Jack Scavone

Project Biography

[Link to Final Project](https://afterpattern.com/wb/jacknscavone/scavone-staffing-solutions)

**Early Stages**

The early stages of this project involved a lot of trial and error in terms of finding an idea to work with. I had planned for several weeks to make a program for a firm that deals with nuclear energy, however, that plan fell through after weeks of planning. The plan for that project was to create a program that, based on legislation regarding nuclear energy assignment, would determine whether or not a specific area would be eligible to receive energy from nuclear power plants. But as mentioned, this plan fell through leaving me back at square one. While I had several other potential projects planned, I decided that I would work with my brother to help him with his staffing business.

**The Problem**

My partner runs a staffing business in which he seeks out both employers looking for potential employees to hire as well as job candidates looking to be placed into an employment position. When meeting with my partner, I asked him to lay out some potential problems that he may run into during the course of business that are of a legal nature. He responded that his only legal issues have to do with his contractual obligations following the termination or resignation of clients that he provided to employers. When an employee is terminated, my partner is under no contractual obligation to help the employer find a new candidate, though he may do so to protect the business relationship he has with the employer. When a candidate resigns within a certain time period, my brother is contractually obligated to find a replacement candidate for that position within a timely manner.

**Rubric Checklist**

**Intro Pitch**

For my intro pitch I outlined the problem (as stated above). I identified the potential users for this project (my partner for this project). I identified the current options available to me which were Afterpattern and Docassemble as well as database options such as google sheets. And I described how I planned to solve the problem through using a document automation program to provide my partner with a streamlined way of creating correspondence documents and storing those documents and their relevant data securely in a database.

**Research**

Having identified the problem at hand I needed to conduct research into how to solve this problem, or if this was even a problem in need of a solution. I decided that the best way that I could address this problem would be to provide my partner with a way to automate and securely store these correspondences.

Given that my brother’s staffing business is not a law firm, there was the question of whether or not my proposed project would be an example of unauthorized practice of law, however, after research and correspondence with a professor and an attorney, I was able to conclude that my project was not in any way an example of unauthorized practice of law. While the project does reference contractual obligations, it is not providing my partner any positive or advisory action regarding contract law. It is instead automating and storing regular correspondences that are not legal documents. While these are not legal documents, having these correspondences archived could potentially be useful to my partner in a potential civil suit as evidence, but again, they are not legal documents.

**Ideation**

Having identified a problem and confirming that it is not something I am unauthorized to do, the next step was to decide what tools to use to solve the problem. My first thought was to use either Afterpattern or Docassemble (the only programs of this nature that I am aware of) alongside something like a cloud storage program such as google drive or onedrive. I met with Professor Colarusso after class one day to discuss ways in which I could attack this problem, and we came to an agreement that either Afterpattern or Docassemble would be the best approach. Considering that I lack any substantial background in coding and with using these systems, I settled on Afterpattern because it was a program I was now familiar with using. On top of this, based on a conversation with Professor Colarusso, I was also made aware that Afterpattern has an integrated database feature, which solved my other problem of how to store these documents. Before settling on Afterpattern, my other options were to use something such as google sheets or a comparable microsoft program to store and catalogue these documents. My issue with this however was that I was not fully convinced of the security of using these programs to store important documents. With Afterpattern being a legal tech company, I had more faith in the security of their database given the nature of the documents they typically deal with. Therefore, I settled on using Afterpattern for the document automation, as well as the database for storing the documents.

**Prototyping**

Having decided on what tools to use for the project, the next step was to start using them. The first thing I did was create a skeleton using the information I received from prior meetings with my partner. While this was in no way a finished product, it allowed me to get a lot of the hard work out of the way (or so I thought) and then make minor refinements in the future based on feedback from my partner. Though my prototype was functional, it was more convoluted than it needed to be, and there were multiple instances where I was using way too many blocks to get a desired outcome. I could tell from my own testing that I was using too many question blocks and prompting too many user typed responses. I could tell that this would lead to fatigue on the user end, and likely defeat the purpose of automating a document like this in the first place.

**User Testing**

Despite the concerns about my prototype, I still had my partner test the initial prototype as is, and he shared the exact concerns as I did in terms of how convoluted the initial product was. On top of this, he addressed several issues with the project in regards to the structuring of the questions. My partner also requested that he draft many of the blocks of text so that it would read more closely to something he would write. Based on the initial feedback on the prototype, I made the appropriate refinements (see: **Refinements**). With the refinements made, I had my partner test the program several more times and requested feedback, my partner’s final requests were to include an option for if he wished to still help a company find a candidate even if he was under no contractual obligation to do so (this would be done to encourage companies to continue working with his business) and to allow him to include additional user-written comments towards the end of the document in case there was a unique case where more information would be needed. With my near final product I had my partner test it several more times, and I also allowed several friends and family members to test it as well. While the program was initially designed to be used solely by my partner, by allowing others to test it would give me an idea of how intuitive the program is to someone who was not already involved in the creation of it, and the response was largely positive in terms of the intuitiveness, however there were still some concerns that needed to be addressed, such as the document using my partners name by default.

**Refinements**

Based on the feedback I was given by my partner and by other testers, I was able to better refine my project and cater it more effectively to my partner’s needs. The documentation of feedback will be attached to this document, and of the feedback given I was able to implement nearly all of it except for some requests that were either not possible to do given the tools I was using, or after consulting with my partner, were not ideal for how he intended to use the program. Some of the major refinements I made were:

(1) Allowing an option for my partner to communicate that he would like to find a replacement candidate for a business despite being under no contractual obligation to do so. To do this I had to rework much of my project, as it was initially left up to a yes or no question as to whether he would provide assistance, and instead I had to use a radial question format and then define 3 variables for potential text blocks, and with those I had to make a conditional logic block that would output the correct text block based on my partners answers in the question prompt.

(2) Cutting back on the amount of questions asked to avoid user fatigue/defeating the purpose of automating a document. To accomplish this I had to start nearly from square one and also I had to learn how to better implement logic blocks and conditionals into my project. With the help of Professor Colarusso and students in the Coding the Law course, I was able to find a great catalogue of videos with in depth tutorials on how to use Afterpattern logic blocks, and with this new info I was able to restructure nearly my entire project and make it more efficient in terms of the amount of questions asked. This involved using logic blocks that would reuse certain variables from previous questions, and making certain questions appear conditionally rather than having them always appear regardless of whether they would be used in the final document.

**Complexity/Robustness**

This project makes use of a document automation system as well as a system for storing and cataloguing these documents. Both the document automation and the data storage are achieved using Afterpattern.

**Impact and Efficiencies**

Knowing that efficiency was an important part of this project’s evaluation in the rubric, I asked my partner to estimate how long it takes him to write out a letter to an employer, and he told me that writing anything of this nature takes him far longer than it should. A one page email/letter similar to that of which the project creates could take him anywhere from 10 to 20 minutes. He claims that this is because he spends too much time ensuring that what he is writing is appropriate and comes off as professionally as possible. He also claims that he second guesses his word choice when addressing representatives from the companies he works with and that having an automated document like this that is properly worded each time will ultimately save him hours of time he would have otherwise wasted drafting these correspondences from scratch. With the final product of this project, the process of creating a document will take no longer 3 minutes as opposed to the 10-20 minutes he claims writing these documents would otherwise take.

**Fit/Completeness**

In regards to fit and completeness, my partner has already stated that this is something he wishes had had access to because his work has ramped up in the past few months and he has already run into situations where clients have been fired or resigned. One of the biggest advantages that my partner has suggested will result from this project (one that I did not even consider when creating it) is that he will now have uniformity among these correspondences. One of the disadvantages of writing an email from scratch each time, is that the correspondences take a more conversational approach given that the entirety of the document is catered to that specific employer. My partner gave me an example of a time where he sent a short email to an employer regarding a client being let go from a job, and the employer, in a casual/conversational manner asked my brother if he could “do him a solid” in replacing the candidate. Had my partner used this program to automate the document, he feels that employers would understand that their relationship is strictly a business relationship, and that they are not in a conversational correspondence with my partner but rather in a more professional and by the numbers business relationship. He also feels that if an employer receives an automated document rather than a document catered to them specifically, it will come off more as if they were being addressed by a detached authority rather than by my partner himself, and he hopes that this dynamic will lead to more strict compliance with the terms of their agreements.

**Documentation**

All of the necessary documentation is included within the solution itself. From the start the user is addressed with clear and concise questions/directions for properly filling out the questionnaire. On top of this, Afterpattern does a great job of making the process of filling out the questionnaire and receiving the finished document as easy as possible, leaving very little room for confusion on the user end. That said, I have still provided my partner with the proper steps for using the program on his own computer and will be available for assistance at any time considering that my partner for this project is also a family member.

**Real World Viability**

I was initially unable to get the app in my partner's possession as Afterpattern has suspended new users from making accounts. Afterpattern did this to avoid users of Netdocuments signing up for Afterpattern expecting it to already be fully integrated with Netdocuments. However, after speaking with a representative from Afterpattern I was able to make a new account for my partner. My partner now has full access to the app and can use it for his job. He has tested it

**Sustainability**

The project as is will be sustainable for use indefinitely provided that my partner’s contracts remain the same and provided he does not hire additional employees who may need to use the program. However, in the likely scenario where one of these things do change, I have already mapped out how I will address these problems.

1. In the case of a change in the terms of the contracts, I will adapt the questions and outcomes of the program to match the changes. This will involve me taking a similar approach to when I started working on this project in that I will have to map out the new contract in a way that can be translated into an afterpattern document/questionnaire.
2. In the event that my partner hires new employees who will be in contact with employers, I will remove the static text from the document template that caters to my partner, make a logic block with nearly identical text but using information specific to the new employee, and I will also need to add an initial prompt to determine who is creating the document, and from there the questions and text will be catered to that specific employee.

**User Feedback**

The following is taken from feedback I received during in person testing with my partner for this project, as well as feedback from two family members and a friend who has a background in computer science. The feedback was originally recorded in a notes app on my cell phone and has been categorized in chronological order and is further categorized based on who provided the feedback. I did not include any positive feedback in this document as I did not make any record of positive feedback.

**From Partner:**

First Testing ( 10/22/2021)

* Cut down on the amount of questions if possible
* Add an option to allow for finding candidates despite not being obligated to
* Allow partner to provide names and emails of potential candidates (almost always gives 2)

Second Testing (11/09/2021)

* Allow partner to write the text himself to match his writing style
* Add an option for writing additional comments to the employer if necessary
* Include e-signature instead of text signature
* Refer to employers as “businesses”
* Include a formal ending paragraph (to be written by partner)

Final testing (11/24/2021)

* Ask for candidates in later prompt (was previously in first prompt)
* Remove confirmation of termination
* Add a spot for resumes

Database Feedback From Partner (12/03/2021

* Add a column for the date the letter was written
* Add the new candidates emails to the database

**From Additional Testers (two family members and one friend)**

(All of the following feedback was taken on the same day (11/24/2021

* Allow for other names to be used than just partners name
* Make the entire questionnaire fit to a single page or less pages (not possible to my knowledge)

**User Letter:**

