Initial Project Document

Facial Recognition Locker

University of Central Florida
Department of Electrical Engineering and Computer Science
Senior Design I

Group 4
Bryce Dere - Electrical Engineering
Che' Baptiste - Computer Engineering
Julian Boaz - Computer Engineering
Ryan Wiegman - Computer Engineering

Project Narrative

As technology continues to develop, things are becoming more automated and more safe than ever before. Machines have been developed in order to manufacture the chips that go into our computers and phones, driving assistance AI have been incorporated into newly made cars to help keep drivers in their lane while driving. These are just a few of the many examples where technology has not only made our jobs easier, but has made tasks safer to perform. One of the more popular trends that has come up in recent years are food delivery apps, namely doordash and uber eats. They deliver your food and leave them at your doorstep, and in theory this is fine, however, with this comes the opportunity for someone else to come and steal your food. Our proposal is to make a device that can only be opened by the delivery person and the user so that nobody else can steal your food.

While this locker's sole purpose is not to hold food, it will be small enough for it to be placed outside of one's house or apartment. It will be a small box that will incorporate a cooling mechanism, about the size of a microwave. It will have a camera on the top of the box that will only be active when there is motion in front of the box. When the camera is activated it will start looking for a face that matches that of the picture of the user or of the delivery driver. If the facial recognition decides that the face in front of the camera and that of the picture match, then the box will open. Otherwise it will stay closed. If however, the person decides to try to break into the box, an alarm will be set off and a notification will be sent to the user.

The companion app that was briefly mentioned will only be available to the user. It will contain information like the location of the box as well as the ability to upload an image of the driver's face. Additionally, from the app you will be able to tell if the item that will be placed in the box is an item that needs to be cooled. If it is, when the item is placed in the box, the cooling system will kick in and keep the item cool. The temperature that the box will be at will be a set temperature, so the user does not have to worry about what temperature they need to set the box to. The app will also serve as a way to notify the user when the food has been placed in the box. This will be done through weight sensors in the box, and when the microcontroller registers that something has been placed in the box, a notification will be sent to the user. The application companion that will go along with the device will allow the user to be able to communicate with the device from anywhere they so wish. The only time they will have to touch the box is when they go out to grab the food. Additionally, the application will feature a way to open the box from the device in case of a face recognition failure.

The purpose of the box will not only serve as a way to temporarily store your food, it will serve as a way to help drivers identify your house. This is especially the case for those who live in apartments, as it is very easy for drivers to get lost in complexes where all of the rooms look the same. With drivers having an easier time identifying your room, they will be less likely to call and ask for help and/or directions, making their job easier as well. Additionally, so that the delivery driver does not get confused, when the box senses motion and confirms that the person in front of the box is the delivery driver, the box will say “Please place the food here” and open the door for the food.

Additionally, the device will also have a few anti theft features in order to prevent stealing. Since the box will share its location with the user, the user will be able to know if someone moves the box and where it is at all times. Additionally, if someone tries to forcefully open the lock or break into the box, a siren will be set off.

The overall audience for this device is for any household that gets food delivered often. More and more people are using food delivery apps and when the food is left outside of your door, it is very easy for someone to just walk by and take your food. Additionally, it is not unheard of that some delivery people will simply take your food without delivering it. With this device, it will help deter anybody from stealing your food.

The device will have some constraints, for example, we don’t want the device to be constantly plugged in. If the user has a house, they might have access to a socket on the outside of the house, but this will most likely not be the case for those who live in apartments. Therefore, the box will feature a battery pack, so the user will have to occasionally switch these out. In order to reduce the amount of times the user will have to do this and electricity consumption in general, the features the motion sensing device previously mentioned.

The device is not meant to be expensive or exclusive in any way, but simply a way of securing your food or items using cutting edge technology. The objective of this device is to simply provide users with a safe and efficient way of securing their food. We want to make it a relatively painless and stress free process of getting your food. This is especially the case when the user might not be able to get their food as soon as it gets delivered. While things continue to become more automated in our lives, we must take into account the risks that come along with it. The risks that come along with getting your food delivered is that someone might steal it, and our device seeks to combat and reduce said risk for all users.

Project Requirements:

Project Constraints

Due to the nature of the project, there will be a number of constraints in order for the box to successfully complete its functions with limited resources. One of the constraints for this project as a whole is time. This is especially concerning due to the fact that since we plan on finishing the project during the summer semester, we lose 2 weeks compared to if we were to finish it in spring or fall. Additionally, all members of the group are full time students. This means that all members of the group must split their time between other classes as well as spending time on this project.

Additionally, we don't have a sponsor for this project. This means that we have to pay for all of the components for the project with our own money. As college students, there is not an ample amount of money we can spend, therefore it is a factor we need to take into consideration. For example, the box may not be as accurate as we aim it to be, because we won't be able to afford the best quality camera available.

Space is also a constraint that needs to be considered when planning and making this project. This is especially the case when considering what the box will be used for. We aim for the box to be small and be able to be placed in both homes and apartments, something that you can set out on a table. However, as non mechanical engineers, it is beneficial for us to set aside some of our budget to purchase an anti porch pirate box rather than build one ourselves. As a result, the box we aim to build might take up more space than we might have hoped, however, with both time and money being a concern for this project, this is most likely the best course of action. One of said boxes available for purchase can be seen in Figure 1. Additionally, the places where this box can be utilized is also something of concern. This box can only really be utilized where people live, as companies or schools will not have small objects delivered to them often. Food is also an existing restraint for the project, as food safety standards are also an important factor to consider. As a result, how long food can stay in the box and if it needs to remain cold could be limiting factors as well.

Figure 1. Purchasable anti theft box

The last constraint for this project is that we, for academic purposes, have to use Amazon Cloud Services. Amazon Cloud Services provides us, as students, an amazing opportunity to use a tool widely known and utilized across a variety of companies. By using Amazon Cloud Services, we will be able to say we have the ability to use these services and have even implemented them in our own way.

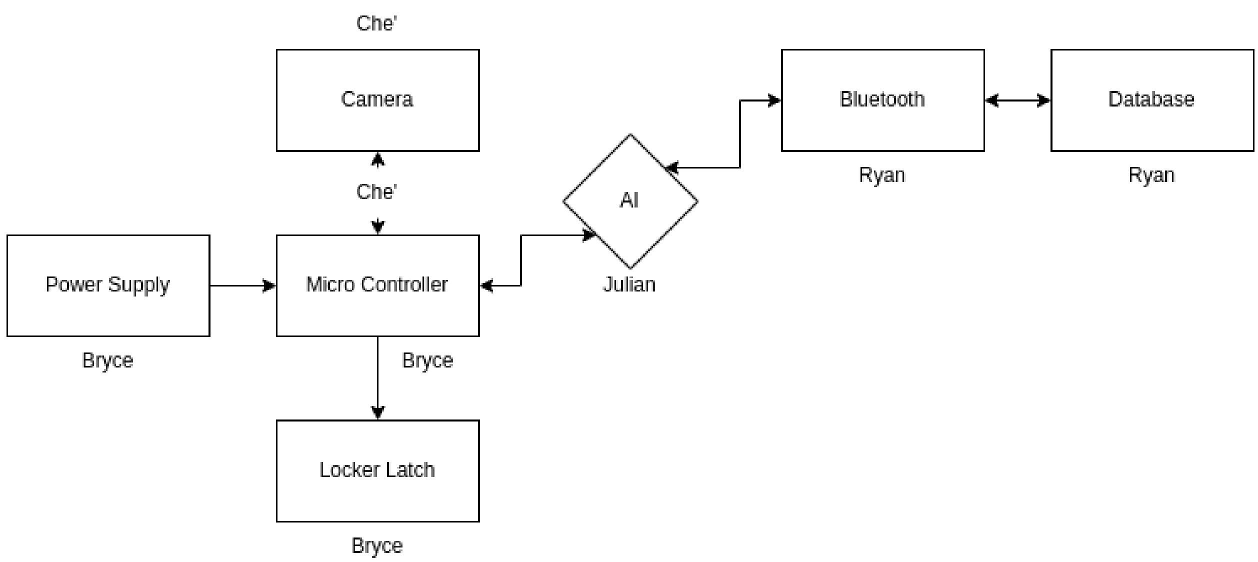
Table 1: Quantitative Measures

1.1	Must keep food at proper temperature (heating 40-60 C)
1.2	Must keep moisture within acceptable levels (<65% humidity)
2.1	Must be secure (locked against unauthorized users; ANSI Grade 3)
3.1	Supply DC voltage for MCU (and other peripherals, etc.)
3.2	Camera needs high enough resolution to detect human face (~100 pixels at medium to short distance, >720p resolution)
3.3	Indoor Bluetooth range (~10 m)

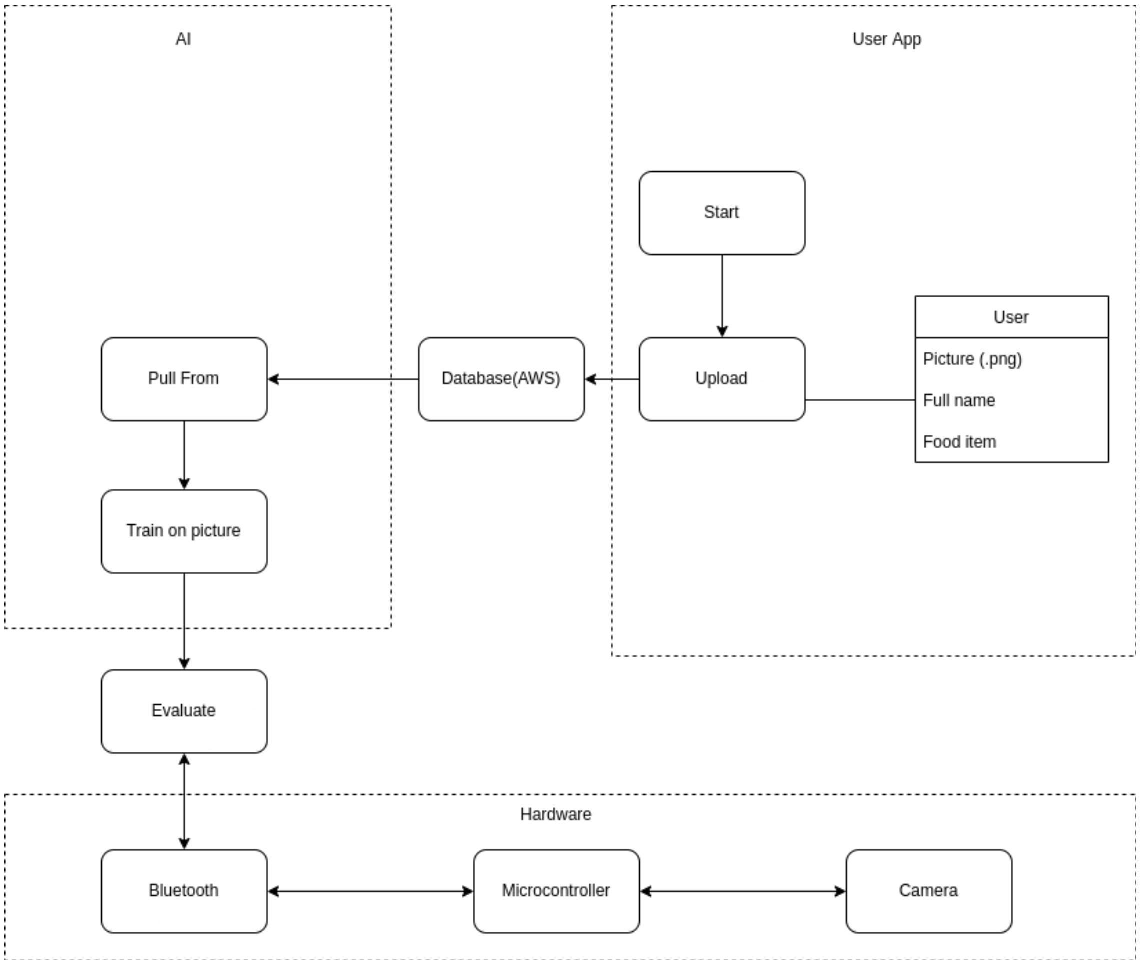
Project Block Diagrams:

This first diagram is to describe how all of the hardware will interact with each other. So the microcontroller is the heart of our system. Bryce will be working to connect a bluetooth module and camera module into the microcontroller. The bluetooth module will be able to send and receive data from the server that contains all of the faces of the users that have orders, one of which will be selected of course, Ryan will be working on this. The camera will be continuously polling to see if the current object on screen is actually the user waiting to pick up the food, and Che' will be working on this. Julian will be working on the AI that works on taking in data from the Camera and matching it to the user that has been selected. Once the AI has determined that the object on screen is the user, the microcontroller will unlock the latch on the locker, allowing the user to take the food inside.

Hardware Diagram (Figure 2):



Software Diagram (Figure 3):



Project Budget:

For the budget, each individual person would like to stay at around the 75 dollar range. This means that our total budget would be around 300 dollars.

Item	Price
Microcontroller	~$50
Lockbox	~$50
Bluetooth module	~$10
Camera module	~$30
Electronic latch	~$30
AWS Server	~$0.016 per hour
Power supply	~$10

Table 3: Planned Budget

This budget highly depends on two factors. The microcontroller and camera will be dependent on how we decide to integrate the AI. Our budget will have to be updated if we decide to use a Raspberry Pi and an Infrared Camera for facial recognition. We could save money by going with a cheap microcontroller, or we could have the Raspberry Pi which make for an easier integration of hardware. The Raspberry Pi's vary greatly in price, as well as camera modules. However, a standard camera module for the Raspberry Pi (RP branded) is approximately 30 dollars, and the Raspberry Pi could end up being over 100 dollars.

Another factor that could contribute to a budget change would be the price of the lockbox, due to this being strictly a Electrical and Computer engineering project, we've decided to buy a pre-built lock box which depending on brand could be anywhere from 30 to 100 dollars as well.

Milestones:

Initial milestones for both semesters

- Gather materials
- Good accuracy with facial recognition
- Working, small scale lock mechanism
- Security feature, in case someone tries to break in
- Incorporate locking mechanism with facial recognition

Week	Task
1	Project idea
3-6	Initial project documentation (D & C I & II)
7-9	Project Research, Collaboration, and Selecting Parts
10-12	Prototype Development and Testing Components
13-14	100 Page Documentation Draft
15	Review Documentation
16	Submit Documentation

Table 4: Senior Design 1 Milestones

Ideally, the most research we will have to do in order to complete this project will be on optimizing the AI model for any given dataset, the only other issue will be integrating all of the software with the hardware, especially using bluetooth to transmit data from one unit to another. This will likely take most of our time in development, however, once our foundation is created, the rest of the work should follow. The prototype in the following weeks will be set up to handle primitive test cases, simply for establishing a proof of concept. Optimization of all systems will come once our test cases have been accomplished efficiently. We have over committed time to testing to allow for debugging within communications and integration, as that has been identified to be the most cumbersome. We could likely continue to perform debugging on the software side once we're at week 13 and just commit new updates to it, to ensure that everything works smoothly. We will take a week to review the documentation we've written to make sure it is congruent. Otherwise the document will be submitted.

Week	Task
1	Retest Components and Double Check Design
2-7	Build Prototype and the Program
8-11	Test Prototype and Fix Bugs
12	Finalize Project and Presentation
13	Prepare for Presentation
14	Final Presentation

Table 5: Senior Design 2 Milestones

When we make our way to Senior Design 2, the first thing we need to do is make sure that everything still works. We need to make sure to do this immediately because if we lack a crucial component, the whole project could go down in flames. In addition to that, we need to make sure the design of the project is what we want is good, and that we did not accidentally miss anything. After those preliminary checks, we can move on to building the prototype and working on the code. We have the most time allocated to this part of the project in Senior Design 2 due to the fact that we want to make sure we have enough time to implement everything we want. Additionally, this will easily take us longer than any other part of the project. After building the project, we will need to test it and fix any bugs we might come across. We've allocated 3 weeks to this, not only to have ample time to identify and to fix any bugs, but to have some room in case building the prototype takes too long. After fixing anything that might be wrong with the project, we'll have to make sure to tidy up any documentation that comes with the class. Not only work on the documentation, but the presentation as well. We'll have another week to prepare for the presentation and practice for it, while the last week of summer is when the final presentation takes place.

<u>Artificial Intelligence Model:</u>

In order to detect the user's face, we will be using an artificial intelligence algorithm. The most effective way of doing so will be using Python. Python has many useful libraries for facial recognition, however, OpenCV seems to accomplish the task required. Ideally we will have the model train itself on each new user's face within the app that will be integrated with the lock box, and thus there will be a separate model for each user' face. Once the user goes to pick up their item from the lock box, the model that was trained on their face will be selected to perform its classification on the face that is within the camera's bounds, unlocking the box if there is a match.

AI Specifications:

Sp1	There must be a minimum of 500 data points per user for training
Sp2	The model must be able to identify a user within 80% accuracy
Sp3	The model must generate a model file that is retrievable for future evaluation
Sp4	The model must be able to train on photos received from the camera
Sp5	The model must train itself on the data points within 5 seconds

Table 6: AI Specifications

These specifications are subject to change in the future while research is being done. Another thing to note about AI is that the model can only be as good as the data, meaning that the quality of each photo will affect the accuracy of the model. Depending on whether we use infrared cameras or a digital camera to take photos of the user will change how the model is developed as well. It is also possible that 500 photos of the user will not be enough to have an accurate model, or perhaps more than enough in which we could reduce the amount of photos taken to save memory. However, the ability to have a model saved for the future will be required as well as taking photos from the camera (even if the photos are a matrix of dot projections from an infrared camera). The specifications that describe integration with the other mechanisms of the project will be discussed in the future once more research has been done. In terms of cost, the research and development of the model will be free.

Application:

The application will be on the user's phone and it is used to train the model on the user's face and also facilitate the interaction with the hardware to ensure the user is the correct user. This entirely depends on whether we can integrate python within the app. The app will be written with React Native so it will be compatible with both iPhone and Android products, without having to write the app in two different languages. The app will have to be able to communicate with the hardware through bluetooth. The data that will be sent will likely just be data about the user and what they ordered. And as mentioned in the previous section, if the model is trained within the app, a reusable model should be transferred as well. Ideally the user's app will store their own model within their device in order to save space on the lockbox's Raspberry Pi, and the React Native app will retrieve the model from memory and pass it to the Pi in order to evaluate the user's face. Otherwise, all models will be trained and stored within the Raspberry Pi. Cost for this could vary depending on how we store data. If we use the AWS servers to store data, the cost is written above, however, everything could potentially be stored locally, allowing for free storage and management as well as a simpler and more secure program.

Security:

Security will be a priority with this program as we will be dealing with user's faces and identification. If the only mode of transferring data is through bluetooth, there is minimal opportunity for interception. The issue will be ensuring that the bluetooth module connects to the correct device, and cannot be connected to without the app. If we are to use an AWS server, we'd have to connect the devices to Wi-Fi which is much more vulnerable to interception unfortunately. We will have to use techniques to encrypt the data that is stored on the Raspberry Pi at that point.

Project: Facial Recognition Locker
Revision: 1
Date: 2-18-2022

Correlations	
Positive	+
Negative	−
No Correlation	

Relationships	
Strong	●
Moderate	○
Weak	▽

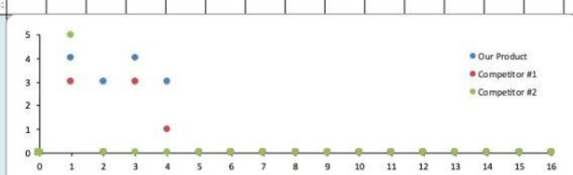
Direction of Improvement	
Maximize	▲
Target	◇
Minimize	▼

− +
+

		Column #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Customer Competitive Assesment						
		Direction of Improvement	◇	▲	▲	▼																			
Category	Weight	Engineering Requirements / Customer Requirements (Explicit and Implicit)	Physically Secure	Camera Resolution for Facial F	Data Security	Bluetooth Range													Our Product	Competitor #1: *CleverMade S100 Lo*	Competitor #2: *BoxLock*	Competitor #3:	Competitor #4:	0 1 2 3 4 5	Row #
Food Safety	8	Food temperature (kept at 40-60 C)	▽	▽	▽	▽													3	0	0				1
	3	Humidity (less than 65%)	○	▽	▽	▽													4	2	1				2
User Experience	7	Must recognize user in less than five seconds	○	●	○	○													4	3	2				3
																									4
																									5
																									6
																									7
																									8
																									9
																									10
																									11
																									12
																									13
																									14
																									15
																									16

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Target	ANSI Grade 3	100px across a face (>720p)	End-to-end encryption via Bluetooth, internal enccryption on Raspberry	10 m range max.												
Max Relationship	3	9	3	3												
Technical Importance Rating	700	1100	500	500	#REF!											
Relative Weight	58%	69%	50%	50%	#REF!											
Weight Chart					#REF!											
Our Product	4	3	4	3												
Competitor #1: *CleverMade S100 Lockbox*	3	0	3	1												
Competitor #2: *BoxLock*	5	0	0	0												
Competitor #3:																
Competitor #4:																

0 1 2 3 4 5



Column #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16