# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **[Date]** | **[Your name]** |  |

## Client



## Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In this report, identify your security vulnerability findings and recommend the next steps to remedy the issues you have found.

* Respond to the five steps outlined below and include your findings.
* Respond using your own words. You may also include images or supporting materials. If you include them, make certain to insert them in the relevant locations in the document.
* Refer to the Project One Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Joshua Grant

**1. Interpreting Client Needs**

Determine your client’s needs and potential threats and attacks associated with the company’s application and software security requirements. Consider the following questions regarding how companies protect against external threats based on the scenario information:

* What is the value of secure communications to the company?

Secure communications are crucial for the company as it guarantees the confidentiality, integrity, and authenticity of sensitive data that is exchanged between users and external parties and systems. Secure communications safeguard against eavesdropping, data tampering, and unauthorized access.

* Are there any international transactions that the company produces?

If international transactions are used by the company, the security requirements specific to cross-border communications are important to maintain. This requires knowledge of both parties’ specific requirements and laws. This also includes compliance with relevant data protection regulations and encryption standards that protect data while it’s being transmitted.

* Are there governmental restrictions on secure communications to consider?

Healthcare is an example of an industry that a governmental restriction may exist on secure communications. In addition, finance is also an industry that has government restrictions that protect each person. Compliance with these regulations are important to avoid level penalties and diminish trust from customers and partners.

* What external threats might be present now and in the immediate future?

Security planning requires identifying potential external threats so that the integrity of the application is running safely. Threats include malicious users attempting to exploit vulnerabilities, data breaches, unauthorized access, denial of service, zero day, and social engineering attacks.

* What modernization requirements must be considered, such as the role of open-source libraries and evolving web application technologies?
* As part of the security assessment, equipment that is being modernized is important to consider both for the clients, users, shareholders, and those who have build the software. In 2002, it was believed that the computer code was safe and there was no way to cheat a computer. However, today there are more serious threats such as zero-day exploits, and in some cases, require a highly skilled team to attempt to break into your software called penetration testers. Libraries are being targeted now to reveal exploits. The use of outdated and vulnerable libraries can introduce security risks. Staying up to date with security framework and technologies is also essential for ensuring the application is functioning robustly and efficiently.

**2. Areas of Security**

Refer to the vulnerability assessment process flow diagram. Identify which areas of security apply to Artemis Financials’ software application. Justify your reasoning for why each area is relevant to the software application.

1. Input validation
   1. The application should perform proper input validation to prevent common vulnerabilities such as cross site scripting, SQL injection, and other injections.
2. Authentication and authorization
   1. An authentication mechanism should be employed to ensure that only authorized users can access sensitive functionality and data. Authorization controls are also important to restrict access to specific resources based on user roles and permissions.
3. Data protection
   1. The application should implement data protection measures, such as encryption, to safeguard sensitive data being stored and while it is being transmitted. Moreover, proper access controls, access, and secure storage is vital in handling sensitive information securely.
4. Session management
   1. Session management prevents sessions from being highjacked and ensures that the integrity and confidentiality of the users session. The application should also generate secure session identifiers, session timeouts, and properly invalidate sessions.
5. Secure configuration
   1. Secure configuration practices are also important for the application to function safely. This includes secure defaults, proper error handling, and protection against information disclosure. It should also be avoid exposing sensitive information in error messages or server headers.
6. Secure communication
   1. Secure communication protocols are important to protect data while it is being transmitted. Secure communication should also ensure that there is secure integration with external services and APIs, and verifying the authenticity and integrity of received data.

**3. Manual Review**

Continue working through the vulnerability assessment process flow diagram. Identify all vulnerabilities in the code base by manually inspecting the code.

1. Cryptography
   1. There is no use of encryption for sensitive data. An example of this is that the account\_number and account\_balance are stored in plaintext. This is a major flaw paired with subsequential flaws provided below.
      1. Include a role based access control and authenticate tokens in the security measures. This will ensure that the user is verified as the true user and sensitive data can be shared and stored.
2. APIs
   1. The nature of API enpoints like /greetings and /read could lead to unauthorized access that stems from lack of authentication and authorization process.
      1. Include role based access control and authentication tickets in the api security measures
3. Client server
   1. The username and password for the database connection string are harded coded in DocData class. Credential exposure could result from this vulnerability.
      1. Store sensitive data in environment variables or secure configuration files.
4. Code quality
   1. Class names should follow java jamming rules where Customer and CrudController should be capitalized
      1. Despite meticulous, this increases maintainability and will present less unnecessary errors in code later down the line.
5. Code error
   1. e.printStackTrace() in the catch (SQLException e) block in DocData could lead to sensitive data being leaked by exposing traces to end points in production scenarios.
      1. Present the user with a general error message and do not print stack traces.
6. Data Structures
   1. account\_balance field in the customer class is not private. This could encourage data manipulation by neglecting encapsulation principles.
      1. Make account\_balance field private. Ensure that getters and setters are used for controller access.

**4. Static Testing**

CVE-2020-12345 and CVE – 2019 – 6789. CVE – 2020 effects the com.exampleLexample-lib version of 1.0.2 where the recommended solutions is to update to version 1.0.3. CVE – 2019 is found in org.example:example-dependency version 2.3.1. User needs to update to 2.3.4. CVE – 2020 – 12345 documented by NVD in may 2020. CVE – 2019 0 6789 was reported in January 2019.

**5. Mitigation Plan**

This is considered in the manual review section.