Tool Rental System

Final Report

Sponsor: WSU Frank Innovation Zone (FIZ)
305 NE Spokane St, Pullman, WA, United States

Mentor: Mae Siev

Team: Judah Davis
Kale Zweig
William VanDyke
Conor Trull

Course: EE416 Electrical Engineering Design
Instructor: Dr. José G. Delgado-Frias
School of Electrical Engineering and Computer Science
Pullman, WA 99164

December 17th, 2022

Table of Contents

Note: The construction manual goes into detail on how the scanner was constructed but is formatted so that anyone at the FIZ can use it to create their own scanner. It is included as supplementary material to parts two and four of the report.

Executive Summary

The members of Team Olympic partnered with the Frank Innovation Zone (FIZ) to create a tool rental system (TRS) for its members. The FIZ provides a centrally located multidisciplinary collaborative space with tools for its members. Before Team Olympics' TRS, the FIZ used excel spread sheets to rent out tools. This system had many flaws, and many tools were going missing. Team Olympic created a tool rental program that could create new tools in a database and rent out those tools to its users. Users provide their email and a picture of their school ID in the program so that if a tool hasn't been returned in a timely manner FIZ members can contact those renters for clarification. RFID tags and barcodes attached to the tools allow the tools to be scanned for renting out and turning in. To do this Team Olympic created a 3D printed scanner that can read and write to RFID's as well as read barcodes. A tool ID is written on a Mifare classic RFID tag through the scanner, barcodes can be generated by the TRS program. The TRS' ease of use makes it a wholly beneficial edition to the FIZ with plans for its expansion in the works for future projects

1. Introduction

Here at Washington State University there are many places for "making", and The Frank Innovation Zone (FIZ) is one of the best creator spaces for students. The FIZ is an easily accessible workshop where students can translate their ideas into reality using a vast amount of space and tools provided by the university. Located in the basement of Dana Hall, the FIZ is centrally located and provides over 3,000 square feet of collaborative space filled with conventional and computer-controlled equipment [1].

With all that space and hundreds of tools one can see how difficult it would be to manage and keep track of tools, who is using them, the condition of the tool, and to check inventory. Currently, the FIZ uses spreadsheets and cell phone pictures to document the flow of these tools. This creates a burden on the users of the system because it involves manually inputting and checking to see if a tool has been checked out for too long. There is also no way to tell the current inventory which makes it harder to tell when tools have gone missing.

The WSU FIZ has tasked us with creating a tool rental system. The system will enable tool tracking to help limit the number of tools that go missing and enable data collection on resources to make purchasing decisions in the future. The system will take advantage of modern technology to make a more streamlined rental process in the service center of the FIZ.

Stakeholders of this project include our mentor Mae, the students who use the FIZ, and the Voiland College of Engineering & Architecture. These are all stakeholders as they will all be affected by the failure or success of this project.

The four main technologies that must be integrated for this project are Barcode, RFID, Arduino, and Python. For guidance on the programming, shielding and other specifics of the Arduino microcontroller we reference “Arduino, A Technical Reference.” The handbook details many different microcontrollers, the arduino programming language, and building around the Arduino which will be invaluable information for the construction of the tool rental system scanner [2]. The RFID tags will be Mifare Classics with 1kb of EEPROM memory and we will be referencing the Adafruit Industries manual for technical specifics [3].

Our main programming language will be python, Python is a very simple language, and has a very straightforward syntax [4]. We will be creating the GUI with Tkinter which is a python

interface to Tk, a popular GUI tool kit [5]. With these tools we will make a tool rental system for FIZ that is intuitive and easy to use.

2. Description of Culminating Design

The main goal of this project is to design, develop, test, and deliver a tool rental system that keeps the FIZ service center organized and operating as smoothly as possible. The system will operate on the FIZ central computer and will consist of 2 main components:

Rental Application

- A database system and a user interface has been designed and constructed and will manage the tool data and handle the rental process. The interface will have the ability to read and write RFID tags and read and print Barcodes concurrently on the same database, quickly and easily add or remove tools/rentals from the database, allow for employee input of the condition of the tool being rented (by taking picture of the tool), allow for employees to add pictures of student ID's to a rental profile, and track how many of each tool remains in inventory and how many are currently on loan. The interface has functionality that will show when a rental is overdue and will show which items are currently checked out by the user. It also includes the ability to search and sort the rental and tool tables based on different criteria and handles selection of different COM ports for selecting the scanner and external camera if needed. When searching rentals, a user can search for a tool that is currently rented out and the system will display the rentals who currently have that tool on loan.

RFID/Barcode scanner

- A RFID/Barcode scanner is used to log and identify each individual tool in the database. The scanner was designed and constructed from scratch using a custom programmed Arduino microcontroller. The microcontroller has been integrated into a 3D printed handheld device with a GM67 barcode scanner, and an RC522 RFID module that will allow for reading and writing RFID tags. The scanner also houses other components such as an RGB LED and a buzzer that will provide feedback to the user, indicating things like if the scanner is currently in a read state, or when the scanner successfully scans or writes an item. The board will communicate with the database system via serial communications through a USB. An early prototype mockup visualization can be seen in the figure below.

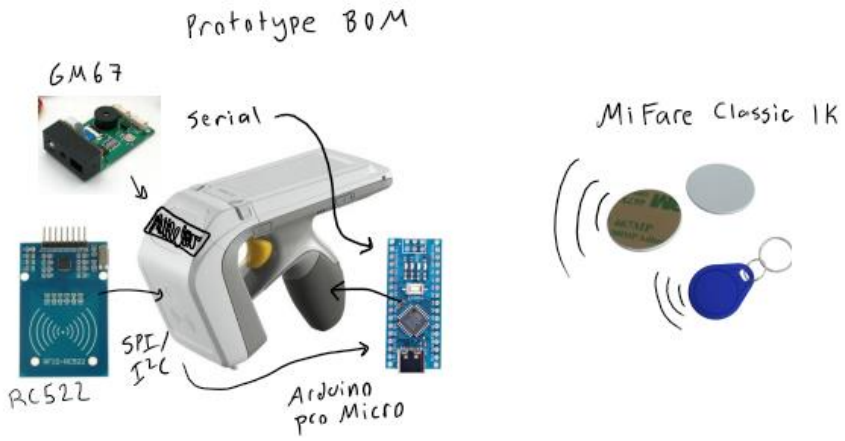


Figure 1. Scanner Mock-Up

The Tool Rental System will operate on the FIZ central computer and will consist of two main components: Rental Application - A database system and a user interface that has been designed and constructed that will manage the tool data and handle the rental process and the Scanner – a hardware system designed to read barcodes and RFID tags and be able to write RFID tags. The computer program will have the ability to read RFID and Barcodes within the same system to be

able to interact with the computer program smoothly. The scanner reads can trigger actions in the database to quickly and easily add or remove tools from each user's checkout. It will also allow employees to take images of the checked-out tools, as well as enter the users email and student ID. In addition, it will track how many of each tool remains in inventory to give status reports of what is in stock and who has checked out which tools.

The RFID/Barcode scanner will be used to log and identify each individual tool in the database. The scanner will be designed and constructed using an Arduino microcontroller, a barcode scanner, and an RFID read / writer. The microcontroller and modules are to be integrated into a 3D printed handheld device with a GM67 barcode scanner, and an RC522 RFID module that will allow for reading and writing RFID tags. The board will communicate with the database system via serial communications through a USB. The handheld device will be easy to use and durable. Even if the equipment is broken instructions will be provided to the sponsor for re-manufacturing. The main goal of this project is to design, develop, test, and deliver a tool rental system that keeps the FIZ service tool inventory organized and operating as smoothly as possible. As forementioned the current problem is that the FIZ has no formal way to keep track of their tools stock, how many of each tool there is, the condition of each tool, or who has possession of what items. This created disarray and operational inefficiency in the FIZ. It also results in financial loss and lack of accountability for when tools are damaged or go missing. To overcome these logistic issues the following solutions were set forth by the client and were implemented for the prototype. The clients specified that they wanted a singular ID for each item so that any tools could be engraved with an ID, have a barcode sticker placed on it or have an RFID sticker on it. An early mock-up of the design of the entire tool rental system can be found below in figure 2.

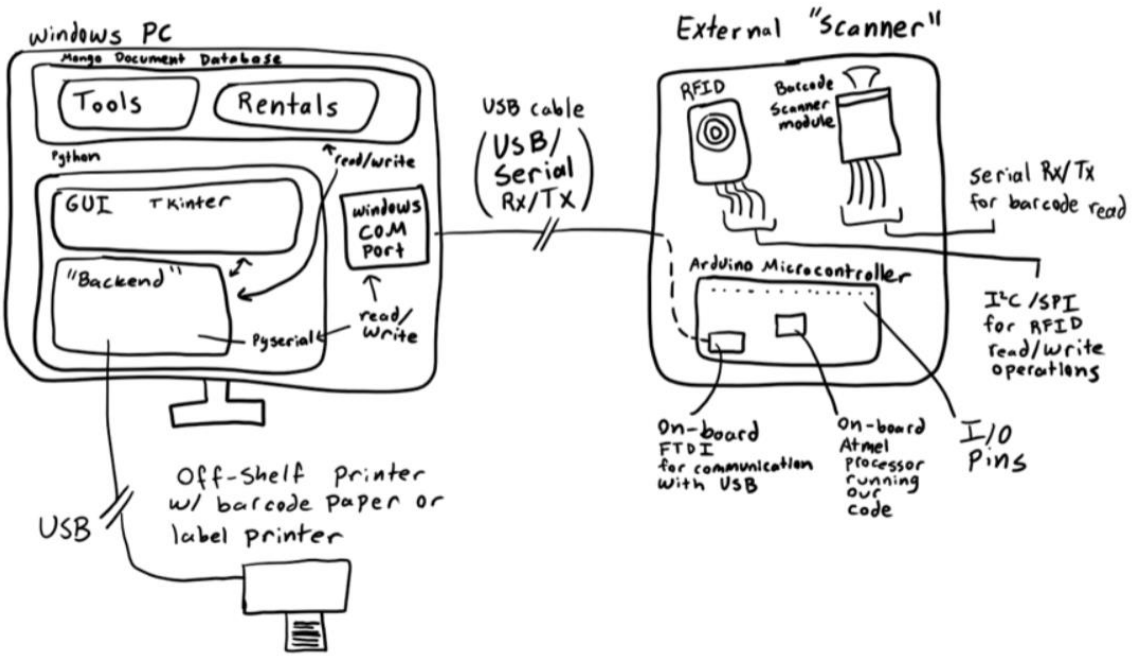


Figure 2. TRS Design

3. Project Management

The members of team Olympic were assigned with these general responsibilities:

- Kale has designed the 3D printed scanner case and aided with developing the software application as well as hardware.
- Conor has developed the software application, implementing a program to execute most functionality of the system and interact with the database. A clean user interface has been constructed for maximum user friendliness and efficiency.
- Judah has collected user feedback and bug reports as well as tested system reliability.

- William has assembled and program micro controller to interact with the software application.

The tool rental system has many small tasks that have been generalized in this tasks list. The goal was to have a complete prototype two weeks before presentations. But it can be completed much sooner with concentrated effort, especially as we have completed the bulk of the system.

	Name	Duration	Start	Finish	Predecessors	Resource Names
1	Closed Beta Prototype	0 days	11/8/22 8:00 AM	11/8/22 8:00 AM		
2	Critical Design Review	0 days	10/11/22 8:00 AM	10/11/22 8:00 AM		
3	Peer Evalutations	0 days	10/18/22 8:00 AM	10/18/22 8:00 AM		
4	Prototype Demo	0 days	12/8/22 5:00 PM	12/8/22 5:00 PM	5;7;13;18	
5	Project Title and Abstract	7 days	11/14/22 8:00 AM	11/22/22 5:00 PM		Conor;Judah;Kale;William
6	User Manual	7 days?	11/23/22 8:00 AM	12/1/22 5:00 PM	5	Judah;William
7	**Testing and Data Collection Complete**	42 days?	**9/12/22 8:00 AM**	**11/8/22 5:00 PM**		**Judah**
8	Test Reliability and Reporting Bugs	7 days?	9/12/22 8:00 AM	9/20/22 5:00 PM		Judah
9	Beta Prototype Live Testing	14 days?	9/21/22 8:00 AM	10/10/22 5:00 PM	8	Judah
10	Collect User Complaints and Bugs	2 days?	10/11/22 8:00 AM	10/12/22 5:00 PM	9	Judah
11	Address Bugs and User Complaints	12 days?	10/13/22 8:00 AM	10/28/22 5:00 PM	10	Conor;Judah;Kale;William
12	Repeat Reliability Testing	7 days?	10/31/22 8:00 AM	11/8/22 5:00 PM	11	Judah
13	**Scanner Prototype Case Design**	35 days?	**9/12/22 8:00 AM**	**10/28/22 5:00 ...**		**Kale**
14	Create Case Design	14 days	9/12/22 8:00 AM	9/29/22 5:00 PM		Kale
15	Test Usability and Durability	7 days?	9/30/22 8:00 AM	10/10/22 5:00 PM	14	Judah;Kale
16	Make Modifications	7 days?	10/11/22 8:00 AM	10/19/22 5:00 PM	15	Kale
17	Finilize Design for ALL time	7 days?	10/20/22 8:00 AM	10/28/22 5:00 PM	16	Kale
18	**GUI and Back End**	49 days?	**9/12/22 8:00 AM**	**11/17/22 5:00 ...**		**Conor**
19	Address Immediate Issues	7 days?	9/12/22 8:00 AM	9/20/22 5:00 PM		Conor;Kale
20	Improve Usability	7 days?	9/21/22 8:00 AM	9/29/22 5:00 PM	19	Conor;Kale
21	Make *PRETTY*	14 days?	9/30/22 8:00 AM	10/19/22 5:00 PM	20	Conor;Kale
22	Address Customer Complaints	14 days?	10/31/22 8:00 AM	11/17/22 5:00 PM	11;21	Conor;Kale
23	**Scanner Hardware**	42 days?	**9/12/22 8:00 AM**	**11/8/22 5:00 PM**		**William**
24	Adress Immediate Issues	7 days?	9/12/22 8:00 AM	9/20/22 5:00 PM		William
25	Improve Data Readibility	7 days?	9/21/22 8:00 AM	9/29/22 5:00 PM	24	William
26	Implement Visual and Audio Ques	14 days?	9/30/22 8:00 AM	10/19/22 5:00 PM	25	William
27	Finialize Design with Team	14 days?	10/20/22 8:00 AM	11/8/22 5:00 PM	26	Conor;Judah;Kale;William

Figure 3. Task List

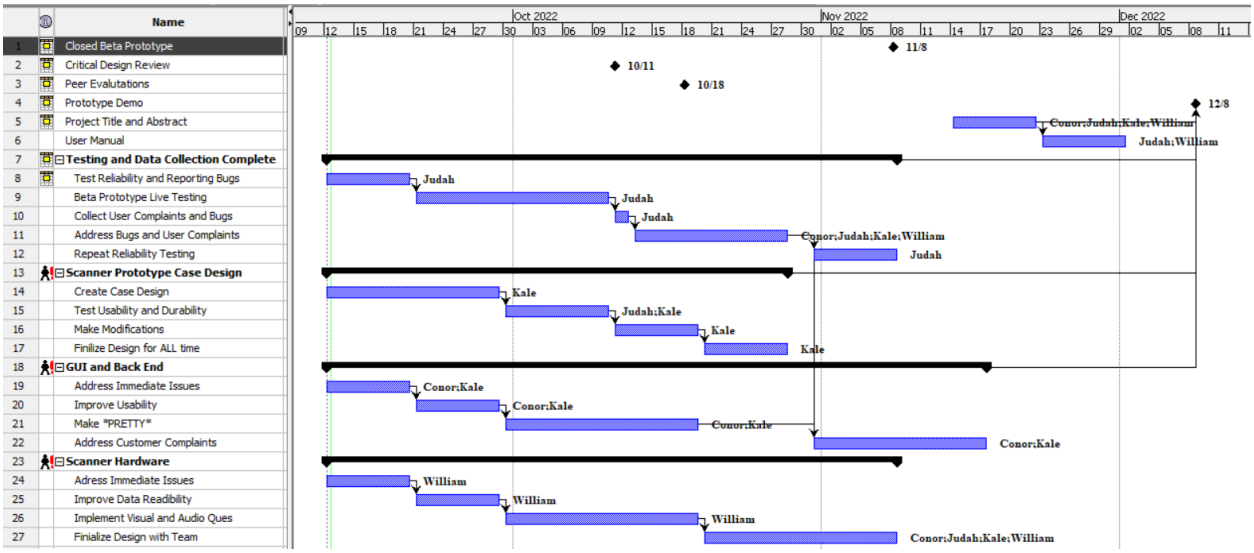


Figure 4. Gant Chart

This project required skills in both the areas of electrical engineering and software engineering. It was necessary that each member of the team had a clear understanding of each component, how it works, and how it would be constructed. In addition to technical skills, it was necessary that each member could communicate clearly and could work fluently in a collaborative environment. The specific skills that needed to be acquired are specific to each part of the project.

The rental application would involve almost all the software development, it will require knowledge of database design and operation as well as knowledge of designing and constructing an interface for users to interact with the database. The RFID/Barcode scanner would require knowledge of Arduino microcontrollers, how to program the microcontroller, and how to

integrate the microcontroller into circuits using the scanners. Knowledge of how to create computer models for 3D printing would be a requirement for construction of the scanner body.

Here are the primary goals of the tool rental system as outlined by our client/mentor Mae.

- Add ability to read RFID and Barcodes concurrently on the same database
- Quickly and easily add or remove tools from database
- Record both tool rental and return with timestamps
- Allow for employee input of the condition of the tool being rented
- Track how many of each tool remains in inventory to give live updates on what is on hand
- Add ability to send out notification emails to rental clients about return deadlines
- Add ability to capture photos using a webcam or other type of camera interface

Additional to these the team produced other goals crucial to the project's success.

- Create and house scanner module.
- Allow the system to be securely used by both administrators and students. (Administrator login)
- Store and display renter info and order by longest rented out item.
- Securely store information and back up data.

Secondary goals include the following.

- Barcodes and RFID tags created through the system using a tool database.
- Connect printer to the system as well as automatically detect the camera and scanner.

- Create an easily assembled 3d printed case in the case that the scanner needs to be repaired.

At the end of the semester the team presented a tool rental system with necessary functionality and a functioning scanner housed in a 3D printed case. The program had a clean and efficient GUI that was able to add tools to the database, add users to the database with their email, rent out tools, search the tools and users, take pictures, adjust the quantity and name of tools in stock, and many more features. The scanner was able to scan barcodes as well as read and write RFID tags to interact with the software application.

The team has presented a fully fleshed out and tested tool rental system that meets all the requirements of the project, with a program interface that is intuitive and functional. The scanner has proper custom housing and will automatically connect to the program and has no issues.

4. Results

In this section, the test results, validation results, and modeling and simulation analysis of the system are discussed to provide a detailed explanation of the tool rental system.

4.1 Beta Prototype Test Results

An Arduino microcontroller was selected to be used to manage the hardware systems since it is cheap and has much widely available documentation. It also easily communicates with the computer via serial communication that is received on windows as a virtual COM port. Several Arduino micro controllers were tested early on, and it was found that the Arduino Nano was the best fit as it provided enough functionality, enough input/output pins, and the overall size of the microcontroller was small enough to fit adequately into a scanner of handheld size.

The barcode scanner selected was the GM67, this was chosen since it is an integrated scanner module that has a laser and camera and outputs the value of scanned barcode data via UART. The Arduino can then receive the UART data and send it via the serial port to the computer. For the RFID reader/writer the RC522 was used. This is based on an NXP chipset and is a cheap and easy to use module with good documentation and libraries for the Arduino. An image of the assembled hardware for the custom handheld scanner can be seen below in figure 5.

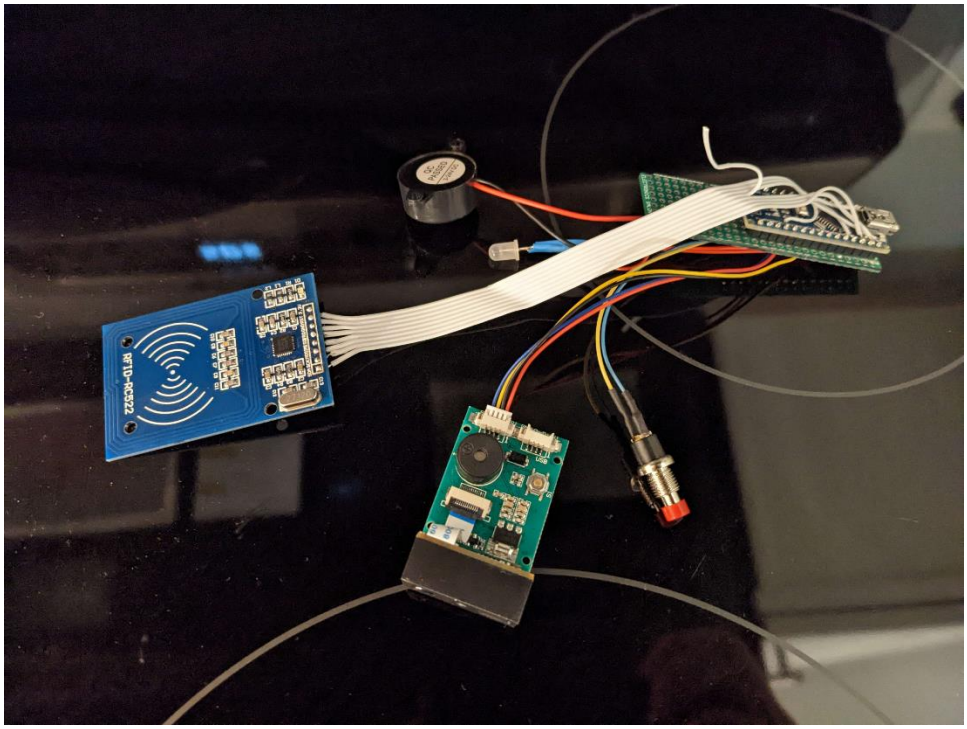


Figure 5. Assembled Hardware

Python was used to make a front and back end for the software system. This was selected since it is a simple programming language that complies easily with applications that run on windows. There is also an open-source library that could be used to interact with the COM port making data transmission to and from the Arduino easier.

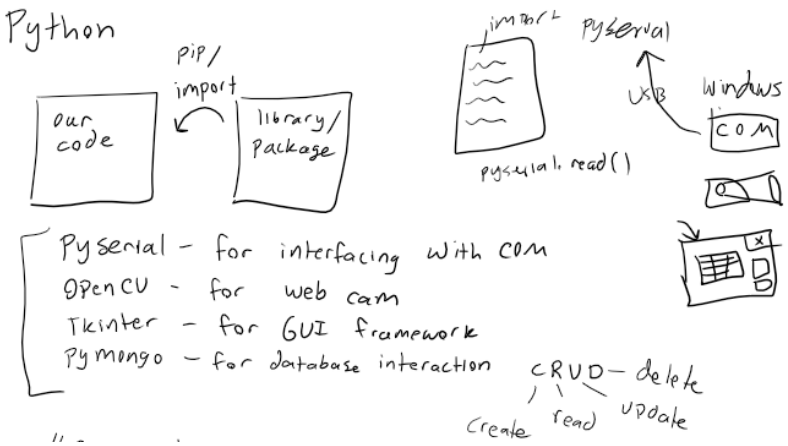


Figure 6. Python Program Design

Python is a remarkably simple language and has a very straightforward syntax and provides a significant number of libraries that can be used. One of which was the Tkinter python library which is a library for creating GUI frameworks and is exceedingly popular. The database used for Python was TinyDB, a very lightweight flexible object-oriented database. It is much more versatile and simpler to use than SQL and can run locally with little extra computing power and no additional software required outside the packaged Python library. It also stores data as JSON so it can easily be viewed and edited in its raw form outside of the code. The code was re-bundled and wrapped in custom functions to make the implementation easier for basic CRUD (create, read, update, delete) functionality. The RFID tags chosen will be Mifare Classics with 1kb of EEPROM (Electrically Erasable Programmable Read-only Memory) memory. These are low cost, commercially available RFID tags that are easy to read and write from while also providing encryption capability. After speaking with the FIZ staff the UI was designed to meet the criteria set out by the project requirements. The following renders in figures 7 & 8 were made to demonstrate the intended look of the final version of the software and provide a good usable interface. It was approved by the project Mentor prior to breaking ground on the Python code.

With these software and hardware tools combined it allows for the creation of a tool rental system for FIZ that is intuitive and easy to use.

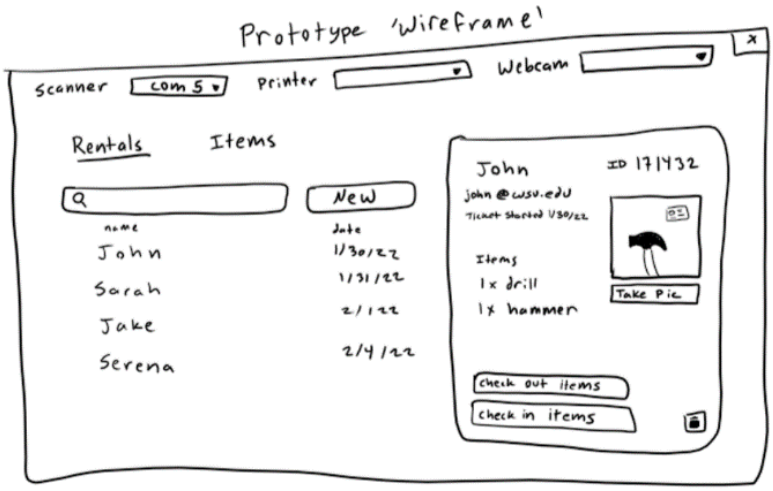


Figure 7. GUI Wireframe: Rentals Page

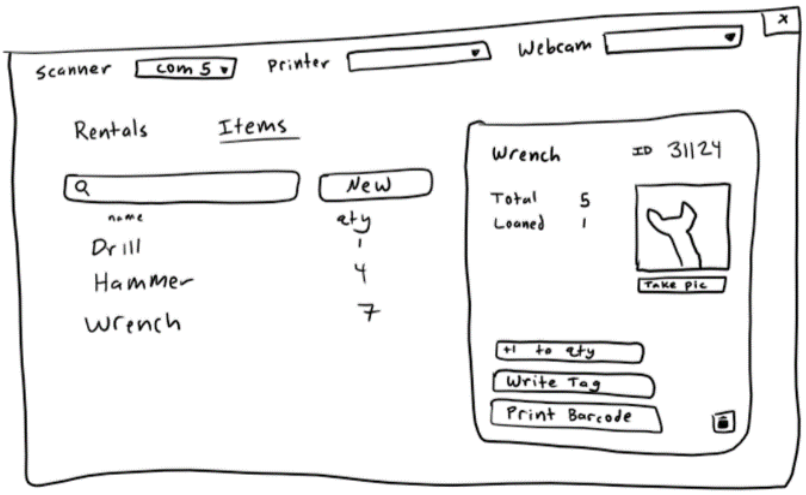


Figure 8. GUI Wireframe: Items Page

The implementation of such a system will allow for the FIZ to increase efficiency by reducing employee time to check in or out tools to students as well as provide data on what tools are where, how many of each there are, and the condition of the tools. The aim of the project is to save the FIZ money and time while providing a service that is necessary to the basic operation

of their day-to-day function as an organization. Each item that is owned by the FIZ will be tagged with an RFID sticker or a barcode and tracked in a database like traditional inventory management. Each item will be searchable in the database providing more information about the items and where they are. As a function of the space, students can check out tools for use on projects, so the rentals aspect is a large part of the project. The rentals database will define who has what items so at any given time a FIZ employee can see which students have checked out which items. This will result in less loss of tools and help with the overall organization of the space.

The tool rental system needs software that any user can easily rent out and turn in tools. A basic program must include a tools tab and a rentals tab. The rentals tab as seen in figure 7, would allow you to rent out tools, turn in tools, and see the details of those who have rented out tools. The scanner would be selected from a comport. The printer would be selected from a list as well as the webcam. You would also be able to search by name or other criteria for those who have rented out items.

The tools tab would allow the user to add and remove tool types from the program as well as increase and decrease the number of tools in the system. As seen in figure 8, when selecting a tool, the software will display details such as total count and loaned out tools. A search bar also allows you to search up the tool or rental you wish to find, the search bar can also be used to find which people have a certain tool loaned to them. As the FIZ adds more tools it may be difficult to find a tool simply by looking through a list. It was necessary to include both an RFID and barcode scanner in the scanner module so both could be scanned at once. The only commercially viable options that were already on the market with this functionality were extremely expensive or came as a separate barcode scanner and RFID reader. Since the budget was set at $1000, it

was decided that a custom scanner would be built to incorporate the needed functionality. The unit must communicate with the computer in the FIZ that is running software that can interact with the user and a database to keep track of inventory and rentals. The computer software had to have databases of locally stored data and a way for the FIZ users to interact with the system to read, write, delete, and modify select data in a user-friendly manner. From this, the following system was initially suggested to incorporate the project specifications.

It was determined that an RFID read/write module could be connected to a microcontroller that communicated via USB. The microcontroller could manage input and outputs of the RFID reader and barcode scanner and aggregate necessary data to interact with the program. The program would run front end code visible to the user for functionality as well as a backend that managed the serial communication with the microcontroller, database interaction, printing, and basic functionality such as barcode and ID generation for the tools and rentals.

The team frequently presented all the prototypes and demonstrated functionality to ensure it aligned with the vision set forth by the FIZ staff. Figure 5 shows the prototype hardware completely assembled outside of the 3D printed case. There is the RFID module, the barcode scanner, the Arduino, a piezo speaker, a button to trigger write and read, and an RGB LED. As of currently the system is functional in that rentals and tools can be created in the database, barcodes can be generated for tools, user data can be stored for students, images of the tools and rentals can be stored, the software can write to RFID tags, tools and rentals can be modified and deleted, and items can be checked in and out to users using the scanner module.

Once the two systems (handheld scanner and software GUI) have been completed, rigorous testing took place to ensure that the entirety of the tool rental system worked 100% of the time. Testing was carried out manually by the team by going through each of the potential use cases

that this system covers and executing them on the program, this process was repeated over and over especially when new changes or fixes were made to the system. The tool rental system was also implemented in the FIZ early on as part of the testing process so that no use cases were left out of consideration for testing. Testing provided crucial insights as to which part of the program needed fixing and which parts needed error handling. Screenshots of the current and completed software GUI can be seen below in figures 9 & 10.

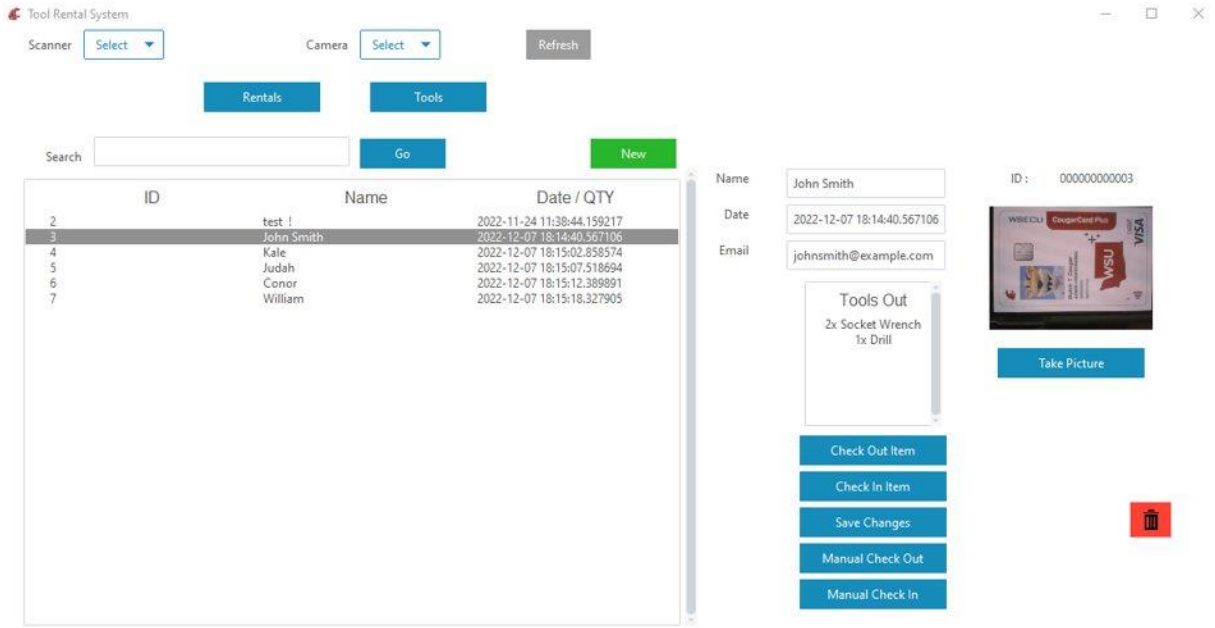


Figure 9. Completed GUI: Rentals Tab

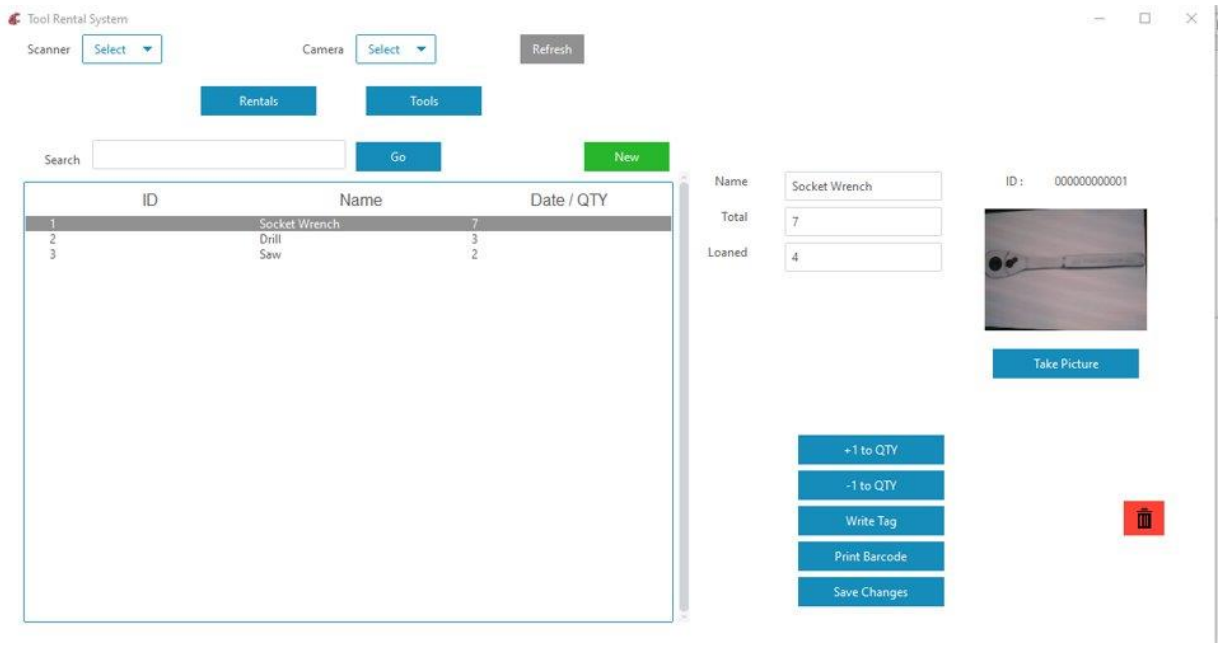


Figure 10. Completed GUI: Tools Tab

Once the hardware was completely assembled the hardware needed to be assembled into its final form. This meant designing a 3D printed handheld enclosure and soldering all connections instead of using a breadboard. The render for the 3D printed scanner housing can be found below in figure 11. The team had created for the project a list of specifications near the beginning of this project. These project specifications have helped to determine if the project is on the right track. The list of specifications can be found below in figure 12. It should be noted that all these specifications have been met and that the FIZ has deemed them to be satisfactory.

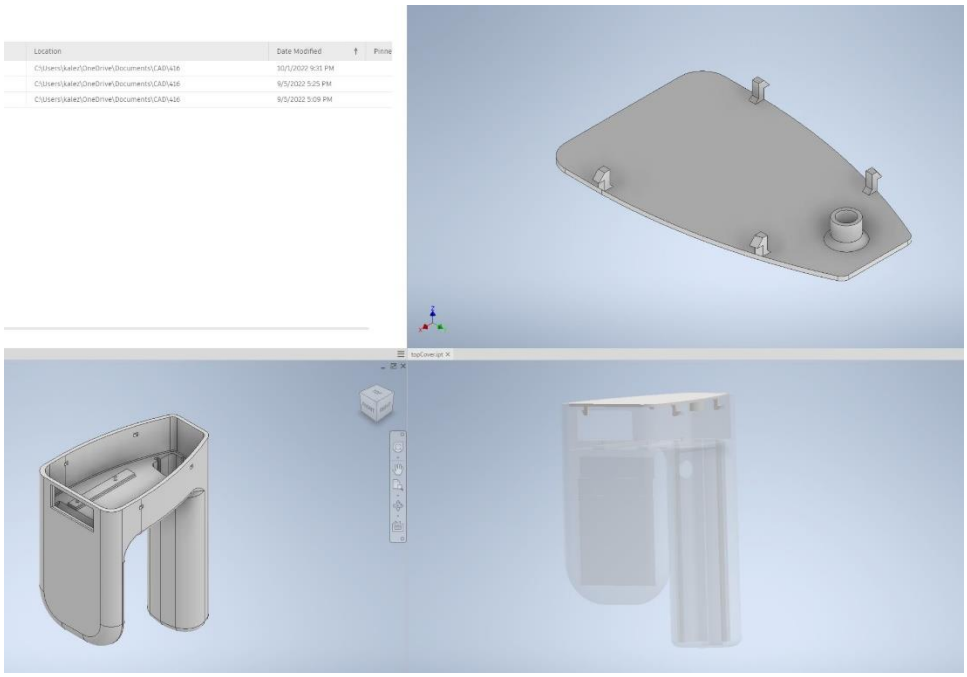


Figure 11. 3D Model for Scanner Housing

Item #	Technical Specification	Needs of technical specification	Metric for determining needs are met.
1	GUI	Intuitive and simple to use.	Any person with no previous knowledge of design can use the GUI.
2	GUI	Responsive to user input.	Switching between tabs will not take more than one second and should not freeze.
3	GUI	Aesthetically Pleasing.	Should not have overly bright colors and show clear differences between clickable and non-clickable items.
4	Main Menu	Hub for all tabs	The main menu will be the hub for all tabs including settings, rent out, turn in, login, etc..
5	Main Menu	Default page to return to	The main menu will be able to be returned into when in the middle of a process without complications.
6	Rent Out Tab	Rent out tools	Tools need to be rented out and a picture taken with their ID without complications. User should not be in doubt that the process is complete.
7	Turn in Tab	Turn in tools	Tools need to be turned in without complications. User should not be in doubt the process is complete.
8	Rented Out Tab	Show Tools Rented Out	Tools rented out and the corresponding ID card should be displayed.
9	Rented Out Tab	Search functionality	Names, dates, tool types, will be able to be searched. Searching should be intuitive and not require exact wording. Alternatively drop-down filters for dates and tool types can be combined with name searches.
10	Tool Tab	Add and remove tools from the data base	Tool types will be added to data base as well as increasing the supply of tool types in the data base.
11	Software & Scanner	Integration with software	Tool will be easily recognizable by software. User should be alerted when scanner is not found.
12	Scanner	Visual Indicators	Users will know visually when to use the scanner. Lights present on the scanner are one option.
13	Scanner	Handheld device	Scanner enclosure will be connected to grip and trigger.
14	Scanner	Enclosure	Scanner enclosure that will protect hardware from drops above 10ft. Functionality after multiple drops.
15	Scanner	Printable Enclosure	Enclosure will be able to be printed and assembled easily.

Figure 12. Project Specifications

4.2 Beta Prototype Validation Results

Prototype validation was done by checking datasheets of each component and running physical

and simulated tests to validate functionality. The scanner itself has been designed to last 4 years and the software has been designed to require no major updates up to 10 million database entries. The scanner has is built around the Arduino microcontroller with an RFID reader, barcode scanner, button, LED, and case. The Arduino, RFID reader, and scanner each had lifecycle estimates that exceeded the proposed lifespan estimates as per the datasheets. Usage was estimated by surveying existing data from the FIZ on number of checkouts monthly. Those numbers were compared to hours or actions on each of the components to ensure compliance. For instance, LED lifespan is rated by number of hours used and buttons are rated by press cycles. All the components used were determined to be reliable enough to meet customer requirements. The scanner case itself was designed in Autodesk Inventor, a powerful 3D CAD modeling software. Built-in tools allow stress tests to be conducted on the part to determine areas of weakness and simulate forces to see breakage. This was done on the model to determine the ideal wall thickness of the scanner and curved edges were added in joins identified as weak to better distribute load force. Breaks were then simulated and determined to be non-dangerous (I.e., the weakest areas just have cables, no sharp edges or exposed power sources. Extra slack was added in wiring accordingly to allow flexibility in case of bends or breakage in production. The final model passed beyond forces that are expected from dropping the scanner from 5ft above ground level onto a hard surface and well as resalable force from user handling. When the final model was printed, a break test was done on the actual item to ensure it would not break. Ultimately the first sign of failure was the front of the scanner caving in following forcefully ramming the item into a table at high speed. The customer is not expected to subject the device to conditions such as these. The software is designed to work using a local database and the program supports up to 99999999 entries in rentals or in items. The amount of space these

entries take up on the computer is minimal as the files are stored in plain text JSON. Images can take up space but are limited in size due to storage at low resolution and therefore as long as space exists on the drive that holds the program, operation will continue.

The program was tested with unit testing that validated outputs of functions used by the program, but physical models were tested as well. Each build was used to perform a succession of check in and checkouts with the barcode scanner and RFID reader. RFID writing was tested as well. During each test run the LED was observed for normal operation and the button was tested. After the first unit was qualified, it was released to the FIZ for use.

Integration testing was done by giving the device to the FIZ and collecting user feedback. The system is currently being used by FIZ staff with no issues or errors reported so far. Some suggestions have come in such as ensuring that rentals checked out for more than one week are marked as late. These suggestions have been implemented and updated program files have been provided to the customer. Ultimately integration testing has yielded successful results and the program is actively being used to improve FIZ processes.

4.3 Analysis, Modeling, and Simulating

This project being made up of two separate major sections. software and hardware. It is important to make sure each is working as they are supposed to. Unit tests can be written and then used to check that the individual pieces within the software are behaving as they should. This would allow us to make sure that less bugs will be present when testing the software and hardware together. The hardware itself is made of electronic components and the case for those components. We can run basic structural integrity modeling for the 3D design for this case. This will ensure that the case won't run structural issues since it should last a long time before

needing to be replaced. The 3D model for the case can also be optimized for printing using some of this information so the case doesn't have unnecessary weight from excess material. Keeping in mind the way in which additive manufacturing happens, the design needs to be catered to good user experience and its ability to be successfully printed. The best way for us to get information on the software and hardware together is to use feedback from those who are using the product. This means giving the client the product early so it can be used in its intended work environment. This would allow the client to give the necessary feedback on any issues or bugs that were not found previously.

5. Impact Analysis

The need for proper organization and tracking is crucial for any rental system. Without a reliable system in place items can be lost and the process of renting anything becomes muddled. Quality rental systems are a must in the fast-moving markets like we have today. The tool rental system that we have created makes the process of renting tools seamless. The system is as user friendly and as organized as possible, so it is not difficult or confusing to use. The system is also very cheap and will provide organization and tool tracking that will make the process of renting tools efficient and worry free. The tool rental system will ultimately save lots of money. Without one, it is easy for tools to go missing therefore needing to be replaced. Many of the tools at the FIZ are much more than the scanner to build, including labor. Truly, the tool rental system is an asset to the FIZ.

6. Ethical Issues affiliated with the design

To keep the design simple the team designed a tool rental system that is not very secure. The design as it is, was intended to be as simple as could be, only a tool for the FIZ, and because of that there are no guards against bad actors. Below is a list of security risks.

a. **No login requirement,** there is nothing preventing random people from using the tool rental system. It's possible for anyone to edit anything.
b. **No guards to deletion,** there is nothing to prevent anyone from deleting anything, much information could be lost because of negligence or malice.
c. **No guards to false check in**, there are no in-built systems of preventing people from checking in a tool and simply stealing the tool right afterwards.

For future projects it would be best to address some of these concerns in future projects involving the tool rental system. Until then having a FIZ leader check in and check out tools to its members would be the best solution to prevent theft or abuse.

7. Engineering Economic Analysis

With the advancements of technology today, the team was able to minimize the budget for this project and the cost of the tool rental system that has been created is extremely cheap and easy to manufacture. The total cost for the tool rental system is roughly $35. In table 1, the costs of each individual part that was used to build the tool rental system have been displayed. Since team Olympic used free and open-source software, there was no cost to develop it. The handheld scanner is the only portion of the system that will be paid for, and it is low cost. These prices are subject to variations however relatively and in the future the scanner will be very low cost especially compared to market equivalents. The cost calculation is based on the price of 1 scanner unit.

Component Name	Quantity	Cost
Software Application	1	Free
GM67 Barcode Scanner	1	$16.42
RC522 RFID Module	1	$1.50
Arduino Nano	1	$12.00
Standard (small) Push Button	1	$2.00

220 Ohm Resistor	3	$0.10
10k Ohm Resistor	1	$0.10
Standard RGB LED	1	$1.00
Piezo Buzzer	1	$1.50
3D Printed Scanner Case	1	Price Varies

Table 1. Tool Rental System Unit Cost

The tool rental system will come with an instruction manual that goes over each part mentioned above and how it is assembled with the other components. The cheap cost of this system is a major advantage to customers. With a tool rental system like this in place, businesses will not be losing tools, saving them hundreds if not thousands of dollars on lost tools that must be replaced. Without a tool rental system like the one constructed in this project, it can be very costly when tools are being lost, so having a system in place to track all the tools will be worth the cost. This system will provide an extremely important service at an extremely low price point.

8. Limitations and Recommendations

As discussed in the ethical issues section, the current system is not secure. Security was not a concern of the FIZ but even so addressing this issue should be a goal for future projects. Another concern is the lack of a printed PCB for the tool rental system scanner Arduino. Soldering everything onto a perf board was a difficult task, of the two models that were created only one worked [6]. The capacity of human error should be addressed by creating a PCB that can be easily assembled. If the PCB is as wide as the Arduino and about 1.5 times as long there will be no issue and making multiple of these scanners would not be a problem.

If the tool rental system is going to be used by groups on the WSU campus adding a unique ID to tools may be a requirement. This would require re-working much of the tool rental system. All

the code has been provided to the FIZ for them to re-work the TRS as they wish.

9. Conclusions and Future Work

The tool rental system was designed to be simple and easy to use. Mae Siev was satisfied with the team Olympics' results and now the tool rental system will aid FIZ members in keeping track of their tools. The members of team Olympic can proudly say that they have learned much from this project and the experience that the team has gained will be helpful in every collaborative project to come. The tool rental system also has a complicated future ahead. FIZ plans to make a conveyor system that will connect to the tool rental system to automatically select and rent out tools. There are also plans for the tool rental system to be used in other clubs at Washington State University. The tool rental system will be the basis of many projects and its future is bright.

Acknowledgements

[1] Mae Siev

[2] Dr. José G. Delgado-Frias

References

[1] “Washington State University.” Frank Innovation Zone | Washington State University, https://vcea.wsu.edu/fiz/

[2] J. M. Hughes. Arduino: A Technical Reference. O'Reilly Media, 2016. https://ntserver1.wsulibs.wsu.edu:2171/lib/wsu/detail.action?docID=4543968

[3] “Adafruit.” Adafruit PN532 RFID/NFC Breakout and Shield | Lady Ada adafruit-pn532-rfid-nfc.pdf

[4] Learn Python - Free Interactive Python Tutorial

[5] Tkinter Tutorial (pythontutorial.net)

[6] Breadboard To Perfboard | The Soldering Station

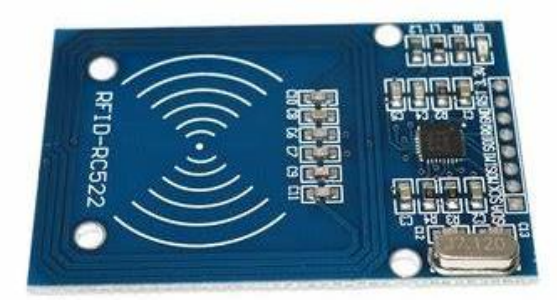
Tool Rental System Instruction Manual

Contents

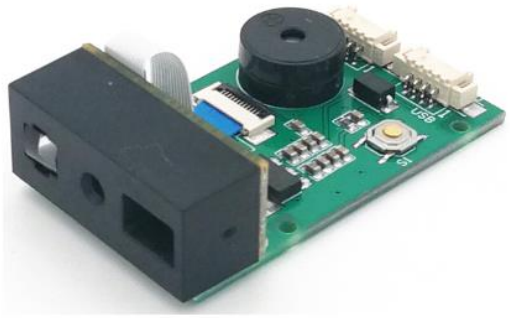
Scanner Part List

RC522 RFID Reader, GM67 Barcode Scanner, Arduino Nano C, standard RGB Led, Standard piezo buzzard, three 220-Ohm resistors and one 10k-Ohm resistor.

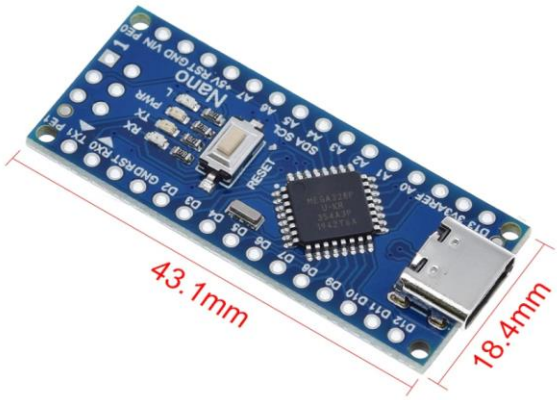
The RC522 can read and write to RFID to MIFARE RFID cards and tags.



GM67 Barcode Scanner



The Arduino Micro will connect all components of the board and is the processing unit of the scanner.



Standard RGB Led and Piezo Buzzard.

Three 220 Ohm resistors.

One 10k Ohm resistor.

Standard (small) Push Button

Scanner Construction Preparation

The Tool Rental System scanner does not have a PCB and creating one would save a lot of time. To wire the scanner parts together you will need a **Perf Board** and to connect all the components together you will have to cut the perf board to the size of the Arduino Microcontroller.

You will need the following items. **Kynar wire**: multiple colors will help identification. **Wire Cutter, Wire Strippers,** and **Needle nose Pliers**: these will help with general manipulation of wires. **0.8mm Rosin Core Solder, Soldering Iron, and Solder Paste**: to connect wires to the **Perf Board**. **Tweezers** might also help. **"Helping Hands"**: are not necessary but will be extremely helpful for keeping the **Perf Board** still while soldering wires onto it.

Scanner Wiring

Most components will be wired directly onto the board, but other components will be connected to resistors first. Connect the following

$$\boldsymbol{RFID \rightarrow Arduino}$$

$$IRQ \rightarrow NA$$

$$SCK \rightarrow D13$$

$$MISO \rightarrow D12$$

$$MOSI \rightarrow D11$$

$$SDA \rightarrow D10$$

$$RST \rightarrow D9$$

$$GND \rightarrow GND$$

$$3.3V \rightarrow 3V3$$

$$\boldsymbol{Barcode\ Scanner \rightarrow Arduino}$$

$$5V \rightarrow 5V$$

$$TX \rightarrow D5$$

$$RX \rightarrow D6$$

$$GND \rightarrow GND$$

$$\boldsymbol{Piezo\ Buzzard \rightarrow Arduino}$$

$$Pin_{+} \rightarrow D8$$

$$Pin_{-} \rightarrow GND$$

$$\boldsymbol{Button \rightarrow Arduino}$$

$$Pin_1 \rightarrow D2 \rightarrow 10k\Omega$$

$$Pin_2 \rightarrow GND$$

$$\boldsymbol{RGB\ LED \rightarrow Arduino}$$

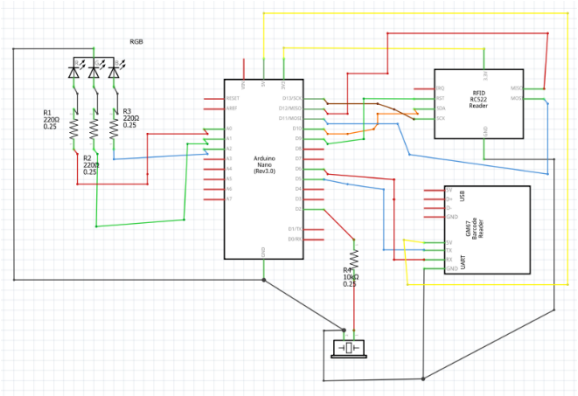
$$Pin_R \rightarrow 220\Omega \rightarrow A0$$

$$Pin_G \rightarrow 220\Omega \rightarrow A1$$

$$Pin_B \rightarrow 220\Omega \rightarrow A2$$

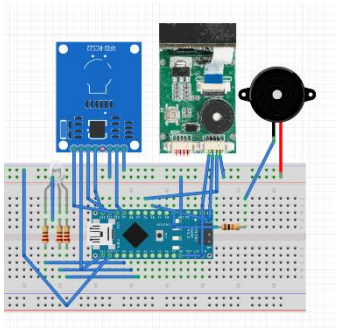
$$Pin_{GND} \rightarrow GND$$

It's important that all wiring happens on the inside edge of the Perf Board if you are using it. Make sure also the Perf Board is only as wide as the Arduino and not too much longer. If everything is wired correctly it should match the following schematic.



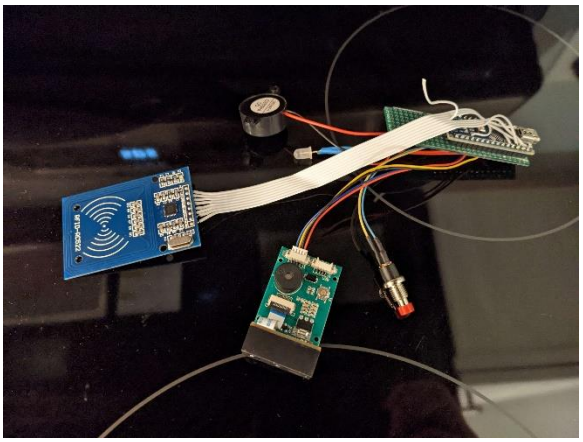
Scanner Testing

It's a good idea to test components before soldering. If you have never constructed this



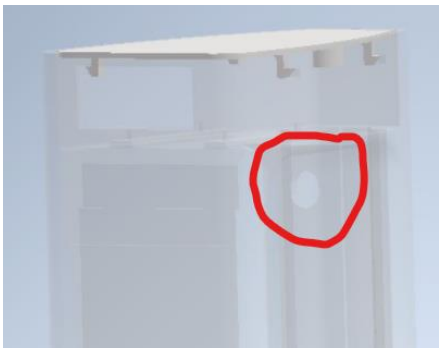
scanner, being familiar with the layout will lead to less mistakes. For example, using a bread board before assembly is good practice.

Compare your design also to the following picture of the soldered hardware.



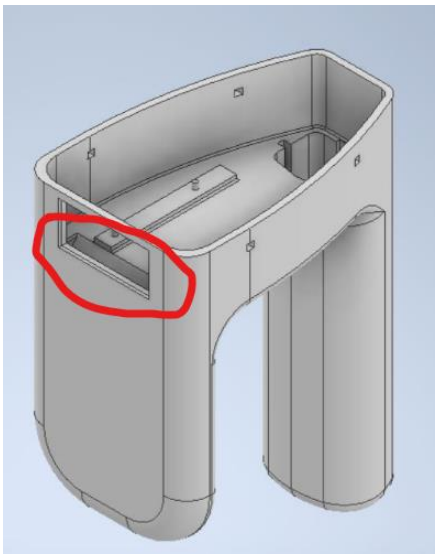
Scanner Final Assembly

In the trigger spot of the case insert the button making room for the next step.

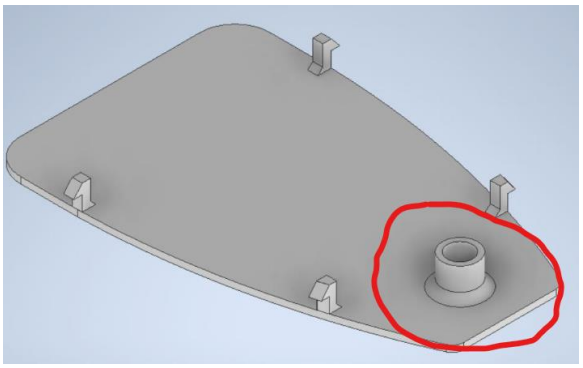


Once construction of the wiring is complete you will first have to insert the Arduino into the **hilt** of the scanner.

The RC522 will be placed in a slot in front of the scanner. Make sure that there is enough room for the GM67 above it. If there is not you may need to reprint the scanner case to a higher quality setting.



In the box-like area above RC522 you will put the GM67. Make sure you aren't forcing it in. Next you will place the RGB Led in a circular slot on the case.



One everything is in place snap on the lid, and you’re done!