# 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

Input Validation: The system requires input from the user.

APIs: The system requires input from outside an intranet.

Cryptography: Because information is being transferred.

Client/Server: Use of security certificates between client and server.

Code Error: Make sure that all code errors are caught and the exceptions are handled.

Code Quality: High-quality code is easy to maintain, read, and therefore debug.

Encapsulation: Important in reducing a user’s interactions with the inner workings of the system.

## Areas of Security Justification

Input Validation: Validating input from the user prevents SQL Injection attacks.

APIs: APIs such as RESTful can handle HTTP web requests securely. The requests from a client to a server must contain all the information needed to process the request and the server does not store client information between these requests.

Cryptography: Secures communication through algorithms, it involves transforming data into non-understandable language to unauthorized users or encrypting. This is essential when the transfer of data is done.

Client/Server: Through the client/server relationship, information is encrypted during transfers and with the use of certificates that information can be unscrambled at its destination.

Code Error: To prevent circumventing rules and roles through input, the code must be able to handle all exceptions thrown by errors.

Code Quality: To ensure that security is in place throughout the code, the code must be authored efficiently and logically to reduce the risk of open vulnerabilities.

Encapsulation: It is extremely important to only allow users to interact with variables that absolutely require input. This is achieved by encapsulating attributes and functions into classes.

## Code Review Summary

The code is done on an out-of-date version and could falter with potential obsolete or lacking security definitions.

## Mitigation Plan

Update the code to the newest version of spring data.