Product Vision Document:

Product Vision

Problem Core:

Transcription management is complex and there are a lot of moving parts. Plus, with the fact that law as an industry is in the dark ages, for good reason, for the same reasons that airlines still run critical apps on COBOL, this means there's tons of room for improvement in transcript management. This software solves that huge problem.

Additionally, frequently, transcribers are single entities and outsourcing entities like to take a huge cut of the page rate when they contract with people to do transcribing. Having software like this would mean the transcriber is able to cut out the middleman MUCH easier, when it might have been impossible for many before, and automate nearly all major benefits an outsourcing company would provide to an individual transcriber. So for the individual transcriber, this means, for example, going from a page rate of $1.35 with an outsourcing company to $2.50 per page or even more, up to $5 or $6 per page. It's a huge amount of money and a huge problem which this software solves.

Last but certainly not least, having a speech recognition engine along with this software would further help the individual transcriber by making it more profitable and more of a proofing job than typing.

Problem Side Effects:

Slower turnaround times w/o automation and/or w/o speech recognition

TONS of details involved in transcript management that need to be PERFECT because it is sometimes literally a life-or-death matter for the defendant or at least loss of liberty in some fashion.

Most transcribers aren't capable of making a complex piece of software DIY style; they're great at Office and related products, can make their way around a computer easily enough, but don't have the technical skills needed to create something like this themselves

Streamlines every part of transcript production and makes every edge case fit into a streamlined framework (production in stages one through four)

Objectives:

Customers

Target customers:

Release Canvases

Features in Production

Defects in Production

Solution Details

Future features to research

Release Canvasses

Analytics

Architecture

Technical: frameworks/platforms/other architectural decisions

Api integrations (broken into front end/back end)

Styling considerations:

User Research

Competitor Research

1. Is someone else already working on this problem? If so, how are they doing? (can be another company, or all the ways an individual is solving their problem)
2. If others have failed in solving this problem, why did they fail?
3. What will it take for users to change from their current solution to our solution?
4. What will your application do that is different from competitors? Why is your product an improvement over your competitors?
5. Summary
6. What other ways is the target persona solving the problem?
7. What are you going to do differently than your competition?
8. Why should users use your product vs. an alternative?
9. Competitors List

Technical Research

## Back-End

* 1. What are you using for your database? (SQLite3, PostgreSQL, MongoDB, etc…)
  2. What framework/language are you using? (many teams use NodeJS/express/knex)
  3. How do you plan on interacting with your APIs? (REST API, GraphQL, etc...)
  4. What alternatives did you explore?
  5. Why did you decide on this combination? (what are the advantages?)
  6. What potential challenges can you foresee? (what questions do you have about this implementation to which you *don’t* have answers?)

## Front-End

(if your product has more than 1 platform, have a section like this for each platform)

* 1. What Framework are you using for your FE? (React, React Native, Angular, Vue, etc...)
  2. What alternatives were considered?
  3. Why did you decide on this framework over the alternatives (what are the advantages?)
  4. What potential challenges can you foresee using this framework?

## Styling Considerations

(if your product has more than 1 platform, duplicate this for each platform)

* 1. What styling library are you using?
  2. What alternatives did you consider?
  3. Has the team UX designer weighed in? Are they bought in to this implementation?
  4. Why did you decide on this library?
  5. What potential challenges can you foresee?

## API Integrations

(Create a section that answers these types of questions for each new API integration)

* 1. What Feature(s) is/are this API connected to?
  2. What will this API integration accomplish in the feature(s) to which it’s connected?
  3. What alternatives did you consider?
  4. Why did you decide on this solution? (what are the advantages?)
  5. What are the potential challenges can you foresee?

User Personas

General Bio Info, Scenario, Goals/Interests, Pain Points/Concerns

Design Inspiration

Wireframes

Mockups

Prototype

Design System

Usability Testing