# CS 305 Module Two Written Assignment

## Areas of Security

From the first level of the vulnerability assessment flow diagram the input validation, API and code quality are the three most relevant areas of security for this software application.

## Areas of Security Justification

**Input Validation:** One of the most common threats to security are injection attacks and proper input validation helps to prevent these sorts of attacks. Additionally, if there are good input validation practices in place then we can make sure that malicious inputs or situations where a user inputs the incorrect information are properly handled.

**APIs:** We will likely be using APIs in our program we need to ensure that we are following the best practices in order to enable secure communication between the database and the API. We need to ensure that the proper countermeasures are in place so that bad actors can’t use the same APIs to gain access to or manipulate our data.

**Code Quality:** Writing proper code with security in mind serves as a prophylactic measure against future security threats. It doesn’t matter how secure the rest of the factors are if the source code itself is poorly written. For example, a properly written prepared statement in the code could help mitigate and prevent SQL injections, while code written without prepared statements are vulnerable to SQL injection attacks. Additionally, it is important to use the best coding practices like testing the code to ensure that small and easy to make mistakes are detected early on and fixed before release.

## Code Review Summary

One issue I found with the code is that in the pom.xml file the team is using version 2.6.5 of the spring-data-rest-webmvc which is outdated and could cause compatibility issues in the code. Another potential issue with the code is there are no input validation measures in place specifically in the GreetingsController class. This means that if the user enters incorrect or potentially malicious inputs then the program does not know how to handle it.

## Mitigation Plan

To resolve the issue of the outdated spring data API we can simply update the API to it’s most recent version 5.0.0-M1, and use that instead. To help fix the issue of input validation you could include a try and catch block to handle errors.