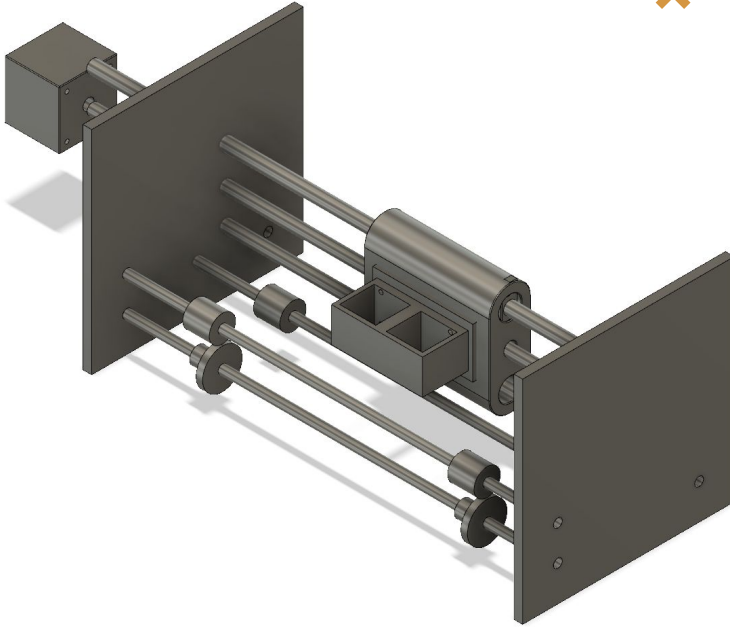


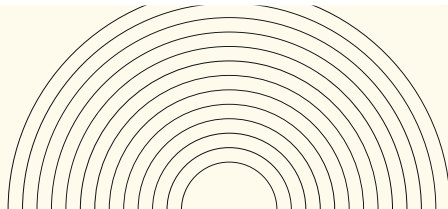


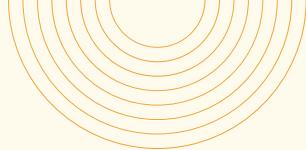
Becky Button  
Joshna Iyengar  
Zeynep Ozkaya



# A4: Digibraille

*A braille printer directly accessible  
from your phone!*





# Problem Statement



*Current technologies for blind people do not allow for easily accessible printed braille for simple everyday use.*



Braille is the preferred format for blind users



Braille writing slates have character limits



Computers not easily accessible



Electronic braille readers and printers are expensive



Electronic braille readers and printers are not portable



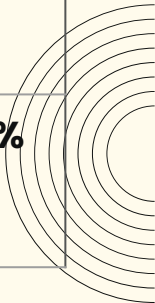

Can't connect to an embosser from a phone





# Quantitative Design Requirements

User Requirement	Design Requirement
Speed	<ul style="list-style-type: none"><li>• Embossing an <b>8.5" x 11"</b> sheet of braille (768 char) should take <b>less than 16 mins</b> to print out</li><li>• <b>1 second</b> latency</li></ul>
Accessibility	<ul style="list-style-type: none"><li>• User should learn to use device in <b>&lt;10 min</b></li></ul>
Accuracy	<ul style="list-style-type: none"><li>• Embossed braille should be <b>100% readable</b> by printing to spec'd out tolerances</li><li>• <b>95% accurate</b> when translating to braille.</li></ul>
Completeness	<ul style="list-style-type: none"><li>• Web-scraping algorithm must produce a result <b>95%</b> of the time.</li><li>• Web-scraping should produce results in <b>&lt;.5 sec</b></li></ul>

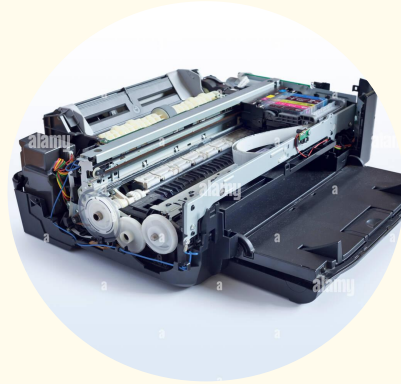




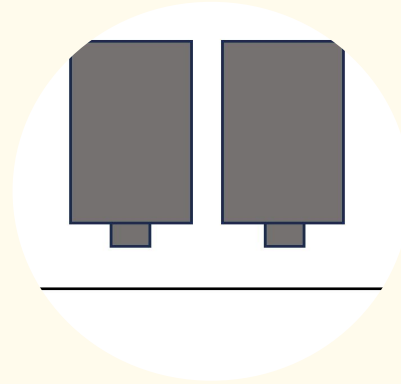
# Solution Approach



Web Interface  
accessible  
from a phone



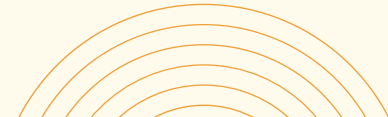
x-gantry system  
with rollers to move  
paper



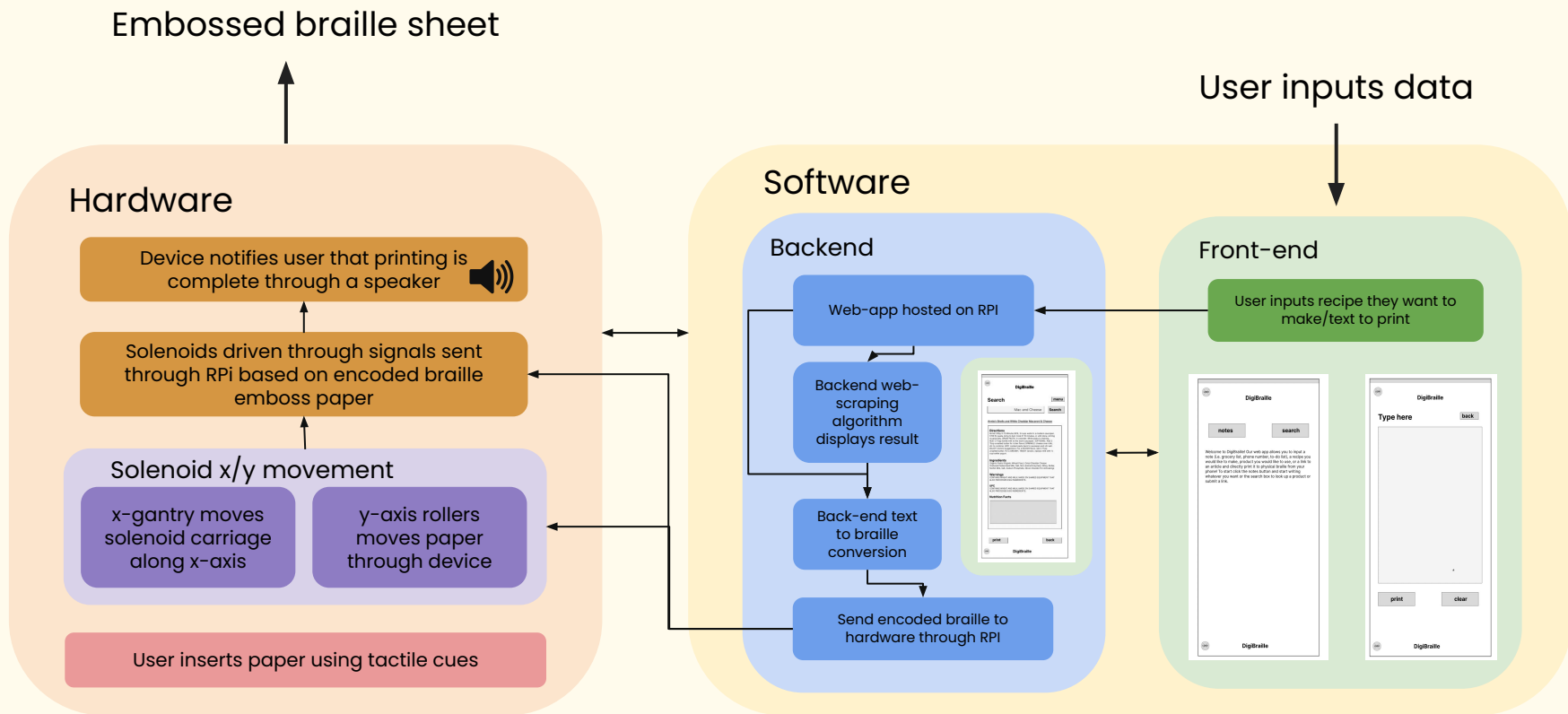
Braille cells will  
be embossed  
using solenoids



An embossing system  
designed with  
accessibility and long  
term use in mind.



# System Block Diagram



# Software Implementation

1

Web-app using Flask hosted on RPi takes user input

2

Web-scraping to find entered product recipe using BeautifulSoup and 10 sites that have product data bases

3

Text is parsed and processed to follow contracted braille format and grammatical rules

3

Parsed contracted braille input is interpreted as a combination of positions on stepper motors and solenoid activations

4

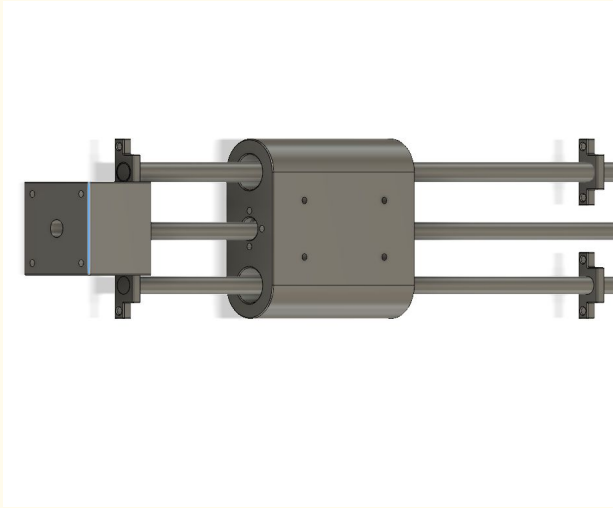
Output signal sent to hardware using RPi



Flask

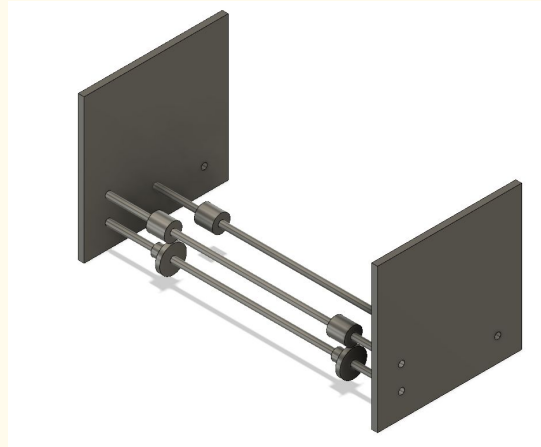
BeautifulSoup

# Mechanical Implementation



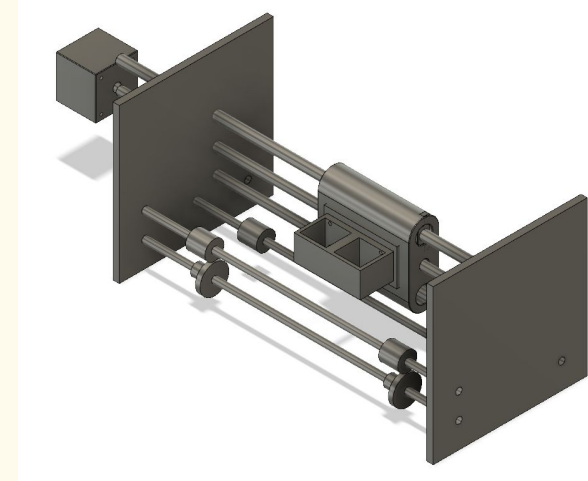
## X-Gantry

Lead screw driven  
Modular solenoid  
attachment to make  
system easily iterable



## Y-Axis

Printer style Y movement



## Overall

Decoupled X/Y  
movement  
Easy paper registration  
points

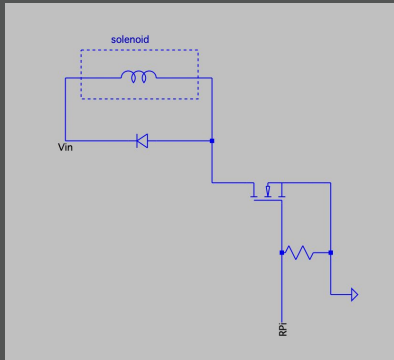


# Electrical Implementation

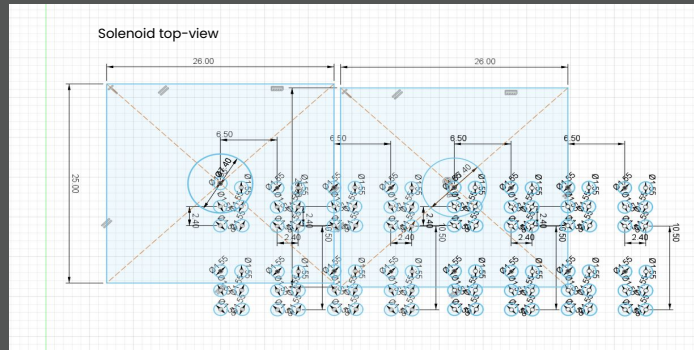


- 2 solenoid embossing system based on timing requirements and braille specs
- Solenoids will emboss to tolerances defined by spec'd out tolerances
- Stepper motors will receive signals to move x/y system based on spec'd out measurements
- Solenoids will be activated by binary voltage signal

## Solenoid Circuit



## Braille Cell Specs



## 2 Solenoid Math

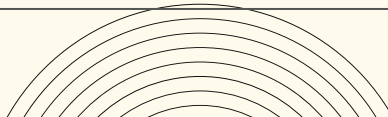
# of Solenoids	Time/sheet
1	29 minutes
2	11 minutes
3	11 minutes



# Implementation Plan



	Materials	Design
Hardware	<ul style="list-style-type: none"><li>● 12V Solenoids</li><li>● Flyback diode</li><li>● Resistors</li><li>● Mosfet</li><li>● Stepper motor</li><li>● Lead screw</li><li>● 12V power supply</li></ul>	<ul style="list-style-type: none"><li>● PCB</li><li>● RPi</li></ul>
Software	<ul style="list-style-type: none"><li>● Python</li><li>● BeautifulSoup (web-scraping from select websites such as <a href="http://directionsforme.org">directionsforme.org</a>)</li><li>● Flask (front-end)</li></ul>	<ul style="list-style-type: none"><li>● Web-application</li></ul>



# Testing, Verification, Validation

Requirement	Testing Plan	Mitigation
Speed	<ul style="list-style-type: none"><li>• Time worst case scenario printing times (full sheet of braille)</li><li>• Time how long it takes to connect to printer, search for directions, print a sheet of braille, etc.</li></ul>	<ul style="list-style-type: none"><li>• Add 2 more solenoids as a separate system to increase number of cells embossed/sec</li><li>• Ensure proper sourcing of x/y gantry parts</li></ul>
Accessibility	<ul style="list-style-type: none"><li>• User test the product and website using accessibility features available on phone</li><li>• Debug using Apple's Accessibility Debugger</li></ul>	<ul style="list-style-type: none"><li>• Read accessibility documentation and conduct extensive user testing</li></ul>
Accuracy	<ul style="list-style-type: none"><li>• Test output of text to braille conversion on 100+ scripts of text with output from Duxbury Braille Translation software</li><li>• User testing for readability of printed braille</li></ul>	<ul style="list-style-type: none"><li>• Ensure proper sourcing of x/y gantry parts</li><li>• Modularize design such that embosser system can be redesigned as needed</li></ul>
Completeness	<ul style="list-style-type: none"><li>• Test web-scraping algorithm on 100+ different products</li></ul>	<ul style="list-style-type: none"><li>• Generate database based on webscraping</li></ul>



# Project Management

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Background research	All	All	All											
Build basic framework for website			Joshna	Joshna	Zeynep	Zeynep								
Design solenoid system				Zeynep	Zeynep	Zeynep								
Design x/y gantry				Becky	Becky	Becky								
Develop algorithm for braille translation						Joshna	Joshna	Joshna						
Test solenoid system						Zeynep								
Design and fabricate PCBs and enclosure						Becky	Becky	Becky						
Develop frontend							Zeynep	Zeynep	Zeynep	Zeynep				
Integrate hardware and software components									All	All	All	All	All	
Slack			Zeynep	Joshna	Becky	Becky	All	Zeynep	Joshna	Zeynep				





**Digibraille: A smart web connected braille embosser system that can easily print grocery lists and product information right from the user's phone!**

