**PySoftExcel App Pitch**

Xavier Caldwell

College of Western Idaho

SWDV – 152 – Systems Analysis and Design

Professor Ricky L Del Vecchio

April 8, 2022

**PySoftExcel App Pitch**

Being certified in the Microsoft Suite, I am fairly intimate with the nuances of each different app, especially that of Excel and Access. My family and many of my colleagues use Excel especially because of its initial ease of use, and the fact that spreadsheets are great at managing projects or personnel. However I have seen firsthand how long and time-consuming committing changes can be on larger spreadsheets, or for those who aren’t extremely proficient in using the built-in tools on these apps, the amount of experimenting or clicking around just to find the correct function to organize their data. Today I am proposing a new way to append or organize spreadsheets without even needing to open the document. Using an app that I have dubbed PySoftExcel, spreadsheet management will be made nearly seamless. This app, while seemingly simple, is able to save the user an amount of time that is proportional to how long they will be using Excel, or how much information they need to either remove/add/sort. Optimally, this, according to the system request pulled by myself could save up to 45 minutes worth of time for those who work on spreadsheets all day, by removing a lot of the “finding and clicking” of certain data. The finished app will have a user-friendly interface that can provide a list of functions along with explanations of what they do and be able to perform them on specific data sets, such as rows, columns, or the names of specific people and whatever information pertains to them.

Generally, the production requirements for this project is small. The same goes for the production window. To be exact, its about 2-4 weeks, the two weeks for a working prototype, and the four for a bit of polish to be applied to it. However, because of the scope of the app, it doesn’t necessarily need to be huge, and looking at the feasibility study, the only considerations that seem to really take precedence before I start production are the ones stating that proficiency in Python is paramount. The rest of the feasibility study almost reads itself as a total green-light for this app, considering that the resources needed to start and finish production are minimal, and easily accessible at any time. All that I will need to start production is some free time and access to my own computer and some LinkedIn Learning videos, which I already pay for.

In regards to the security of this project, this whole deal should be nearly trivial. According to the risk management plan attached to this document, the only real this project should be a low-risk venture, considering that there is only myself working on this app, with very little threats to the development of it, and it wont cost anything. The only issues I may come across when working on this app is simply poor time management, but because of the fact that I need to pass this class to actually get a job in the long run, I feel that decent time-management will easily find its own place in the list of priorities.

A very worthwhile question to ask is how to measure success and when to know when the app is done. The simple answer is that, once it can at least access and edit spreadsheets, which is its intended function in a nutshell, it basically is done. From there, I can add more functions from what is already present, and include user feedback for how to make the UI prettier. The easiest and probably most reliable way to get feedback for the UI should be just asking outside sources such as my friends and even maybe my instructor. I think personally, a completed app will have a comprehensive list of options that can always be expanded based on feedback from users, so determining a definite “complete” is a little harder than normal, but as long as it has primitive functionality, wherein “primitive” can be defined by users, then I think we are good to go.

Overall, I hope that PySoftExcel sounds like a promising and useful little utility app that can improve the quality of life for people working around the office, maybe even some of those working at CWI. With the low opportunity cost and plenty of room for it to grow once the foundation has been set, alongside its nearly airtight security and almost no capital needed to begin work, I feel like I have made a comprehensive case as to why I should be able to begin production on PySoftExcel.

**References**

Allen, J. (2015, October 8). *The importance of client-side JavaScript security*. SitePoint. Retrieved February 11, 2022, from https://www.sitepoint.com/importance-client-side-javascript-security/

Chen, B. (2019, January 22). *How programming languages can hurt your application's security*. TechBeacon. Retrieved February 11, 2022, from https://techbeacon.com/security/how-programming-languages-can-hurt-your-applications-security

Wayner, P. (2013, February 4). *Safeguard your code: 17 security tips for developers*. InfoWorld. Retrieved February 11, 2022, from https://www.infoworld.com/article/2078701/safeguard-your-code--17-security-tips-for-developers.html