# CS 305 Module Two Written Assignment Template

## Areas of Security

This project requires me to build a “complex web application that uses Spring Framework. The team has been tasked with implementing an expressive command input function for the application. The team is currently using Version 2.6.5 of the spring-data-rest-webmvc in Spring Framework, also want to use the Spring Expression Language to accomplish the task.”

Based on this tasking, I have reviewed the first layer of Vulnerability Assessment Process Flow Diagram and concluded there are three areas of security that are relevant to this scenario: input validation, APIs, and code quality.

## Areas of Security Justification

Input validation: since this website will be implementing an expressive command input function. The application will need to strictly validate all input to make sure that the new input is not conflicting with the current function in the system, change the system in anyway, or returning invalid output. This way we can prevent an injection attack and make program more secure.

APIs: API allow website to make data public so that others can use the application or analyze the data and help make the website more secure. The application using Spring Network can implement Restful API’s dependencies to allow access to the system features, but we need to make sure the access is monitored, and input is validated.

Code quality: it is important to ensure that the project is running smoothly, easy to debug, free from error that can be exploit such as a DDOS attack, accurately displaying return for input.

## 3. Code Review Summary

* Outdated version of Spring-data-rest-webmvc: In pom.xml, the version listed is out of date “2.6.5. RELEASE”. The current available version is 4.0.3. The latest version should be used to ensure the application using the most up to date dependency that may include more security feature than the older version.
* The “GreetingController.java” insecure input: in line 27-30 of the code, the parser data is put into a string directly to the message. The parser input can be format and since it not checked, it will affect the GreetingController.java.

## 4. Mitigation Plan

* Update the current Spring-data-rest-webmvc version to the most up to date.
* Secure code practice called for implementing data validation, validate data range, and data length prior to using the input. Therefore, to use the data from the parser as a string into our template, we need to first validate the data again and possibly limit the size of the string which prevent overload of data from passing in (could cause a DDOS attack).