Portal Box Design Report

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## Project Manager: Axel Andrews

## Systems & Performance Engineer: James Powick

## Design Engineers: Peter Beaudette, Kevin Duong, Aidan Flynn, Colton Jiorle

# **0: Executive Summary**

This report serves to document the progress made during the first semester of the Senior Design project focused on the fourth generation of the Portal Box system. The Portal Boxes are a machine-access control solution for makerspaces, designed and deployed at Bucknell University in several makerspaces and fabrication labs. This project aims to build on an already unique and innovative product, enhancing usability, reliability, and adaptability while reducing costs, benefiting stakeholders, and broadening the scope of the deployments

The report outlines the team’s research and development towards the project’s objectives, design challenges, and proposed solutions. The team has six members – a Project Manager, a Systems and Performance Engineer, and four Design Engineers, and their respective responsibilities are laid out below in the later sections of the report. The existing system has three main components, a physical box that handles the user interaction, a database that stores all of the required information, and a website that allows users and admins to interact with machinery and view their permissions. The system tracks users using RFID cards which they insert into the physical box, which then pings the database to check whether they are authorized for a given machine. For machines that operate at 120 VAC, the box can turn power on and off between the wall and the box to control access, while higher power machines can be controlled with a USB interlock that blocks data transmission for unauthorized users.

Existing Portal Boxes are built using Raspberry Pi microcontrollers, and the new generation will be ESP32 based in an effort to cut cost and reduce complexity. The current microcontrollers are more powerful than required for the application, and have more functionality than is needed. The user interface on the physical boxes is currently streamlined, with just one button, LED lights, and a buzzer. We seek to implement a keypad and a screen to provide more configurability, information, and control at the point-of-use. These components will also enable two-factor authentication, eliminating existing security loopholes that could be a deterrent for new makerspaces and labs from deploying the system.

As will be explained in more detail throughout the report, these additional features seek to improve both the security and accessibility value added by Portal Boxes. Entities that provide machine access to users must balance their need to protect against misuse with the need to train and authenticate users. The existing Portal Boxes are a great solution to this problem, and thus our changes do not seek to disrupt the general operation of the system. Providing more information to users and admins will allow for simpler training, data analysis, and troubleshooting, while plugging any holes in the security functionality of the boxes will bolster confidence in their efficacy and reliability.

Looking ahead to next semester, our team plans to deliver a high technical-readiness-level project to our client, Matt Lamparter, as well as labs and spaces outside of Bucknell. The Portal Box projects of the past have laid a tremendous foundation for our project to build from, and we hope to honor all those who have developed and maintained the previous iterations by spreading the value they created into the maker community. Portal Boxes have an opportunity to greatly impact universities, lower-education, private makerspaces, and more through their innovative approach to machine access control, and we want to share the benefits that Bucknellians have reaped with a wider audience.

# **1: Project Context & Stakeholders**

Many people, from casual users to dedicated professionals, rely on machines to hone their craft efficiently. However, these machines are not always accessible or realistic for individuals to own, which is where makerspaces offer a more affordable solution [1]. While beneficial, these spaces have flaws, such as limited hours of operation, accessibility issues, and potential security concerns if open 24/7 without constant supervision. The issue of balancing security and accessibility effectively often results in a one-sided approach, which leads to outsiders being excluded while insiders roam free [2]. Trust plays a huge factor in how these spaces are set up and how much freedom is granted.

Many spaces with specialized equipment are built around the trust of different users to respect and uphold the guidelines of a communal space [3]. However, given the level of risk involved, it often becomes necessary to ensure users are trained and skilled at operating these machines for their own safety, that of the machine, and others in the space [4]. As more machines become feasible to deploy to wider audiences, this has created a dynamic of accessibility and security for authentication of user credentials in spaces shared by many users of different skill levels. The goal of these spaces is to create a balance between these two factors where users can be both protected and knowledgeable of best practices while still feeling invited in to learn, create, and share.

Bucknell University has deployed several versions of the Portal Box in its makerspaces, an access-control device that utilizes RFID cards to authenticate and provide access to trained users. Currently, in spaces such as the Maker-E at Bucknell, there are 3D printers, laser cutters, and all sorts of machines available to users at Bucknell with 24/7 access [5]. Users of the space scan into these Portal Boxes, which communicate with an online database to verify that they are trained for a particular machine. Portal Boxes are well-liked by their users and administrators, and they directly address the identified problem of balancing security and accessibility in makerspaces by making it easy for new users to realize they need training, get that training, and then operate machinery confidently and safely. The current boxes serve as a proof of concept, and we look forward to iterating on top of a successful product in a way that makes them more user friendly, operator friendly (maintenance/setup), secure, and adoptable, all while decreasing the overall cost from the last generation.

## Stakeholders

Because of their current deployment and development for and in the Maker-E, its administrators and supervisors are some of the biggest supporters and stakeholders in our project. Matt Lamparter, our client, is the most knowledgeable individual about the existing systems due to his role as Director of Bucknell’s Computer and Electrical Engineering labs, the main home of Portal Boxes. Matt’s knowledge about how Portal Boxes were developed and what it takes to maintain them has guided and will continue to influence our development, although we are cognizant of the fact that we must consistently seek out and consider outsider opinions in order to achieve our goal of broader adoption. Thus, we are actively looking for people and spaces outside of Bucknell’s makerspaces that may be interested in deploying Portal Boxes with their own machinery. Summarized in *Table 1.1* on the following page are individuals and entities that will or may be impacted by our project.

*Table 1.1: Stakeholders*

|  |  |
| --- | --- |
| Who it Impacts Directly | Who it Might impact |
| Matt Lamptarter | Non-Bucknell Makerspaces |
| Lab Directors | Other Spaces that Provide Equipment |
| Maker Council | Bucknell L&IT |
| Makerspace/Maker-E users |  |

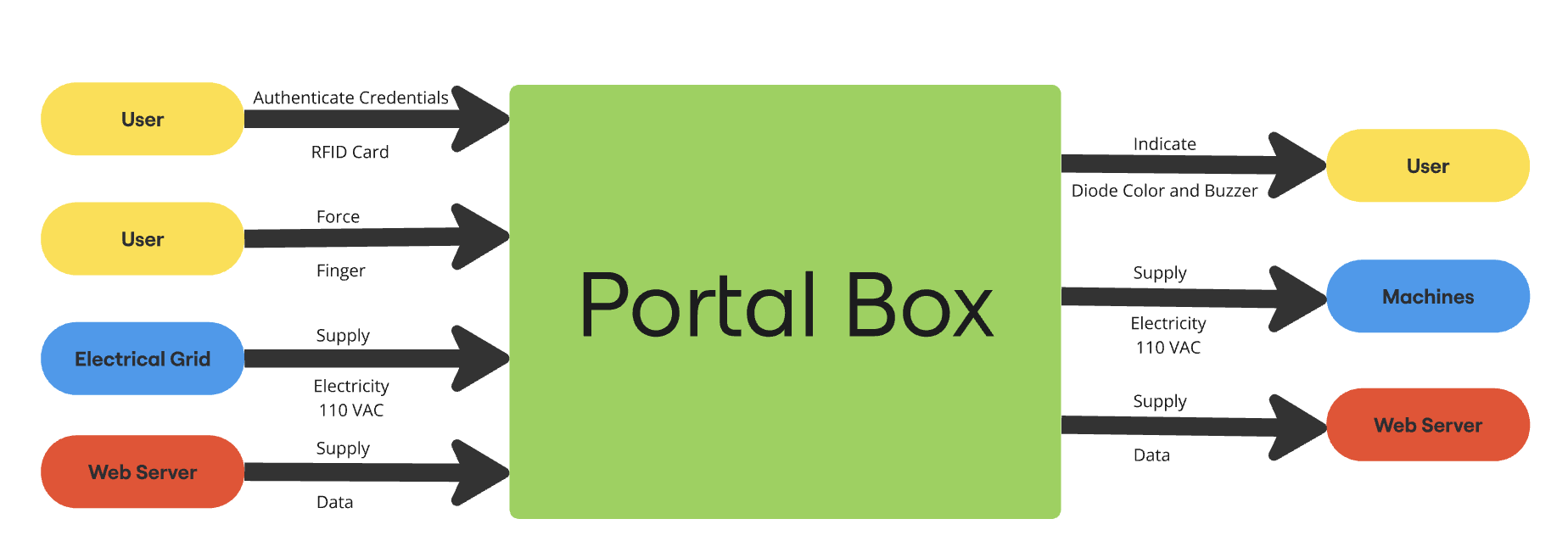
We strongly feel that the existing Portal Box system improves the operational efficiency of makerspaces while also providing enhanced security and safety measures, and many who have used the system note that it is more advanced than other makerspace systems they’ve encountered [6][7]. The strain on administrators and technicians is lowered by removing the necessity to actively monitor who is using what machines, and users are quickly able to start building without waiting for human permission. Furthermore, there are many spaces that operate in a similar fashion to what we see in makerspaces – such as workshops, labs, and hospitals, all which have sensitive equipment that often requires training or authentication to use. We believe that the Portal Box is capable of revolutionizing the way these spaces operate, greatly simplifying authentication and increasing confidence for users. One of the biggest benefits of Portal Boxes is eliminating the need for mental or physical tracking of users and what trainings they have. Furthermore, the system improves the overall security of a space by physically stopping unauthorized users from operating machinery. We hope to bring the benefits seen mostly in the Maker-E to more places on campus, and hopefully more places around the world.

We also hope to decrease the difficulty and strain required to set up and maintain the Portal Box system, which should contribute to broader adoption and make the existing caretakers’ lives easier. Jeremy Dreese is Bucknell’s Computing Systems Integrator, and he currently manages the database and web server which host the Makerportal on Bucknell L&IT VMs. By switching to a cloud-provider with automated flows and error detection we can simplify the maintenance and setup simultaneously.

## Portal Box Functionality

### Physical Box

The most recognizable component of the Portal Box system is a white box that allows equipment to be operated only if a user is authorized to use a certain machine. The box has male and female power headers which allow it to act as an intermediary block between a power source and a machine. Access is denied to the equipment unless an authorized user inserts their RFID card into the slot for it at the front of the box, at which point the box makes a network request to the database to check if the user is trained and authorized for the machine. If the user is denied access, no power will be delivered to the equipment. If authentication is successful then the power switches on, providing access to the machine. There are flows that handle timeouts, removals of cards, training new users, and resets. The boxes are currently operating with Raspberry Pi microcontrollers, which serve as the brains of the project. The main method of blocking access is with a AC-AC power relay which can be switched on and off, but the boxes also feature a USB interlock which is used as a secondary means to restrict access to machines by blocking the transmission of data. The boxes contain an RFID scanner for identifying users, a button for user inputs, a buzzer for audible feedback, and lights for visual feedback. Shown below in *Figure 1.1* is our Level 0 Block Diagram which summarizes the functionality of the Portal Box.

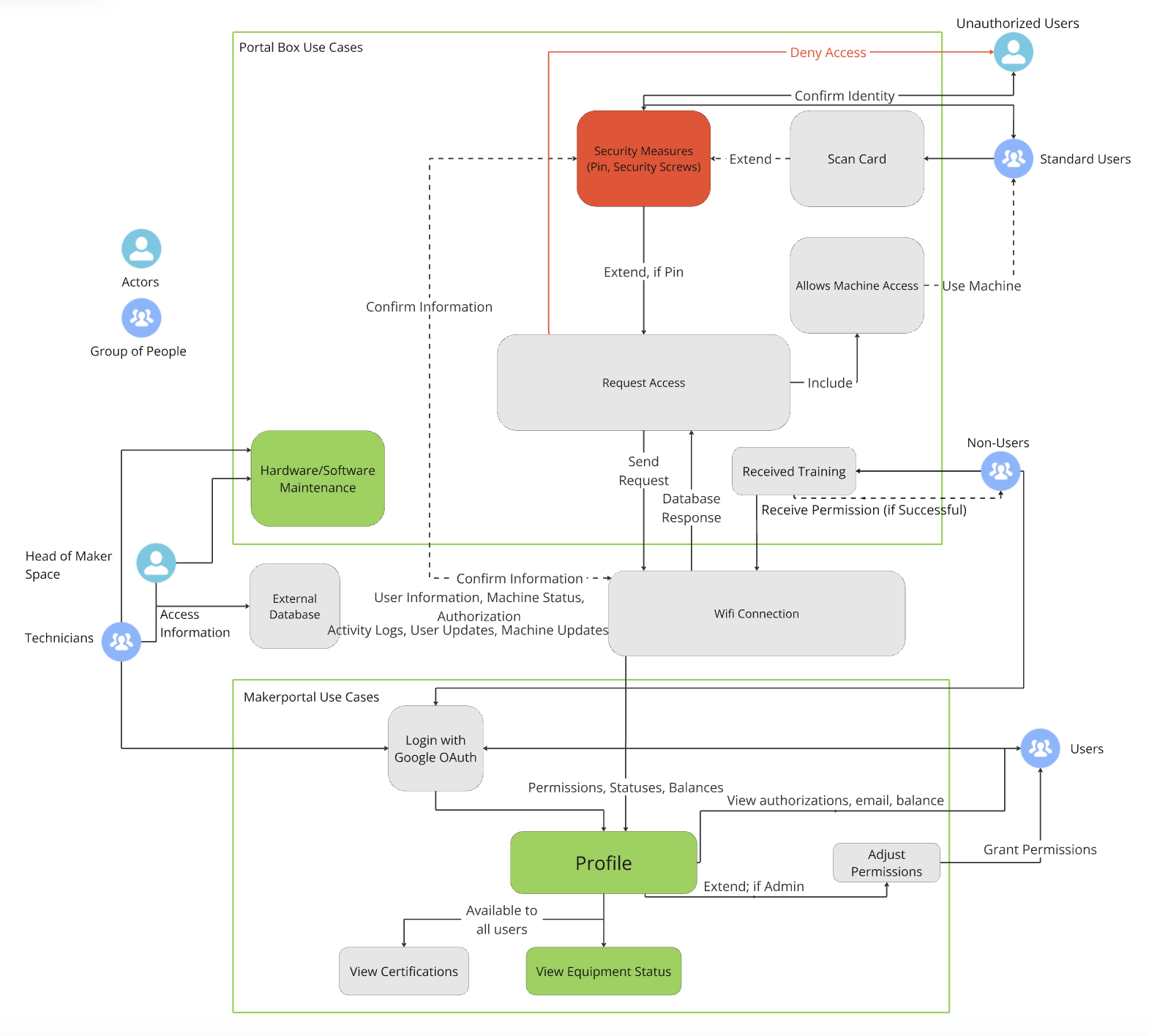


*Figure 1.1: Level 0 Block Diagram*

### Makerportal

In order for users to be created, granted permission, and operate machinery, the Portal Boxes interface with an SQL database over a websocket, and the Makerportal serves as the web front-end for the Portal Box system. When an RFID card is inserted into the Portal Box, the database is queried to check if the user ID attached to that card is authorized to use the machine, and only if the user has the correct authorization does the power turn on. The process of making a database request each time a user needs to be authorized can lead to some latency issues, and also makes the boxes highly dependent on their internet connection, so we hope to find ways to improve the reliability with local storage and granting of permissions. These permissions are currently granted by admins through the web front-end. The Makerportal also provides basic information to users and admins about the current status of each machine, and users can be charged for usage on specific machines depending on the configuration. Users can view their permissions and any balances on their account in the profile tab. Users login to the Makerportal with Google OAuth, and only users who have passed the Makerspace Access Quiz are able to login on the current system. When someone passes the Makerspace Access Quiz, their Bucknell email is logged in an Excel spreadsheet, and there is a script that pulls the emails from that sheet and assigns them a user ID in the user pool.

Shown below in *Figure 1.2* is our Use Case Diagram, with red highlighted blocks representing areas for security improvements and green blocks representing areas for accessibility or user-friendliness improvements. We hope to improve security by better organizing the existing system, simplifying the maintenance process, and updating both the Makerportal and physical box to provide more information to users. More specifics about these upgrades will be provided later in the report.

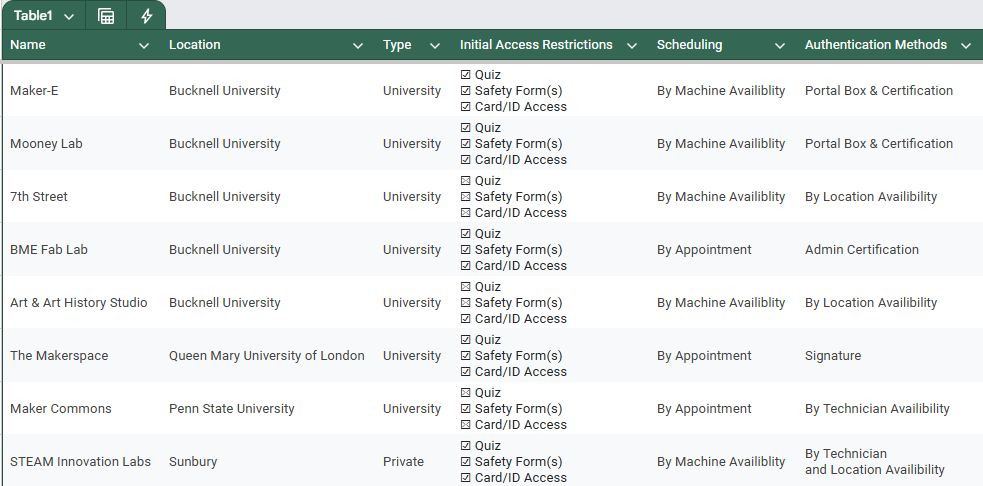


## *Figure 1.2: Use Case Diagram*

## Other Authentication Methods in the Space

The following page contains *Table 1.2*, which illustrates the operational structure of some of the makerspaces we’ve visited or interviewed representatives from. Further detail is provided after the table about the spaces, and all together we were able to pick up on strategies commonly employed in these spaces. The initial access restrictions, authentication methods, and scheduling protocol all work together to define the culture and flow of a makerspace, and there is a wide spread in what these places feel like.

*Table 1.2: Comparison of Different Spaces [5][6][7][8][9][10]*



At Queen Mary University of London, users must go through several training classes and exams in a variety of areas (fire, health and safety, first aid, etc.) to gain access to the space, before going through additional classes to be certified for each piece of equipment [7]. While QMUL has stringent training protocols, they need help with documenting this training and rely on something that many makerspaces rely on - trust. Each user’s credentials are tracked on paper, with signatures from an admin of the space to authenticate that training was received on a given date. However, physical access to machinery is not restricted once a user has access to the room, and the director of the space relies on his trust with his students that they will not act unsafely or unethically in the space [7]. Prior to accessing any equipment, students are issued an extensive checklist that begins with five health and safety induction courses. They also have to specify their purpose for wanting access, be it for classes, clubs, or recreation. The prospective user then must attend several other lectures on topics such as incident management, first aid, and PPE [11]. Once this checklist is completed, it is signed, dated, and stamped, meaning the user can now access the makerspace with their school ID.

This is quite similar to how Bucknell manages the base-level access to some of its labs and makerspaces, and restricting access to the space itself is a common practice to mitigate risk. At professional spaces, like STEAM Innovation Labs in Sunbury, members pay a subscription fee to enter the space, and prior to acceptance they must sign a waiver [6]. Their space is different from Bucknell’s spaces, which are typically fairly specialized, and there is a large contrast between arts and crafts versus dangerous machinery.

Once you purchase a membership from STEAM, you have access to the entirety of the space without any other required training (e.g. 3D printers). There are some machines in separate rooms like laser and water pressure cutters that require supervision or certification to use, however there is nothing actually enforcing this mandate. Those authorized to use these machines have their names written down in a physical list that may not be easily accessible to those who monitor the space, and the 24/7 hours of the lab make it possible that anyone *could* bypass the regulations [6]. The fact that users pay for admittance goes a long way in the trust department, as their payments go to waste if their membership is revoked. There have been no noted issues with individuals accessing machinery they are not supposed to, but there have been some issues with individuals building inappropriate or dangerous items [6]. STEAM has no established policy on infractions, and the noted incident resulted in a stern talking to. However, other spaces have stringent policies for how to handle improper usage or access of machinery, and these rules are responsible for the security that Portal Boxes can provide.

At QMUL there is a three-strike system for improper use of the makerspace, with a verbal warning preceding each strike. The first strike is usually for improper use of PPE and strike 2 is a written warning taking into account the student’s name and ID number. Strike 3 is for serious abuse of equipment that endangers students which results in an instant ban followed by a disciplinary meeting [7]. Bucknell’s Makerspace Agreement is incredibly similar, with varying levels of violations that result in varying levels of repercussions [12]. In both cases, the rules are rarely, if ever, used to punish individuals. This is because people simply do not break the rules, at least not in serious ways, when they are under threat of being caught or prevented from doing so. At QMUL, users cannot operate machinery at will, as the space is not open 24/7, is always staffed when it is open, and requires reservations for most of its equipment. In the Maker-E, Portal Boxes prevent unauthorized access to machines, and cameras serve as a potent backup if one were to bypass them. The glaring difference in the two spaces is that the Maker-E achieves the same, if not higher, levels of security, while simultaneously providing constant access to machinery and greater accessibility. Without Portal Boxes, this distinction would not be possible.

## Portal Box Evolutionary Timeline

### Portal Box V1

The V1 was developed in 2016. According to Joseph Kale, member of the original Portal Box team, the primary goal of the Portal Box was still to be able to authenticate users. Their thought process is to focus on a way to shut off power to the machines since they all share the need for power. The remainder of the project was to focus on power and making sure that it won’t overheat [13][14].

### Portal Box V2

The V2, which is currently used in the Portal Boxes as of 2024, developed in 2018 the primary goal of this iteration is to improve the functionality of the system. Primarily the code was revised and cleaned up as well as to improve the enclosure. Professor Joe Hass had a lot of influence on the embedded code as he developed the code using 15 different states which improved the functionality of the authentication with a lesser focus on cooling [15]. James Howe, a Computer Engineering student at the time, worked over the summer on improvements to the codebase. He wrote the Python FSM implementation that is currently in use with the Portal Boxes [16].

### Portal Box V3

The V3, developed in 2021, was an extension of the V2 which was never deployed at Bucknell. The enclosure once again had substantial changes which went back to a focus on controlling heat with an optional position for a small fan in addition to more ventilation in the enclosure to support the component swaps, such as swapping the Neopixel LEDs for DotStars to increase the functionality of the LED colors, made to the Portal Box which increases the strain on the PCB and microcontroller, but overall increases the accessibility if it finished development and were deployed [15][16].

### Current Development

Our Senior Design Project revolves around iterating and expanding on the currently existing Portal Box V2 & V3 and using it as a foundation to improve. One of the looming issues with these versions is that they are overbuilt using expensive components and by switching to cheaper components [5]. The cost of the Raspberry Pi used in the V3 Portal Box is 300% that of the microcontroller we plan on using, so making that switch presents a great opportunity to cut costs. Aside from cost, we intend to broaden the adoption of the system while simultaneously improving the user friendliness, security reliability, as well as the monitoring capabilities mentioned in the next section. Taken from the V3, we intend to adopt the code and make maintenance simpler and more reliable with a focus on user accessibility. We hope to move the Makerportal and database to a cloud provider where we can create a multi-tenant system based on the original instance. This should greatly simplify the setup process for the entirety of the system, aiding in our quest for broader adoption. The combination of a cloud deployment and upgraded hardware will hopefully allow us to achieve our goal of getting Portal Boxes outside of Bucknell’s doors.

## Deployments

### Current Deployments

Portal Boxes in their current state are implemented within Bucknell’s campus primarily in makerspaces. The Maker-E and Mooney spaces are extensive users of the Portal Boxes with a large portion of their equipment, however, a Portal Box is also currently deployed within the Biomedical Engineering Department’s Microscopy Suite with their Scanning Electron Microscope, located in Academic East [17]. However, our team believes that there is ample opportunity for Portal Box implementation in other spaces on campus and beyond.

### Potential Deployments

There are spaces on campus that currently have a similar set up to the Maker-E and Mooney, including the 7th Street Makerspace, the Art/Art History Digital Sculpture Studio, and the BME Fabrication Lab. Additionally, the STEAM Innovation Labs in Sunbury have been identified as another potential site for Portal Box deployment. This location would be a change from our current convention of Portal Boxes on university campuses, but after a site visit, we believe that much of the needs and operations of this space directly mirror our own set up. In our tour, we identified various high risk machines that would make sense to authenticate users on before granting them access.

In addition to makerspaces, an aspirational goal of the project is to broaden the scope of how Portal Boxes can be implemented. In a similar fashion to the BME Scanning Electron Microscope, we have received outside interest in this project, both at the aforementioned STEAM Innovation Labs as well as the makerspace director of Queen Mary University of London. While at this stage the former is significantly more feasible due to their proximity, interest of this kind from other universities shows the strength of the Portal Box concept. It is rare that people tell us they have no interest in Portal Boxes, but convincing them to actually adopt them is a larger challenge that we seek to lessen with our improvements to the boxes. As mentioned above, our development will focus on simplifying the setup, maintenance, and operation of the Portal Box system, which will make it simpler for new spaces to onboard. In addition, we seek to add value in areas that directly benefit the mission of the boxes, which are to improve user and operator friendliness, adoptability, and security while simultaneously decreasing the cost per deployment.

# **2: Project Value**

## 

*Figure 2.1: Accessibility & Security*

### Compromise of Security & Accessibility

The primary challenge addressed by the Portal Box project is the tradeoff between accessibility and security in makerspaces [18]. Our initial research helped us build a system map of how makerspaces operate, showing the relationships between the directors, technicians, users, machines, and more.[[1]](#footnote-1) We analyzed our map and realized that security and accessibility form a key loop that ties all of the agents of the system together. These two concepts are at the top of the mind of makerspace managers and directors, as they impact the flow, usability, and community of a space, all of which are critical for operability [19]. The choices that makerspaces make about their machine access structure often result in a linear spectrum, illustrated by the reinforcing loop above in *Figure 2.1*. Focusing heavily on security, like is done at Penn State’s OriginLab where technicians operate the machines for students, limits the accessibility and availability of machines. On the other hand, decreasing security and increased freedom can introduce vulnerabilities, potentially leading to improper or unsafe usage of machinery. Neither of these are good outcomes, and focusing too many resources on either security or accessibility can stall the productivity of a makerspace. The Portal Box system changes this relationship, transforming the linear relationship into a circular spectrum that enables for both enhanced security and accessibility without sacrificing aspects of either. This is illustrated as a balancing loop, where both elements can improve with each other, fostering a safe, supportive, and inclusive community of makers. This approach improves the usability of makerspaces while maintaining safe usage and lowering the barrier to entry for new or inexperienced users. From our group’s own experience as users of the Maker-E, the Portal Box system allowed us to quickly train on new machines and create powerful deliverables for class and independent projects. With our and others’ positive experiences in mind, we do not want to make significant changes to the overall flow and operation of the boxes, because they already do a good job of intervening in this loop. We feel that the existing Portal Boxes have begun the transformation of this loop from reinforcing to balancing, and the changes we seek to make are designed to not inhibit either of these essential elements of makerspace management. The existing boxes do a good job in both regards, but there are certain loopholes that limit security, and certain features that confuse users. By eliminating bypasses and introducing an upgraded UI, we plan to make the boxes more secure and more accessible, allowing them to greatly impact makerspaces in a positive way.

### Cut Costs

A secondary challenge is the high cost of manufacturing the Portal Box. The current design costs $202 for V3 and $140 for V2. The high cost of fabrication restricts the system's wide adoption. In order to make the Portal Box more accessible to a broader audience of makerspaces, we aim to reduce fabrication costs by roughly 20% where a majority of the cost reduction would be utilized to better improve the portal box giving a rough net 10% price reduction to cost. Changing out expensive materials and replacing expensive hardware, such as Raspberry Pi microprocessors with more affordable alternatives like the ESP32 microcontroller, will lower the overall cost of implementing a Portal Box into a space. These adjustments will make the system financially viable for smaller makerspaces with limited budgets. In the larger scheme of a makerspace, $200 is not much money when compared to the cost of a 3D printer or laser cutter, however, a preemptive investment in a Portal Box can extend the lifespan of a machine by preventing those who are inexperienced from breaking the machine and thus requiring repairs and incurring more costs down the line. By being able to implement Portal Boxes in these makerspaces, we can reduce the need for constant technician supervision, reducing the human cost of operating a space. Assuming a $10/hour wage, one technician has to work just 20 hours for their wages to match the cost of a Portal Box, and thus will reduce the number of technicians and the hours technicians need to work, thus reducing the cost to maintain a Makerspace in the long term.

### Broaden Expansion

Another area we seek to improve on the existing Portal Box projects is to get it out of Bucknell’s doors. We hope to deploy Portal Boxes in other makerspaces, and in order to do so we must add value that simplifies setup and operation of the system. Each makerspace already has a system that works for them, so convincing them to switch things up will be difficult if the setup is tedious or challenging. Penn State for example takes advantage of the abundance of students who are willing to be technicians in these spaces to maintain and approve requests to use the machines [8]. Many makerspaces rely on appointments with technicians for usage, where the technician themself operates the machine. Incorporating a scheduling system into the Makerportal would provide value for these spaces and others, and is a stretch goal for our team. In terms of facilitating a simpler setup, we aim to deploy the database and web portal on a cloud-provider such that each new space has essentially no front-end setup required. This will make things easier for those who might not have experience with managing a server, and will allow for maintenance flows to be carried out simultaneously for all deployments. Automating as much of the update and maintenance procedures as possible is also key, as installing a system that requires lots of debugging could place strain on the makerspace and distract the employees from helping makers. Portal Boxes also open the door for 24/7 access to spaces because there is no need for a technician to be present if someone is trained, and we hope to provide a set of guidelines for any space that wants to deploy Portal Boxes. Providing a centralized source of documentation with our recommended guidelines, fabrication files, and codebase will ensure that adoption is simplified as much as possible, thus aiding in our quest to get the product out of the university and into the field.

### Technical Competence

The new iteration of the Portal Box will improve things for makerspace and lab directors. The Portal Box system supplies directors with the peace of mind that machines are secure and protected by ensuring that only users who have been certified through training are allowed to use machines while those that have not been trained do not have access. There are currently a few technical issues with the Portal Boxes, including loopholes with the card mechanism, power relay, and a non-functional USB interlock. By eliminating these loopholes and fixing existing issues, we will push the technical competence of the project forwards. The base-level acceptance criteria for our project is to replicate the existing functionality while closing these loopholes, and we intend to provide further value by leveraging the existing framework. Machine usage is already tracked with timestamps, and formatting this data will allow directors to allocate resources to the correct machines, cutting out wasted spending on machines that are used less in a space. If it is noticed that machines are being constantly used or vice versa, more or less money will be spent on new versions or improvements to that machine. A major point still under consideration is the enclosure. We would like our project to have a professional look, but more important than that it needs to be simple to build with commonly available tools and resources. We want to explore resin printing and other ways to build a more refined enclosure, but we must balance this or provide options for the various spaces Portal Boxes could be deployed in.

### Ease of Operation (Admins/Technicians)

Technicians of these spaces are often paid to monitor machine usage and verify that end-users are using machines properly, not violating the terms of the space’s user agreement. The Portal Box reduces the need for technicians to monitor machines at all times, freeing them up to spend their valuable time helping makers with their projects. When combined with cameras, Portal Boxes act as the wall and cameras act as the inner moat around a castle – if you circumvent the Portal Boxes you will be caught by cameras. By eliminating the stress that can come with supervising a space, technicians become more approachable and available. The database and Makerportal provide a tremendous amount of value through their logging and permission tracking functionality, which we will continue to improve on with data analytics for both users and admins. There is no need for mental or physical notes about who is allowed to use what, who was operating something when it broke, and so on – all of that information is present on the Makerportal and ready with a few clicks. By switching to a cloud-hosted platform, we will greatly increase the ease of operation of the front-end, making it easy for even someone with zero experience to launch their own Makerportal.

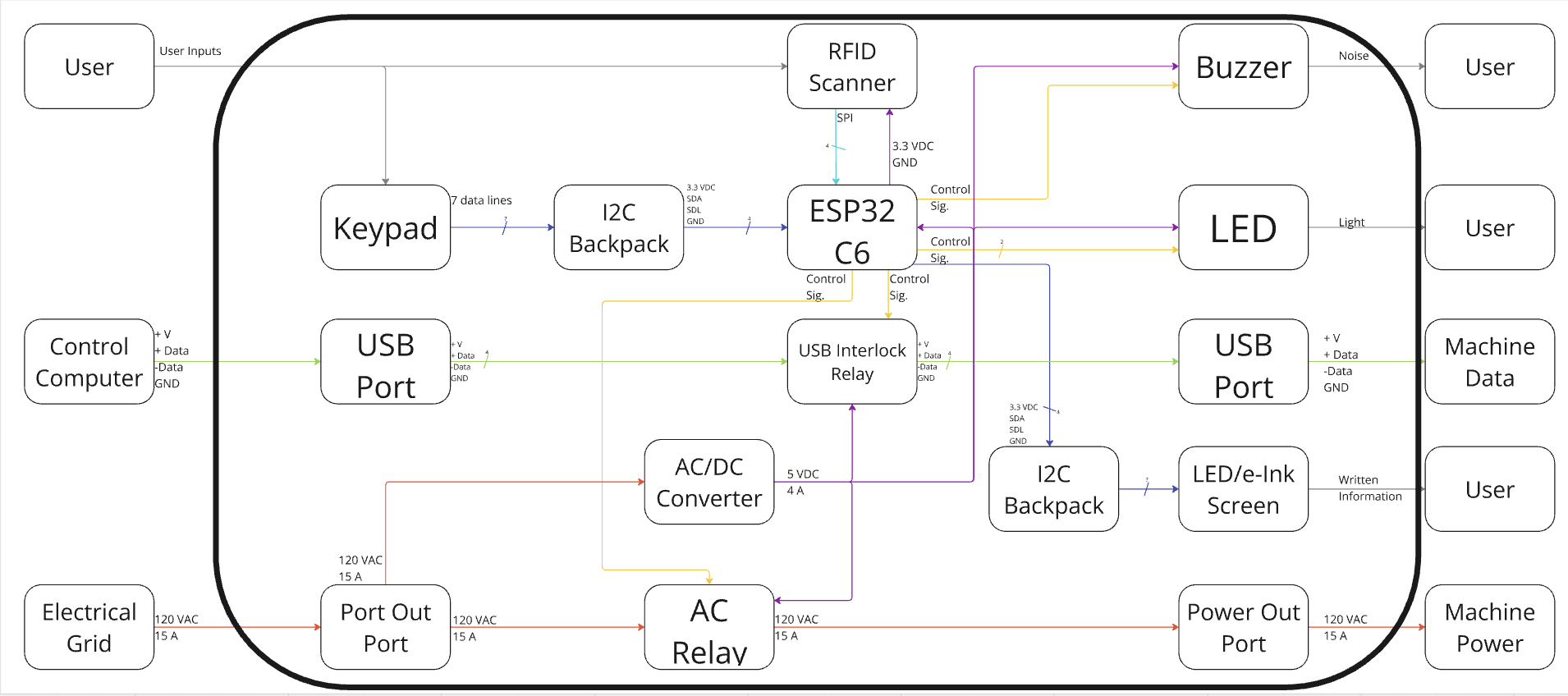
### User Friendliness

The new iteration will also improve usage for end-users (those accessing machines), with improved user friendliness, training flows, and latency. Each box will locally store a list of users that are authorized to use it, decreasing the number of database requests required for operation. Furthermore, users will be granted permissions through an automated flow, rather than requiring an admin to log into the Makerportal and manually grant permissions. This change will cut down on the effort needed to train users and the wait-time for users to get trained. Right now, the Makerportal serves as the hub of operations for Portal Boxes. Admins can assign an RFID card to a user, grant and remove permissions, and monitor machine status. End-users have access to this portal, but many are not aware of the feature that shows machine availability. We plan to bring more attention to this type of information in an effort to make using machines as convenient as possible. We will implement a Dashboard with per-machine statistics, something that both technicians and end-users have expressed a desire for. We hope to look at historical data and create a predictive feature where users can select a machine and get an idea of when it will be available for usage, or at least see what times of day over the last few weeks the machine was available. At the point-of-use, we will implement a screen that will provide more information than the existing lights and sounds, which can confuse some users [5]. The boxes will be designed to have QR codes linking to training modules displayed directly on them, allowing for quick and direct access.

# **3: Project Functions & Implementation Planning**

### Overall Functionality

The overall functionality of our Portal Boxes will look very similar to the existing system, with all changes aimed at directly improving on one of our goals (security, accessibility, cost, broader acceptance, user friendliness). *Figure 3.1* below shows how we plan to achieve the functionality, with notable changes from past Portal Boxes being the keypad, screen, and current-sense IC. These are the new components required to implement our point-of-use security and accessibility improvements, as well as measure power data to be displayed on our dashboard. Another significant change from the earlier Portal Boxes is the microcontroller, which will now be an [ESP32-C6](https://www.espressif.com/sites/default/files/documentation/esp32-c6_datasheet_en.pdf) [20]. Switching to the more professional and less complex ESP32 allows us to significantly reduce the cost of the Portal Box, and should allow for a smaller footprint due to its lesser size. We also anticipate the switch cutting down on the power consumption of Portal Boxes, and the Raspberry Pis used in past iterations are power-hungry and overbuilt for this application.

**

*Figure 3.1: Level 2 Block Diagram*

Our Bill of Materials, derived from *Figure 3.1,* is not finalized, but the table shown below in *Table 3.1* gives you an idea of where we are headed and what we are using for our initial prototyping. Many of the components are the same as were used in the V3 Portal Box, but we intend to do further research to ensure the reliability of the components (both performance and availability). The new components introduced into the schematic currently are outweighed by the reduction of cost in the microcontroller, and we are trying not to obsess over cutting cost. That being said, we recognize that introducing an incomplete or poorly developed screen or keypad could set the boxes back from their last iteration, or make the product appear less complete. If we are unable to adequately develop those new functions, they will be dropped in favor of further cutting costs and reducing complexity of both setup and operation.

*Table 3.1: Bill of Materials*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Quantity per PCB** | **Unit Cost** | ID |
| Membrane 3x4 Matrix Keypad | 1 | $3.95 | 419: Adafruit |
| Reset Button | 1 | $0.49 | TL3336AF160Q: Digikey |
| Buzzer | 1 | $1.27 | AI-1223-TWT-3V-2-R: Digikey |
| LED Screen | 1 | $12.95 | 398: Adafruit |
| RFID Scanner | 1 | $8.99 | RC522: Amazon |
| AC Plug (Male) | 1 | $2.02 | Q219-ND: Digikey |
| AC Plug (Female) | 1 | $6.54 | 161-PX0675/PC/12599: Mouser |
| AC-DC Convertor | 1 | $13.19 | 1866-5327-ND: Digikey |
| USB Port | 1 | $1.53 | USB1135-15-9-L-B: Digikey |
| LED | TBD | $0.17 | NS107S: Superlightingled |
| ESP32-C6 | 1 | $9.95 | 5672: Adafruit |
| I2C Backpack | 2 | $9.95 | 292: Adafruit |

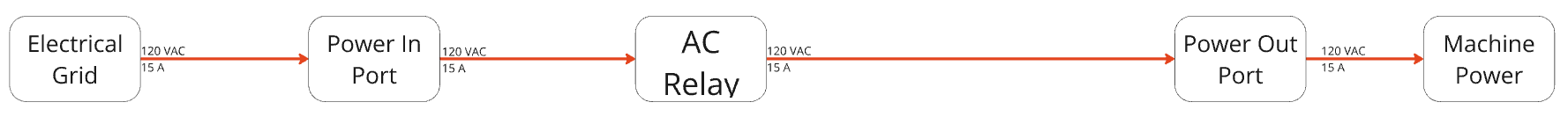
## 

## Specific Functions

### Power Relay (Colton)

#### **Description:**

The power relay is one of the most critical components within the Portal Box. For machines whose access is controlled via cycling power, the power relay acts as the medium which physically allows or blocks power from routing from the wall supply to the machine, shown in *Figure 3.2.* This component is intended to handle 120V and up to 15A flowing through the box. This relay is controlled by an output from the microcontroller which gives inputs into whether authentication was successful or not.



*Figure 3.2: AC Relay Blocks*

#### **Development:**

The components and layout from the Portal Box V3 will be kept the same for our iteration. However, one improvement I believe could help extend the lifetime of the Portal Box is adding a layer of modularity to our design. Because relays are mechanical components, there is the possibility for degradation over time. Since the power relay is the central component for the usefulness of the Portal Boxes, a relay failure is unacceptable. However, if we were to add a socket to allow for easy replacement of the relay, this could greatly improve the lifetime of our design and protect against obsolescence.

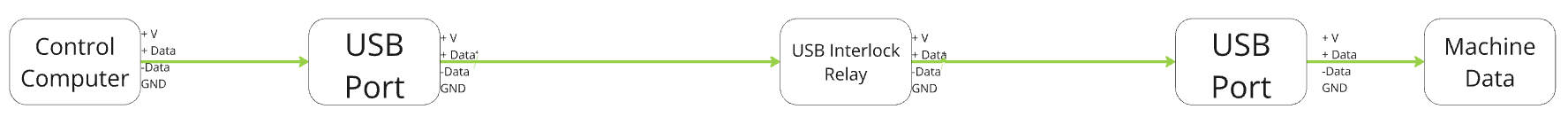
#### **Roadmap:**

Because the existing components and design for the power relay are conventional and known to operate well, more time is anticipated to be spent on investigating the socket design for the power relay. In addition to this, if a socket is implemented, this will then need to be tested under full power to ensure that the component is still able to operate as intended.

### USB Interlock (Colton)

#### **Description:**

Similar to the power relay, the USB Interlock is also used for blocking or allowing machine access. However, the USB interlock is intended for machines whose power cannot be cycled on or off for safety or maintenance reasons. In order to resolve this issue, the USB interlock is used to stop data from being sent from the computer to the machine, depicted below in *Figure 3.3*. This way, the machine is always powered, however, in order to interface with the machine, a user needs to authenticate to allow information to be sent and give the machine commands.



*Figure 3.3: USB Interlock Blocks*

#### **Development:**

The V2 and V3 Portal Boxes have a USB Interlock built into their system. However, it was discovered that the current design does not work as intended. The system had a cut off relay for the 5V power line between the computer and machine. However, because both the computer and certain machines have the ability to power this 5V port independently this method was not successful for actually denying data transmission. For the next iteration, the USB interlock will instead use a double throw single pole reader relay to cut off the data and clock lines of the USB connection. This way, even if both ends are independently powered, no data can be sent through the box until the connection is closed.

#### **Roadmap:**

The reader relay on the data and clock lines has successfully been tested and appears to work. This system needs to be tested further to ensure that the switch can properly accommodate data rates and also be implemented onto the PCB board.

### Current-Sense IC (Colton)

#### **Description:**

The current-sense is a new feature being added to our iteration of the Portal Box. One problem we found with the previous version of the Portal Box is that machines stay powered for the entire duration that a verified card is inserted in the box. This means that it hasn’t been possible to distinguish between the time that the machine is powered and how long the machine is actually in use. The current-sense addition will monitor the power consumption use of the machine both to monitor overall consumption during normal operation while also allowing data collection for how long the machine is actually in operation throughout its life.

#### **Development:**

This component needs to be tested under full power to ensure that it is sufficiently capable of handing 120 V up to 15 A and accurately transmitting power consumption information to the microcontroller. This component is a hall-effect sensor that sends a quantized signal from the IC to an analog pin on the microcontroller.

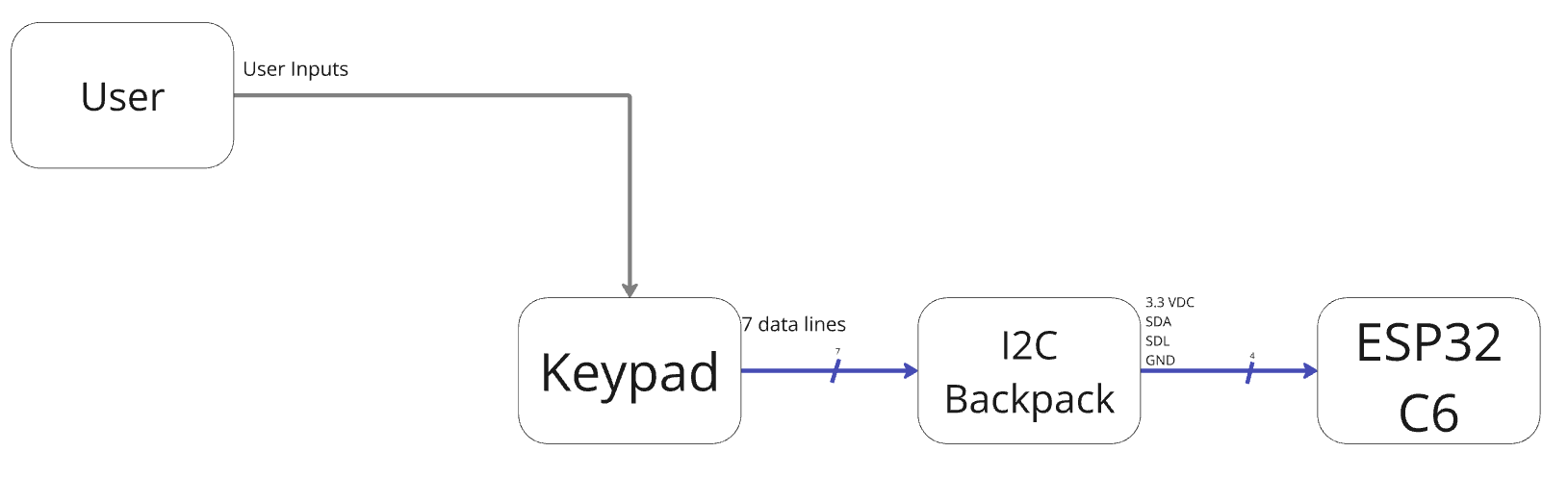
#### **Roadmap:**

The current-sense is in the early stages of testing. The component select should be operable under AC power and is rated for up to 21.5 A. This system remains to be tested under full power and with a machine. If this component is included in the final design, it would require significant reworking of the PCB since all of the current from the wall to the machine needs to pass through this IC.

### Keypad (Aidan)

#### **Description:**

The Portal Box will feature a significant upgrade with the integration of a keypad, replacing the existing single button interface. This will improve the user interface and also enable two factor authentication (2FA) by ensuring that the RFID card inserted actually belongs to the person who inserted it. Updating the box with a form of 2FA will increase the system's security and safety [21][22]. The keypad will allow a user to input a personal PIN number in addition to using their RFID card which provides this extra layer of verification. 2FA is a trusted practice in many modern fields where equipment is shared, including the medical field where it is used to limit access to machinery as well as to verify that samples belong to specific patients [22]. Drawing from what works in another field where many different nurses require quick and immediate access will help us ensure new onboarders are confident that the Portal Box achieves its goal of restricting access only to authorized people. By implementing a standard flow that users are already familiar with, we will not introduce significant blockers to accessibility. The information inputted on the keypad by the user will be communicated to the ESP32-C6 as shown in *Figure 3.4*.



*Figure 3.4: Keypad Blocks*

#### **Development:**

The keypad has been directly wired to a microcontroller and was able to read different button presses on the keypad. We are able to read the different numerical characters and keep the functionality of the one button interface expanding into using a PIN number.

#### **Roadmap:**

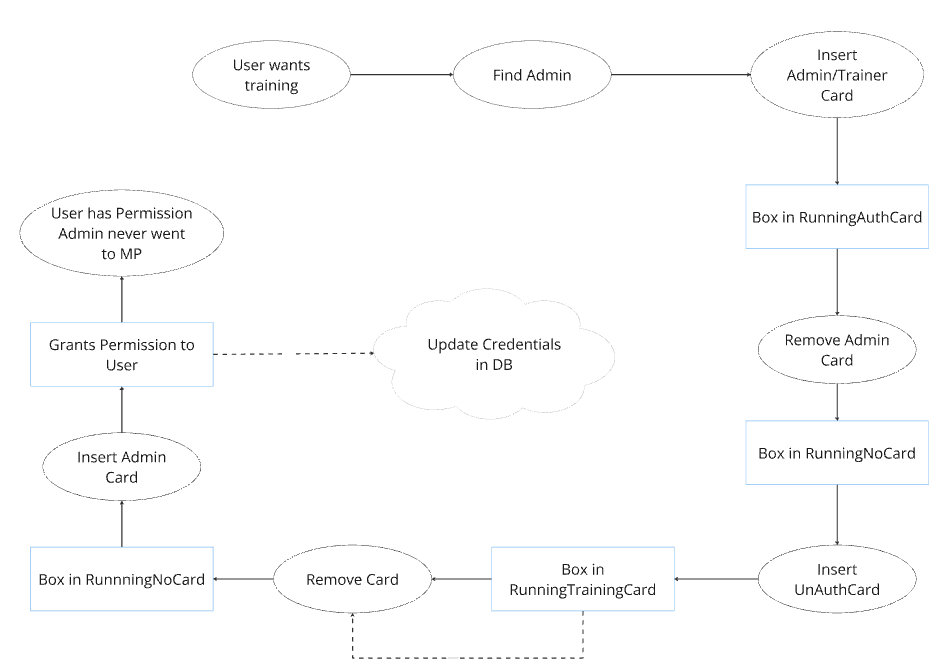
Implementing this into the main code of the Portal Box is a big step. The PIN number has to be sent to the user quickly, while also being uploaded to the Portal Box, so it knows what code it needs to verify. When this addition is made into the Portal Box it will add another level of security that allows the head of makerspaces to have more trust in its users.

### 

### RFID Scanner (Aidan)

### Description:

The RFID scanner is the primary authentication method used by the Portal Box. A user will insert their RFID card, which is subsequently read by the RFID scanner, which reads the embedded UID and verifies authorized access. The RFID scanner can also write to the RFID cards, which allows us to add authentication and store information on the cards themselves. Adding a new user as an authenticated user involves using the RFID scanner with the Portal Box’s local storage. As shown in *Figure 3.5*, after reading the ID of a non-certified user, then a training or administrator ID, will put the Portal Box in training mode. Once the same uncertified user card is in training mode, scanning the same uncertified card ID enables the Portal Box to update its local storage with the new user’s RFID credentials.



*Figure 3.5: New Training Flow*

#### **Development:**

The RFID scanner will function largely the same as it did in previous versions of the Portal Box. However, we will be using the RFID card for slight changes, like with our new training flow. The scanner is capable of reading an RFID card’s ID number in the correct format. Furthermore, the new training system will store the new trained user’s data and UID directly locally on the Portal Box, and then upload it to the cloud database.

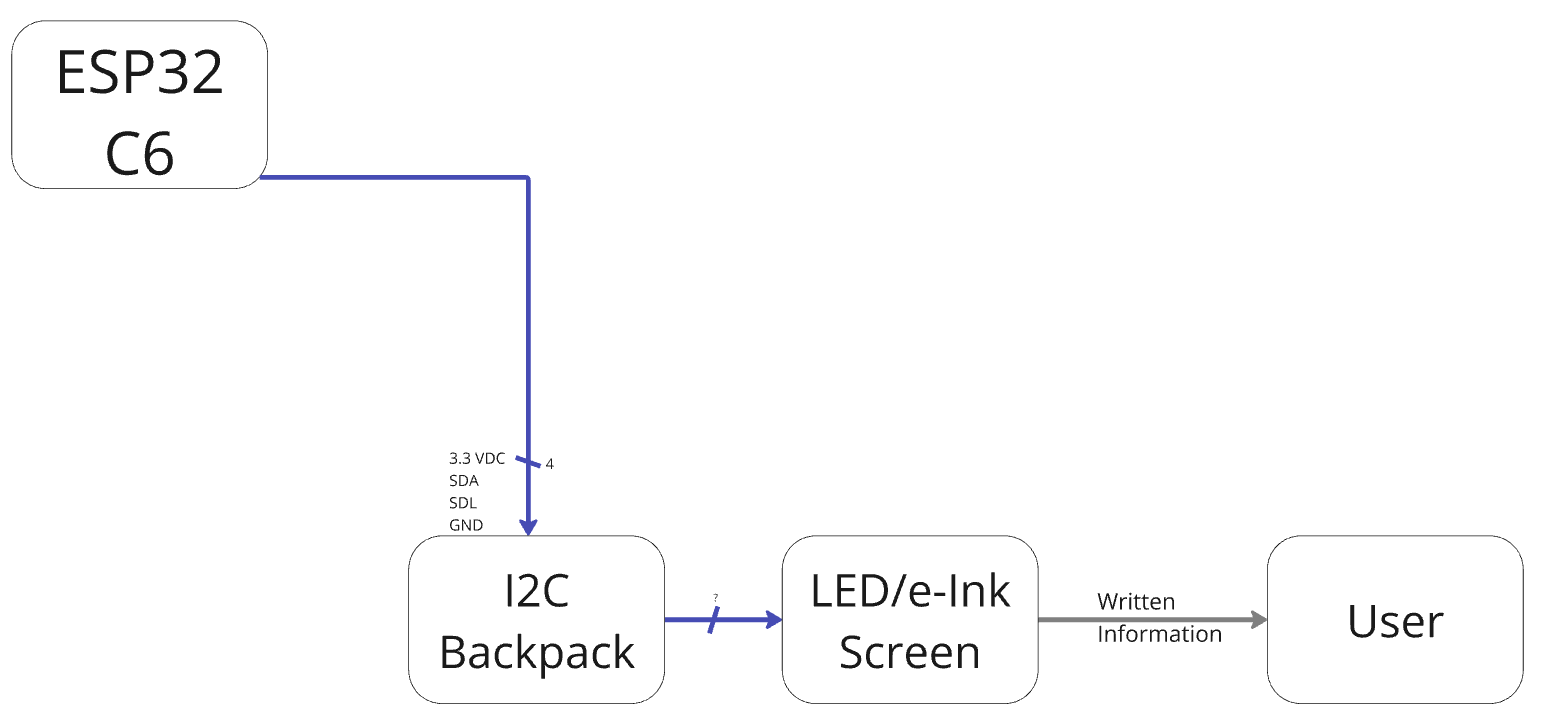
#### **Roadmap:**

The RFID scanner and keypad are implemented to verify that both security measures work together seamlessly. The RFID scanner stores IDs locally on the ESP32. The current system the Portal Box has in place does not have 2FA. By adding another level of security, ensures that the Portal Boxes are still accessible but adds another layer of protection that ensures the safety of the space and its users.

### Screen (Peter)

#### **Description:**

Using a screen attached to the Portal Box will allow it to convey information to users and staff more efficiently than the current system relying on different LED colors. The use of a screen will also make the box more visually appealing and more interesting to interact with. The two main competitors for this are the [Adafruit RGB backlit 16x2 LCD screen](https://www.adafruit.com/product/398#technical-details) [23] and the [Adafruit IL0373 2.9” Tri-Color E-Ink display](https://www.adafruit.com/product/4778) [24]. We have decided to use the E-Ink display as it is much less limited in regards to the data that can be shown compared to the LCD screen despite the E-Ink’s slower refresh time. At all times, the left of the screen will display the Dozuki QR code for the piece of equipment the Portal Box is connected to, which can be easily changed as the QR code is simply stored as a bitmap image in the ESP32’s root directory. The right side of the screen will display various messages that vary depending on the state of the Portal Box.



*Figure 3.6: LED/e-Ink Screen Blocks*

#### **Development:**

One of the most important factors in regards to testing the screen is ascertaining the limits of both its time to update and the extent of what can be shown on it. The E-Ink screen has a resolution of 296x128 pixels, which allows for much more comprehensive display options than the 16x2 LCD screen. We currently have five different screen display states designed, all showing a QR code with a message representing a different state of the box.



*Figure 3.7: E-Ink*

#### **Roadmap:**

While some displays are designed, there are still more that need to be done to cover every separate visual state of the Portal Box. The screens have also yet to be integrated into the full embedded system. We would also like to further test the E-Ink display to see if it can update more quickly. The libraries it uses are currently unfamiliar which is the primary factor impeding this.

### Card Insertion Mechanism (Peter)

#### **Description:**

One reason we plan to improve user opinion of the Portal Box is by revising the method that the user inserts their access card into it. We currently plan to design a small 3D - printed holder that the card will slot into. This holder will be attached to the RFID reader and held in place with an internal support structure inside the box. As a result, the user will insert their card into the Portal Box itself, as opposed to placing it in a compartment on the outside of the box.

#### **Development:**

As of now, we have preliminary sketches of how the card insertion mechanism will function. So far, we have prioritized setting up and testing the screen. We also have sketches of how it might be internally supported.

#### **Roadmap:**

We need to create 3D models of the card holder, as well as model how it will be internally supported within the box. We also need to make sure that it won’t physically conflict with the other internals of the box, like the PCB.

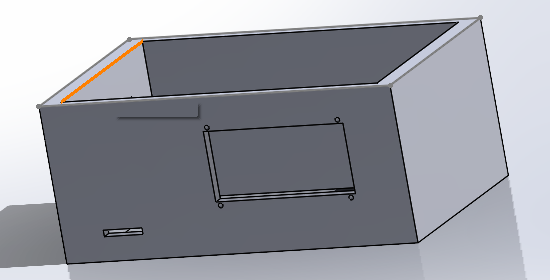
### Enclosure (Peter)

#### **Description:**

The enclosure for the Portal Box is a simple box container that houses all of the circuitry and components. Its goal is to look as professional and premium as possible while simultaneously protecting the PCB, microcontroller, card reader, and other parts inside. It features cutouts to allow for mounting of the keypad, screen, card insertion mechanism, and power outlets from the interlock. Our goal is to make it possible to be quickly assembled by most makerspaces.

#### **Development:**

We currently have several sketches of how the finished enclosure will look both with and without the screen installed. We also have 3D models for how the finished enclosure will look with cutouts for both the LCD and E-Ink screen, shown below in *Figure 3.8.*



*Figure 3.8: 3D Model of Potential Enclosure Layout*

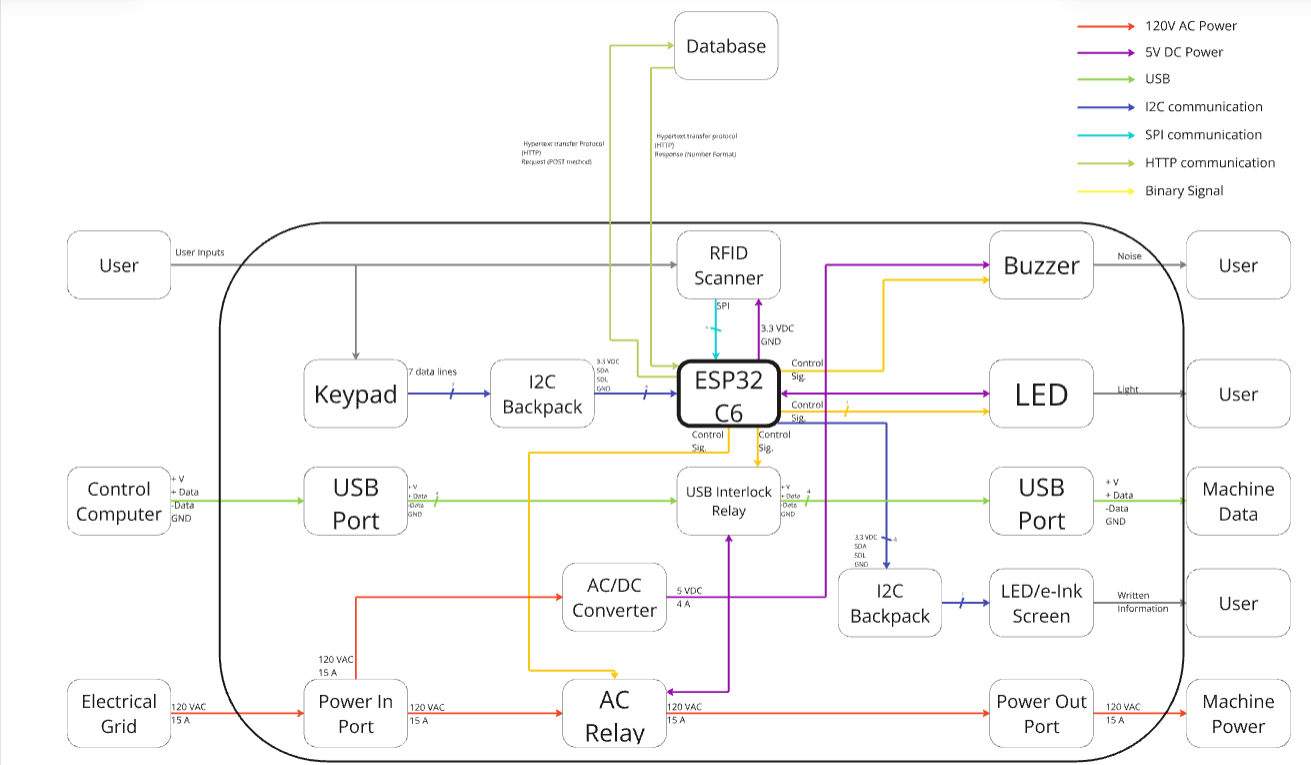
#### **Roadmap:**

Our next objective is to turn these 3D models into files that can be laser cut and assembled into a physical prototype. We also want to explore more orientations for the box. Another avenue we would like to explore would be using more materials to construct the box. While the ideal goal is to have the entire box able to be constructed in most makerspaces, we want to investigate more professional options for fabricating the enclosure, such as resin printing. While this is more expensive than laser cutting or 3D printing the box in the Maker-E, this would help add a more premium field to the Portal Box as well as make it more suited for mass production.

### Embedded Software (Kevin)

#### **Description:**

Residing on the ESP32-C6. The Python code, derived from the python code stored previously on the Raspberry Pi Zero in the V2, takes inputs from the user through the peripherals and communicates to different parts of the Portal Box through the other functions and is only accessible by those with admin permission. The code acts as the communication between the other functions and the ESP32 is how software can communicate to hardware as shown in bold in *Figure 3.9* below. From the figure, all of our components share a connection to the ESP32 as represented by the color of the arrows coming from and to the ESP32. Users of these makerspaces will indirectly use the code as anyone using the Portal Box would receive access to the machines through the ESP32 and will be automatically deployed once the device receives power through the 120V Female Port and will constantly be running as long as it has power.



*Figure 3.9: ESP32’s Role in Block Diagram*

#### **Development:**

We have successfully flashed the ESP32-C6 and are able to communicate with physical components and read/control them through the software. We can currently read a Keyboard, control a button, and control Neopixel LEDs. Additionally, we have been able to establish a WIFI connection to Bucknell.iot. We have also begun broadly outlining the software and will be soon translating the Circuit Python code from the Raspberry Pi to MicroPython on the ESP32.

#### **Roadmap:**

Our next step in the software is to continue to test individual devices together as well as start to develop code for the other components. The next immediate step is to receive user information from the database over WIFI, which we can then store locally on the ESP32 in which we can start once we can read RFID cards and get them connected to the ESP32-C6. Our goal going into the next semester is that we want to be able to send and receive data from the API website as well as get the majority of the software in some sort of functioning system.

### Database (Axel)

#### **Description:**

The system functionality is currently achieved by leveraging a MySQL database, hosted on Bucknell L&IT’s server. Both the Makerporal and the Portal Box interface with the database through HTTP requests and responses which allow for authentication, updates to permissions, and logging. The schema is publicly available on GitHub, and we do not intend on changing a whole lot because it is effective.

#### **Development:**

For our initial development, we intend on creating a locally accessible database with the same schema and updating it to allow for the new pin-authentication flow. This will allow us to test flows and conduct unit tests without the entirety of the system in place.

#### **Roadmap:**

Using Docker containers, we can store the contents of a database in a volume, allowing for continued functionality each time a design engineer spins up the container. As we move from prototyping to our final deployment, we hope to create a multi-tenant platform where each instance has access to its own database, allowing directors to have full control over their space without impacting others. In the next month and a half we will explore what solution is the most effective and reliable for our proposed cloud-implementation, which will be touched on further below in the *Deployment* section.

### Makerportal (Axel)

#### **Description:**

The Makerportal is the front-end web application that allows users to view machines and their profile. We spoke to James Howe, who worked on the current codebase, and he told us that he explored creating an informational dashboard during his development but never got to it. Another feature he mentioned was allowing for more control over the Portal Box from the Makerportal, such as placing a box in an “Out of Order” state where its access is shut off remotely. Both of these features were on our mind already, and we intend to implement a dashboard showing peak times of usage, machine power consumption, and more.

#### **Development:**

We are pushing development of the Makerportal off for now, as the current implementation is effective and generally well-liked [5]. However, a dashboard could provide immense value to both users and managers, so as we move past prototyping and towards a high TRL project we will implement it. In order to create the dashboard, we first must see what representations we can create with the data available, and then integrate those representations into the application.

#### **Roadmap:**

First, we will create graphs using the timestamps and access logs that exist in the database. Using plugins, we will integrate these into the Makerportal and create an interface to control what is being shown to the user. We can then refine the graphs and ensure that our client and users find them readable and useful, and that they can reliably be created and refreshed.

### Code & Documentation Deployment (Axel)

#### **Description:**

The current presentation of the project is slightly scattered, with various elements of the documentation residing in Google Drive folders, webpages, and blogs. The codebase lives on GitHub where it is easily accessible, yet the process of actually deploying it and getting it working is strenuous, especially if the user is inexperienced with web apps. We will explore cloud-based options to create a multi-tenant platform that would encapsulate the existing Makerportal and database. A makerspace director would visit the homepage, register their space with an OAuth flow, and then a copy of the database and Makerportal would be created for their space. We will also consolidate the existing documentation into a shared point-of-access for each iteration of the Portal Boxes, allowing new users to see the history of the project and start building their own.

#### **Development:**

The development of our multi-tenant system is something totally new to everyone in our group, so we will lean on experts in the field to guide us. A commercial option is likely the best and simplest to set up, and we can start by deploying the existing codebase using our chosen infrastructure to ensure functionality.

#### **Roadmap:**

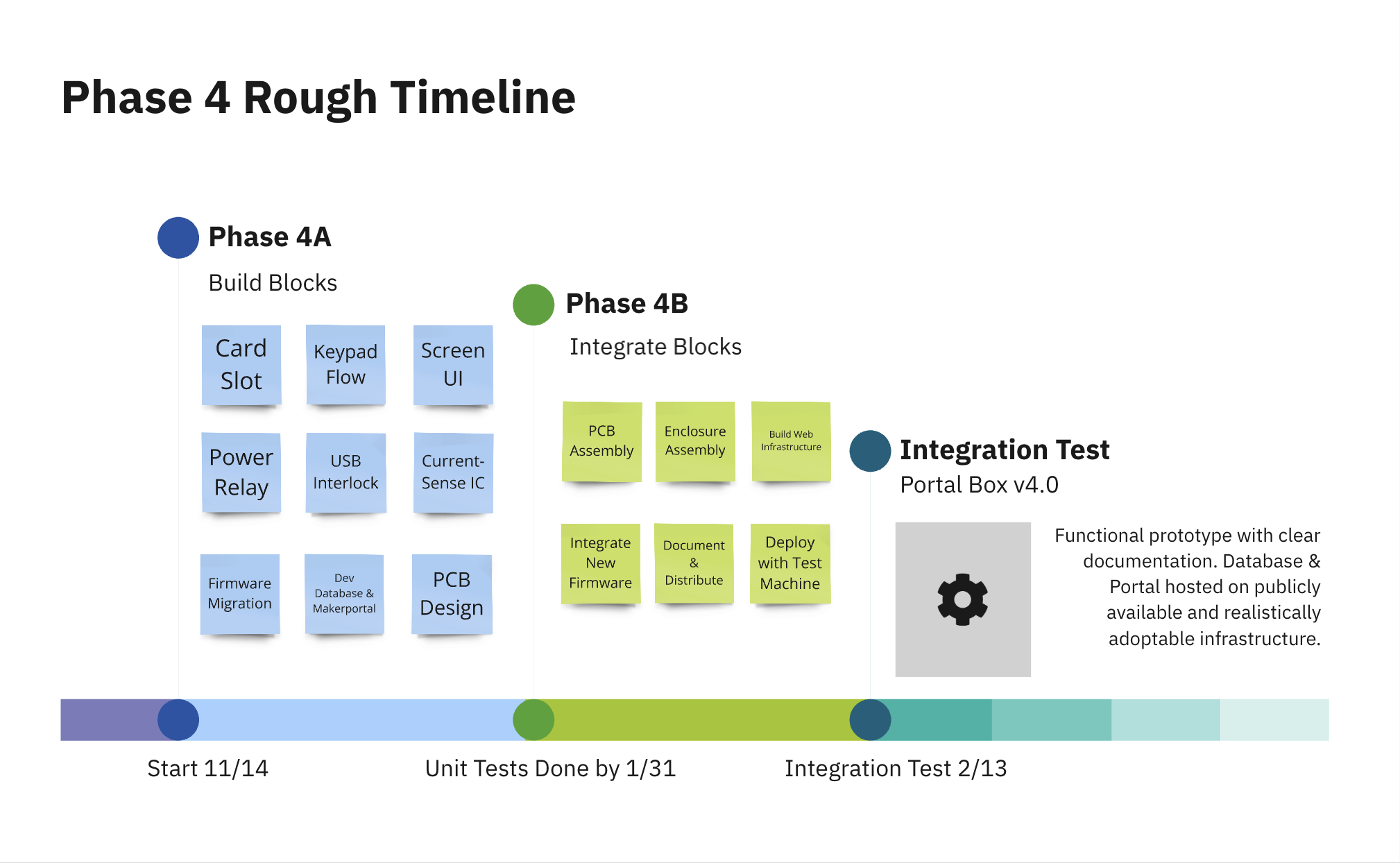
This deployment is not crucial for our unit or integration tests conducted in Phase 4A and Phase 4B, but as we move through Phase 5 we will rapidly move our existing work to the cloud provider of our choice. We have funds allocated for any expenses, which we do not anticipate to be significant due to the small initial size requirements. By the time we reach our final acceptance test in April, we should be able to demonstrate the simplicity of setting up a new instance of the multi-tenant platform and the ease at which new makerspaces can join the Makerportal system.

# **4: Fabrication & Performance Evaluation**

## Implementation Planning

### Phase 4 Breakdown

Leading up to the end of the fall semester, our design engineers have all been making progress towards the implementation of their blocks. We are several weeks into Phase 4A, which is where all of the individual blocks will be built, datasheets will be prepared, and then the blocks will be reviewed in unit tests with the professors. We began the phase on November 14th, and we will undergo our final unit tests by January 31st. We will then move into Phase 4B, integrating the blocks already developed by the design engineers with the goal of a functional prototype by February 13th. In this phase we will integrate hardware modules with the PCB, build the initial cloud-infrastructure, prepare our public documentation structure, and deploy our box with a test machine (likely a lamp). At this point the system should operate as we are used to with Portal Boxes, with the additional security, training, and screen features that we’ve added. The dashboard will be in early development, and at the integration test we will demonstrate that meaningful representations can be generated from the data. Shown below in *Figure 4.1* is a rough outline of the two parts of Phase 4. Additional information about the timeline of tasks can be found in our Gantt Chart.[[2]](#footnote-2)



*Figure 4.1: Phase 4 Timeline*

### Phase 4A: Unit Tests (Ends 1/31)

#### **Aidan (1/29):**

#### Demonstrate ESP32 reads card when instructed, accepts/denies pin-codes, and manages the new training flow (stores locally, pushes/pulls with database).

#### **Colton (1/31):**

Demonstrates the power relay and USB interlock appropriately block/enable connections. Demonstrate the current-sense IC measuring useful data and what representations we can create with it. Show and justify PCB design.

#### **Kevin (1/31):**

Demonstrate functional ESP32 with internet connection, FSM implemented, pins assigned and functional as needed. Code the the RFID module may already be integrated for testing purposes, although it is not critical at this stage and inputs can be simulated. Show that there is sufficient memory and storage on the ESP32, or specify how much more we will need.

#### **Peter (1/29):**

Demonstrate improved feel/user-friendly features. Card insertion, screen display, and keypad integration should all be demonstrated. Overall enclosure should be modeled and maybe physically demonstrated, although it is not critical for the entire enclosure to be assembled at this point. Designs should be ready to move forward with fabrication during Phase 4B.

### Phase 4B: Integration Test (2/1 - 2/14)

After completing our unit tests, we will rapidly integrate the system. Inputs and outputs will be connected between blocks, the PCB will be assembled, and the enclosure will take form. Since Aidan and Peter will conduct their unit tests slightly earlier than Colton and Kevin, they can pivot to integrating the new features that they created with the existing functionalities that Colton and Kevin are migrating. While we expect the enclosure to still be in development at this stage due to changing constraints, it is important that we demonstrate how the screen and keypad will interface with the rest of the components. At this stage we will determine whether or not these new hardware components provide a sufficient amount of value in terms of security, accessibility, and user friendliness, or if we feel that the project would be better off without them. There will still be time to refine the UI and flows as we move through Phases 5 and 6, but if these components are to be eliminated we may need to redesign the PCB at this point to reduce complexity.

### Phase 5: Rapid Reiteration (2/15 - 3/7)

During this phase we will hopefully begin with a working prototype, then work to refine it. We will explore professional enclosure and PCB fabrication opportunities, balancing the need for a high TRL final product with the need for simple and reliable fabrication. Anything that we pursue should be possible to replicate for another space who wants to implement Portal Boxes. We will also finalize our code deployment during this stretch, creating the multi-tenant infrastructure that should allow for simple setup of the database and Makerportal for new spaces. With this move to the cloud, we will also look at automated maintenance flow and what we can do to simplify the continued operation of Portal Boxes. We will begin compiling our documentation (what will be provided to new adopters), ensuring completeness across all aspects of the project. We will create a checklist showing what is available for the V1, V2, V3, and our V4 to ensure that we encapsulate all relevant information in one location. At the end of Phase 5 we are heading into Spring Break, which will provide us with a natural pausing point to evaluate our project. As we leave for break, we should ensure that we have ordered the supplies or fabrication of our newest iteration such that we can get user feedback in Phase 6.

### 

### Phase 6: Final Convergence (3/17 - 4/11)

In this final phase of active development, we will converge on the highest TRL product possible and deliver it to potential adopters. We’ve been very tight with our spending so far and have benefitted from gifted supplies from Matt Lamparter, so we estimate to have room to build at least three additional Portal Boxes with our budget, and are confident in obtaining more funds if we need more deployments. We will make plans with adopters during the months leading up to this point, so that during this phase we can conduct experiments and get input about our new design. These user tests are better defined below in *Performance Metrics*. We aim to conduct our final acceptance test on April 1st, but we have the buffer up until the 11th in case we need more time for users to get back to us.

### Phase 7: Wrap Up (4/11 - End)

By this time of year we will be rapidly approaching graduation. Seeing our friends last year grappling with balancing the end of the year and finishing their projects, we’ve chosen to give ourselves a solid amount of time here. We will finalize our technical and organizational reports, our public documentation, and our deployment plan based on the user feedback and results of the acceptance test in Phase 6. We are aware that success is not certain, and if we fail the acceptance test this time will be spent reflecting on our failures and what went wrong. Ideally, this will be time for putting bows and ribbons on our project, improving small UI details, and packing it all up. In the following section we detail our initial plans for evaluating whether our project meets the acceptance criteria.

## Performance Metrics

### Colton (Design Engineer – Circuit Block)

#### **Power Relay:**

* **Metric:** A digital signal from the microcontroller should allow or block 120V power within 5 seconds of the signal being sent.
* **Test Method:** Measure relay activation time using a test jig.
* **Target:** ≤ 5 seconds.

#### **USB Interlock:**

* **Metric:** A digital signal from the microcontroller should allow or block data/clock connections via USB within 5 seconds of the signal being sent.
* **Test Method:** Test interlock functionality with a known USB connection.
* **Target:** ≤ 5 seconds.

#### **Current Sense:**

* **Metric:** The system should report current flow through the wall-to-machine connection with 90% accuracy.
* **Test Method:** Compare reported data against actual current measured with calibrated equipment.
* **Target:** ≥ 90% accuracy.

### Kevin (Design Engineer – Firmware Block)

#### **Execution Time:**

* **Metric:** Average time between state transitions in the finite state machine should range from 200 to 500 ms.
* **Test Method:** Use a timer module in MicroPython to measure state transition durations.
* **Target:** 200–500 ms.

#### **Extended Runtime Performance:**

* **Metric:** The Portal Box should operate continuously without failure for at least 7 days.
* **Test Method:** Monitor for errors during prolonged operation and log runtime until failure.
* **Target:** ≥ 7 days of continuous operation.

#### **Data Movement Accuracy and Stability:**

* **Metric:** Data should be accurately stored locally and transmitted to the database without errors, with proper functionality afterward.
* **Test Method:** Simulate user data input, verify local storage, and confirm database synchronization.
* **Target:** 100% accuracy and stability.

### Peter (Design Engineer – Enclosure Block)

#### **Enclosure Cost:**

* **Metric:** The cost of materials for the new enclosure should not exceed the cost of the current acrylic enclosure.
* **Test Method:** Compare the material cost for the redesigned enclosure against the current cost.
* **Target:** ≤ Current enclosure cost.

#### **User Opinion:**

* **Metric:** A majority of interviewed users must prefer the new enclosure design functionally and stylistically over the current design.
* **Test Method:** Conduct user interviews and surveys.
* **Target:** ≥ 51% positive preference.

#### **Ease of Assembly:**

* **Metric:** The redesigned enclosure should be as easy or easier to assemble than the current enclosure.
* **Test Method:** Time assembly processes and survey MakerE users assembling model enclosures.
* **Target:** Assembly time equal to or less than the current design, with high usability ratings.

### Aidan (Design Engineer –Keypad & RFID Scanner)

#### **RFID Scanner:**

* **Metric:** The RFID Card should display the same UID no matter what scanner it is being read on
* **Test Method:** Compare the reading of the RFID Cards on two different scanners
* **Target:** 100% Accuracy

#### **Keypad:**

* **Metric:** Users of the current Portal Box would like another way to verify the user of the card is the actual person who the card is assigned to, while making verification fast still
* **Test Method:** Test response time of the code being administered, entered and given access
* **Target:** Fast and reliable access to the Portal Box and machine

#### **Backend Verification:**

* **Metric:** Once the UID of an authorized person is stored locally it should be uploaded to the database.
* **Test Method:** Ensure that the UID is uploaded to the database with the correct information of the card and the User.
* **Target:** Information is stored locally and uploaded to the database within just a few minutes, with 100% correct information.

### James (Systems and Performance Engineer)

#### **Power Consumption:**

* **Metric:** The Portal Box should draw less than 1.2A (current draw of the Raspberry Pi-based system).
* **Test Method:** Measure current draw with a multimeter or current-sense IC.
* **Target:** < 1.2A at 5V.

#### **Fault Condition Test:**

* **Metric:** The system must turn off power, display a warning, and log an uncertified card’s usage in the database during misuse.
* **Test Method:** Simulate an uncertified card input during machine operation and verify system responses.
* **Target:** Accurate response in all cases.

#### **Non-Ideal Inputs:**

* **Metric:** Power should only be supplied when a certified card and correct pin are entered; all other cases should block power.
* **Test Method:** Simulate certified and uncertified cards with correct and incorrect PINs, measuring voltage output.
* **Target:** 100% correctness in all input scenarios.

### Axel (Project Manager)

#### **Documentation:**

* **Clarity and Usability:**
  + **Metric:** Average usability rating of 4.0 or higher on a 1–5 scale.
  + **Test Method:** Provide documentation to testers and have them set up the system independently.
  + **Target:** ≥ 4.0 rating.
* **Completeness:**
  + **Metric:** 100% of required topics (setup, maintenance, troubleshooting) covered.
  + **Test Method:** Cross-check documentation against a predefined checklist.
  + **Target:** 100% completeness.

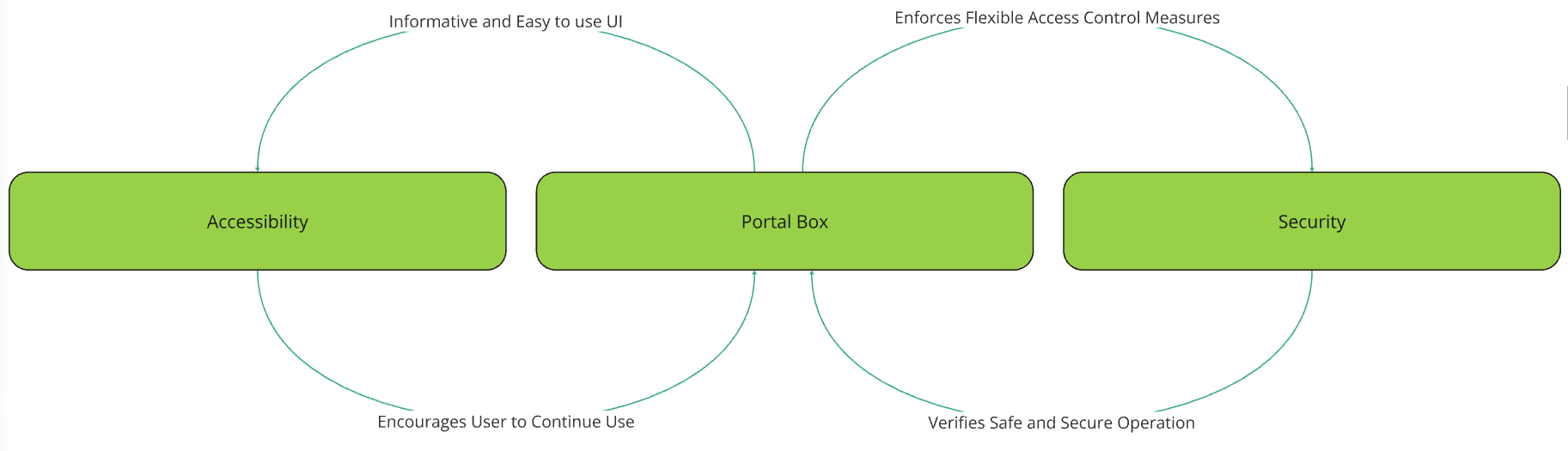
#### **Code Deployment:**

* **Ease of Deployment:**
  + **Metric:** Average setup time using Docker containers should be less than 15 minutes.
  + **Test Method:** Time independent users deploying the system.
  + **Target:** ≤ 15 minutes.
* **Update Difficulty:**
  + **Metric:** Time required to apply firmware updates and resume normal function should be under 15 minutes.
  + **Test Method:** Simulate a firmware update and measure recovery time.
  + **Target:** ≤ 15 minutes.

#### **Analytics and Displays:**

* **Dashboard:**
  + **Metric:** Average usefulness rating and user inclination rating ≥ 4.0 on a 1–5 scale.
  + **Test Method:** Present dashboard to users for evaluation.
  + **Target:** ≥ 4.0 average.
* **On-Box UI:**
  + **Metric:** Average readability, ease-of-use, and improvement ratings ≥ 4.0 on a 1–5 scale.
  + **Test Method:** Deploy prototypes and gather user feedback via surveys.
  + **Target:** ≥ 4.0 average.

### Reasoning Behind Metrics

The purpose of these performance metrics is to measure and verify the Portal Box system will reach a high technology readiness level (TRL) and achieve the flow outlined in the flow diagram.[[3]](#footnote-3) The current Portal Box system is actively deployed at the Bucknell Maker-E and other university spaces, and it is a working system that is used daily in a real world setting. In order for our project to be deemed successful, the V4 Portal Box must meet a high TRL. It must meet these defined performance metrics to validate the system’s readiness and capability to operate reliably in any implementation environment and be more efficient than its predecessors. Not only is a high TRL needed for the ability to implement Portal Boxes into a space, but to address the negative reinforcing loop we identified between accessibility and security in Makerspaces. The Portal Box is a dynamic system that is easily configurable by administrators to fit their access control needs, and a dynamic system is the key to alleviating strains between accessibility and security in these spaces [18]. This separates security and accessibility from a spectrum, where one benefits as the other is hurt, into individual elements that overlap with each other. The Portal Box intervenes to create a space in which there is both accessibility, through an easy to use interface that encourages a user to return to the space, and security, that the Portal Box provides and verifies those measures work. This creates two separate positive relationships that both benefit without hurting each other as shown in *Figure 4.2*. To implement this change effectively, the Portal Box must achieve a high TRL. Reaching a high TRL demonstrates the Portal Box is practical for widespread adoption to address the critical balance between accessibility and security in makerspaces.

*Figure 4.2: Portal Box Effect on Accessibility & Security*

# **Bibliography**

[1] “How MIT’s fab labs scaled around the world,” MIT News | Massachusetts Institute of Technology, Jun. 05, 2023. https://news.mit.edu/2023/how-mits-fab-labs-scaled-around-world-0605

[2] C. Cahoon and K. Furlong, “Interview: L&IT and Makerspaces,” Sep. 16, 2024.

[3] E. L. T. Davies, R. D. Morris, and A. S. Jariwala, “Trust as the Foundation for a Successful Balance of Power in a Student Run Academic Makerspace”.

[4] G. Rokavek, “Interview: Risk Management,” Sep. 27, 2024.

[5] M. Lamparter, “Interview: Maker-E,” Sep. 11, 2024.

[6] T. Derk, “STEAM Site Visit,” Nov. 15, 2024.

[7] J. Wayland, “Interview: QMUL Makerspace,” Sep. 12, 2024.

[8] R. Wetzel, “Penn State Makerspace Visit,” Sep. 19, 2024.

[9] E. Jablonski, “Interview: Center for Entrepreneurship and Innovation,” Sep. 26, 2024.

[10] A. Clark, “Interview: Mooney Innovation Lab,” Sep.12, 2024.

[11] QMUL, School of Engineering & Material Science, “Access to Makerspace Checklist,”

[12] “Bucknell Makerspace User Agreement,” Google Docs. Accessed: Sep. 23, 2024. [Online].

[13] J. Hale, “Interview: 2016 Portal Box,” Nov. 19, 2024.

[14] A. Sellers, “Interview: 2016 Portal Box,” Sep. 26, 2024.

[15] J. Hass, “Interview: Firmware Owner,” Oct. 22, 2024.

[16] B. Buentello, “Interview: V3 Portal Box,” Sep. 4, 2024.

[17] J. Nickel, “Interview: BME Labs,” Oct. 7, 2024.

[18] [J. Logas, R. Zhong, S. Almeida, and S. Das, “Tensions between Access and Control in Makerspaces,” Proc. ACM Hum.-Comput. Interact., vol. 4, no. CSCW3, pp. 1–33, Jan. 2021, doi: 10.1145/3432914.](https://www.zotero.org/google-docs/?9s8JSU)

[19] Z. Li, H. Qi, Z. Sun, C. Shi, and D. Zhu, “[1 point] Research on an Open and Shared Maker Space and Curriculum System,” in *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, Opatija, Croatia: IEEE, May 2024, pp. 1234–1237. doi: [10.1109/MIPRO60963.2024.10569945](https://doi.org/10.1109/MIPRO60963.2024.10569945).

[20] Espressif, “ESP32 Series Datasheet Including,” 2024. Available: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>

[21] TechTarget, “What is two-factor authentication (2FA)?,” TechTarget, 2024. [Online]. Available: <https://www.techtarget.com/searchsecurity/definition/two-factor-authentication>.

[22] M. Blank and A. Lehman, “How secure electronic authentication mitigates risk at the point of care,” Analog Devices, vol. 57, no. 1, pp. 1–7, 2024. [Online]. Available: <https://www.analog.com/media/en/analog-dialogue/volume-57/number-1/how-secure-electronic-authentication-mitigates-risk-at-the-point-of-care.pdf>.

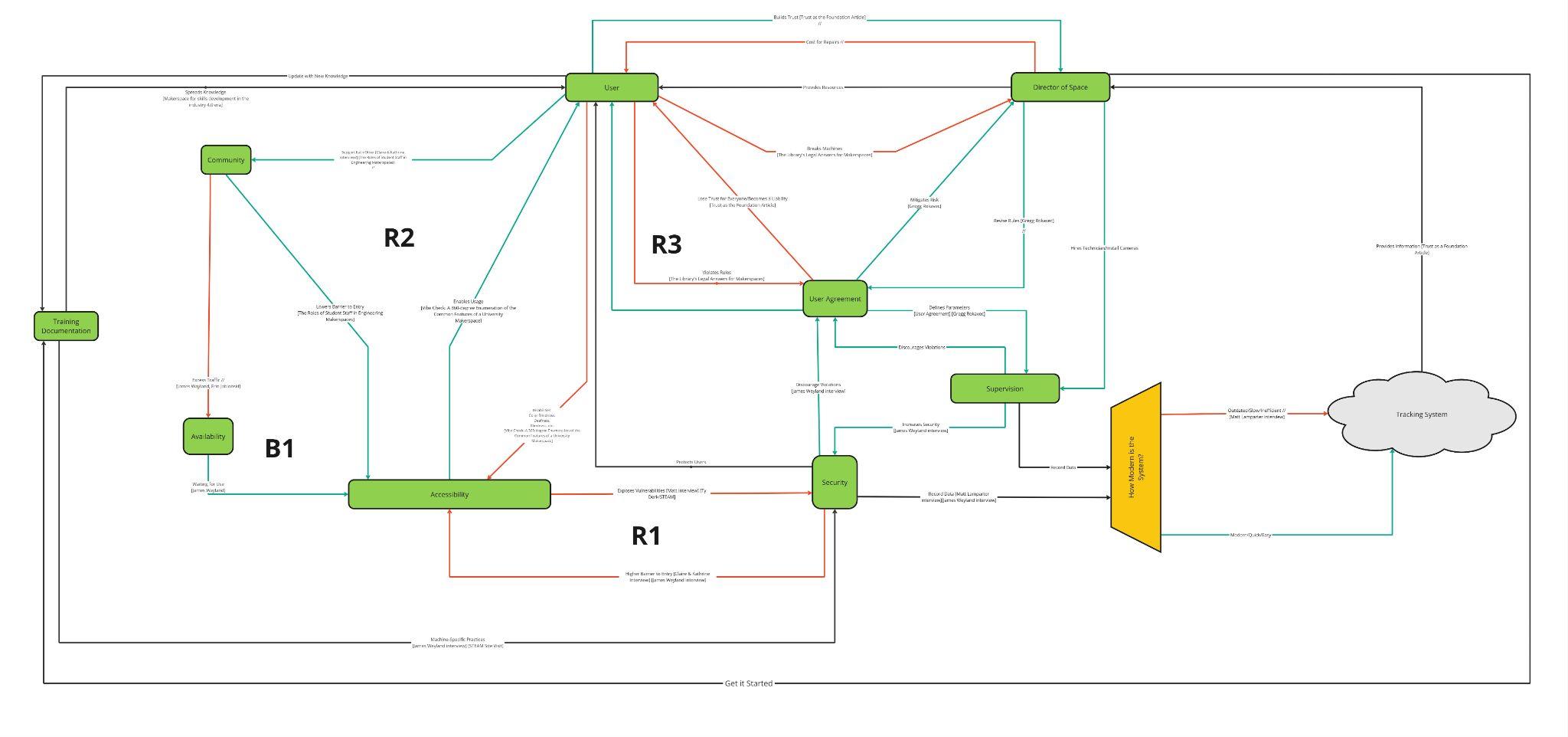
[23] “Adafruit eInk Display Breakouts and FeatherWings Created by lady ada.” Accessed: Dec. 10, 2024. [Online]. Available: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-eink-display-breakouts.pdf

[24] “HD44780U (LCD-II), (Dot Matrix Liquid Crystal Display Controller/Driver)”.

# **Appendices**

### Appendix A: System Map

The system map shown below is our representation of the relationships between many interconnected components within the realm of Makerspaces and spaces that have elements of access control within them. Highlighted in this system map is how elements within these spaces relate to access and security for their users. Within our system map there are loops identified as R and B symbolizing reinforcing and balancing respectively. We feel these are important loops within the system map and change or are affected by the implementation of the Portal Box system within these spaces. As a whole the system map helps guide why the Portal Box is important and adds value in a larger perspective.



URL to system map in Miro: <https://miro.com/app/board/uXjVL9IOtOY=/?moveToWidget=3458764606846789561&cot=14>

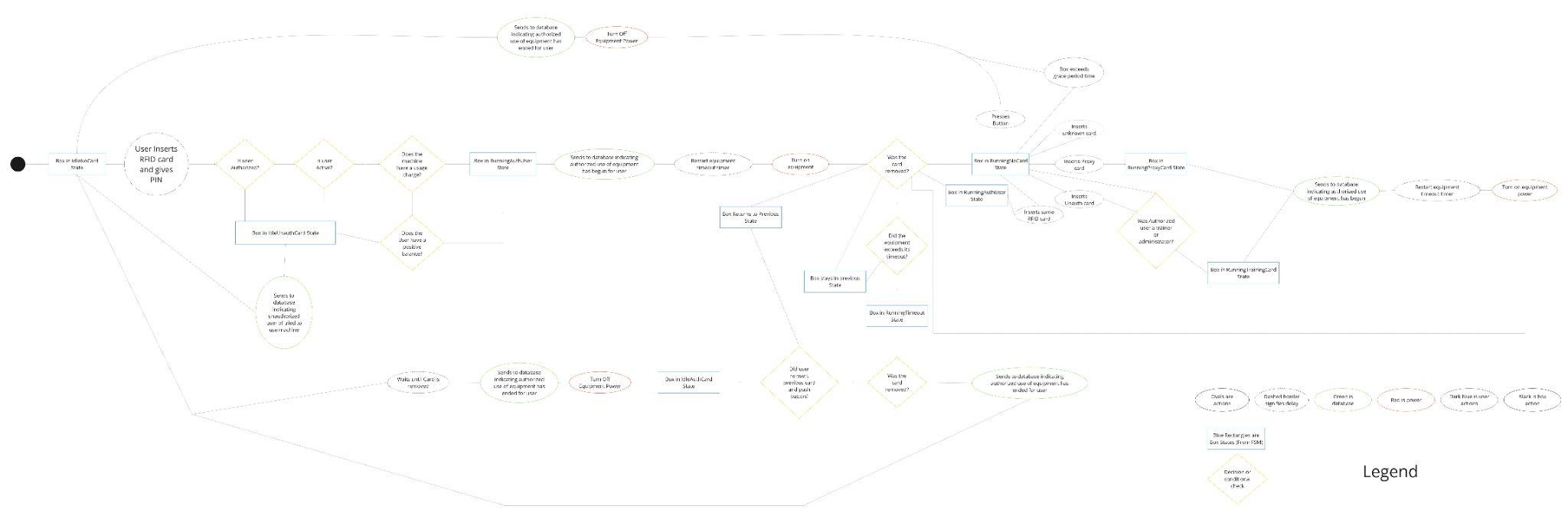
### Appendix B: Gantt Chart

Linked here is the working copy of our Gantt chart. I chose to use SmartSheet to build and maintain the chart because of its advantages and ease-of-use when compared to Google Sheets or Microsoft Excel. Phases 4A & 4B are fleshed out, while start and end dates are given for later phases. This chart will continue to grow and develop as we progress through the year, and it will be reviewed with the team as we approach each deadline.

URL: <https://app.smartsheet.com/sheets/MqQXcG5wjWhWVM2vf2qxx46qJF4MjXP7mw5mWGj1?view=gantt&timelineGroupBy=3010581315284>

### Appendix C: Flow Diagram

The flow diagram below represents the operational life cycle of the Portal Box. It forms a continuous flow that begins with the Portal Box in its idle state, transitions through its active usage states and finishes when the Portal Box use ends, setting it back into an idle state. The diagram illustrates the software states of the Portal Box as it is in use. It is a diagram that has guided our design and consideration of hardware components and software design. The ability for the Portal Box to successfully go through this flow has also been considered for our performance metrics. The ability of Portal Box to go through this flow without error is critical for ensuring secure and safe machine access in any space the Portal Box is implemented.



URL to flow diagram on Miro: <https://miro.com/app/board/uXjVL9IOtOY=/?moveToWidget=3458764605676973124&cot=14>

Appendices

Bibliography

1. Appendix A: System Map [↑](#footnote-ref-1)
2. Appendix B: Gantt Chart [↑](#footnote-ref-2)
3. Appendix C: Flow Diagram [↑](#footnote-ref-3)