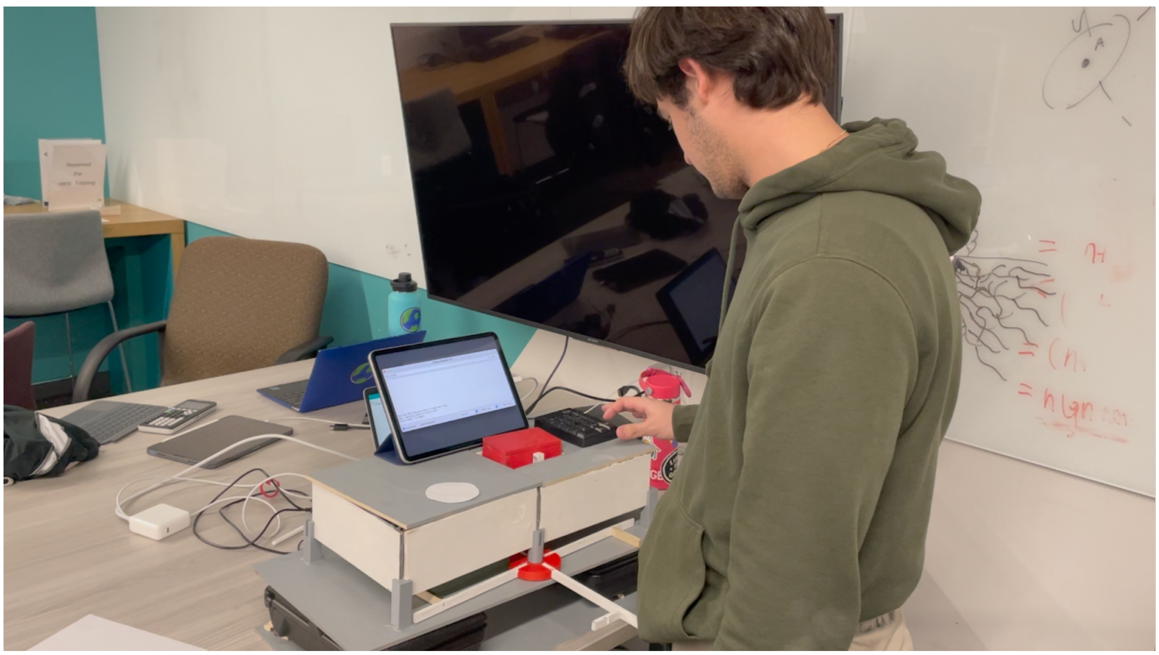
Cornerstone of Engineering

TO: Susan Freeman, Northeastern University

FROM: Team 1

DATE: 22nd April 2022

SUBJECT: Final Project Report



The To-go Tracker System

Our solution to the Dining Hall Problem

Leon Michael Frank Alex

Cover Letter

Professor Freeman,

This is a brief summary of the design process used on our project and some lessons we learned. We hope you enjoy this recap.

Problem statement

"Needing to eat within dining halls makes dining a greater challenge for students because it decreases their flexibility—also limiting accessibility to existing dining infrastructure. Alternatives currently require too much foresight or are environmentally wasteful."

Ideate

When ideating, we used the Gallery Method and focused most on how to dispense a stack of boxes one at a time. Together, we generated six possible solutions: gears, a carousel of hooks, a row of wheels, conveyor belts, a ramp, and a wheel. Using a Kepner-Tregoe Decision Matrix we decided on the wheel because it was the most easily scaled, it looked least likely to jam.

Build

First, we designed each section using CAD. We manufactured the parts with laser-cutting, 3D printing, and woodworking tools. Once an initial prototype was constructed from wood and plastic, we iterated each section multiple times to improve the final device.

Features

To use our system, the user types in their NUID, the machine scans a box, and the user turns the wheel to dispense it. Then, to return: the user scans the box and places it back in the stack.

Lessons

Our primary take-away was how important simplifying can be. After removing the central motor and greatly downsizing our project, we were able to better focus on the primary functionality.

Thank you for reading our report, and thank you for a fantastic two semesters in Cornerstone.

With gratitude,

Alex, Frank, Leon, and Michael.

Table of Contents

Figures

Figure 1. Duncker Diagram [pg 22]

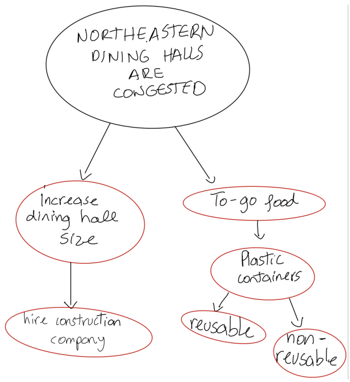


Figure 2. Promotion for the UCSB eco-to-go program [pg 25]

Figure 3. Harvard's to-go program. [pg 25]

Figure 4. Starbucks' reusable cup kiosk. [pg 25]

Figure 5. Bibliotheca's library checkout terminal. [pg 26]

Figure 6. RFID reader [pg 26]

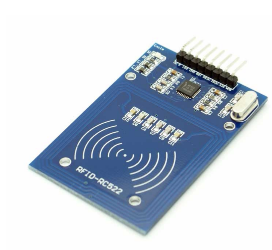


Figure 7. Gears design [pg 27]

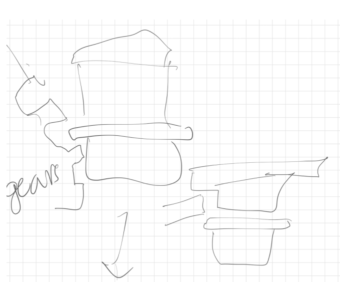


Figure 8. Hook design [pg 27]

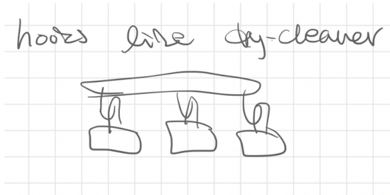


Figure 9. Pitching design [pg 27]

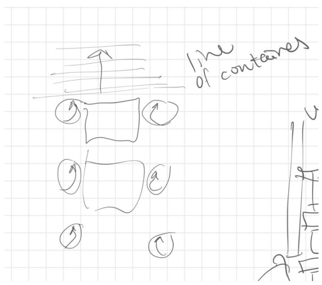


Figure 10. Conveyor design [pg 27]

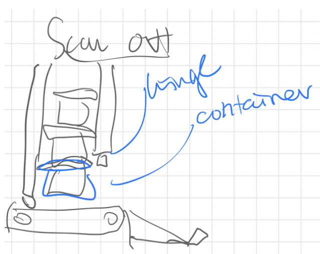


Figure 11. Ramp design [pg 27]

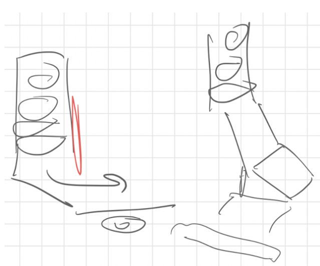


Figure 12. Wheel design [pg 27]

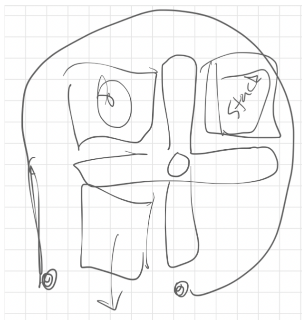


Figure 13. Initial idea for the wheel [pg 28]

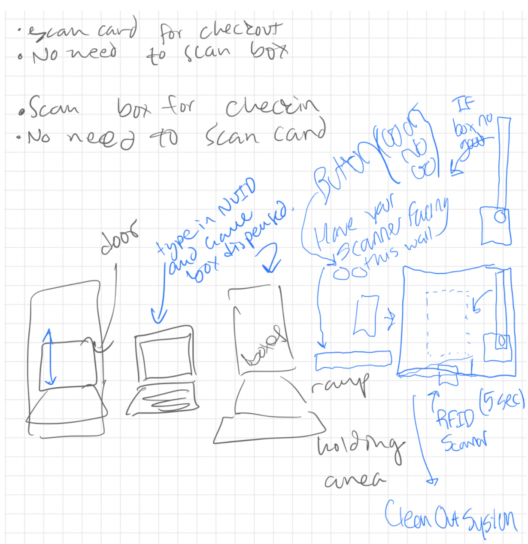


Figure 14. Flowchart [pg 28]

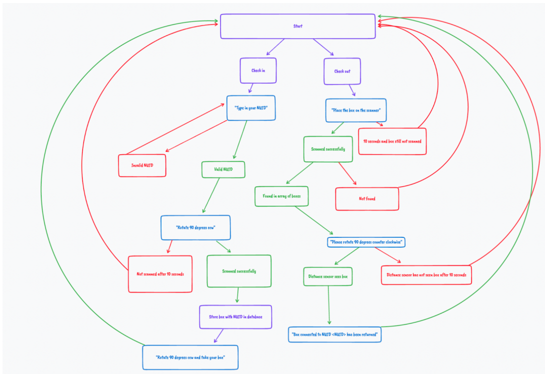


Figure 15. Prototype [pg 31]

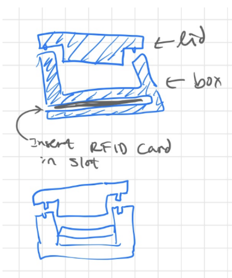


Figure 16. CAD [pg 31]

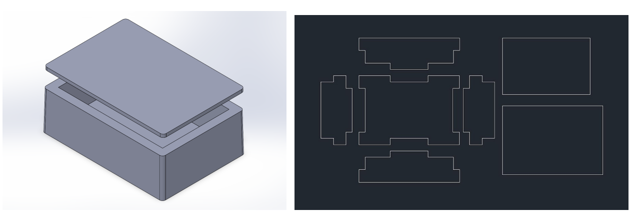


Figure 17. Laser cut [pg 31]

Figure 18. Painted boxes (final result) [pg 31]

Figure 19. Initial CAD Idea, wheel [pg 32]

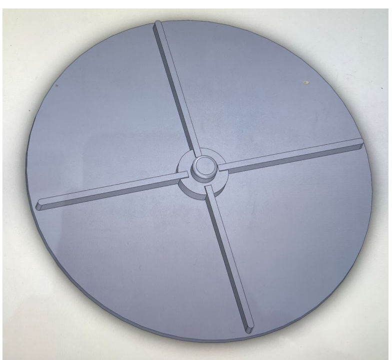


Figure 20. Solidworks model to be 3D printed [pg 32]

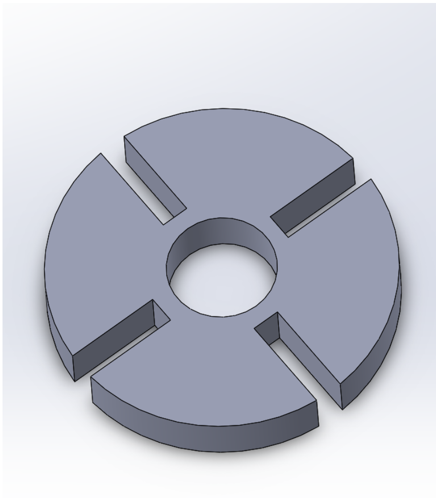


Figure 21. 3D Print of the result [pg 32]

Figure 22. Frank's Physics notes on Circular Motion [pg 32]

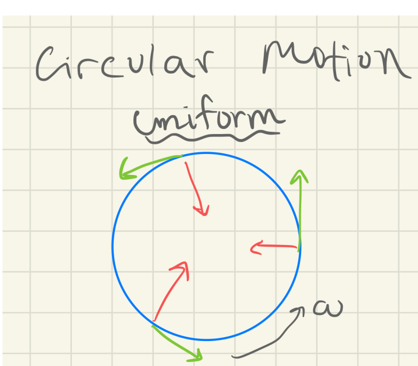


Figure 23. Idea for our solution [pg 32]

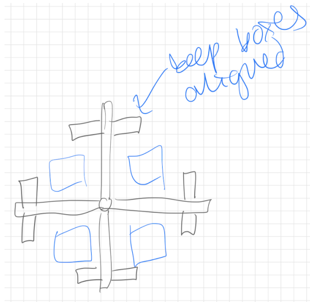


Figure 24. Implementation [pg 32]

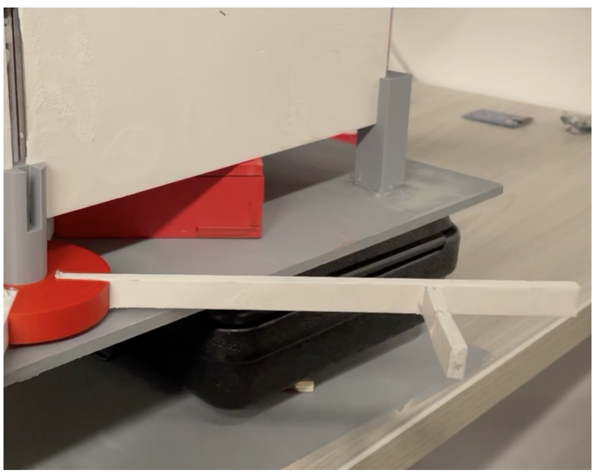


Figure 25. Initial sketches [pg 33]

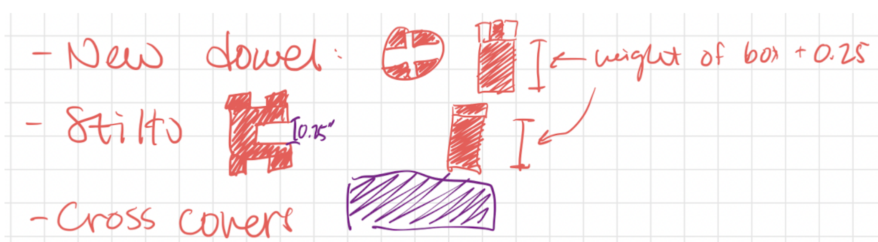


Figure 26. CAD Model [pg 33]

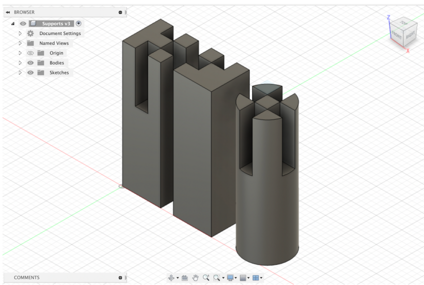


Figure 27. Final Prints [pg 33]

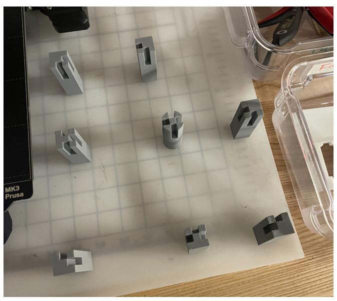


Figure 28. Initial Sketch with wheel [pg 33]

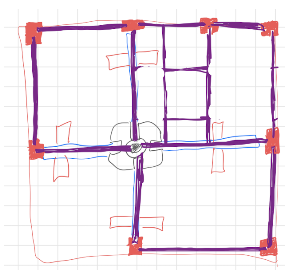


Figure 29. Paper model [pg 33]

Figure 30. Build [pg 33]

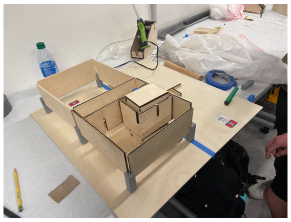


Figure 31. Painting [pg 33]

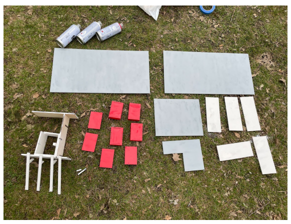


Figure 32. Wiring [pg 34]

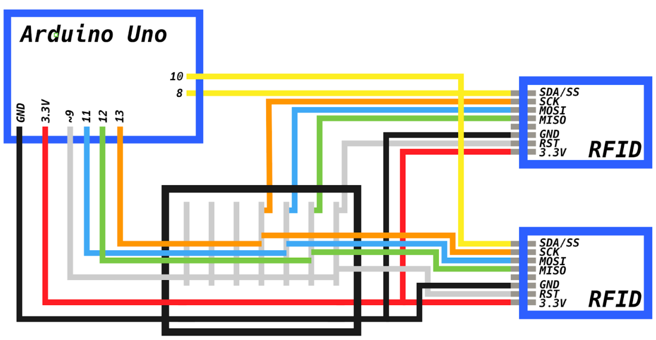


Figure 33. Code output [pg 34]

```
If there's input waiting:
    Read it: Either NUID or "."
        New box path (NUID):
        Validate NUID
        Read Box ID
        Conjoin NUID and Box ID in 2D Array
    Return path ("."):
        Read Box ID
        Find NUID for Box ID
        Mark box as returned
    Wait a bit, reset
```

Figure 34. Top view of product [pg 36]

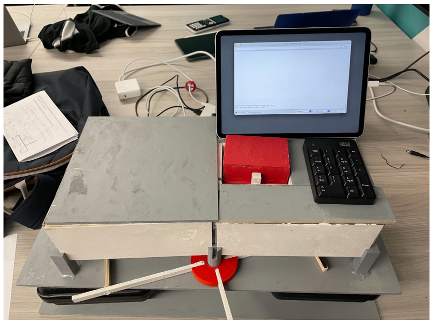


Figure 35. Front view of product [pg 36]

Figure 36. Side view of product [pg 36]

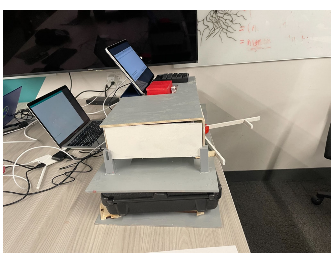
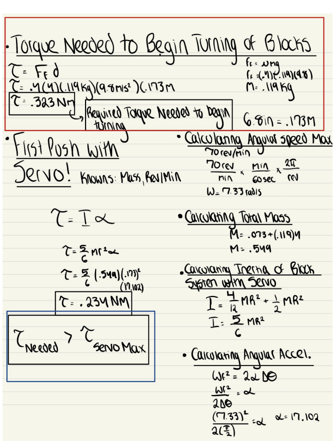


Figure 37: Physics work [pg 43]



• Torque Needed to Begin Turning of Blocks

$\tau = F_F d$

$\tau = .4(4)(.119\,Kg)(9.8\,m/s^2)(.173\,M)$

$\boxed{\tau = .323\,Nm}$ → Required Torque Needed to begin turning

$F_F = \mu mg$

$F_F = (.4)(.119)(9.8)$

$M = .119\,Kg$

$6.8\,in = .173\,M$

• First Push with Servo! Knowns: Mass, Rev/Min

$\tau = I\alpha$

$\tau = \frac{5}{6} Mr^2 \alpha$

$\tau = \frac{5}{6}(.549)(.173)^2 (17.102)$

$\boxed{\tau = .234\,NM}$

$\boxed{\tau_{Needed} > \tau_{servo\,Max}}$

• Calculating Angular speed Max

70 rev/min

$\frac{70\,rev}{min} \times \frac{min}{60\,sec} \times \frac{2\pi}{rev}$

$W = 7.33\,rad/s$

• Calculating Total Mass

$M = .073 + (.119)4$

$M = .549$

• Calculating Inertia of Block System with Servo

$I = \frac{4}{12} MR^2 + \frac{1}{2} MR^2$

$I = \frac{5}{6} MR^2$

• Calculating Angular Accel.

$W_F^2 = 2\alpha\Delta\theta$

$\frac{W_F^2}{2\Delta\theta} = \alpha$

$\frac{(7.33)^2}{2(\frac{\pi}{2})} = \alpha$ $\alpha = 17.102$

Tables

Table 1: Kepner Tregoe Problem Analysis (p. 10-11)

	Is	Is Not	Distinction	Possible Cause
What (Identity)	Dining halls are congested and there are no good alternatives	The quality of the food, the locations of the dining halls, and the hours	The process of dining is the trouble, not the food itself or access to it.	You have to eat in the dining hall.
When (Timing)	At peak hours and when the patron is very busy	At closing, between meal-times, on lighter-workload days.	When a lot of people need to eat at the same time or when you don’t have a lot of time to eat, dining in the dining hall can be challenging.	Society has defined “standard” mealtimes that everyone tries to eat during; sometimes one’s schedule is exceedingly busy with few gaps. And those gap times fall in line with most students on campus
Where (Location)	Stetson West, International Village, Stetson East	Stetson West Outtakes, Restaurants near campus	Places where you don’t have to eat in the dining hall	The ability to take food to go frees up valuable space for the people who really want to dine at the establishment.
Extent (Magnitude)	Long lines outside dining halls at peak hours; students roam inside the dining hall waiting for seats; dining takes at least half an hour. Overcrowdedness prolongs people's exposure to possible airborne illnesses.	Finding a way to take out food from restaurants since that is already effective and taking place currently through students going to the restaurants themselves, or ordering online and having the food delivered directly to them.	Dining halls on campus suffer from overcrowdedness and make dining take significantly longer.	You have to eat inside the dining hall, so time per transaction is very high compared to a restaurant where diners take out their food.

Table 2: Rank Order Chart and KT Decision Matrix (p. 16)

Rank Order	Score	Description				
Simple	2	How easy is the system to manufacture and hence maintain?				
Resilient	5	How well does the system hold up to abuse?				
Fixable	3	How easy is it to fix the system? Can a user unjam it on their own for example?				
Small	1	How much space does the mechanism take up?				
Scalable	4	How easy is it to add capacity upon the success of the proof-of-concept?				

Decision Matrix							
Category	**Simple**	**Resilient**	**Fixable**	**Small**	**Scalable**	**Total**	
Weight	2	5	3	1	4		
Hooks	1	1	3	1	1	7	Unweighted
Gears	5	5	3	3	5	21	
Conveyor	3	5	1	3	5	17	
Ramp + Door	3	3	1	5	5	17	
Wheel	5	5	5	3	5	23	
Hooks	2	5	9	1	4	**21**	Weighted
Gears	10	25	9	3	20	**67**	
Conveyor	6	25	3	3	20	**57**	
Ramp + Door	6	15	3	5	20	**49**	
Wheel	10	25	15	3	20	***73***	

Table 3: Design Recap

Design Recap	Reasoning
Hooks	Ruled out for feasibility (obviously fragile, hooking the boxes was a complex problem in itself
Gears	Limited capacity and requires lots of space. *(In retrospect could have been made to work, a la needle dispenser)*
Conveyor	Research showed this was very hard to implement well; placing RFID readers unwieldy.
Ramp + Door	Very likely to jam with no mechanism for relieving the jam; not sure how to quickly raise and lower a door/release
Wheel	Footprint and capacity unrelated; less likely to jam, with recourse to unjam; simple, buildable mechanism.

Table 4: Bill of Materials (p. 24)

Item	Cost	Retailer	Purchaser	Use
Total	**$99.95**	**-**	**-**	**-**
Remaining	**$0.05**	**-**	**-**	**-**
Number Pad	$13.27	Microcenter	Frank	Blocking user from doing non-numeric inputs
1/4" Laser Cutter Wood (4 sheets)	$21.17	NU Bookstore	Frank	Boxes and dividers for spinning part
1/4" Laser Cutter Wood (2 sheets)	$10.58	NU Bookstore	Frank	More wood for frames
12" x 24" Craft Sheets (3 sheets)	$19.10	True Value	Frank	Wood for base
Spray Paint (White, Gray, Red)	$23.94	Blick Art Materials	Alex	For decoration
Gloves	$6.90	Target	Alex	For working with spray paint
Extra wires	$4.99	Microcenter	Frank	They were pretty and Frank so wanted them but fronted this cost himself instead of making his group members fund his indulgences.

Table 5: Ethical Arguments and Counter Arguments (p. 27)

Utilitarian Pro	Utilitarian Con
More one-time-use plastics will be saved than waste created from lost boxes.	More strain will be placed on dining halls, and boxes will still be lost whereas to eat in dining halls before reusable plates had to be used.

Deontological Pro	Deontological Con
Giving students more options and flexibility is good. Reducing use of one-time-use plastics is good.	Letting students take food and the possible waste that comes with that is bad. Creates the opportunity for excess waste where there once was none.

Definitions

CAD: Computer-Aided Design

Refers to using computers to help design components for manufacturing or design. Makes modeling to test and build more efficiently and precisely.

FYELIC: First-year Engineering Learning and Innovation Center

Makerspace for First-year Engineering students at Northeastern where they can receive help from upperclassmen students, use a variety of hand tools, power tools, laser cutters, and 3D printers to work on their Cornerstone Projects.

NUID:

A unique 9-digit number that serves as the key identifier for each student on Northeastern University records and systems.

Redboard:

Sparkfun Redboard is a circuit board created by Sparkfun with inputs and outputs based on the Arduino Uno which can be programmed and interfaced with using the Arduino IDE to connect and control different electronic components.

RFID: Radio Frequency Identification

Wireless system of readers and tags where the reader emits radio waves and receives signals from a tag carrying data. The data can be decoded into a message. In our project, we used the unique data on each tag to identify each to-go container.

Executive Summary

Congestion in Northeastern's dining halls is difficult at peak times due to congestion. The current take-out alternatives that the university provides either require foresight or are environmentally wasteful. The implementation of a new, environmentally-friendly take-out system would increase productivity and give the students an option to eat outside while enjoying a freshly cooked meal.

To create the product, we first researched patents and similar systems. We based the core idea of our product on Starbucks' cup checkout system and on the automated checkout and return system found in libraries. Using our newly gained knowledge of the topic, we started the build process. With the use of a KT-Decision Matrix, we decided on the wheel design for dispensing lunch boxes. After using CAD to design each component, we used a mixture of laser-cutting, 3D printing, and woodworking to manufacture the parts of our system. With the pieces ready to go, we wired the electronics, constructed the device, and wrote the code. Once it was constructed, we tested our device and analyzed it for changes we should make.

We found the wheel had trouble keeping the boxes aligned while spinning, so we added protrusions to help guide them. The frame and boxes required more fit and finish, so we added a fresh coat of paint. Our RFID readers were very delicate so we learned to be more careful with power when using them after replacing a few readers due to failures. After these changes, our prototype was ready to showcase!

To use our system, the user types in their NUID, the machine scans a box, and the user turns the wheel to dispense it. Then, to return the box, the user scans the box and places it back in the stack.

Section 1: Problem and Definition

Objectives, Functions, and Constraints

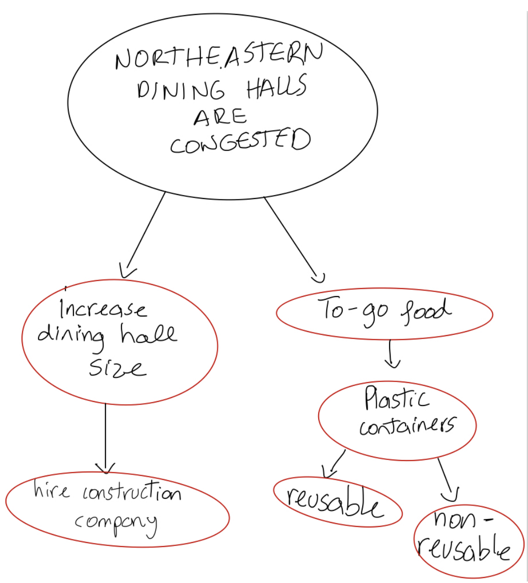
We had three main objectives for our project. First, we wanted it to help relieve congestion in the dining halls by making dining flexible for students. Second, we wanted it to make it easier and more efficient for students to take freshly made food to-go. Finally, we wanted to build an environmentally friendly system.

With this in mind, the essential functions of our project were to be able to let a student checkout their box, while tracking who checked it out, and to let a student return their box, and have the system know who returned which box.

Finally, we faced four primary constraints while working on the project. First, scale: we had to down-scale the project because a full-size model would not fit in any of our dorm rooms and would be nigh impossible to transport. Second, testing: our project was limited because ideally we would implement the system in the dining halls and get feedback, but due to other constraints this was not possible, so we had to compensate by receiving feedback from peers and by doing extensive research. Third, budget: due to the constraints of the course we were given a one hundred dollar budget which we could not exceed. That said, we were still able to make use of existing resources, such as Frank’s iPad and the Sparkfun kits we all had to purchase last semester.

Problem Statement

Figure 1: Duncker Diagram



To help identify the problem, we created a Duncker diagram, in which we stated two possible solutions. The first one was increasing dining hall sizes and the other was what if we didn't. Instead of increasing dining hall sizes, we could create a system that let students not have to eat in the dining hall. This led us down the track of wanting to make a device that would let students check out a to-go container to take their food elsewhere to eat, giving students another option when dining halls are too congested.

Original Problem Statement

Our initial problem statement, based on our defined objectives, desired functions, and identified constraints was this:

"Dining halls at Northeastern are too congested, but current to-go options either require foresight or are ecologically wasteful."

Table 1: Kepner-Tregoe Problem Analysis

	Is	Is Not	Distinction	Possible Cause
What (Identity)	Dining halls are congested and there are no good alternatives	The quality of the food, the locations of the dining halls, and the hours	The process of dining is the trouble, not the food itself or access to it.	You have to eat in the dining hall.
When (Timing)	At peak hours and when the patron is very busy	At closing, between meal-times, on lighter-workload days.	When a lot of people need to eat at the same time or when you don't have a lot of time to eat, dining in the dining hall can be challenging.	Society has defined "standard" mealtimes that everyone tries to eat during; sometimes one's schedule is exceedingly busy with few gaps. And those gap times fall in line with most students on campus
Where (Location)	Stetson West, International Village, Stetson East	Stetson West Outtakes, Restaurants near campus	Places where you don't have to eat in the dining hall	The ability to take food to go frees up valuable space for the people who really want to dine at the establishment.
Extent (Magnitude)	Long lines outside dining halls at peak hours; students roam inside the dining hall waiting for seats; dining takes	Finding a way to take out food from restaurants since that is already effective and taking place currently	Dining halls on campus suffer from overcrowdedness and make dining take significantly longer.	You have to eat inside the dining hall, so time per transaction is very high compared to a restaurant where diners take out their food.

	at least half an hour. Overcrowdedness prolongs people's exposure to possible airborne illnesses.	through students going to the restaurants themselves, or ordering online and having the food delivered directly to them.		

After identifying our problems by making a Kepner- Tregoe Problem Analysis chart, we ended up with the following Problem Statement:

Improved Problem Statement:

"Needing to eat within dining halls makes dining a greater challenge for students because it decreases their flexibility-also limiting accessibility to existing dining infrastructure. Alternatives currently require too much foresight or are environmentally wasteful"

Background

We performed background research to better understand checkout systems and food distribution in dining halls. We wanted to be better informed to answer the question: "What's the best way to allow students to take their food to go from dining halls?" Our research was broken down into six separate parts, 1) stakeholder needs, 2) foundational information, 3) best practices, 4) materials and components, 5) regulations and standards, and 6) intellectual property. For each topic, we asked a series of questions and did research to find answers to these questions and more that popped up during the research process. Based on this research, we are now able to summarize the following important aspects of our topic.

At first we wanted to understand if containers that would work with our system existed–we found patents for two: a Rigid Biodegradable Food Container [1], a Microwavable Food Container [2]. We also gained an understanding that making environmentally friendly and reusable containers is a really complicated branch of material science and that we would be better off spending our resources elsewhere after reading about Environmentally Safe Composites in Food Packaging Industry [3] along with food safety recommendations and requirements for packaging from the FDA [4]. There was a concern about room temperature leftovers [5], based on CDC guidance and about the safety of plant based products [6] as well, but we plan to refer to NU Dining for guidance around communicating these risks if this project is ever implemented since the safety of food is far beyond the domain of our project, and our project does not substantially increase these risks.

Next, we looked at existing solutions such as Food Lockers at Northeastern [7] and other institutions [8]. The single-use plastics in these existing solutions [9] was of particular concern to us. With this concern in mind, we learned about Harvard's [10] and Santa Barbara Community College's [11] systems for reusable container checkout systems. A concern when thinking about trays is that switching from trays to plates in recent years has been shown to reduce food waste [12], so we don't want to back-track on that progress. There is discussion that reusable containers can actually be more harmful [13] since they require the user to reuse the item a certain number of times before it offsets the additional cost of producing the product, but because our product makes reusing really easy and because reuse will be on a university-wide scale, these concerns are adequately addressed.

Figures 2 and 3: Promotion for the UCSB eco-to-go program and Harvard's program.

Starbucks [14] has also begun to implement a similar system to what we had in mind, just for coffee cups, in a few of their stores. We examined their video for inspiration [15], noting the dishwasher rack and sanitation methods especially and how they can already be performed using the equipment in the dining halls. Making use of the existing methods also ensures concerns such as allergies are addressed. Hence, we were able to narrow our scope to the specific mechanics of checking in and out.

Figure 4: Starbucks' reusable cup kiosk.

Automated checkout and return at libraries was our next source of inspiration. Initially, we found the websites for two companies that create checkout systems for libraries: Bibliotheca [16] and Envisionware [17]. A key takeaway from this aspect of the research was Bibliotheca's tagline:

“Self-service is only convenient if it's easy”, which is something we need to keep in mind as we design our own solution. We also learned the scanners use RFID to read books in a stack.

Figure 5: Bibliotheca’s library checkout terminal. Uses an RFID reader to scan the books the user wants to checkout.

A write-up by Paul Pandian [18] and a video by the Pasadena library [19] explain this process in detail. In short, a user signs in on a kiosk, the RFID reader in the station reads the RFID pads embedded in or placed on the book, and the user confirms the checkout. When it’s time to check-in, the user places the item in the kiosk, the kiosk looks up who had the item last then checks it back in, and the user confirms they’ve checked the item back in and receives a receipt. Somewhat technically, the Snell Locker system provides a good starting framework for how to incentivize students. Their policy is students have 7 days to return items, and then they are charged the full cost of them [20].

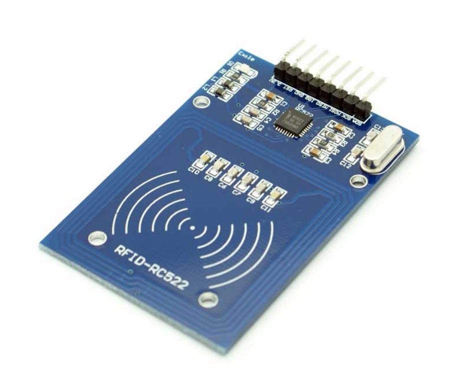


Figure 6: RFID reader which can interface with a Redboard.

All this background research was helpful in clarifying the problem we wanted to solve, and how we wanted to go about doing so.

Section 2: Generate and Decide

Generating Design Alternatives

The first step in Generating Design Alternatives was to reference the research we had performed. Key inspiration was conveyor belts, library automated checkout systems, Starbucks' cup checkout system, and Harvard's reusable container program.

The first problem to solve was how to dispense one cup at a time. To solve this problem we used the Gallery method, where we each made sketches for 3 minutes, and then afterwards collaborated to improve each member's ideas and combine some of our ideas. The main ideas we came up with were gears, hooks, pitching machine or conveyor, ramp with door, and wheel. These ideas were based on a variety of inspirations including the TSA Baggage Screening Conveyors, utensil dispensers, automated checkout systems at libraries, clothing carousels at laundromats, pitching machines, Pez dispensers, and more.

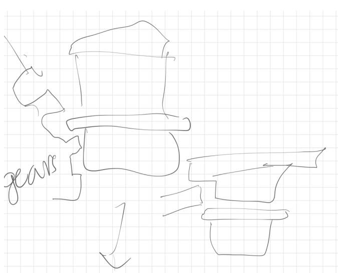


Figure 7: Gears

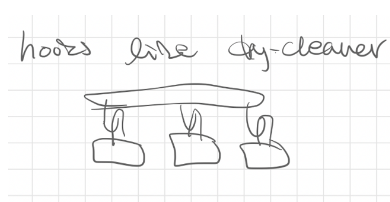


Figure 8: Hooks

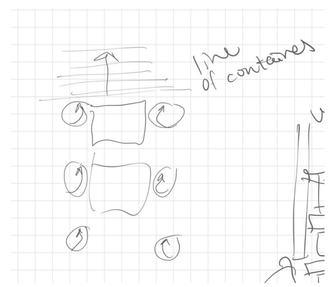


Figure 9: Pitching Machine

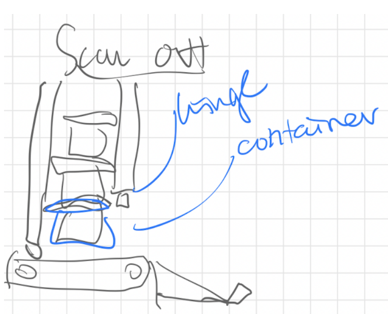


Figure 10: Conveyor

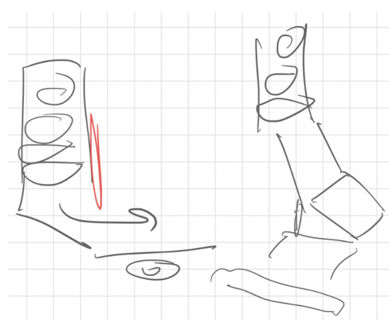


Figure 11: Ramp

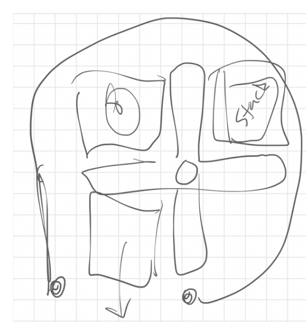


Figure 12: Wheel

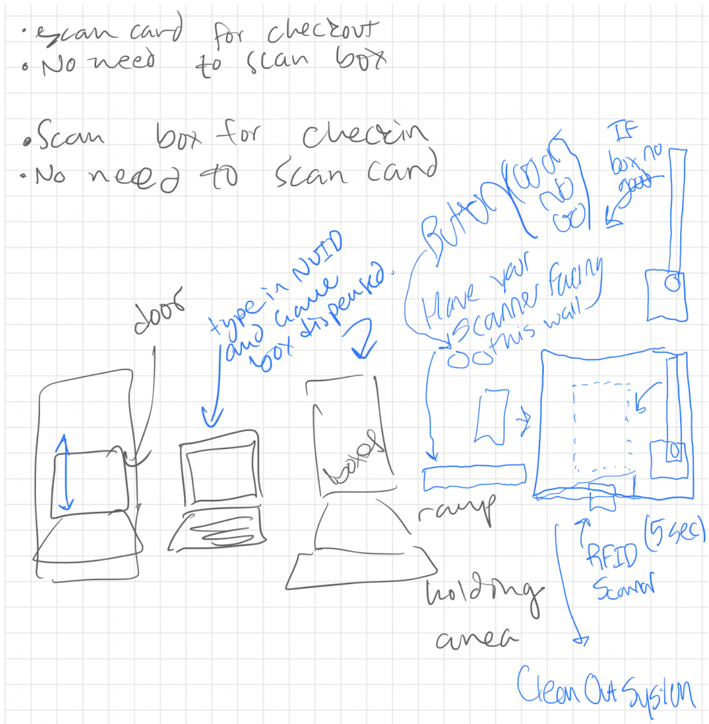


Figure 13: Thinking through the door, and initial idea for the wheel.

We also analyzed different reader components such as QR codes, barcodes, and RFID. We very quickly settled on RFID, both because Frank really wanted an excuse to use them and because they were very flexible while also not requiring things to be clean in order for them to work (thinking food could get all over these things, ew). Thinking through the user experience also helped us think through our code. To help us understand the user experience of our device we made a flowchart (Figure 14). This was the framework for our code moving forward.

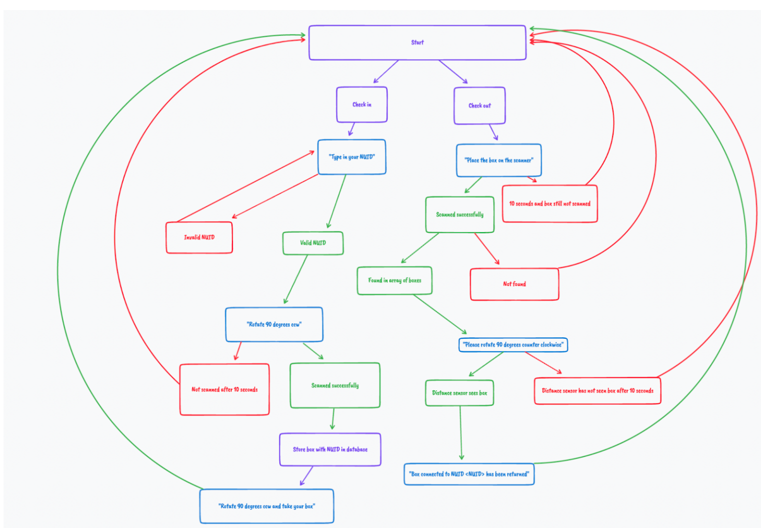


Figure 14: Initial flowchart for our code and user experience.

Evaluating Design Alternatives

To evaluate our ideas, we created a Rank Order Chart and used a Kepner-Tregoe Decision Matrix.

Table 2: Rank Order Chart and KT Decision Matrix

Rank Order	**Score**	**Description**					
Simple	2	How easy is the system to manufacture and hence maintain?					
Resilient	5	How well does the system hold up to abuse?					
Fixable	3	How easy is it to fix the system? Can a user unjam it on their own for example?					
Small	1	How much space does the mechanism take up?					
Scalable	4	How easy is it to add capacity upon the success of the proof-of-concept?					
Decision Matrix							
Category	**Simple**	**Resilient**	**Fixable**	**Small**	**Scalable**	**Total**	
Weight	2	5	3	1	4		
Hooks	1	1	3	1	1	7	Unweighted
Gears	5	5	3	3	5	21	
Conveyor	3	5	1	3	5	17	
Ramp + Door	3	3	1	5	5	17	
Wheel	5	5	5	3	5	23	
Hooks	2	5	9	1	4	**21**	Weighted
Gears	10	25	9	3	20	**67**	
Conveyor	6	25	3	3	20	**57**	
Ramp + Door	6	15	3	5	20	**49**	
Wheel	10	25	15	3	20	***73***	

We also evaluated each design's merits and drawbacks to make sure we hadn't missed anything.

Table 3: Design Recap

Design Recap	**Reasoning**
Hooks	Ruled out for feasibility (obviously fragile, hooking the boxes was a complex problem in itself
Gears	Limited capacity and requires lots of space. *(In retrospect could have been made to work, a la needle dispenser)*
Conveyor	Research showed this was very hard to implement well; placing RFID readers unwieldy.
Ramp + Door	Very likely to jam with no mechanism for relieving the jam; not sure how to quickly raise and lower a door/release
Wheel	Footprint and capacity unrelated; less likely to jam, with recourse to unjam; simple, buildable mechanism.

We chose the wheel, both because it won in our KT Decision Matrix and because it appeared to be the best return on scalability and time needed to make it work. We also had concerns about the gears because none of us were very familiar with manufacturing something that required both that precision and scale at the same time.

Section 3: Implement and Iterate

Now it was finally time to make our ideas come to life. We broke the project down into a few specific components and ensured each one worked before bringing the whole project together. Our components were the boxes, the wheel, the frame, and the code and electronics.

Boxes

First, we built the boxes because all the other components were reliant on specific dimensions of the boxes and how the boxes interacted with each other. To start, we figured out how we wanted the RFID card to fit in the box. We settled with putting it in the base. Next, Leon created a CAD model in Solidworks of the idea box we wanted. Then, we decided to laser cut the boxes because it was both cheap and time-efficient, so Frank translated the Solidworks model into AutoCAD to create something as close as possible that was also able to be manufactured. In FYELIC, we laser cut the design and taped the box together to check fit and function. Finally, to add fit and finish and based on feedback about our prototype, we decided to paint the boxes red and glue them together. We decided to build the boxes ourselves because we wanted to scale down the project, and needed something with a flat top so that they could slide on top of each other easily. This was hard to find because most boxes you purchase are designed to stack, so we made them ourselves since we've made enough laser-cut boxes in our time, that four more didn't hurt.

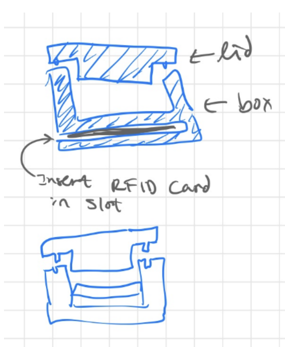


Figure 15: Prototype

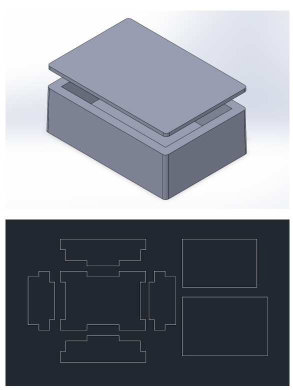


Figure 16: CAD

Figure 17: Laser cut

Figure 18: Painted (Final Result)

Wheel

The wheel was the core component of our project. First, we wanted to understand if a motor could be used to drive it, so we performed calculations to understand how much torque would be required and learned it was more than any of our Sparkfun motors could produce. The specific math we did to find this can be found in the Appendix. Knowing this, we needed to make the wheel such that it was large enough to move boxes and have the user push it. Michael created a CAD model in Solidworks to get us started. Then we 3D printed a version and realized it was too small. Michael created another iteration which was larger and 3D printed that one. We 3D printed because we wanted a complex and precise material that was not easily purchased.

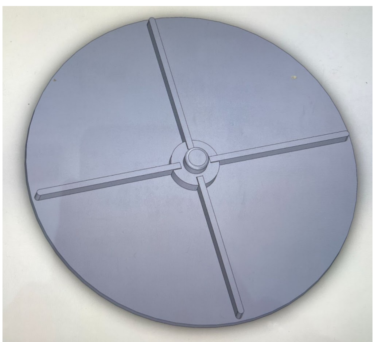


Figure 19: Initial CAD idea

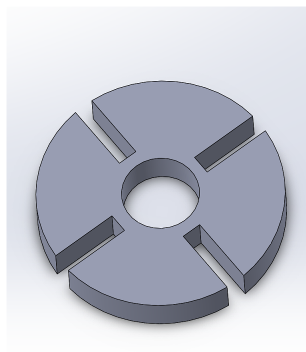


Figure 20: Solidworks model to be 3D printed

Figure 21: 3D Print of the result

Next, we put it all together and demonstrated circular motion and tangential motion are not one and the same (the boxes moved tangentially because the wheel only provided a one-directional force). To solve this, we added protrusions to the wheel to guide the box. This kept the boxes aligned as the wheel rotated.

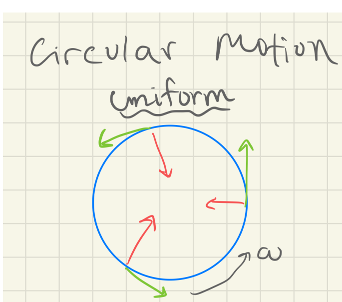


Figure 22: Frank's Physics notes on Circular Motion

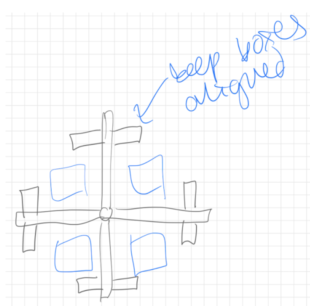


Figure 23: Idea for our solution

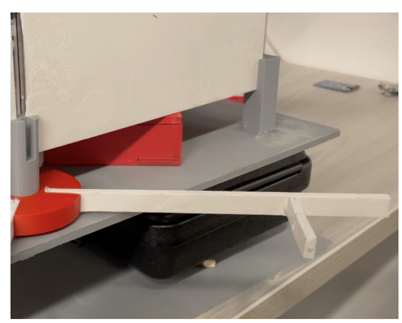


Figure 24: Implementation

Frame

The frame encapsulated both the tower to hold the boxes and the box which covered the project and made it look presentable. The main restriction with the frame was that it had to be off the ground so that the boxes and wheel could go under it. We thought about this problem for a while, and eventually decided to use stilts on the edges and the center of the wheel to hold up the rest of the structure. We used 3D printing to make the stilts because they were complex small parts that were not able to be purchased. We used laser cutting and the bandsaw to make the frame because we wanted precise parts and we needed to iterate quickly. Wood was also less expensive than we realized and let us build large, sturdy structures quickly.

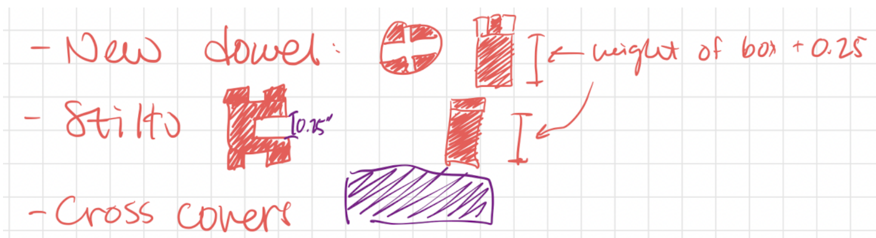


Figure 25: Initial sketches

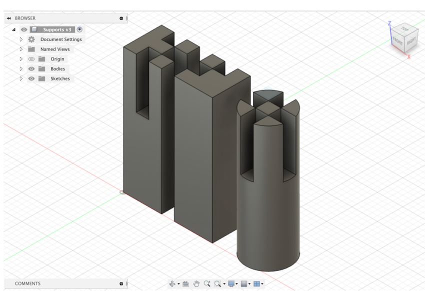


Figure 26: CAD Model

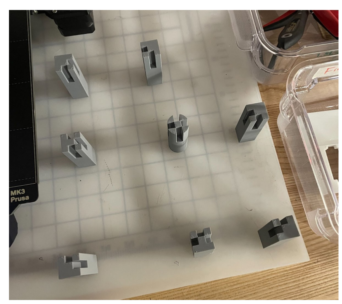


Figure 27: Final Prints

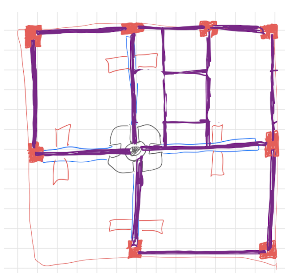


Figure 28: Initial Sketch

Figure 29: Paper model

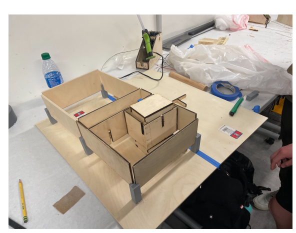


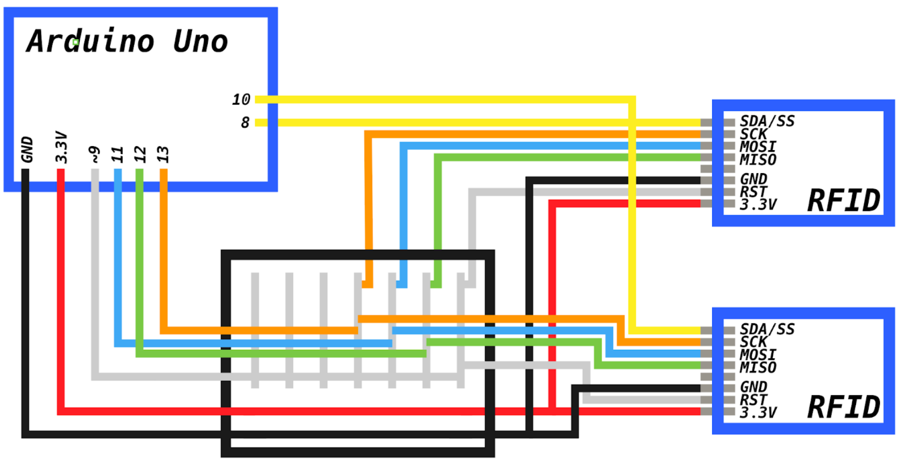
Figure 30: Build

Figure 31: Painting

Wiring and Code

To scan the boxes, we needed to wire up the RFID readers. Wiring up one was easy, but getting two to work on the limited number of inputs the Arduino provides was a challenge. We made extensive use of the examples provided by Annaane on their [Github page] about reading four RFID readers for a project. They provided a wiring diagram and code samples. We also took lots of inspiration from Miguel Balboa on their [Github page] where they provide numerous examples for using a single RFID.

We wired up the two readers using this wiring diagram (Figure 32), adapted from Annaane's example:



With the Arduino and RFID readers wired, we were able to start coding. Working from the flowchart designed in the previous section of the project, we were quickly able to complete the logic sections of the code—reading serial inputs and scanning the RFID cards as well as printing useful messages to the Serial interface. What repeatedly created difficulty however was the reliability of the RFID readers. The readers were incredibly sensitive to power changes, so we went through many of them before realizing we had to be very careful to unplug the board from power before making changes to wiring. Below is very simple pseudo code (Figure 33), but a full code breakdown can be found in the appendix.

```
If there's input waiting:
    Read it: Either NUID or "."
        New box path (NUID):
        Validate NUID
        Read Box ID
        Conjoin NUID and Box ID in 2D Array
    Return path ("."):
        Read Box ID
        Find NUID for Box ID
        Mark box as returned
    Wait a bit, reset
```

Table 4: Bill of Materials

Item	Cost	Retailer	Purchaser	Use
Total	**$99.95**	**-**	**-**	**-**
Remaining	**$0.05**	**-**	**-**	**-**
Number Pad	$13.27	Microcenter	Frank	Blocking user from doing non-numeric inputs
1/4" Laser Cutter Wood (4 sheets)	$21.17	NU Bookstore	Frank	Boxes and dividers for spinning part
1/4" Laser Cutter Wood (2 sheets)	$10.58	NU Bookstore	Frank	More wood for frames
12" x 24" Craft Sheets (3 sheets)	$19.10	True Value	Frank	Wood for base
Spray Paint (White, Gray, Red)	$23.94	Blick Art Materials	Alex	For decoration
Gloves	$6.90	Target	Alex	For working with spray paint
Extra wires	$4.99	Microcenter	Frank	They were pretty and Frank so wanted them but fronted this cost himself instead of making his group members fund his indulgences.

Our final Bill of Materials came in just under budget, which demonstrates how we maximized within our restrictions and were able to take advantage of what we could. We did also make use of the Sparkfun kit, Frank's iPad, and a lot of computer cables to connect those devices to the checkout system too. Finally, we made use of many RFID readers from FYELIC (because they kept breaking) and those are not factored into our spending budget.

Section 4: Evaluate

It's safe to say, our final robot achieved our primary goals because it possesses the functionality we desired to theoretically satisfy our project's objectives. The robot can check boxes in and out and correctly mark who checked them out and which box they checked out. The system is easy to use, and if built to the proper scale and with modifications to turn it from a proof-of-concept into a full-size product, could be placed in a Northeastern dining hall for students to use.

Not only does the system perform within the obvious use case, but it also accounts for edge cases such as NUID inputs of invalid length and trying to return a box that was never checked out. The wheel is not too fragile, and the system rarely jams—when it does, it's a very simple fix.

Compared to our original ideas, the system changed in size and scope a lot, but the core check-out system stayed mostly the same after the decision process. Choosing to narrow the scope to focus on the check-out was a good decision in retrospect because it allowed us to ensure we fulfilled the core objectives to the best of our ability. If we hadn't done this, our monetary and time budget would have been stretched far too thin and the project would not have been able to fulfill its original objectives.

To showcase our project, we made an infomercial to showcase the problem it solves and its functionality. The video can be viewed [here](#).

Figure 34: Top View

Figure 35: Front View

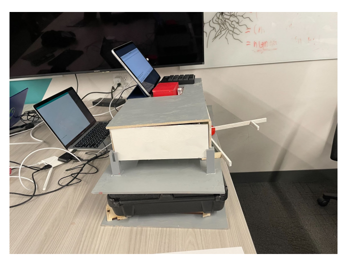


Figure 36: Side View

Section 5: Reflection

Ethical Implications to decisions

The ethical decision behind the to-go containers was to create an environmentally friendly method of food storage and to limit the excessive use of plastic in our dining halls, along with the goal of limiting waste of food within our dining halls. The to-go containers ethically serve as betterment for society, which was the main reason behind the development of the To-Go System. There are a variety of viewpoints and stances to take from an ethical perspective, generally, we believe our device provides a net good—our arguments and possible counter arguments are provided in the table below:

Table 5: Ethical Arguments and Counter Arguments

Utilitarian Pro	Utilitarian Con
More one-time-use plastics will be saved than waste created from lost boxes.	More strain will be placed on dining halls, and boxes will still be lost whereas to eat in dining halls before reusable plates had to be used.

Deontological Pro	Deontological Con
Giving students more options and flexibility is good. Reducing use of one-time-use plastics is good.	Letting students take food and the possible waste that comes with that is bad. Creates the opportunity for excess waste where there once was none.

Safety

The design of our to-go system does the opposite of harm in society, allowing for an environmentally friendly method of food storage, and limiting the waste of food within dining halls in itself.

While the system in itself requires the user to spin the wheel upon retrieving a to-go container, such a device poses no substantial harm to the user. The only implication being the system having sharp corners on the design layout may cause the user to cut themselves however to do so would require deliberate misuse of the device.

The material involved in the design of the to-go system is wood, which could pose a threat for cuts and splinters upon turning the wheel which is an area of concern. At the same time though, the design presented is only a prototype of the device we plan on creating, as such it is important to note that wood doesn't have to be the material in use and softer material such as plastic can be incorporated into the design of the true system in the future to prevent injury.

Service to Humanity

Simply put: our robot does provide a service, admittedly not particularly life-changing, to humanity. It provides a method of food storage that promotes reuse instead of single-use—a positive impact on the environment, which in turn has a positive impact on humanity.

To-go containers provide a valuable service to cleaning the environment from excessive overuse of single use plastic and food waste, providing a better source for people to store their food for the long term through the use of biodegradable containers.

The system also makes life more convenient and flexible for students. Having to eat inside a dining hall is the smallest of hardships; technically this is an additional humanitarian benefit our device provides.

Engineering Codes

No codes were violated in the development of our device. According to the NSPE Code of Ethics for Engineers, "Engineers shall hold paramount the safety, health, and welfare of the public." Safety was achieved with our to-go containers by providing humanity a safe way to store their food while having convenient access to the containers using an easy-to-turn wheel system. Health was achieved by designing our system to make use of existing NU Dining safety infrastructure, to ensure our system adds no more risk than current options. Welfare is simply doing well for society as a whole and our to-go containers provided an environmentally friendly method of storing food while providing access to all students, the ability to obtain a to-go container for their own needs.

In the Code of Ethics for Engineers, the aspect of Professional Obligations was achieved as our device does not attempt to harm anyone or anything and our best wishes are honest and truthful as the whole motive behind the development of the to-go containers is to better society, not to profit ourselves.

Lastly, no deceptive acts were initiated in the process of creation. In the development of the to-go containers, the goal is honest and narrowly focused and that is through the aspect of bettering society and the environment as a whole. Nothing is misled in the promotion of our project, and everything is centered around bettering society.

Reflection and Learning Process

To create the most efficient design, we were tasked with running through several iterations of designs in an effort to single out the best approach to handling our problem. The most enlightening part of this process was connecting concepts from other classes, mostly Physics, to help explain observations and questions we had about how our device functioned or could function.

In our first iteration of the design process, we decided upon two concept applications that are no longer part of our final design: servo motors and ramps. When looking at these approaches through the lens of physics, we realized that the amount of torque that would need to be provided for these functions to suffice was far beyond the capabilities of the equipment provided to us. We also realized that with a ramp our project was likely to jam based on how the boxes would tend to fall. What we learned from this scenario is that projects often need to be scaled down, and do not need to be necessarily as complicated as they seem logistical. We easily maneuvered around this complication by using gravity and vertical alignment as opposed to a conveyor belt, and user intervention as opposed to autonomous spinning using servo motors. We arrived at the same outcome as intended with our new implementations, which showed us that more is not always better and that the most pragmatic solution is often the simplest and best.

Changes and Future Improvements

Throughout our design process, we all shared brilliant ideas, but these ideas needed to be pieced together in a way which would allow us to build a formidable device. Because the mechanics of our system were very difficult to draw in 2D, it was hard to quickly draw ideas and communicate them. On numerous occasions we would have to go as far as creating detailed CAD models of an idea, just to communicate it effectively enough for the others to understand and provide feedback, wherein we would decide to take the project in a different direction. As we got further down the road in the project, pivoting required disposing of more and more sunk cost, or instead changing the goals of the project to refocus our efforts and resource use.

In retrospect, using something like the Roxbury Robotics Lego kits would have been a simple and effective way to quickly model ideas because drawing them on paper was so difficult. Another solution would have been to learn a little more Solidworks so that we could use it to create functional prototypes with CAD. Working as a group is not always the easiest, but if done correctly it can yield great results. This is a concept that we found out more as we continued to work as a group, which provided more efficiency to our work as we reached the final stages of our project.

If we had more time to complete our project as well as a greater budget, we would improve upon the mechanical design of our project through the use of better materials. Firstly, we would use a larger motor with enough torque to spin our wheel with a maximum of four boxes included. We would also look into the concept of having this wheel be vertical instead of horizontal to decrease the footprint, with the capability of holding more than four boxes per rotation. A greater budget would allow us to build a full sized model, letting us do real user testing of the device to gain better, more accurate feedback.

Currently, we implemented a system in which students can type in their NUID and receive a box based on their personal information. We realize that this is not the safest method since anyone can type in another person's NUID to be granted access to a box. With more time, we could potentially allow users to scan out a box using their Husky Card or the CBORD mobile app. This would allow for greater security of our boxes, which is something that we did not focus on greatly throughout the design process.

Scale and security are two concepts which we would choose to focus on greatly if given more time to work, and a much greater budget. These would allow our design to have the potential of real implementation in Northeastern University dining halls.

References

[1] B. Kuswandi, "Environmental friendly food nano-packaging | SpringerLink," Environmental Chemistry Letters, vol. 15, pp. 205-221, Feb. 2017.

[2] Packaging Concepts Inc, 2022. Microwavable Food Container. US2021114793A1.

[3] Dagmara Bajer and Aleksandra Burkowska-But, Innovative and environmentally safe composites based on starch modified with dialdehyde starch, caffeine, or ascorbic acid for applications in the food packaging industry." doi: 10.1016/j.foodchem.2021.131639.

[4] “ECFR :: 21 CFR Chapter I Subchapter B -- Food for human ...” [Online]. Available: https://www.ecfr.gov/current/title-21/chapter-I/subchapter-B. [Accessed: 16-Feb-2022].

[5] “Food Delivery Safety,” Centers for Disease Control and Prevention, 15-Jul-2021. [Online]. Available: https://www.cdc.gov/foodsafety/communication/food-safety-meal-kits.html#restaurant. [Accessed: 16-Feb-2022].

[6] “ECFR :: 21 CFR Chapter I Subchapter B -- Food for human ...” [Online]. Available: h https://www.ecfr.gov/current/title-21/chapter-I/subchapter-B. [Accessed: 16-Feb-2022].

[7] “Dine On Campus at Northeastern University || Boost Mobile Ordering."

[8] “Lunch to Go | Princeton Campus Dining."

[9] S. Long, “Sustainable Food and eco-friendly food packaging,” *Green Business Bureau*, 20-Jul-2021. [Online]. Available: https://greenbusinessbureau.com/business-function/facilities/food-service/sustainable-food-and-eco-friendly-food-packaging/. [Accessed: 16-Feb-2022].

[10] “Reusable Container Program | Sustainability at Harvard.”

[11] Jessica Schmitt, Eco-to-Go, a Reusable Container Program, has officially launched at Coral Tree Cafe! | UCSB Sustainability." sustainability.ucsb.edu, https://sustainability.ucsb.edu/news/eco-go-reusable-container-program-has-officially-launched-coral-tree-cafe.

[12] Alexa Davis, Eliminating College Dining Hall Trays Cuts Water, Food Waste - ABC News." abcnews.go.com, Oct. 2022, https://abcnews.go.com/OnCampus/story?id=6087767&page=1.

[13] A. Foundation, "Reusable containers aren’t always better for the environment than disposable ones - new research," Theconversation, Sep. 7, 2021. https://theconversation.com/reusable-containers-arent-always-better-for-the-environment-Than-disposable-ones-new-research-166772.

[14] "Starbucks To Offer Reusable Cup-Share Program In All Europe, Middle East and Africa Stores By 2025 - Starbucks Stories EMEA," Stories, Jun. 3, 2021. https://stories.starbucks.com/emea/stories/2021/emea-cup-share-program-2025/.

[15] Starbucks Coffee. "New Test: Borrow a Cup, Save a Cup," Youtube, Jun. 4, 2021. https://www.youtube.com/watch?v=3Pn4UTaXNGw&ab_channel=StarbucksCoffee.

[16] “Library Self Service Checkout makes borrowing & returning simple,” *bibliotheca*, 18-Mar-2021. [Online]. Available: https://www.bibliotheca.com/solutions/self-service-checkout/.
[17] “Self-service checkout,” *EnvisionWare*, 26-Feb-2020. [Online]. Available: https://www.envisionware.com/self-service-checkout/.
[18] M. P. Pandian and M. P. Pandian, “RFID for libraries,” in RFID for libraries: A practical guide, Oxford, England: Chandos Publishing, 2010, pp. 43–73.
[19] Justin Brasher. *Library Self Checkout.* Youtube 2017. URL: https://www.youtube.com/watch?v=xt9JGaAkxaQ&ab_channel=PasadenaChannel
[20] Northeastern University. "Locker and Vending Services," Northeastern University Information Technology Services, https://its.northeastern.edu/hardware/.

Appendices

Physics Work

This is the math we did to calculate how much torque would be required from a motor to spin the wheel, as compared to how much one of the Sparkfun motors can produce.

- Torque Needed to Begin Turning of Blocks

$\tau = F_F d$

$\tau = .4(4)(.119\,Kg)(9.8\,m/s^2)(.173\,M)$

$\boxed{\tau = .323\,Nm}$ Required Torque Needed to begin turning

$F_F = \mu mg$

$F_F = (.4)(.119)(9.8)$

$M = .119\,Kg$

$6.8\,in = .173\,M$

- First Push with Servo!

Knowns: Mass, Rev/Min

$\tau = I\alpha$

$\tau = \frac{5}{6} mr^2 \alpha$

$\tau = \frac{5}{6}(.549)(.173)^2 (17.102)$

$\boxed{\tau = .234\,NM}$

$\boxed{\tau_{Needed} > \tau_{servo\,Max}}$

- Calculating Angular speed Max

70 rev/min

$\frac{70\,rev}{min} \times \frac{min}{60\,sec} \times \frac{2\pi}{rev}$

$W = 7.33\,rad/s$

- Calculating Total Mass

$M = .073 + (.119)4$

$M = .549$

- Calculating Inertia of Block System with Servo

$I = \frac{4}{12} MR^2 + \frac{1}{2} MR^2$

$I = \frac{5}{6} MR^2$

- Calculating Angular Accel.

$W_f^2 = 2\alpha\Delta\theta$

$\frac{W_f^2}{2\Delta\theta} = \alpha$

$\frac{(7.33)^2}{2(\frac{\pi}{2})} = \alpha$ $\quad \alpha = 17.102$

Gantt Chart

<table>
<tr><th>Activity \ Week</th><th>2/18</th><th>2/25</th><th>3/4</th><th>3/11</th><th>3/18</th><th>3/25</th><th>4/1</th><th>4/8</th><th>4/15</th><th>4/22</th></tr>
<tr><td>Background Research</td><td colspan="2">Background Research</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Project Proposal</td><td></td><td colspan="2">Project Proposal</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Imagine</td><td></td><td colspan="2">Imagine</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Decision Matrix</td><td></td><td colspan="2">Decision Matrix</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Design Outer Structure</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">Design Outer Structure</td><td></td><td></td></tr>
<tr><td>Build Outer Structure</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">Build Outer Structure</td><td></td></tr>
<tr><td>Design Electronics</td><td></td><td></td><td></td><td colspan="3">Design Electronics</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Initial Code</td><td></td><td></td><td></td><td></td><td colspan="2">Initial Code</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Edit Code / Electronics</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">Edit Code / Electronics</td><td></td></tr>
<tr><td>Design Boxes</td><td></td><td></td><td></td><td colspan="2">Design Boxes</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Build Boxes</td><td></td><td></td><td></td><td></td><td></td><td colspan="2">Build Boxes</td><td></td><td></td><td></td></tr>
<tr><td>Design Wheel</td><td></td><td></td><td></td><td></td><td colspan="2">Design Wheel</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Build Wheel</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">Build Wheel</td><td></td><td></td></tr>
<tr><td>Testing and Improvements</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">Testing and Improvements</td></tr>
</table>

Code with Documentation Inline

```
/**
 * Team 1 - Meal Box Checkout System
 * Frank Anderson
 */

/**
 * Based on code found here for a 4 RFID system: https://github.com/Annaane/MultiRfid
 */

#include <SPI.h> // serial interface library for RFID
#include <MFRC522.h> // RFID library

/**
 * PIN & READER ASSIGNMENTS
 * Reader 0 = check out box
 * Reader 1 = return box
 */

// PIN Numbers : RESET + SDAs
#define RST_PIN         9
#define SS_0_PIN        10
#define SS_1_PIN        8

byte ssPins[] = {SS_0_PIN, SS_1_PIN};

// Create an MFRC522 instance for each reader:
#define NUM_READERS   2
MFRC522 mfrc522[NUM_READERS];

// String to go down RETURN path
#define RETURN_KEY "."

/**
 * Global vars to track program state
 */
String tagID = "";  // id of currently read tag, "" if no tag read yet
String nuid = "";  // "" if waiting for input; "r" if in return process; "#########" if in checkout
process
String boxes[10][2]; // store boxes and nuid's (technically can only hold 10 entries, but would use SQL
database if actually used in real world)
int numBoxes = 0; // track how many boxes have been added (for efficiency)

/**
 * Set up initial state of the program
 * - Open Serial and RFID interface
 * - Setup each reader
 * - Flush the Serial monitor and print a welcome message
 */
void setup() {
    Serial.begin(9600);           // Initialize serial communications with the PC
    while (!Serial);              // Wait until port is open
    SPI.begin();                  // Init SPI bus

    setupReader(0);               // Setup readers
    setupReader(1);
```

```
    flushSerial(-1);              // Clear the Serial monitor of previous run content
    printInstructions();          // Print a welcome message
}

/**
 * Run state of the program
 * Checks if information is in the Serial flow and then passes down either flow path.
 */
void loop() {
    if (Serial.available() > 0) { // is information in the Serial path...
        while (nuid == "" && nuid != "r") { // if not currently on a path... figure out which one to go
down
            if (Serial.available() > 0) { // if there's something to read, read it
                readSerial();
            }
        }

        if (nuid == "r") { // return path
            Serial.println("Place your box on the white circle. Wait for the box to scan.");
            while (!getID(1)); // repeat until reading something for checkout

            if (findNUIDfor(tagID) != "-1") {
              Serial.println("Box ID: " + tagID);
              Serial.println("Marked box connected to NUID: " + findNUIDfor(tagID)) + " as returned.";
              Serial.println("Place box back in stack.");
              removeNUIDfor(tagID);
            } else {
              Serial.println("Box not currently checked out.");
              Serial.println("Try returning again if you believe this is an error.");
            }

        } else { // check out path
            Serial.println("NUID: " + nuid);
            while (!getID(0)); // repeat until reading something for checkout
            Serial.println("Box ID: " + tagID);
            addBox(tagID, nuid);
            Serial.println("Rotate the wheel 90° clockwise and and take your box!");
            Serial.println("Remember to return it within 7 days, or you will be charged.");
        }

        delay(3000); // 1 second delay, then clear instance variables...
        nuid = "";
        tagID = "";

        flushSerial(-1); // reset
        printInstructions(); // print instructions for a new user
    }
}

/**
 * Print instructions for user.
 */
void printInstructions() {
  Serial.println("\nType your NUID and press Enter to check out a box.");
  Serial.print("Type ");
  Serial.print(RETURN_KEY);
  Serial.println(" and press Enter to return a box.");
}
```

```
/**
 * Read input from Serial stream and clean input / throw errors
 */
void readSerial() {
    String incomingString = ""; // string to be read
    String cleanedString = ""; // string to be output

    incomingString = Serial.readString(); // read the serial port

    // if return path indicator ...
    if ((incomingString.substring(0, 1) == RETURN_KEY || incomingString.substring(0, 1) == "R") &&
        incomingString.length() < 4) {
        nuid = "r"; // set the current path
        return; // exit the function
    }

    // otherwise ... clean the string for NUID input
    for (int i = 0; i < incomingString.length(); i++) { // only take numbers as valid chars
        if (incomingString.substring(i, i + 1) == "0" ||
            incomingString.substring(i, i + 1) == "1" ||
            incomingString.substring(i, i + 1) == "2" ||
            incomingString.substring(i, i + 1) == "3" ||
            incomingString.substring(i, i + 1) == "4" ||
            incomingString.substring(i, i + 1) == "5" ||
            incomingString.substring(i, i + 1) == "6" ||
            incomingString.substring(i, i + 1) == "7" ||
            incomingString.substring(i, i + 1) == "8" ||
            incomingString.substring(i, i + 1) == "9") { // if valid...
            cleanedString += incomingString.substring(i, i + 1); // add to cleaned string
        }
    }

    if (cleanedString.length() == String("000000000").length()) { // if it's the right length too
        nuid = cleanedString; // set the NUID
    } else { // otherwise, reset and tell them to try again
        cleanedString = ""; // clear instance variables
        incomingString = "";
        Serial.println("Invalid input. Try again.");
        return; // exit  function (this return isn't necessary, but it's good practice in case we add
another state below)
    }
}

/**
 * Set up the reader to be readable.
 * Used to take complex line and make it more readable, as well as usable for different readers.
 * @param reader - the reader to setup
 */
void setupReader(int reader) {
    mfrc522[reader].PCD_Init(ssPins[reader], RST_PIN); // init function for mfrc522
}

/**
 * Make a reading if possible, #t if read, #f if nothing to read/could not read
 * @param reader - reader to read from (same process for return/checkout so...yeah)
 * @return - true if read, false if nothing to read/could not read
 */
boolean getID(int reader) {
    if (!mfrc522[reader].PICC_IsNewCardPresent()) { // is there a card there to read?
```

```
        return false;
    }

    if (!mfrc522[reader].PICC_ReadCardSerial()) { // can it be read?
        return false;
    }

    tagID = ""; // clear the tag id before it gets reassigned

    for (uint8_t i = 0; i < 4; i++) { // read the tag id
        tagID.concat(String(mfrc522[reader].uid.uidByte[i], HEX));
    }

    tagID.toUpperCase(); // make the tag id uppercase

    mfrc522[reader].PICC_HaltA(); // stop reading

    return true; // Yay! It read something
}

/**
 * Print a bunch of blank lines to the Serial monitor so that the next user doesn't see the previous
user's information
 * @param numLines - number of lines to print, if negative prints an amount that totally flushes the
monitor.
 */
void flushSerial(int numLines) {
    int linesToPrint = numLines; // how many lines should be printed
    if (numLines < 0) { // if it's a negative number, use a big value
        linesToPrint = 50;
    }

    for (int i = 0; i < linesToPrint; i++) { // print x-many blank lines
        Serial.println("");
    }
}

/**
 * Reassign boxes array with new size for new boxes
 * @param box - id of the box
 * @param nuid - id of user with box
 */
void addBox(String box, String nuid) {
  boxes[numBoxes][0] = nuid; // store nuid
  boxes[numBoxes][1] = box; // and boxid
  numBoxes += 1; // add one to the box count
}

/**
 * Find the box ID in an array of box id's
 * @param boxID - id of box to find
 * @return nuid connected to box, or "-1" if box not found
 */
String findNUIDfor(String box) {
  for (int i = 0; i < numBoxes; i++) { // linear search the array
    if (boxes[i][1] == box) {
      return boxes[i][0]; // return box id if found
    }
  }
```

```
  return "-1"; // if not found, -1 is the serial value
}

/**
 * Remove the box ID in an array of box id's by setting the id to "0"
 * @param boxID - id of box to find
 * @return nuid connected to box, or "-1" if box not found
 */
String removeNUIDfor(String box) {
  for (int i = 0; i < numBoxes; i++) {
    if (boxes[i][1] == box) {
      boxes[i][1] = "0"; // "removing is just setting an impossible value"
      return "1"; // 1 == success code
    }
  }
  return "-1"; // -1 == value code
}

/**
 * Prints all boxes in Boxes
 */
void dumpBoxes() {
  for (int i = 0; i < numBoxes; i++) { // for all the boxes in the array...
    Serial.println(boxes[i][0] + "   " + boxes[i][1]); // print nuid and box id
  }
}
```