

## **Problem Statement**

Today, a large portion of written content is created and stored electronically. However, there remains a substantial amount of handwritten documents that aren't easily accessible or searchable in digital formats. The need to convert these handwritten materials into a more convenient, accessible, and persistent digital form is becoming increasingly pressing. As people become more accustomed to reading and interacting with text on digital devices, the ability to decipher and interpret handwritten content is diminishing. This shows the necessity for a reliable method to make handwritten documents accessible online, ensuring that important information is not lost or overlooked because of the challenges of reading handwritten text.

## **Vision Statement**

FOR educators WHO need a method to reliably transcribe handwritten text for improved accessibility, Scribble Scan (name in progress) is an accurate handwriting digitization AI, to improve legibility and accessibility of writings. UNLIKE Pen to Print, MY PRODUCT has increased digitization accuracy, especially with more difficult to read handwritings including cursive or children's writing.

## **Stakeholders**

### Specific Use Cases:

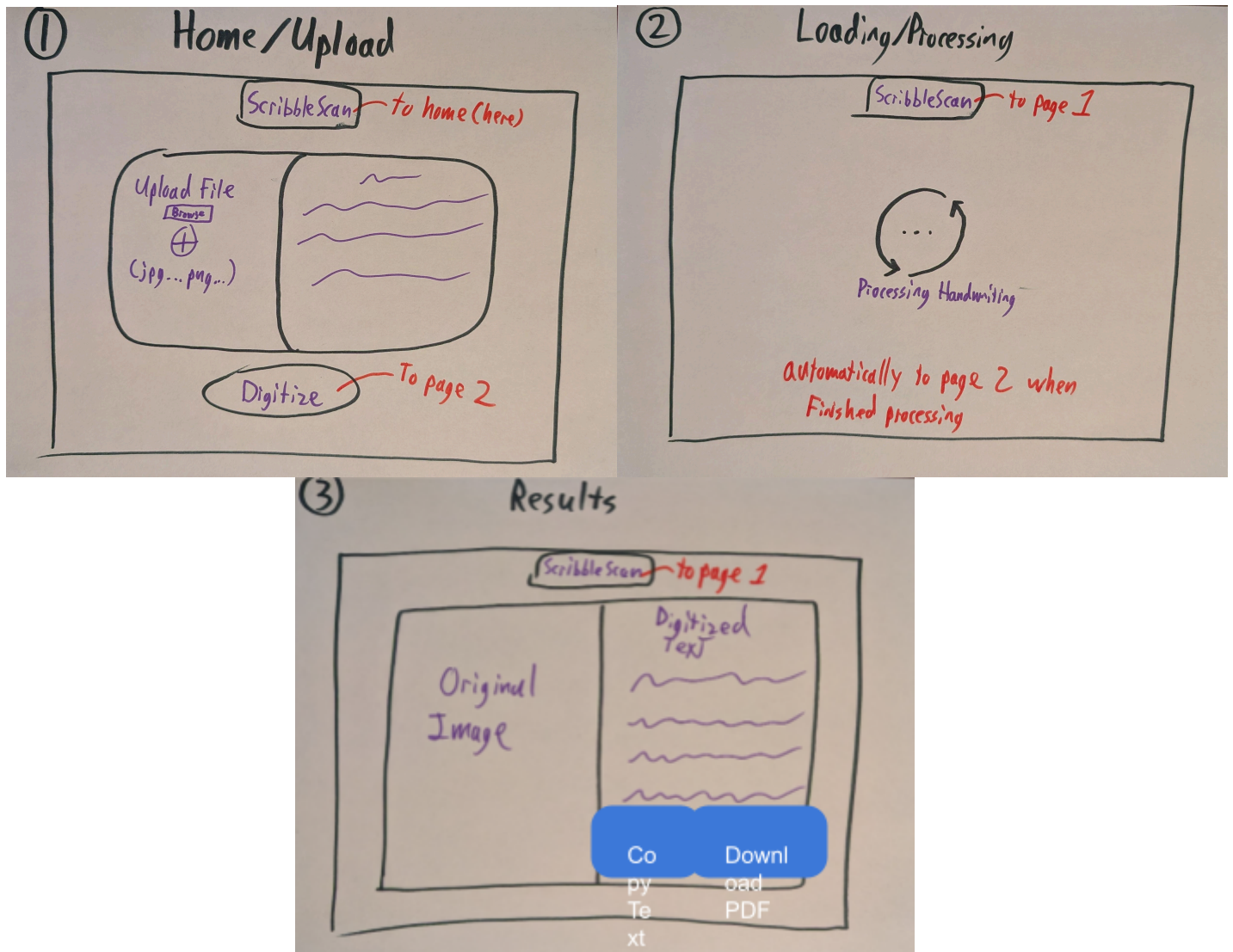
- Professors looking to translate old field notes
  - Taking old field notes and digitizing them for easier storage and reference.
  - Generally older audience (with old written notes that need to be translated.  
Pre-personal computer work)
- Like \_\_\_\_\_ and their field notes
- Not super comfortable with technology. But knows enough to want to translate writing into an online format. Even if they have no idea how to do it

- Academic. Well educated. Professional
- Will likely use once with a lot of pages
- Students looking to translate lecture notes
  - Taking recent lecture notes and digitizing to keep work in one place.
  - Generally young. People who have grown up using technology
  - Like me or my classmates who take notes on paper. An easy way to digitize these notes could prove helpful later, both in storage, and future reference.
  - People who know how to use basic technology comfortably, including tools like google drive and smartphone cameras
  - Academic and tech comfortable
  - Will likely use more than once for smaller amounts of writing

## Functionality

- Web-app to make accessible
- Take user file input
  - PNG, JPG, WEBP(?), PDF
- [Redacted]
- [Redacted]
- Downloadable result (PDF, Copy/paste)
- Text to speech result option

## Wireframes:



## **Timeline:**

Week 1(1/20-1/26):

- Set up server and docker/containerization
- Start Web-development with fast-API python-based framework

Week 2(1/27-2/2):

- Continue web-development
- Figure out drag and drop files

Week 3(2/3-2/9):

- Continue web-development
- Convert proof-of-concept code to runnable code in API

Week 4(2/10-2/16):

- Finish full implementation of barebones handwriting digitization web-app
- This includes user file upload, and basic digitization process with copy-pasteable result

Week 5(2/17-2/23):

- Brief testing of current work
- R&D for methods of gathering word locations in image

Week 6(2/24-3/2):

- [Redacted]

Week 7(3/3-3/9):

- [Redacted]

Week 8(3/10-3/16):

- [Redacted]

## **SPRING BREAK**

Week 9(3/24-3/30):

- [Redacted]

- Brief testing of current work

Week 10(3/31-4/6):

- Add autocorrect as final step in process
- Add text to speech on result page

Week 11(4/7-4/13):

- Add support for downloadable types such as docx... pdf... md... copy/paste...

Week 12(4/14-4/20):

- Expand input types: pdf? Other image types?

Week 13(4/21-4/27):

- Finishing touches on website and UI
- Testing

Week 14(4/28-5/4):

- Final testing and tweaks

## **Resources:**

Programming Languages:

- Python (backend)
- JS, HTML, CSS (frontend)
- SQL (database)

Frameworks:

- Fast-API
  - For building python-based web-app API

IDE:

- Visual Studio Code

Server:

- Probably Google Cloud (I have experience with this)

- Potentially AWS (I want to learn this)
- Depending on how it works, and if it could run with my docker containerized project, I would maybe be open to using Tech Services Server

Database:

- I will use PostgreSQL. As far as I understand it will work with the setup of my application, and will work with Google cloud or AWS

## **Evaluation**

The most straightforward method to evaluate my work is accuracy. I will demonstrate that I have solved the problem, by creating text digitization software with higher accuracy than the competitors when given a variety of handwritings. I think 2 weeks of full testing will be more than enough given how my project is split into parts, and will be easy to test as I go. I am not too sure, but I think I might need to do 25-50 iterations of testing to make sure my product is working as desired. I will be looking to get feedback through both in-person interviews, and through google surveys for people who I cannot meet.