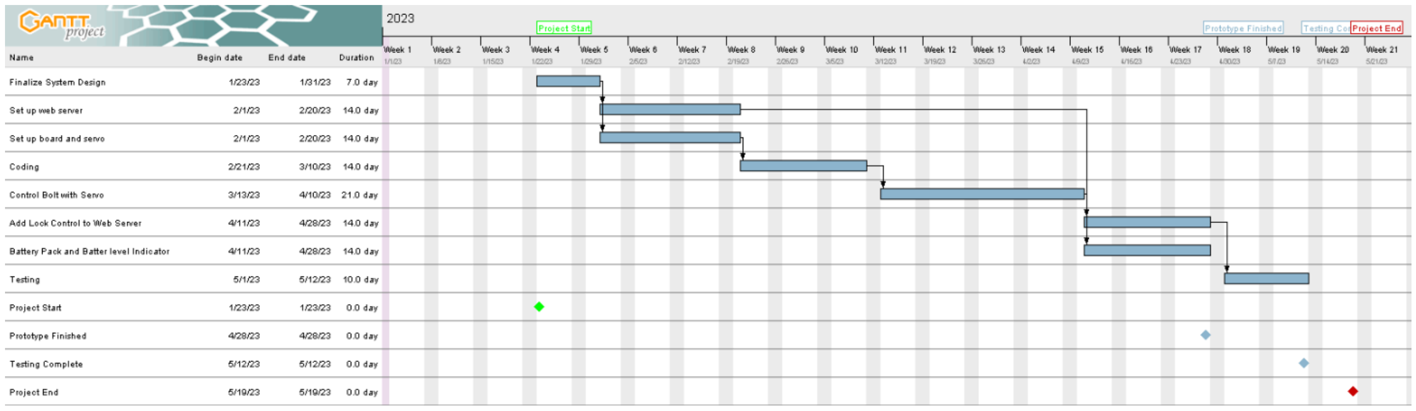
Materials	**Cost**
Sparkfun ESP 32	$30
Gears	$13
Case	$5
LiPo Battery	$18

Servo module	$7
Jumper Wire	$1
Deadbolt Lock	$12
LCD	$20
Total	$106

Economically the project should be a little less than a hundred dollars if we got the parts from another site. Initially we thought that the costs would've been a little less than the final cost but due to timing we had to get the faster shipping site instead. The prototype can be reduced or exchanged parts with other similar cost effective parts. Usually a WiFi deadbolt lock costs about $119.99 or higher and our prototype is a little less than the cheapest one on the market. This being said our prototype can be cheaper if we find parts at a lesser price then the final costs would be around $85-$95.

Project Management

Proposed Gantt Chart



Name	Begin date	End date	Duration
Finalize System Design	1/23/23	1/31/23	7.0 day
Set up web server	2/1/23	2/20/23	14.0 day
Set up board and servo	2/1/23	2/20/23	14.0 day
Coding	2/21/23	3/10/23	14.0 day
Control Bolt with Servo	3/13/23	4/10/23	21.0 day
Add Lock Control to Web Server	4/11/23	4/28/23	14.0 day
Battery Pack and Batter level Indicator	4/11/23	4/28/23	14.0 day
Testing	5/1/23	5/12/23	10.0 day
Project Start	1/23/23	1/23/23	0.0 day
Prototype Finished	4/28/23	4/28/23	0.0 day
Testing Complete	5/12/23	5/12/23	0.0 day
Project End	5/19/23	5/19/23	0.0 day

Actual Timeline of Project:

- Project start and Finalize System Design - Week 1
- Website UI designed - Week 4

- Web Server set-up - Week 4
- Website UI implemented - Week 5
- Servo control added - Week 7
- Connecting gears to servo and bolt - Week 10
- Battery level reader added to website - Week 12
- Gears placed and tested - Week 14
- Integration - Week 14

Tasks proposed in the gantt chart were done in a different order during the actual implementation, due to some tasks being easier to do before others. Controlling the bolt with the servo motor ended up taking longer than expected. The original timeline stated that the task would take four weeks to complete, but instead it ended up taking seven weeks. The bolt control task was started right after the servo control was added, and it was completed in week 14. The main reason why this task took so long to complete was due to the servo not being strong enough to turn the gears. One of the previous servo motors used was also defective and broke when trying to put the gears together. Due to this, we had to do some research in order to find a stronger motor that fit our requirements. The LCD ended up taking up more time than expected to set up, due to missing parts and damage from shipping.

In the project plan, each team member was expected to work on the project for around 5 hours per week. The actual hours spent on the project were 6 hours a week by each team member, since lab hours were also used to work on the prototype. Uriel was in charge of creating the door lock controls page to engage or disengage the bolt, update the door status, and show the battery level on the site. Daniel was responsible for implementing the servo motor control and calibration, implementing the LCD to display the local ip address and show battery level on the LCD. Pablo was in charge of preparing the gears to be placed in the lock and setting up the bolt. All three team members worked together on the gear placement, layout in the casing, integration, and testing.

Summary and Future Work

Strengths

- Connects to WiFi
- Unlock and lock button functioning
- Shows battery level and rechargeable

Weakness

- Design of final prototype
- Key functioning
- Security purposes
- Mechanical parts degrading

The final prototype we went with functioned and delivered the function that we wanted it to do. But there could've been more improvements, for example the design and placement of the parts onto the door to make it more nice and natural. Having more knowledge about gear placements would've helped with using the key in order for the user to unlock the door from the outside. Having a better security system in place so not anybody can access the door as long as they have your ip address, so setting up a login after inputting the ip address on the web server could help against this.

References

- https://randomnerdtutorials.com/esp32-servo-motor-web-server-arduino-ide/ Date accessed: February 22, 2023
- https://www.w3schools.com/css/css3_buttons.asp Date accessed: February 22, 2023
- https://randomnerdtutorials.com/esp32-wi-fi-manager-asyncwebserver/ Date accessed: March 1, 2023
- https://dronebotworkshop.com/WiFimanager/ Date accessed: March 1, 2023
- https://learn.sparkfun.com/tutorials/lipo-fuel-gauge-max1704x-hookup-guide/all Date accessed: Marc 15, 2023
- https://github.com/sparkfun/OpenLCD Date accessed: May 5, 2023
- https://learn.sparkfun.com/tutorials/avr-based-serial-enabled-lcds-hookup-guide/serial-uart-example-code---basic Date accessed: May 5, 2023