# CS 305 Module Two Written Assignment Template

## Instructions

Replace the bracketed text with the relevant information in your own words. If you choose to include images or supporting materials, make certain to insert them in all the relevant locations in the document.

## Areas of Security and Justification

* **Input Validation –** Input validation is a must, defining what constitutes good data and reject everything else. Leveraging patterns such as RegEx can help strengthen input validation.
* **APIs –** The Spring Framework relies on APIs to function so must our app. To make sure the APIs and the data sent and received remain secure, measures such as secure JSON and content security policies must be instituted.
* **Cryptography –** Any public facing web application, the client is counting on us to keep any sensitive information secure so we must ensure all data is encrypted before transmitted by using TLS or SSL.
* **Client/Server –** Separating how data is processed is a simple way of creating some security as the client should never be allowed to process certain information. With any client/server setup security patterns must be practiced preventing attacks or manipulation of data.
* **Code Error –** Error handling must be done so carefully, standardizing error messages and return the same error message despite the type of failure is one way of securing password validation. Going hand in hand with input validation code error is a must.
* **Code Quality –** Adhering to coding standards, best practices and proper conventions will increase code quality. The Spring Expression Language (SpEL) will help with this in our app.
* **Encapsulation –** We must ensure our application is properly encapsulated within the entire system and this will allow us to follow best practices which mitigates the risk for attacks.

## Code Review Summary

* 1. In opening the project there is a warning the version 2.6.5 of the spring-data-rest-webmvc is out of date. Using outdated frameworks opens our application up to vulnerabilities that were fixed in later versions of the framework. The outdated version could also lead to incompatibility issues as well as compilation issues.
  2. There is a lack of input functionality is another issue, while the parser, “Hello World” message is coded correctly, no user input functionality exists
  3. Spring Boot is also outdated and must like the spring-data-rest-webmvc, using outdated versions of a framework opens our application up to vulnerabilities.
  4. The constructor for the Greeting.java file is public which is a security risk because opens the door to attacks.

## Mitigation Plan

* 1. The largest concern is the outdated version of the framework, so updating the framework would be one of the first actions to take. The latest version of the spring-data-rest-webmvc is 5.3.22 and Spring Boot is 2.7.3. This will resolve incompatibility issues and protect our application from the latest vulnerabilities.
  2. The next area of concern is the Greeting.java file, we first need to make the constructor private which follows secure guidelines.
  3. Another issue to address is the user input and make sure to include input validation on all user input to prevent attacks or injection.
  4. The final task would be to look at some kind of access control for the application for encapsulation and security.