# IntelliRack

Authors: Ryan Lin, Doreen Valmyr, Surafel Tsadik
Affiliation: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—This project introduces an innovative solution for streamlining the personal item check-in process at large events and conventions. By integrating facial recognition with a hardware stand, users can easily and seamlessly check-in and retrieve their items. This approach can reduce process time compared to the traditional manual process.**

***Index Terms*—Actuators, Arduino, Sensors, Facial Recognition, Hardware-Software Interface, OpenCV, User Interaction**

## 1 INTRODUCTION

At various events and conventions worldwide, attendees often need to check in their belongings temporarily. Whether for security purposes or simply to organize the event space, this process typically involves interaction with a personal item checking attendant. Attendees exchange their items for an identifying claim ticket, after which the attendant stores the items in a designated backroom area.

There are several problems that can arise. For example, an attendee can lose their identifying ticket. This usually results in a long and irritating game of the attendee describing various features of their personal item while the attendant scrambles to identify it. With no clear identification, the attendee can also steal an item that may not belong to them. A more general problem is the sheer amount of time this process takes. Attendants, given a ticket, have to search through possibly hundreds to thousands of personal items, resulting in long queue times and frustration among attendees.

This project aims to address these challenges by implementing an automated system for personal item retrieval, utilizing facial recognition as a secure and reliable identifier. Such an approach has the potential to significantly streamline the personal item checking process, enhancing security and efficiency while ensuring attendees' belongings are safely managed.

Currently, there seems to be one notable competing product called CoatChex. CoatChex eliminates the need for a claim ticket by providing a kiosk where attendees can sign in using their phones. Our approach provides more functionality and is easier to use. Firstly, CoatChex still relies on numbering a personal item and manually placing them on a rack. It still relies on an attendant to search for an item, which could slow down the process as described earlier. Secondly, CoatChex requires the user to sign in on their phones which takes some time, while our project seamlessly uses facial recognition to automatically recognize a user and their personal item by simply walking up to the system.

## 2 USE-CASE REQUIREMENTS

### 2.1 Facial Recognition

To ensure that users can check in their items using their face, our solution must include an accurate face detection and recognition system. For facial recognition, our system must be able to recognize users from at most 0.5 meters away within 5 seconds. Not only does this create a smooth check-in process for users, but makes sure that bystanders walking around in the background don't accidentally start the check-in process. For face recognition, we aim to have an accuracy rate of at least 90%, erring on the side of false negatives. Correctly recognizing a person's face with high accuracy is crucial so that we can give users their correct item, but we also want to avoid giving users someone else's item.

### 2.2 Item Deposit/Retrieval

We aim to have our system detect when an item has been placed or removed from the rack within 1 second. We plan to have the physical rack be able to rotate such that the next open rack position faces forward toward the next user. To make sure that the next user in line doesn't have to wait long to check in their item, we want our rack to quickly detect when an item has been placed onto/removed from the rack so it can start (and eventually finish) the rotation process quickly.

### 2.3 Weight

We want to allow users to place heavier items like their backpacks onto our rack with no issue. This is why we aim to have our rack be able to withstand weights of up to 25 lbs on each hook without slowing down the rotation of the rack. Since our rack will have 6 hooks to hang items onto, this means that the rack has to be able to hold a load of at most 150 lbs total without any loss to the stability of the rack. There is also the possibility of there being weight imbalances on the rack depending on what gets placed on/removed from the rack. We will make sure that our system implements a way to deal with these imbalances by determining the next best rack position with item weights in mind.

# 3 ARCHITECTURE AND PRINCIPLE OF OPERATION

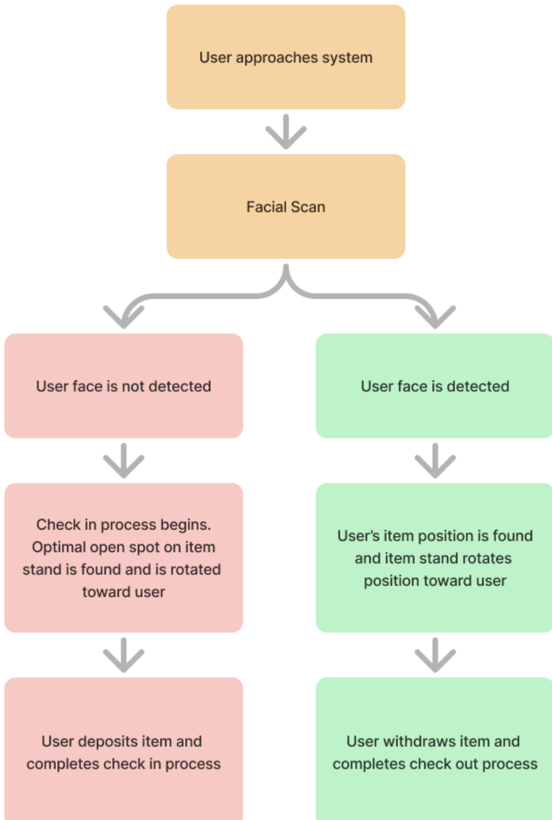


Figure 1: System Flow Chart

The user flow diagram is illustrated in Figure 1 and the system block diagram is shown in Appendix Figure 5. A user initiates interaction by approaching the camera and positioning within a distance of half a meter. Subsequently, upon user authentication, the check-in process commences. In the system's back-end, available open slot positions on the coat stand are assessed, and an optimal location for the user's item placement is determined. The rack then rotates accordingly, facilitating item placement by the user. Once completed, the user can proceed to their event activities. The check out process involves a similar procedure, where the user repeats the initial steps but removes their item from the rack instead.

## 3.1 Facial Recognition

When the user approaches our system, they will scan their face to begin the facial recognition process. When their face is detected for a sufficient amount of time, our system will take about 10 frames of their face and obtain their face landmarks for each image. With the facial landmarks in each image, we will check our recognition model to see if we have seen the face before. If we have then our system will recognize the face and communicate to the rack which item belongs to the user. If the user is checking in then our system won't recognize the face, which is when their face information will be used to train a new recognition model that includes the new user's face. They will then be prompted to put their item on the rack.

## 3.2 Item Stand

The rotation of the rack is made possible though the NEMA Stepper motor, which will slowly rotate the rack until stopping at an empty location where the user can place their item. Once an item is placed, the readings from the load cell are passed through an ADC and processed by an Arduino Mega. These read weights will be stored in a MySQL database, and used to later compute the optimal locations for other users to place their items.

# 4 DESIGN REQUIREMENTS

We have some design requirements related to each of our subsystems that ensure ease of use and a reduction of frustration for all potential customers - event attendees and the event managers.

## 4.1 Facial Recognition

The first design requirement pertains to our facial recognition system. Facial recognition is one of the key pillars of making the experience of interacting with our system easy and worry free. Our design goal for the facial recognition system is a 95% facial recognition accuracy rate. The aim for this high accuracy rate because we map faces to people's personal items and keeping attendees' items secure is very important.

While we want to have high accuracy, speed is very important as well. Our second design requirement is that a user's face should be recognized within 5 seconds. If a facial recognition system takes too long to recognize your face, a user can get frustrated. Extra time can also slow down the overall check-in process if dozens to hundreds of people have to check in their items. We choose 5 seconds because it is a relatively short time, was used by past capstone projects, and gives us flexibility with improving accuracy.

Lastly, the two previous design requirements should apply only to users that have come within 0.5 meters of the camera- outside of that nothing should happen. The system is designed for a fast and seamless user experience, but has to determine who is actively trying to use the system. Because the user has to place their item on the physical stand, we choose 0.5 meters as a reasonable distance a user would stand to check in their personal items.

## 4.2 Hardware Item Stand

The second set of design requirements revolve around the hardware item stand, which still store physical user items and interface with the facial recognition system for item retrieval.

The first design requirement is the quick detection items added to or removed from the system- within 1 second. This design requirement coincides with the general idea of reducing frustration for the user. The quick detection of physical changes can allow the rest of our system to know if it can move on to supporting the next person. 1 second is rather arbitrary, but is a very short time and entirely doable with our hardware.

The second design requirement is the quick display of users' items within 1 second, after the user's face has been recognized. This requirement, along with the relatively fast facial recognition requirement, allows the user to retrieve their personal items as fast as possible and allows for the steady flow of users checking in or checking out their items.

We want to support a larger variety of items other than traditional coats, resulting in the last design requirement necessitating our item stand to withstand 150 pounds of weight. In contrast to coats, items such as backpacks can weigh up to or even exceeding 20 pounds- the reason we chose a 25 pound support goal on each of our 6 hooks. We want cater to hardware integrity by supporting a large weight.

Overall, our design requirements focus on making the software and hardware optimized enough to serve the customer, without any frustration or difficulty.

# 5 DESIGN TRADE STUDIES

There were several times the team made specific choices to narrow down the design. For example, the choice of a facial recognition library and its trade offs, or what kind of microcontroller or computer to use to read and control our sensors and actuators.

## 5.1 Facial Recognition

The first main component is our facial recognition system that will check user in and out. It is crucial that whatever detection and recognition model we used is not only fast enough, but accurate enough to meet our design requirements. We were able to narrow down our options to two toolkits, OpenCV or Dlib. After doing initial testing with manually inputted images as training data, both OpenCV and Dlib had their pros and cons. With Dlib, it had access to a shape predictor with 68 face landmarks, which would provide more accurate face alignment for each image processed (leading to more accurate and consistent recognition). Where Dlib falls short is its inability to easily implement real-time facial recognition through a video stream (which is what we are mainly looking for). For OpenCV, although their shape predictor doesn't have as many landmarks, OpenCV comes with built-in functions to easily run a video stream and start detecting/recognizing from them. Since this test also gave similar result for OpenCV as it did with Dlib, we decided to go with OpenCV for our toolkit of choice.

OpenCV itself has two different ways of implementing face detection. One implementation is based on the Haar Cascade Algorithm, while the other implementation is a DNN face detector that is based on Single-Shot MultiBox (SSM) with the ResNet-10 architecture as the backbone. It was clear after a bit of research that the Haar Cascade face detection was very outdated and produced a lot of false positives (which we want to reduce as much as possible). Not only is the SSM face detection much more accurate overall, it has a greatly reduced false positive rate compared to Haar implementation and works well in real-time. So we chose to go with the SSM face detection. For the face landmarks and recognition, the clear choice for us to use is the recognizer created by the OpenFace toolkit. As the name suggests, OpenFace is a toolkit built on top of OpenCV that allows for facial landmark detection. This is the first of its kind, and it is perfect for what we need based on our requirements.

## 5.2 Hardware Item Stand

The second main subsystem is our hardware stand. Because hardware is a substantial part of our project, a lot of potential approaches and ideas were discarded and revised.

### 5.2.1 Item Stand Design

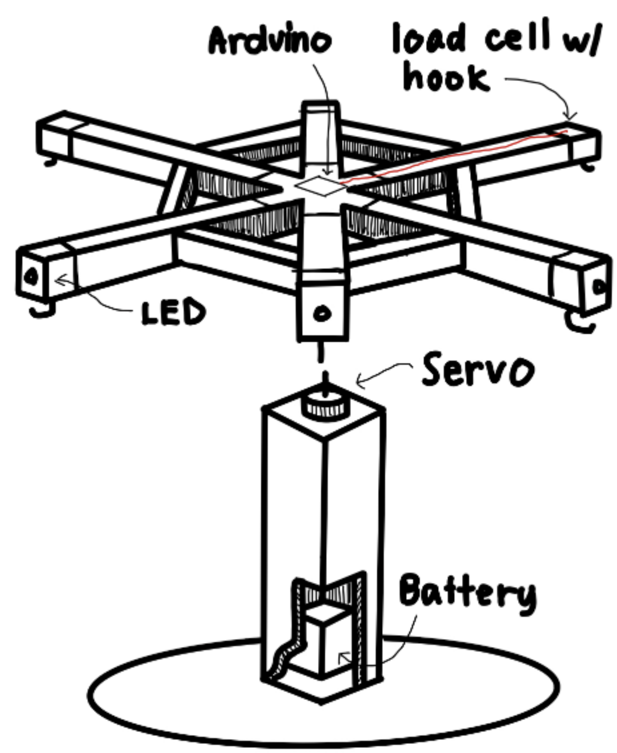


Figure 2: Original item stand design from the proposal presentation

The original item stand design is shown in Figure 2. It features a large, circular base, a smaller and thinner rectangular prism for the main shaft, and a large, hexagonal-shaped upper rack with 6 prongs that house hooks to store user items. The upper rack contained load cells, LEDs, hooks, and an Arduino to process sensor data and control the servo. A large battery in the base of the stand provided weight stability and portable power. After a qualitative assessment of the design and feedback from peers, we made shifted the design to abandon some original components.

The most glaring design flaw was the single axle support of the stand. With the possibility of 150 pounds on the upper rack in accordance with our design specifications, a single axle combined with the fact of no guarantee of even weight distribution could have been disastrous. Secondly, because there are no other connection points, there is no way to control the servo with the Arduino that is on the upper rack. The same reason applies to the inability of the battery at the base of the item stand to power the upper rack.

The second design flaw was the size of hexagonal upper rack. Due to the 6 upper prongs being much longer than the width of the mid rack, weight imbalance and tipping can happen easily, even if the single axle described earlier was reinforced. This ratio was originally chosen to allow for generous clearance of items that are hooked- up to 10 inches. For example, an very thick item could be hung on a hook without it brushing up against the middle main structure of the stand during rotation. But, after examining several common items brought to events, such as backpacks, purses, bags, and coats, we determined that this clearance does not have to be nearly has large. Just 3-4 inches is enough.

#### 5.2.2 Servo Versus Stepper Motor

The project required various components to support the design requirements. For the item stand to rotate upon the user check-in or check-out, we required an actuator in the form of a servo or a stepper motor. Originally, the team planned on acquiring and using a large servo, as shown in Figure 2, but there were some problems with this. Firstly, most servos can only rotate 180 degrees, which would not work for our use case of rotate the item stand from any position to face the user. Secondly, a servo, due to its precision angle-setting, is not as robust as a stepper motor. This is not ideal specifically for our project where the actuator has to potentially rotate 150 pounds. Lastly, a servo costs more than a stepper motor. Though the rotation of the rack is very important, the purchase of a component that could be more than $100 along with the other reasons mentioned resulted in our team choosing a NEMA 17 stepper motor. Peer feedback from the proposal presentation reinforced our choice. A servo was originally involved in the design because our project required the precise rotation and a servo would allow us to achieve that relatively easily.

#### 5.2.3 Load Cell and Load Sensors

Next, to detect item deposit or withdraw, some kind of weight measuring mechanism must be used. There were 2 sensors that could achieve this goal- load cells or load sensors. Load cells are more suited to measure tensile force because it measures different resistances on different sides of a slightly bendable bar. Load sensors measure compression weight, meaning they are the kinds of sensors used in a traditional floor scales. The load sensors could support the measurement of much more weight, while being cheaper. Our team devised various ways of using the compression load sensors to measure tensile weight. For example, if the hook was connected to a ring which was placed on each of the six prongs in Figure 2, then a downward force would compress a potential load sensor placed on top of the prong. Eventually, a load cell was chosen after balancing the extra layer of hardware complexity against the slightly higher cost.

#### 5.2.4 Microcontroller and Computer

With these component choices, we needed to choose a processing unit to read and control them. Two potential choices were gathered- an Arduino Mega, or an Raspberry Pi (or any other small computer). Using a Raspberry Pi could change the solution approach drastically, because a facial recognition model could be built directly into the rack instead of running the model on a personal computer and wirelessly transmitting to an Arduino. With this approach, the system could be entirely self-contained within the rack. This solution route is a viable option, but was abandoned due to time constraints of the project, lack of experience with Raspberry Pi, and being over-engineered for our solution approach. This means that the extra complexity will not bring any new features to our predetermined design requirements or use case. An Arduino is a easier component to use to control and read from our other components.

#### 5.2.5 Power

Our original design, as shown in Figure 2, featured a battery to power the a servo and Arduino, allowing portability. But further deliberation and evaluation of our use case resulted in the team abandoning this component. The target customers of event organizers largely have no need of a portable item stand. Firstly, most events have ample power outlet or power transmission infrastructure. Secondly, if the item stand was populated with items up to 150 pounds, it would not be portable anyways. Lastly, a battery as the only source of power meant that the item stand could only be used for a certain period of time, compared to indefinitely when plugged into an outlet.

## 6 SYSTEM IMPLEMENTATION

While our product solution is not primarily aimed at enhancing public health, safety, or welfare, there are aspects of our project that contribute positively to this domain. For instance, eliminating the need for a ticket system and simplifying the process of checking in personal items can alleviate stress for individuals attending events. Moreover, our product offers customers peace of mind by ensuring that their items are securely tracked and organized on our item stand.

Furthermore, the adoption of our system by organizations can lead to significant cost savings by reducing the need for event staffing. Additionally, our product eliminates the requirement for external identifiers such as tickets, further lowering event costs for organizations or event creators.

Our facial recognition algorithm will use tools like OpenCV to characterize and distinguish different faces, so things involving a person's social group, whether cultural, political, or economic, will not be taken into consideration when checking in or checking out a user. In addition to this, our product can be used in a variety of social contexts and within all types of social gatherings that necessitate a attendee having to set aside particular personal items when attending that particular event. Without strict rules being set on the items that can be on the rack, simply the assumption that it will mostly be used for coats, users will not be limited to the items they can set aside, but mostly the weights of those items. As a result, users are able use our product for their required social gatherings without limitations.

## 6.1 Software Implementation

The current software component of our project is split up into two parts, the facial recognition system and the web application. For the facial recognition system, we will be using various toolkits/libraries to implement facial detection, face landmark detection, and facial recognition. The main toolkit that will be providing all that we need is OpenCV and OpenFace (a toolkit built on top of OpenCV). We will be using the DNN face detector that is included in OpenCV version 3.3+, which is implemented using Single-Shot Multibox. Additionally, we will be using the face recognizer that is provided with OpenFace, which will give us their facial landmarks for face alignment. With these two technologies, we will be able to take in user input (their face being scanned in), obtain their face landmarks (while doing some face alignment), and represent the user's face as a vector that we can use to compare to other faces in the system.

For the web application, we will be using Django as the web framework of choice as it is a framework that uses Python as the back-end language of choice (we have the most experience with Python). Other than the Django framework, we will be using the vanilla web stack (HTML, CSS, JavaScript) for the actual construction of the web application as something more complex isn't needed for the requirements of our project. To serve as our database for both the user's items and the facial embeddings that will be created during the facial recognition phase, we will use MySQL because of its high performance and that it comes built into MySQL. The purpose of the web application is to mainly log and display the current items on the rack and where they are located, but also allow possible staff of an event to manually change the state of the rack at any time. The web application should be able to manually check in/out items at any time and pinpoint the correct position of an item on the rack regardless of where it may have rotated to, so the web application needs to communicate with the rack so it knows when the rack rotates and for how long. This communication will be created using a wireless transceiver connected to a laptop through an Arduino (more on this later in Section 6.2.3).

## 6.2 Hardware Implementation

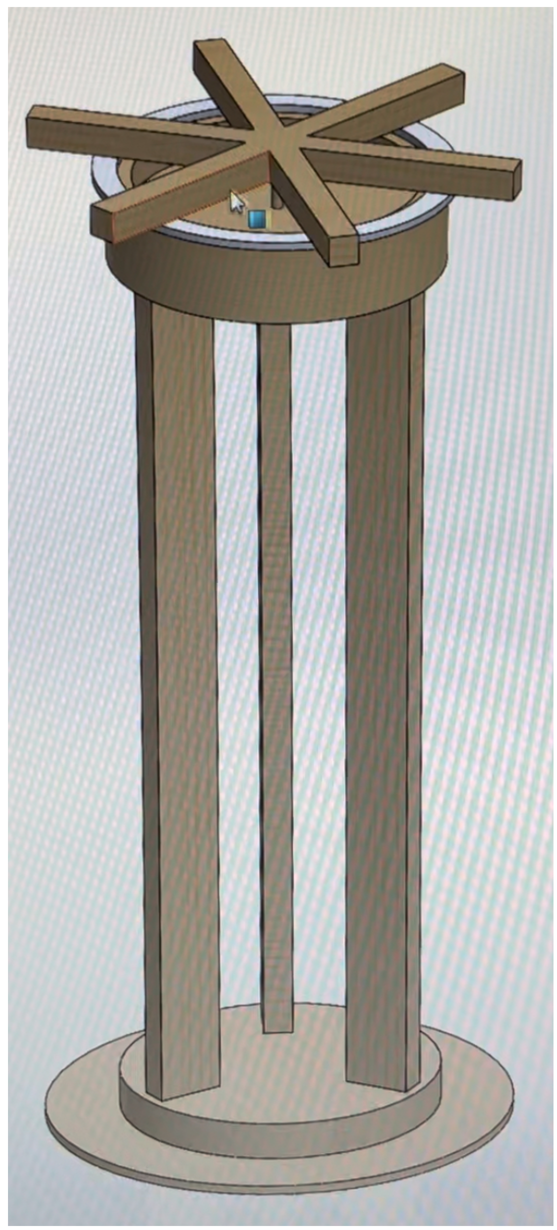

Figure 3: 3D model of current item stand design

#### 6.2.1 Design

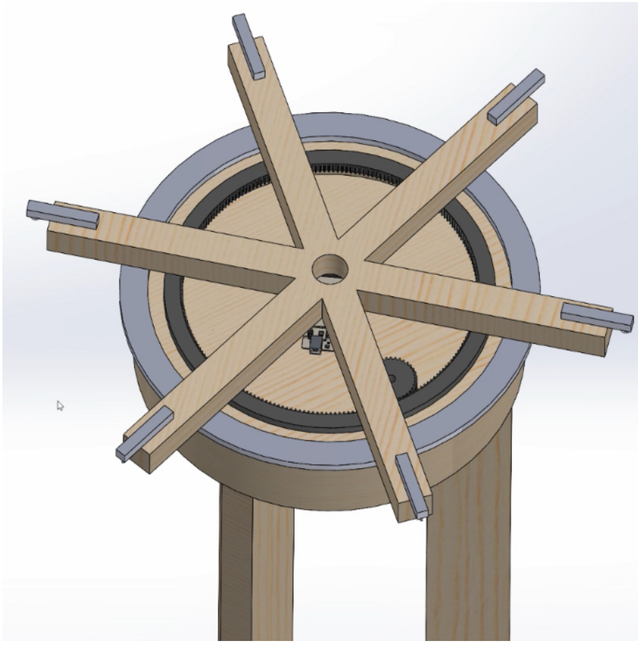

Figure 4: Top view of item stand

The current hardware design resembles Figure 2, featuring a hexagonal, 6-sided rotating prong structure as the upper rack. This rotating component will be mounted on a turntable bearing, facilitating both support for the items placed and rotational movement to display specific positions for user item placement. Positioned at the endpoints of the hexagonal structure will be LEDs and hooks. LEDs will serve to guide users to the designated item placement areas, while hooks will enable users to securely hang their items on the rack. Additionally, an internal gear mechanism will be integrated with this structure to facilitate smooth rotation.

Below the top hexagonal structure is a circular area containing the stepper motor and Arduino Mega, along with an internal gear connected to the upper rack as shown in Figure 4. The internal gear, with a diameter of 11 inches, will be connected to a smaller 2-inch diameter spur gear. This smaller spur gear will then be connected to the stepper motor, serving as the mechanism to provide rotational power for the rack.

The bottom of this circular area is then connected to 4 planks (only 3 legs are displayed in the CAD model) and a large wooden base which will provide stability for the rack in the case of uneven weight distributions. The entire rack stands at about 4.5 feet tall, allowing users to easily place their items while standing upright.

#### 6.2.2 Electronics

The item stand incorporates several electronic components, including an Arduino Mega, six load cells, six HX711 ADC amplifiers, a NEMA 17 stepper motor, LEDs, and an A4988 motor driver. Due to the small analog signals produced by the load cells, each load cell is paired with an HX711 ADC. The wires from each ADC will be routed through a central hole in the rack to connect with the Arduino. We are considering the use of a slip ring if the rack rotation exceeds 360 degrees from the reference point. Additionally, the Arduino controls the NEMA 17 stepper motor, with power provided by the A4988 motor driver. Both the Arduino Mega and motor driver will be powered by a wall adapter.

Since load cells output analog values, their readings must be calibrated using known weights. We intend to calibrate each load cell during our component testing phase to ensure accurate measurements. Additionally, because all six connections to the HX711, which interface with the load cells, require power and ground lines, we plan to consolidate these connections. We also consider joining the clock lines if we opt to implement a slip ring mechanism.

#### 6.2.3 Connection to Software

The item stand does not contain any user processing logic, which is done exclusively on a personal laptop. In our system, the Arduino Mega is attached to an NRF24L01 wireless transceiver, which will communicate with an Arduino Uno with the same component connected by serial port to a personal laptop. The laptop will be connected to a camera and runs the facial recognition model along with other logic to ensure items are mapped to faces and that the rack is balanced (balanced rack algorithm). Once a user is has either has their face scanned, the software will determine where the a user can deposit or withdraw their item. Then, the software component will write to the Arduino Uno which will transmit rotation and LED data to the rack.

The rack can then read if an item has been placed or removed via load cell reading, and transmit this information the same way to the software component.

# 7 TEST & VALIDATION

To verify the integrity and robustness of our system, we will conduct unit and functional tests. This includes thorough testing of both the facial recognition software and the physical coat stand for integrity and sturdiness.

## 7.1 Facial Detection and Recognition

We envision our autonomous coat rack system being utilized in large events characterized by high volumes of attendees. Our system should detect passing faces without attempting facial recognition, as these individuals are unlikely to be checking in their personal items. We aim to recognize only users within a distance of 0.5 meters or less from our camera. If the distance exceeds 0.5 meters, our system should still detect them, indicated by a bounding box surrounding their face in the camera feed, but refrain from trying to determine whether the person has already checked in an item or is currently attempting to do so.

For testing purposes, we define three distances between potential system users and our camera: 0.5m, 1m, and 2m. At a distance of half a meter, our system should detect that a person is in the vicinity of our camera and within the specified range distance ([0m, 5m]), indicating that a user is likely attempting to check in or check out a personal item. At distances of 1m and 2m, our system should be able to detect a person's face without attempting to check them in or out, as they are likely to be passersby. To conduct this experiment, we will gather 20 volunteers. For each volunteer, we will ask them to stand at all three specified distances ([0.5m, 1m, 2m]), and we will examine our system's log to determine whether they have been successfully checked in when standing at a distance of 0.5m for that trial, or if they have been successfully detected but not checked in whether standing at the 1 meter and 2 meter marks.

Additionally, our facial recognition system should achieve a 95% accuracy rate and be capable of correctly identifying a user within 5 seconds. These requirements ensure that the process of checking in and out personal items is efficient and precise, contributing to user satisfaction with our product. As mentioned earlier, we will enlist 20 volunteers to validate our recognition system. When a user stands within 0.5 meters of the camera for at least 3 seconds for the first time, they should be checked in within 5 seconds. On a subsequent instance of standing within 0.5 meters of the camera for at least 3 seconds, the user should be identified, initiating the check-out process.

By considering the distinctions between facial detection and recognition, defining specific criteria for user recognition based on distance, and setting clear performance benchmarks for our facial recognition system, we aim to ensure the effectiveness, reliability, and user satisfaction of the facial recognition portion of our autonomous coat rack system.

### 7.2 Item Placement/Removal & Item Position

Upon recognizing a user, we anticipate that the user moves towards the coat rack to either deposit or retrieve their personal item. It is crucial that we accurately discern weight changes on the rack to determine the availability of storage space for additional users. Therefore, we aim to promptly ascertain, within 1 second, when a user picks up or places down their item. This enables us to promptly update our program's data and accommodate the needs of subsequent users who are likely waiting in line.

To test this, we will repeatedly place and remove items on hooks 20 times and monitor whether the system successfully identifies when an item is placed or removed. This can be accomplished by displaying an alert message on a terminal and observing the speed at which these alerts appear after adjusting the weights on the hooks of our rack. If we receive the messages within 1 second, we are within our specified latency range. However, if the messages are not received within 1 second, we will adjust our weight threshold parameters to enhance precision in detecting weight fluctuations.

In a similar regard, we aim to display items or available slot positions within 1 second, achieving this goal 90% of the time. This implies that our rack should commence rotating within 1 second of identifying the user, facilitating their retrieval of personal items during check-out or placement during check-in. This ensures minimal waiting time for users interacting with our system, while also allowing for error correction in cases where potential errors may occur. This test will also be conducted 20 times to ensure the robustness of our algorithm.

### 7.3 Rack Integrity

The rack component of this project consists of six hooks, evenly distributed around the center of the stand. Each hook is designed to withstand weights of up to 25 pounds individually, with a maximum load capacity of 150 pounds for the entire rack. To validate this, we will conduct comprehensive testing involving the placement of varying weights on each hook, ranging from 0.5 lbs to 25 lbs. This extensive testing regimen ensures that each hook can reliably support items of different sizes and weights. Additionally, we will assess the rack's performance under conditions of weight imbalance, including tests with maximum load and rotations, as well as one-sided imbalance tests with a load of 75 pounds.

Ensuring the durability and stability of the rack is paramount to its functionality. By subjecting it to rigorous testing, including weight distribution and imbalance scenarios, we aim to confirm its ability to withstand real-world usage conditions. This testing approach not only validates the rack's capacity to securely hold items but also enhances user confidence in its reliability. Furthermore, by identifying potential weaknesses and areas for improvement during testing, we can refine the design and optimize the rack's performance for optimal user experience.

## 8 PROJECT MANAGEMENT

### 8.1 Schedule

The Gantt Chart provided in Appendix Figure 6 delineates the project timeline for the semester. It features three distinct rows, each attributed to a team member, detailing their individual responsibilities. The schedule incorporates designated break times and slack periods, offering flexibility to address any unforeseen issues or to allocate additional time for specific components as needed.

### 8.2 Team Member Responsibilities

Surafel will lead the facial recognition aspect of the project. Additionally, he will be responsible for creating the UI component, which will display logs related to users who have items currently checked in on the rack.

Ryan and Doreen will take the lead on the hardware components, which involve building the physical rack and integrating electrical components such as load cells and an Arduino Mega. This responsibility also includes writing the associated code to program these components and specifying algorithms for how the rack should behave when interacting with users.

The integration of these sub-systems, along with the preparation of presentations and reports, will be a collaborative effort among the team. Additionally, team members will provide support to one another in resolving any issues that may arise across various project components.

### 8.3 Bill of Materials and Budget

The bill of materials can be found in Appendix Table 1, detailing the materials acquired thus far in the semester and those scheduled for purchase later. These materials are essential for the successful execution of our project.

Additional materials, including wires, breadboard, and small circuit components, were repurposed from previous class projects.

Overall, the projected cost for our materials is $354.27.

### 8.4 Risk Mitigation Plans

The main aspects of this project involve the facial recognition sub-system and the physical hardware rack. The main risk associated with these components are accurate facial recognition and robust rotation. In order to satisfy users and meet use case requirements, users should be correctly matched and the rack should safely rotate to optimal hook positions, ensuring that our system is both usable and safe for potential users.

In the event that our facial recognition system does not meet the specified requirements, we will explore alternative libraries beyond OpenCV, such as dlib and MediaPipe. Dlib utilizes a more sophisticated approach for face detection through Histogram of Oriented Gradients (HOG) features and a linear Support Vector Machine (SVM) classifier, while MediaPipe allows developers to incorporate facial recognition models using TensorFlow. By leveraging these diverse methods for facial recognition, we can determine which approach best satisfies our requirements if OpenCV falls short of initial expectations.

To mitigate the potential failure of the coat rack, we will restrict the types of items that users can place on the rack. Only lightweight items such as jackets will be permitted, while heavy items like backpacks will be prohibited. This measure ensures that the items are not excessively heavy, thereby facilitating smoother rotation of the rack.

Furthermore, if users attempt to place items on the rack that are deemed too heavy, we will activate a red LED to alert the user. These LEDs will be strategically positioned in close proximity to their respective hooks on the rack, allowing users to easily identify any errors.

## 9 RELATED WORK

To initiate our preliminary research, we examined a project completed by a team in Spring 2021, which developed Smart Wardrobe [3]. This project integrated a rotating stand and a clothing recognition model to suggest outfits to users. Drawing inspiration from this project, we designed our rotating rack system, enhancing it with additional components and specifications such as load cells to measure the weight of items on our rack and LEDs to notify users where to place their items.

A team from Spring 2019 [1] developed facial recognition algorithms for their project and we took inspiration from their timing goals and application of facial recognition in every day life.

## 10 SUMMARY

IntelliRack aims to revolutionize the coat check system at large events by addressing key pain points experienced by attendees and organizers.Traditional methods involve handing over personal items to attendants in exchange for a ticket, leading to inefficiencies such as long wait times and the risk of losing tickets. Our solution leverages facial recognition technology and a hardware stand for seamless item retrieval, eliminating the need for attendants and streamlining the check-in and retrieval process.

Stakeholders concerned with event efficiency and attendee experience will benefit greatly from our design. Attendees will experience faster check-in and retrieval times, enhancing their overall event experience. Organizers will see improved operational efficiency and reduced staffing needs, leading to cost savings and smoother event logistics.

While our solution offers promising benefits, several challenges must be addressed during implementation. These include ensuring the accuracy and reliability of the facial recognition system and seamlessly building and assembling the hardware stand, while also ensuring its sturdiness and safety during rotation. Successfully navigating these challenges will be essential in delivering a usable and satisfactory product for event attendees.

## References


[1] Team A3. *Door ID*. 2019. URL: `https://course.ece.cmu.edu/~ece500/projects/f19-teama3/`.

[2] GreatScott! *Electronic Basics 33: Strain Gauge/Load Cell and how to use them to measure weight*. YouTube. 2017. URL: `https://www.youtube.com/watch?v=lWFiKMSB_4M&t=390s`.

[3] Fred Lee, Henry Lin, and Sung Hyun Back. *Team B2: Smart-Wardrobe*. 2021. URL: `https://course.ece.cmu.edu/~ece500/projects/s21-teamb2/`.

# APPENDIX

Table 1: Bill of materials

| Description | Model # | Manufacturer | Quantity | Unit Cost | Total |
|---|---|---|---|---|---|
| Arduino Mega 2560 Rev3 | Arduino Mega 2560 Rev3 | Arduino | 1 | $48.40 | $48.40 |
| 10kg Load Cell + Amplifier 2 piece set | S22X2+S19X2 | ShangHJ | 4 | $11.99 | $47.96 |
| Dailydanny Heavy Duty Lazy Susan | AluLS | Dailydanny | 1 | $35.99 | $35.99 |
| Nema 17 Stepper Motor | 17HS19-2004S | OSM Technology Co.,Ltd. | 2 | $13.99 | $27.98 |
| 4pcs NRF24L01+ Wireless Transceiver Module | 3-01-0416 | HiLetgo | 1 | $7.89 | $7.89 |
| 2x8 Feet 1/2 Inch Sande Plywood | - | - | 3 | $45.55 | $136.65 |
| 2x4 Inch, 8 Feet Prime Stud Wood | - | - | 3 | $3.25 | $9.75 |
| Stepper Motor Driver Carrier | A4988 | Pololu Corporation | 2 | $13.95 | $27.90 |
| Power Supply AC Adapter | 4336304932 | smooth-elec | 1 | $9.99 | $9.99 |
| Screw-In Hooks | - | BuyMagnets | 8 | $0.22 | $1.76 |
| | | | | Grand Total | $354.27 |

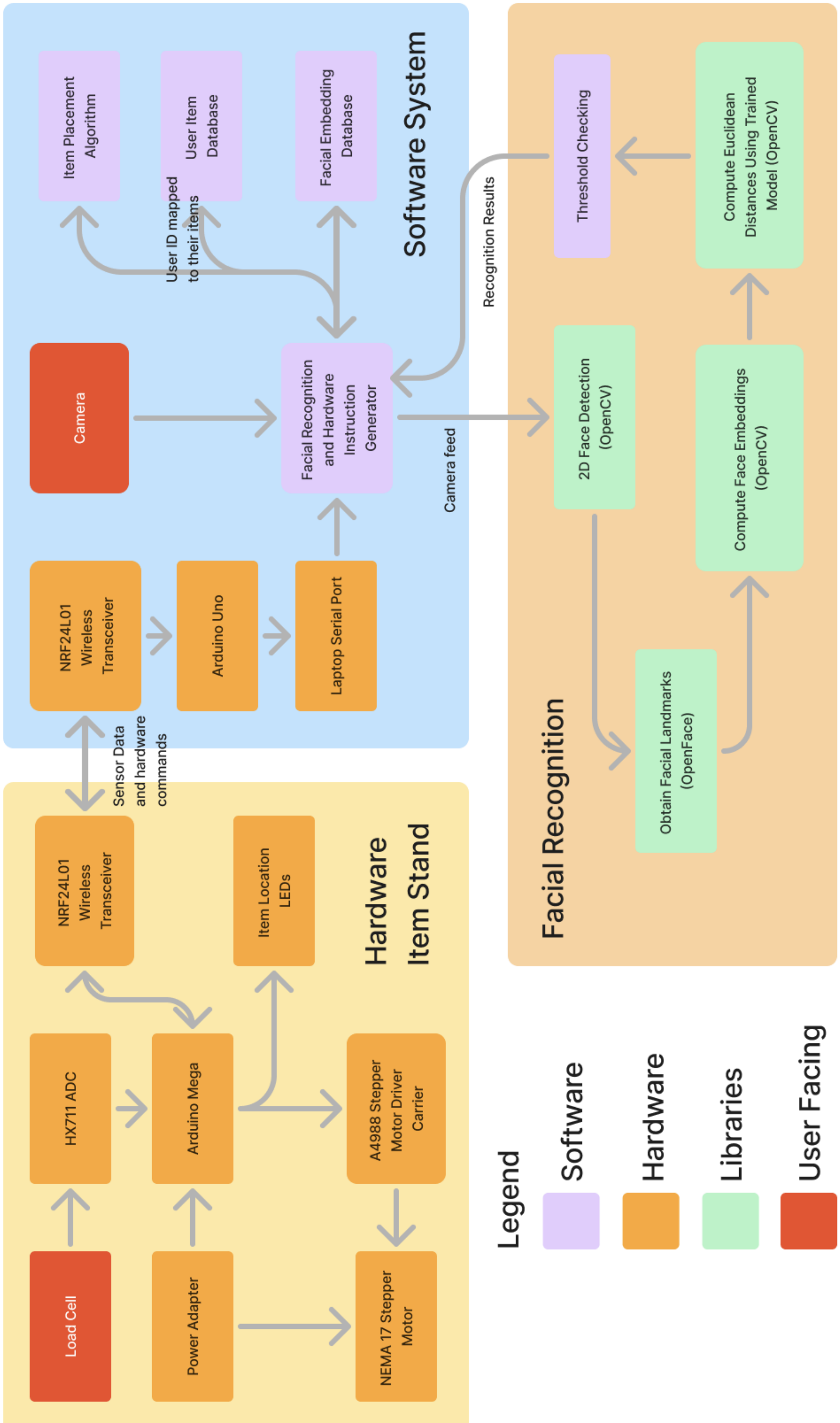


Figure 5: System Block Diagram

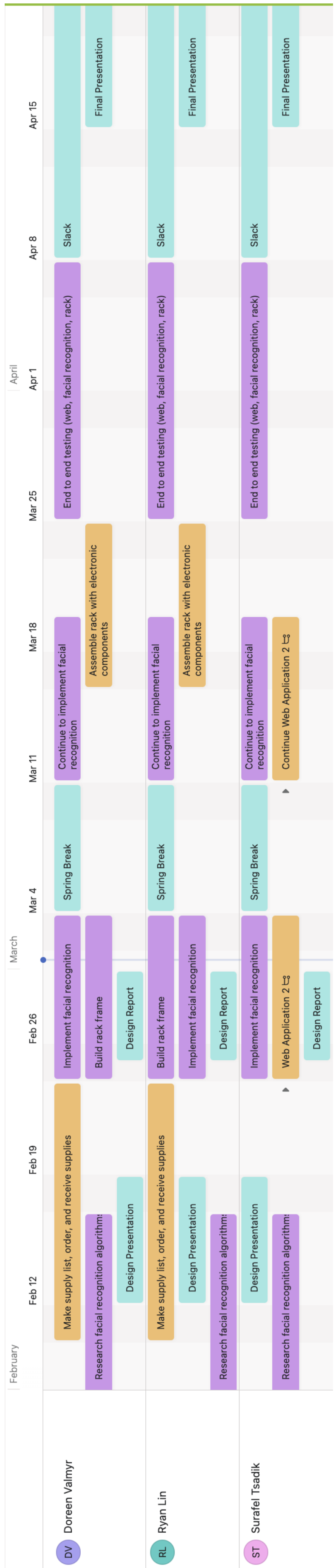


Figure 6: Gantt Chart