**Facial Recognition Rack Mount System**

CPE 495 Interim Report

December 2016

**Project Summary**

The Facial Recognition Rack Mount project is a system-level hardware/software design with a primary intention of keeping server rooms safe and secure. With physical security of modern datacenters being often overlooked, the goal of this project is to provide the means of increasing that security.

**Team Description**

Jared Nixon – **Team Lead**

Interests: GUI Design and Design Patterns

Garrett Eledui – Software Lead

Interests: Graphics Development and Image Processing

Jason Parker – Hardware Lead

Interests: Cybersecurity, System Administration, and Machine Learning

Daniel Hasty – Software Support

Interests: Software Design and Cybersecurity

**Introduction**

Facial Recognition Rack Mount – Sponsored by **Emerson Network Power**

**Marketing Requirements:**

* The system should have a camera, processing unit, and means of turning a lock on a server rack.
* The system should be easy to use with minimal experience
* The system should have the ability for the user to request entry and for the uer to know whetehr they were authenticated or not.

**Engineering Requirements:**

* Performance – The system should unlock the rack door in no more than ten seconds from the point when access is requested.
* Functionality – The system will process the photo and relay an access granted or access denied signal to the actuator.
* Functionality – The system will auto lock after 60 seconds regardless of if the user is finished with the rack. Upon finishing with the rack, the user may have to press the button again to unlock the rack (if the timeout period has already passed), then close the door.
* Usability – Users of the system should be able to unlock with the push of a button and have user feedback.

**Background**

**Competition:**

* AMG Product: contains facial and fingerprint recognition, and has a database.
  + Pros: Product is pre built with very impressive features.
  + Cons: Product costs $499.00

**Projects:**

* Windows Team IoT: Facial Recognition Door.
  + Team development project that uses facial recognition to unlock a door.
  + This is accomplished using Microsoft API: Project Oxford.
  + API seems to be open to use.

**Tradeoff Analysis of Design Alternatives**

*Rasp Pi vs Microcontroller* – The Raspberry Pi 3 is the processing unit of choice because it already has facial recognition software (OpenCV) readily available for use with it. It also has more processing power with a clock speed of 1.2 GHz and 1 GB of RAM.

*Camera Module vs. Standard Camera* – The Raspberry Pi Camera Module is the camera of choice because it plugs directly into the Raspberry Pi and is designed for specific use with it. It is capable of capturing static images at 3280x2464, which is more than enough for the camera need of this project, and the camera software is supported with the latest version of the Raspbian Operating System.

*OpenCV vs Windows API* – OpenCV is the approach decided on for the facial recognition software. Though the Windows 10 API has already been proven to work, OpenCV is already directly compatible with the Raspberry Pi 3.

*Push Button vs Automatic Photograph* – Push Button is the more viable approach here because it allows the user to notify the system when they are ready to have a photo taken. Using automatic photographing can be very CPU inefficient depending on the polling time, whereas a push button will allow one photograph, and involve significantly less code.

*C++ vs Python* – C++ is the more efficient language to use with this system, as Python is more oriented around a scripting language. Algorithms with OpenCV can be written in either C++ or Python, so the API is not the problem. Compiled languages also tend to be much more efficient than interpreted languages. The team is also signficantly more familiar with C++ than with Python, which will save research when implementation begins.

*Push vs Pull Actuator* – For the sake of power consumption, an actuator that stays outward and retracts when current is passed through would be the most efficient actuator to use. This will ensure that power does not need to be applied 99% of the time.

**Response to Feedback – Responses to main constructive feedback only.**

1. Engineering Requirements – We decided to not use this feedback as only a basic prototype was asked of us. Our project could lead to other existing projects in the future, such as recognizing specific faces, after the framework is created. Setting that as a requirement in this semester could set us up for not completing in time, but we would still like to look into other features granted time permits the team to do so.
2. Existing Projects – We decided to definitely take this feedback and put references into our powerpoint as to where the existing projects exist (though we didnt because that was a resource we wanted to keep to ourselves). This is very beneficial to someone wanting to look more into it.
3. Functional Decomposition – We decided to take this feed back in its entirety. Our functional decomposition diagram should have been shown while talking about the system, rather than leaving the slide on the description of the system. It would have been very beneficial for everyone to SEE the system visually as the speaker talked about it.
4. Risk Analysis – We decided to indirectly use this feedback. We most definitely need to add some risks to our presentation and assess those. In concerns to the software being too large for the raspberry pi, many people have previously put it on a raspberry pi, so we know this is a definite thing that can be done.
5. Test Plan – This feedback we fully implemented and narrowed down some specific test cases. Originally, we did not realize we needed to state specific cases that needed to be tested, but only approaches/types of testing that we intended on using.
6. Cost Estimation – We are giving this feedback consideration. We don’t think it is necessary to have another Raspberry Pi just in case, but it is good to have extra components for the outer works of the system. Things such as extra RGB LEDs will be good to have on hand in case one blows.
7. General Questions and Comments – The team will be using OpenCV on a Raspberry Pi, during our presentation, some confusion was thrown with the reference to the Windows 10 project. We have not already tested any facial recognition software because we have nothing to test it with. The Raspberry Pi has 1 GB of RAM with a 1.2 GHz clock speed. OpenCV has been used on its linux based Raspbian operating system, so we have no concern that this piece of hardware will not have enough processing power. The team does plan to have some kind of physical element for demonstration purposes in April.

**Global/Societal Impact of Project**

**Impact of the datacenter environment:**

* The project will have an impact on datacenters from a space management standpoint. Most server racks are filled to the rim to get the most efficiency out of them, meaning there is not much room to mount a raspberry pi. Our design will be placed in what is called the ZERO U space of the rack (i.e, the left or right side where space is not allocated for servers / power management units), so server space will not have to be sacrificed.

**Impact of modern securities:**

* The goal of this project is to increase security on rack mounts. Bearing in mind this is a basic prototype, there still will be vunerabilities such as the door behind the rack not being locked, authorized / non-authorized faces not being distinguished, and brute force break-ins. This project is meant to begin the R&D process of an eventual secure system.

**Test Plan**

*Unit Testing*:

* Can the button be pushed and acknowledged?
* Can a picture be taken and sent to the CPU?
* Can CPU recieve the picture when taken?
* Can the CPU verify a face is present?
* Can the CPU send voltage command to actuators?
* Can the CPU send voltage command to LED?

*Integration Testing*:

* Can the button be pushed and trigger a picture to be taken?
* Can the CPU recieve the photo and verify a face is present / not present, and send voltage out to appropriate peripheral?
* Can the actuators / LED recieve voltage and take appropriate action?

\*\* The integration testing is stringing together multiple unit tests to be sure they can work as a result of an action.

*Regression Testing*:

* Not specific tests can be listed for regression testing because the team does not know problems that will happen during the development process. The point is to ensure fixing a problem does not introduce other problems.

*Acceptance Testing*:

* See marketing requirements and compare to actual functionality. The marketing requirements are the only guidelines for this testing process.

**Work Breakdown Structure and Project Schedule**

*Milestone One* – Deadline of November 4th

* Research hardware/software integration and compatibility.
* Research facial recognition algorithms.
* Research actuators that are likely to be successful.
* **DELIVERABLE: Project Proposal Presentation (11/2)**

*Milestone Two* – Deadline of November 11th

* Submit hardware components needed to Thom based upon research findings.
* Decide on appropriate coding language (C++, Python, etc.)

*Milestone Three* – Deadline of January 27th

* Facial Recognition lock in the most basic form (will open for anyone).
* Ensure the lock will NOT open for objects, and strictly for faces of people.
* Validate the camera takes successful samples of person accessing rack.
* Validate current flows only one way to the actuator before full hookup (use a diode).
* **DELIVERABLE: Preliminary Design Review (January 2017)**

*Milestone Five* – OPTIONAL – Deadline of March 31st

* Distinguish between authorized and non-authorized faces.
* Log for successful IDs when access is granted.
* Manual override for possible hardware failures.
* Implement the three tier system from project proposal.

*Milestone Six* – Deadline of April 14th

* Test the final prototype of the project.
* Verification / Validation processes

*Milestone Seven* – Complete Final Design

* **DELIVERABLE: Final Design Review (April 2017)**

**Course Team-Specific Deliverables**

* Project Proposal Presentation – November 2nd, 2016
* Preliminary Design Review – January 2017
* Final Design Review – April 2017

**Professional and Ethical Statements of Responsibility**

* Jared Nixon – Ensure deadlines are made, keep team headed in the right direction, and support the software development process.
* Garrett Eledui – Develop / Integrate facial recognition software.
* Jason Parker – Develop communication between system components.
* Daniel Hasty – Develop software where help is needed to maintain deadlines.

**Project Cost Evaluation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item Needed** | **Quantity** | **Cost** | **Total** |
| **Raspberry Pi 3** | **1** | **$39.95** | **$39.95** |
| **Case** | **1** | **$5.00** | **$5.00** |
| **Case Lid** | **1** | **$3.00** | **$3.00** |
| **Camera Module** | **1** | **$29.95** | **$29.95** |
| **SD Card (comes w/ Raspbian Jessie OS)** | **1** | **$11.95** | **$11.95** |
| **Solenoid** | **2** | **$4.95** | **$9.90** |
| **RGB LED (25 pack)** | **1** | **$12.50** | **$12.50** |
| **Push Button** | **2** | **$1.50** | **$3.00** |
| **TOTAL:** | **10 items** | **$108.80** | **$115.25** |

**Conclusion**

The final prototype of this project will result in a functional rack mount system that triggers actuators / leds based upon a face being in the view of the camera and a button being pushed. This prototype has an enormous amount of room for expansion in future projects as team Face-Off and future teams lead Emerson Network Power in the direction of a marketable, secure system that can be integrated into datacenters for increased physical secure.

**Appendix A – Design Schematic:**

