# 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

* To design this new functionality safely and securely for expressive command input we will of course need to ensure inputs are validated and API interactions are handled securely. If the inputs are anticipated to carry sensitive data, then cryptography will need to be addressed as well.
* Since there is an input that is being used to fetch data of some kind, there is a risk of input injection and thus each interaction will need to be sanitized or otherwise handled properly. The application is a web service, receiving and transmitting data over the internet. Therefore, API interactions are happening and will need to be handled securely. Lastly, if sensitive information is handled over these inputs, good practice will be needed to ensure that information is not leaked.

## Code Review Summary

* Having now looked at the code base, I can see that this application will not be used to transmit sensitive data so cryptography will not likely be needed. I also see no input expected from a user at any point in the program. User is maybe expected to type in the proper URL, but that cannot be validated until it reaches the application. Proper API handling will still need to be implemented.
* Another point to make is that there does not seem to be any error handling. For example, in GreetingController, we see that an ID is parsed from the URL. We expect an integer but have no handling if we unexpectedly get a string instead.

## Mitigation Plan

* API Interactions
  + Validate any URLs that hit this server. Sanitize any characters that cannot be utilized for this service. Handle unexpected URLs
  + Is it intended that anyone can make a greeting? If not, then this functionality will need to be locked down to only certain users meaning authentication will need to take place.
* Error Handling
  + Ensure the application can handle unexpected IDs. Currently, the id endpoint creates a plain variable. This could come as anything. Handle other types by creating case logic to decide whether to use the data or return an error. This can also be said with the greeting endpoint. Should the parameter not come as a string, that would need to be handled.