# CS 305 Module Two Written Assignment Template

Abdulrahman Al-Nachar

CS305 / 2-1

## Areas of Security

1. Input validation: It’s imperative for our program to have have strong efficient inputs validation coded into it, to work against being exposed to injection attacks.
2. API’s Interactions: use SSL/HTTPS to secure interactions between client and server, set rate limits, implement authentication and authorization controls to prevent unwanted access and users having too much authority in the program.
3. Cryptography: make sure key data is secured from potential attackers
4. Distributed system: make sure that a working system is secure in a distributed environment that can control authentication, authorization and each part of the system works together.
5. Error Handling: Test and handle all errors are properly worded and are useful to the necessary parties, that helps us protect against any attack through this angle.
6. Code Quality: make sure the program and code follow secure coding guidelines and best practices.
7. Encapsulation: make sure the data structures in the program is secure and that they cannot be accessed or modified by unwanted actors.

## Areas of Security Justification

It’s important to make sure that the program is secure and safe to use, successful attacks of any kind can expose private data, modify the backend, change how the program works, and allow for unauthorized use of the program.

## Code Review Summary

1. In the greetingController.java file, the **public** Greeting greeting() method takes an input without validating the input that can expose the program to an injection attack
2. In the same java file, with the number() method , this method can get an out of bound error
3. Our API Endpoints have no authentication or authorization controls on them.
4. The code lacks any cryptography.
5. Lastly this version of the spring framework is old.

## Mitigation Plan

1. Ensure that the **public** Greeting greeting() method validates inputs using length limits or limiting the characters that users can input.
2. Ensure that the number() method safely handles the error out of bound error.
3. Setup authentication and authorization controls, add spring framework’s dependencies for it in the POM.xml file.
4. I understand that the code doesn’t take in any sensitive data from users, but if the program was to be scaled and later on take in sensitive data, then there would be a need to have strong encoding to that data.
5. Use the latest version of the spring framework.