Secure Door Opener Milestone 3

Team Members:

Name	Email
James Pabisz	jpabisz2020@my.fit.edu
Christopher Kiefer	ckiefer2019@my.fit.edu
Warren Smith	wsmith2019@my.fit.edu
Luke Bucher	lbucher2017@my.fit.edu

Faculty Advisor:

Dr. Marius Silaghi - msilaghi@fit.edu

Client:

Dr. Marius Silaghi - Graduate Professor at Florida Institute of Technology

Progress Matrix:

Task	Percent Completed	James	Christopher	Warren	Luke	To Do
Camera Drivers	40%	40%	20%	20%	20%	Implement on 2k camera
Implement Image Recognition	80%	20%	40%	20%	20%	Incorporate model on server
Learning to interface with Raspberry Pi	10%	35%	25%	20%	20%	Unlock lock. Connect to server, and communicate between server, camera, and doorlock.
.apk & server interaction	50%	20%	20%	40%	20%	Finish connecting features to the server.
Backend endpoint development	60%	20%	20%	20%	40%	Need websockets and IOT backend.

Task Discussion:

Camera Drivers:
Implemented a small camera driver for the Logitech C920 camera available in the technology lending service at the library using C and a raspberry pi. This was done to get a feel of how to write a driver. Modifications were made using inline assembly language to try to make optimizations which will be needed if streaming is expected. This process will be repeated and refined for the 2k cameras that we have for the project. The drivers for this camera may already be included but optimizations will still need to be made.

Image Recognition:
Data set for facial recognition was refined and low lighting images were added. Various pretrained models were tested and the models were refined. The next step is to finish doing metric testing to see what models will perform best on our server and then implement it on the server.

Finished creating a model for image and facial recognition on AWS. Ended up going with AWS lambda to handle the execution of commands to actually process the images. The lambda functions are using the AWS object recognition to find all of the things in the image. If there is a face involved then check to see if the face is actually one of the accepted people. The model is hard to use without a lot of images to compare the face to and as such it has a low confidence when comparing specific peoples faces. This is the only thing holding this part of the recognition back and as such the comparing feature is not implemented in the milestone.

Also created a bucket system for the facial recognition software to pull pictures from an uploaded source, mainly the database. Started the development of using the recognition software to be able to process videos rather than just pictures. This is a more complicated process and involves having a steady feed to fully implement but a majority of the ground work is done.

Created the necessary rules and roles for AWS to allow the use of the buckets with the lambda functions. This was a necessary switch from using a local host to just have all of the images/videos stored on there.

Raspberry Pi:
Wrote small programs for the Raspberry Pi and got a feel for integrating it with various hardware such as a camera and a light. Learned to connect it to wifi. The next step is to connect it to the server and doorlock. So unlocking the door with the raspberry pi must be done first. Eventually, We will have the raspberry pi communicate with the lock, camera, and server.

.apk and server interaction:
Created a login screen and the capability to add users to the database. Aesthetics were refined and Add approved visitor and Camera feed buttons were added (currently unconnected). The next step is to get full interconnectivity between the app and the server. This will include a way to add pictures of approved visitors to the database. An Add unapproved visitor is going to be added so that the app can alert users of concerning visitors.

Begin Backend Endpoints:
Refined routing for handling incoming requests. Current routes handle Authentication, Login, Dashboard navigation and reAuth. Finished setting up a SQLite database to store the needed information for both user profiles and device information. Finished the integration of the Login UI to authenticate users through JWT (JavaScript Web Token) to focus on token based authentication within the app.

Began to interconnect the Raspberry Pi to the backend and tested communication between them.

While working on the facial recognition experimented with using the different backend tools that AWS offers such as its media processing template to handle the video feed from an IOT device and also alerting the admins of the device if there was a person detected in the video feed. This was not fully explored because of cost as it was close to $1200 a month for full video processing and the other services. This can be an option to look into if we would like to streamline the application and find a way to reduce the amount of processing and cost.

Member Contributions:

James Pabisz:
Implemented a driver for Logitech C920 in C
Optimized driver with in-line assembly
Experimented and got comfortable with programming with a Raspberry Pi
Added additional features and buttons (some unconnected to server) to the app.
Connected some features and buttons to the app.
Refined the aesthetics of the app to match project branding.
Experimented with wifi features on Raspberry pi.

Christopher Kiefer:
Created AWS Lambda functions for facial recognition
Created bucket system for facial recognition
Finished standard facial detection/image processing
Started work on video processing
Created Roles/Rules on AWS to allow lambda functions to pull images and videos from other sources
Laid foundation for facial recognition to be start being integrated into server
Found tool to allow users to be notified when a face is detected

Warren Smith:
Added additional functionality to the app
Helped with Facial recognition
Focused on implementing security minded features to the app
Made Milestone 3 Presentation

Luke Bucher:
Finished login Component within the .apk
Finished navigation from Login to main Dashboard
JWT Handling for User Authentication and Request Management
Polished the established Routing Through Express.JS
Refined Node.JS backend to handle requests
Adapted SQLite Database to Handle User accounts and Store

Milestone 4 Task Matrix:

Task	James	Christopher	Warren	Luke
Implement 2k Camera drivers	40%	20%	20%	20%
Connect Raspberry Pi to server and hardware	30%	20%	20%	30%
Implement Facial Recognition on server	20%	40%	20%	20%
.apk & server connectivity	20%	20%	40%	20%
Finish backend endpoints	20%	20%	20%	40%

Milestone 4 Task Discussion:

Camera:
Driver will be implemented on the 2k camera.

Raspberry Pi:
Connect the Server with the Hardware

Facial Recognition:
Implement the model on the server.

.APK:
Finish Connecting features to the server

Backend Endpoints:
Finish the Backend Endpoints and begin communicating with the hardware.

Meeting Dates:

Date	Topic
November 17, 2022	Discussion of progress on the assigned tasks and current status of the hardware component and reviewed current difficulties and their possible solutions.

Faculty Advisor Feedback below:

Task 1:

Task 2:

Task 3:

Task 4:

Task 5:

Approval from Faculty Advisor

"I have discussed the milestone with the team. I have evaluated the progress and will assign a grade for this milestone."

Signature: _____ Date: _____

Evaluation by Faculty Advisor:

Score (0-10)

James	0	1	2	3	4	5	6	7	7.5	8	8.5	9	9.5	10
Chris	0	1	2	3	4	5	6	7	7.5	8	8.5	9	9.5	10
Luke	0	1	2	3	4	5	6	7	7.5	8	8.5	9	9.5	10
Warren	0	1	2	3	4	5	6	7	7.5	8	8.5	9	9.5	10