Magic Door Sensors

Design Document

Team 09
Daji Qiao (EE, ISU)
Team Members/Roles

Ryan Lanciloti-Report Manager
Ben Pierre-Webmaster
Abdelaziz Hassan-Power Systems
Chinar Kaul-Meeting Facilitator
Alyssa Marshall-Meeting Scribe and Team Manager
Jared Hermon-Test Engineer

sddec21-09@iastate.edu
https://sddec21-09.sd.ece.iastate.edu

Revised April 17th, 2021 Version 3

Executive Summary

Development Standards & Practices Used

- Circuit
  - US power circuit wiring color codes
- Hardware
  - CISPR and IEEE Electromagnetic Interference Standards
- Software
  - PEP 8 and Arduino Styling Guides

Summary of Requirements

- Detect a door's rotation at 15-degree angles.
- Detect a door state from 10ft away (with the line of sight)
- No active sensors on the door
- Arm and disarm the system
- Notify user within 5 seconds of a door opening

Applicable Courses from Iowa State University Curriculum

- Com S 573
- Com S 228
- Com S 309
- CPR E 288
- CPR E 281
- CPR E 488
- CPR E 489
- E E 417
- E E 230
- E E 201
- E E 311
- E E 332
- ENGL 250
- ENGL 314
- STAT 330

New Skills/Knowledge acquired that was not taught in courses

- Wifi CSI Packet collection
- ESP32 programming

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

Figures:

Tables:

Definitions:

- **ESP32** - highly - integrated and low -powered microcontroller with inbuilt antenna switches and added Wifi Module
- **Raspberry Pi Model 3 B**- a miniature single-board computer
- **CSI** - "This information describes how a signal propagates from the transmitter to the receiver and represents the combined effect of, for example, scattering, fading, and power decay with distance" *(Channel state information 2020)*
- **Antenna Design** - a system of multiple antennas and a reflector that is used to transform and reflect radio frequency (RF) signals

1 Introduction

1.1 Acknowledgment

We would like to acknowledge and thank our facility advisor and client, Dr. Daji Qiao, for all of his help throughout this project. He provided us with information on the different technologies we could utilize as well as providing feedback on a weekly basis for where to go next. We would also like to thank Dr. Andrew Bolstad for his input on CSI and antenna design. We would like to thank Dr. Mani Mina and Ph.D. Student Wei Shen Theh for their continuous input and guidance for electromagnetic questions.

Additionally, we would like to thank Steven M. Hernandez and Jonathan Muller for access to their open-source GitHub repositories. Access to these templates has made development significantly more efficient [5],[6].

1.2 Problem and Project Statement

Picture: You're driving home from the airport after a long trip. Before leaving, you installed this new security system that utilizes a wireless door sensor that is powered off of a 9-volt battery and interfaces with a base station somewhere else in the room. As you're pulling up to your house, you notice the door is slightly ajar. You approach the door curious; you're confident you closed it before you left. The security system would have alerted you if someone had broken in, right? Well, the TV, latest-generation gaming console, and most of your valuables are gone. Panicked, you check the door sensor to see what's wrong and notice the light is off. After some tests, you realize the battery you put in your sensor was already partially drained, and it died only moments after you walked out of the door for the trip. If only there were a modern, non-intrusive, wireless, batteryless security system that was easily affordable, attractive, and easy.

Introducing the Magic Sensor, we are proposing a wireless sensor that doesn't utilize a battery of any sort. Modern security systems with a physical sensor attached to doors require a battery that needs to be changed every so often. Checking and replacing the battery adds hassle to the system and can prove extremely problematic if you're away and you get alerted that a sensor has died.

Our solution is to use Channel State Information (CSI) included within WiFi to track the movement of a door to determine if it's opened or closed. Essentially, we would use three transmitters and one receiver to send and receive information about the angle and time of flight of the WiFi signals as they propagate through the room. There will be a reflective piece of aluminum on the door and the door frame that will deflect the signals differently depending on if the door is open or closed.

Our project aims to improve upon solutions to the age-old problem of security in the modern age. WiFi is cheap, and the technology is very well established, so we aim to solve an old problem with modern solutions.

1.3 Operational Environment

Our product is expected to be used inside of a building, away from most harsh environments. Our sensor must work on any arbitrary door; that could be an interior door and an exterior door of the building. The system consisting of the base station and door sensor is expected to not face any outdoor elements. With that statement, it does not need to be waterproof, withstand large impacts, or handle extreme temperature changes. As such, it must work under a semi-wide range of temperatures, and we chose between -10° C and 40° C inclusively.

1.4 Requirements

Functional:

- Detect if a door is open o r closed with 85% accuracy with false-positive reporting around 5% to 10% and false-negative reporting less than 5%
- Detect a door's state up to 10 feet away from the base unit
- No powered no wired sensor on the door
- Alert client via an application on their phone within 5 seconds if a door opens or closes
- Capable of arming and disarming system

Non-Functional:

- $300 budget for total system
- The door module must be less than $30 to allow a user to add more sensors to their system in the future
- UI must conform to current design stylings
- The door module must weigh less than one pound and be less than 4" by 4" when installed

1.5 Intended Users and Uses

This product will be deployed by a homeowner/business owner/property owner who wishes to have a non-powered, tamper-resistant system of detecting door states. This user wants to have the system installed by a professional and then never touch it again. The only interaction needed between the user and the system is arming and disarming the system via the application on their phone.

1.6 Assumptions and Limitations

Assumptions:

- The system will only be used inside
- Doors will be within 10 feet of the base station
- The building will have WiFi so that the ESP32 is capable of relaying data to the server
- Expecting the user to understand English as our application will be only written in English
- The user owns an Android phone

Limitations:

- The door module will be no more than 4" x 4"
- The door module will not exceed more than 1 lb

1.7 Expected End Product and Deliverables

- Security Hub
    - A suite of ESP32s and a Raspberry Pi, packaged into one tamper-proof package.
- Android Application
    - Installable from the app store, the only configuration is signing in.
- Server
    - Capable of being deployed on any Linux box, should only need port forwarding to set up.
- Setup Guide
    - A short instruction manual that shows the maintenance person how to set up our product.

Due to the fact that all of our deliverables will require constant modification as other deliverables evolve our final iterations will be delivered roughly simultaneously. We hope to deliver this mid-Q4 2021.

2 Project Plan

2.1 Task Decomposition

- Develop Door Module
    - Analyze room layout
    - Develop antenna system layout
    - Test aluminum door module with ESP32s
- Develop ESP CSI Harvesting Network
    - Develop Code for ESP32s
        - Refactor original code

        - Establish harvesting routine
        - Allow multiple channels
    - Interface with a remote server
- Develop Inferencing on Server
    - Feed data from ESP32s
        - Interface with ESP32s
        - Establish a protocol for handling several streams
    - Generate prediction as to door state
        - Feed normalized CSI ESP data into TensorFlow or equivalent
    - Interface with android app
- Develop Android Application
    - Interface with sever
    - Allow arming and disarming
    - Setup notifications upon state change
    - Polish UI if time allows

2.2 Risks And Risk Management/Mitigation

- Develop Door Module
    a. The reflective piece of metal will not be able to create the binary representation that we need, so we will test various ideas as early as possible to determine the best design to proceed forward with for this project.
- Develop ESP CSI Harvesting Network
    a. Packet unreliability will be mitigated with multiple sensors
    b. Sensors could be tampered with and as such will be put in tamper-proof boxes
- Develop Inferencing on Server
    a. Machine learning could be unreliable, as such we will analyze several different algorithms
    b. The server could be compromised, as such, we will implement passwords and use Iowa State's built-in protections.
- Develop Android Application
    a. Android Apps could be susceptible to not receiving alerts, as such we will implement a two-way handshake for each notification to ensure the end-user is notified.
- Team Related Risk
    a. A team member quits. To mitigate this we have assigned multiple developers to each deliverable, ensuring that if a team member leaves there is someone knowledgeable on the task

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

1. Develop a concrete design for our system as a whole
2. Put together functional code on our ESP32s
3. Parse through data from ESP32s into human-readable output
4. Develop a machine learning algorithm that can interpret and train on data from ESP32s
5. Achieve an inference accuracy of 70%
6. Improve inference accuracy to 85%
7. Design and develop a way to retrain the model efficiently

2.4 Project Timeline/Schedule

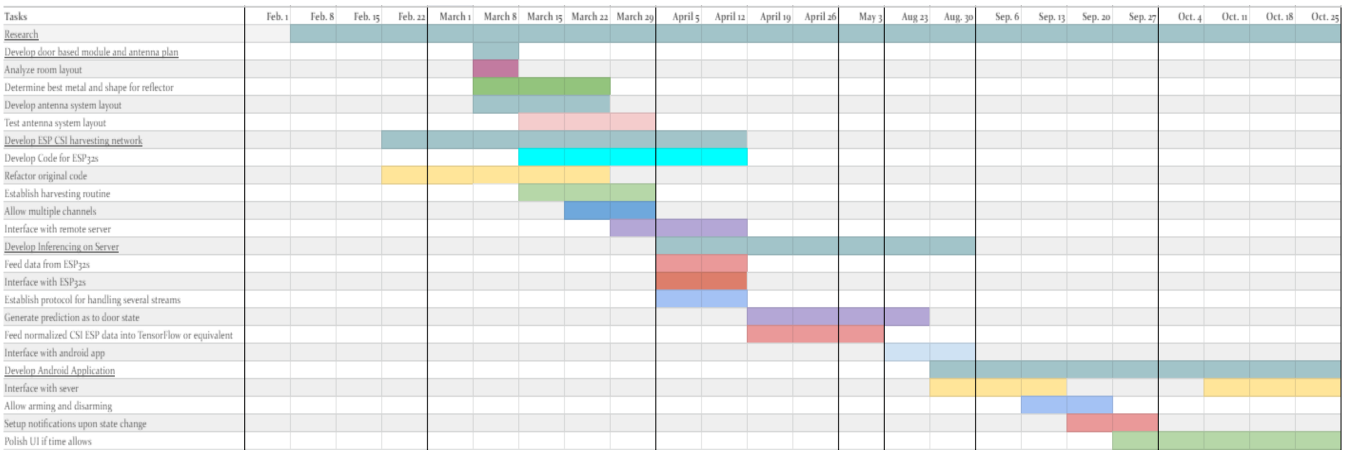

Figure 1: Gantt Chart; https://drive.google.com/file/d/1ewsoJ3gYMU2Lc9ELXNJ_bIwuRRAUpxsC/view?usp=sharing

2.5 Project Tracking Procedures

Our group utilized Github for keeping track of all software we used for this project. We made use of Google Drive for the sharing and editing of all documents which we have created for this project. Once we were able to develop a solid plan for how to execute this project, we used Trello to keep track of features and milestones in development.

2.6 Personnel Effort Requirements

Table 1: Time estimation for each task.

Task	Number of hours
Writing ESP 32 Code	15
491 presentations and documentation	16
Researching	24
Establish machine learning on ISU servers	18
Develop Android Front End	15
Develop Aesthetic Front End	10
Link systems via MQTT	10

2.7 Other Resource Requirements

- 2 ESP32
- 1 Raspberry Pi Model 3 B
- Aluminum sheet for door module

2.8 Financial Requirements

We do not have a concrete budget for this project, however, our client has specified that our final iteration must cost $300 or less for one base station and one sensor, connected to our server. The cost per additional sensor must be less than $30.

3 Design

3.1 Previous Work And Literature

From the research we have done, there currently does not appear to be a wireless home security system that utilizes CSI for detecting door states. The current door sensors on the market either require the physical sensor to be wired in or require a battery to operate and use a reed switch for detection. This project is both riveting and daunting. It is riveting because we are essentially researching the technology and its applications as it is still an abstract technology. However, it is daunting because, going into the project, we have no idea about the feasibility of CSI in a real-life security application. We are doing the R&D and experiments to figure out if we can develop a system that can detect the state of a door. For the relevant documents, please see section 6.2.

3.2 Design Thinking

There were various ideas that were brought up during the ideation process. One potential idea was imaging processing using OpenCV and Tensorflow. However, we did not proceed with this idea as it did not fit our client's needs and there was privacy concern. We also considered adding a servo to the hinge of the door to generate a current that would power a relay when it was opened, but this was deemed too invasive. Additionally, we intend to allow the user to install this product themselves, which would not be possible with this approach. Our final plausible idea was to put a pressure-sensitive switch on the door that would relay data to a base station however this idea was also deemed too invasive. Once we determined CSI was the best system to use, we generated multiple ideas for the door module. We thought about using an RF harvesting system with a hall effect sensor on the door, but this was deemed too complex to complete in our necessary timeline.

3.3 Proposed Design

There are two possible methods of design: one involves CSI packets with a constant feed of phase and amplitude values, and the other method is putting a pressure-sensitive sensor on the door that emits a pulse when the door changes.

Upon suggestion from the client, we have opted to take the CSI approach, and have started testing this approach with a blend of open source code and our homegrown code. The system is composed of the ESP32 acting as a transmitter, an additional ESP32 functioning as a receiver, and a reflective piece of aluminum on the door in a triangle formation. The input of the system is the state of the door, and the output of the system is an analysis based on the door state resulting in an "open" or "closed" decision. The movement of the reflective piece of metal is used to create a change in amplitude to the RF waves that our CSI system will be able to detect and measure. The CSI data will be analyzed by a deep learning algorithm to accurately detect the door state. This whole system will be able to fulfill our functional goals of accurately detecting the door state with low false-positive and false-negative readings.

The specific metal chosen for the door module was aluminum as it fulfilled the functional requirement of not having an active door sensor but still helping provide the door state due to its high reflection ability. Additionally, aluminum filled all of our non-functional requirements for this project. Aluminum was one of the most cost-effective choices when considered against copper. Additionally, aluminum was chosen as it does not rust over time compared to steel, which was another considered choice. Aluminum provides the minimum alteration to the appearance of the door and fits our weight and size requirements.

As detailed in our functional requirements and the diagram below, the base station will be up to 10 feet away from the door.

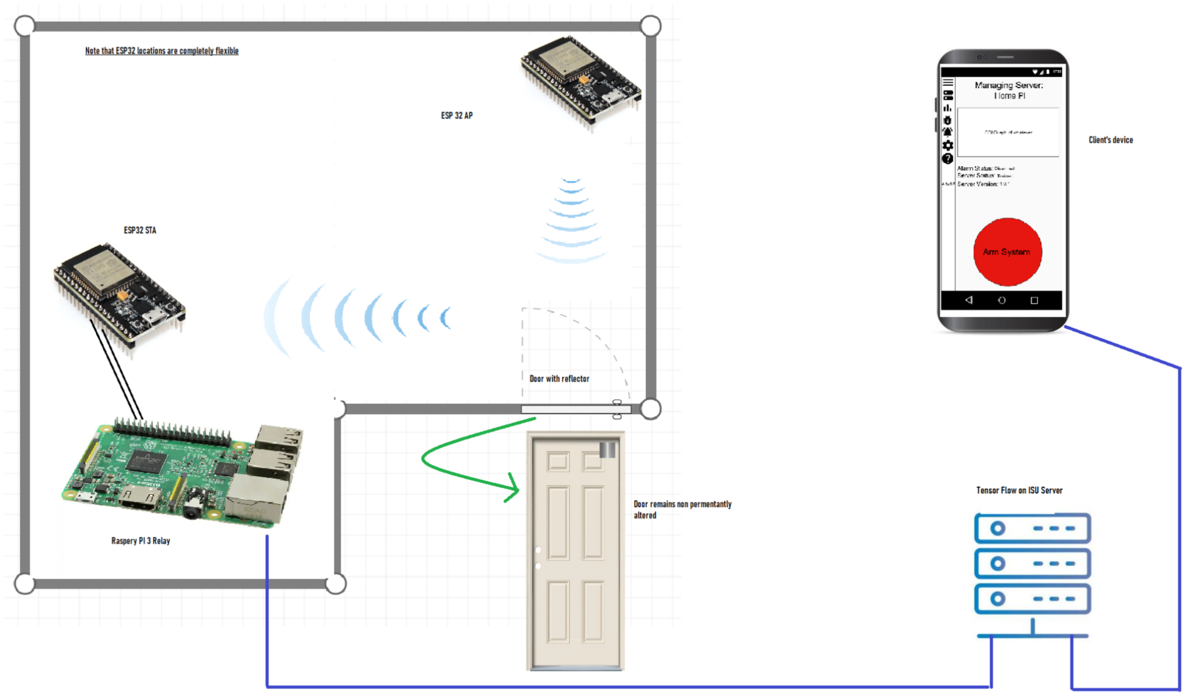

Figure 2: Design Sketch

3.4 Technology Considerations

One big issue we are running into is that thus far only one consumer-level microcontroller is available to analyze CSI packets, and to the best of our knowledge only one or two papers exist that actually implement this technology. Additionally, CSI Preambles seem to be notoriously unreliable and thus require several sensors for redundancy. The upside of this technology is that they are cheap and very easy to duplicate should the need arise.

Other solutions, such as placing an RF harvesting circuit on the door would increase reliability and deployability. However, they would negatively affect the client's vision and would require an unaesthetic implementation of the sensor design on the door.

3.5 Design Analysis

Our proposed design should work; however, we cannot comment on its accuracy at this time. We have yet to reach a point where we can assess this. If we see a lack of precision or functionality, we can tighten our machine learning system's bounds. If this does not work, we can get drastic and increase the number of transmitters while decreasing the frequency. This will give us a larger array of more robust systems. However, it will require a dedicated transmitter module if we stray from the bounds of typical wifi frequencies.

3.6 Development Process

As a group, we are familiar with Waterfall, Agile, and Test-Driven Development (TDD). From what we understand, Waterfall is a linear form of development and the least iterative out of the three. With Waterfall, we would gather requirements, design a solution, test, and deliver a final product. A major problem with Waterfall development is that if during the later stages of development, a bug is found or if a client changes the project requirements, engineers would have no way of integrating changes in the project unless they completely start over.

On the other hand, Agile is an iterative development process that encourages team collaboration and quick deliverables. The benefit of Agile is that it accounts for flexible client requirements. Agile teams are structured and include roles such as Scrum Master and Product Owner. Additionally, Agile teams work in a structured timeline. For example, teams have a Daily Standup every day during the sprint, then reflect on the past and future activities in the Sprint Review, Sprint Retrospective, and Sprint Planning meetings.

Test-Driven Development tightly integrates testing into the development process. A new test is written every time new technologies are introduced to guarantee every part of the project is thoroughly tested. A key feature of TDD is developing as little of the project and as little of a unit test as possible during each iteration to ensure that no part of the project goes untested.

Our team believes that a flexible version of Agile development will best suit our needs. Agile development will allow our team to produce deliverables quickly and regularly. Additionally, Agile will allow us to modify our project easier in case of bugs or changed requirements.

3.7 Design Plan

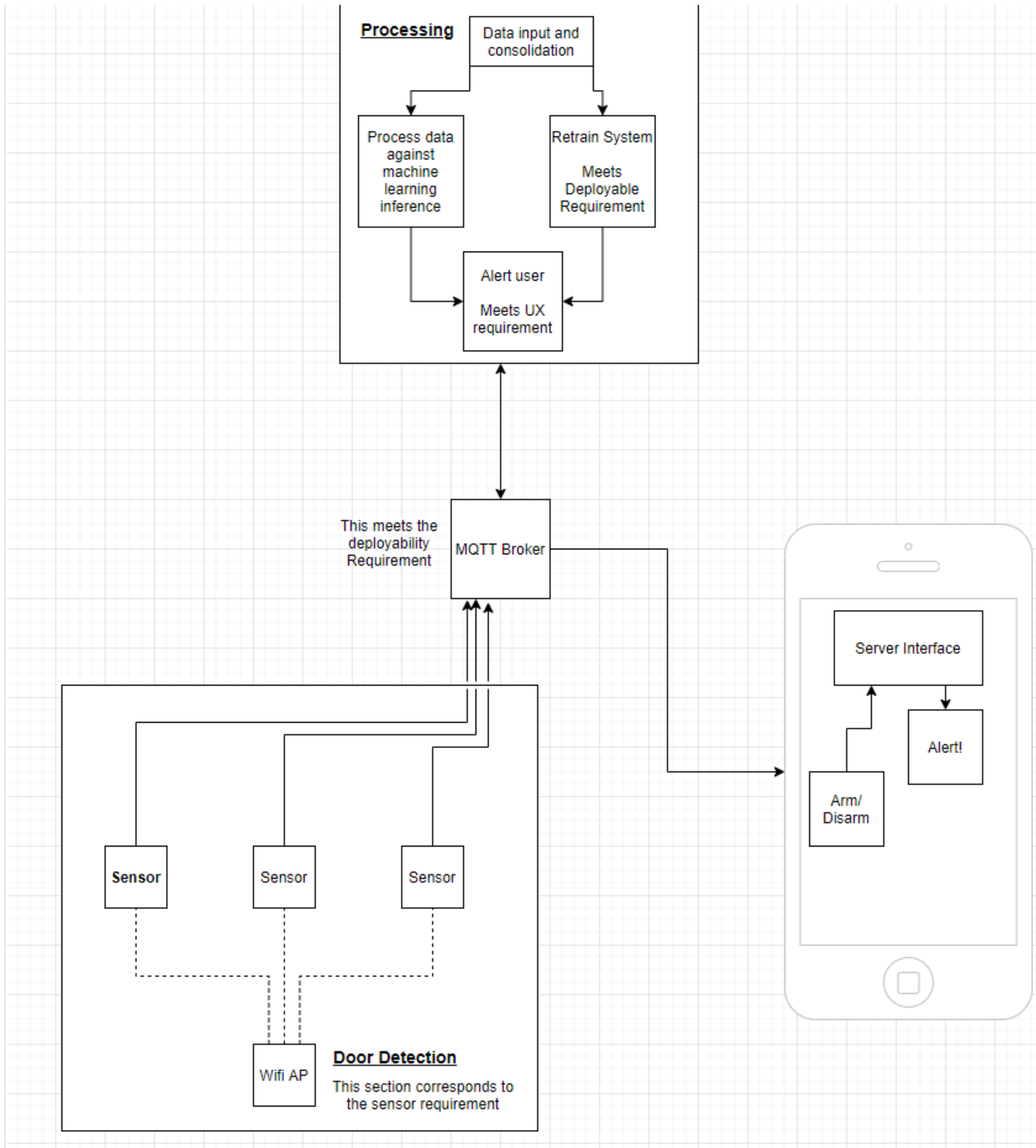

Figure 3: Design Plan

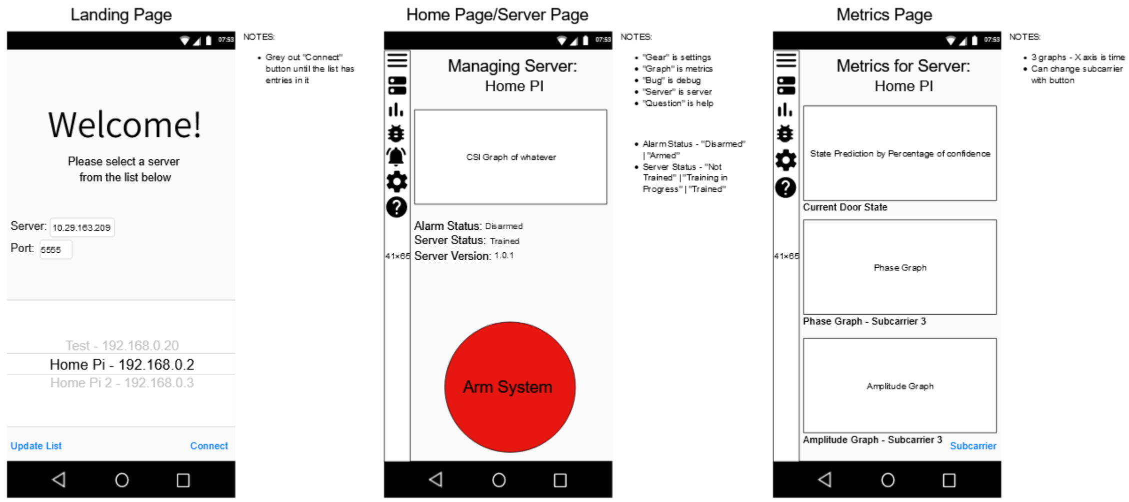

Figure 4: App Design Mockup (Screens 1-3)

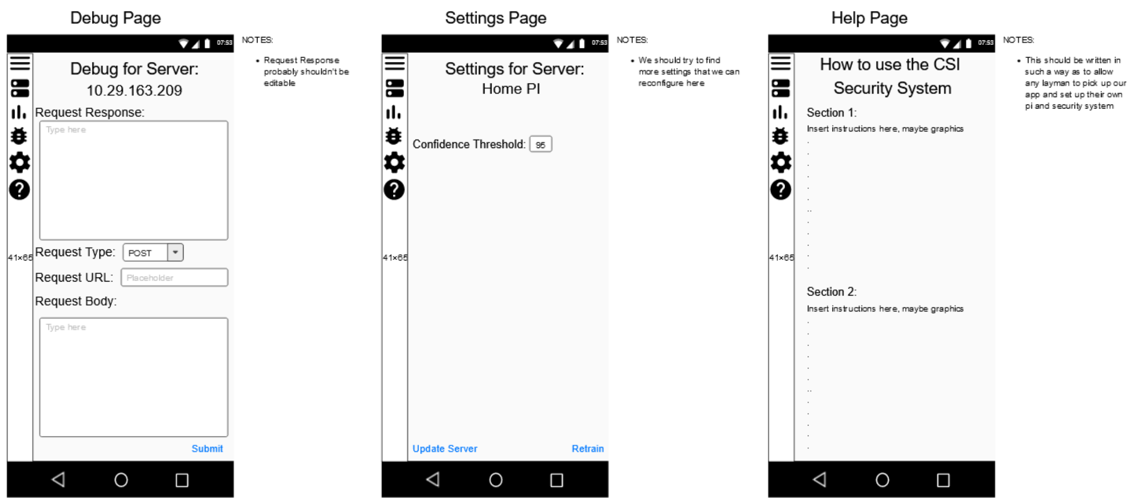

Figure 5: App Design Mockup (Screens 4-6)

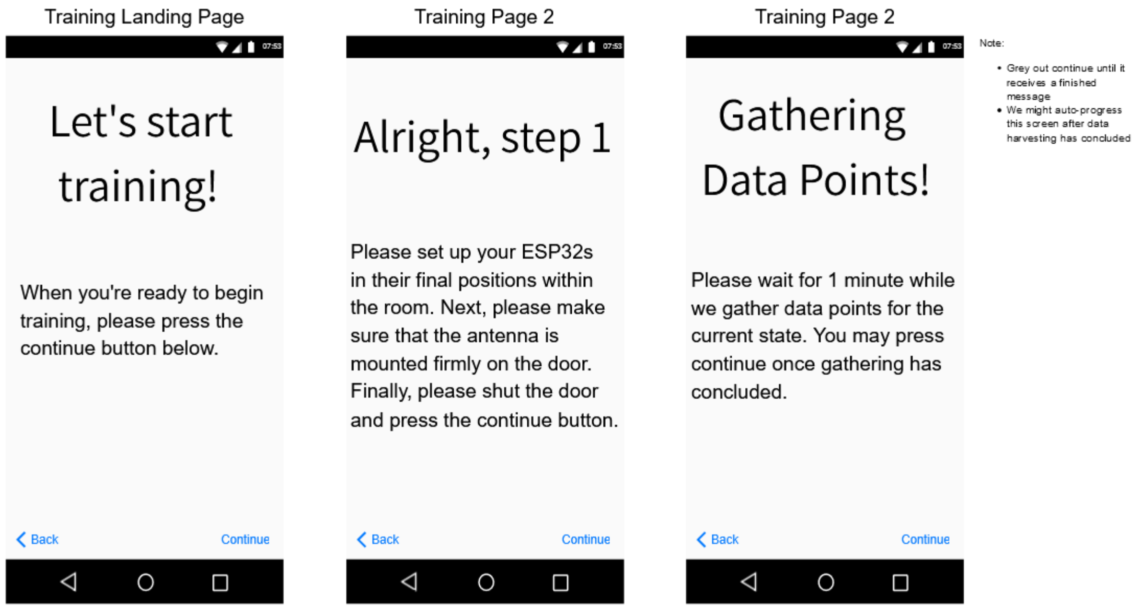

Figure 6: Model Training Mockup (Screens 1-3)

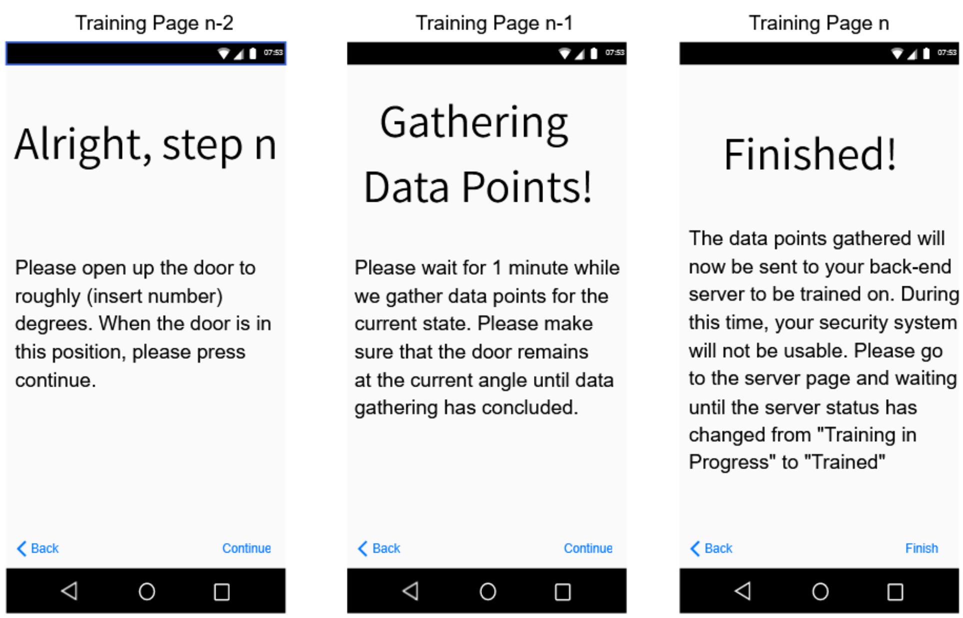

Figure 7: Model Training Mockup (Screens 4-6)

This design meets all of our functional requirements, while also breaking the system into smaller modules that can be built and tested individually.

4 Testing

4.1 Unit Testing

Currently, we are doing unit testing for our ESP32s, backend API, and Android app. One ESP32 establishes an Access Point (AP) and the other ESP32 connects to said Access Point; thus both subsystems must be tested. Unfortunately, the two are intertwined, and as such, both must be tested together.

Moving forward, we must test the component that handles our python link between the ESP32s and our server. Our server is located on Iowa state's network, while our tools are external. As a result, ISU ETG has set up a reverse proxy to allow us to forward our data onto the network securely. This link will require flow testing. To accomplish this we intend to use a 'ping' like system where we send packets and measure the successful delivery percentile.

Another component that must be tested is the effectiveness of the door module. This component can be tested individually once we have meaningful CSI data implemented with the ESP32s. The various designs consisting of different materials were considered. With research, aluminum was deemed the best material as it reflects the RF waves most effectively for our purpose. Additionally, we have started with a 4" by 4" flat piece of aluminum. Once we are ready with the CSI system, we will test a parabolic shape instead of a flat shape to see if we can maximize the CSI system's efficiency. Additionally, we will begin decreasing the size of the metal until we determine the best size to use where we still receive meaningful results but minimizing the alteration to the appearance of the door.

While not established, the final component that needs to be tested will be our machine learning system. This component will receive packets from our python relay and interpret them to determine door states. This component can be tested individually with constant inputs to verify we generate the expected information.

4.2 Interface Testing

We will use two interfaces in our project; one will be in the form of a Message Queuing Telemetry Transport (MQTT) server that routes packets in the appropriate direction. The second will be our python bridge between our ESP32s and our MQTT server. Our final interface will be what handles the android application's communication. Each interface will be tested individually.

We will test our MQTT interface by sending packets to the box and determining if the software correctly interprets them. We will monitor this interface in real-time, so we intend to send it real packets from our clients and observe how it handles each to ensure it is functional. While this will not allow us to test it in isolation, it will expose bugs that will only show up with scale testing.

Our python bridge will be easy to test as we will use a Serial Communication Simulator to simulate serial data and then observe how it handles packets. This set-up will allow us to verify this interface is working correctly with a large scale of packets.

The final interface to test will be between our android application and our MQTT server. While currently not implemented, we plan to test this by pushing text messages to the phone and verifying that notifications are working as expected.

4.3 Acceptance Testing

We intend to prove acceptance testing by setting up a demonstration to our client in which we have the entire system running and set up. We will demonstrate that opening and closing the door does update the state on the android application. This will prove functional requirements have been met.

Our non-functional requirements are easy to demonstrate as they consist of visible features such as the sensor is less than 4 by 4 inches in size. These non-functional goals will be obvious at a demonstration, as mentioned earlier.

4.4 Results

Since we don't have any tests run on or against our system, we do not have any results to report at this time. We expect to eventually have inference accuracy scores from our machine learning model when it is coupled with CSI.

5 Implementation

We have divided our project into categories: ESP Design, Server Design, Antenna Design, and Machine Learning Design. We have assigned individuals to each of these categories and will be working independently with frequent meetings to test and synchronize development. The physical aspect will be deployed in spare space at the house of several group members, while the server side aspect will be deployed to a server hosted on Iowa State's servers.

6 Closing Material

6.1 Conclusion

Thus far we have taken the previous group's project and redeployed it within our own testing environment. Additionally, we have explored several routes of detecting the door state that differ from the previous group's implementation. Through this, we arrived at the conclusion that CSI data is an acceptable method of detection, and will be going forward with it. This method allows us to reposition sensors with no additional engineering, and scaling the deployment can be done at little to no cost to the user. Additionally, this method is the least intrusive to a user's home.

6.2 References

[1] Al-Qaness, M. A., Li, F., Ma, X., & Liu, G. (2016). Device-free home intruder detection and alarm system using wi-fi channel state information. International Journal of Future Computer and Communication, 5(4), 180-186. doi:10.18178/ijfcc.2016.5.4.468

[2] Brescia, F., Gringoli, F., Brescia, U., Darmstadt, M., Schulz, M., Darmstadt, T., . . . Authors: Francesco Gringoli University of Brescia. (2019, October 01). Free your CSI: A Channel state information EXTRACTION platform for MODERN Wi-fi Chipsets. Retrieved March 04, 2021, from https://dl.acm.org/doi/abs/10.1145/3349623.3355477

[3] Ma, Y., Zhou, G., & Wang, S. (2019). WiFi sensing with Channel state information. ACM Computing Surveys, 52(3), 1-36. doi:10.1145/3310194

[4] "Channel state information," Wikipedia, 18-Aug-2020. [Online]. Available: https://en.wikipedia.org/wiki/Channel_state_information#:~:text=In%20wireless%20communications%2C%20channel%20state,and%20power%20decay%20with%20distance. [Accessed: 03-Apr-2021].

[5] "ESP-IDF" Jonathan Mueller[Online]. Available : https://github.com/jonathanmuller/esp-idf

[6] "ESP32 CSI Toolkit" Steven M. Hernandez[Online]. Available: https://stevenmhernandez.github.io/ESP32-CSI-Tool/

[7] H. Guan and D. D. L. Chung, "Absorption-dominant radio-wave attenuation loss of metals and graphite," Journal of Materials Science, vol. 56, no. 13, pp. 8037–8047, Jan. 2021, doi: 10.1007/s10853-021-05808-2.

6.3 Appendices

N/A