# CS 305 Module Two Written Assignment Template

## Areas of Security

The application that we are looking at uses a Spring-based framework to help weed out any possible vulnerabilities. When it comes to security is crucial to make sure any outside data that could corrupt the application is handled appropriately. Looking at this type of application it seems the Input Validation would be the top part of VAPFD that needs to be handled. Other areas of security that have to be taken into consideration for this application are the API Interactions, Code Errors, and Encapsulation.

## Areas of Security Justification

A screenshot of a computer

Description automatically generated

Any input that enters the application must be sanitized to make sure that it is safe before being used by the application. Data in input form can potentially do harm whether on purpose or not and must always be treated like a threat and must be validated. API is used as a communication tool to speak from the application to the user and or server. Therefore, this communication must be secured to make sure all data being shared is trustworthy. Encapsulation will be used to help from data being changed that could cause unwanted errors.

## Code Review Summary

Currently in the code, we can see that it is planning on parsing through a string before outputting the string. The parsing uses a getValue() method that passes the string to an API that then validates that the input is not corrupt or harmful. We also notice there is a greeting class and a greeting controller. The greeting class is rather simple. It has two get methods within it. One to return an ID and one to return content.

The greeting controller is a little more complex. This class isn’t used to create an object, but rather has more functionality. The greeting controller has two greeting objects. One greeting object is labeled as greeting and parses through a string and returns the string that has been sanitized. The other is a greeting object named number. This object sends in an id to a function that returns and prints a value from an array based on the id that is was given. The application also has a pom.xml file that it utilizes. This file allows the application to use a Spring framework that works as a tool for security.

## Mitigation Plan

In order to mitigate any issues that may occur we should work around what security protocols should be used. The first being any and input must be validated and sanitized. Encapsulation can be enhanced in the greeting class. This is possible by utilizing the get methods within the greeting class. Using the API to help fix any code errors is also a must. I will help with a spillage of data that can be misused in a malicious way or possibly by accident. The last thing to consider in order to mitigate any problems is making sure everything is up to date on the Spring framework end. While utilizing the updated framework all dependency vulnerabilities should be taken care of.

**Citations**

*How to read the reports*. dependency-check – How To Read The Reports. (n.d.). https://jeremylong.github.io/DependencyCheck/general/thereport.html

Manico, J., & Detlefsen, A. (n.d.). *Iron-clad Java*. O’Reilly Online Learning. https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/?sso\_link=yes&sso\_link\_from=SNHU