

Quinton Miller

Caleb White

Rocky Chen

Dr. William Plymale

ECE 4564

Network Applications

Final Project Proposal

Concept of Operations

In mine and Caleb's experience in the Marching Virginians over the past four years, one of the most important things to the band's success is for the member to attend rehearsals. However, it can be difficult to incentivize members to attend rehearsals even with the group being a class with a grade. This is because the directors have not had a very robust system for tracking which band members are at each rehearsal. The current system has student leadership taking attendance through a google survey for each rehearsal which is a system that is cumbersome and difficult to work with once the season is over because the survey does not track automatically track each student's attendance, the directors need to go into each survey and manually track the attendance. Our system will be an improvement on this by allowing the band directors to automatically tally the attendance of students through software instead of them having to manually tally attendance. Students will interact with the system by scanning their fingerprint on an fingerprint reader to mark themselves as present. We expect this system to make the grading process much easier for the directors because they will not have to spend the time to manually tally each student's

attendance throughout the entire season. It will also increase the accuracy of the grading because the system will remove the human error that comes with tallying each student's attendance manually.

System Overview

Our system will consist of two raspberry pies, one acting as a server and one acting as a client. The client raspberry pi will be equipped with a fingerprint reader that will be used to scan student's fingers. The user will tell the client Pi that it needs to take a scan then the user will scan their finger with the scanner. The client Pi will take the information from the scan, reference a MongoDB collection with the registered fingerprint location to ensure the student is registered and send a post request including the system time and date of scan which will then get sent to the server Pi. The server Pi will then take this information and decide whether the student was on-time, tardy, or absent. Once the server has received the data and saved it to the database, it will send a response back to the client that the data was received. Once the client receives this signal, it will light an LED light through the GPIO pins as well as make a beeping sound to tell the user on the client end that they were marked present. While this is happening, the server will store the post request information (attendance info) as well as whether the student was on-time or tardy in a MongoDB database. If the student has not scanned their finger before the end of rehearsal or a time determined by the director, they will be marked absent. Finally, the director will be able to print out a dump of the database to facilitate the assignment of final grades at the end of the semester using get requests from curl commands or a web browser.

Testable Requirements

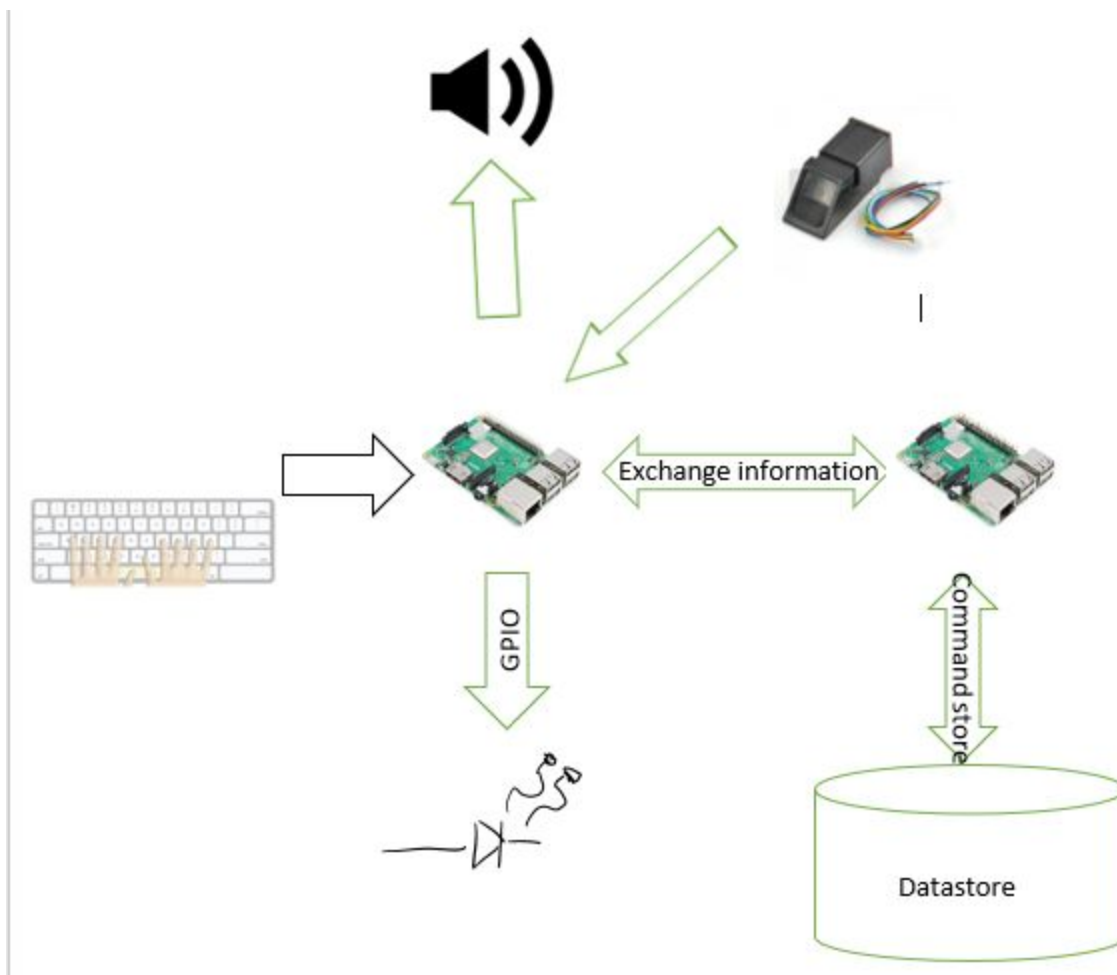
- The system will be able to scan a fingerprint to get an image and match it with a registered student's fingerprint model
- The system will be able to send post to a server through the python R and send a signal back to the client that the data was received.
- The system will be able to light an LED and make a beep when the client gets the message received signal.
- The system will be able to store the data transferred to the server Pi to a MongoDB database in a way that will make the director's job of assigning final grades much easier.
- The system will be able to store the data from the post request to a MongoDB database and return a database summary through a get request.

Hardware List

- Two raspberry pi's
- [Small beeper for reader pi](#) (\$~2)
 - <https://www.ebay.com/p/Portable-3-5mm-Pillow-Speaker-For-MP3-MP4-CD-iPod-Phone-Samsung-Radio-White/6011374186?iid=322078878008>
- Indicator LED for reader pi
- simple [fingerprint](#) reader kit

- Use adafruit_fingerprint python [library](#)
- <https://www.adafruit.com/product/751>

Diagram of System



Timeline and Milestones

- Sound and LED **4/14/19**
 - Produce the sound we will use to confirm reading
 - Produce output to LED for a controlled period of time ($\frac{1}{2}$ second)
- Setup database server and tables on server pi **4/18/19**
 - Get MongoDB running on Raspberry Pi
 - Design the data tables we will use
 - Student table
 - Has 'location' associated with a fingerprint model and student ID and Name
 - InOut table
 - Links in/out events with a studentID from Student table
- Flask Webserver
 - Basic HTTP authentication
- Fingerprint reading **4/28/19**
 - Register students' fingerprints for attendance taking later
 - Properly read a fingerprint in python, matching the student Name and ID from MongoDB database and collection

- Package date and time and student info retrieved from fingerprint query and send over post 4/30/19
 - Save record (document) to MongoDB database on server pi
 - Send response to confirm document was saved to database to the reader pi
 - Here the client will use the LED to tell whether read was saved or not
- Design a 'report' that you can run (get) to view attendance summary of that day in a text file. 5/4/19