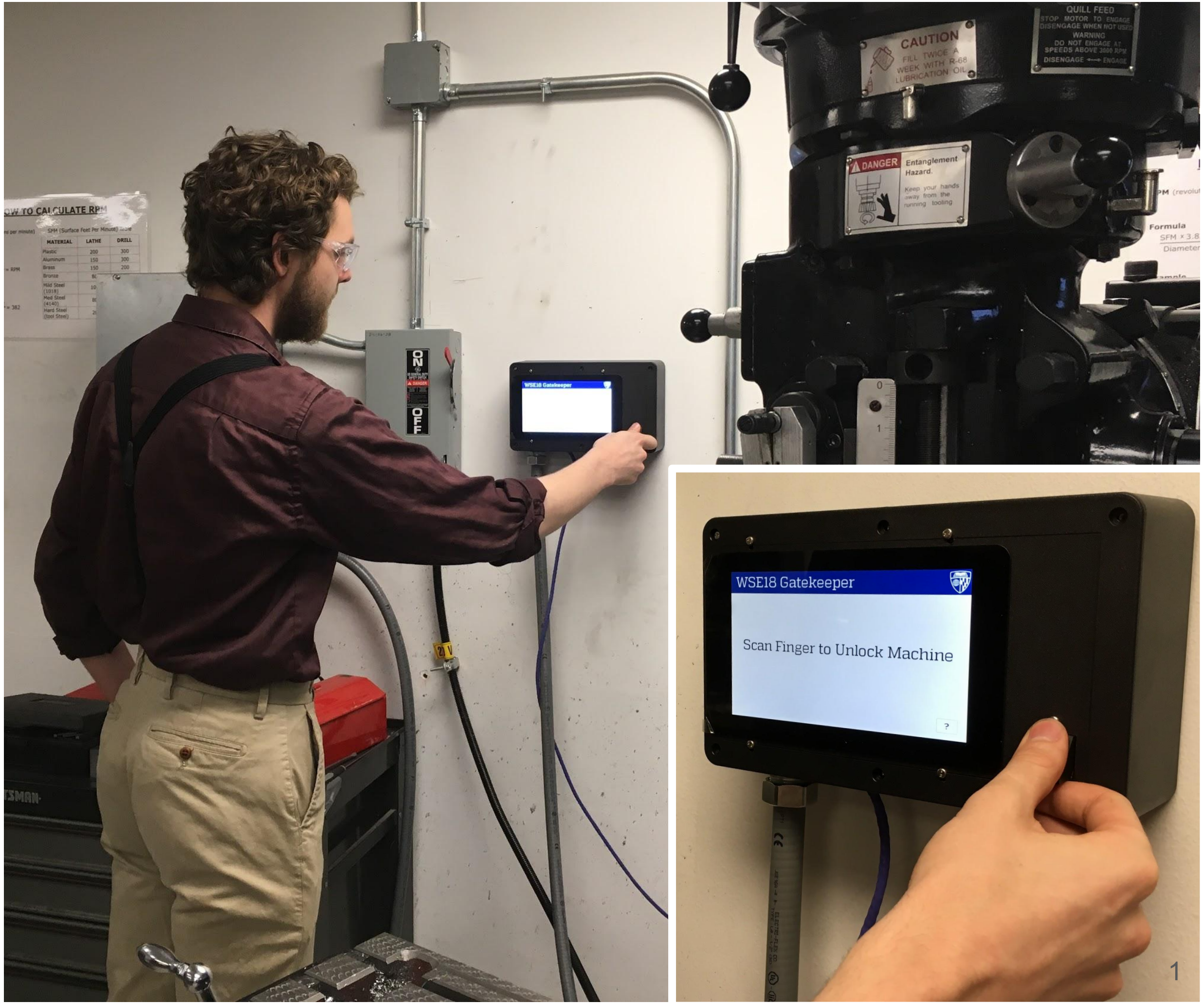
WSE18

Enhancing the Self Service Shop experience

Geordan Gutow
ggutow1@jhu.edu
920-376-1608

David Samson
dsamson4@jhu.edu
267-221-3054

Stephane Teste
steste1@jhu.edu
248-376-7922



Sponsors and Support Structure

	Name	Email	Phone
Sponsors	Rich Mejia	rmejia2@jhu.edu	(410) 516-7146
	Rich Middlestadt	rmiddle4@jhu.edu	(410) 516-7710
	Cynthia Larichiuta	clarich1@jhu.edu	(410) 516-3937
WSE IT	Colleen Cusimano	colleen.cusimano@jhu.edu	(667) 208-6860
	Shawn Suter	ssuter3@jhu.edu	(410) 516-7443
JHU ISI	Joe Carrigan	joseph.carrigan@jhu.edu	(410) 516-6454
Junior Assistant	Kimberly Koon	kkoon1@jhu.edu	(631) 689-7524

The following contains intellectual property owned by the WSE18 team. By accessing this document, you agree not to use information gained from it in ways that would harm the IP rights of the team. In particular, you may not use information gained from this document to reproduce the described system, or to develop one substantially similar to it, unless you are granted an explicit separate license to do so.

Table of Contents

Section 1: Executive Summary

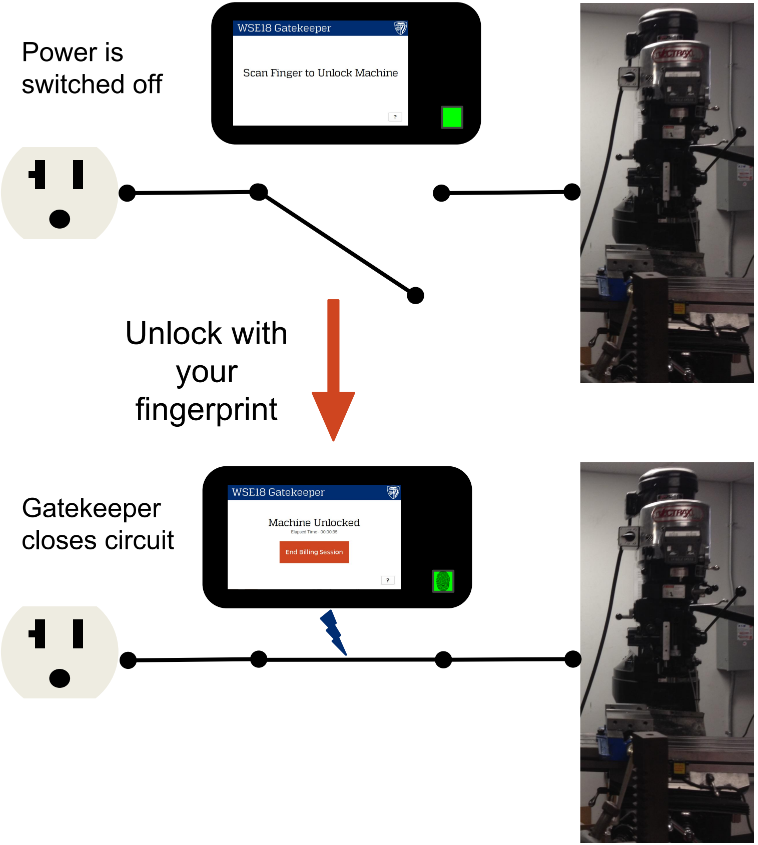
Motivation

The WSE self-service shop possesses equipment requiring training before use. Currently, training is enforced solely by the honor system: a student is granted access to the room upon completion of basic shop training, and it is expected that they will not use the mills, lathes, or EDM until they have proper training. This system has been abused by students in the past; Shop Manager Rich Mejia estimates that in a typical year, machines are used on the order of a hundred times by users who are not trained or who fail to log their usage.

Problem Statement

In an effort to **make the shop safer** and improve tracking of shop usage, team WSE18 is tasked with implementing a **tamper-evident system** that **prevents users from activating equipment they are not trained on**. The system is also expected to reduce the manual effort necessary for billing by automatically tracking equipment usage.

Solution: The Gatekeeper™


Power is switched off

Unlock with your fingerprint

Gatekeeper closes circuit

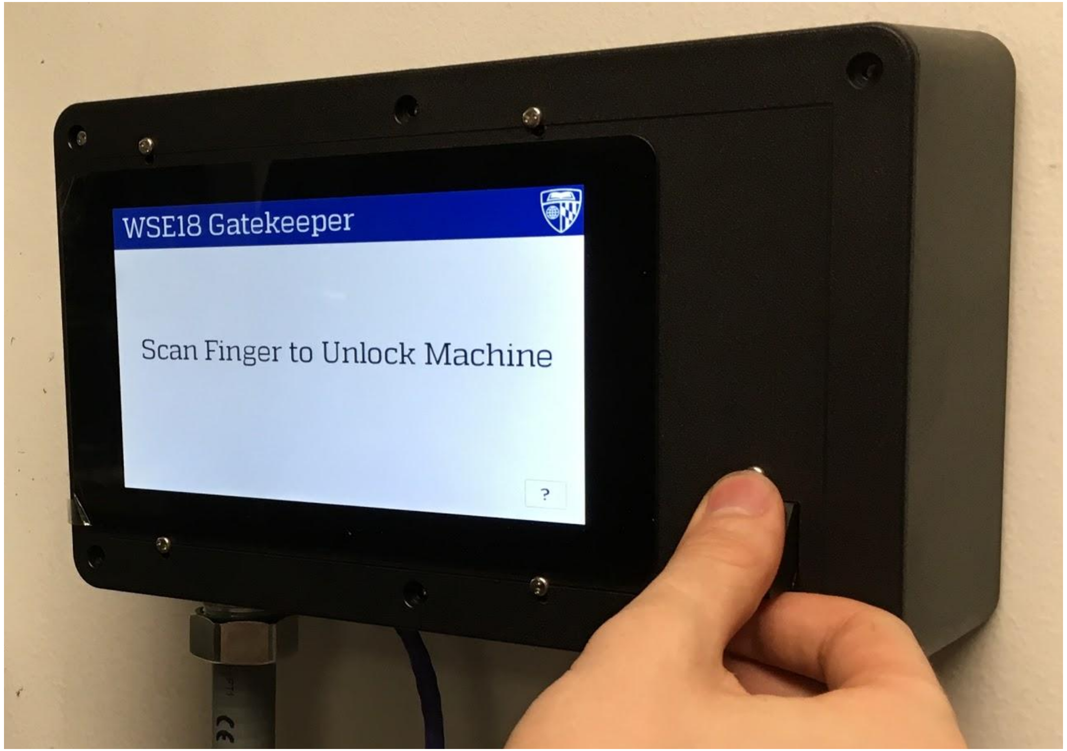
An interlock device (The Gatekeeper) goes on each machine. Each interlock has a fingerprint sensor for identifying users, and has the ability to interrupt power to the equipment until a trained user authenticates

A server maintains a database of users, training records, budget records, and fingerprints

A kiosk allows managing that database through an intuitive graphical user interface

Using the Gatekeeper

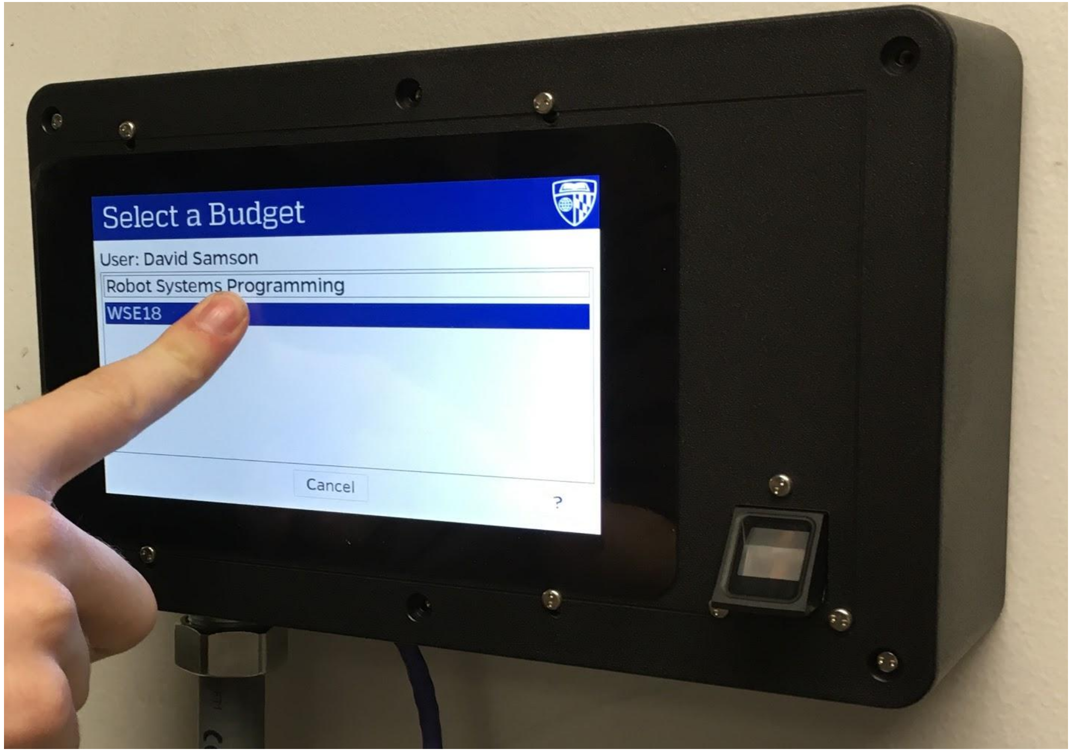
1.



Present your finger.

Using the Gatekeeper

2.



Select a Budget

User: David Samson

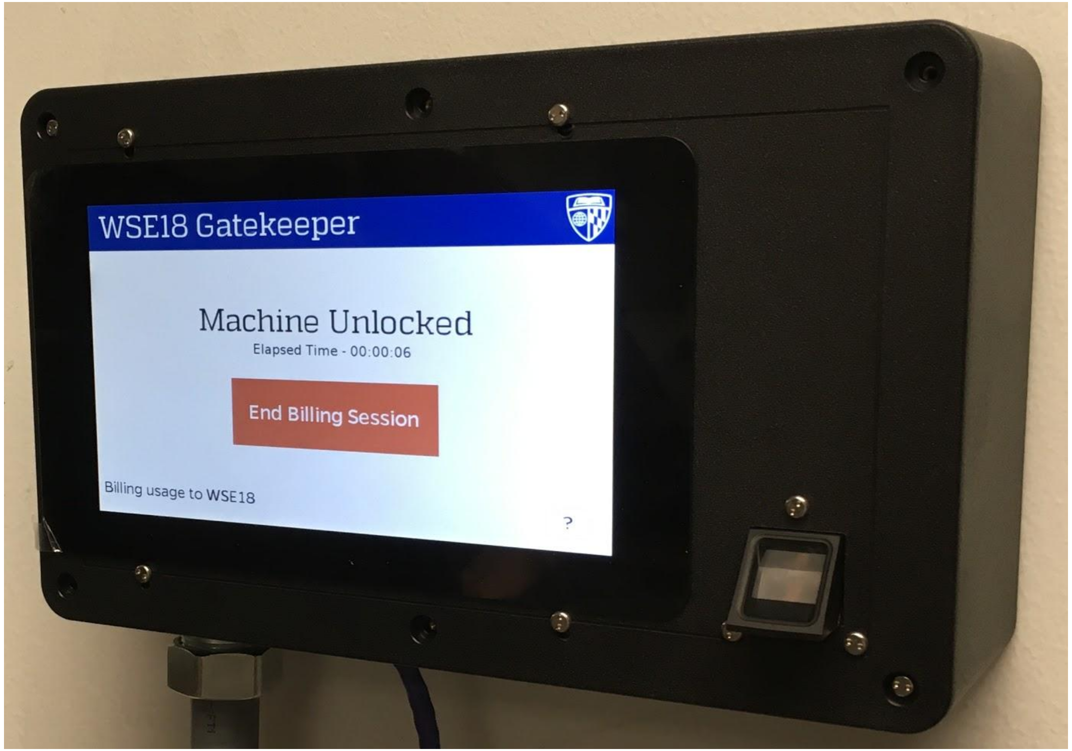
Robot Systems Programming

WSE18

If trained, pick the budget to charge.

Using the Gatekeeper

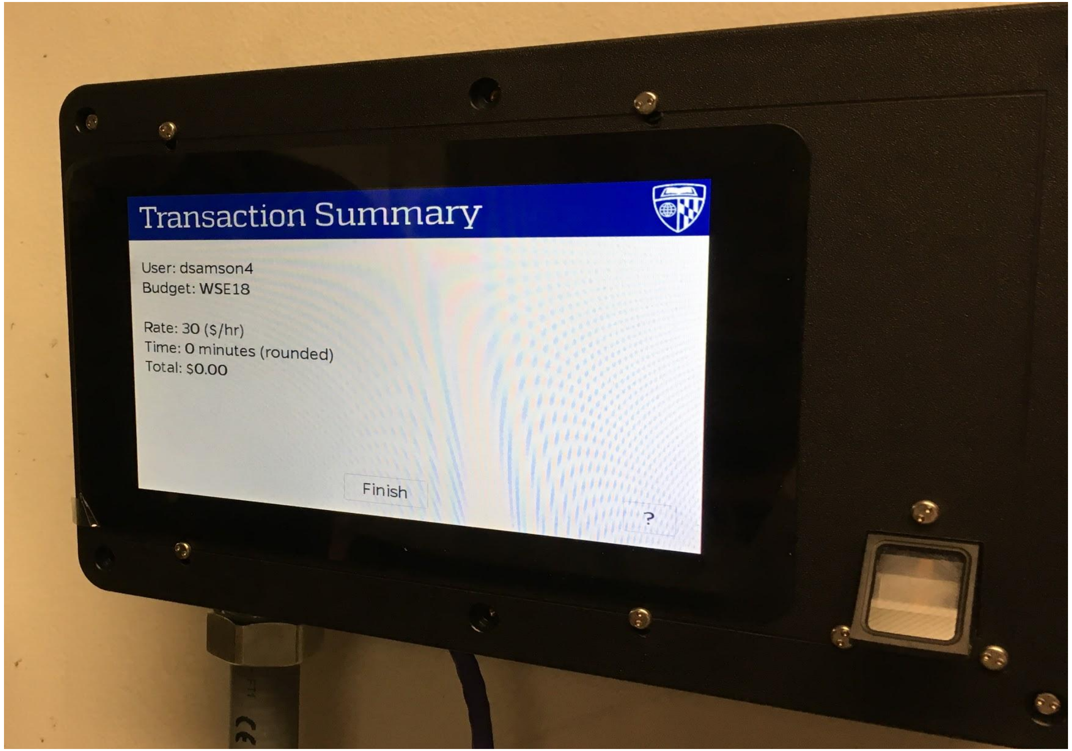
3.



Use the machine.

Using the Gatekeeper

4.



Transaction Summary

User: dsamson4
Budget: WSE18

Rate: 30 ($/hr)
Time: 0 minutes (rounded)
Total: $0.00

A summary is displayed when finished.

Project Requirements		High Level Test Plan	Status
1	prevent untrained users from activating machinery	In-house testing with unauthorized prints, live user testing	1 user of 50 was misidentified once; raised required confidence in matches to avoid in the future.
2	be difficult for a determined user to bypass	Penetration testing by Information Security Institute, live user testing	Penetration tests found no exploitable holes; two unexploitable vulnerabilities found
3	be tamper-evident	Penetration testing, Live user testing	No explicit test conducted, but system detected some tampering attempts
4	aid in billing	Live user testing	Automatically billed over 100 hours of machine usage
5	be minimally intrusive to the user’s ability to utilize the shop efficiently	Live user testing	5 weeks of testing completed; user feedback positive apart from sensor responsiveness
6	require minimal maintenance	Accelerated life-cycle testing of hardware	Contactor survived 10,000+ cycles; one fingerprint sensor failed after 4 weeks of testing
7	be expandable to additional machines as the shop grows	Stakeholders setup a machine using only documentation	Assembly successful on 5/3/18
8	not violate relevant electrical and safety codes	Expert evaluation from electricians	Electricians completed install
9	not disconnect power to a machine while in use	In house testing and live user testing	No incidents during testing
10	not cost more than $10,000 to install in student shop	Cost tracking	Component cost: $427.49

Section 2: Problem Definition

Original Problem Statement

The WSE18 project aims to install tamper-resistant biometric authentication and iLab in the self-service machine shop.
The main objective is to, first and foremost, ensure the safety of students using the machine shop, then to streamline the billing process.

What is iLab?

iLab is a commercial software product sold by Agilent. It provides tracking of user training on equipment and has the capacity to enable/disable machines via a small set of commercially available controlled outlet devices, like the one to the right.

iLab was removed from the project in September, as it lacked the functionality necessary to implement a biometric authentication system. A revised problem statement, presented on the next page, was used to guide our work from that point onwards.

Revised Problem Statement

In an effort to **make the shop safer** and improve tracking of shop usage, team WSE18 is tasked with implementing a **tamper-evident system** that **prevents users from activating equipment they are not trained on**. The system is also expected to reduce the manual effort necessary for billing by automatically tracking equipment usage.

Requirements Definition

Requirements were developed by engaging with the project stakeholders and sponsors. We formally met with the clients (Rich Mejia and Richard Middlestadt, the managers of the Self-Service Shop, and Cindy Larichiuta, the shop’s finance guru) every two weeks, as well as on an ad-hoc basis since they were on-site. These meetings showed five primary needs for the solution we developed.

1. Tamper-Evidence

The solution we present needs to be tamper-evident. It is okay if a very determined user can circumvent the interlocking system as long as this causes detectable damage.

“I want to know if they tampered.”-Rich Middlestadt

2. Billing Effort Reduction

It is highly desirable that the system make billing easier than it currently is.

The current process is:

1. User writes down which machine they used, how long they used it, and which budget to charge in a paper log.
2. Periodically, Cindy retrieves the log and manually enters budgets and charges into the KB15N spreadsheet.
3. This spreadsheet is then submitted to the JHU Interentity accounting department.
4. Interentity enters the charges into the university's SAP finance system

3. User Experience

System should not interfere too much with shop usage. The process of logging into a machine and using it needs to be intuitive and fast, so that the efficiency of the shop is not impaired. Furthermore, Rich Mejia indicated that, since multiple users get trained at one time, the process for enrolling new users needs to be smooth and fast in order to not waste time.

4. Longevity

The solution we present needs to actually *work* indefinitely. This is something that will be sitting in the machine shop, controlling access, potentially for decades. It must do so long after we and everyone associated with the project are gone. Thus, the system needs to be able to operate for extended periods of time with limited maintenance by non-experts.

5. Extendable

The project brief is only for installing the system on the mills, lathes, and EDM in the current Self-Service Shop. However, if the shop ever expands in the future, the solution we present must be able to be easily installed on new machines without requiring an expert on-site to handle the installation and setup.

Formal Design Requirements

Functional

1. Solution shall prevent untrained users from activating machinery
2. Solution shall be difficult for a determined user to bypass

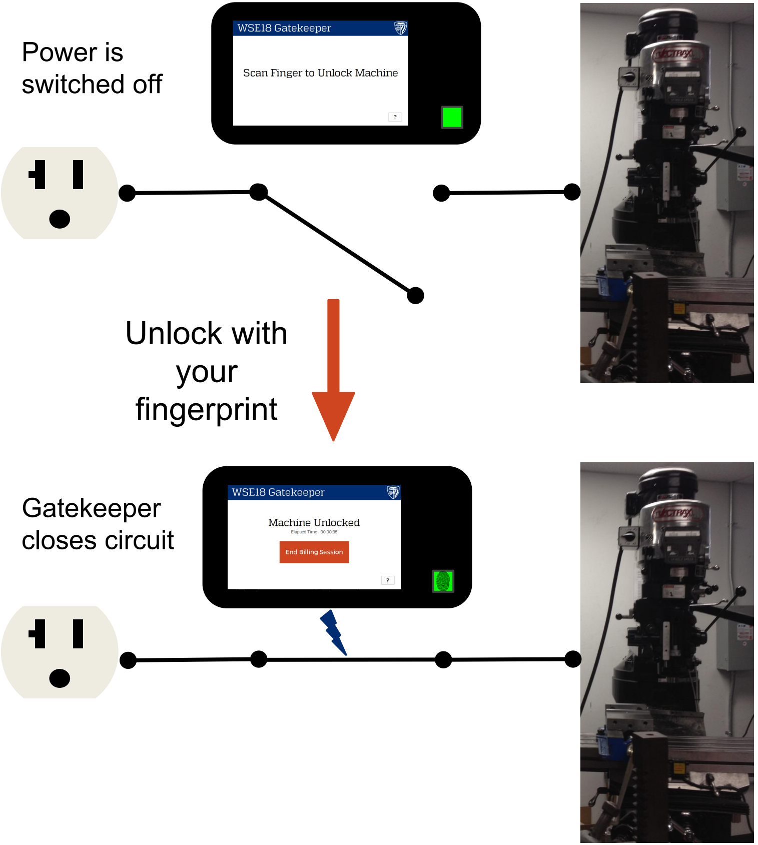
Stakeholder Needs

3. Solution shall be tamper-evident
4. Solution shall aid in billing
5. Solution shall be minimally intrusive to the user’s ability to utilize the shop
6. Solution shall require minimal maintenance
7. Solution shall be expandable to additional machines as the shop grows

Formal Constraints

1. Solution shall not violate relevant electrical and safety codes (specifically, the National Electric Code)
2. Solution shall not disconnect power to a machine while in use
3. Full project, including the installation of five Gatekeepers and a single Kiosk, shall not cost more than $10,000

Solution: The Gatekeeper™



Power is switched off

Unlock with your fingerprint

Gatekeeper closes circuit

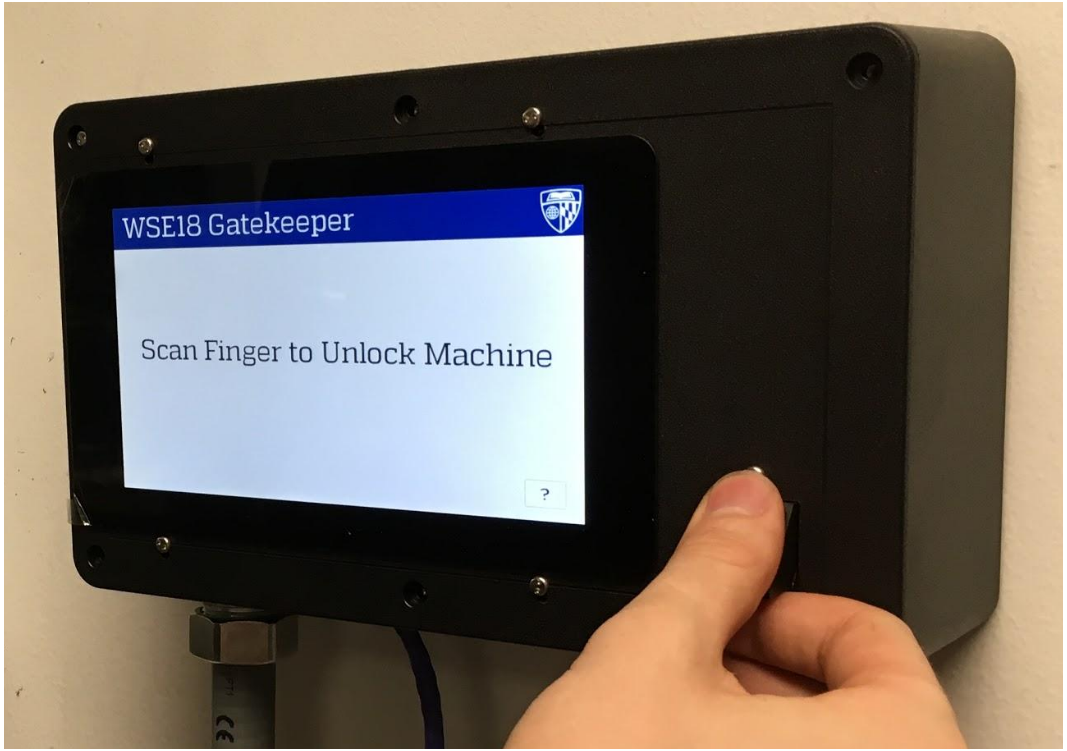
An interlock device (The Gatekeeper) goes on each machine. Each interlock has a fingerprint sensor for identifying users, and has the ability to interrupt power to the equipment until a trained user authenticates

A server maintains a database of users, training records, budget records, and fingerprints

A kiosk allows managing that database through an intuitive graphical user interface

Using the Gatekeeper

1.



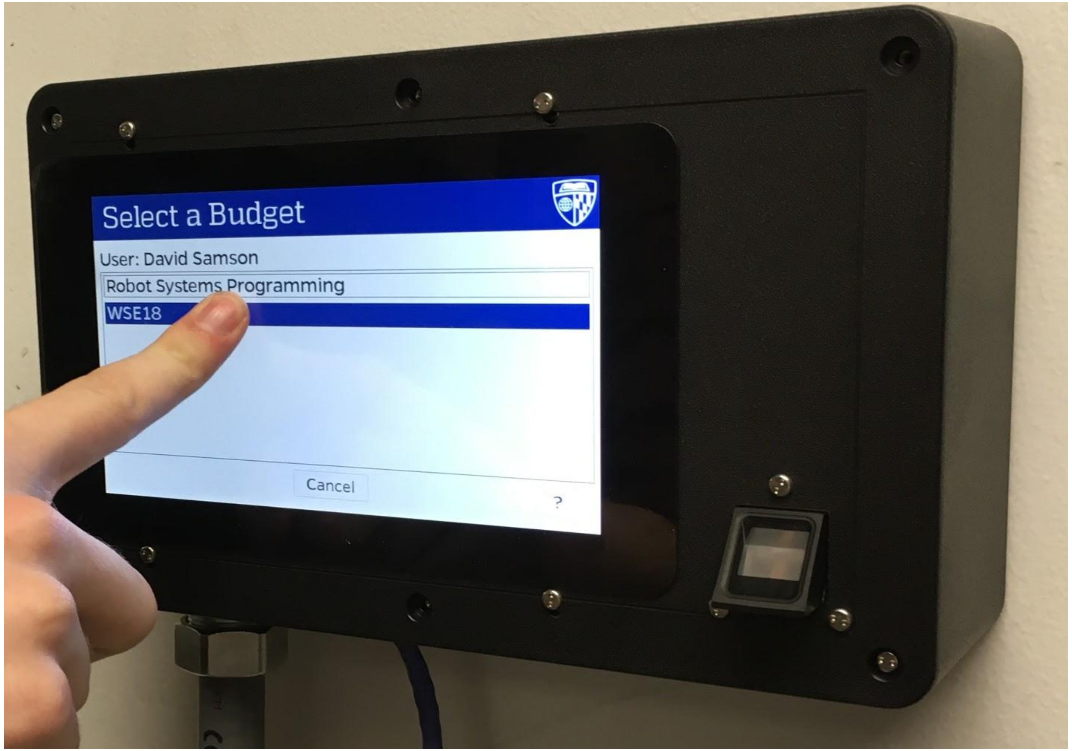
WSE18 Gatekeeper

Scan Finger to Unlock Machine

Present your finger.

Using the Gatekeeper

2.



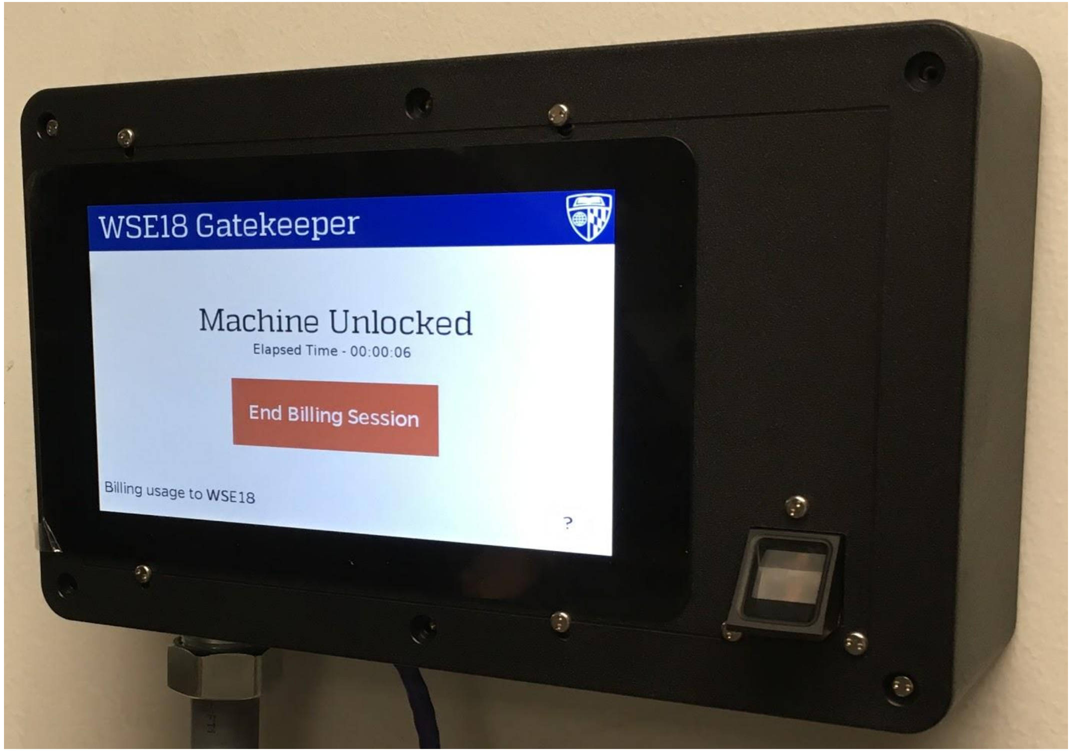
Select a Budget

User: David Samson

If trained, pick the budget to charge.

Using the Gatekeeper

3.


WSE18 Gatekeeper

Machine Unlocked

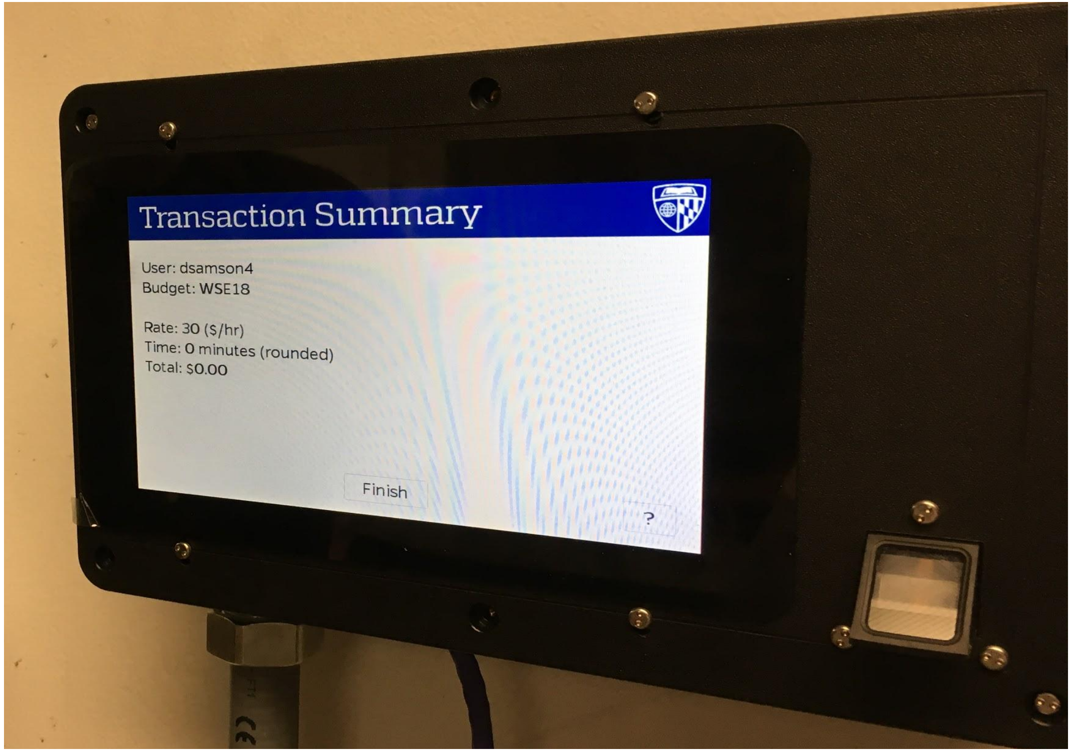
Elapsed Time - 00:00:06

Billing usage to WSE18

Use the machine.

Using the Gatekeeper

4.



Transaction Summary

User: dsamson4
Budget: WSE18

Rate: 30 ($/hr)
Time: 0 minutes (rounded)
Total: $0.00

A summary is displayed when finished.

Project Requirements		Status
1	prevent untrained users from activating machinery	one misidentification occurred during testing. The identification threshold was raised and no misidentifications have occurred since
2	be difficult for a determined user to bypass	bypasses require special tools and modifying high-voltage electronics
3	be tamper-evident	all known tampering approaches are monitored
4	aid in billing	automatically generates billing documents
5	be minimally intrusive to the user’s ability to utilize the shop efficiently	shop usage is not practically impaired, but users express annoyance at system responsiveness.
6	require minimal maintenance	one sensor failed during testing; no other maintenance issues are known
7	be expandable to additional machines as the shop grows	sponsors have demonstrated ability to assemble new devices based on the documentation provided
8	not violate relevant electrical and safety codes	JHU electricians approved and installed electronics
9	not disconnect power to a machine while in use	during 5 weeks of testing, this never happened
10	not cost more than $10,000 to install in student shop	components cost $430/device; installation labor < $500

Section 3: Solution Development

Brainstorming: Authentication Schemes

	Something you have (e.g. JCard)	Something you know (e.g. password)	Something you are (passive) (e.g. gait, or facial scan)	Something you are (active) (e.g. fingerprint)
Pros	● “100%” reliable ● Everyone already has a JCard ● Already need card out to get into shop->not disruptive	● Cheap ● “100% reliable” ● Might be less likely to share	● User doesn’t have to do anything ● Can’t give away ● Always have	● Can’t give away ● Always have
Cons	● Can give away to untrained user ● Can lose ● Expensive if not JCard ● Reader may be expensive	● Can give away ● Easy to forget ● Annoying to enter	● Accuracy ● Unintentionally unlocking machine ● Expensive cameras/sensors/software ● Privacy concerns ● Speed of identifying a user	● Accuracy ● Privacy ● Annoying to have to provide

Brainstorming: Authentication Schemes

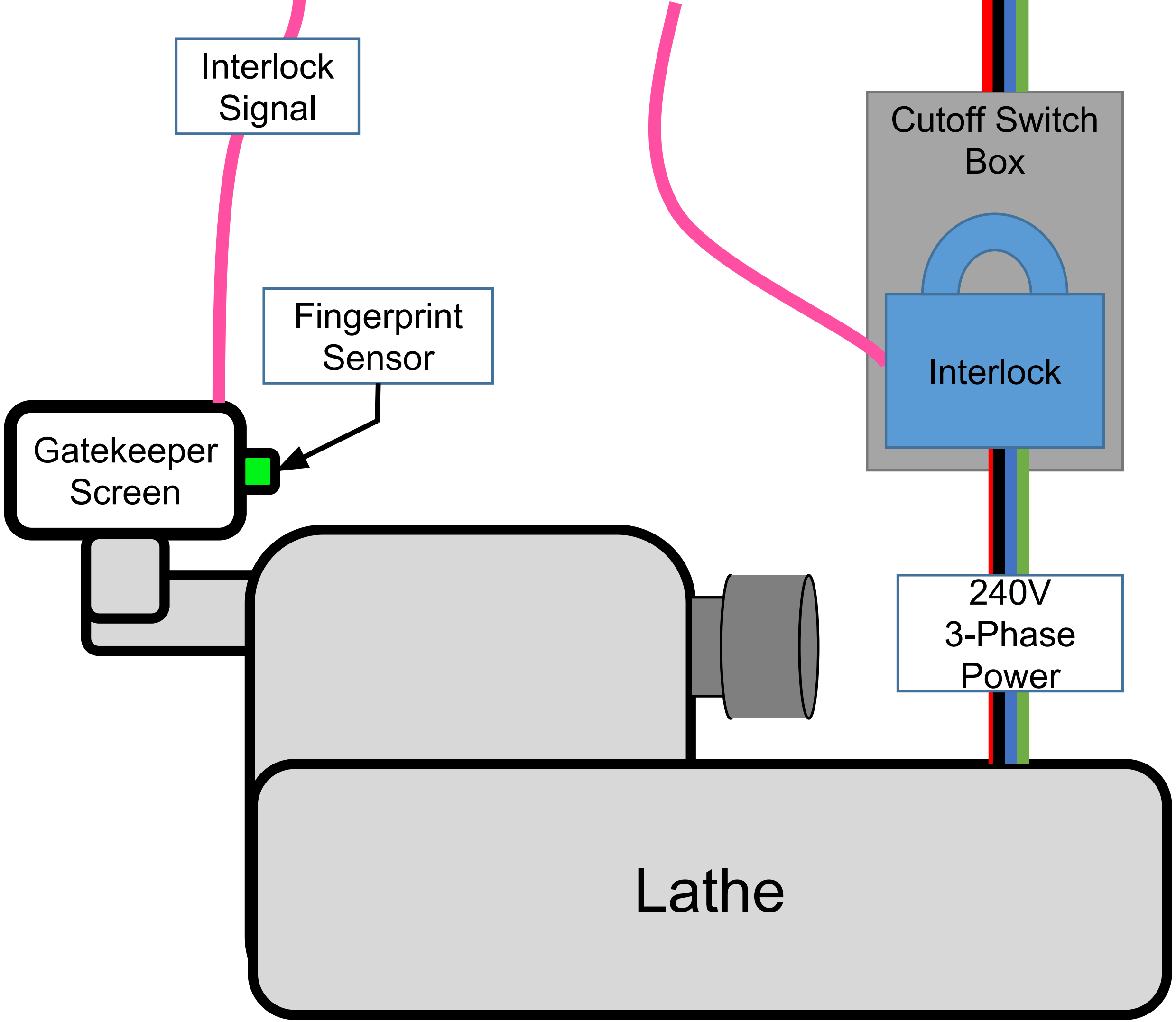
The first step was to decide how users would be authenticated.

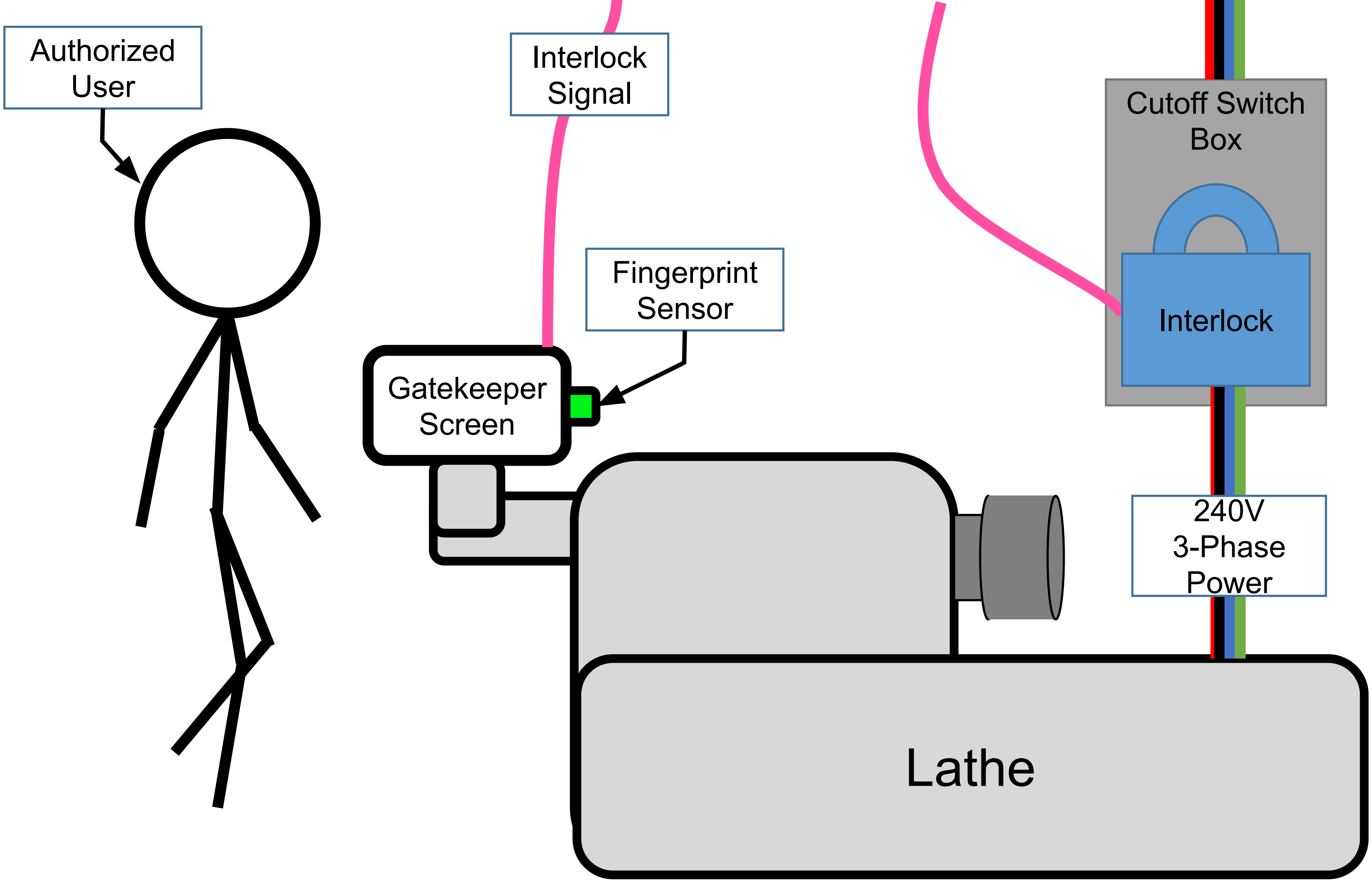
Broadly speaking, means of authentication can be divided into three schemes that rely on 1) something the user has (like a key), 2) something the user knows (like a password), 3) or something the user is (biometrics, like a fingerprint). For our purposes, the ability of a solution to correctly identify individuals (reliability), even in the event of collusion between users (security), must be balanced with the level at which a system interferes with shop usage (convenience)

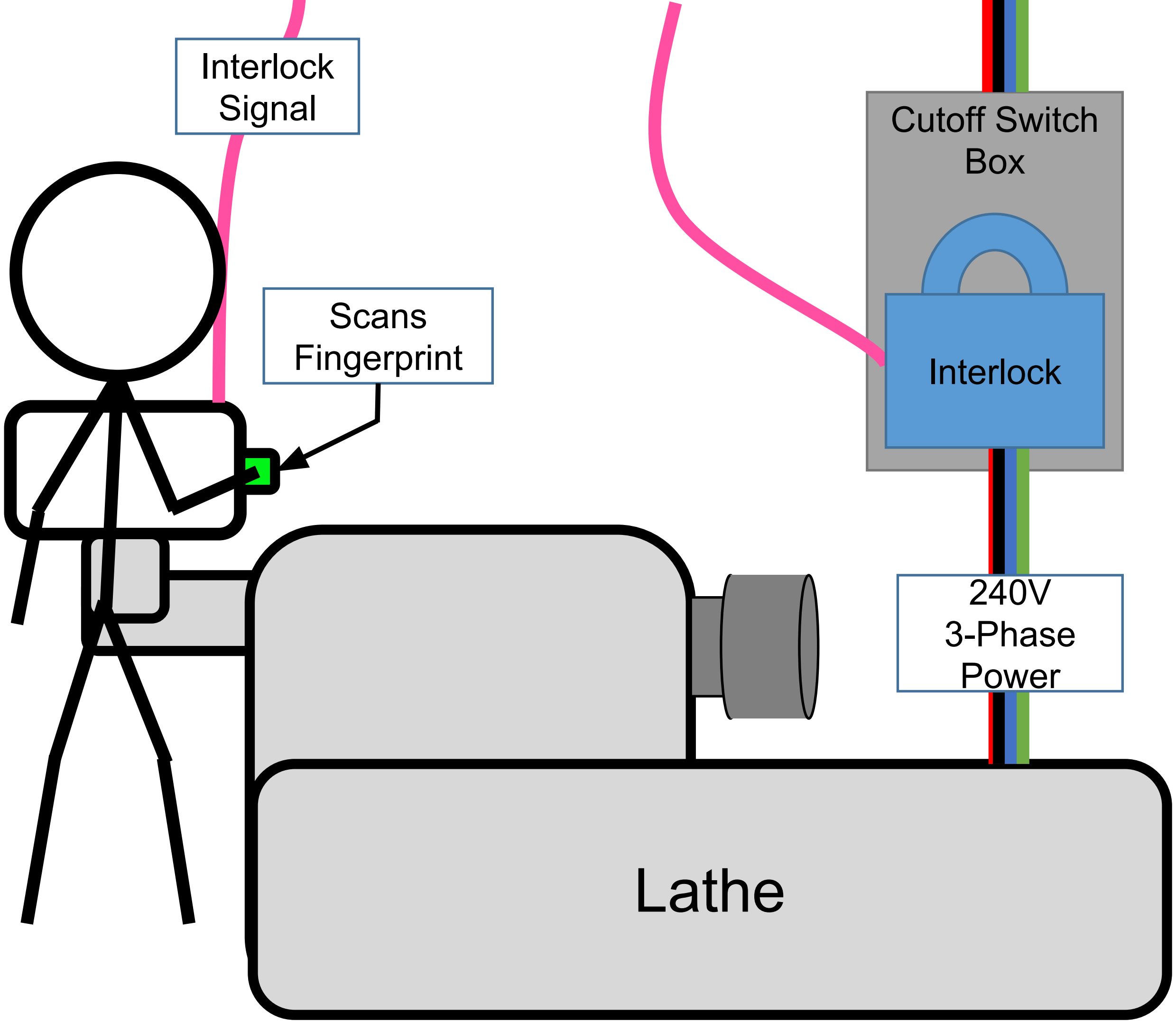
Authentication Scheme Downselection

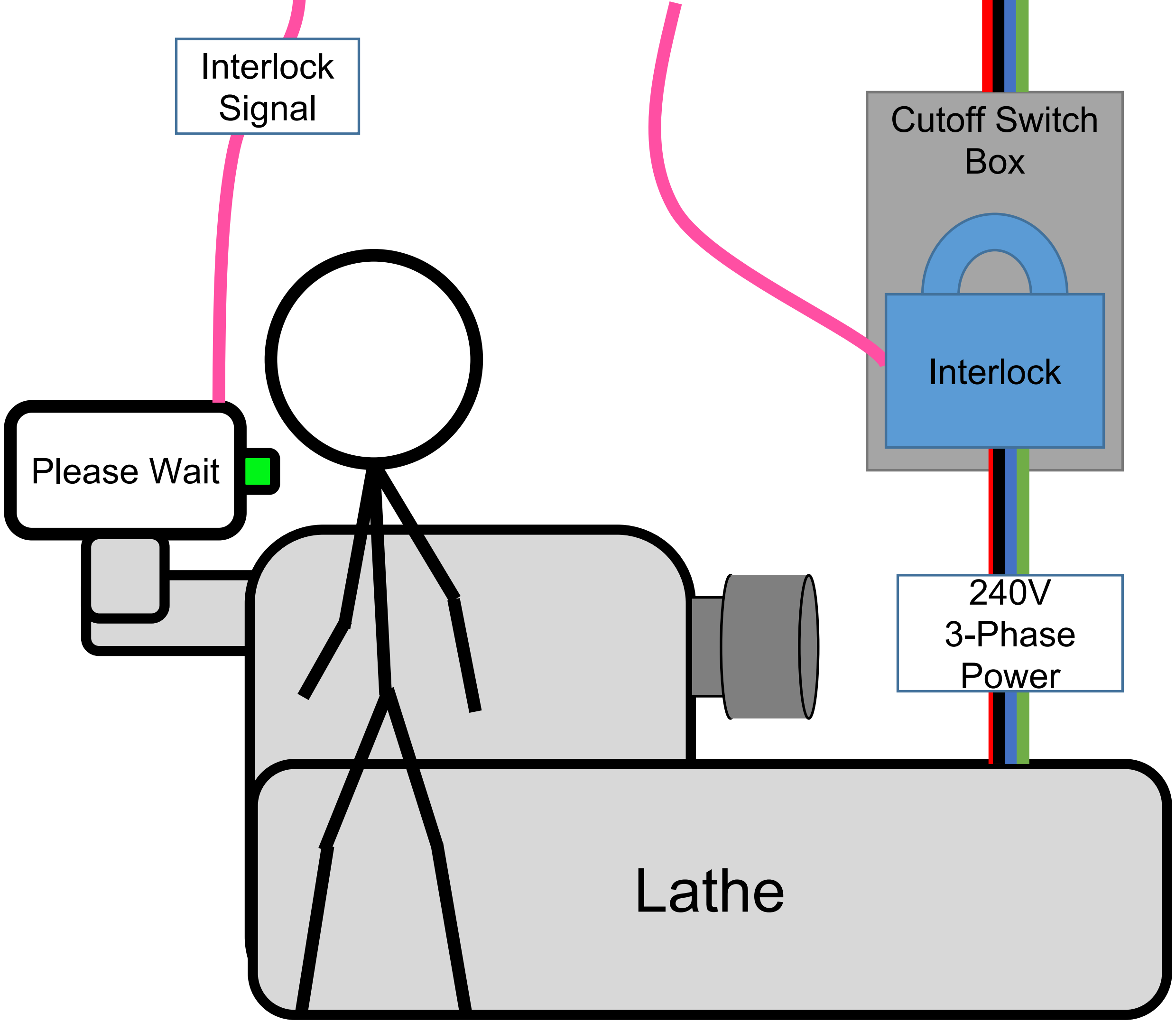
- Have and Know systems (keys, JCards, passwords) are too easy for a trained user to give away
  - Trained users allowing untrained users to access the shop is a known problem
  - So these candidates are eliminated
- Passive biometric systems (camera that matches your face or gait to a database, for example) offer the best user experience
  - User just walks up and starts using the machine
  - But they could accidentally unlock a machine if there is another trained user in the room
  - So these candidates are eliminated
- An active biometric system is selected as a compromise between user convenience, security, and reliability

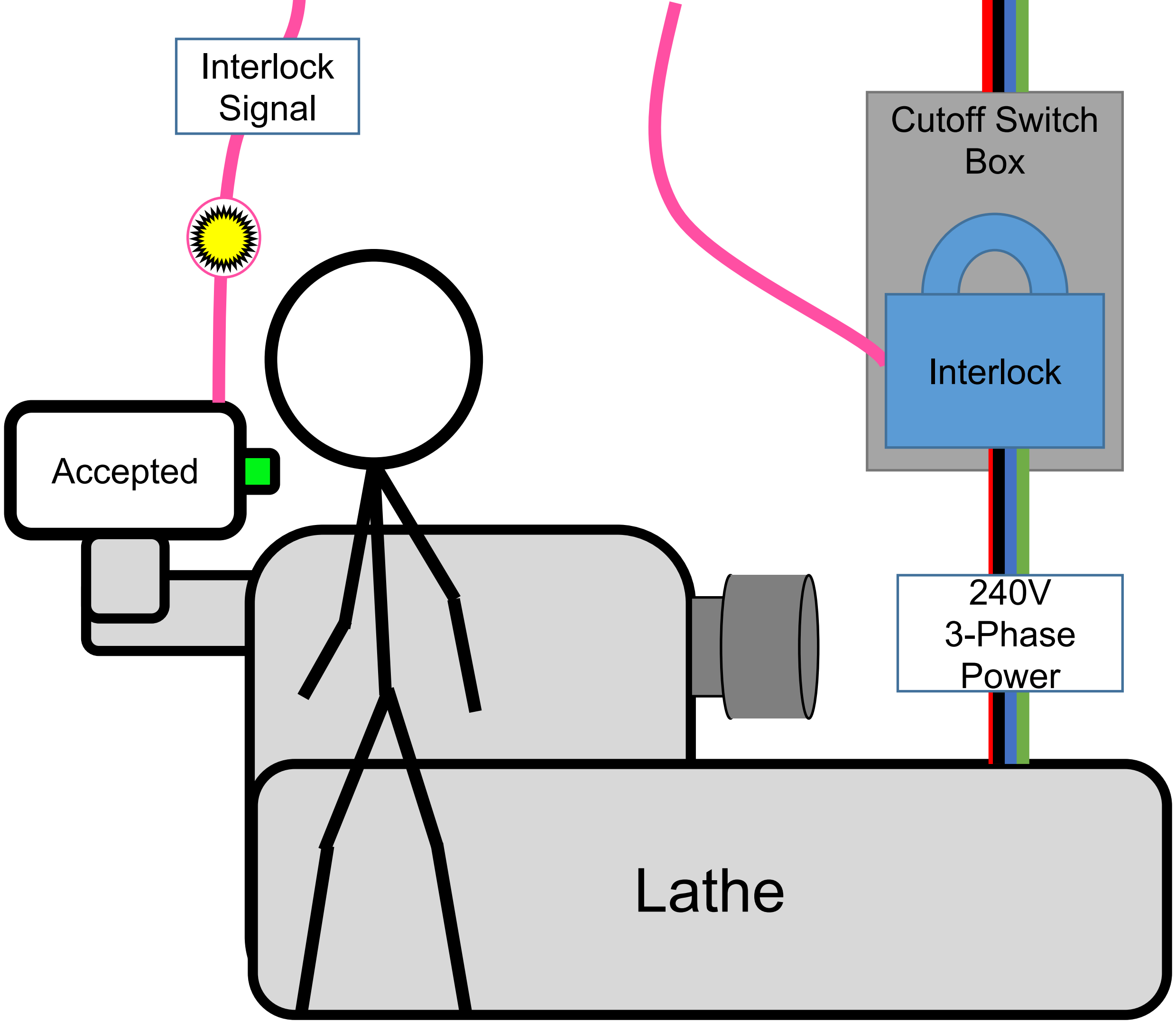
3.1: The Vision

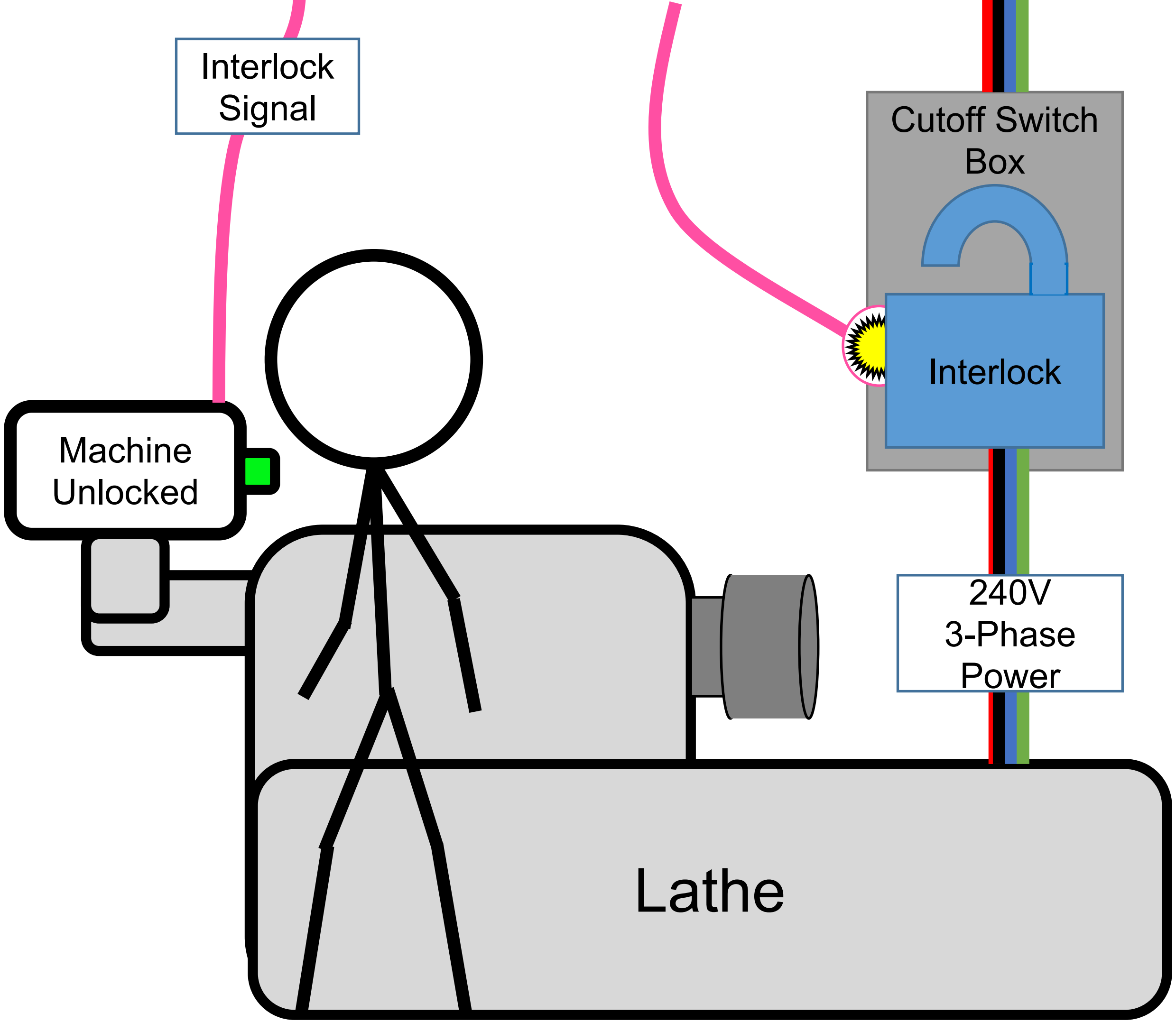

Interlock Signal
Cutoff Switch Box
Fingerprint Sensor
Interlock
Gatekeeper Screen
240V 3-Phase Power
Lathe


Authorized User
Interlock Signal
Cutoff Switch Box
Interlock
Fingerprint Sensor
Gatekeeper Screen
240V 3-Phase Power
Lathe


Interlock Signal
Cutoff Switch Box
Scans Fingerprint
Interlock
240V 3-Phase Power
Lathe


Interlock Signal
Cutoff Switch Box
Interlock
Please Wait
240V 3-Phase Power
Lathe

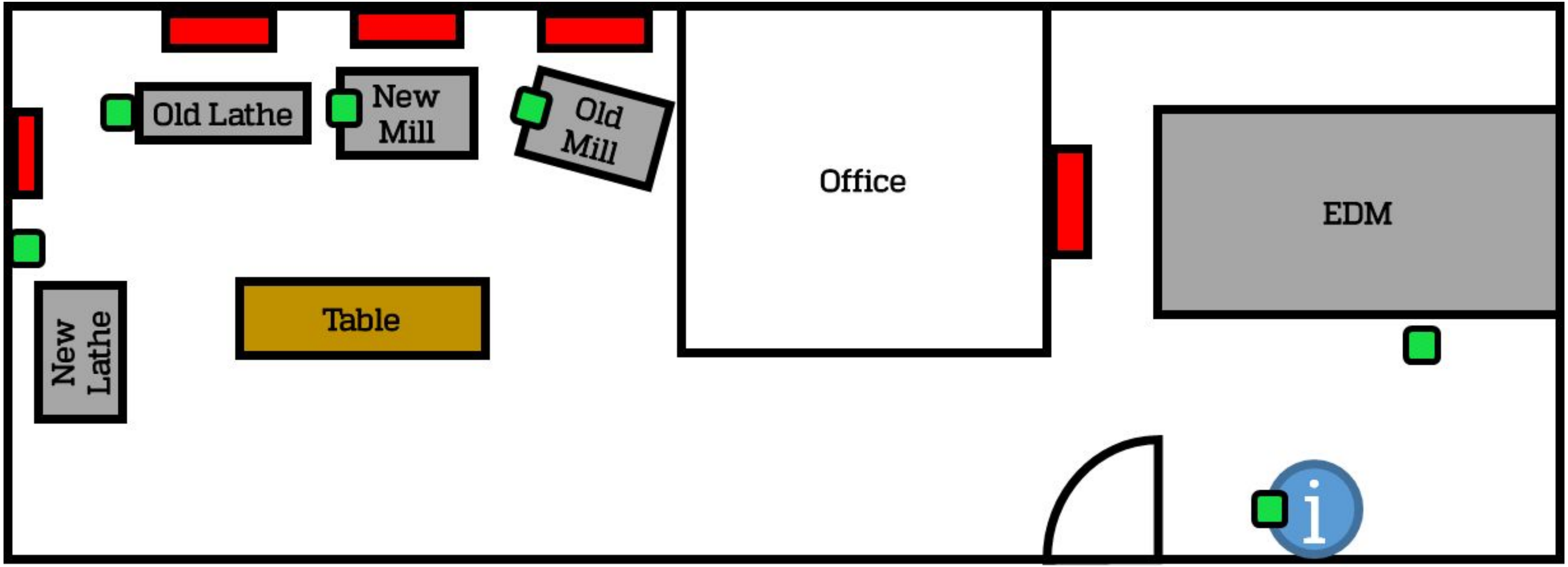

Interlock
Signal
Cutoff Switch
Box
Interlock
Accepted
240V
3-Phase
Power
Lathe


Interlock Signal
Cutoff Switch Box
Interlock
Machine Unlocked
240V 3-Phase Power
Lathe

The Vision

1. Trained user approaches the machine
2. User presents their finger to a device attached to the machine (“Gatekeeper”)
3. The print is checked against a database, confirming the user is trained
4. A signal is sent to a relay that connects power (or some other essential supply, e.g. compressed air for the EDM) to the machine
5. With power (or other supply) connected, the machine can be use normally

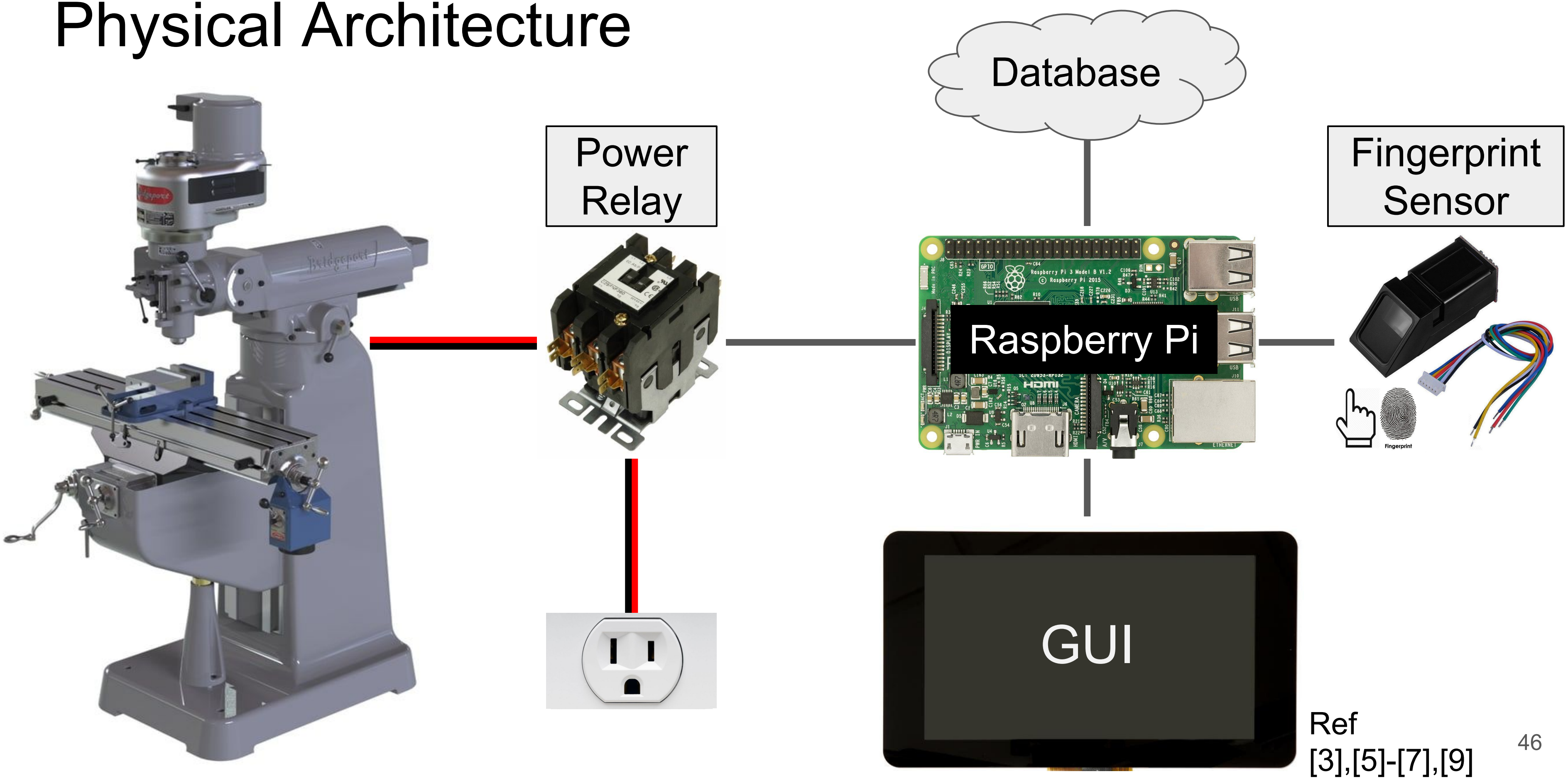
Our System in the Machine Shop



- ▨ - Machine
- 🟩 - Gatekeeper
- 🟥 - Power Relay
- ℹ - Information Kiosk

Section 4: Physical Implementation

Physical Architecture



Ref [3],[5]-[7],[9]

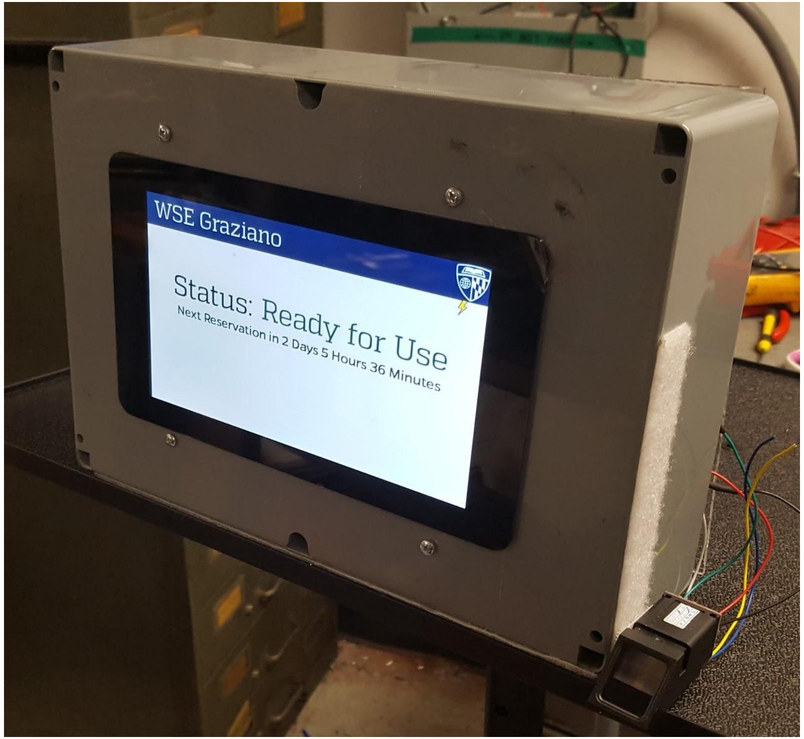
Architecture Explanation

Our system works by interrupting power from the wall to the machine. A “contactor” relay can connect and disconnect power, controlled by an input signal. The input signal is provided by a Raspberry Pi, running our Gatekeeper interface. When a user scans their fingerprint, the Pi checks a database to determine if the user possesses the necessary training to use the machine. If the user is suitably trained, the Pi then commands the contacter to close the connection from the wall to the machine. Otherwise, the connection remains broken, preventing untrained users from accessing the machine. When the connection is first closed, the Pi checks a machine state circuit to ensure that the machine does not immediately start to move (this would be dangerous). If the machine does start to move too quickly, power is immediately removed.

First Prototype: The Gatekeeper

Fully functional Gatekeeper system. Able to run the entire authentication loop:

- Users can scan their finger
- Database can recognize users
  - Connected to server using a static IP address
  - WSE IT maintained server
- Trained users can select a budget
  - Warning is displayed for untrained users
- Selecting a budget unlocks the machine
- User can then lock the machine when finished

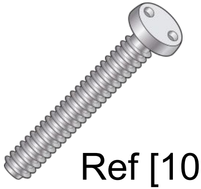


First Prototype (continued)

Gatekeeper back end electronics are fully functional

- System was able to use a 3.3V logic signal from the Raspberry Pi to activate a contacter, connecting wall power to a machine
- Prototype implemented activation circuit on a breadboard
  - Developed a custom PCB in the spring to replace breadboard
- Entire electronics back end, including circuitry to detect if the machine is on, was installed on the new Vectrax mill in the shop
- This is the prototype that was used for all live user testing

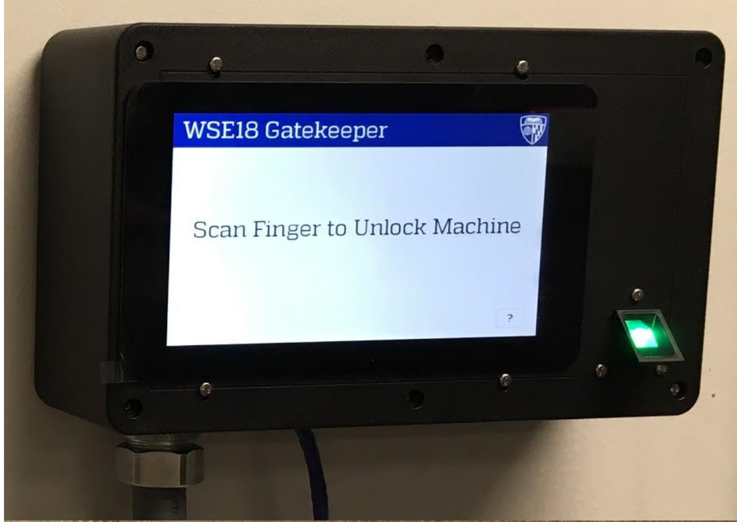
Final Gatekeeper Device

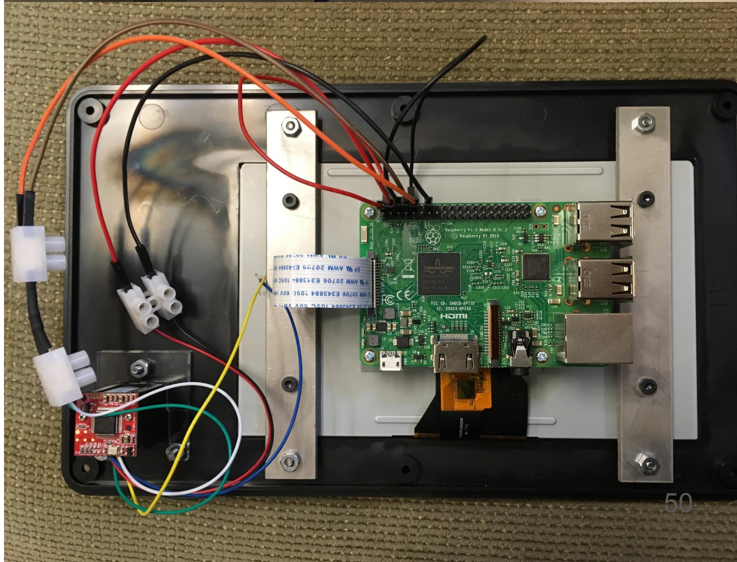


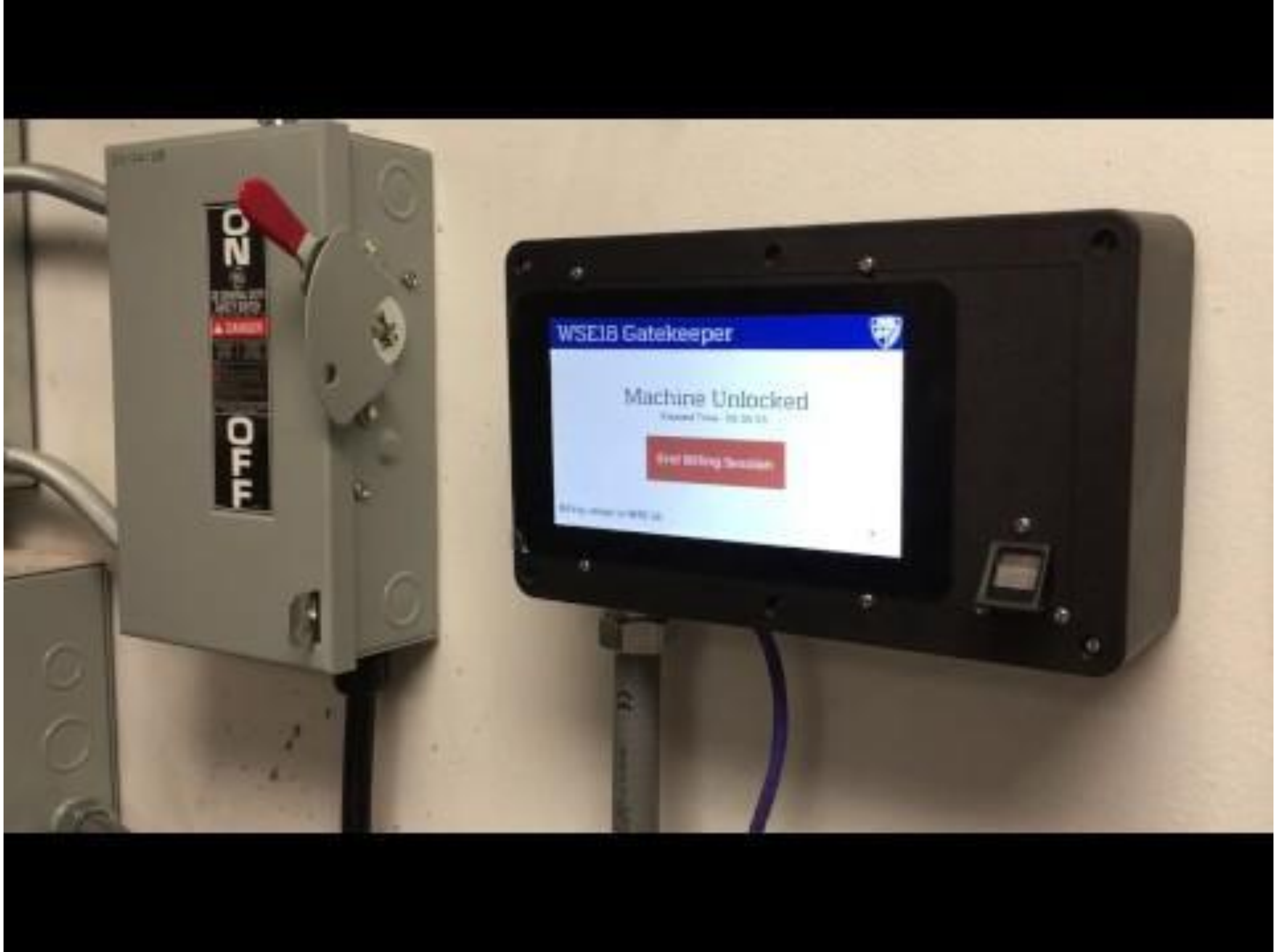
Ref [10]

Wall mounted in box closed with tamper-proof screws

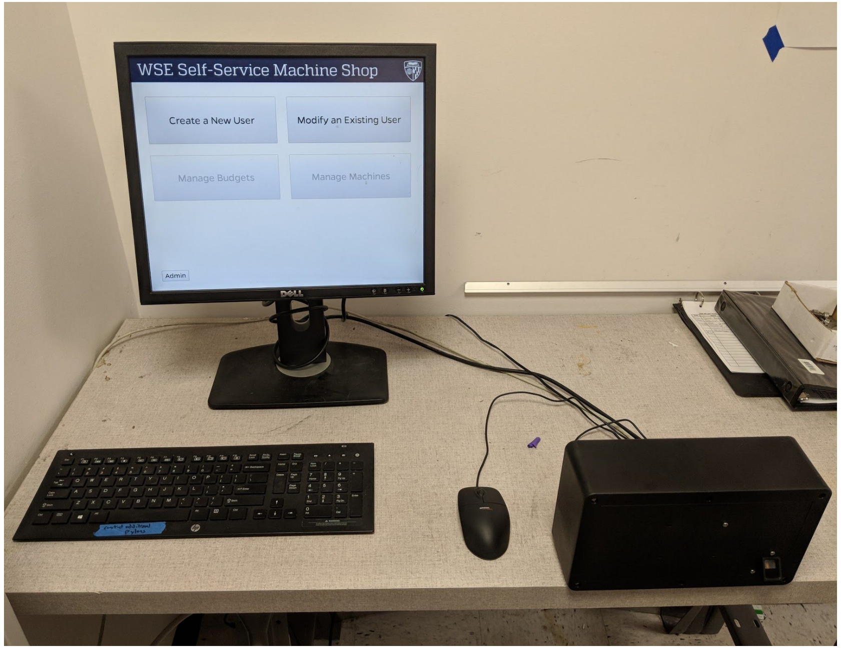
- User experience is the same as the prototype
- Runs a cleaner, updated version of the GUI
- More robust sensor mounting and improved wiring harness reduces error rates
- Electronics moved to PCB/solderboard
- Uses fixed but DHCP allocated IP address instead of static, making setup easier






ON
DANGER
OFF
WSE18 Gatekeeper
Machine Unlocked
End Billing Session

Final Kiosk Device



The Kiosk is run on a Raspberry Pi just like the Gatekeepers.

Instead of the Pi 7” touchscreen, the Kiosk uses a desktop monitor with mouse and keyboard peripherals so that users can enter information into the system.

The fingerprint sensor is mounted in a regular Gatekeeper housing minus the cutout for the touchscreen.

Section 5: Software

5.1: Software Requirements

Software Requirements

1. Manage fingerprint sensor
   a. Trigger print captures
   b. Read print from sensor
   c. Extract features
2. Manage training database
   a. Match captured prints to users
   b. Match users to training status on machines
   c. Match users to budgets they have access to
   d. Match users to shop usage
3. Provide a user interface
   a. Let users select from multiple budgets
   b. Inform users why a machine has refused to turn on

Software Requirements (cont.)

4. Provide a means for adding and modifying users
   a. Add fingerprints to the user
   b. Give access to budgets and machines
5. Switch Machine on and off safely
   a. Monitor state of machine and refuse to deactivate power if machine is unsafe
   b. Similarly, refuse to activate power if machine is unsafe
   c. This generally means not turning power on or off while the motor is engaged

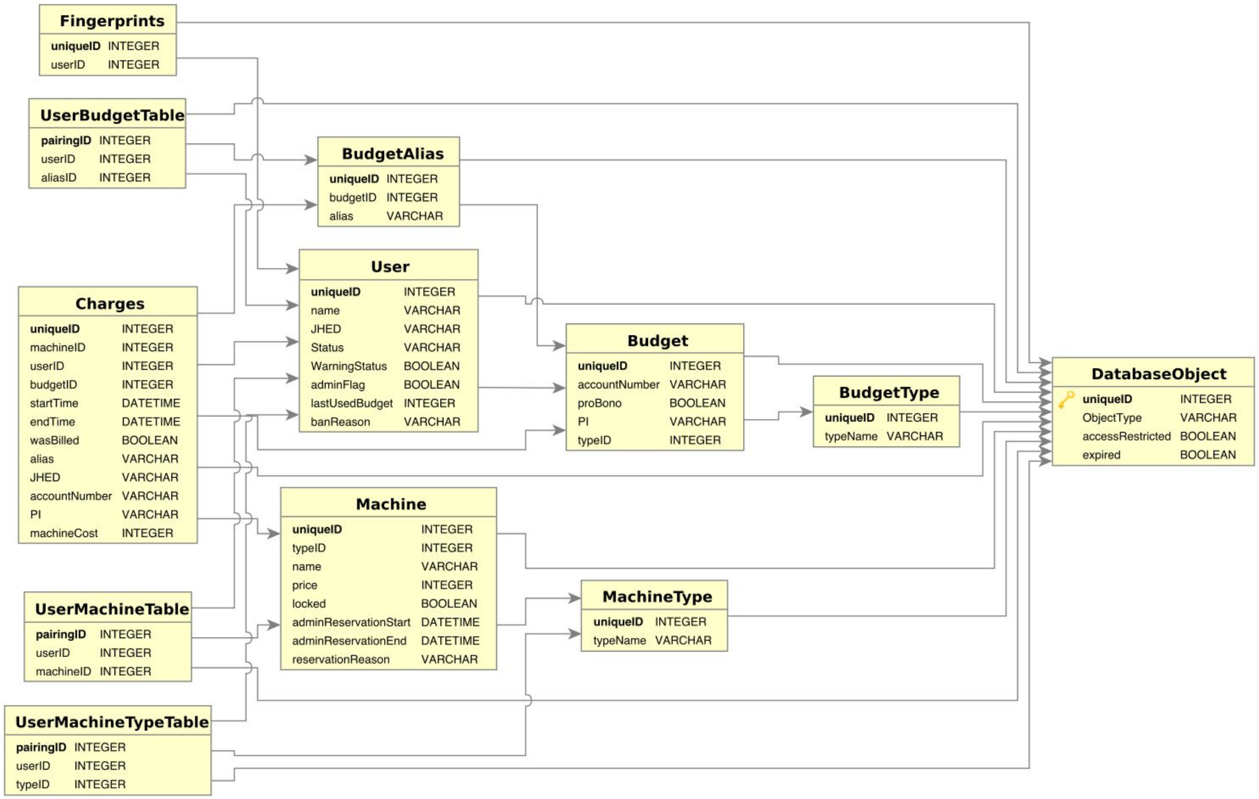
Data Management: What does the system need to remember?

- Primary Information related to machines
  - What machines are in the shop
  - How much they cost
  - Who is trained on them
  - If they are locked by an administrator for any reason (e.g. maintenance)
- Primary Information related to users
  - Identifying information (name, JHED, fingerprints)
  - What budgets they can access
  - What machines they are trained on
  - If they are restricted from shop usage (warned, no longer active, banned, etc.)
- Primary Information related to budgets
  - The budget code
  - “Aliases” for the budget (i.e. Mechatronics, WSE18 Senior Design Project)
  - the Principal Investigator

5.2: Software Architecture

Data Management Solution: SQL Database

The data includes information (such as training) that needs to be shared with multiple types of “objects.” A relational database can capture this efficiently



Fingerprints	
uniqueID	INTEGER
userID	INTEGER

UserBudgetTable	
pairingID	INTEGER
userID	INTEGER
aliasID	INTEGER

BudgetAlias	
uniqueID	INTEGER
budgetID	INTEGER
alias	VARCHAR

User	
uniqueID	INTEGER
name	VARCHAR
JHED	VARCHAR
Status	VARCHAR
WarningStatus	BOOLEAN
adminFlag	BOOLEAN
lastUsedBudget	INTEGER
banReason	VARCHAR

Charges	
uniqueID	INTEGER
machineID	INTEGER
userID	INTEGER
budgetID	INTEGER
startTime	DATETIME
endTime	DATETIME
wasBilled	BOOLEAN
alias	VARCHAR
JHED	VARCHAR
accountNumber	VARCHAR
PI	VARCHAR
machineCost	INTEGER

Budget	
uniqueID	INTEGER
accountNumber	VARCHAR
proBono	BOOLEAN
PI	VARCHAR
typeID	INTEGER

BudgetType	
uniqueID	INTEGER
typeName	VARCHAR

DatabaseObject	
uniqueID	INTEGER
ObjectType	VARCHAR
accessRestricted	BOOLEAN
expired	BOOLEAN

Machine	
uniqueID	INTEGER
typeID	INTEGER
name	VARCHAR
price	INTEGER
locked	BOOLEAN
adminReservationStart	DATETIME
adminReservationEnd	DATETIME
reservationReason	VARCHAR

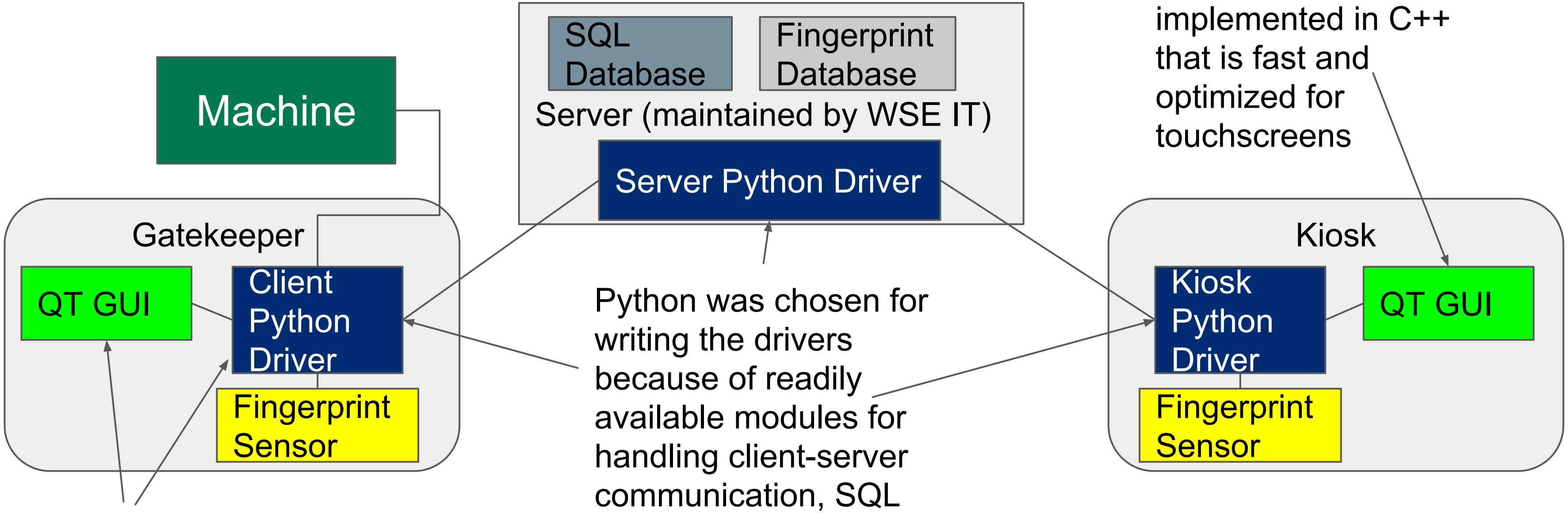
MachineType	
uniqueID	INTEGER
typeName	VARCHAR

UserMachineTable	
pairingID	INTEGER
userID	INTEGER
machineID	INTEGER

UserMachineTypeTable	
pairingID	INTEGER
userID	INTEGER
typeID	INTEGER

Visualization of actual database structure

Solution Architecture



QT is a GUI builder implemented in C++ that is fast and optimized for touchscreens

Python was chosen for writing the drivers because of readily available modules for handling client-server communication, SQL databases, and GPIO devices (like the fingerprint sensor and the machine switches)

The separation of the driver and GUI into separate processes on each device makes the system more robust. If the GUI fails, there is no risk to important data.

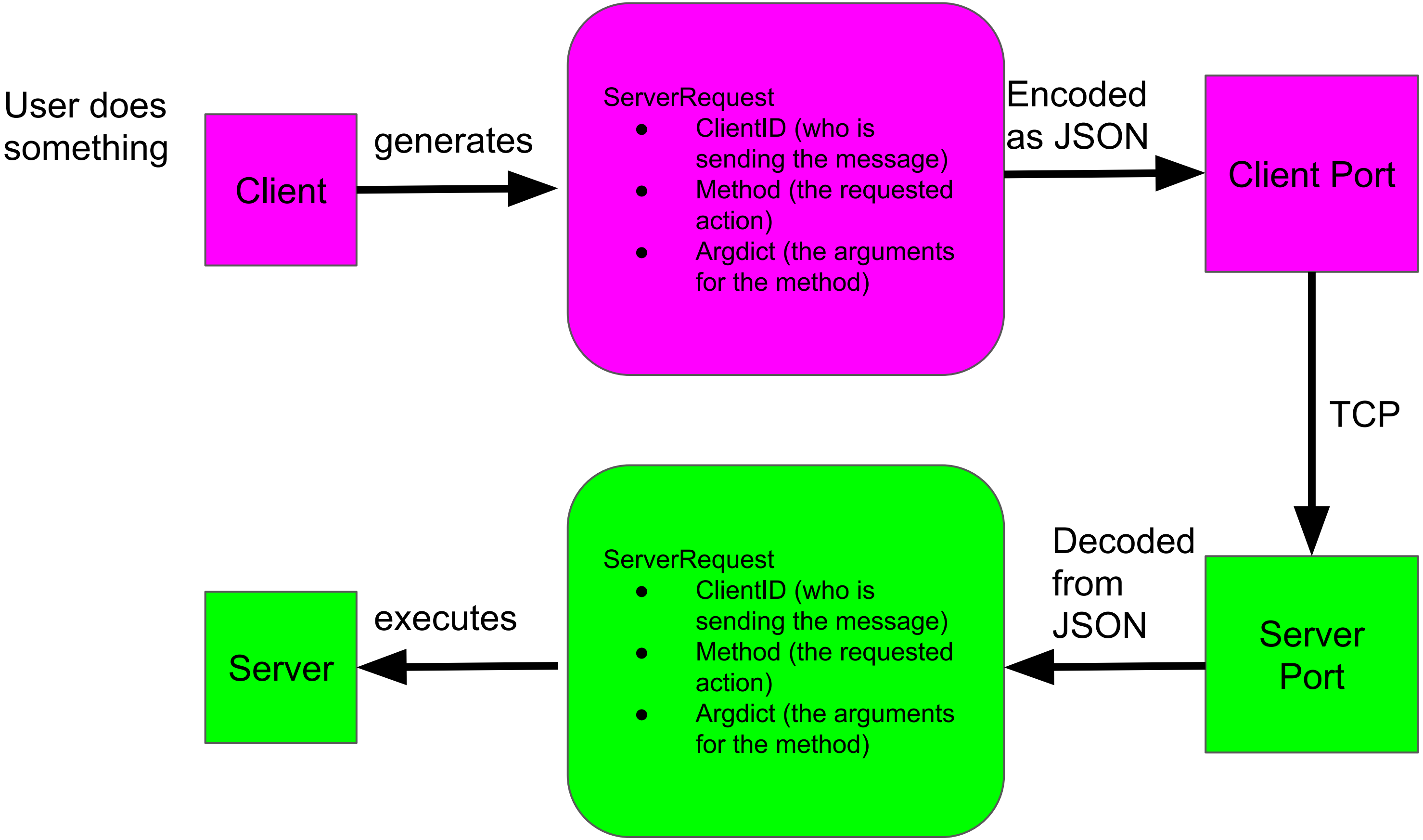
Software Components

- Client Python driver
  - manages the fingerprint sensor (req 1)
  - Handles communication with the server
  - switches the machine on and off (req 5)
  - Controls Gatekeeper GUI
- Kiosk Python driver
  - Manages its fingerprint sensor
  - Handles communication with the server
  - Provides a means of adding and modifying users (req 4)
  - Controls Kiosk GUI
- Server Python driver
  - Manages SQL and fingerprint databases (req 2)
- QT GUI
  - Provides a user interface for both Gatekeeper and Kiosk (req 3)

Interprocess Communication

- The Python drivers need to communicate with their GUI and with the Server Driver
- This is supported by a custom Application Layer Protocol implemented using the TCP Transport Layer Protocol
- Client-GUI and Client-Server communication uses the same protocol on different ports
- The protocol transmits Javascript Object Notation (JSON) messages encoding an instruction, information necessary to carry out that instruction, and the results of the instruction back and forth
- The communication with the server is authenticated using x509 certificates and encrypted using TLSv1.2

Client to Server Communication Protocol



User does something

Client

generates

ServerRequest
- ClientID (who is sending the message)
- Method (the requested action)
- Argdict (the arguments for the method)

Encoded as JSON

Client Port

TCP

Server Port

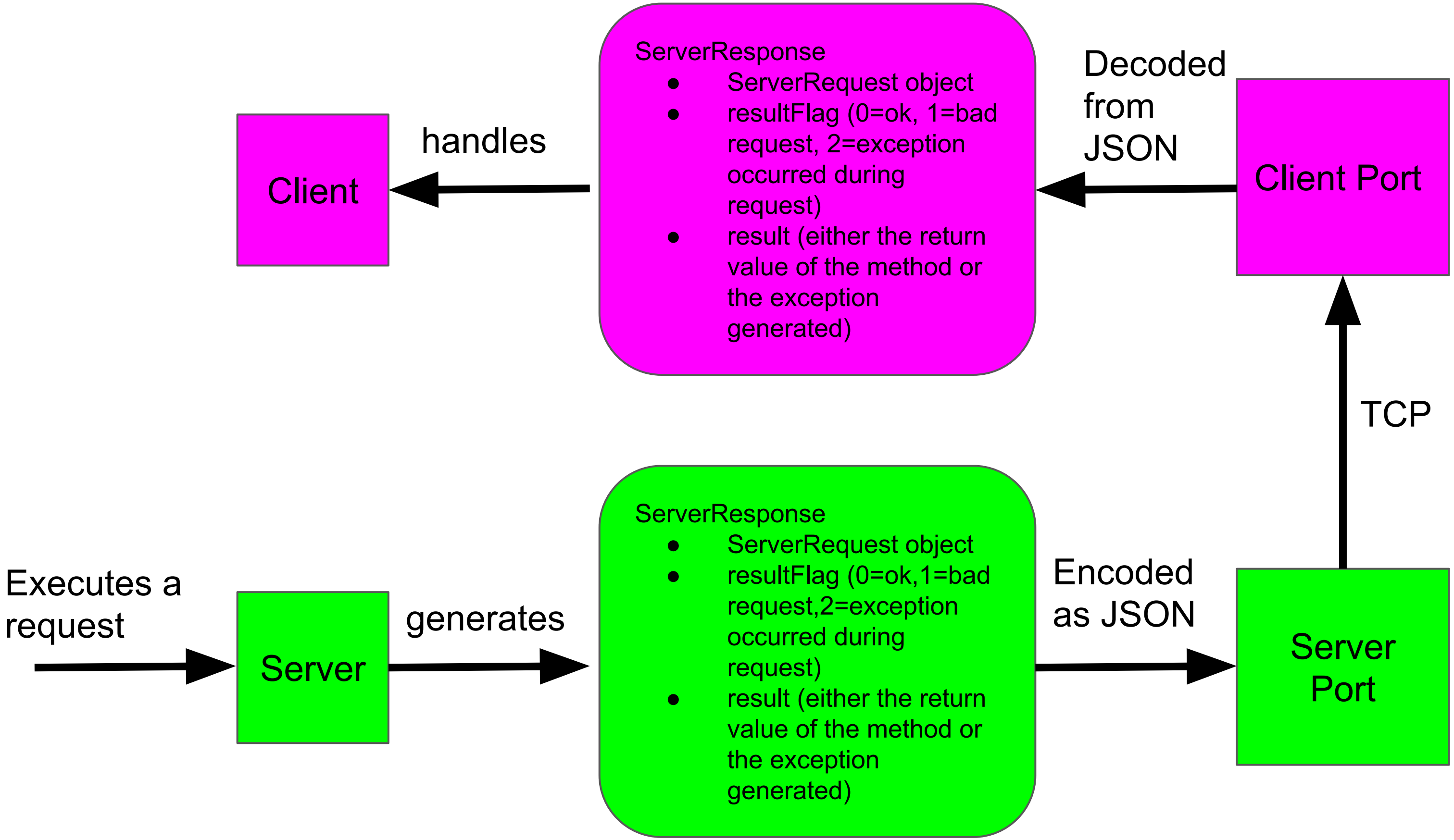
Decoded from JSON

ServerRequest
- ClientID (who is sending the message)
- Method (the requested action)
- Argdict (the arguments for the method)

executes

Server

Server to Client Communication Protocol



Executes a request

Server

generates

ServerResponse
- ServerRequest object
- resultFlag (0=ok,1=bad request,2=exception occurred during request)
- result (either the return value of the method or the exception generated)

Encoded as JSON

Server Port

TCP

Client Port

Decoded from JSON

ServerResponse
- ServerRequest object
- resultFlag (0=ok, 1=bad request, 2=exception occurred during request)
- result (either the return value of the method or the exception generated)

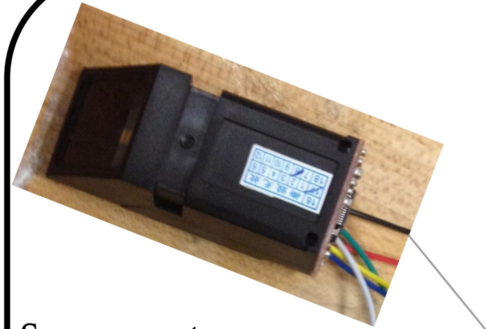
handles

Client

Fingerprint Handling

- When a new fingerprint image is required, either for enrolling a user or for trying to log into a machine, the Python driver signals the fingerprint sensor to take a fingerprint image and upload it to the Raspberry Pi for processing
- Characteristic features (called minutiae) are extracted from the image using the mindtct software from the National Institute of Standards and Technology
- These features are sent to the server using the protocol outlined earlier, to either be recorded in the “gallery” of fingerprint images or compared against existing prints
    - Comparison against existing prints is conducted by NIST’s bozorth3 software
- Actual fingerprint images NEVER leave RAM on the Pi that captured them and they are immediately deleted. Only feature data is retained.

Recording a fingerprint



Sensor captures an image, and...

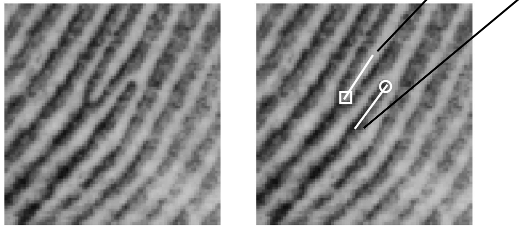
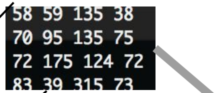


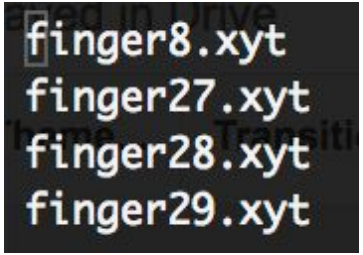
Figure 3. Minutiae: bifurcation (square marker) and ridge ending (circle marker).

...mindtct executable extracts minutiae...

...which the server saves as .xyt files...



```
58 59 135 38
70 95 135 75
72 175 124 72
83 39 315 73
```

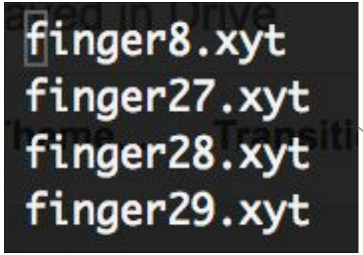


```
finger8.xyt
finger27.xyt
finger28.xyt
finger29.xyt
```

...that are tracked in the gallery file and SQL database

Ref [11]

Recognizing a User

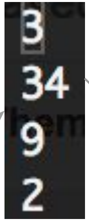


```
finger8.xyt
finger27.xyt
finger28.xyt
finger29.xyt
```

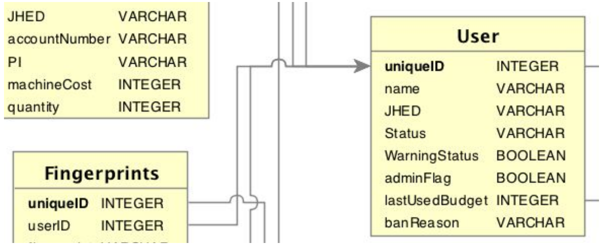
Gallery of fingerprints

Bozorth3 executable generates match scores...

...which are stored in a text file



```
3
34
9
2
```



JHED	VARCHAR
accountNumber	VARCHAR
PI	VARCHAR
machineCost	INTEGER
quantity	INTEGER

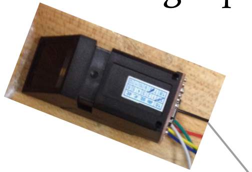
Fingerprints

uniqueID	INTEGER
userID	INTEGER

User

uniqueID	INTEGER
name	VARCHAR
JHED	VARCHAR
Status	VARCHAR
WarningStatus	BOOLEAN
adminFlag	BOOLEAN
lastUsedBudget	INTEGER
banReason	VARCHAR

Finger presented to Gatekeeper



Sensor captures an image, and...

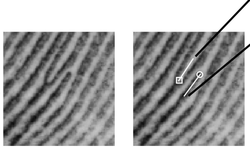
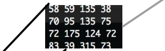


Figure 3. Minutiae: bifurcation (square marker) and ridge ending (circle marker).

...mindtct extracts minutiae...



```
58 59 135 38
70 95 135 75
72 175 124 72
83 39 315 73
```

...which are saved as a .xyt file...

Scores above a threshold indicate a match with a gallery fingerprint (and thus a user)

Ref [11]

Server Tasks

1. Maintain database
    a. Database stored as a sqlite3 file
    b. Python Database object “owns” the database and contains all methods for modifying it
2. Ensure database integrity
    a. Make daily backups of the database contents in case of data corruption
    b. Accomplished using a scheduled task (cronjob)
3. Carry out the task of billing
    a. At a specified interval (monthly), construct a spreadsheet with all the new charges and email it to shop administrators
    b. Accomplished using a scheduled task (cronjob)
4. Securely interact with Gatekeepers and Kiosk
    a. Firewall configured to only allow connections from pre-authorized IP addresses on specific ports
    b. Python secure socketserver that responds only to clients with pre-authorized SSL certificates

Server Configuration File

- All four server tasks need some information about the environment in which they are operating
- This shared, persistent information is configured by admins when setting up the system by editing the server's configuration file
- The python executables run by the cronjobs read this settings file in to determine how to carry out their tasks, as does the socketserver that communicates with the clients
- Of special note is the folder in which logs are stored. By making this a path to a remotely accessible drive, administrators can easily monitor the server's activity

Entries in the Server Configuration File

#1. list of email addresses to send billing spreadsheets to
#2. logging mode-True to store logs in a file in logging folder, False to just dump them to stdout
#3. logging folder-folder to store log files in
#4. sql dialect-the dialect of sql to use to manage the database
#5. sql driver-should be left default
#6. database URL-prepend a '/' to the absolute file path to the database file
#7. server ip address
#8. port to use for communication with the clients
#9. path to folder for storing fingerprints
#10. filename for the fingerprint gallery file
#11. matching threshold value for fingerprint matches
#12. path to server's certificate file
#13. path to folder containing acceptable certificates for clients
#14. String to display to user when they lack training on a machine
#15. Path to folder to store database backups in

Client Configuration File

- To make deployment of new devices easier, Kiosks and Gatekeepers use the exact same executable, StartClient.py
- The differences between Kiosks and Gatekeepers are captured in the client configuration file that is passed to this StartClient.py
- The client config file tells the client if it is a kiosk or a Gatekeeper, where the appropriate GUI executable is stored, and (if it is a Gatekeeper) which machine it is controlling
- Administrators customize this file when setting up a new device
- If the device is told it is a Gatekeeper, every time it is rebooted, it asks the server to email all admins to let them know
    - This is done to inform the admins of possible malfunction or tampering in real-time

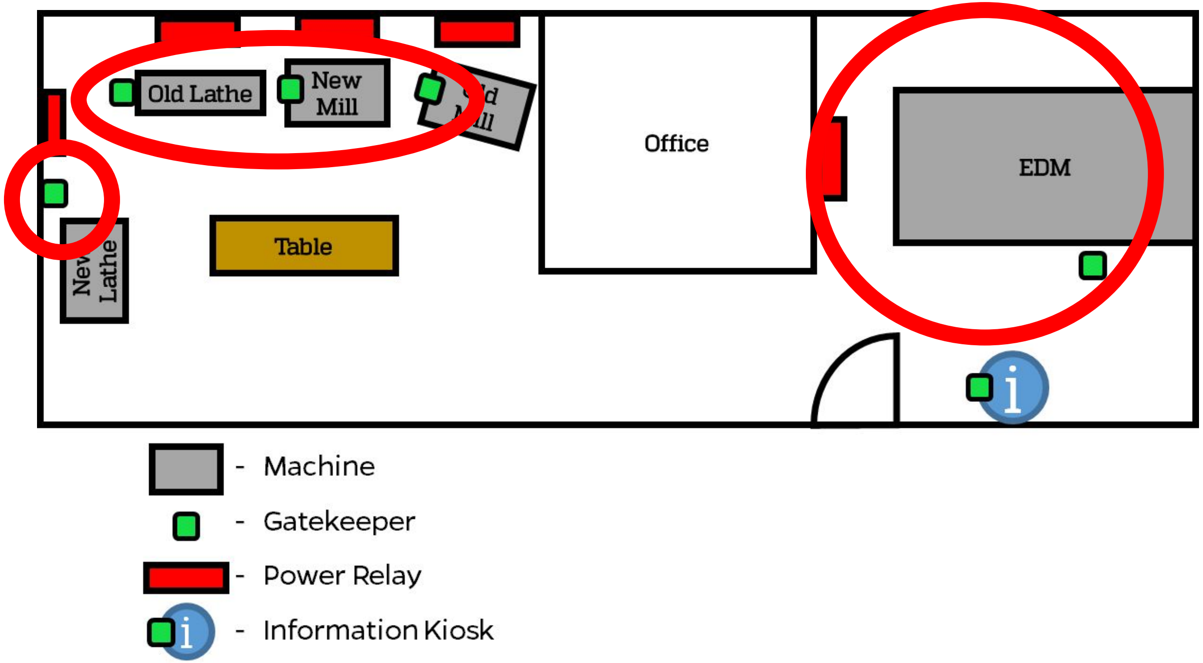
Entries in the Client Configuration File

#1. host is the ip address/hostname of the server
#2. port is the port to contact the server on
#3. machineID is the uniqueID of the machine this corresponds to. Does nothing if this is a Kiosk
#4. logdir is the folder to store the logs in
#5. production tells it to log to file if True or to command line if False
#6. certfile is the path to the certificate to use when talking to the server
#7. cafile is the folder in which acceptable server certificates are stored
#8. type is kiosk or client. If kiosk, this pi is a kiosk; if client, this pi is a machine client.
#9. gui_path is the path to the gui executable. Default is empty, which results in no GUI.

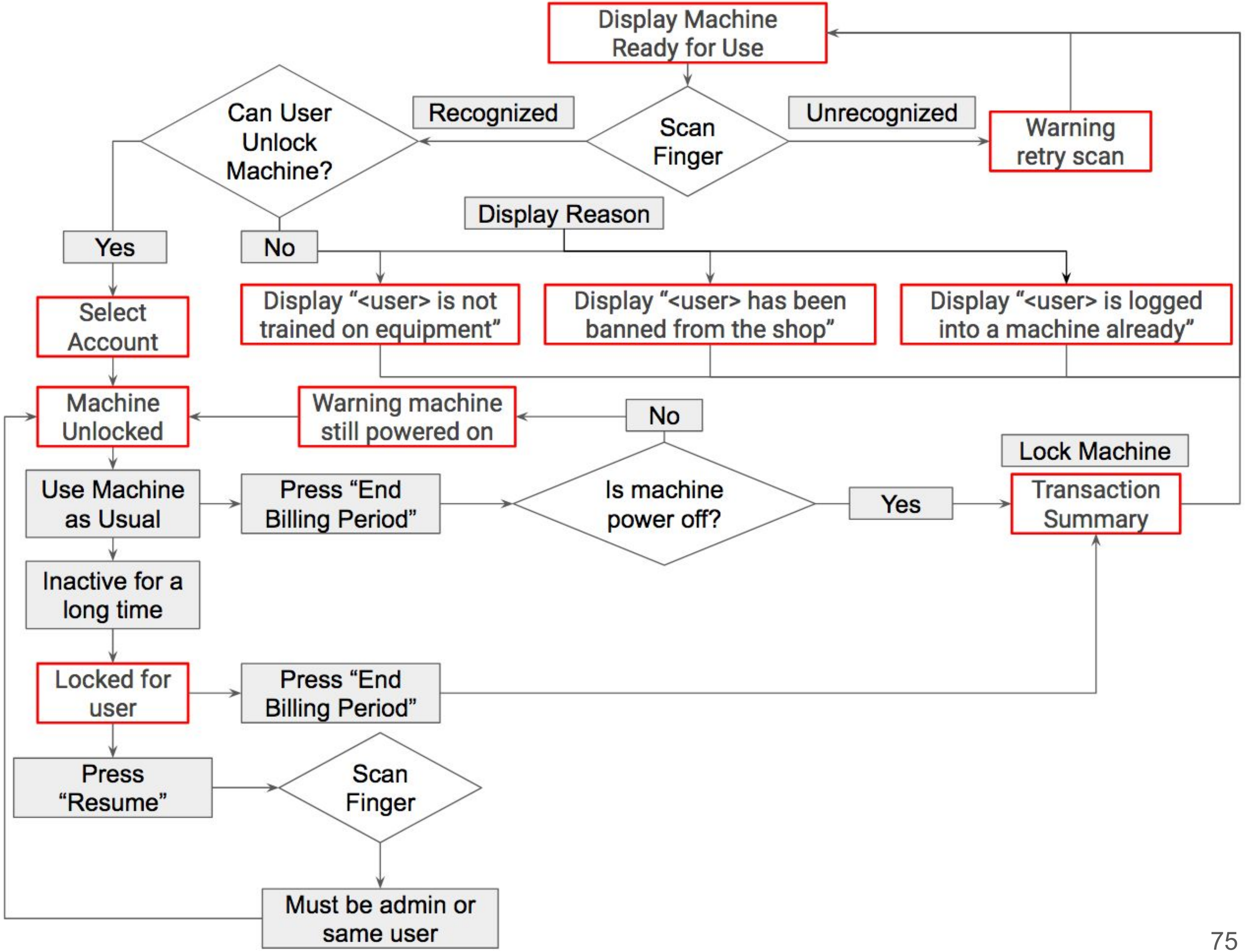
5.3: User Interfacing

The Gatekeeper

A Gatekeeper is the system component mounted to a machine. It provides the GUI, authentication, budget selection, power control/monitoring, and automated billing

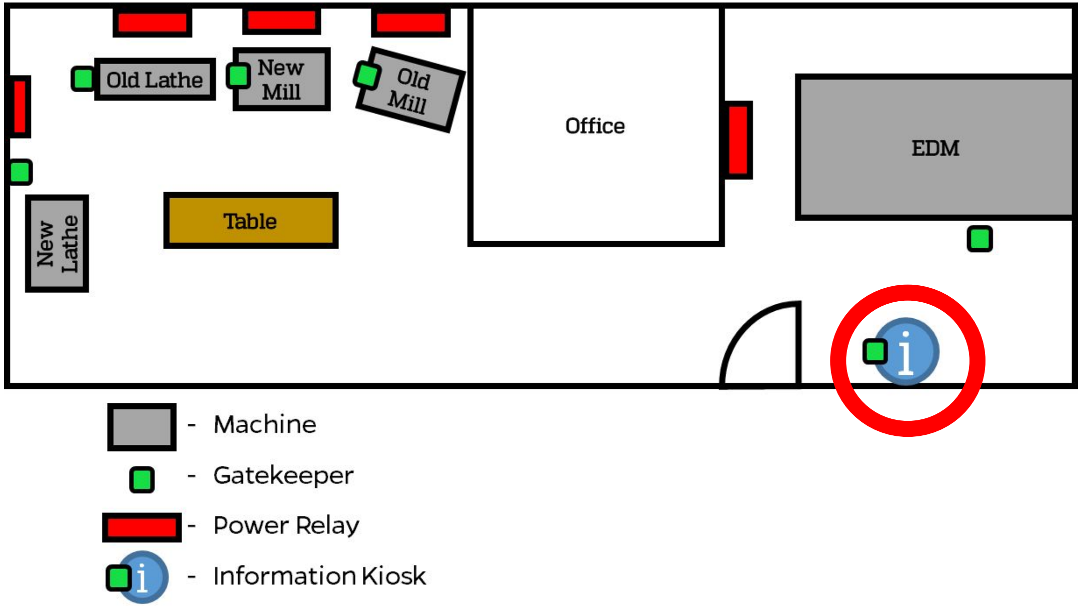


Flowchart of Events at the Gatekeeper

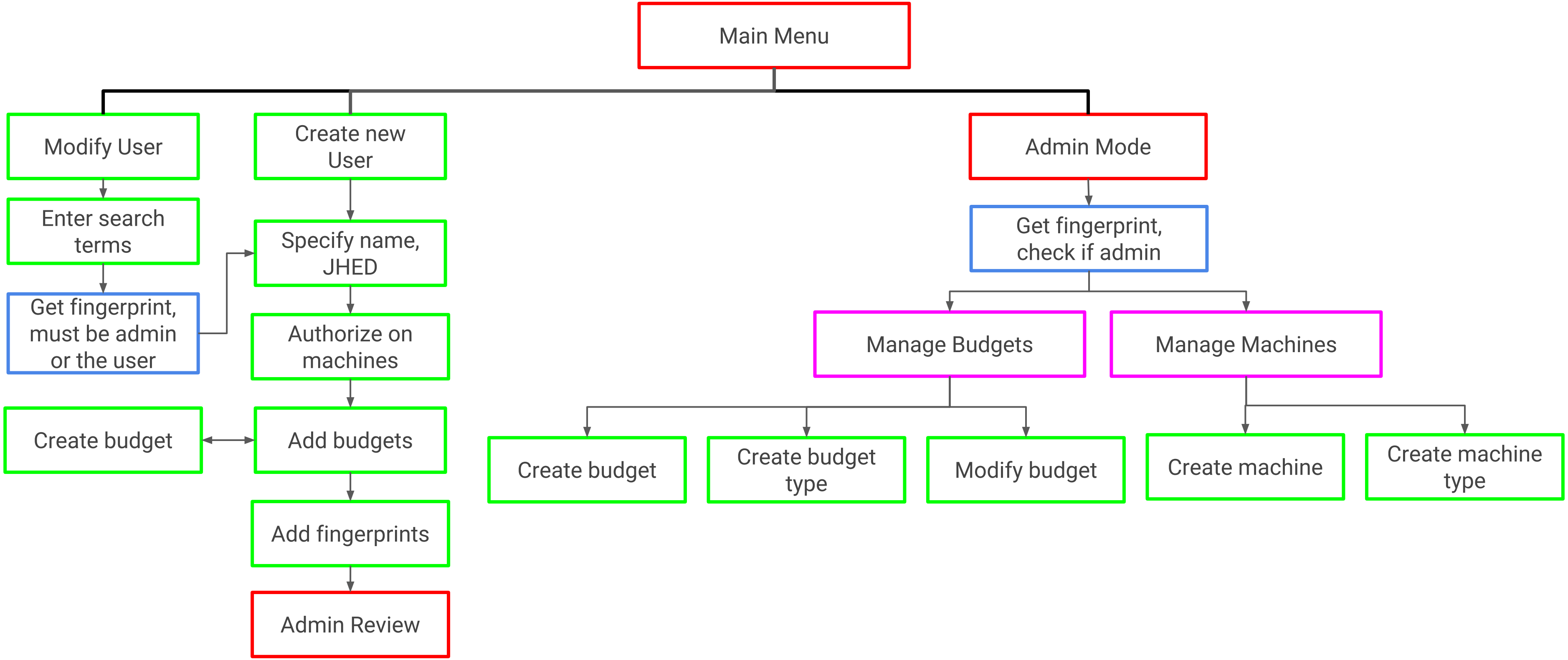

Display Machine Ready for Use
Scan Finger
Recognized
Unrecognized
Warning retry scan
Can User Unlock Machine?
Yes
No
Display Reason
Select Account
Display "<user> is not trained on equipment"
Display "<user> has been banned from the shop"
Display "<user> is logged into a machine already"
Machine Unlocked
Warning machine still powered on
No
Lock Machine
Use Machine as Usual
Press "End Billing Period"
Is machine power off?
Yes
Transaction Summary
Inactive for a long time
Locked for user
Press "End Billing Period"
Press "Resume"
Scan Finger
Must be admin or same user

The Kiosk

The kiosk is the primary mechanism for managing users, budgets, and machines



Flowchart of Events at the Kiosk



Return to main menu at end of sequence

5.3.1: Gatekeeper GUI

Gatekeeper User Interface (GUI)

The GUI provides essential feedback to users while they interact with the system

GUI should:

- Explain to unauthorized users why they cannot access a machine
- Allow authorized users to select budget to charge
- Notify users when the machine is unlocked

GUI should be as simple as possible so as to not burden users when they use the machines

Original GUI Concept

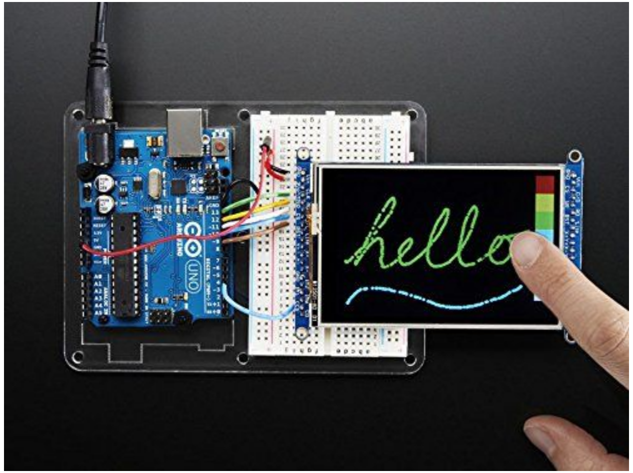
Initial plan was to use 3.5” Touchscreen display

Microcontroller would directly write the pixels to the display

- Included Adafruit GUI library for basic graphical elements (lines, shapes, text, etc.)

Required separate analog read capability to detect finger presses on the screen

Adafruit library provides no high level GUI tools, which would have lengthened development time



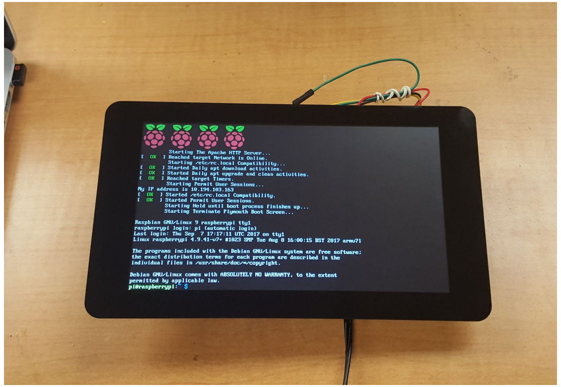
Also, a 3.5” was determined to be too small to easily read

Ref [2]

GUI Final Screen Hardware

Raspberry Pi 7” touchscreen display

- Full color
- Capacitive Touch Screen
- 800x480 (480p) resolution
- Designed to interface with the Pi
    - Screen standoffs connect to pi mounting holes
    - Screen connects directly to Display Serial Interface
- Easy to mount to an external enclosure

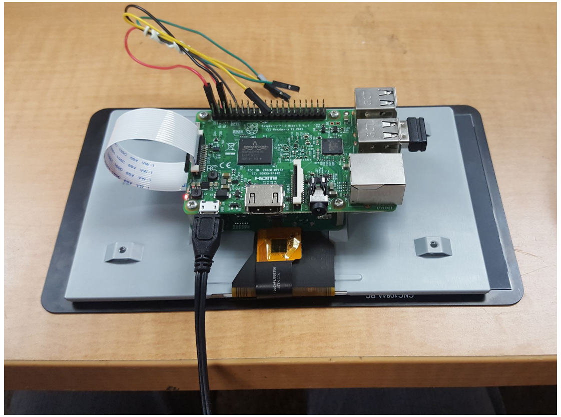


```
         Starting The Apache HTTP Server...
[  OK  ] Reached target Network is Online.
         Starting /etc/rc.local Compatibility...
[  OK  ] Started Daily apt download activities.
[  OK  ] Started Daily apt upgrade and clean activities.
[  OK  ] Reached target Timers.
         Starting Permit User Sessions...
My IP address is 10.194.103.163
[  OK  ] Started /etc/rc.local Compatibility.
[  OK  ] Started Permit User Sessions.
         Starting Hold until boot process finishes up...
         Starting Terminate Plymouth Boot Screen...

Raspbian GNU/Linux 9 raspberrypi tty1
raspberrypi login: pi (automatic login)
Last login: Thu Sep  7 17:17:11 UTC 2017 on tty1
Linux raspberrypi 4.9.41-v7+ #1023 SMP Tue Aug 8 16:00:15 BST 2017 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi:~ $
```



GUI Software Implementation

The Gatekeeper User Interface was implemented using the Qt framework. Qt is a cross platform development framework designed specifically for running GUIs on embedded touchscreen systems.

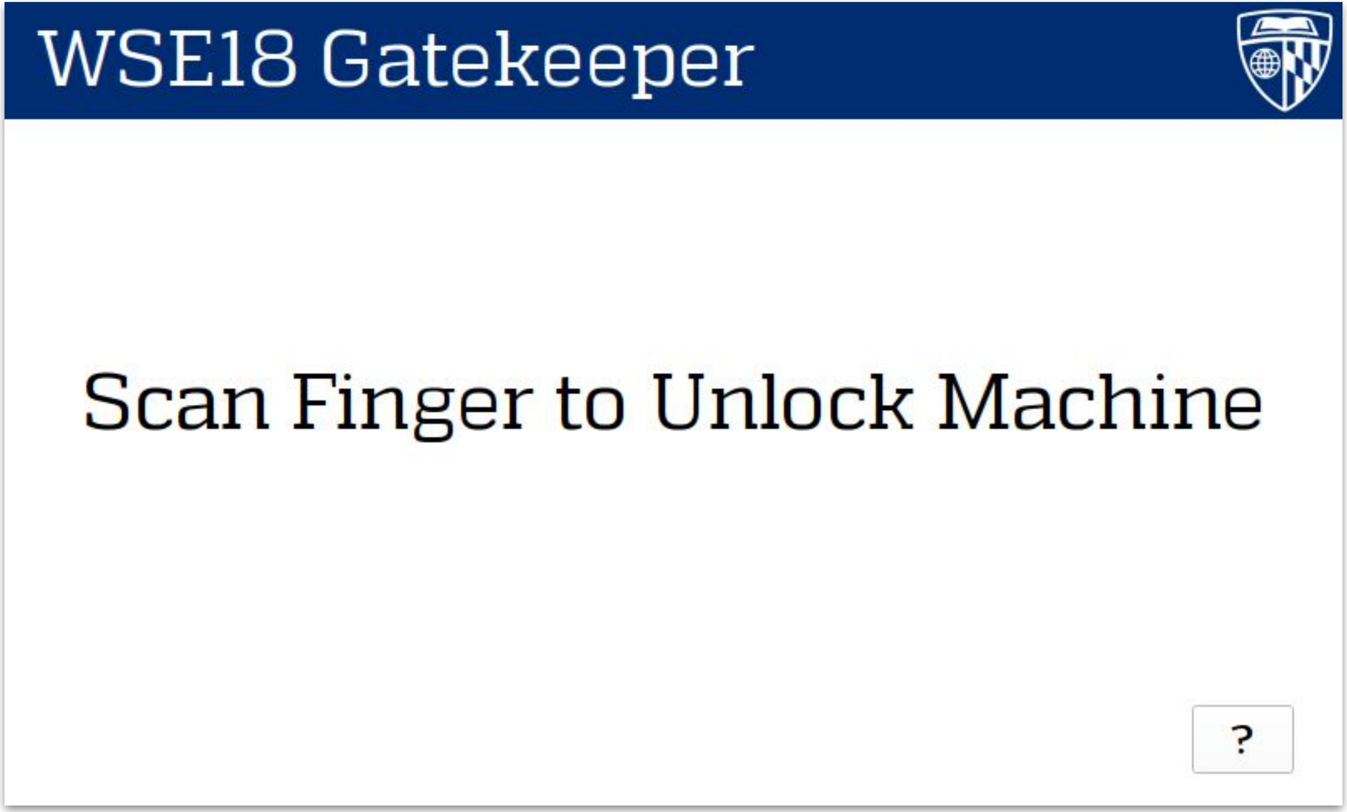
Qt is written in C++ ensuring that the GUI is fast and responsive. Additionally, it will run on just about any device architecture due to C++'s portability. Lastly the framework contains many common GUI features such as dropdowns, textboxes, buttons, etc. which decreases development time.

Ref [4],[8]

GUI Screens

The following are the GUI screens as they appear on an actual Gatekeeper device. Each screen is followed by an explanation of how it functions.

GUI Screen: Present Finger



WSE18 Gatekeeper

Scan Finger to Unlock Machine

Present Finger Screen Explanation

Default screen, stating that a fingerprint is required to unlock the machine

The fingerprint sensor flashes green while this screen is displayed

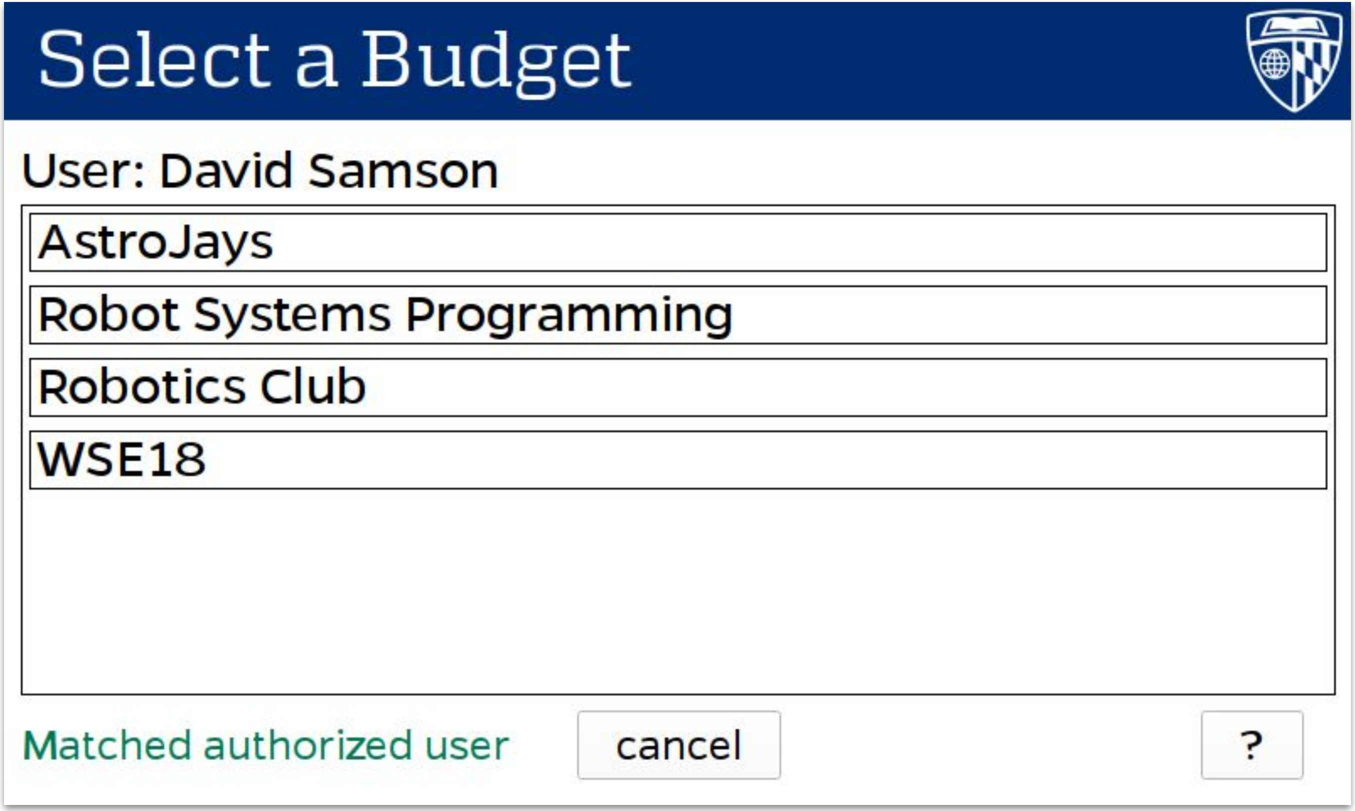
From here, a user can scan their finger to begin the authentication process

After a print is scanned, the following can occur

- The user is recognized, and allowed to use the machine
- The user is recognized, and not allowed to use the machine
- The user is not recognized

The first option goes to the select budget screen, while the others display error messages

GUI Screen: Select Budget



Select a Budget

User: David Samson

AstroJays

Robot Systems Programming

Robotics Club

WSE18

Matched authorized user

Select Budget Screen Explanation

This screen is displayed if the user’s fingerprint was recognized, and they have received training for this machine

Displays budgets the user can select

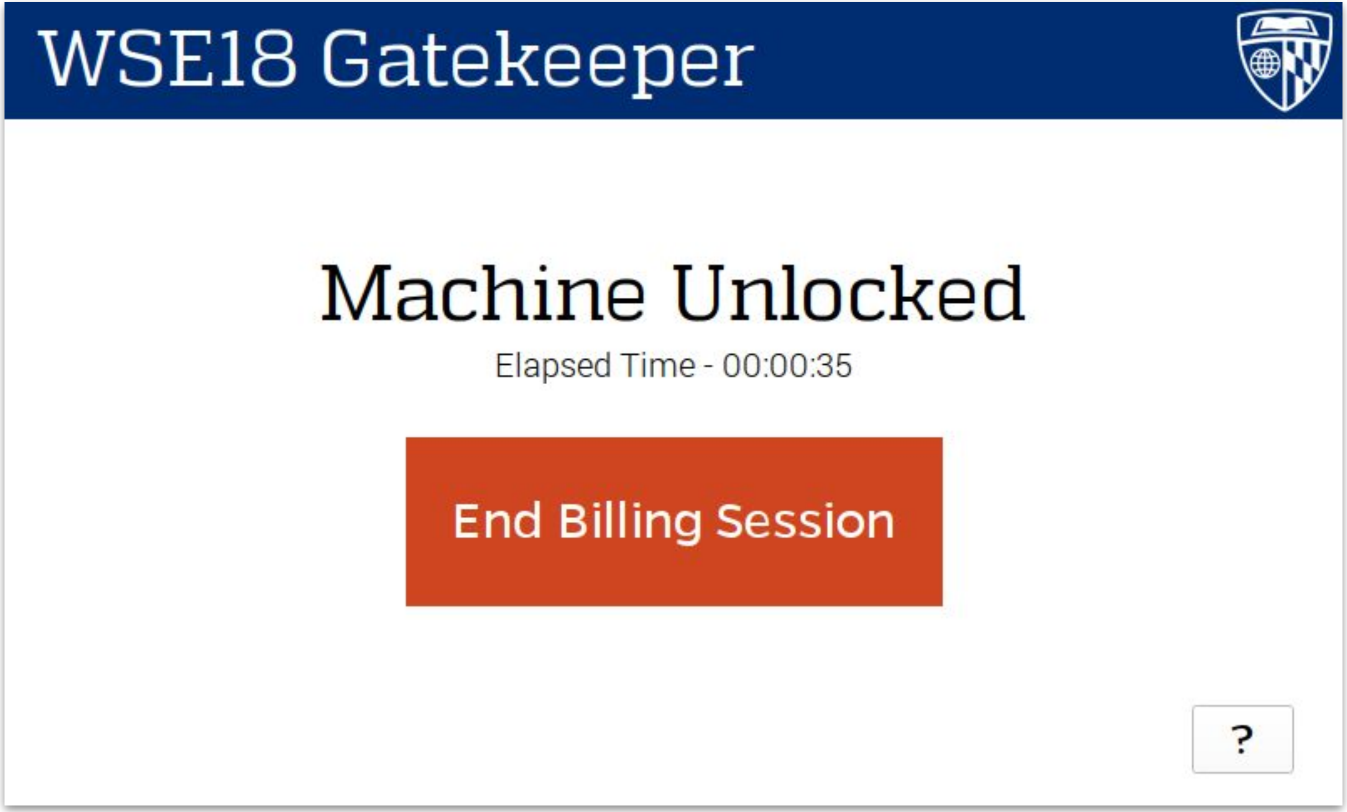
If a user has multiple machine shop budgets, they can select the correct one for the given task

Displays the name of the budget instead of the budget number (if desired, a user can instead set the budget number to display when they register at the Kiosk)

If there are more budgets than can fit, the user can scroll through the list

Selecting a budget unlocks the machine

GUI Screen: Machine Unlocked



WSE18 Gatekeeper

Machine Unlocked

Elapsed Time - 00:00:35

Machine Unlocked Screen Explanation

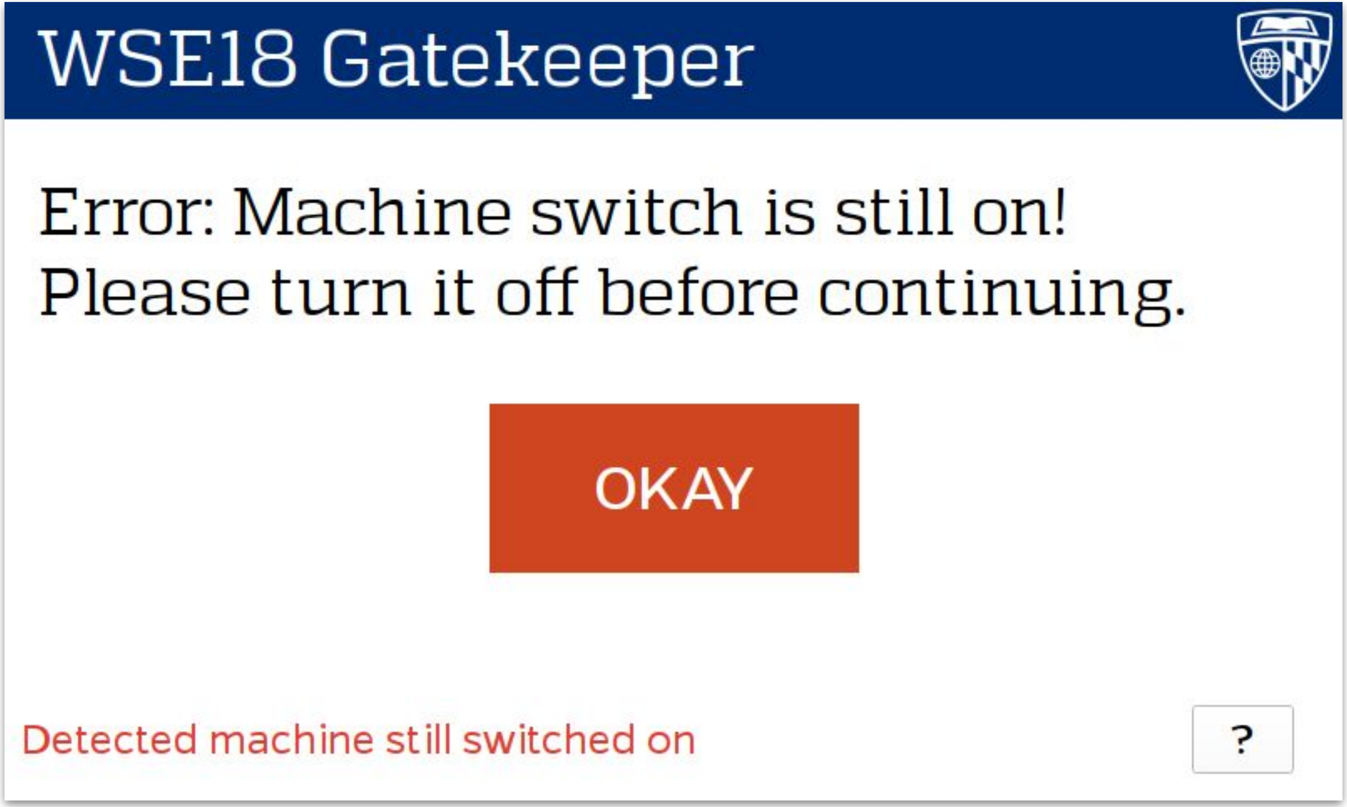
Once a budget is selected, the machine is unlocked

Display how long the current billing session is

Allow the user to end the billing session

- If the machine is off, lock the machine and move to the transaction summary
- Otherwise display a warning message that the machine is not turned off

GUI Screen: Machine Switch On



WSE18 Gatekeeper

Error: Machine switch is still on! Please turn it off before continuing.

Detected machine still switched on

Machine Switch On Screen Explanation

This screen is displayed in two possible situations:

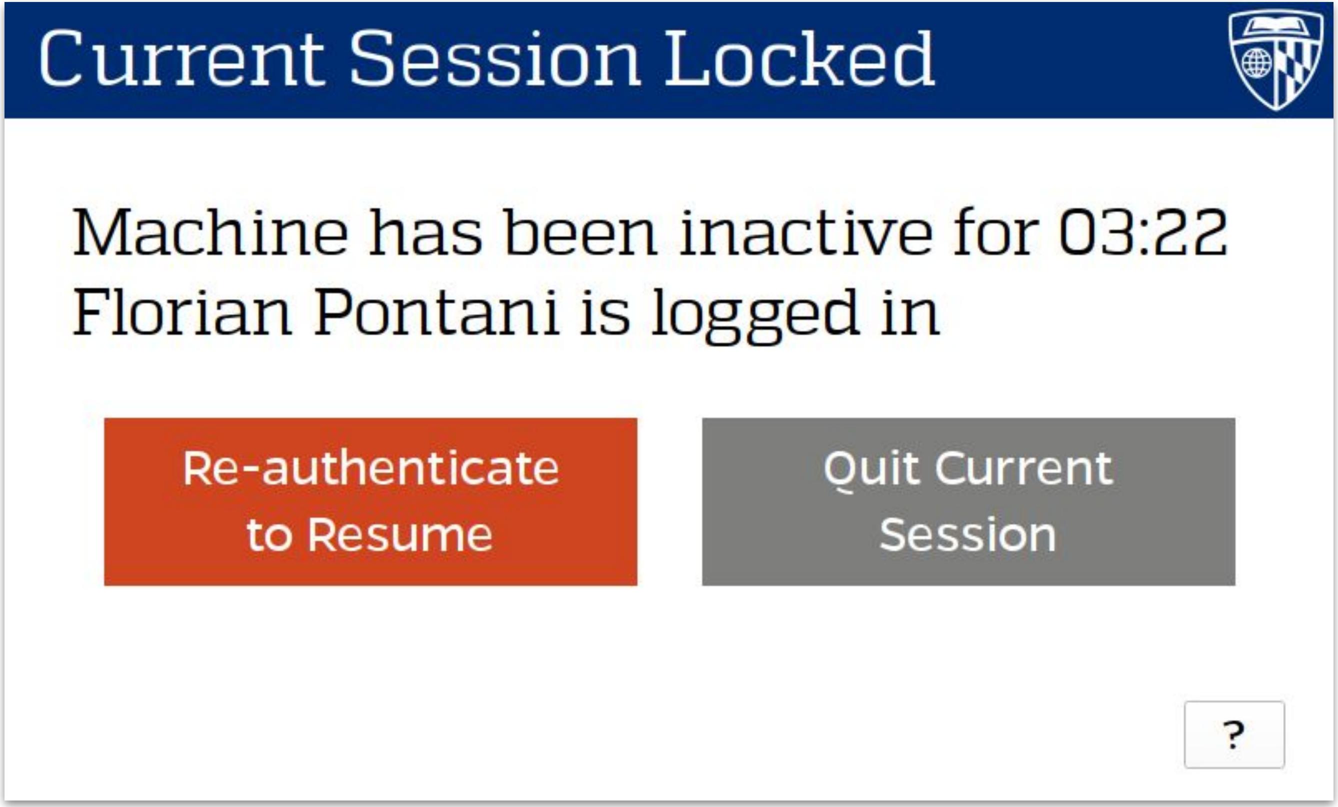
- A user tries to unlock a machine when the power switch is in the “On” position
- A user tries to end the billing session when the machine is unlocked and running

This screen informs the user of this dangerous situation, and directs them to turn the machine off before proceeding

A user cannot start a billing session until the machine is switched off

A user cannot end their billing session until the machine is completely powered off

GUI Screen: User Away



Current Session Locked

Machine has been inactive for 03:22
Florian Pontani is logged in

User Away Screen Explanation

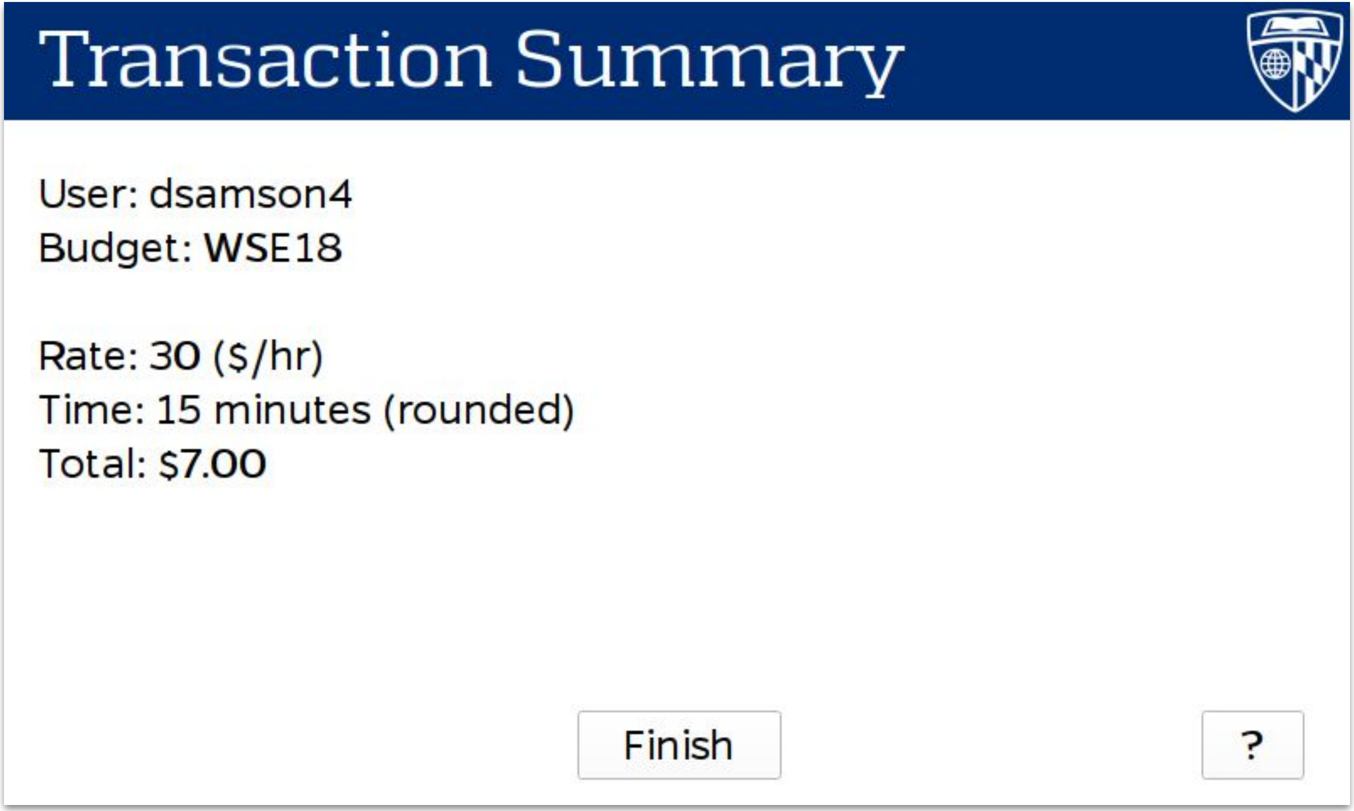
This screen is used to lock a machine if it becomes inactive during a session

This screen appears if a session is logged in and the machine hasn’t been turned on for at least 3 minutes

At this screen, the currently logged in user (or an admin) can authenticate to continue the session, or the session can be quit by anyone

Reauthenticating will take you to the machine in use screen, while quitting will take you to the transaction summary screen

GUI Screen: Transaction Summary



User: dsamson4
Budget: WSE18

Rate: 30 ($/hr)
Time: 15 minutes (rounded)
Total: $7.00

Transaction Summary Screen Explanation

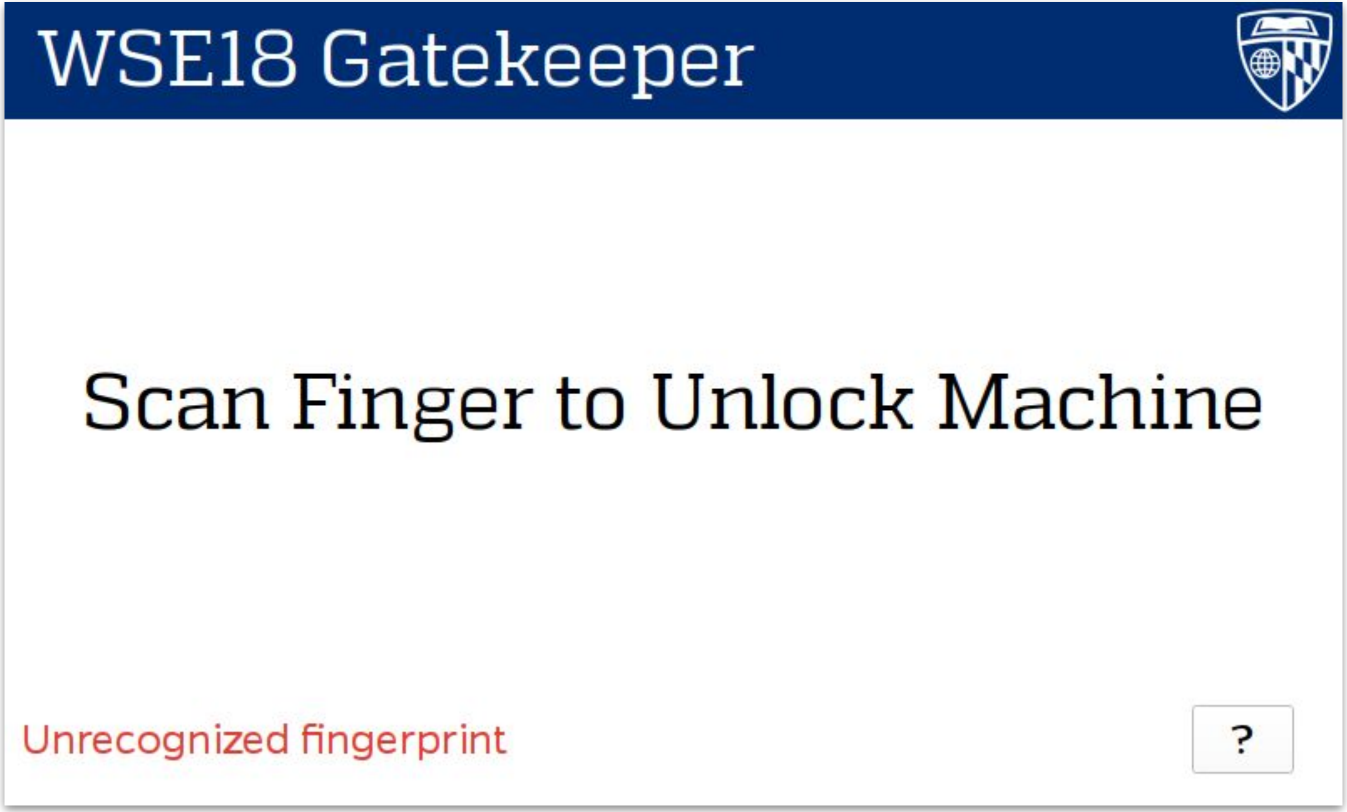
Display a summary after machine use is complete

- Who used the machine
- What budget was billed
- Machine unit cost
- Machine usage time (rounded up to the nearest 15 minutes)
- Total amount billed

In the background, the transaction is logged to the database

After this screen is displayed, the Gatekeeper resets for a new session (i.e. a new user can authenticate and use the machine)

GUI Message: Unrecognized Fingerprint



WSE18 Gatekeeper

Scan Finger to Unlock Machine

Unrecognized fingerprint

Unrecognized Finger Message Explanation

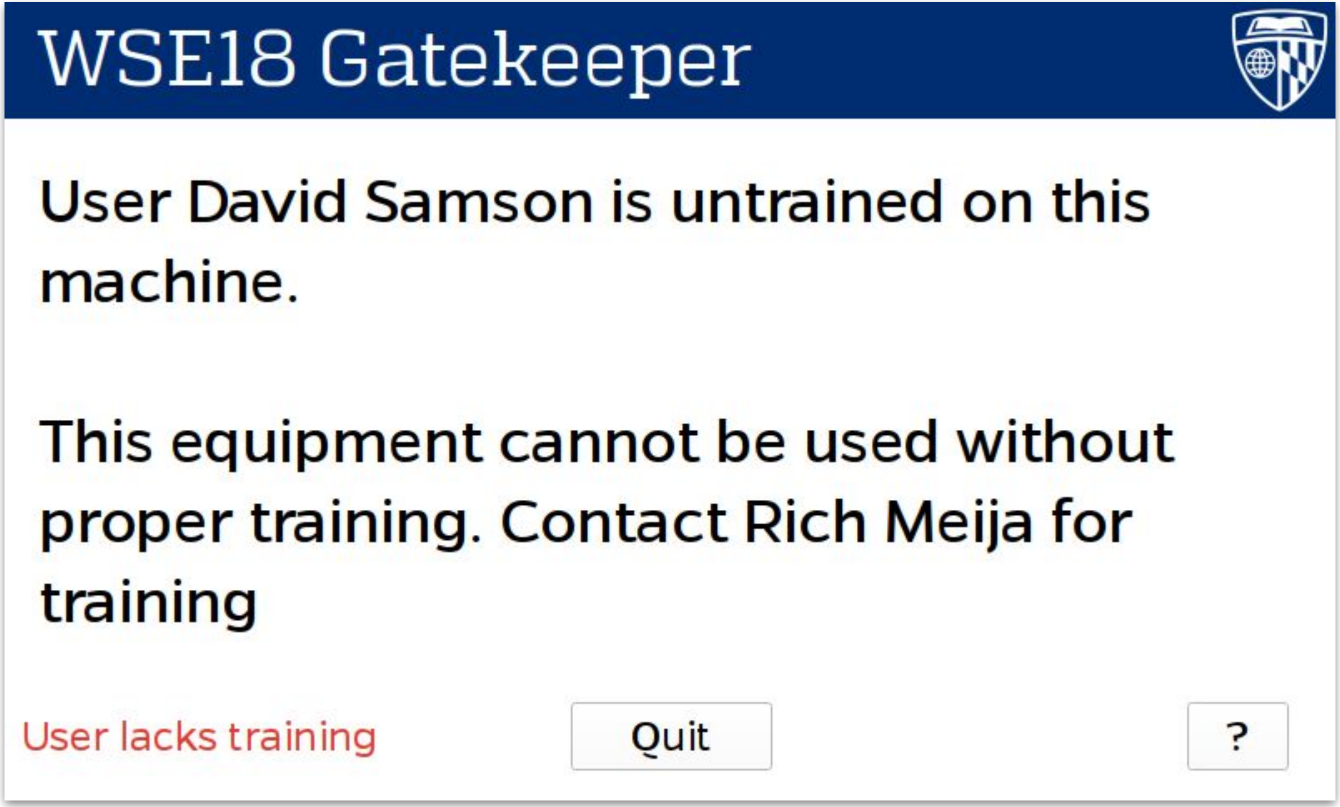
This error message is displayed whenever user scans a fingerprint that is unrecognized

Possible causes:

- The user is not in the database, and thus has no matching prints in the database
- The user didn't make a good scan of their fingerprint, and the system was unable to make a match

This was originally an entire screen, but was replaced with the message at the bottom of the Present Finger screen based on feedback from live user testing

GUI Screen: Untrained User



WSE18 Gatekeeper

User David Samson is untrained on this machine.

This equipment cannot be used without proper training. Contact Rich Meija for training

User lacks training

Untrained User Screen Explanation

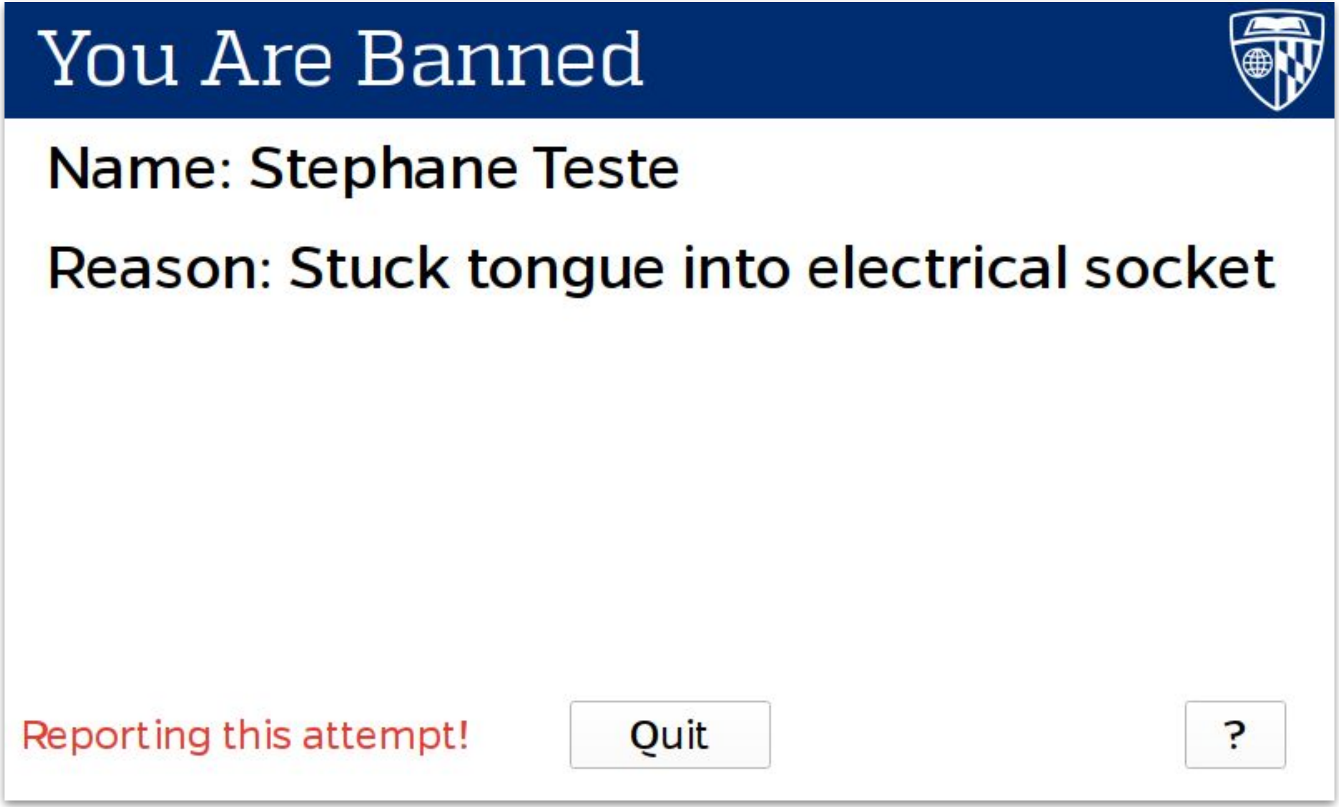
This screen is displayed if a user is recognized, but they do not have training to use the specific piece of equipment

Informs the user that they are untrained

Informs the user how they can obtain training for the equipment (This message can be updated by an administrator using the server configuration file)

After this screen, the Present Finger screen is displayed

GUI Screen: Banned User



You Are Banned

Name: Stephane Teste

Reason: Stuck tongue into electrical socket

Reporting this attempt!

Banned User Screen Explanation

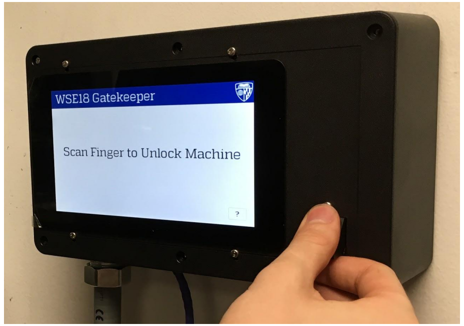
This screen is displayed if a user is recognized, but they have been banned from using the machine shop

Informs the user that they are banned

Informs the user why they are banned

After this screen, the Present Finger screen is displayed

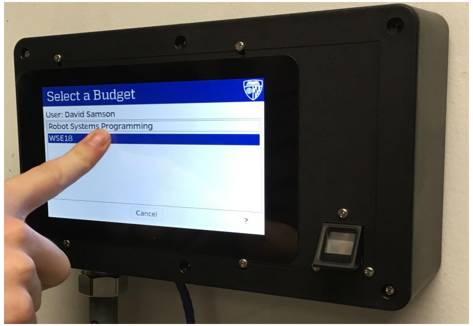
Gatekeeper GUI on Final Hardware



WSE18 Gatekeeper

Scan Finger to Unlock Machine

1. Present your finger



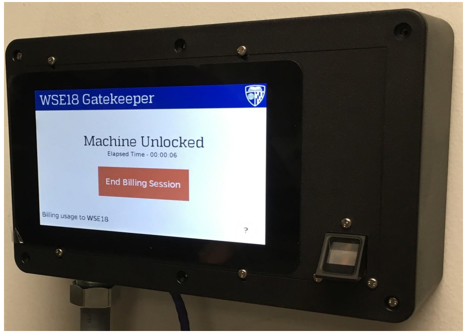
Select a Budget

User: David Samson

Robot Systems Programming

WSE18

2. If trained, pick the budget to charge



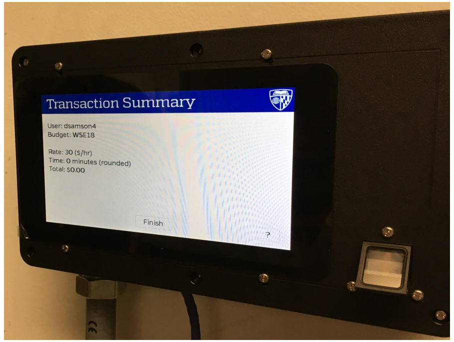
WSE18 Gatekeeper

Machine Unlocked

Elapsed Time - 00:00:06

Billing usage to WSE18

3. Use the machine



Transaction Summary

User: dsamson4
Budget: WSE18

Rate: 30 ($/hr)
Time: 0 minutes (rounded)
Total: $0.00

4. When finished, a summary is displayed

Deployment Device showing GUI screens during the regular authentication process

5.3.2: Kiosk GUI

Kiosk User Interface

The Kiosk UI provides an interface for interacting with the database, in addition to displaying machine shop information

Kiosk should be able to:

- Add new users to the database
    - Specify new budgets, machines the user is trained on etc.
- Modify existing users in the database
- Manage budgets in the system (admin only)
- Manage machines in the system (admin only)

Kiosk User Interface Implementation

The Kiosk UI software is implemented using Qt, same as the Gatekeeper

The Gatekeeper and Kiosk share much of the same codebase

The Kiosk UI follows the same visual style as the Gatekeeper UI

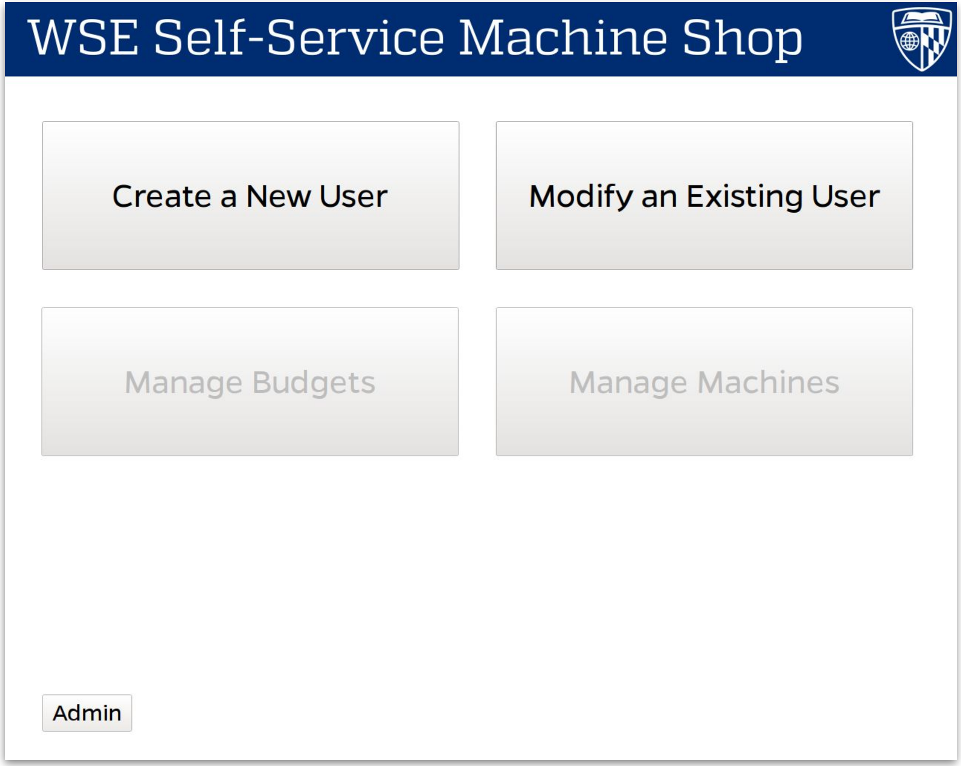
- Large easy to read fonts
- Simple design layouts with only important information
- Color schemes and fonts that meet Hopkins Brand Guidelines

The Kiosk runs on a single raspberry Pi, connected to a regular desktop monitor with mouse and keyboard peripherals

Kiosk GUI Screens

The following are the Kiosk GUI screens as they appear on an actual Kiosk device. Each screen is followed by an explanation of how it functions.

Kiosk GUI Screen: Main Menu



Main Menu Screen Explanation

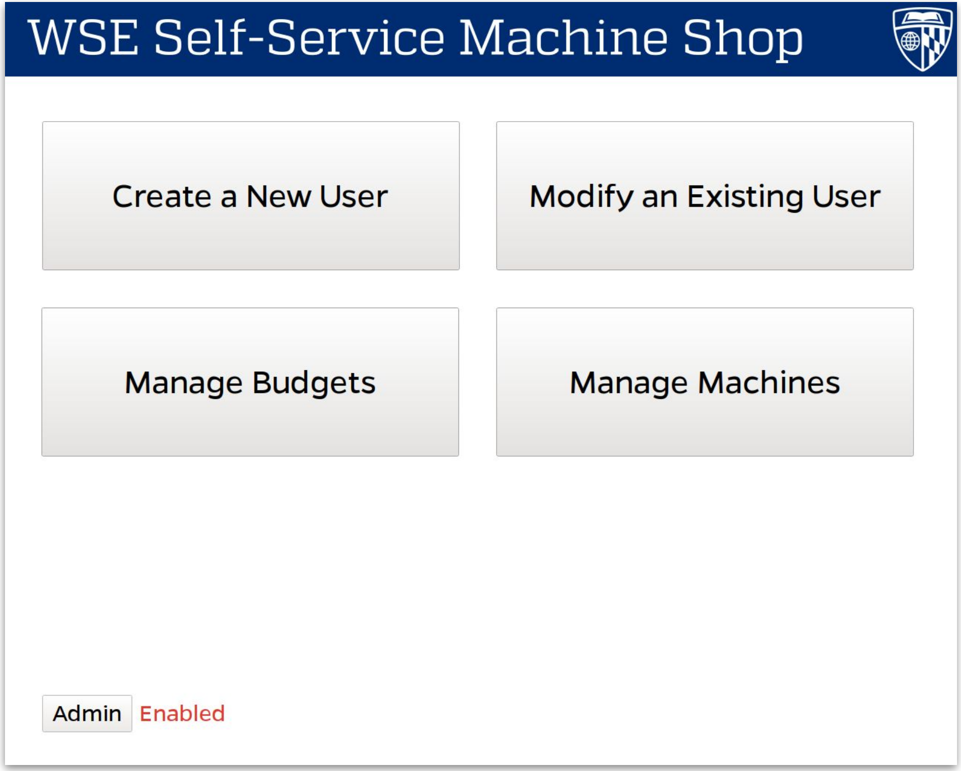
This is the home screen for the kiosk

Four large buttons allow a user to select the task the would like to do

- Create a new user
- Modify an existing user
- Manage Budgets (admin only)
- Manage Machines (admin only)

Additionally, in the lower left there is an “Admin” button which allows an admin to enable admin mode by verifying their fingerprint

Kiosk GUI: Admin Mode



WSE Self-Service Machine Shop

Admin Mode Explanation

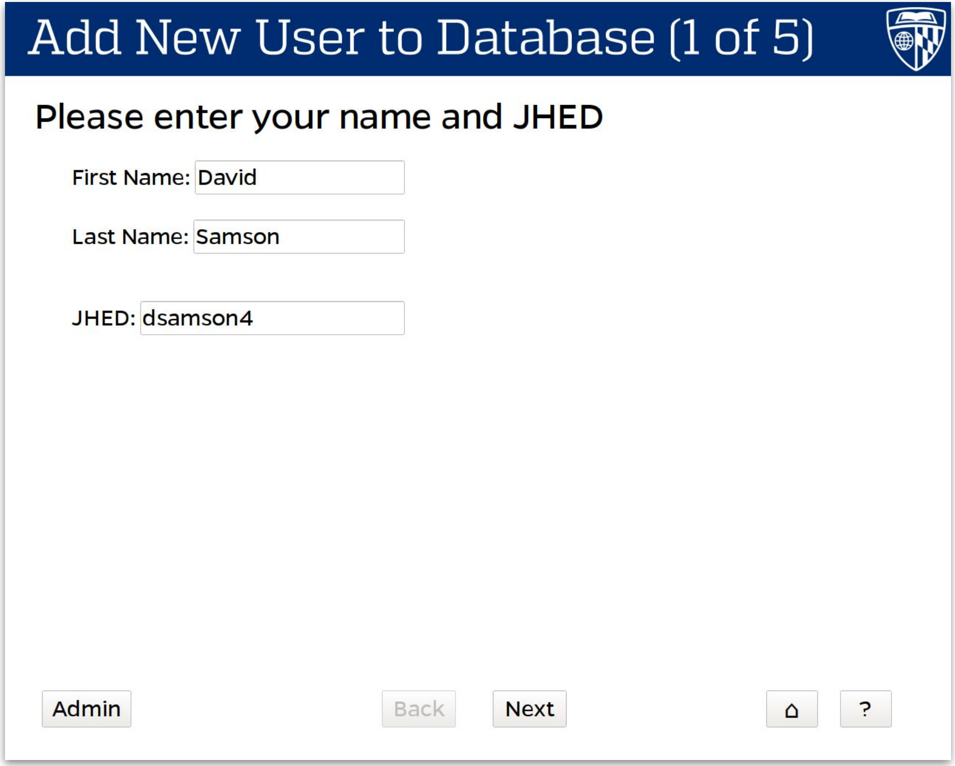
Because certain functionality of the kiosk should be limited to use by only the shop administrators, the kiosk provides a mode available only to admins which allows access to such functions.

To initialize admin mode, from any screen push the “Admin” button in the lower left. The fingerprint sensor will then attempt to verify an admin. If successfully verified, the text “Enabled” will appear next to the button indicating that admin mode is on. Admin mode can be quit by pushing the button again.

Admin mode allows access to the Manage Budgets, and Manage Machines screens, as well as providing extra options during the modify user process.

In admin mode, no fingerprint verification is required for any other processes

Kiosk GUI Screen: Create New User (1 of 5)



Add New User to Database (1 of 5)

Please enter your name and JHED

First Name: David

Last Name: Samson

JHED: dsamson4

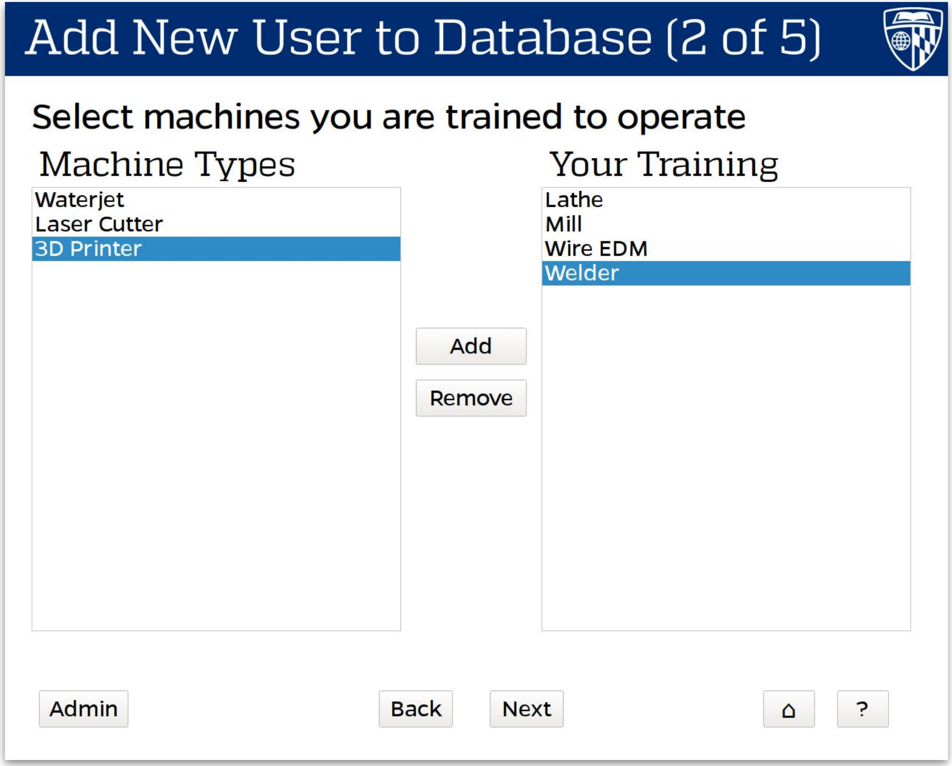
Create New User (1 of 5) Screen Explanation

This screen is reached when Create New User is selected from the main menu

This screen has fields for entering a new user’s name, and JHED

During the user creation process, the Next button will continue to the next step, and the back button will take them to the previous step. Additionally the home button will return to the home screen

Kiosk GUI Screen: Create New User (2 of 5)



Add New User to Database (2 of 5)

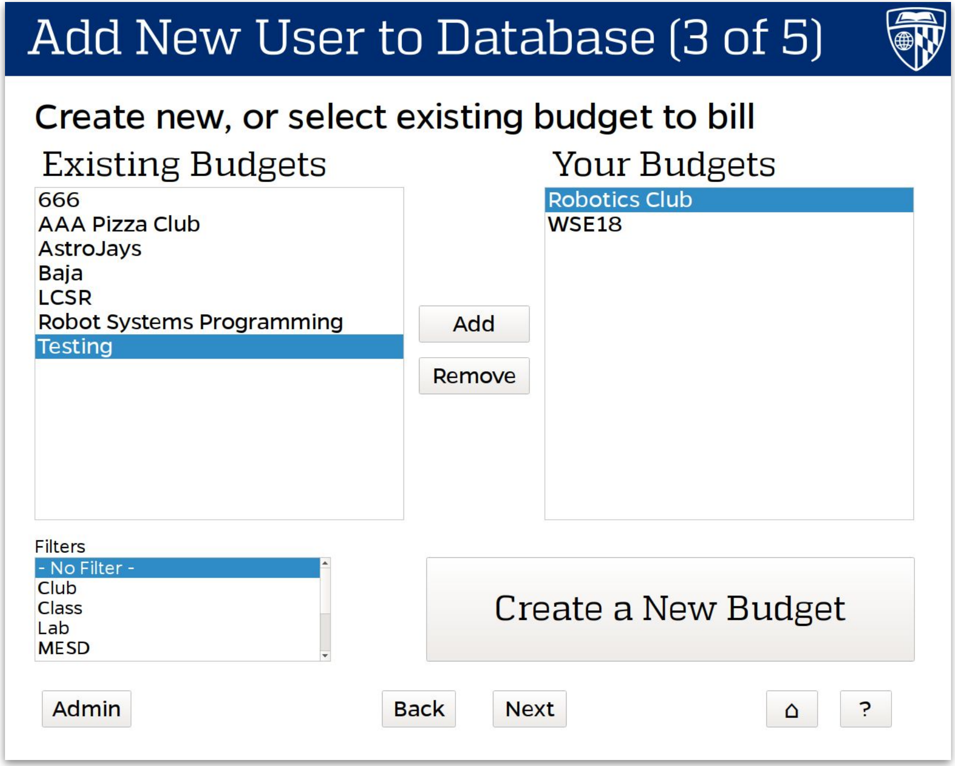
Select machines you are trained to operate

Create New User (2 of 5) Screen Explanation

The second screen in the user creation process controls which machines a user is allowed to use (i.e. what training do they posses)

Machine types can be selected from the list on the left and added to the user’s list of trainings on the right

Kiosk GUI Screen: Create New User (3 of 5)



Add New User to Database (3 of 5)

Create new, or select existing budget to bill

Existing Budgets

- 666
- AAA Pizza Club
- AstroJays
- Baja
- LCSR
- Robot Systems Programming
- Testing

Your Budgets

- Robotics Club
- WSE18

Filters

- - No Filter -
- Club
- Class
- Lab
- MESD

Create New User (3 of 5) Screen Explanation

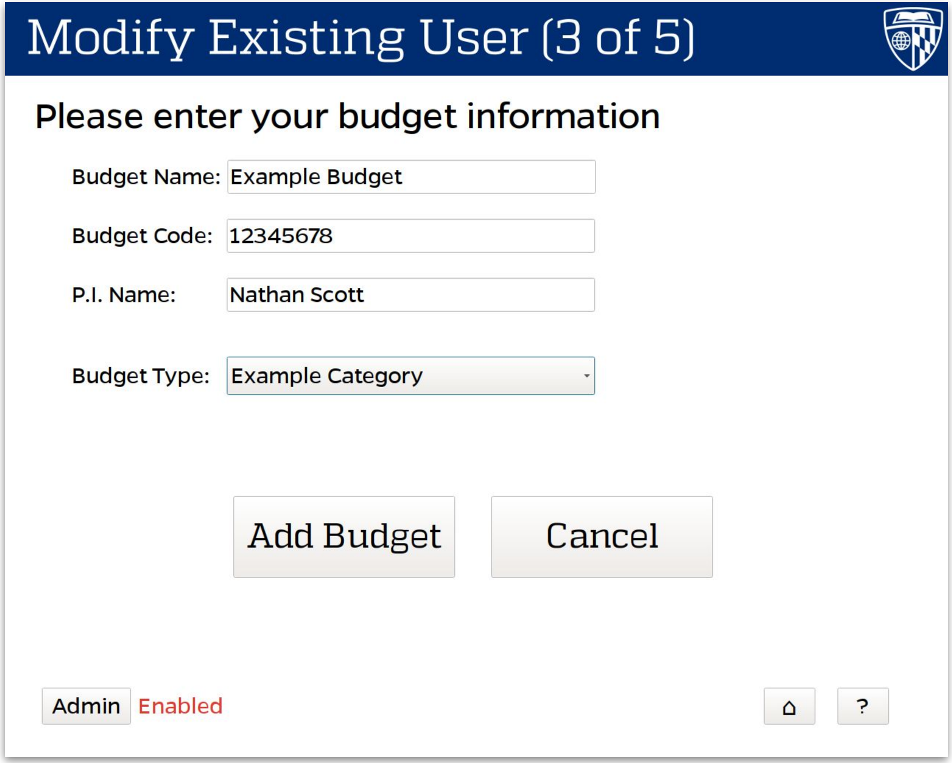
The third screen in the user creation process controls what budgets a user has access to for billing machine usage

Budgets can be selected from the list on the left, and added to the user’s own list on the right

The budgets can be filtered by selecting a budget category from the lower left list

Additionally if a the user’s budget does not exist in the list, it can be created with the Create a New Budget button

Kiosk GUI Screen: Create New Budget



Modify Existing User (3 of 5)

Please enter your budget information

Budget Name: Example Budget

Budget Code: 12345678

P.I. Name: Nathan Scott

Budget Type: Example Category

Create New Budget Screen Explanation

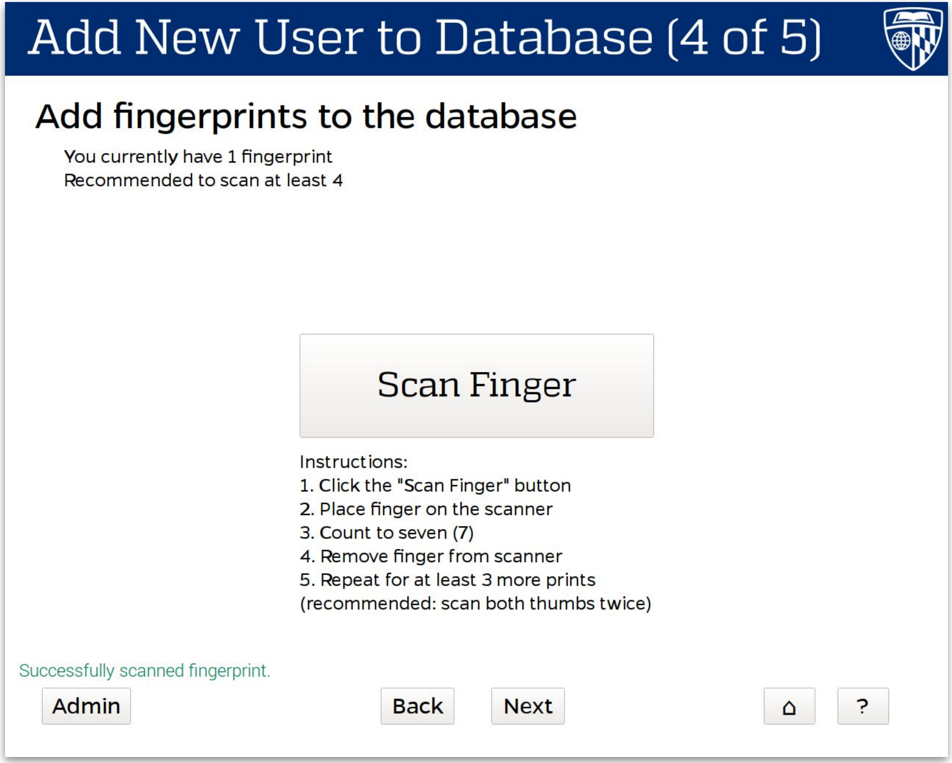
This screen handles the creation of new budgets

A user specifies the budget name, budget code, PI, and a category for the system to use to sort the budget with

If admin mode is active, the ability to create new budget categories is enabled

Creating a new budget on this screen from during the user creation process will add it to the user’s list of budgets

Kiosk GUI Screen: Create New User (4 of 5)



Add New User to Database (4 of 5)

Add fingerprints to the database

You currently have 1 fingerprint
Recommended to scan at least 4

Instructions:
1. Click the "Scan Finger" button
2. Place finger on the scanner
3. Count to seven (7)
4. Remove finger from scanner
5. Repeat for at least 3 more prints
(recommended: scan both thumbs twice)

Successfully scanned fingerprint.

Create New User (4 of 5) Screen Explanation

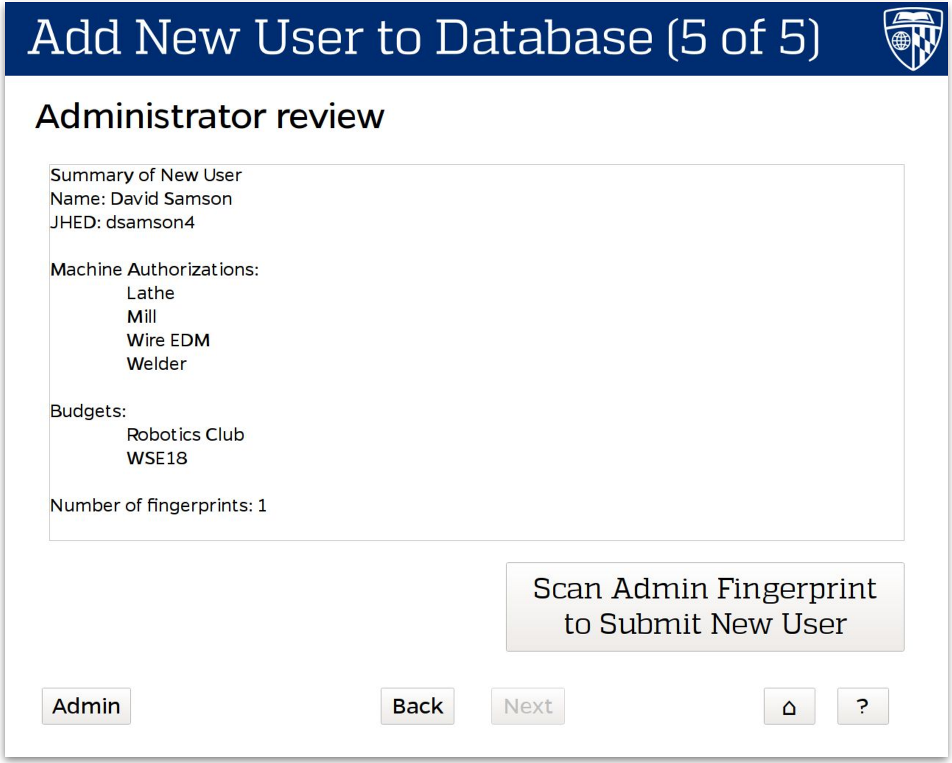
The fourth screen in the user creation process manages adding a user’s fingerprints to the database

The Scan Finger button is pressed, and then the fingerprint sensor flashes green indicating that it is ready to scan a fingerprint

This process can be completed for as many fingerprints as the user wishes. The more prints, the more likely the system will recognize that user

The user is required to scan at least one fingerprint before being allowed to continue

Kiosk GUI Screen: Create New User (5 of 5)



Add New User to Database (5 of 5)

Administrator review

Summary of New User
Name: David Samson
JHED: dsamson4

Machine Authorizations:
- Lathe
- Mill
- Wire EDM
- Welder

Budgets:
- Robotics Club
- WSE18

Number of fingerprints: 1

Create New User (5 of 5) Screen Explanation

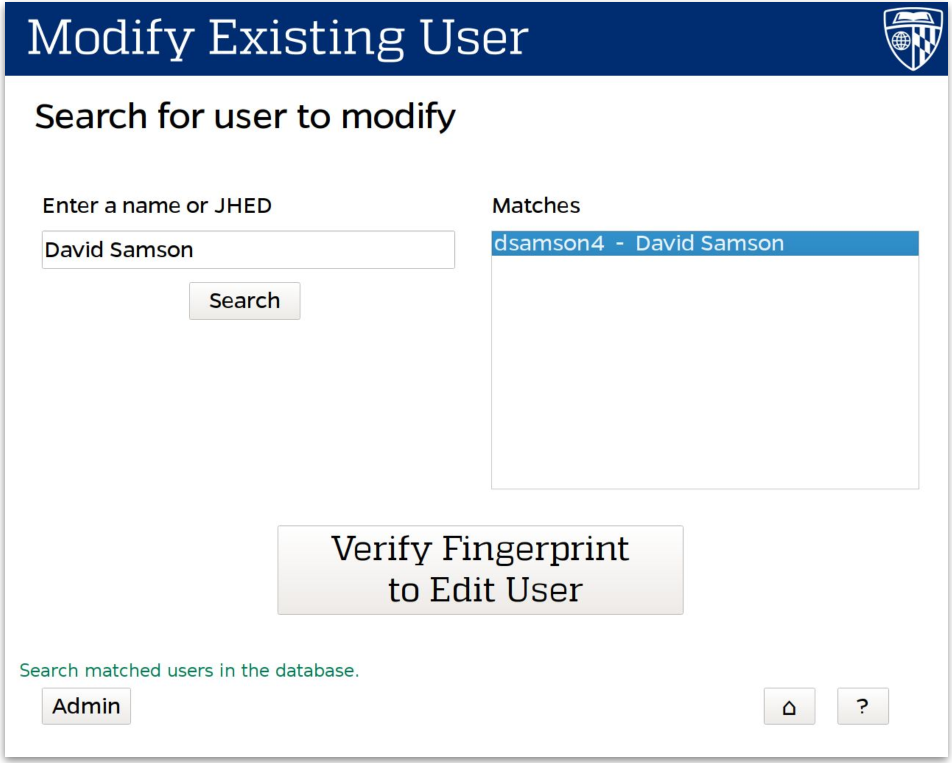
The final screen in the user creation process allows an admin to verify that the user information is correct before the new user can be added to the database

A summary of all user information is displayed

The submit button requires an admin to verify their fingerprint. This mainly ensures that the user has not added training for machines that they have not actually been trained to use

Once the user is submitted to the database, the Kiosk flashes a success message, and then goes to the home screen

Kiosk GUI Screen: Search for Existing Users



Modify Existing User

Search for user to modify

Enter a name or JHED

David Samson

Matches

dsamson4 - David Samson

Search matched users in the database.

Search for Existing Users Screen Explanation

This screen is displayed when the Modify Existing User button is selected from the home screen

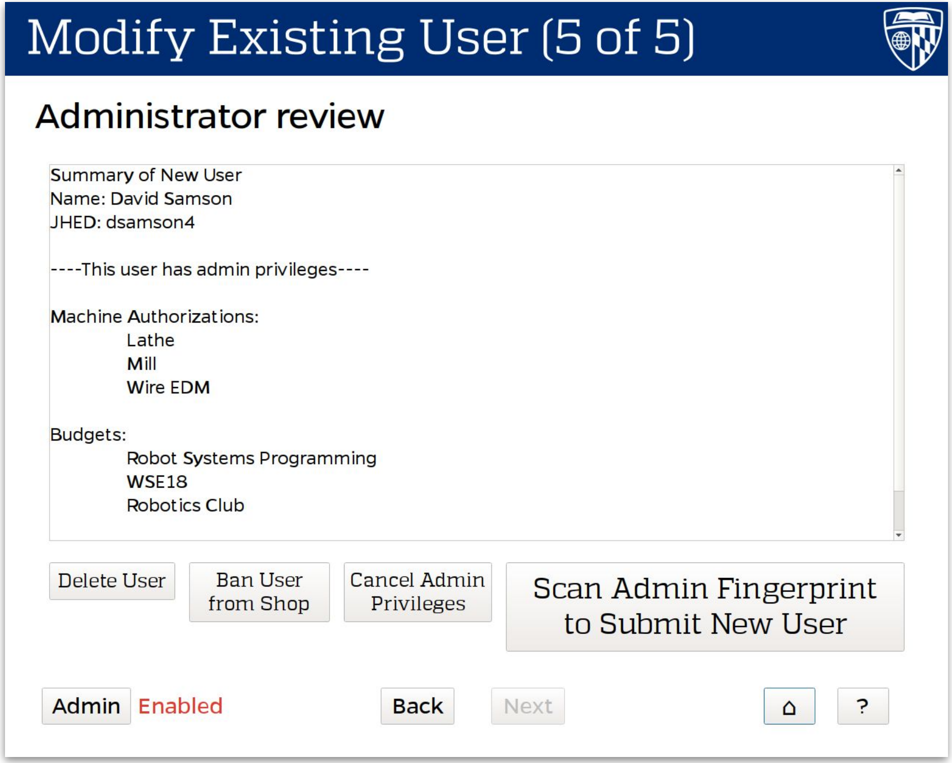
This screen handles searching for what user to modify

A user can be identified by their full name or JHED. The search must match exactly the information in the database, but is case insensitive

Users can only be modified by themselves or admins. When a user is selected for modification, either that particular user, or an admin must verify their fingerprint. Either way, an admin must authenticate to submit changes to the database.

The screens to modify a user are identical to the user creation screen, however they contain the user's information pre-populated in each data field

Kiosk GUI Screen: Admin Modify User (5 of 5)



Administrator review

Summary of New User
Name: David Samson
JHED: dsamson4

----This user has admin privileges----

Machine Authorizations:
Lathe
Mill
Wire EDM

Budgets:
Robot Systems Programming
WSE18
Robotics Club

Admin Modify User (5 of 5)

The fifth screen in the modify user process has extra options available if admin mode is enabled

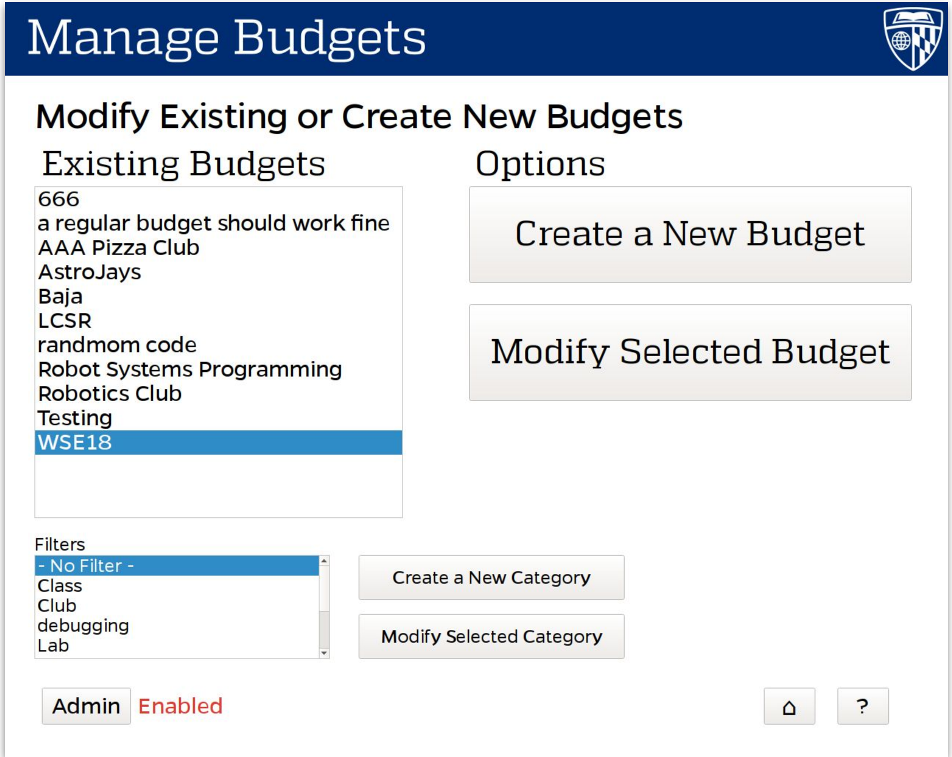
Delete User allows the user to be removed from the system

Ban User from Shop allows the user to be banned, and a dialog appears that prompts for an explanation of why they were banned. If the user is already banned, this will be an Unban User from Shop button

Grant Admin Privileges allows a normal user to be converted into an admin. If the user is already an admin this will be a Cancel Admin Privileges button

Again, in order to submit any changes to a user, an admin must be verified

Kiosk GUI Screen: Manage Budgets



Manage Budgets

Modify Existing or Create New Budgets

Existing Budgets

- 666
- a regular budget should work fine
- AAA Pizza Club
- AstroJays
- Baja
- LCSR
- randmom code
- Robot Systems Programming
- Robotics Club
- Testing
- WSE18

Options

Filters

- - No Filter -
- Class
- Club
- debugging
- Lab

Admin Enabled

Manage Budgets Screen Explanation

This screen manages all budgets stored in the database

This screen can be reached if admin mode is enabled and the Manage Budgets button is selected from the home screen

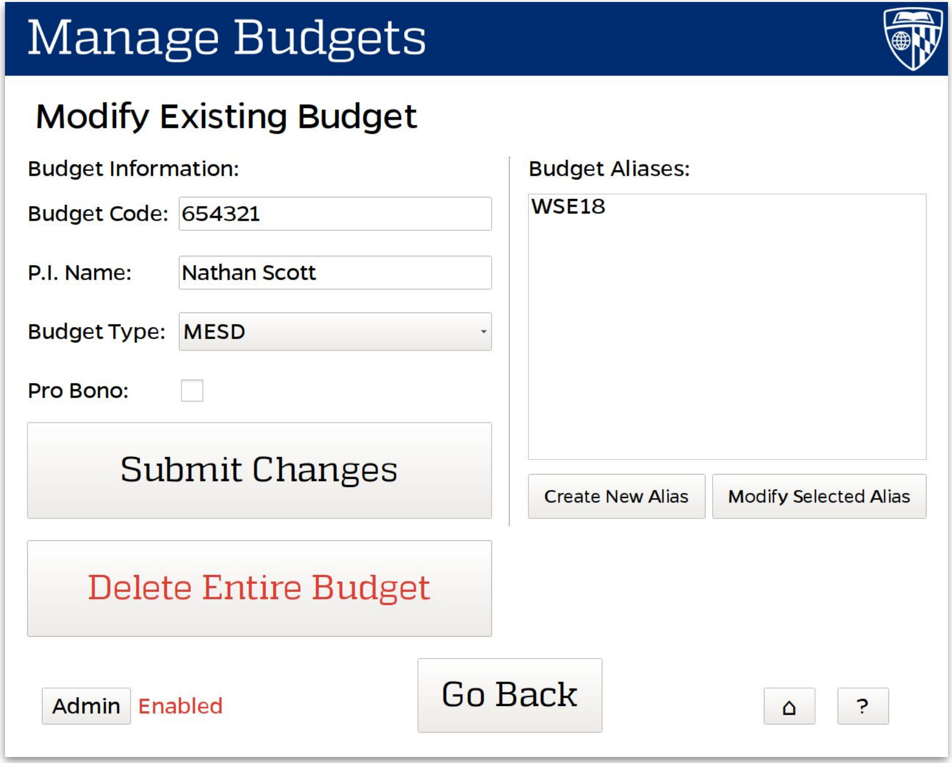
All budgets in the database are listed to the left, and the list can be filtered by selecting a budget category from the list of filters

The Create New Budget takes the user to the budget creation screen

The Manage Existing Budget takes the user to a screen to modify information about an existing budget

Budget categories can be created or modified from this screen

Kiosk GUI Screen: Modify Budget



Modify Existing Budget

Budget Information:

Budget Code: 654321

P.I. Name: Nathan Scott

Budget Type: MESD

Pro Bono: □

Budget Aliases:

WSE18

Modify Budget Screen Explanation

This screen allows the modification of information about a specific budget

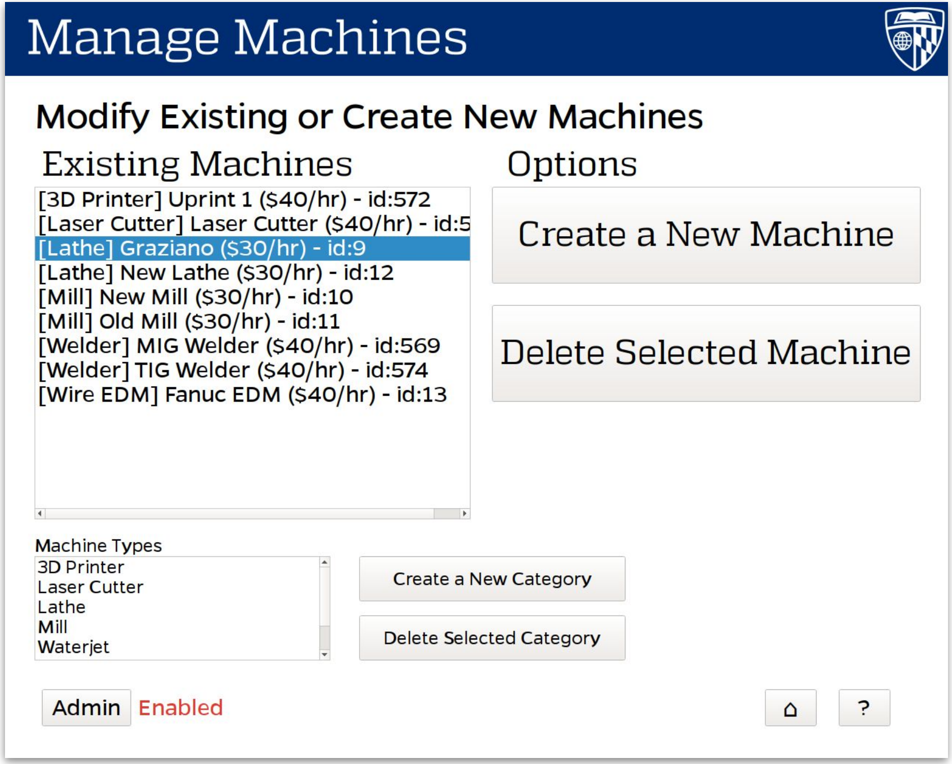
Budget code, PI, and category can be updated

A budget can be marked as pro bono (this is just cosmetic, and doesn’t affect any functionality for when a gatekeeper bills a budget)

A budget can be deleted

The list of names that refer back to the budget (aliases) can be modified

Kiosk GUI Screen: Manage Machines



Manage Machines

Modify Existing or Create New Machines

Existing Machines

[3D Printer] Uprint 1 ($40/hr) - id:572
[Laser Cutter] Laser Cutter ($40/hr) - id:5
[Lathe] Graziano ($30/hr) - id:9
[Lathe] New Lathe ($30/hr) - id:12
[Mill] New Mill ($30/hr) - id:10
[Mill] Old Mill ($30/hr) - id:11
[Welder] MIG Welder ($40/hr) - id:569
[Welder] TIG Welder ($40/hr) - id:574
[Wire EDM] Fanuc EDM ($40/hr) - id:13

Options

Machine Types

3D Printer
Laser Cutter
Lathe
Mill
Waterjet

Admin Enabled

Manage Machines Screen Explanation

This screen manages the list of machines in the database

This screen can be reached if admin mode is enabled, and the Manage Machines button is selected on the home screen

The create new machines button creates a dialog that prompts the user for the machine name, machine type, and billing rate

The list of machine types (i.e. what types of training users can have) can be updated with Create New Machine Category and Delete Selected Category

Machine IDs are listed to allow an admin to refer to a specific machine when setting up a new Gatekeeper device for that machine

Gatekeeper/Kiosk GUI General Design Notes

The Gatekeeper and Kiosk GUIs follow the same design principles, in addition to sharing most of the same codebase

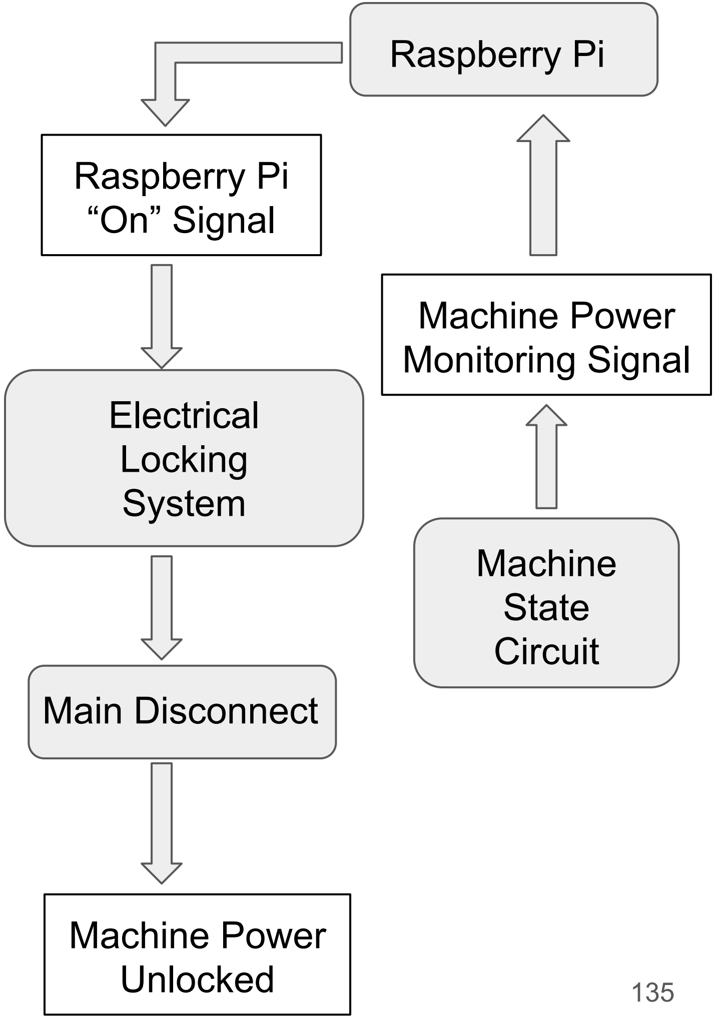
- Large text that is easy to read
- Simple uncluttered design
  - Communicates only essential information
- Design meets Hopkins Visual Brand Guidelines
  - University logo properly displayed (correct color, background contrast, spacing, etc.)
  - Colors used in UI are all official hopkins colors
  - Fonts are official Hopkins fonts Quadon and Gentona (not reproduced in this presentation)
- Colored status messages for warning/success messages
- Help button on every screen
  - Displays helpful information about what each screen does

Section 6: Electrical System

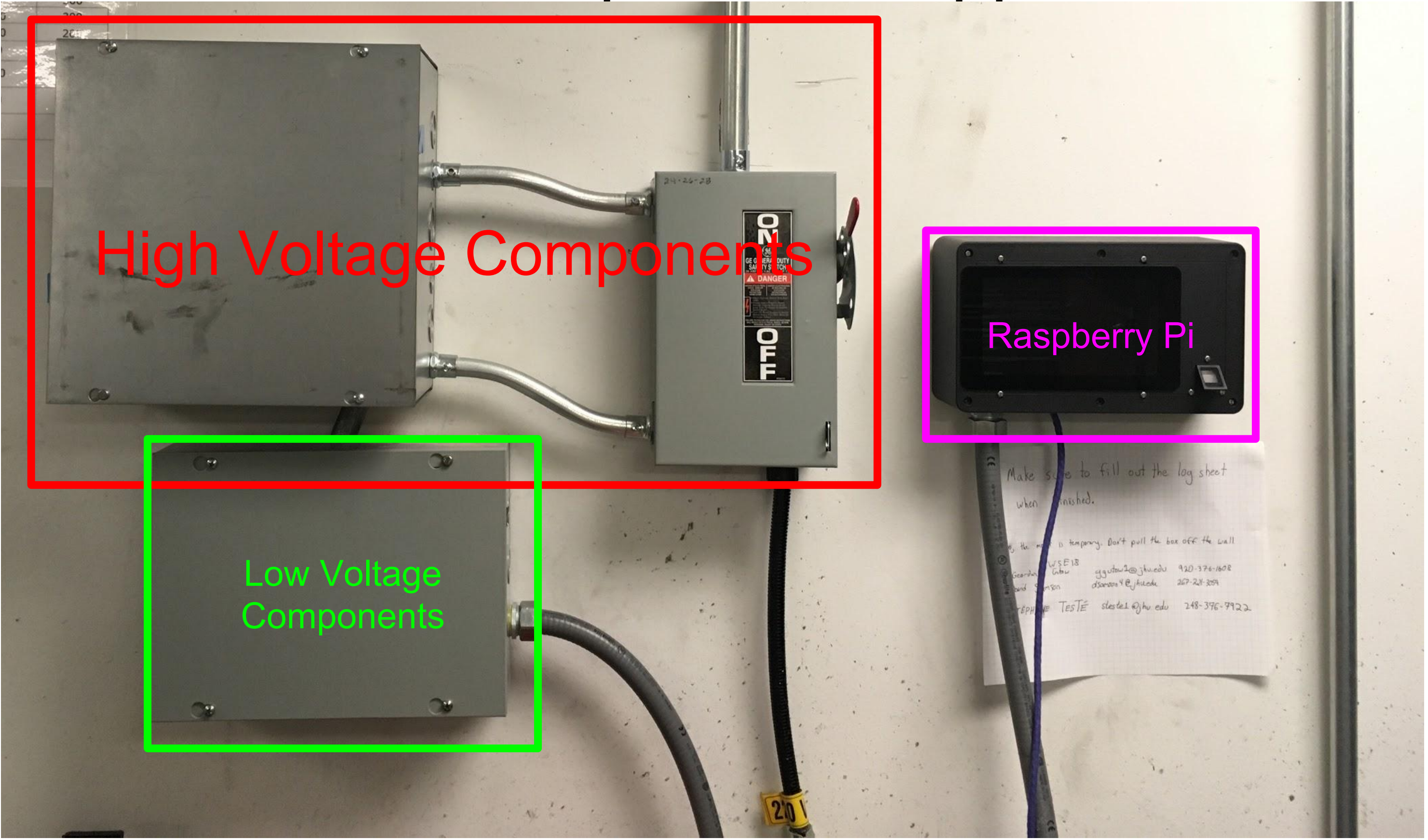
Electrical System Requirements

Requirements

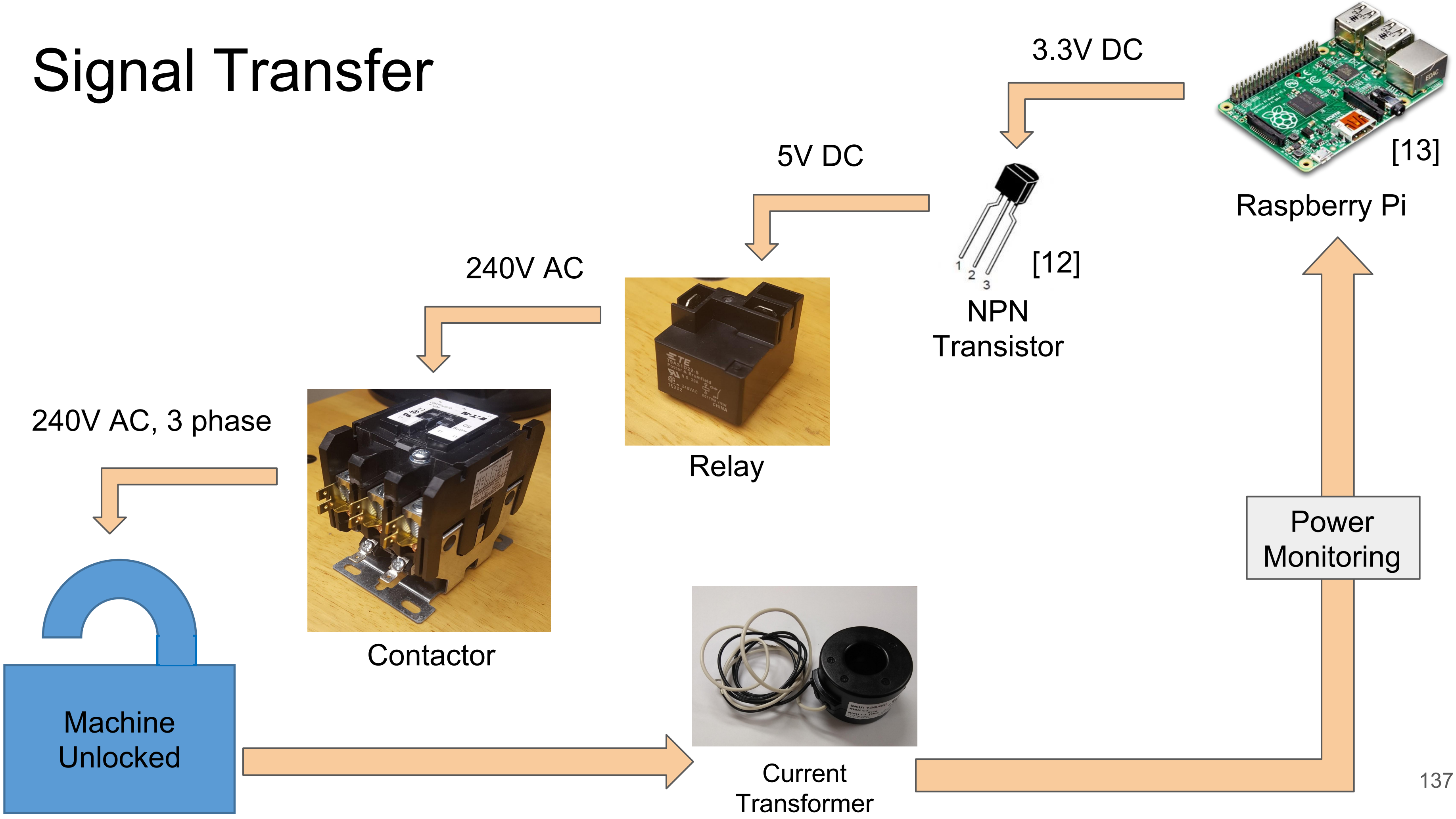
- Switch machine power based on Raspberry Pi
    - Input from Pi 3.3V, 6mA
    - Output to machine is 240V, 60A
- Monitors machine status
    - Detect when power drawn by machine
    - Avoid disconnecting while machine in use
- Does not interfere with current safety protocols
    - Lockout/Tagout
    - Mechanical circuit breaker avoids electrical anomalies
        - Machine draws significant power



Fully Installed Gatekeeper and Support Electronics

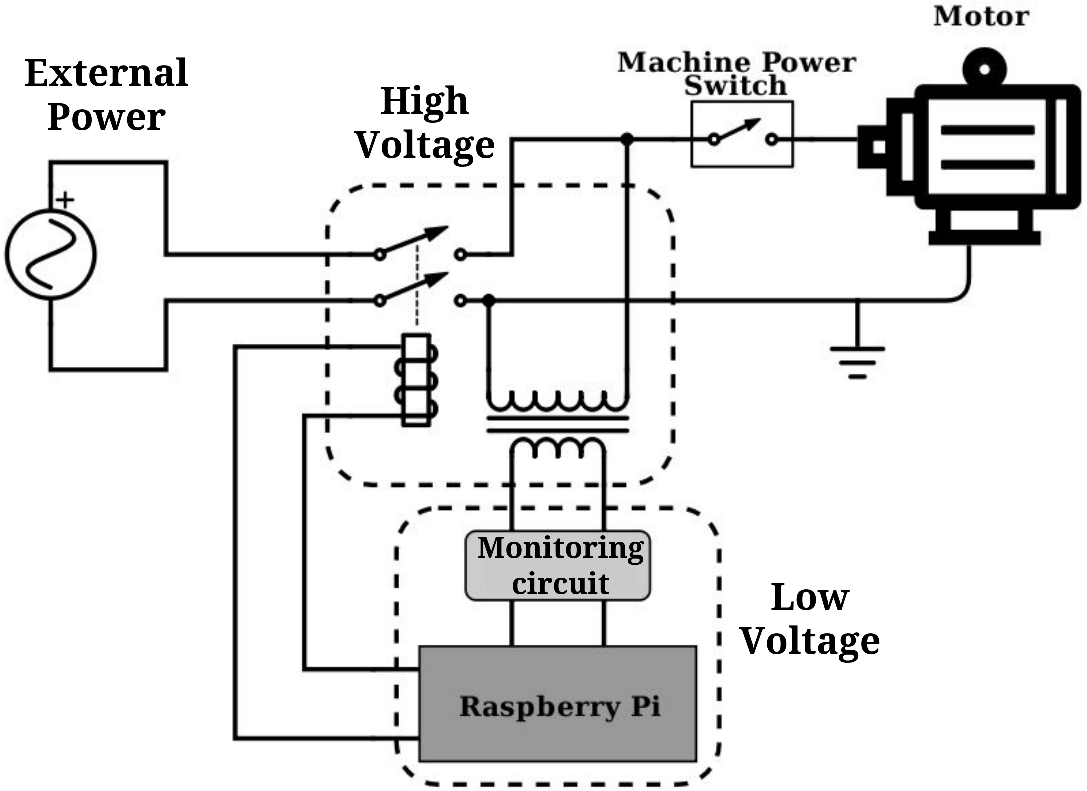

High Voltage Components
Low Voltage Components
Raspberry Pi
ON
OFF
DANGER
Make sure to fill out the log sheet
when finished.
the is temporary. Don't pull the box off the wall
WSE18
Gutow ggutow1@jhu.edu 920-376-1608
dsamson4@jhu.edu 267-2?-3054
TESTÉ stestel@jhu.edu 248-376-7922
20

Signal Transfer



Electrical System

- Pi activates contactor
- Power becomes available to machine
- User uses the machine as normal
- Pi monitors power with monitoring circuit



Electrical System

The Raspberry Pi’s sends out a 3.3V signal to a transistor. This switches closed an intermediate circuit powering a relay. The relay in turn closes the circuit powering the contactor. Finally, the contactor unlocks the machine’s power, allowing it to be used.

Adjoining is the power monitoring circuit which is completely separate from the machine’s on/off signal circuit. It is however connected to the same power supply as the machine. A current transformer monitors a single 120V lead of the 3-phase wires. The signal is rectified, filtered, and processed by a custom PCB before being delivered to the Pi as a 3.3V logical signal that is high when the machine is off.

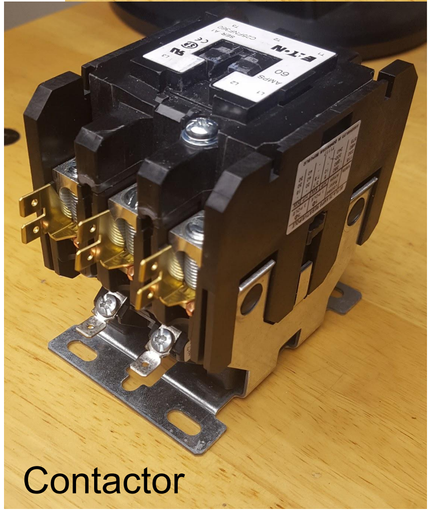
Apart from the Pi’s 3.3V logical signals, all low voltage signals in the electrical circuitry are at 5V.

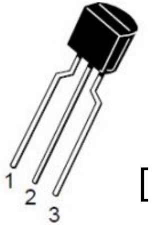
System Specifications

- There are 4 levels of signal power used in the circuit to ultimately control the machine. Here are specifications for which each critical piece of hardware is designed to handle.
  - Current Transformer
    - 240V, 100A
  - Contactor
    - 240V, 60A
  - Relay
    - 240V, 1A
  - NPN Transistor
    - 1 Watt, 5V
  - Digital Input (Raspberry Pi)
    - 3.3V, 8mA

Relay

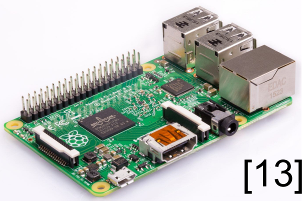
Current Transformer


Contactor



[12]

NPN Transistor


[13]

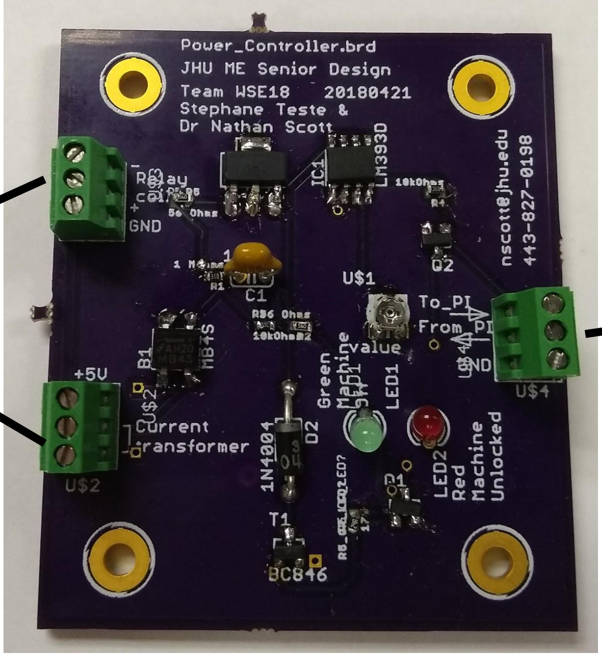
Raspberry Pi

System Specifications (cont.)

- Type of power needed by system
    - Constant power, even while machine not on- several components must have power delivered at all times without interfering with the safety measures.
    - Additional power draw occurs intermittently
        - The machine may not always be on but its immediate power cannot be hampered in any way by our system.
        - Logic Signals will add to the system power demands and may deplete the available power for low voltages.
- Insulation requirements for each location
    - High voltages require a minimum of 300V insulation.
        - Required insulation within the high voltage box
        - All connections between containers must use this insulation strength
    - For the range of lower voltages that is being handled, 75V insulation is required.
        - Limited to the low Voltage box

Low Voltage Control System

- Digital signal input
  - From Raspberry Pi
  - Unlocks transistor
- 1W (5V) intermediate signal
  - Controlled by transistor
  - Unlocks relay
- 240V signal output
  - Controlled by relay
  - Unlocks contactor
- LED status light
  - Alerts technician system status
  - Decreases safety risk

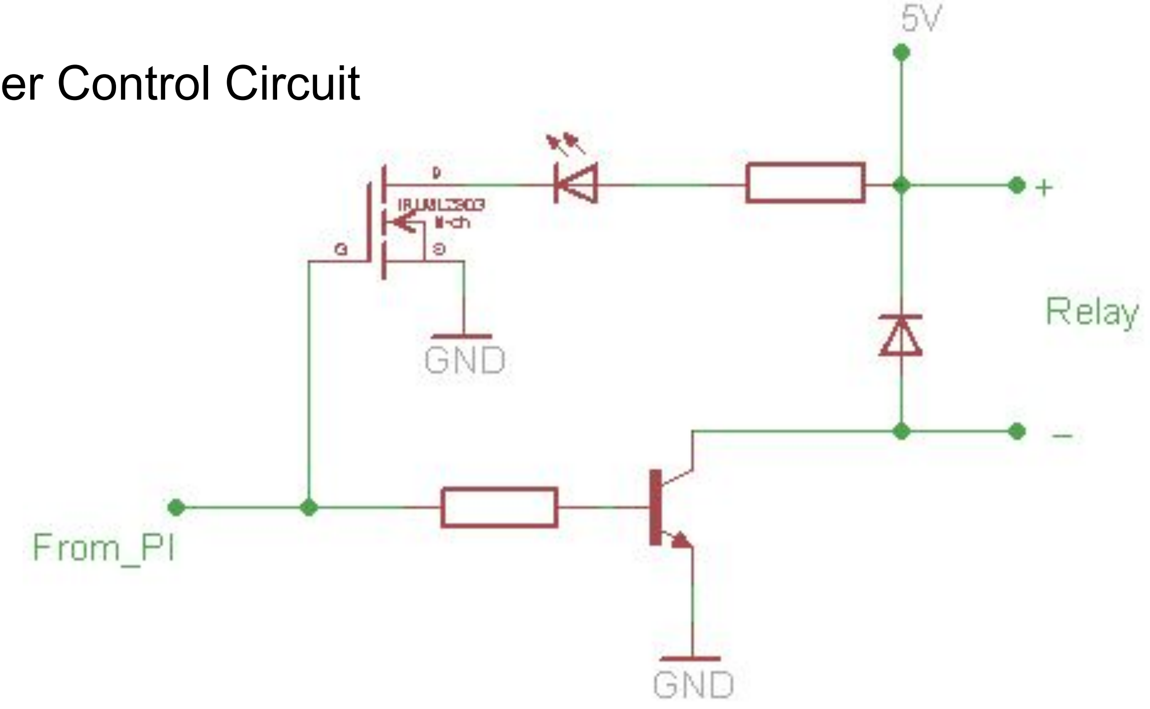


Wires going to the high voltage box

Wires going to the Gatekeeper

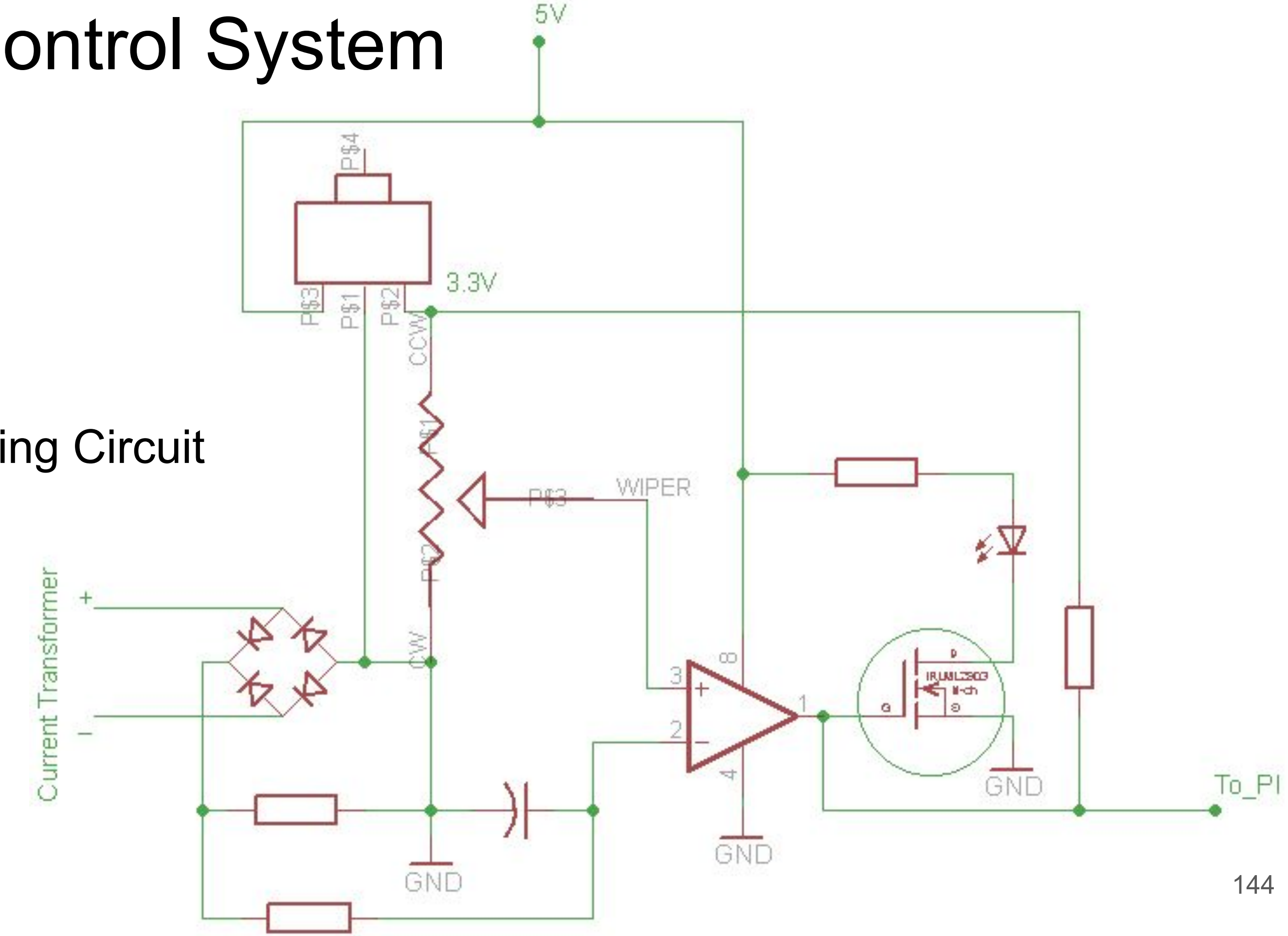
Low Voltage Control System

Machine Power Control Circuit



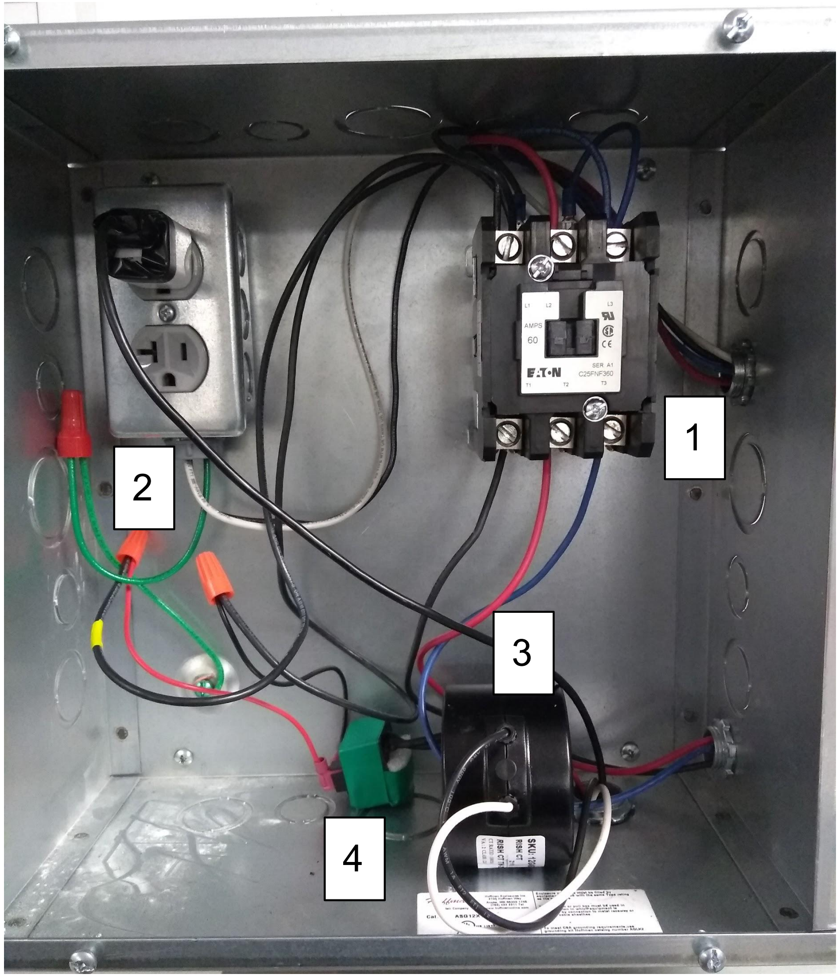
Low Voltage Control System

Machine State Monitoring Circuit



High Voltage Installation

1. Contactor
   a. 240V, 60A
2. Internal Outlet
   a. 120V
3. Current Transformer
   a. 240V, 100A
   b. Thread a main voltage wire through
4. Relay
   a. 240V, 30A
   b. Serves as the intermediate power logic gate



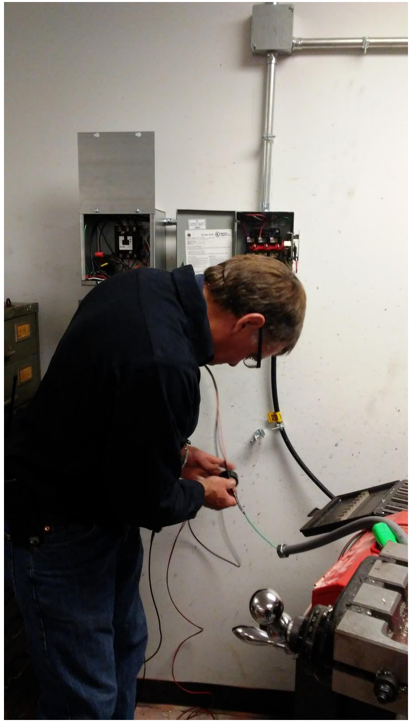
National Electric Code (NEC)

Requirement	Solution	Compliance
Certification required to handle more than 75V	Electrical technicians handle installation of high Voltage	Yes
Maintain Lockout-Tagout	Protocol unchanged by additions	Yes
Consistent insulation	Separate high voltage and low voltages boxes connected by conduit	Yes
Electrical system rated for 6x required current	Use appropriately rated components	Yes
Leads must be capped	Wire connectors	Yes

Ref [1]

Electrical System Installation

- Electricians set up high voltage box
  - Lockout/Tagout procedure followed
  - Ensured safe installation process
  - Capped high voltage leads
- Installed low voltage box
- Interconnected boxes
  - Common Grounding
    - Metal Tubing
  - Tamper Evident
- Flexible tubing
  - From electrical container
  - Leads to Raspberry Pi
  - Tamper Evident



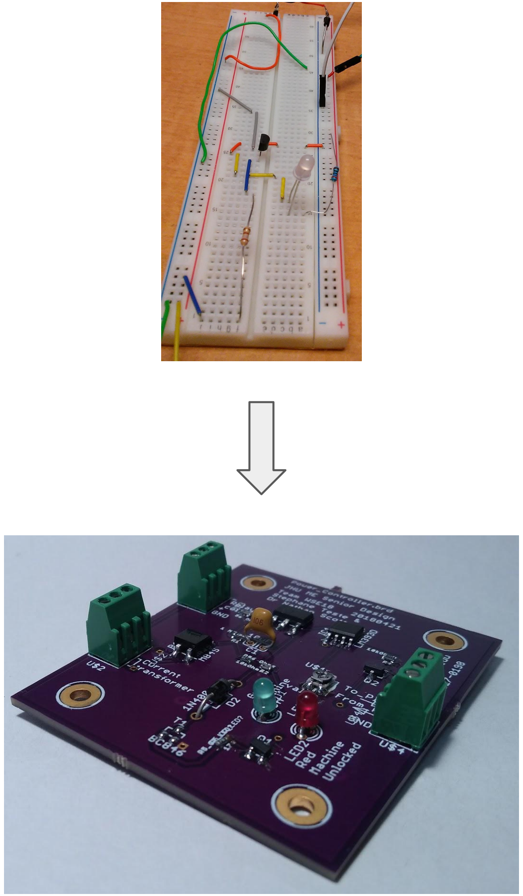
Certified Electrical
Technician

Electrical Tests

- Individual
  - Tested each component
  - Verify capabilities
- Low voltage
  - Tested low voltage control system
  - Verified
- High Voltage
  - Supervised by electrician
  - Tested and fixed issues in circuit
- Safety
  - Confirm with technician that system does not interfere with safety protocol

Improvements

- Shared voltage nodes
  - 240V, 5V
  - Ease of installation, maintenance
- Machine monitoring system installed
- Improved power delivery to Raspberry Pi
  - Eliminate the Pi's "brownouts" -a lack of power supplied
- Non-interface connections
  - Currently within electrical boxes
  - Safer/more stable connection if leads are directly connected to one another
- Permanent control boards (PCB)
  - Integrated circuit boards to replace current breadboard
  - More compact solution
- Consolidate boxes
  - Putting all components in single electrical box



Breadboard
replaced by PCB

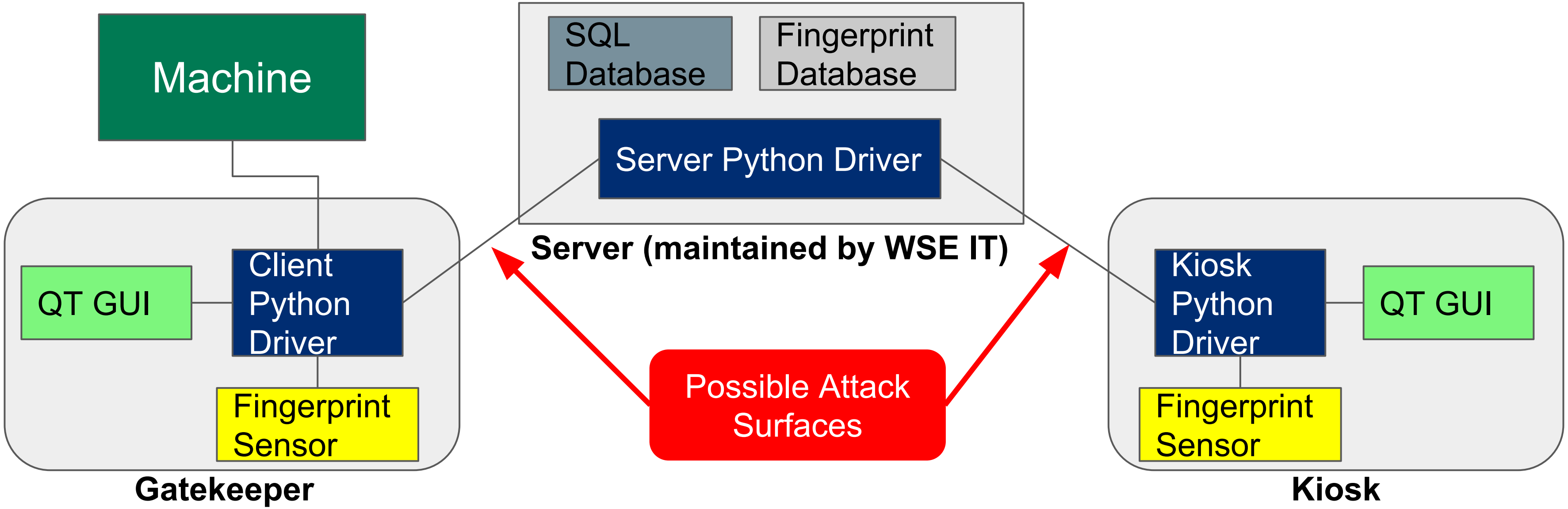
Section 7: Testing

Project Requirements		High Level Test Plan	Status
1	prevent untrained users from activating machinery	In-house testing with unauthorized prints, live user testing	Achieved
2	be difficult for a determined user to bypass	Penetration testing by Information Security Institute, live user testing	Achieved
3	be tamper-evident	Penetration testing, Live user testing	Achieved
4	aid in billing	Live user testing	Achieved
5	be minimally intrusive to the user’s ability to utilize the shop efficiently	Live user testing	Achieved
6	require minimal maintenance	Accelerated life-cycle testing of hardware	Achieved
7	be expandable to additional machines as the shop grows	Stakeholders setup a machine using only documentation	Achieved
8	not violate relevant electrical and safety codes	Expert evaluation from electricians	Achieved
9	not disconnect power to a machine while in use	In house testing and live user testing	Achieved
10	not cost more than $10,000 to install in student shop	Cost tracking	Achieved

Test Regime	Requirements Tested	See...
Penetration Testing	1, 2, 3, 4	7.1
Accelerated Life-cycle testing	6	7.2
Safety Evaluation	8	7.3
In-house Performance Evaluation	1, 2, 4, 9	7.4
Live user-testing	1, 2, 3, 4, 5, 6, 9	7.5
Stakeholder Setup Test	7	7.6

7.1: Penetration Testing

Penetration Testing



Penetration Testing (Details)

The WSE18 system requires network communication between the devices in the machine shop and a central server to function properly. These communication pathways are subject to network attacks. Penetration of these pathways places user information, including fingerprints and budget codes, at risk, as well as allowing a determined user to bypass the system and use equipment without training or paying [Req 1,2,3,4]. These pathways are secured using public/private key encryption, but our implementation of this must be checked. The JHU Information Security Institute subjected a clone of the system to extended attacks by 4 teams of volunteer “red team” hackers

Penetration Test Process

Provided to testers	Process by testers	Information Obtained
● 2 Raspberry Pis with custom interlock code ● Disk Image of the server configured with all security and custom code ● Copies of source code and documentation	● Setup a sandbox version of the WSE18 system on an isolated network ● Identify potential vulnerabilities via documentation and source code ● Attempt to exploit vulnerabilities on sandbox	● Potential security flaws ● Techniques that worked to break into system and steal/alter information ● Recommendations for patching flaws

As the testers found flaws, we patched them out and conducted additional rounds of testing on the upgraded system

Penetration Test Timeline

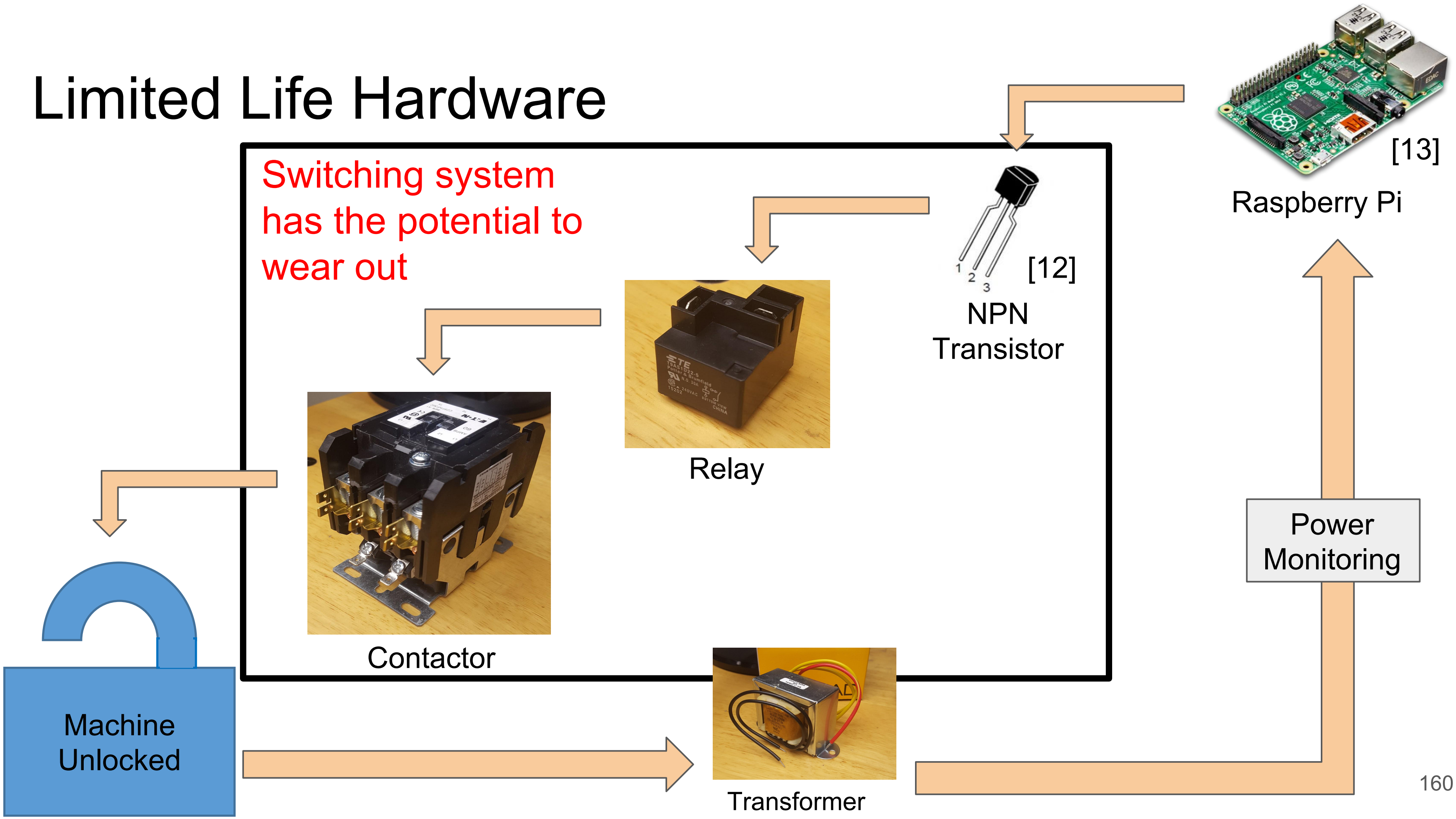
- Preparation of Pis for testing use was completed 2/20/2018
- Server clone was taken 2/21/2018
- Sandbox system set-up on 2/27/2018
- Testing occurred from 3/10/2018 onward
- Penetration testing concluded on 5/10/2018

Penetration Test Results

- 4 low-severity (non-security relevant) bugs detected in our custom code and resolved
- 1 low-severity security-relevant bug detected in Raspberry Pi OS configuration
  - Not exploitable
- 1 moderate-severity information security relevant bug detected in server OS
  - Patch will be applied once it is available
  - Merely allows reading of traffic between devices and server, not injection of faked traffic

7.2: Accelerated Life-Cycle Test

Limited Life Hardware



Switching system has the potential to wear out

Limited Life Hardware (Details)

- The circuitry used to supply and remove power from the machine includes solid-state and electromechanical switches that have limited cycle counts before failure.
- Low maintenance need is required [Req. 6], so we must ensure that the circuitry can stand up to sufficient cycles
- 10,000 cycles is estimated to be more times than the a machine will be toggled on and off over the lifetime of the system (15 years)
  - ~13 uses/week*52 weeks*15 years
- Testing is thus conducted by rapidly toggling the machine on and off for 10,000 cycles at the rate of one cycle/second

Lifecycle Results

- This test was conducted on 2/21/2018
- Contactor was cycled on then off 10,000 times at 1 full cycle/second
- System retained full functionality after test completion
- This same contactor was used without issue for the full duration of the live user tests

7.3: Safety Evaluation

National Electric Code (NEC)

Requirement	Solution	Compliance
Certification required to handle more than 75V	Electrical technicians handle installation of high Voltage	Yes
Maintain Lockout-Tagout	Protocol unchanged by additions	Yes
Consistent insulation	Separate high voltage and low voltages boxes connected by conduit	Yes
Electrical system rated for 6x required current	Use appropriately rated components	Yes
Leads must be capped	Wire connectors	Yes

Ref [1]

National Electric Code (Details)

Each requirement from the NEC must be complied with during all stages of development, testing, and release of product

- Electrical technicians handle high voltage
    - Any time touching a connected high voltage components/wires -even when off
- Maintain Lockout-Tagout. When an electrician works on the electrical system, they must be able to:
    - Lock the machine via main disconnect switches
    - Tag the machine by putting a warning label/ bag over it
- Separate high voltage and low voltages
    - 2 different boxes to categorize risk
- Electrical system rated for 6x required current:
    - All components must have a 6x safety factor for the current drawn
- No exposed wires can be present on a functioning product
    - Insulation around all end wires or connections
        - Heat Shrink
        - Box around component

7.4: In-House Performance Evaluation

In House Testing Description

When all components of the system were completed, the team informally evaluated the system as a whole to ensure that it meets the project requirements.

Specifically, the team confirmed requirements 1, 4, and 9:

1. The system cannot be bypassed without leaving behind physical evidence
4. The system is able to generate the billing spreadsheet in context
9. It is not possible to cut power to a machine except by explicitly flipping the machine switch

We also conducted a cost audit to determine the installation price of a Gatekeeper. Components cost a total of $427.49.

7.5: Live User Beta Testing

Purpose of Live User Testing

- Requirements 1-5 are qualitative and depend on the experiences and opinions of the users:
    1. prevent untrained users from activating machinery
    2. be difficult for a determined user to bypass
    3. be tamper-evident
    4. aid in billing
    5. be minimally intrusive to the user’s ability to utilize the shop efficiently
- They can only be evaluated by having people actually use the system
- Requirement 9 is best evaluated in actual use cases, as it is hard to replicate the variety of situations users can create during actual use

Process for Live User Testing

1. All shop users informed of upcoming deployment of test system on a single mill in the student shop.
2. Enrollment period during which users can add their fingerprints to the database and obtain machine and budget authorizations
3. Live testing period during which users can only use the mill via our system. An override button is in-place in case of malfunction
    a. User feedback recorded via polling
4. Updates made to system based on feedback
5. Second round testing with no override

Schedule for Live User Testing

1. User Registration started March 13th
2. System went live on new mill on March 26th; generated and emailed billing spreadsheet weekly for testing purposes
3. Collected feedback for two weeks; system deactivated April 9th for updates
4. Updates implemented by April 13th; system went live again with no override

Live User Testing Results

System Usage:

- Completed more than 5 weeks of live testing
- 50 users registered
  - 342 total fingerprints
- 109 hours of machine time billed
  - 152 usage sessions
- 5 KB15N spreadsheets generated
  - Also generated (empty) spreadsheets for all other machines, demonstrating the system will work when all gatekeepers are up and running

Live User Testing Results

Positive Feedback:

- Machine never had power removed while it was active, nor did the system allow users to apply power while the switch was on
- Majority of people were able to use with no issues
- System was able to bill people who did not fill out the log sheet
- The user interface is generally easy to understand

Live User Testing Results

Learning Opportunities:

- One fingerprint sensor failed--cause unknown
- One user was misidentified as a different user, forcing us to increase the match threshold
- Users express frustration with the 7 seconds needed to process a fingerprint
- One user has fingerprints with very few features; system has trouble recognizing him
- Lots of users had small UI suggestions which were implemented over the testing period

7.6: Stakeholder Setup Test

Stakeholder Setup Test

- Requirement 7 means that stakeholders will need to be able to set up a completely new system relying only on the provided documentation
- To demonstrate this, we had them go through the setup process while we were not present on May 3rd
- Provided:
    - Version 1 of Gatekeeper Setup Guide
    - Box containing all Gatekeeper components
        - Raspberry Pi
        - Touchscreen
        - Housing box
        - Fingerprint sensor
        - Various electrical components
    - Folder of files containing custom OS image and laser-cutter files for the housing

Setup Test Results

- Rich Mejia was able to get a Gatekeeper device fully assembled
- He provided many suggestions for changed explanations, imagery, and order of operations to make future setups easier
  - In particular, he asked for formal wiring diagrams in addition to the step-by-step directions and photographs of the completed wiring harness
  - He tried to rely on the provided photos, which were confusing
- He also discovered that the Pi cannot drive DisplayPort monitors
  - He thought that the OS image wasn't working because it wouldn't drive the monitor he was connecting it to
  - In reality, it was fine but the Pi just couldn't handle the DisplayPort monitor

Section 8: Review and Next Steps

8.1: Requirements Review

Project Requirements		High Level Test Plan	Status
1	prevent untrained users from activating machinery	In-house testing with unauthorized prints, live user testing	Achieved
2	be difficult for a determined user to bypass	Penetration testing by Information Security Institute, live user testing	Achieved
3	be tamper-evident	Penetration testing, Live user testing	Achieved
4	aid in billing	Live user testing	Achieved
5	be minimally intrusive to the user’s ability to utilize the shop efficiently	Live user testing	Achieved
6	require minimal maintenance	Accelerated life-cycle testing of hardware	Achieved
7	be expandable to additional machines as the shop grows	Stakeholders setup a machine using only documentation	Achieved
8	not violate relevant electrical and safety codes	Expert evaluation from electricians	Achieved
9	not disconnect power to a machine while in use	In house testing and live user testing	Achieved
10	not cost more than $10,000 to install in student shop	Cost tracking	Achieved

Requirement 1: prevent untrained users from activating machinery

Testing indicates achievement of this requirement. During in-house testing we were never able to get the device to authorize us when we were not listed as trained. Furthermore, during the more than 5 weeks of live user testing, only one person was ever identified as someone other than themselves. This failure occurred because we had been steadily reducing the match threshold to make it easier for the system to identify users, which had the effect of also increasing the probability of false matches. In response to this, we increased the threshold. The failure occurred again with an unregistered user after the test ended, so we raised the threshold again in response.

Requirement 2: be difficult for a determined user to bypass

We are in total compliance with this requirement. Physical access to the device is blocked by tamper-resistant screws securing the housing box AND the electrical boxes. ISI testing indicates that there are no overt weaknesses in the software setup that would allow someone to bypass the system. The entire design is also “fail safe;” if you disable the system by cutting its power or interrupting internet access, it becomes impossible to activate the controlled equipment.

Requirement 3: be tamper-evident

The idea behind this requirement is that, while a determined user may in principle be able to bypass the system, they cannot do so without being detected. In particular, the system cannot practically be bypassed without causing the Gatekeeper to restart, since you need to get into the electrical boxes and must first cut power to the Pi. When power is restored, the administrators will receive an email saying the Gatekeeper was restarted, and can then review the camera files to discover the cause.

Additionally, due to the use of security screws in the construction of each Gatekeeper, it is impossible to access the internal electronics of each box without detectably damaging the box.

Thus, **we are in full compliance with this requirement.**

Requirement 4: aid in billing

During live user testing, the system generated charges for over 100 hours of usage on the mill. The billing scheduled task on the server successfully executed every week during the test and sent correctly formatted and correctly charged spreadsheets to the desired mailing list. In fact, the system demonstrated its worth in detecting people trying to hide usage; we noticed during the last week of the test that no one had filled out the physical log sheet, yet our billing records proved that several different groups had used the mill for many hours during that week. **This requirement is achieved.**

Requirement 5: be minimally intrusive to the user’s ability to utilize the shop efficiently

The shop has functioned normally and effectively during the 5 week user test, but users are unhappy with the relatively slow fingerprint scanners used. The scanners used in the deployed system require roughly 7 seconds to process a fingerprint. In the timeframes involved in machining in the student shop, this is not a meaningful amount of time...but it *is* very annoying to have to wait. Users express frustration with the delay. Additionally, users have found that providing good scans takes some practice; new users of the system require many tries to successfully authenticate, while experienced users such as the system designers are able to reliably authenticate on the first try. This is exacerbated by the fact that we require periodic reauthentication from users (to prevent someone for logging in for someone else, then leaving). **The shop is still usable, so we are technically in compliance with this requirement. However, a faster, more accurate fingerprint sensor would be very valuable.**

Requirement 6: require minimal maintenance

The accelerated life-cycle test of the switch hardware was a resounding success, indicating that the switching circuitry has far more life than the system will require. Furthermore, the 5 weeks of live testing give confidence that the software is capable of running unsupervised for extended periods. Less encouraging is the fact that the fingerprint sensor we had installed on the Kiosk failed partway through the user testing. However, it was NOT the most heavily used sensor (the sensor on the prototype Gatekeeper had far more cycles on it), and no other sensor we purchased failed. We suspect that it was damaged by the temporary mounting to the table, which involved a c-clamp that was possibly over-tightened and crushed the sensor. The actual mounts now in place will not do this. Additionally, replacing a sensor is not a very involved task, so **we consider this requirement met.**

Requirement 7: be expandable to additional machines as the shop grows

The system was designed from the ground up to allow adding arbitrary numbers of machines to the system. The assembly test carried out by Rich Mejia on 5/3/18 indicates that the developed documentation, especially with the enhancements suggested by the tester, will permit expanding the system to new machines in the future. **This requirement is achieved.**

Requirement 8: not violate relevant electrical and safety codes

Electrical installation was handled by licensed electricians who confirmed that the system is compliant with the National Electric Code. **We are in compliance with this requirement.**

Requirement 9: not disconnect power to a machine while in use

During over 5 weeks of live user testing, the system never disconnected power from a machine that was in use. **We are in compliance with this requirement.**

Requirement 10: not cost more than $10,000 to install in student shop

The team has spent $5043.90. This includes the cost of 5 Gatekeeper installations and 1 Kiosk. Additional costs that are not yet final, including the electrical technician’s installation fee, are estimated at $500. **We are in compliance with this requirement.**

Review

The simplest means of evaluating a project is to consider how well it met the requirements laid out for it. By this metric, the project is a major success. All ten requirements were achieved, and most requirements were passed with flying colors. The requirement that the system not interfere heavily with the normal operation of the shop is negatively impacted by the poor user reception of the relatively slow fingerprint scanners used. While the requirement is met, better performance on this requirement is desirable. Furthermore, better evidence of the tamper-evidence, difficulty of bypassing, and low-maintenance nature of the system is desirable. The first and second points result mostly from the difficulty of proving a negative; just because no one (that we know of) was able to tamper, that does not mean no one ever will tamper, nor does it mean that no one tampered and got away with it. The third results from the limited time frame available to us; the only way to be sure of low-maintenance requirements over the long term is to actually be installed for an extended period.

Conclusion

The WSE self-service shop administration asked for something that would improve safety by keeping people from using equipment they are not trained on, while simultaneously making it easier to track shop usage for billing purposes. We have delivered a system that virtually guarantees the person activating a machine is trained on it. This system also completely automates the tracking of shop usage, asking administrators only to sign off on the charges when they are delivered in an email at the end of the month. While shop users would like a faster authentication mechanism, the system completely achieves its primary goals of improving safety and reducing administrative overhead.

8.2: Next Steps

Continuing Work

The system in its current state meets all requirements. It could be left as is and function for years to come in the shop. However, there are a number of “nice to have” features and improvements that, if implemented, would turn the system from something that merely works well into a truly great product. In particular, an upgraded fingerprint sensor would improve the system’s feel and make shop users much happier with it. We propose to design a custom sensor to meet this need.

A detailed proposal for extended work may be found here:

https://docs.google.com/document/d/1aEYoiR-guMrZw-_geF9awzPsY7ggRi4B4II_9mE5PE0/edit?usp=sharing

Continuing Work Highlights

The main item in proposed summer work is the development of a better fingerprint sensor for the system. The biggest bottleneck in the responsiveness of the system is the speed of data transfer of fingerprints from the current sensor to the server.

During the semester, cursory work was done evaluating the feasibility of integrating the custom fingerprint sensor found here: https://www.raspberrypi.org/blog/raspireader-fingerprint-scanner/. During this work, it was determined that the implementation was not suitably complete, and too much work would have needed to be done to integrate it into the current solution. Thus the team decided to stick with the current sensor.

Because the current sensor is the biggest cause of frustration for users of the system, work will be done Summer 2018 to integrate the above sensor, or develop a new custom sensor using a similar architecture (i.e. raspberry pi camera, prism, 3D printed housing). The intended goal is to outfit the Gatekeepers with a fingerprint sensor that responds much more quickly than the current ones.

Ref [14]

Commercialization

Stephane Teste and David Samson are considering marketing this technology to other university machine shops. Presently no concrete plans for this exist. Over the summer, both will research the feasibility of commercialization, and contact university machine shops to gauge interest in the product

Acknowledgements

Thank you to:

- Rich Middlestadt
- Rich Mejia
- Cynthia Larichiuta
- Colleen Cusimano
- Shawn Suter
- Dr. Nathan Scott and Soraya Bailey
- JHU Shops Electricians
- Joseph Carrigan
- The volunteer penetration testers (all 20+ of you!)
- Kimberly Koon
- Sebastian Yllanes
- Michael Mihalick

References

[1] “Applied Codeology: Navigating the NEC.” Delmar Cengage Learning. 10 Dec 2017.
[2] “Adafruit Display.” Chicago Electronic Distributors. 10 Dec 2017.
[3] “Contactor Stock Photo.” Crescent Electric Supply Company.
[4] Kratz, Jeremy. “C++ Logo.” The Standard C++ Foundation. Published 26 June 2017. 2 May 2018.
[5] Gould Studios. “Bridgeport Milling Machine.” Turbosquid. 8 Dec 2017.
[6] “Kookye Optical Fingerprint Reader.” Amazon. 8 Dec 2017.
[7] “Outlet Stock Photo.” Fox News. 8 Dec 2017.
[8] “QT Logo 2016.” Qt Project. 2 May 2018.
[9] “Raspberry Pi Touchscreen.” Pihut. 8 Dec 2017.
[10] "Snake Eyes Machine Screw." Tamperproof Screw Co of New York, Inc. 3 May 2018.
[11] “User’s Guide to Nist Biometric Image Software.” NIST Interagency/Internal Report (NISTIR) - 7392. Published 21 Jan 2007.
[12] “BC337 NPN Transistor” Semiconductor Components Industries, LLC. Published November 2013. 5 May 2018.
[13] “Raspberry Pi 2.0” Raspberry Pi Foundation. 5 May 2018.
[14] “RaspiReader: Build Your Own Fingerprint Reader” Raspberry Pi Foundation. 10 May 2018.

Appendix 1: Finances

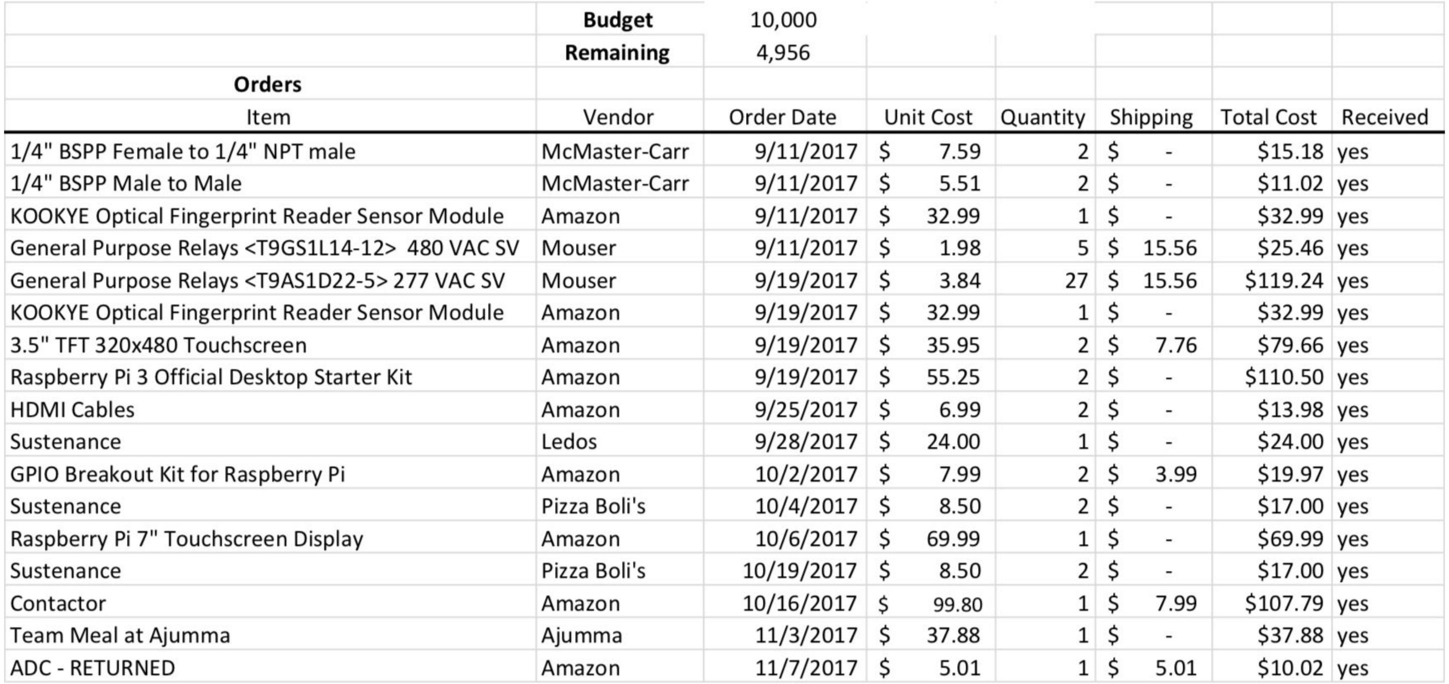
A note from Dr. Scott: Engineering design & prototyping work is inherently risky from a financial point of view. We did not know at the start of this work what technology we would pursue. As a result, some of our projects run under budget and some over budget. Be assured that your sponsorship dollars all go where they should: to student teams building stuff, even if your own team did not use every dollar.

Expenses (Next 5 Pages)

Budget: $10,000

Total Expenditures:
$5,043.90

Estimate of additional costs (i.e. electrician bill): $500.00



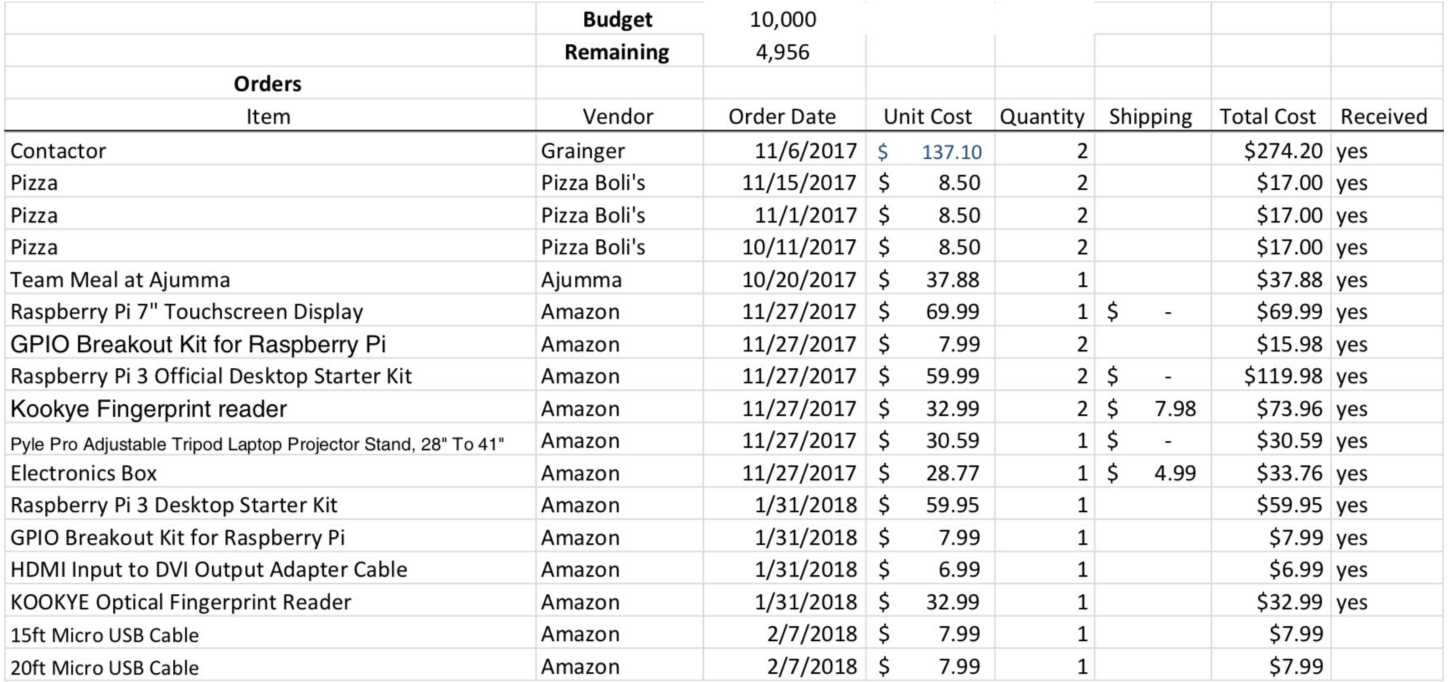
	Budget	10,000					
	Remaining	4,956					
Orders							
Item	Vendor	Order Date	Unit Cost	Quantity	Shipping	Total Cost	Received
1/4" BSPP Female to 1/4" NPT male	McMaster-Carr	9/11/2017	$ 7.59	2	$ -	$15.18	yes
1/4" BSPP Male to Male	McMaster-Carr	9/11/2017	$ 5.51	2	$ -	$11.02	yes
KOOKYE Optical Fingerprint Reader Sensor Module	Amazon	9/11/2017	$ 32.99	1	$ -	$32.99	yes
General Purpose Relays <T9GS1L14-12> 480 VAC SV	Mouser	9/11/2017	$ 1.98	5	$ 15.56	$25.46	yes
General Purpose Relays <T9AS1D22-5> 277 VAC SV	Mouser	9/19/2017	$ 3.84	27	$ 15.56	$119.24	yes
KOOKYE Optical Fingerprint Reader Sensor Module	Amazon	9/19/2017	$ 32.99	1	$ -	$32.99	yes
3.5" TFT 320x480 Touchscreen	Amazon	9/19/2017	$ 35.95	2	$ 7.76	$79.66	yes
Raspberry Pi 3 Official Desktop Starter Kit	Amazon	9/19/2017	$ 55.25	2	$ -	$110.50	yes
HDMI Cables	Amazon	9/25/2017	$ 6.99	2	$ -	$13.98	yes
Sustenance	Ledos	9/28/2017	$ 24.00	1	$ -	$24.00	yes
GPIO Breakout Kit for Raspberry Pi	Amazon	10/2/2017	$ 7.99	2	$ 3.99	$19.97	yes
Sustenance	Pizza Boli's	10/4/2017	$ 8.50	2	$ -	$17.00	yes
Raspberry Pi 7" Touchscreen Display	Amazon	10/6/2017	$ 69.99	1	$ -	$69.99	yes
Sustenance	Pizza Boli's	10/19/2017	$ 8.50	2	$ -	$17.00	yes
Contactor	Amazon	10/16/2017	$ 99.80	1	$ 7.99	$107.79	yes
Team Meal at Ajumma	Ajumma	11/3/2017	$ 37.88	1	$ -	$37.88	yes
ADC - RETURNED	Amazon	11/7/2017	$ 5.01	1	$ 5.01	$10.02	yes

Expenses (cont.)

Budget: $10,000

Total Expenditures:
$5,043.90

Estimate of additional costs (i.e. electrician bill): $500.00



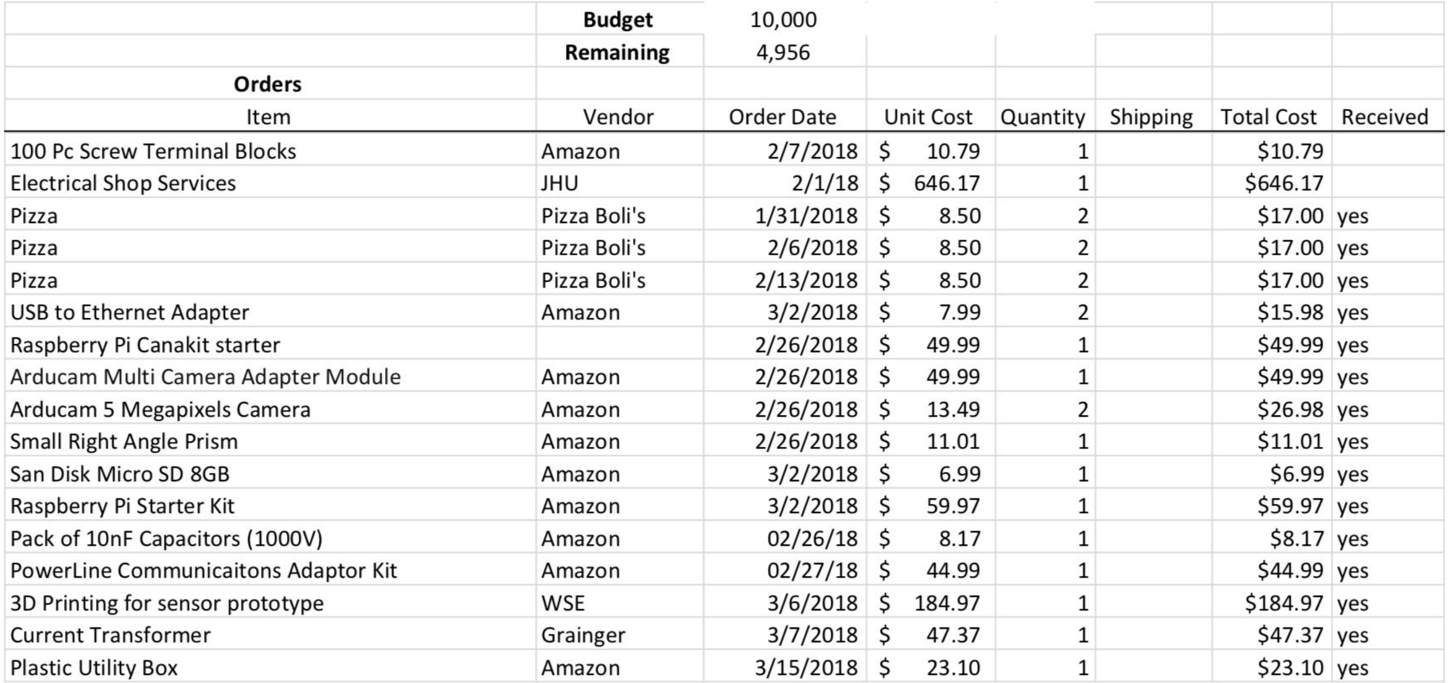
	Budget	10,000					
	Remaining	4,956					
Orders							
Item	Vendor	Order Date	Unit Cost	Quantity	Shipping	Total Cost	Received
Contactor	Grainger	11/6/2017	$ 137.10	2		$274.20	yes
Pizza	Pizza Boli's	11/15/2017	$ 8.50	2		$17.00	yes
Pizza	Pizza Boli's	11/1/2017	$ 8.50	2		$17.00	yes
Pizza	Pizza Boli's	10/11/2017	$ 8.50	2		$17.00	yes
Team Meal at Ajumma	Ajumma	10/20/2017	$ 37.88	1		$37.88	yes
Raspberry Pi 7" Touchscreen Display	Amazon	11/27/2017	$ 69.99	1	$ -	$69.99	yes
GPIO Breakout Kit for Raspberry Pi	Amazon	11/27/2017	$ 7.99	2		$15.98	yes
Raspberry Pi 3 Official Desktop Starter Kit	Amazon	11/27/2017	$ 59.99	2	$ -	$119.98	yes
Kookye Fingerprint reader	Amazon	11/27/2017	$ 32.99	2	$ 7.98	$73.96	yes
Pyle Pro Adjustable Tripod Laptop Projector Stand, 28" To 41"	Amazon	11/27/2017	$ 30.59	1	$ -	$30.59	yes
Electronics Box	Amazon	11/27/2017	$ 28.77	1	$ 4.99	$33.76	yes
Raspberry Pi 3 Desktop Starter Kit	Amazon	1/31/2018	$ 59.95	1		$59.95	yes
GPIO Breakout Kit for Raspberry Pi	Amazon	1/31/2018	$ 7.99	1		$7.99	yes
HDMI Input to DVI Output Adapter Cable	Amazon	1/31/2018	$ 6.99	1		$6.99	yes
KOOKYE Optical Fingerprint Reader	Amazon	1/31/2018	$ 32.99	1		$32.99	yes
15ft Micro USB Cable	Amazon	2/7/2018	$ 7.99	1		$7.99	
20ft Micro USB Cable	Amazon	2/7/2018	$ 7.99	1		$7.99	

Expenses (cont.)

Budget: $10,000

Total Expenditures:
$5,043.90

Estimate of additional costs (i.e. electrician bill): $500.00



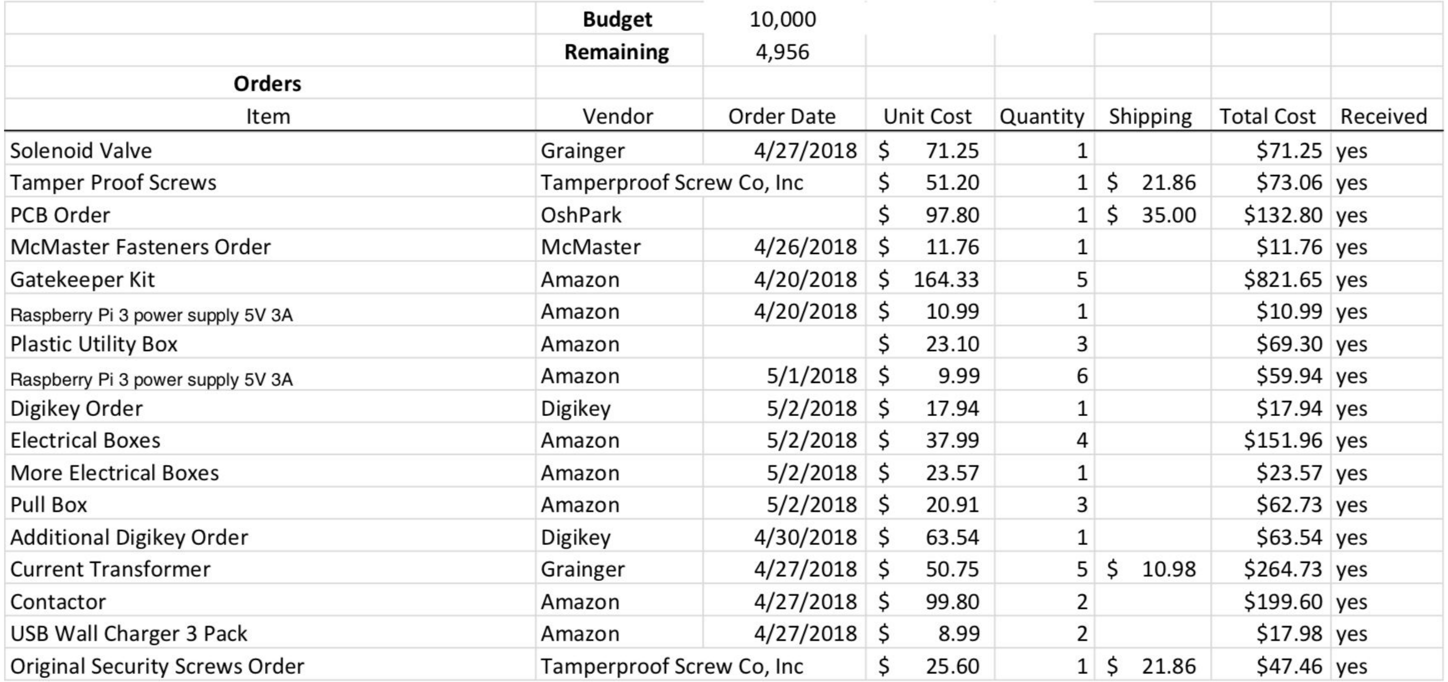
	Budget	10,000					
	Remaining	4,956					
Orders							
Item	Vendor	Order Date	Unit Cost	Quantity	Shipping	Total Cost	Received
100 Pc Screw Terminal Blocks	Amazon	2/7/2018	$ 10.79	1		$10.79	
Electrical Shop Services	JHU	2/1/18	$ 646.17	1		$646.17	
Pizza	Pizza Boli's	1/31/2018	$ 8.50	2		$17.00	yes
Pizza	Pizza Boli's	2/6/2018	$ 8.50	2		$17.00	yes
Pizza	Pizza Boli's	2/13/2018	$ 8.50	2		$17.00	yes
USB to Ethernet Adapter	Amazon	3/2/2018	$ 7.99	2		$15.98	yes
Raspberry Pi Canakit starter		2/26/2018	$ 49.99	1		$49.99	yes
Arducam Multi Camera Adapter Module	Amazon	2/26/2018	$ 49.99	1		$49.99	yes
Arducam 5 Megapixels Camera	Amazon	2/26/2018	$ 13.49	2		$26.98	yes
Small Right Angle Prism	Amazon	2/26/2018	$ 11.01	1		$11.01	yes
San Disk Micro SD 8GB	Amazon	3/2/2018	$ 6.99	1		$6.99	yes
Raspberry Pi Starter Kit	Amazon	3/2/2018	$ 59.97	1		$59.97	yes
Pack of 10nF Capacitors (1000V)	Amazon	02/26/18	$ 8.17	1		$8.17	yes
PowerLine Communicaitons Adaptor Kit	Amazon	02/27/18	$ 44.99	1		$44.99	yes
3D Printing for sensor prototype	WSE	3/6/2018	$ 184.97	1		$184.97	yes
Current Transformer	Grainger	3/7/2018	$ 47.37	1		$47.37	yes
Plastic Utility Box	Amazon	3/15/2018	$ 23.10	1		$23.10	yes

Expenses (cont.)

Budget: $10,000

Total Expenditures:
$5,043.90

Estimate of additional costs (i.e. electrician bill): $500.00



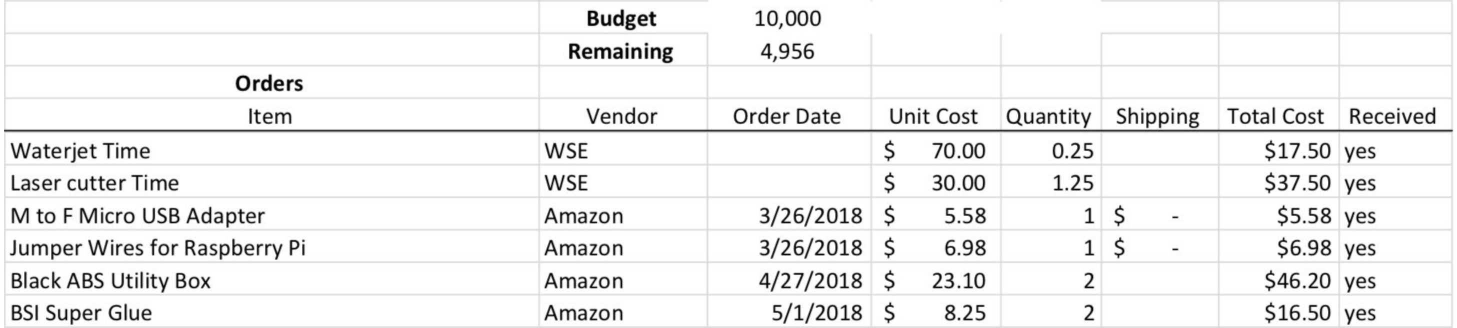
	Budget	10,000					
	Remaining	4,956					
Orders							
Item	Vendor	Order Date	Unit Cost	Quantity	Shipping	Total Cost	Received
Solenoid Valve	Grainger	4/27/2018	$ 71.25	1		$71.25	yes
Tamper Proof Screws	Tamperproof Screw Co, Inc		$ 51.20	1	$ 21.86	$73.06	yes
PCB Order	OshPark		$ 97.80	1	$ 35.00	$132.80	yes
McMaster Fasteners Order	McMaster	4/26/2018	$ 11.76	1		$11.76	yes
Gatekeeper Kit	Amazon	4/20/2018	$ 164.33	5		$821.65	yes
Raspberry Pi 3 power supply 5V 3A	Amazon	4/20/2018	$ 10.99	1		$10.99	yes
Plastic Utility Box	Amazon		$ 23.10	3		$69.30	yes
Raspberry Pi 3 power supply 5V 3A	Amazon	5/1/2018	$ 9.99	6		$59.94	yes
Digikey Order	Digikey	5/2/2018	$ 17.94	1		$17.94	yes
Electrical Boxes	Amazon	5/2/2018	$ 37.99	4		$151.96	yes
More Electrical Boxes	Amazon	5/2/2018	$ 23.57	1		$23.57	yes
Pull Box	Amazon	5/2/2018	$ 20.91	3		$62.73	yes
Additional Digikey Order	Digikey	4/30/2018	$ 63.54	1		$63.54	yes
Current Transformer	Grainger	4/27/2018	$ 50.75	5	$ 10.98	$264.73	yes
Contactor	Amazon	4/27/2018	$ 99.80	2		$199.60	yes
USB Wall Charger 3 Pack	Amazon	4/27/2018	$ 8.99	2		$17.98	yes
Original Security Screws Order	Tamperproof Screw Co, Inc		$ 25.60	1	$ 21.86	$47.46	yes

Expenses (cont.)

Budget: $10,000

Total Expenditures:
$5,043.90

Estimate of additional costs (i.e. electrician bill): $500.00



	Budget	10,000					
	Remaining	4,956					
Orders							
Item	Vendor	Order Date	Unit Cost	Quantity	Shipping	Total Cost	Received
Waterjet Time	WSE		$ 70.00	0.25		$17.50	yes
Laser cutter Time	WSE		$ 30.00	1.25		$37.50	yes
M to F Micro USB Adapter	Amazon	3/26/2018	$ 5.58	1	$ -	$5.58	yes
Jumper Wires for Raspberry Pi	Amazon	3/26/2018	$ 6.98	1	$ -	$6.98	yes
Black ABS Utility Box	Amazon	4/27/2018	$ 23.10	2		$46.20	yes
BSI Super Glue	Amazon	5/1/2018	$ 8.25	2		$16.50	yes

Full expenses spreadsheet can be found at: https://docs.google.com/spreadsheets/d/12Nj-vAkb6Q3DfC7pEC3eFCeiTTDK5vQEFtAUXouBF-I/edit?usp=sharing

Projected Deployment Cost

Each Gatekeeper uses components costing $427.49+shipping. Bulk ordering can potentially reduce these costs further.

The work conducted by the electricians to install a system is a variable cost, depending on the state of the electrical system surrounding the machine to be controlled. However, labor is unlikely to cost more than $400 for 4 installations.

A detailed cost breakdown, with links for ordering more components, is available here: https://docs.google.com/spreadsheets/d/1RR1LcHQNt9hPBADR2CIPDq5knEZoumeF0wtRRccP1DI/edit?usp=sharing

Budget: $10,000

Total Expenditure: $5,043.90

Excess Funds: $4,956.10

40%

Of total budget as projected reserve funds/cost savings

Appendix 2: Bill of Materials

Spreadsheet with components is at:

https://docs.google.com/spreadsheets/d/1RR1LcHQNt9hPBADR2CIPDq5knEZoumeF0wtRRccP1DI/edit#gid=0

Appendix 3: Documentation

Documentation Files

File	Link
Gatekeeper Setup Guide	https://docs.google.com/document/d/14GKjbu0SB9ID-966V0SumIsd_CKAYpvC1rpLqSFpTLM/edit
System Documentation	https://docs.google.com/document/d/1PsotDg2mYMbk5d5mN8Vk359-g8yNYYQzfsVWbSh8Tn0/edit
Troubleshooting Guide	https://docs.google.com/document/d/1AHeuT1GP8BzznzZZpDUDJuF8OVpj5NNTh8zTMZNFg4o/edit
Gatekeeper Parts List	https://docs.google.com/spreadsheets/d/1RR1LcHQNt9hPBADR2CIPDq5knEZoumeF0wtRRccP1DI/edit#gid=0

Appendix 4: Design Files

Design Files

Folder	Contents
CAD Files	SolidWorks files and DXFs for manufacturing
Code Snapshotted 3_10_18	A copy of the bitbucket repository updated 3/10/18
Documentation	PDFs of Troubleshooting Guide, Assembly Guide, and System Documentation
Meetings	Minutes and presentations from fortnightly meetings
OS Images	A compressed image file to use when flashing new Raspberry Pi SD cards
PCB	Files for making more PCBs
Photos	Collection of photos and graphics produced throughout the project
Reports	PDFs of the Preliminary Design Review, the Critical Design Review, Fall Design Day presentation, the Plan for Testing and Verification, the Spring Design Day Presentation, and the Final Report

Appendix 5: Manufacturing Files

Manufacturing Files

File	Where to obtain
Raspberry Pi Image file	Machine Shop Share Drive/WSE18 Files/OS Images/Kiosk_Image_04_13_18.img.zip
Screen Bracket DXF	Machine Shop Share Drive/WSE18 Files/CAD Files/Screen Bracket.dxf
Sensor holder drawing	Machine Shop Share Drive/WSE18 Files/CAD Files/fingerprint_holder.SLDDRW
Faceplate Cutout drawing	Machine Shop Share Drive/WSE18 Files/CAD Files/Bracket on cutout.zip
Gatekeeper Parts List	https://docs.google.com/spreadsheets/d/1RR1LcHQNt9hPBADR2CIPDq5knEZoumeF0wtRRccP1DI/edit#gid=0
Contactor Mount Plate drawing	https://drive.google.com/open?id=1p4JA6bInl0FTtHu_BeChEbu7b3cCCTMO

Appendix 6: Extra Components

Item	Location	Quantity
Unimaged SD cards	Pi Stuff	4
Spare Imaged SD cards	Pi Stuff	1
Spare SD card adapters	Pi Stuff	5
Raspberry Pi + imaged SD card	Pi Stuff	2
Raspberry Pi w/out imaged SD card	Pi Stuff	2
Spare fingerprint sensors	Pi Stuff	2
M3 Washers	Fasteners	lots
4-40 Tamperproof Screws+spanners	Fasteners	lots
10-32 Tamperproof Screws+spanners	Fasteners	lots
4-40 nuts	Fasteners	lots
M3 screws	Fasteners	lots
4-40 phillips head screws	Fasteners	lots

Item	Location	Quantity
30-row breadboard	Pi Stuff	2
64-row breadboard	Pi Stuff	1
Pi GPIO-to-extender ribbon cables	Pi Stuff	3
Pi GPIO Extension Board	Pi Stuff	3
5.1V 3A usb power supply, no cable	Power Supplies	5
Official Raspberry Pi power supply	Power Supplies	2
5V 1A usb power supply, no cable	Power Supplies	6
20 ft USB to microUSB	Power Supplies	1
15 ft USB to microUSB	Power Supplies	1
120 Ohm resistor	Pi Stuff	10
Spare sensor holders	Pi Stuff	9
Jumper wires	Pi Stuff	1 bag

Item	Location	Quantity
Spare screen brackets	Fasteners	4
USB2.0 to ethernet adapters	Miscellaneous	2
1.5meter usb to micro usb power, with switch	Power Supplies	6
Multiple styles of screw terminals and other connectors	Miscellaneous	lots
Switchable relay, 5VDC in, controls 240V 30Amps	Miscellaneous	23
Adafruit touchscreen display	Miscellaneous	1
Raspberry Pi touchscreen	Miscellaneous	2
Low viscosity cyanoacrylate glue	Fasteners	1
Raspberry Pi Housing	Pi Peripherals	4
RES SMD 174 OHM 1% 1/10W 0603	Electronics Box	11
MOSFET N-CH 30V 1.2A SOT-23	Electronics Box	5
RES SMD 56 OHM 0.5% 1/16W 0603	Electronics Box	4

Item	Location	Quantity
RES SMD 10K OHM 0.5% 1/10W 0603	Electronics Box	3
TRANS NPN 65V 0.1A SOT-23	Electronics Box	2
RES SMD 1M OHM 0.1% 1/16W 0603	Electronics Box	6
DIODE GEN PURP 400V 1A DO41	Electronics Box	2
CAP CER 10UF 25V X7R RADIAL	Electronics Box	6
IC DUAL DIFF COMP 8-SOIC	Electronics Box	5
LED RED DIFF 3MM ROUND T/H	Electronics Box	5
LED GRN DIFF 3MM ROUND T/H	Electronics Box	5
IC RECT BRIDGE 0.5A 400V 4SOIC	Electronics Box	5
TERM BLOCK 3POS SIDE ENT 2.54MM	Electronics Box	2
TRIMMER 10K OHM 0.1W SMD	Electronics Box	2
IC REG LINEAR 3.3V 1A SOT223-4	Electronics Box	1

Item	Location	Quantity
CAP CER 47UF 25V X7R RADIAL	Electronics Box	10

Reflection

Geordan Gutow

Learning Python was valuable, as was improving familiarity with relational databases. I also learned a lot about the basics of the git version control tool and about how Unix-style operating systems are configured and controlled. I realized late how valuable it was to have a standardized, consistent interface between the various executable codes in our system; if I had realized sooner how useful that would be, a lot of minor bugs would have been avoided.

David Samson

I learned a lot about developing interfaces that are meant to be used by people, and how people will always find ways to break your system. I also gained a lot of practical experience in the development of software systems. If I could redo the project, I think the biggest change I would have made would have been to make the Gatekeeper be run by a single monolithic program rather than separate programs that have to communicate with each other. For this project that would probably have meant developing everything in Python since the team lacked the experience to implement it in C++, but honestly, if we could have managed it, making the whole thing a single C++ application would likely have yielded the best results in terms of responsiveness of the interface. Also, this project could have used another team member as the three of us barely had enough time to complete the main requirements; many nice to haves had to be cut due to time.

Stephane Teste

This project has given me a depth into electrical engineering that I hadn’t thought I would attain as a mechanical engineer. I learned of in-rush currents, the importance of stabilizing electrical flows, and how to apply pull-up resistors for logic. This culminated into a single printed circuit board for which all the logic aspects are incorporated.

Furthermore, the development of a final -and possibly commercial- product has been a valuable experience that I feel will serve and guide me in the future. The importance of accommodating client requests and addressing design/manufacturing concerns stand out most prominently.

Client Grade

Report read by ________________________

Suggested grade scale

- As good as could be expected from a team of young engineers; entirely satisfactory 90-100%
- Pretty good but project not quite finished to commercial standards 80-90%
- Project itself was pretty well done but the report is not so readable 70-80%
- Team seem to have missed the mark a bit, or output seems lower than it should be 60-70%
- Disappointing work or very poorly communicated, or both 40-60%
- Abject shame 0-40%

Grade % ______

Please feel free to comment for the team.