**Technical Feasibility Draft**

**October 23, 2022**

**Team Teacher To-Do**

**Project Sponsor:** Chris Aungst

**Faculty Mentor:** Michael Leverington

**Team Members:**

Sam Gerstner (Team Lead)

Alexander Frenette

Noah Nannen

Shlok Sheth

Bronwyn Wedig

**Overview**

The purpose of this technical feasibility document is to outline the high-level project requirements, technologies to be used, as well as the technical feasibility and any foreseeable problems with any project requirement or chosen product or technology. This document aims to provide an in-depth analysis of all MVP requirements and the technologies and platforms that will be used to help us accomplish our goals.

This document will be used as a reference point for the remainder of the project and will be one of the primary documents that guides our development and decision-making processes. One of the primary goals of this document is to outline potential problems, their solutions, and to help all stakeholders gain a better understanding of road blocks that may be encountered during the development process.

**Scope of Project**

**MVP Tasks & Goals**

**Plan for Development**

**Languages and Frameworks**

One of the first challenges we faced with this project when coming up with our development plan, was what programming language we should use. Ultimately, we decided to use Java for this project because of it’s wide use in enterprise applications, extensive documentation, long-term support, familiarity among our team members, and the wide variety of tools available to extend Java’s functionality. Java is an extremely popular choice for enterprise web applications because it is platform independent, has built-in memory management, is very cost-effective, and is easily scalable. By using Java to develop this application, we are also conforming to some industry best-practices by using a widely supported language that is unlikely to go away any time soon.

In order to streamline the development process, we chose to use the Java Spring Boot Framework which provides a wide array of tools for web application development, templating, security, and more. Spring Boot incorporates a style of web application development known as Model-View-Controller or MVC which we will be utilizing for this application. MVC allows for the separation of web and code components, allowing for easy division of development tasks among our team members based on individual’s strengths. By being able to divide tasks based on front-end/back-end development allows us to assign development tasks to the team member whose skills best match the specific task. In most web applications, the front-end and back-end development are very closely coupled, and don’t allow for the easy division of tasks based on team member strengths.

The Spring Boot framework will allow us to write our application in languages that are familiar to us (Java, HTML, CSS, etc.) and easily package our application into a JAR or WAR file that can easily be deployed to an Apache Tomcat web server. By using the Spring Boot tools, we are also ensuring coherency among components of our application, and greatly increasing the likelihood of all application components working together correctly to hopefully minimize the amount of work needed to implement basic components of the web application.

**APIs**

**Authentication**

**User Interface & User Experience**

For our user interface, we are planning on using Spring Boot, a Java addon that allows us to generate HTML5-based webpages from our backend and use them to import any data that the user needs to import into their profile. Spring Boot will act as a sort of observer, seeing what webpage would best suit the needs of the importer and generating the page from a series of templates that our UI developer will create and implement into the Spring Boot language.

**Potential Problems**

For our project, most of the potential problems stem from us being unable to gain access to the student database for resources. If we are unable to use real student data, we will have to try and spoof our data set and test from there, which may lead to inaccuracies in the program’s data storage system. We would also not be able to successfully implement the system, meaning that we would create the project to the fullest extent that we are able, and then it would most likely sit on the sidelines for a few years until ITS gets the chance to modify it to work within the current system. We have done all we can to minimize this risk, even going as far as to become FERPA certified to be able to responsibly handle student data, however it is ultimately a choice left up to ITS on whether they see it as to great a risk on their part. We do have an alternative option to directly importing in student data, which is to have students enter in via text and Boolean checks into the system, and then have an administrator verify the data, however that comes dangerously close to the manual-input system the College of Education already has in place, and may not win over the administrators by having them learn an entirely new system that does essentially the same thing as the system that is already in place.

**Conclusion**