# CS 305 Module Two Code Review and Mitigation Plan Assignment

## Instructions

Replace the bracketed text with your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Areas of Security

* [Input validation: It’s always good to validate your input validation. This is just another way to decrease the vulnerability of SQL injection.
* API(Your is what allows your apps to converse with one another. DAO layers should be used in Greetings.java. Because you have methods like getID, and getContent. DAO must be parameterized in order to prevent SQL injection.
* Cryptography (Since you have two applications talking to one another, you should ensure that you data is always protected from SQL injection.
* Client/Server (When using API, we must always ensure that data transfers are secure. There is definitely risk of some kind of URL https SQL injection.
* Code Error (Always ensure that your code is neatly set up with no spelling errors. Make sure that all input functions are reviewed thoroughly.
* Encapsulation (If we have a method that calls a function within the system, this is untrusted input. It is important to apply query parameterization at this point to protect your code from SQL injection.
* [Be sure to justify your reasoning for why each area is relevant to the software application.]

## Code Review Summary

* [1) The 2.6.5 version of the Spring framework will need to be updated to the most current version of itself.

2) in GreetingController.java, There is not a SpringbootApplication here->@GetMapping("/greeting")

**public** Greeting greeting(@RequestParam(value = "name", defaultValue = "World") String name) {

Expression exp = parser.parseExpression(name);

String message = (String) exp.getValue();

System.***out***.println(message);

**return** **new** Greeting(counter.incrementAndGet(), String.*format*(***template***, message));

}

Not having a SpringbootApplication here makes the input vulnerable to SQL injection

* .]

## Mitigation Plan

* [The first layer of security, is to update the Spring frame work to the latest version of itself. This will ensure that the software can adapt to the latest and greatest threats against your code. Secondly, add SpringBootApplication to GreetingController.java. This will ensure that when parsing, the input information can not be manipulated by an intruder. You are basically query parameterizing your code with this.