Secure Application Development: Java Security and Application Testing

NCI Post Graduate Diploma Cyber Security

Author: Jonathon Taaffe

Title: Secure Application Development: Java Security and Application Testing
Author: Jonathon Taaffe

Warning and Disclaimer
The information is provided on an “as is” basis. The author shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this publication. The opinions expressed in this publication belong to the author.

Trademark Acknowledgments
All terms mentioned in this publication that are known to be trademarks or service marks have been appropriately capitalised. The author cannot attest to the accuracy of this information. Use of a term in this publication should not be regarded as affecting the validity of any trademark or service mark.

Contents

Java Introduction

Java is the most widely used computer programming language in the world. According to TIBOE[1], the Software Quality Company, Java has been the dominant programming language since 2001[2]. Java is a Write Once, Run Anywhere (WORA)[3] Object Orientated Programming language. It is a Cross-Platform Language; it's not platform specific. Its core architecture utilises Java Virtual Machine (JVM)[4] technology, an environment in which Java loads, verifies, executes and runs code abstracted from the underlining Operating System. This JVM architecture enables Write Once, i.e. develop and code Java applications, and Run Anywhere, i.e. JVM will manage Java applications on all major Operating Systems[5].

State of the Art – Latest Programming Paradigms

Security of Java [6]

Java's security architecture provides API's for Platform Security, Cryptography, Authentication, Access Controls and Secure Communications with Public Key Infrastructure (PKI). Developers can integrate these security API's into their Java applications, which provides developers with:

- Platform Independence: Developers do not need to code security into their applications, the Java platform provides security API's which can be integrated into Java applications.
- Interoperability: Applications are not constrained to a particular security provider, and, conversely, providers are not constrained to a particular Java application.
- Algorithm Extensibility: Java provides an extensive set of algorithmic services that can be extended to include custom algorithms.

Platform Security

The Java Virtual Machine (JVM) and Java Compiler offer type-safe or strong data typing, automatic memory management, bytecode verification and secure class loading.

- Type-Safe or Strong Data Typing: Ensures coding errors and exceptions are detected at compile time prior to run time; defined variables must be assigned to a data type.
- Automatic Memory Management: Java manages memory allocation by automatically allocating and de-allocating objects; called Garbage Collection. This process reduces both memory corruption and the potential for vulnerabilities.
- Bytecode Verification: Java bytecode is the coding instruction set used by Java, which is not native machine code, but Java specific. JVM performs bytecode verification prior to code execution to ensure non-execution of unsafe code.
- Secure Class Loading: Prevents untrusted code from impacting Java programs runtime.

---

[1] TIOBE.com [2019] *TIOBE (the software quality company)* https://www.tiobe.com/ [Accessed 11th December 2019].
[2] TIOBE.com [2019] *The Java Programming Language* https://www.tiobe.com/tiobe-index/java/ [Accessed 11th December 2019].
[3] ComputerWeekly.com [2002] *Write once, run anywhere?* https://www.computerweekly.com/feature/Write-once-run-anywhere [Accessed 11th December 2019].
[4] Wikipedia.org [2019] Object-Oriented Programming https://en.wikipedia.org/wiki/Object-oriented_programming [Accessed 11th December 2019].
[5] Java.com [2019] *Java Downloads for All Operating Systems* https://www.java.com/en/download/manual.jsp [Accessed 11th December 2019].
[6] Oracle.com [2019] *Java SE Security* https://www.oracle.com/java/technologies/index-jsp.html [Accessed 11th December 2019].

Cryptography

The Java security architecture supports the following cryptographic features:

- Services: Digital Signatures, Message Digests, Ciphers, Message Authentication Codes, Key Generators and Factories
- Algorithms: RSA, DSA, AES, Triple DES, SHA, PKCS#5, RC2, and RC4

Authentication and Access Controls

The authentication and access controls mechanisms Java includes are:

- Extensive Login Mechanisms: Single-Sign-On (SSO)
- Policies and Permissions: Role and Identity-base Access Control.

Secure Communication

Java security architecture supports secure network communication protocols to ensuring privacy and integrity of data in transit including:

- Transport Layer Security (TLS)/Secure Sockets Layer (SSL)/ HTTPS over SSL/TLS
- Kerberos Directory Authentication
- Simple Authentication and Security Layer (SASL)

Public Key Infrastructure (PKI)

The Java security architecture also supports the following core PKI services essential in providing data confidentiality and integrity:

- Certificates and Certificate Revocation Lists
- Certification Path Validators and Builders
- KeyStores
- Certificate Stores

Java Object Orientated Programming (OOP)[7]

Java is an Object Orientated Programming language which utilises the construct of Objects in which data properties or attributes are stored and Code Procedures, known as methods, are executed. Methods can access and modify Objects as required. Java is a class-based OOP, where classes of objects are defined allowing for inheritance of classes rather than inheritance of objects.

Java OOP Features

Java utilises the following OOP concepts to enhance secure programming:

- Encapsulation: Combining variables (data) and methods (code acting on data) into a single unit.
- Inheritance: Classes can obtain or inherit methods and data fields of other classes.
- Access Modifiers: Restrict access to class implementations
- Type-Safe: Code errors and exceptions detected at compile time prior to run time.

Encapsulation: Not only does this combine the variables and methods into a single unit, encapsulated classes can be made private classes. This is known as data hiding and ensures the variables of a private class are hidden from other classes. Setter and Getter methods are then used to access the data in private classes. Encapsulation also allows class fields to be read-only or write-only, granting the class full control over class data fields.

Inheritance: This ensures pre-defined classes can be re-used reducing the potential for coding errors.

Access Modifiers: Access modifiers can be assigned to classes, methods and fields which enables class access restriction. Java has 4 access levels: public, protected, private, and package:

- Public: Access by anyone
- Protected: Access to subclass, or any other class within the specific package
- Private: Access restricted to within a specific class
- Package-level: Access to classes in the same package

Type-Safe: A data type strictly determines the operations that can be performed on data. This ensures that only the valid operations for a specific data type can be performed against that specific data type.

[7] Wikipedia.org [2019] *Object-oriented programming* https://en.wikipedia.org/wiki/Object-oriented_programming [Accessed 11th December 2019].

Java Virtual Machine (JVM) Architecture

The Java Virtual Machine architecture provides many security features. The following diagram is a high-level overview of the Java Virtual Machine architecture

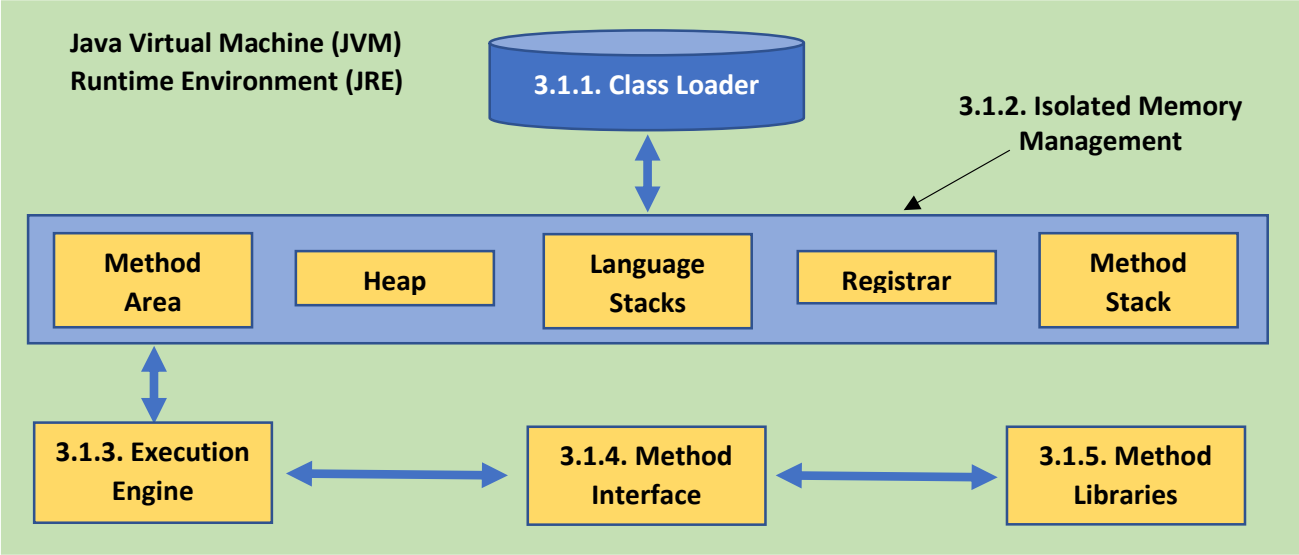

Figure 1. Java Virtual Machine (JVM) Runtime Environment (JRE)[8]

JRE employs a zero-trust policy; it does not trust incoming code.
JVM abstracts and separates processes providing defense-in-depth security.

JVM Security Related Features[9]

Class Loader

- Load, Link and Initialise Java classes
- Separates packaged classes from the underlying Operating System.
- SecurityManager Class: Applies security policies to ensure code execution is performed in a secure context.[10]

Isolated Memory Management: Defined isolated memory area in which to load, verify, execute and run code

- Method Area: Store class structures and provide runtime environment
- Heap: Objects, related instance variables and array storage
- Language Stacks: Dynamic variable and results storage
- Registrars: JVM instruction execution address storage
- Method Stack: Native code instruction and library storage

Execution Engine: Bytecode verifier checks code for illegal code violations and managed code execution.
Method Interface: Programming framework to call native libraries and applications
Method Libraries: C, C++ libraries required by the Execution Engine

[8] Taaffe, Jonathon [2019] *Figure 1. Java Runtime Environment (JRE)* [Created 11th December 2019].
[9] Guru99.com [2019] *What is JVM (Java Virtual Machine)? with Architecture: JAVA Programming Tutorial* https://www.youtube.com/watch?v=G1ubVOl9IBw [Accessed 11th December 2019].
[10] Oracle.com [2019] *Class SecurityManager* https://docs.oracle.com/javase/7/docs/api/java/lang/SecurityManager.html [Accessed 11th December 2019].

Class Loader[11]

The Java Class Loader ensures only secure classes are loaded and verifies the classes ensuring valid Java code is processed. The Class Loader maintains class loading into Java runtime execution, and determines the following code characteristics:

- Code source location
- Code signing properties
- Default permissions

This Class Loading process applies to all class loading into Java runtime execution, regardless of whether the code was from the local file system or from a remote network location.

Isolated Memory Management

Java automatically manages memory allocation by allocating and de-allocating objects as needed. This process reduces both memory corruption and the potential for vulnerabilities and is called the Garbage Collector.

Bytecode[12]

Java bytecode is the coding instruction set used by Java, which is not native machine code, but Java specific. JVM performs bytecode verification prior to code execution to ensure unsafe operations are not being executed. The bytecode verification process includes:

- Code fragment format verification
- Theorem prover to determine code is non-malicious ensuring
  - Pointers are not forged
  - Access restrictions are not violated
  - Accesses objects defined functions

Java's bytecode verification process is a core feature of Java's security design, enforcing run-time code verification preventing interface violation.

Interface[13]: The front-end GUI of a Java application constructed from Java methods.

[11] Oracle.com [2019] *Java™ Security Overview* https://docs.oracle.com/javase/8/docs/technotes/guides/security/overview/jsoverview.html [Accessed 11th December 2019].
[12] Oracle.com [2019*] The Java Language Environment* https://www.oracle.com/technetwork/java/security-136118.html [Accessed 11th December 2019].
[13] Oracle.com [2019] *What Is an Interface?* https://docs.oracle.com/javase/tutorial/java/concepts/interface.html [Accessed 11th December 2019].

Bytecode Verification Process
The bytecode verifier inspects the bytecode, assembles state information of that code and verifies the bytecode parameters.

he following diagram outlines the Java bytecode verification process, and the controls Java enforces during verification.

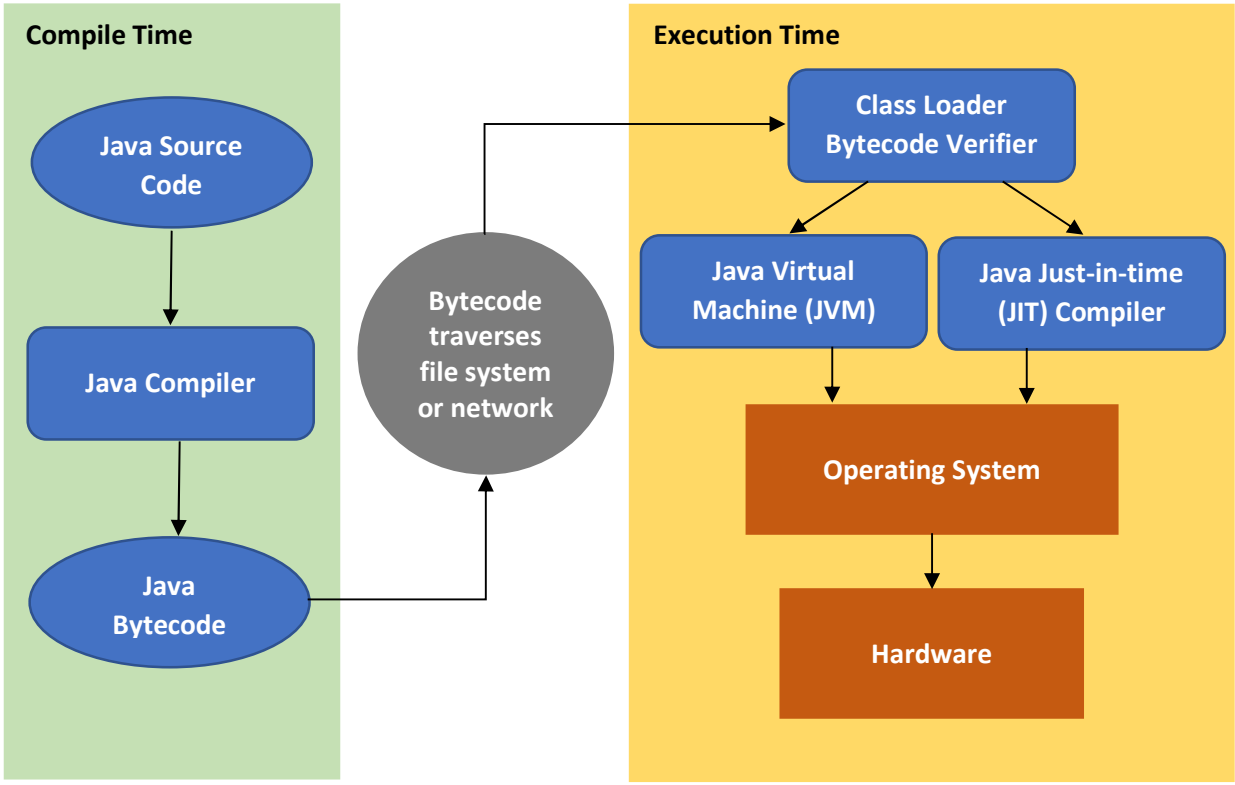

Figure 2. Bytecode Verification Process[14]

The figure above outlines the data flow control from Java Source Code, through the Java Compiler and Java Bytecode verifier and on to the Java Virtual Machine (JVM).

The Java Class Loader and Bytecode Verifier function on a trust-less basis and do not make any distinctions on the bytecode source, be it local or network-based bytecode. The Bytecode Verifier analyses the bytecode before it is passed to the Java Interpreter (JVM/JIT).

The Bytecode verifier analysis ensure the bytecode is properly formatted for execution and can be processed by the Java Interpreter.

The bytecode verifier ensures:

- No operand stack overflows, or underflows are contained in the code
- Bytecode parameter types are correct
- Object fields are private, public or protected

With the Bytecode verifier completing all operand and stack overflow analysis, the Java interpreter can process the bytecode as optimally as possible.

[14] Taaffe, Jonathon [2019] *Figure 2. Bytecode Verification Process* [Created 11th December 2019].

Java Language Security

In addition to the security features outlined so far in this report, Java also includes a security Package with security Classes available for programming. The java.security Package[15] includes API's for algorithms, services and protocols. Java security API's provide functionality for the following core security services:

- Cryptography and Public Key Infrastructure (PKI)
- Authentication and Access Control
- Secure Communication

Integrating Java Security Services

The current Java Development Kit (JDK)[16] for Java 11 includes a Provider Architecture[17] from which Security Services can be implemented. Through the provider architecture developers can integrate security into their applications without having to understand their underlying configuration. JDK Modules that include security API's are:

Module	Description
java.base [18]	Foundational APIs: java.security, javax.crypto, javax.net.ssl, javax.security.auth
java.security.jgss [19]	GSS-API: Kerberos v5 and SPNEGO
java.security.sasl [20]	SASL: DIGEST-MD5, CRAM-MD5, and NTLM
java.smartcardio [21]	Java Smart Card I/O API
java.xml.crypto [22]	Cryptography XML API
jdk.jartool [23]	JAR File Signing API's
jdk.security.auth [24]	javax.security.auth.* interfaces and authentication modules
jdk.security.jgss [25]	GSS-API and SASL GSS-API extensions

Table 1. JDK Modules including Security API[26]

[15] Oracle.com [2019] *Package java.security* https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/security/package-summary.html [Accessed 11th December 2019].
[16] Oracle.com [2019] *Java SE Development Kit 11 Downloads* https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html [Accessed 11th December 2019].
[17] Oracle.com [2019] *Introduction to JDK Providers* https://docs.oracle.com/en/java/javase/11/security/oracle-providers.html [Accessed 11th December 2019].
[18] Oracle.com [2019] *JDK Security API: java.base* https://docs.oracle.com/en/java/javase/11/docs/api/java.base/module-summary.html [Accessed 11th December 2019].
[19] Oracle.com [2019] *JDK Security API: java.security.jgss* https://docs.oracle.com/en/java/javase/11/docs/api/java.security.jgss/module-summary.html [Accessed 11th December 2019].
[20] Oracle.com [2019] *JDK Security API: java.security.sasl* https://docs.oracle.com/en/java/javase/11/docs/api/java.security.sasl/module-summary.html [Accessed 11th December 2019].
[21] Oracle.com [2019] *JDK Security API: java.smartcardio* https://docs.oracle.com/en/java/javase/11/docs/api/java.smartcardio/module-summary.html [Accessed 11th December 2019].
[22] Oracle.com [2019] *JDK Security API: java.xml.crypto* https://docs.oracle.com/en/java/javase/11/docs/api/java.xml.crypto/module-summary.html [Accessed 11th December 2019].
[23] Oracle.com [2019] *JDK Security API: jdk.jartool* https://docs.oracle.com/en/java/javase/11/docs/api/jdk.jartool/jdk/security/jarsigner/package-summary.html [Accessed 11th December 2019].
[24] Oracle.com [2019] *JDK Security API: jdk.security.auth* https://docs.oracle.com/en/java/javase/11/docs/api/jdk.security.auth/module-summary.html [Accessed 11th December 2019].
[25] Oracle.com [2019] *JDK Security API: jdk.security.jgss* https://docs.oracle.com/en/java/javase/11/docs/api/jdk.security.jgss/module-summary.html [Accessed 11th December 2019].
[26] Taaffe, Jonathon [2019] *Table 1. JDK Modules including Security API* [Created 11th December 2019].

java.security.Provider Class

This is the security provider class in Java and allows for multiple security API's to be configured simultaneously. Providers are listed and implemented based on priority; highest priority provider is implemented first.

To request a security service from a provider, firstly import the java.security.*class and use the getInstance method, for example to create a cryptographically secure RSA key use the 'KeyFactory.getInstance("RSA")'[27] method.

Cryptography – Java Cryptography Extension (JCE) [28]

JDK 11 Cryptography framework assists access to and development of the following core cryptographic services:

- Message Digest and Digital Signature Algorithms
- Symmetric, Asymmetric and Stream Encryption
- Elliptic Curve Cryptography (ECC)
- Key-based Algorithms
- Message Authentication Codes (MACs)
- Secure Random Number Generators (RNG's)

JDK 11 includes providers for the RSA, DSA and ECDSA signature-based algorithms, and providers for the AES, SHA-2, Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) encryption-based algorithms.

Public Key Infrastructure (PKI)[29]

PKI infrastructure enables data-in-transit encryption using Public and Private cryptographic key pairs[30]. The java.security, java.security.cert and the java.security.KeyStore packages provide API's for Java's PKI integration. This provides developers the ability to integrate PKI, or more specifically include secure cryptographic functions into their Java programs and applications. PKI integration is provided through the following tools:

- keytool: used for the management of key stores including
  - Creating cryptographic key pairs
  - X.509 certificate management
  - Submit certificate requests to Certificate Authorities (CA's)[31]
  - Certificate management
- Jarsigner: used to digitally sign and verify Java Archive (JAR)[32] files

---

[27] Oracle.com [2019] *Module java.base, Package java.security, Class KeyFactory* https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/security/KeyFactory.html [Accessed 11th December 2019].
[28] Oracle.com [2019] *Java Cryptography* https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-C6D250FC-F147-4284-A6BF-8384DFD39DA6 [Accessed 11th December 2019].
[29] Oracle.com [2019] *Java Public Key Infrastructure* https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-054AD71D-D449-47FF-B6F7-F416DA821D46 [Accessed 11th December 2019].
[30] Techopedia.com [2019] *Public Key Infrastructure (PKI)* https://www.techopedia.com/definition/4071/public-key-infrastructure-pki [Accessed 11th December 2019].
[31] Techopedia.com [2019] *Certificate Authority* (CA) https://www.techopedia.com/definition/29742/certificate-authority-ca [Accessed 11th December 2019].
[32] Oracle.com [2019] *Java Archive (JAR) Files* https://docs.oracle.com/javase/8/docs/technotes/guides/jar/index.html [Accessed 11th December 2019].

Authentication – Java Authentication and Authorization Service (JAAS)[33]

This process is fundamental in providing authentication, i.e. user identity verification, to a Java program. Included in the javax.security.auth.login package is the LoginContext class which provides secure authentication through the javax.security.auth.spi.LoginModule interface.

The following figure provided by Oracle highlights the independence between Java Applications, the Authentication Framework and the Authentication mechanisms including Smartcard, Kerberos, Username/Password mechanisms, being used:

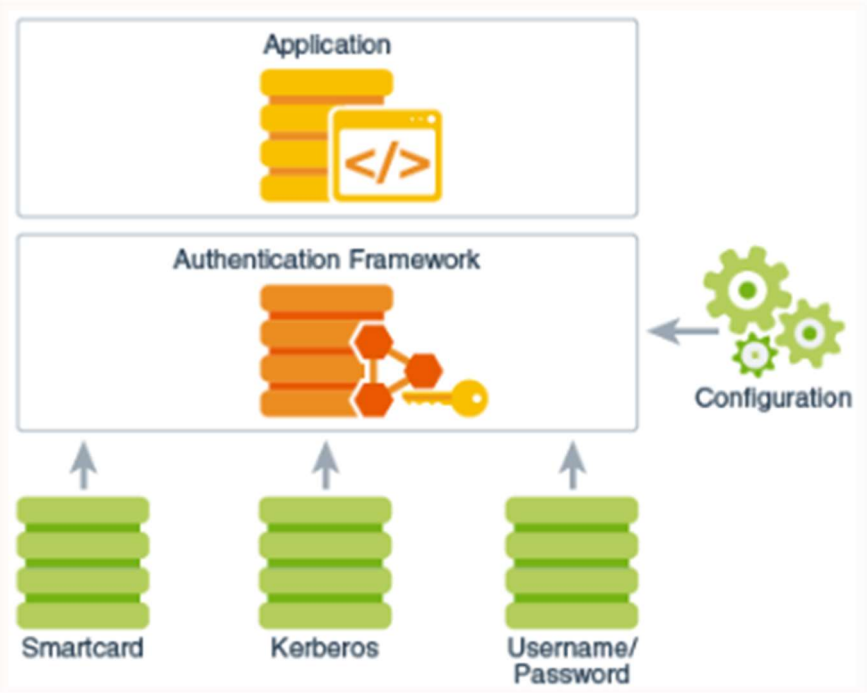

Figure 3. Java Application, Authentication Framework and Authentication Mechanism Independence[34]

The Java com.sun.security.auth.module package provides for the following Authentication Mechanisms

- Krb5LoginModule: Kerberos authentication
- JndiLoginModule: LDAP/NIS Username/Password Authentication
- KeyStoreLoginModule: Certificate-based authentication

[33] Oracle.com [2019] *Java Authentication* https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-F8BE6C49-3506-4D1A-8E4E-053CA439D1E2 [Accessed 11th December 2019].
[34] Oracle.com [2019] *Figure 3. Java Application, Authentication Framework and Authentication Mechanism Independence* https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-F8BE6C49-3506-4D1A-8E4E-053CA439D1E2 [Accessed 11th December 2019].

Secure Communication – Java Secure Socket Extension (JSSE)

SSL/TLS/DLTS[35]

To ensure secure data-in-transit, Java utilises the cryptographic services outlined above in conjunction with the following secure end-to-end transport communication protocols to facilitate server and client authentication, data confidentiality and data integrity:

- Secure Sockets Layer (SSL) [36]
- Transport Layer Security (TLS) [37]
- Datagram Transport Layer Security (DTLS) [38]

The following specific Java classes provide for these secure communications:

- javax.net.ssl.SSLSocket: Encapsulated TLS network socket support
- javax.net.ssl.SSLEngine: TLS/DTLS packet management
- javax.net.ssl.KeyManager: Authentication key management
- javax.net.ssl.TrustManager: Certificate key store management

Simple Authentication and Security Layer (SASL)[39]

The java.security. sasl module provides Simple Authentication and Security Layer (SASL) support for authentication and client/server security and includes support for:

- Client-side: CRAM-MD5, DIGEST-MD5, EXTERNAL, GSSAPI, NTLM, PLAIN
- Server-side: CRAM-MD5, DIGEST-MD5, GSSAPI, NTLM

Access Control[40]

Java access control provides protection for sensitive resources and Java application code. The java.lang.SecurityManager class manages this protection mechanism, which it activates by installing a Java SecurityManager at runtime.

Access control can be configured for the following properties:

- Permissions: Securely manage access to system resources
- Security Policy: Extend system resource permissions
- Access Control Enforcement: Call stack evaluation to verify requested access

---

[35] Oracle.com [2019] *Java Secure Communication* https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-4E5FEEF5-A541-4222-AD18-31AE184F38E4 [Accessed 11th December 2019].
[36] SSL.com [2019] *What is SSL?* https://www.ssl.com/faqs/faq-what-is-ssl/ [Accessed 11th December 2019].
[37] Techopedia.com [2019] *Transport Layer Security (TLS)* https://www.techopedia.com/definition/4143/transport-layer-security-tls [Accessed 11th December 2019].
[38] Wikipedia.org [2019] *Datagram Transport Layer Security* https://en.wikipedia.org/wiki/Datagram_Transport_Layer_Security [Accessed 11th December 2019].
[39] Oracle.com [2019] *Simple Authentication and Security Layer (SASL)* https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-7C74ECA4-3645-4756-B8FB-63D84480F4AD [Accessed 11th December 2019].
[40] Oracle.com [2019] *Java Access Control* https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-BBEC2DC8-BA00-42B1-B52A-A49488FCF8FE [Accessed 11th December 2019].

Java Security Classes[41]

The following table outlines the packages, classes and modules specific to Java security:

Package	Class/Interface Name	Module
java.lang	SecurityException	java.base
java.lang	SecurityManager	java.base
java.lang	System	java.base
java.security	AccessController	java.base
java.security	DomainLoadStoreParameter	java.base
java.security	Key	java.base
java.security	KeyStore	java.base
java.security	MessageDigest	java.base
java.security	Permission	java.base
java.security	Policy	java.base
java.security	Provider	java.base
java.security	Security	java.base
java.security	Signature	java.base
java.security.cert	Certificate	java.base
java.security.cert	CertStore	java.base
java.security.cert	CRL	java.base
javax.crypto	Cipher	java.base
javax.crypto	KeyAgreement	java.base
javax.net.ssl	KeyManager	java.base
javax.net.ssl	SSLEngine	java.base
javax.net.ssl	SSLSocket	java.base
javax.net.ssl	TrustManager	java.base
javax.security.auth	Subject	java.base
javax.security.auth.kerberos	KerberosPrincipal	java.base
javax.security.auth.kerberos	KerberosTicket	java.base
javax.security.auth.kerberos	KerberosKey	java.base
javax.security.auth.kerberos	KerberosTab	java.base
javax.security.auth.login	LoginContext	java.base
javax.security.auth.spi	LoginModule	java.base
javax.security.sasl	Sasl	java.security.sasl
javax.security.sasl	SaslClient	java.security.sasl
javax.security.sasl	SaslServer	java.security.sasl
jdk.security.jarsigner	JarSigner	jdk.jartool
org.ietf.jgss	GSSContext	java.security.jgss
com.sun.security.auth.module	JndiLoginModule	jdk.security.auth
com.sun.security.auth.module	KeyStoreLoginModule	jdk.security.auth
com.sun.security.auth.module	Krb5LoginModule	jdk.security.auth

Table 2. Java Security Classes[42]

[41] Oracle.com [2019] *Java Security Classes* https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-CF323502-F719-4618-91FE-4D37CB57FF24 [Accessed 11th December 2019].

[42] Taaffe, Jonathon [2019] *Table 2. Java Security Classes* [Created 11th December 2019].

Secure Coding in Java Guidelines[43]

The previous section of this report outlined the security features available with the Java platform. But it is the responsibility of the Java developer to integrate these security features into their Java applications and programs. Oracle have provided extensive guidelines on how to securely code in the Java programming language which developers must take heed of to ensure their Java applications and programs are as secure as possible.

Fundamentals [44]

The following guidelines are fundamental when programming in Java:

1. Develop transparent Java code that can be statically analysed to confirm security posture.
2. Ensure security is integral in all API development.
3. Duplicated Java code can execute differently to intended outcome.
4. Always apply least privilege to permissions.
5. Data crossing trust boundaries must be both sanitised and validated
6. Perform SecurityManager security checks only at key code execution points.
7. Always define object fields as private. Method, class, package and module interfaces should have clearly defined behaviors
8. All API security-related information should always be documented and kept up to date.

Denial of Service (DoS) [45]

Denial of Service can cause the resources of a system to become exhausted and unable to process valid requests. To mitigate DoS attacks, adhere the following guidelines.

1. Sanitise and validate activities that require excessive resources including:
   - Large file access
   - Integer overflow causing size checking processing to fail.
   - Parsing text or binary streams
   - Zip decompression bomb
   - XML entity expansion
   - Hash table attack
   - Backtracking of regular expressions
   - XPath expressions
   - Java and Java Beans XML deserialisation
   - Extensive unusual behaviour logging
   - Infinite loops
2. All resources must be promptly released
3. Ensure resource limit checks, i.e. bounds checking on arrays, do not cause integer overflows

[43] Oracle.com [2019] *Secure Coding Guidelines for Java SE* https://www.oracle.com/technetwork/java/seccodeguide-139067.html [Accessed 11th December 2019].
[44] Oracle.com [2019] *Secure Coding Guidelines – 0. Fundamentals* https://www.oracle.com/technetwork/java/seccodeguide-139067.html#0 [Accessed 11th December 2019].
[45] Oracle.com [2019] *Secure Coding Guidelines – 1. Denial of Service* https://www.oracle.com/technetwork/java/seccodeguide-139067.html#1 [Accessed 11th December 2019].

Confidential Information[46]

All measures should be taken to ensure Confidential Information is not exposed. All Confidential Information should always be read-only and, if the Confidential Information must be readable, limits of exposure must be defined. Abide by the following guidelines when handling Confidential Information:

1. Ensure exception information is sanitised
2. Prevent sensitive information logging
3. Clear sensitive information from memory after use.

Injection and Inclusion[47]

Malicious Code Injection and Malicious Code Inclusion can cause unintended results. Malicious code can be injected into a Java application through an interface allowing the malicious code to run in the context of the application. The following guidelines should be incorporated into Java code development to mitigate injection and inclusion.

1. Ensure Formatting Compliance
2. Exclude Dynamic-SQL
3. Sanitise and Validate XML and HTML Generation
4. Exclude Command-line Untrusted Data
5. Restrict XML Inclusion
6. Exclude BMP Files
7. Disable Swing Components HTML Display
8. Sanitise and Validate Untrusted Code
9. Inhibit Exceptional Floating-Point Value Injection

Accessibility and Extensibility[48]

Reducing the accessibility and extensibility of a Java program will reduce the attack-surface of the program. Employ the following guidelines to restrict accessibility and extensibility.

1. Restrict Class, Interface, Method and Field Access
2. Restrict Package Access
3. All Unrelated Code must be Isolated
4. Restrict ClassLoader Instances Availability
5. Restrict Classes and Methods Extensibility
6. Sanitised and Validate Superclass/Subclass Behaviours

Input Validation

Whenever external data can be inputted into a Java application or program as part of the functioning of that application/program, data input sanitisation and validation must occur.

1. Input Validation
2. Output Validation
3. Only expose Native Java Methods through public Java-based wrapper methods

---

[46] Oracle.com [2019] *Secure Coding Guidelines – 2. Confidential Information* https://www.oracle.com/technetwork/java/seccodeguide-139067.html#2 [Accessed 11th December 2019].
[47] Oracle.com [2019] *Secure Coding Guidelines - 3. Injection and Inclusion* https://www.oracle.com/technetwork/java/seccodeguide-139067.html#3 [Accessed 11th December 2019].
[48] Oracle.com [2019] *Secure Coding Guidelines - 4. Accessibility and Extensibility* https://www.oracle.com/technetwork/java/seccodeguide-139067.html#4 [Accessed 11th December 2019].

Mutability[49]

This is the property of a Java object which cannot change state after construction. [50] Adhere to the following guidelines when dealing with mutability:

1. Value Types should be Immutable
2. Mutable Output Value Copies should be made
3. Mutable and SubClassable Input Value Copies should be made
4. Allow Mutable Class Copy Functionality
5. If trust identity can be overridden by input reference objects, do not trust identity equality
6. Untrusted object output should be considered as passing input
7. Untrusted object input should be considered as output
8. Modifiable internal state should have wrapper methods defined
9. Public static fields should be final
10. Public static final field values should be constants
11. Mutable statics should not be exposed
12. Modifiable collections should not be exposed

Object Construction[51]

Objects in the process of being constructed exist but are not usable. Apply the following guidelines when constructing objects:

1. Sensitive class constructors should not be exposed
2. Unauthorised sensitive class construction should be prohibited
3. Partial initialisation of non-final class instances should be protected
4. Avoid Constructors from calling overridable methods
5. Non-final class cloning should be prevented

Serialisation and Deserialisation[52]

Serialisation is the process of converting objects in memory into byte-streams to store the object on disk or transfer it across a network. Deserialisation is the reverse process; converting byte-streams into an object stored in memory.[53]

Important Note: Deserialisation of untrusted data is fundamentally hazardous as you are storing untrusted data on disk. Deserialisation of untrusted data must be prevented.

1. Prevent Serialisation of security-sensitive classes
2. Protect serialisation of sensitive data
3. Object deserialisation should be treated similar to object construction
4. For serialisation and deserialisation processes, ensure rigorous SecurityManager checks
5. Serialisation and deserialisation security permissions should be clearly understood
6. All untrusted serial data should be sanitised

---

[49] Oracle.com [2019] *Secure Coding Guidelines - 5. Mutability* https://www.oracle.com/technetwork/java/seccodeguide-139067.html#6 [Accessed 11th December 2019].
[50] Oracle.com [2019] *Immutable Objects* https://docs.oracle.com/javase/tutorial/essential/concurrency/immutable.html [Accessed 11th December 2019].
[51] Oracle.com [2019] *Secure Coding Guidelines - 6. Object Construction* https://www.oracle.com/technetwork/java/seccodeguide-139067.html#7 [Accessed 11th December 2019].
[52] Oracle.com [2019] *Secure Coding Guidelines - 7. Serialization and Deserialization* https://www.oracle.com/technetwork/java/seccodeguide-139067.html#8 [Accessed 11th December 2019].
[53] StackOverflow.com [2019] *What is serialization?* https://stackoverflow.com/questions/633402/what-is-serialization [Accessed 11th December 2019].

Access Control[54]

Controlling access to and performing security checks of Java objects is the role of the Java SecurityManager. Controlling object access includes modifying code permissions.

1. Permission checking should be clearly understood
2. Callback methods should be verified
3. Clearly understand java.security.AccessController.doPrivileged class and use it cautiously
4. Clearly understand doPrivileged privilege restrictions
5. Treat all privileged operation caching results with caution
6. Clearly understand context transfer
7. Clearly understand thread construction context transfer
8. Treat API's that bypass SecurityManager checks with caution
9. Immediate caller class loader API instances should be invoked cautiously
10. Java language access checks against immediate callers by API's should be treated cautiously
11. java.lang.reflect.Method.invoke ignores immediate caller checking; prevent Method.invoke usage
12. Interface classes naming convention should not use caller-sensitive method names
13. Prevent returning privileged operations results
14. API's using immediate caller's module for tasks should be invoked cautiously.
15. InvocationHandlers should be used cautiously
16. Module configuration should always be managed with caution

[54] Oracle.com [2019] *Secure Coding Guidelines - 8. Access Control* https://www.oracle.com/technetwork/java/seccodeguide-139067.html#9 [Accessed 11th December 2019].

Java Security – Recent Updates

Java 11 Standard Edition (SE) and Java Development Kit (JDK) Security API's[55]

In September 2018, Oracle released Java 11 SE and JDK which included a number of security related updates to enhance the Java platform. This clearly demonstrates Oracle's commitment to security in and of the Java platform.

Java 11 security related updates of note include:

Library	Description	JDK Ref
security-libs	JEP 324 Key Agreement with Curve25519 and Curve448	8181595
security-libs/java.security	Added Support for PKCS#1 v2.2 Algorithms Including RSASSA-PSS Signature	8146293
security-libs/javax.crypto	ChaCha20 and Poly1305 Cryptographic Algorithms	8153028
security-libs/javax.crypto	RSASSA-PSS Signature Support Added to SunMSCAPI	8205445
security-libs/javax.crypto	Enhanced KeyStore Mechanisms (not public)	8189997
security-libs/javax.net.ssl	JEP 332 Transport Layer Security (TLS) 1.3	8145252

Table 3. Java 11 SE and JDK Security Updates[56]

Java 12 Standard Edition (SE) and Java Development Kit (JDK) Security API's[57]

In March of this year (2019) Oracle released Java 12 SE and JDK which also included a number of security related updates.

Java 12 security related updates of note include:

Library	Description	JDK Ref
security-libs/java.security	groupname Option Added to keytool Key Pair Generation	8213400
security-libs/java.security	Customizing PKCS12 keystore Generation	8076190
security-libs/java.security	Added Additional TeliaSonera Root Certificate	8210432
security-libs/javax.crypto	Change to X25519 and X448 Encoded Private Key Format	8213363
security-libs/javax.net.ssl	ChaCha20 and Poly1305 TLS Cipher Suites	8140466
security-libs/javax.net.ssl	Removed TLS v1 and v1.1 from SSLContext Required Algorithms	8214140
security-libs/javax.net.ssl	Disabled TLS anon and NULL Cipher Suites	8211883
security-libs/javax.net.ssl	Disabled All DES TLS Cipher Suites	8208350
security-libs/org.ietf.jgss:krb5	Support for dns_canonicalize_hostname in krb5.conf	8210821

Table 4. Java 12 SE and JDK Security Updates[58]

[55] Oracle.com [2019] *JDK 11 Release Notes* https://www.oracle.com/technetwork/java/javase/11-relnote-issues-5012449.html [Accessed 11th December 2019].
[56] Taaffe, Jonathon [2019] *Table 3. Java 11 SE and JDK Security Updates* [Created 11th December 2019].
[57] Oracle.com [2019] *JDK 12 Release Notes* https://www.oracle.com/technetwork/java/javase/12-relnote-issues-5211422.html [Accessed 11th December 2019].
[58] Taaffe, Jonathon [2019] *Table 4. Java 12 SE and JDK Security Updates* [Created 11th December 2019].

Java 13 Standard Edition (SE) and Java Development Kit (JDK) Security API's[59]

In September of this year (2019) Oracle released Java 13 SE and JDK which also included a number of security related updates. The Java 13 security related updates of note are included in the following table.

This list of security related updates again demonstrates Oracle's commitment to ensuring the Java Platform is as secure as possible providing for the latest updates in both cryptographic and Secure Socket's Layer (SSL) functionality.

Library	Description	JDK Ref
security-libs/java.security	Configurable Read Timeout for CRLs	8191808
security-libs/java.security	New keytool -showinfo -tls Command for Displaying TLS Configuration Information	8219861
security-libs/javax.crypto	Support for MS Cryptography Next Generation (CNG)	8026953
security-libs/javax.crypto	Use SunJCE Mac in SecretKeyFactory PBKDF2 Implementation	8218723
security-libs/javax.crypto:pkcs11	SunPKCS11 Provider Upgraded with Support for PKCS#11 v2.40	8080462
security-libs/javax.crypto:pkcs11	Memory Growth Issue in SunPKCS11 Fixed	6913047
security-libs/javax.net.ssl	Support for X25519 and X448 in TLS	8171279
security-libs/javax.net.ssl	Session Resumption without Server-Side State in JSSE	8211018
security-libs/javax.net.ssl	Updated the Default Enabled Cipher Suites Preference	8163326
security-libs/javax.net.ssl	Use Server Cipher Suites Preference by Default	8168261
security-libs/javax.security	Allow SASL Mechanisms to Be Restricted	8200400
security-libs/javax.xml.crypto	New String Constants for Canonical XML 1.1 URIs	8224767
security-libs/javax.xml.crypto	[xmldsig] Added KeyValueEC_TYPE	8223053
security-libs/javax.xml.crypto	Updated XML Signature Implementation to Apache Santuario 2.1.3	8219013
security-libs/org.ietf.jgss	Added a Default Native GSS-API Library on Windows	6722928
security-libs/org.ietf.jgss:krb5	Support for Kerberos Cross-Realm Referrals (RFC 6806)	8215032

Table 5. Java 13 SE and JDK Security Updates[60]

[59] Oracle.com [2019] *JDK 13 Release Notes* https://www.oracle.com/technetwork/java/13-relnote-issues-5460548.html [Accessed 11th December 2019].

[60] Taaffe, Jonathon [2019] *Table 5. Java 13 SE and JDK Security Updates* [Created 11th December 2019].

Security of MySQL

According to StackOverFlow Developer Survey Results for 2019[61], MySQL is the most popular Database Technology used by both general users and professional developers. 54% of all respondents stated that MySQL is their number 1 choice of Database technology.

In DB-Engines Database Management Systems (DBMS) ranking results for 2019[62], which includes paid-for and free Open-Source DBMS's, MySQL comes in a close second to Oracle DBMS.

Points of note:

- MySQL is owned by Oracle
- Oracle is a pay-for DBMS
- MySQL is the top free Open-Source DBMS

MySQL Features

To understand why MySQL is the most popular, free, Open-Source DBMS today, a review of MySQL features is required. The following are the core features available with MySQL:

- High Performance and Scalability to meet the demands of exponentially growing data loads and users.
- Self-healing Replication Clusters to improve scalability, performance and availability.
- Online Schema Change to meet changing business requirements.
- Performance Schema for monitoring user- and application-level performance and resource consumption.
- SQL and NoSQL Access for performing complex queries and simple, fast Key Value operations.
- Platform Independence giving you the flexibility to develop and deploy on multiple operating systems.
- Big Data Interoperability using MySQL as the operational data store for Hadoop and Cassandra.

For users and developers this translates into[63]:

- Ease of use: simple install with only a basic knowledge of SQL language required.
- Free Open-Source: MySQL is free to use, downloadable from MySQL.com
- Secure: data security layer provides sensitive data protection and passwords are encrypted.
- Client/Server: RDBMS MySQL server with many clients available, e.g. mysql-client, for connectivity.
- Scalability: capable of +50 million rows with a theoretical file size limit of 8TB
- Platform Interoperability: supported on all major operating systems
- Roll-back: ability to roll-back transactions.
- Performance: storage engine provides exceptional performance
- Application Interoperability: support for many embedded applications
- Productivity: Triggers, stored procedures and views provide for a high level of productivity

[61] StackOverflow.com [2019] *StackOverflow Developer Survey Results 2019* https://insights.stackoverflow.com/survey/2019#technology-_-databases [Accessed 11th December 2019].
[62] DB-Engines.com [2019] *DB-Engines Ranking* https://db-engines.com/en/ranking [Accessed 11th December 2019].
[63] JavaPoint.com [2019] *MySQL Features* https://www.javatpoint.com/mysql-features [Accessed 11th December 2019].

MySQL Technical Specifications

The MySQL Technical Specifications[64] detail the numerous technical features included with MySQL. Below is a summary of those features which are most sought after:

Feature	Details
Flexible Architecture	Open Source, Multi-threaded, Pluggable Storage-Engine
ANSI SQL Standards	ANSI SQL
MySQL Document Store	Relational Tables, JSON Documents, X Protocol, X DevAPI, MySQL Shell
JSON Support	Native JSON Support
Replication & High-Availability	InnoDB Cluster, Built-in Replication Engine providing Master/Slave, Ring, Tree or Row-based Replication
MySQL NDB Cluster	99.999% Availability, Distributed architecture with Synchronous replication
Security	OpenSSL by Default with SQL Roles and Password management
High-Performance	Performance, Information and SYS Schemas, Resource Groups, Partitioning, Optimisation for high concurrency, Read Only and for use with SSD
OLTP and Transactions	ACID Transactions, Commit and Rollback, Row-level Locking, Repeatable Reads and Automatic Deadlock Detection
Manageability and Ease of Use	Easy Install and Setup with a "3 minutes to Success" with all-in-one Windows Installer
Graphical Tools	MySQL Workbench, Data Modeling, Database Administration and SQL Editor for design database functions
Drivers	MySQL Native C Library, MySQL Drivers for ODBC, JDBC, .Net, Python, C, C++ and Community Drivers for PHP, Perl, Python, Ruby, Go

Table 6. MySQL Technical Specifications Summary[65]

Security In MySQL[66]

To ensure a secure MySQL installation, the following considerations must be taken into account:

- **General Security**
  - Employ complex password security using alpha-numeric with special character combinations
  - Principal of least privilege: only grant the required level of access to users
  - Integrating application security to mitigate SQL injections, data corruption, etc.

- **Installation Security:** Secure the core components of the database by ensuring data, log and installation files are only accessible by authorised users

- **Access Control and Account Management:** Securely administer user access to the database contents

- **Security Plugins:** Numerous security focused plugins are available for MySQL providing additional layers of security for Authentication, Connection-Control and Password Validation.

[64] MySQL.com [2019] *MySQL Technical Specifications* https://www.mysql.com/products/enterprise/techspec.html [Accessed 11th December 2019].
[65] Taaffe, Jonathon [2019] *Table 6. MySQL Technical Specifications Summary* [Created 11th December 2019].
[66] MySQL.com [2019] *Chapter 1 Security* https://dev.mysql.com/doc/mysql-security-excerpt/8.0/en/security.html [Accessed 11th December 2019].

Security In MySQL (contd.)

- **Network Security:** Access over the network to MySQL should be restricted to the defined TCP port 3306 for MySQL. MySQL network access can also be restricted to specific IP addresses and remote connectivity can be restricted to only specific database accounts.

- **Backups:** Ensure regular backups of database, configuration and log files are successfully completed. Ensure backups are securely stored and ensure that recovery procedures can be securely completed using the backups.

General Security[67]

The following general security guidelines should be implemented to ensure a secure MySQL installation:

1. Only grant root access to the mysql user table
2. Employ principal of least privilege for user accounts
3. Always encrypt passwords with a hash and salt
4. Employ complex non-dictionary passwords
5. Deploy MySQL behind a firewall
6. All external input data should be sanitised and verified
7. Always encrypt data-in-transit to/from MySQL

Secure Password Management[68]

The following secure password management guidelines should be adhered to, to ensure password security:

End-Users

- Use mysql_config_editor to store authentication credentials in an encrypted login file
- Use --password or -p to interactively and securely enter a password to connect to MySQL
- Store passwords in a .my.cnf option file and restrict access to .my.cnf with chmod.

Administrators

- Only grant access to mysql.user table to administrative accounts
- Set password expiration policies to ensure passwords are regularly changed
- Use validate_password to enforce password policies
- Prevent user access to the MySQL plugin directory
- Protect all log files by restricting access using chmod

[67] MySQL.com [2019] *Security Guidelines* https://dev.mysql.com/doc/refman/8.0/en/security-guidelines.html [Accessed 11th December 2019].
[68] MySQL.com [2019] *Keeping Passwords Secure* https://dev.mysql.com/doc/refman/8.0/en/password-security.html [Accessed 11th December 2019].

Passwords and Logging

- Ensure statement logging is used to prevent passwords being written to log files in plaintext
- Ensure MySQL log files are stored in a restricted directory
- Ensure MySQL audit log files are stored in a restricted directory
- For replication ensure master info repository can only be accessed by administrators
- Use restricted access mode to secure backups

Hardening MySQL[69]

- Ensure all MySQL accounts have valid complex passwords
- Use MySQL compression protocol to encrypt data-in-transit
- Use MySQL SSL support to further encrypt data-in-transit
- mysqld should only be run under a standard user account
- File, Process and Super privileges should only be assigned to administrative accounts
- Use binlog_encryption to encrypt binary and replay log files
- Use --skip-symbolic-links to prevent symlinks to tables
- Ensure stored objects, i.e. programs and views, are securely written
- Do not use wildcards for hostname entries in the grant table
- Use max_user_connections to restrict connections to a specific account
- Using chmod set plugin_dir to read only
- mysql_secure_installation: use this mysql command to secure your MySQL installation as follows
    - Set root account password
    - Remove non-local root accounts
    - Remove anonymous accounts
    - Remote the test database

mysqld Security[70]

The following mysqld commands should be reviewed and enabled or disabled as required

- allow-suspicious-udfs: Enable to prevent loading UDFs (User Defined Functions) from shared object files
- automatic_sp_privileges: Enable to grant execute and alter routing privileges to stored routine creator.
- chroot: Load MySQL into a change root/jailed secure environment
- local_infile: Enable to control server-side local load data statements.
- safe-user-create: Enable to prevent users from creating MySQL accounts using GRANT statement.
- secure_file_priv: Enable to limit data import and export to users with file privilege
- skip-grant-tables: Disable to ensure MySQL loads using the privilege system
- skip_name_resolve: Enable to force MySQL to only use IP addresses when checking client connections.

[69] MySQL.com [2019] *Making MySQL Secure Against Attackers* https://dev.mysql.com/doc/refman/8.0/en/security-against-attack.html [Accessed 11th December 2019].
[70] MySQL.com [2019] *Security-Related mysqld Options and Variables* https://dev.mysql.com/doc/refman/8.0/en/security-options.html [Accessed 11th December 2019].

Run MySQL Normal User[71]

- On Linux, MySQL should never be run as root
- Configure MySQL to run under a non-privileged user account

Security In MySQL - Access Control and Account Management[72]

MySQL provides extensive granular Account and Role management to control account and access management. The following summarises security guidelines for accounts and roles

Account Management

- Ensure all MySQL accounts have complex passwords assigned.
- MySQL user accounts can be configured up to 32 characters
- MySQL account passwords and operating system passwords are separate
- Passwords stored in the user table are encrypted
- Enable non-ASCII character set usage for complex passwords using mysql_options

Account Privileges

- Ensure the principal of least privilege is employed when creating MySQL accounts
- Use MySQL GRANT command to specify specific privileges to accounts
- When creating MySQL accounts, associate user accounts with system hostnames
- Use MySQL roles to grant specific role privileges to user accounts
- Ensure all anonymous accounts are disabled
- Always DROP, i.e. delete, unused MySQL accounts

Password Management

MySQL provides the following password management capabilities

- Expiration
- Reuse restriction
- Verification
- Dual password use
- Password strength assessment
- Random password generation
- Password failure tracking
- Account Locking

Pluggable Authentication

Pluggable authentication enables the following capabilities:

- Authentication Method Choice
- External Authentication
- Proxy User Authentication

[71] MySQL.com [2019] *How to Run MySQL as a Normal User* https://dev.mysql.com/doc/refman/8.0/en/changing-mysql-user.html [Accessed 11th December 2019].
[72] MySQL.com [2019] *Access Control and Account Management* https://dev.mysql.com/doc/refman/8.0/en/access-control.html [Accessed 11th December 2019].

Security In MySQL - Encrypted Connections[73]
To ensure data-in-transit between clients and servers, MySQL can be configured to use encryption capabilities. By default, MySQL will accept encrypted and non-encrypted connections. To enforce encrypted connections, enable require_secure_transport system variable. MySQL provides support for the following data-in-transit encryption options:

- Transport Layer Security (TLS)
- Secure Sockets Layer (SSL) with RSA Certificates and Keys

Transport Layer Security (TLS)
- MySQL supports TLS v1 to TLS v1.3 protocols
- Clients and servers must have an in-common TLS protocol to allow connection
- Clients and servers must have an in-common cryptographic cipher to allow connection
- MySQL supports an extensive cipher suite

Secure Sockets Layer (SSL) with RSA Certificates and Keys
- SSL and RSA certificates and keys can be created by MySQL
- openssl can be used to create SSL and RSA certificates and keys

Security In MySQL - Security Plugins[74]
MySQL includes many security related plugins that can be enabled and configured as required. Below is a summary of the security related plugins and their features:

Authentication Plugins
MySQL plugins are available for the following secure authentication mechanisms:

- Native Authentication
- Authentication System Variables
- SHA-256 Authentication
- Caching SHA-2 Authentication
- LDAP Authentication
- Windows Authentication
- Test Authentication

Connection-Control Plugins
MySQL connection-control plugins allow MySQL administrators to limit connections from clients after a defined number of consecutive failed connection attempts using:

- Connection Delay Configuration
- Connection Failure Assessment
- Connection Failure Monitoring

[73] MySQL.com [2019] *Using Encrypted Connections* https://dev.mysql.com/doc/refman/8.0/en/encrypted-connections.html [Accessed 11th December 2019].
[74] MySQL.com [2019] *Security Components and Plugins* https://dev.mysql.com/doc/refman/8.0/en/security-plugins.html [Accessed 11th December 2019].

Password Validation Component
This MySQL component enforces the following password requirements:

- Password required
- Password strength testing

MySQL Keyring
This MySQL Plugin allows for the secure management of encryption keys locally on the MySQL server. This plugin provides the following features:

- keyring_file File-Based Key Management
- keyring_encrypted_file Key Management
- keyring_okv KMIP Key Management
- keyring_aws Amazon Web Services Key Management
- HashiCorp Vault Key Management
- Keyring Key Migration
- Configurable Keyring System Variables

State of the Art – Latest Programming Paradigms – Conclusion

So far, this report has highlighted the many security features available for both the Java and MySQL technologies. These features should be included in all Java and MySQL production environments to ensure as secure an environment as possible.

State of the Art – Application Security Testing

Software Development Life Cycle (SDLC)

This is an industry standard software development process resulting in producing high quality code and applications, that are of the lowest possible cost, and developed in the shortest time possible. There are 5 distinct phases to an SDLC cycle:

1. Planning and Requirements Gathering
2. Defining Requirements
3. Building
4. Testing
5. Deployment

An SDLC process will include an evaluation of existing applications to identify any shortcomings. Next, a clear definition of the requirements of a new application will be generated. The new application code will then pass through the Plan, Design, Build, Test and Deploy phases.

SDLC Steps[75]

To ensure an effective SDLC process, the following steps should be taken:

1. **Planning and Requirements Gathering:** This is the initial stage of the process and involves either evaluating existing applications to identify shortcomings, or to generate ideas for a new application.
2. **Defining Requirements:** During this phase the application requirements for a new application are gathered. A Software Requirement Specification (SRS) document is used to document all application requirements.
3. **Designing:** This phase takes the software requirements from the planning phase and develops a Design Specification. The design specification includes all functional requirements of the new applications. A Design Document Specification (DDS) document will be used to document the application architecture.
4. **Building:** This is when the actual application code is created and developed.
5. **Testing:** This phase involves testing the new application to ensure it functions as per requirements.
6. **Deployment:** This is a phased process when the new application code is released in a limited fashion to a select group of users for initial User Acceptance Testing (UAT). Once the new application passes UAT the application can be rolled out to more users.
7. **Maintaining:** This phase involves ensuring the new application is functioning optimally and includes all functionality as per the initial application requirements.

SDLC Models

There are a number of SDLC models in use today including:

- **Waterfall:** Finish one phase, then move to the next
- **V-Shaped:** Similar to Waterfall
- **Iterative:** Initial version developed, then tested and improved through rapid successive version upgrades
- **Spiral:** Similar to the iterative model
- **Agile:** Separate the process into cycles, delivering working applications quickly
- **Big Bang:** All resources deployed at the development stage

[75] Stackify.com [2019] *What is SDLC?* https://stackify.com/what-is-sdlc/ [Accessed 11th December 2019].

Development Operations (DevOps)

The change from developers developing applications and then handing over support of the application to the operations teams, to developers and operations teams working together to develop and support applications has greatly enhanced the SDLC process. DevOps is the term used to describe this combining of developers and operations teams into a single support group.

Developers now support applications in production environment, and operations support teams now provide development support for same applications.

Software Assurance Maturity Model (SAMM)

To develop and enhance the SDLC process the Open Web Application Security Project (OWASP)[76] have produced a Software Assurance Maturity Model (SAMM)[77]. SAMM is an open source framework to assist with the development and implementation of a software security strategy. The SAMM framework aids with:

- Evaluating existing software security practices
- Building a balanced software security assurance program
- Demonstrating security assurance program improvements
- Defining and measuring security-related activities

SAMM is based on the core software development functions integrating security practices at each level as follows:

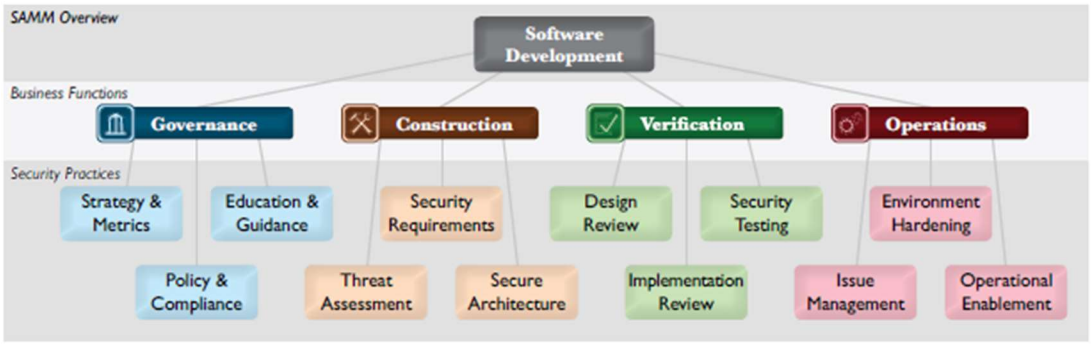

Figure 4. OWASP SAMM Foundation Model [78]

SAMM Business Functions

The SAMM business functions directly relate to the process of software development including

- Governance: Software development management process
- Construction: Software development activities including product management, requirements gathering, architecture specification, design and implementation.
- Verification: Testing phase of programs, applications and/or software
- Operations: Software release management

SAMM Security Practices

The SAMM Business Functions outlined above are then further broken down into the following security practices:

Governance

- Strategy & Metrics: Process of establishing a software security assurance framework
- Policy & Compliance: Attaining legal and regulatory compliance
- Education & Guidance: Providing resources to developers to securely design, develop, and deploy software

[76] OWASP.org [2019] *OWASP™ Foundation. The free and open software security community* https://www.owasp.org/ [Accessed 11th December 2019].
[77] OWASP.org [2019] *OWASP SAMM Project* https://www.owasp.org/index.php/OWASP_SAMM_Project [Accessed 11th December 2019].
[78] OWASP.org [2019] *Figure 4. OWASP SAMM Foundation Model* https://github.com/OWASP/samm/raw/master/Supporting%20Resources/v1.5/Final/SAMM_Core_V1-5_FINAL.pdf [Accessed 11th December 2019].

Construction

- Threat Assessment: Identification and understanding of software and runtime environment functionality risks
- Security Requirements: Analysis of software security behaviour
- Secure Architecture: Focussed on the design and build of secure software

Verification

- Design Review: Software design and architecture assessment focussing on security-related concerns
- Implementation Review: Source code and code configuration analysis to identify security vulnerabilities
- Security Testing: Dynamic software testing in the runtime environment to identify security vulnerabilities

Operations

- Issue Management: Focused on incident management processed
- Environment Hardening: Hardening the software runtime environment
- Operational Enablement: Gathering of software security-related information from developers

Incorporating a robust Software Development Lifecycle (SDLC) process with OWASP's Software Assurance Maturity Model (SAMM) model will ensure developers are provided with the processes, models and frameworks to ensure security is integral to any software development.

Application Security Testing

According to Gartner, Application Security Testing (AST) is the fastest growing information security segment[79]. There are 3 distinct types of Application Security Testing

- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Interactive Application Security Testing (IAST)

In addition to these 3 primary types of AST, there are also

- Runtime Application Self-Protection (RASP): Runtime exploit protection
- Software Composition Analysis (SCA): Identify open-source/3rd-party application components and their known security vulnerabilities
- Mobile Application Security Testing (MAST)
- Application Security Testing as a Service (ASTaaS)

For this report the focus is on SAST, DAST and IAST.

Static Application Security Testing (SAST)

This involves statically analysing application source code, bytecode or binary code to identify security concerns. This type of testing is performed during the application coding phase and the Software Development Lifecycle (SDLC) testing phase. This testing process is a white-box test where application source code and uncompiled code is scanned at rest and then reviewed.

Dynamic Application Security Testing (DAST)

This is the process of running or executing the code and analysing the application code while it is in a running state. This type of testing is carried out during the SDLC testing and operational phases. Malicious attacks against the application can be simulated in a controlled environment to identify application security issues. Dynamic testing of the application will demonstrate how the application code dynamically responds to attack scenarios. This type of testing is considered as black-box testing as the application source code is not readily available to scan. DAST tools using the Fuzzing technique to inject invalid of unexpected test case scenarios at the application.

Interactive Application Security Testing (IAST)

Technology from both the SAST and DAST are used to provide for Interactive Application Security Testing. IAST is normally executed in the applications run time environment. For a Java application, the Java Virtual Machine (JVM) can be configured to assist IAST testing. This type of testing allows for observation of the application in runtime and to simulate attacks against the application code to determine its vulnerability status.

[79] MicroFocus.com [2019] *Gartner 2019 Magic Quadrant for Application Security Testing* https://www.microfocus.com/en-us/assets/security/magic-quadrant-for-application-security-testing [Accessed 11th December 2019].

Application Security Testing Framework

OWASP's Testing Framework[80], provides an extensive list of testing processes that should be carried out when performing Application Security Testing. The following table of OWASP items to be tested relate to security testing a Java application with a MySQL database component:

Area	Item to Test	OWASP Ref
Information Gathering	Identify application entry points	OWASP-IG-003
Information Gathering	Application Discovery	OWASP-IG-005
Information Gathering	Analysis of Error Codes	OWASP-IG-006
Configuration Management	SSL/TLS Testing	OWASP-CM-001
Configuration Management	DB Listener Testing	OWASP-CM-002
Configuration Management	Infrastructure Configuration Management Testing	OWASP-CM-003
Configuration Management	Application Configuration Management Testing	OWASP-CM-004
Configuration Management	Testing for File Extensions Handling	OWASP-CM-005
Configuration Management	Old, Backup and Unreferenced Files	OWASP-CM-006
Configuration Management	Infrastructure and Application Admin Interfaces	OWASP-CM-007
Authentication	Credentials transport over an encrypted channel	OWASP-AT-001
Authentication	Testing for user enumeration	OWASP-AT-002
Authentication	Testing for Guessable User Account Names	OWASP-AT-003
Authentication	Brute Force Testing	OWASP-AT-004
Authentication	Testing for bypassing authentication schema	OWASP-AT-005
Authentication	Testing vulnerable remember password and pwd reset	OWASP-AT-006
Authentication	Testing for Logout Management	OWASP-AT-007
Authentication	Testing for Race Conditions	OWASP-AT-010
Session Management	Testing for Session Management Schema	OWASP-SM-001
Session Management	Testing for Exposed Session Variables	OWASP-SM-004
Authorization	Testing for path traversal	OWASP-AZ-001
Authorization	Testing for bypassing authorization schema	OWASP-AZ-002
Authorization	Testing for Privilege Escalation	OWASP-AZ-003
Business Logic	Business Logic Testing	OWASP-BL-001
Data Validation	Testing for LDAP Injection	OWASP-DV-006
Data Validation	Testing for Code Injection	OWASP-DV-012
Data Validation	Testing for Command Injection	OWASP-DV-013
Data Validation	Testing for Buffer overflow	OWASP-DV-014
Denial of Service	Testing for SQL Wildcard Attacks	OWASP-DS-001
Denial of Service	Testing for DoS Locking Customer Accounts	OWASP-DS-002
Denial of Service	Testing for DoS Buffer Overflows	OWASP-DS-003
Denial of Service	Testing for DoS User Specified Object Allocation	OWASP-DS-004
Denial of Service	Testing for User Input as a Loop Counter	OWASP-DS-005
Denial of Service	Testing for Writing User Provided Data to Disk	OWASP-DS-006
Denial of Service	Testing for DoS Failure to Release Resources	OWASP-DS-007

[80] OWASP.org [2019] *OWASP Testing Guide* https://www.owasp.org/index.php/OWASP_Testing_Guide_v3_Table_of_Contents [Accessed 11th December 2019].

Table 7. OWASP Application Security Testing for Java and MySQL[81]

Application Security Testing Processes

Open Source Security Testing Methodology[82]

This Manual provides a detailed process methodology for Application Security Testing. The primary purpose of this manual is

> *"to provide a scientific methodology for the accurate characterization of operational security (OpSec) through examination and correlation of test results in a consistent and reliable way."*(Herzog, 2016)

The second purpose of this manual is to provide a thorough process to successfully complete an Open Source Security Testing Methodology (OSSTMM) audit of an application and to determine the applications security status.

As per the OSSTMM, there are 7 defined types of Application Security testing as follows:

- Vulnerability Scanning: Automated software vulnerability scanning
- Security Scanning: Manual and Automated network and system weakness scanning
- Penetration Testing: Simulation of a malicious attack
- Risk Assessment: Organisations security risk analysis
- Security Auditing: Applications and Operating System static code analysis
- Ethical Hacking: White hat hacking to expose security vulnerabilities
- Posture Assessment: Combining Security Scanning, Ethical Hacking and Risk Assessments to determine an organisations security posture

For this report the following Application Security Testing types have been identified as relevant to the Java and MySQL testing requirements:

- Vulnerability Scanning: Automated software vulnerability scanning
- Security Scanning: Manual and Automated network and system weakness scanning
- Penetration Testing: Simulation of a malicious attack
- Security Auditing: Applications and Operating System static code analysis

[81] Taaffe, Jonathon [2019] *Table 7. OWASP Application Security Testing for Java and MySQL* [Created 11th December 2019].
[82] ISECOM.org [2019] *The Open Source Security Testing Methodology Manual* https://www.isecom.org/OSSTMM.3.pdf [Accessed 11th December 2019].

Application Security Testing Types

There are many types of application security testing that testers can employ. These testing practices can be divided into Functional and Non-Functional tests. Functional testing is where the actual functionality of the software or application is tested to ensure it functions are per design requirements and specifications. Non-Functional testing relates to testing of Security, Performance, Recovery, Reliability testing.

There are also many different types of software testing types including:

- Password Hacking or Cracking
- Buffer Overflow Testing
- Fuzz Testing
- Penetration Testing
- Vulnerability Scanning
- Security Scanning
- Security Auditing
- Ethical Hacking

Password Hacking or Cracking

This is both a manual and automated process where testers attempt to guess or crack username and password combinations to gain access to a system or software. Once successful a tester can then access restricted areas of the system or software.

Buffer Overflow Testing

A buffer is a finite allocation of memory allocated to an application or program. This buffer is used by an application to store data and respond to user input. As most applications require a buffer to function, this is a common function to be tested. For a buffer overflow to occur, a process or execution needs to occur that adds more data to the buffer than storage allocated to the buffer. When a buffer overflow occurs, the application can and will respond in unexpected ways.[83]

Fuzz Testing

Fuzz testing can be either automated or a manual process to identify errors in software code or security vulnerabilities. Fuzz testing is a black-box test, meaning very little is known about the software code. Fuzz testing requires inputting invalid or unexpected input into a software application to determine vulnerabilities or weaknesses in the software. Fuzz testing is one of the most widely used testing processes as it can identify high risk vulnerabilities.

Fuzz Testing Process

The following steps are required to successfully complete a fuzz test

1. Identify target system or software
2. Identify inputs that can be inputted into the software
3. Generate fuzz data, i.e. the data that will be inputted into the software
4. Execute a fuzz test by inputting the fuzz data into the software
5. Monitoring the system or software to identify any impacts caused by the fuzzing process
6. Log any vulnerabilities detected.

[83] OWASP.org [2019] *Buffer overflow attack* https://www.owasp.org/index.php/Buffer_overflow_attack [Accessed 11th December 2019].

Penetration Testing

This is a simulated attack on a software or system to identify security vulnerabilities. A penetration test attempts to simulate a real-world attack in a controlled process. Penetration testing is also a black-box test where very little of the software code is known to the tester.

There are 5 stages in a penetration test as follows

1. Planning and reconnaissance: identify the target system or software
2. Scanning: automated and manual tools are used to identify details about the system or software
3. Access: testers attempt to access the software to determine vulnerabilities
4. Maintain Access: testers attempt to maintain continual access to the system or software.
5. Results Analysis: a detailed analysis of the results is generated to document software vulnerabilities.

Vulnerability Scanning

Vulnerability scanning is an automated process using a scanning application to scan software for vulnerabilities. This process probes and system or software to identify any weaknesses or vulnerabilities. There are many different types of vulnerability scanners which probe different components of a system or software including

- Port Scanners: scanning for open ports on a system or software.
- Network Enumerators: scanning the system or software to identify and retrieve information about the system or software.
- Web Application Scanner: scanning to identify vulnerabilities within a web application or its infrastructure

Port Scanners

A port in a computing system is an endpoint through which information passes into and out of the system. Ports are required to allow applications communicate with systems and to allow those systems communicate over networks including the internet. Most common internet ports include HTTP port 80 and HTTPS port 443.

A port scanner can be used to probe the ports of a system or software to identify which ports the system or software is listening on for communications. The most common and widely used port scanner is nmap[84]. Nmap is a free open-source network mapping command line tool which can provide extensive information on a system or software configuration and can be used to identify any vulnerabilities in the software or system.

Network Enumerators

Network enumerators probe a systems network configuration to identify its configuration. A network enumerator will also attempt to identify additional configurations of the system including username, account groups, network shares and services running on the system.

[84] Nmap.org [2019] *Nmap Network Mapper* https://nmap.org/ [Accessed 11th December 2019].

One of the most widely used network enumerators is netstat[85]. Netstat is a command line utility which can be used to identify the Transport Control Protocol (TCP)[86] network connectivity of a system or software. This tool can also provide details on routing tables, network interfaces and the network protocol configuration or stack.

Security Scanning

This is an automated the process of scanning a system or software for security specific vulnerabilities. There are a number of automated security scanners available, but the most widely used is Nessus[87] from Tenable[88]. The Nessus automated security scanner scans operating systems, network attached devices, databases and applications. The Nessus scanner will identify system or software vulnerabilities and misconfigurations which could lead to system or software compromise.

Security Auditing

This is a manual process where the security of a system or software is audited for vulnerabilities. Audit items can include all facets of security from physical access to systems, to application access controls.

Ethical Hacking

Ethical hacking simulates a real-world cyber-attack on a system or software, but the ethical hacker does not cause any malicious damage and any vulnerabilities identified are reported to the system or software owner.

[85] Microsoft.com [2019] *netstat* https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/netstat [Accessed 11th December 2019].
[86] Wikipedia.org [2019] *Transmission Control Protocol* https://en.wikipedia.org/wiki/Transmission_Control_Protocol [Accessed 11th December 2019].
[87] Tenable.com [2019] *Tenable Nessus* https://www.tenable.com/products/nessus [Accessed 11th December 2019].
[88] Tenable.com [2019] *Tenable.com* https://www.tenable.com/ [Accessed 11th December 2019].

Secure Application Development – Java Application Testing

Project Configuration

To analyse the Java Project Code, the following setup was used:

1. Installed Oracle JDK 11[89]
2. Installed Eclipse IDE 2019-09 Windows 64-bit Integrated Development Environment (IDE)[90]
3. Imported the Java trainingProject.zip Project into Eclipse

   File -> Import -> General -> Projects from Folder or Archive -> Next -> Import Source -> Archive: trainingProject.zip_expanded -> Finish

4. Installed MySQL Developer 8.0.18[91]
5. Installed Connector/J 8.0.18
6. Imported the Java Project Database using MySQL WorkBench

   MySQL Workbench->Local Instance MySQL80->Server->Data Import->Import from Dump Project Folder -> securityProject\SQL ->Load Folder Contents->library->Start Import

7. Configured DB.java to used MySQL username and password

Java Application Functional Test

Account Login

User Login: Successfully logged into the Java application from the main library interface MainLibrary.form using the credentials provided.

Librarian Login: Successfully logged into the Java application from the main library interface MainLibrary.form using the credentials provided.

Add User Account

After successfully logging into the Java application using the Librarian account, adding a new user to the application was successful.

Issue a Book

After successfully logging into the Java application using the Librarian account, issuing a book to a student was successful.

Add Book

After successfully logging into the Java application using the Librarian account, adding a book failed receiving the error 'The Book is not added!'.

---

[89] Oracle.com [2019] *Java SE Development Kit 11 Downloads* https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html [Accessed 11th December 2019].
[90] Eclipse.org [2019] *Eclipse Foundation Download Eclipse Technology that is right for you* https://www.eclipse.org/downloads/ [Accessed 11th December 2019].
[91] MySQL.com [2019] *MySQL Community Downloads* https://dev.mysql.com/downloads/windows/installer/8.0.html [Accessed 11th December 2019].

Java Code Analysis

Visual Code Grepper (VCG)[92]

With the Java application functioning analysis of the Java code could start. The first tool chosen was Visual Code Grepper (VCG) which is a command line tool that analyses Java code for issues.

In total 154 lines of Java code were listed as incorrect. Below is a summary of the VCG findings from severity high to low:

PTY	Sev.	Title	Description
7	Potential Issue	Public Class Not Declared as Final	The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.
6	Suspicious Comment	Comment Indicates Potentially Unfinished Code	The comment includes some wording which indicates that the developer regards it as unfinished or does not trust it to work correctly.
5	Low	Operation on Primitive Data Type	The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

Table 8. Visual Code Grepper Summary Results[93]

The Java files in which errors were reported include:

About.java	LibrarianDao.java	UserForm.java
AllStudent.java	LibrarianLogin.java	UserLogin.java
BookDao.java	LibrarianSuccess.java	UserLoginSuccess.java
BookForm.java	MainLibrary.java	UsersDao.java
DB.java	NewView.java	UserView.java
DeleteBook.java	ReturnBookForm.java	UserViewBook.java
IssueBookForm.java	TransBookDao.java	ViewBook.java

[92] SourceForge.net [2019] *Visual Code Grepper* (VCG) https://sourceforge.net/projects/visualcodegrepp/ [Accessed 11th December 2019].

[93] Taaffe, Jonathon [2019] *Table 8. Visual Code Grepper Summary Results* [Created 11th December 2019].

JArchitect Java Static Analysis Tool[94]

Using the JArchitect Java Static Analysis Tool the following results were reported for the Java project

Application Metrics

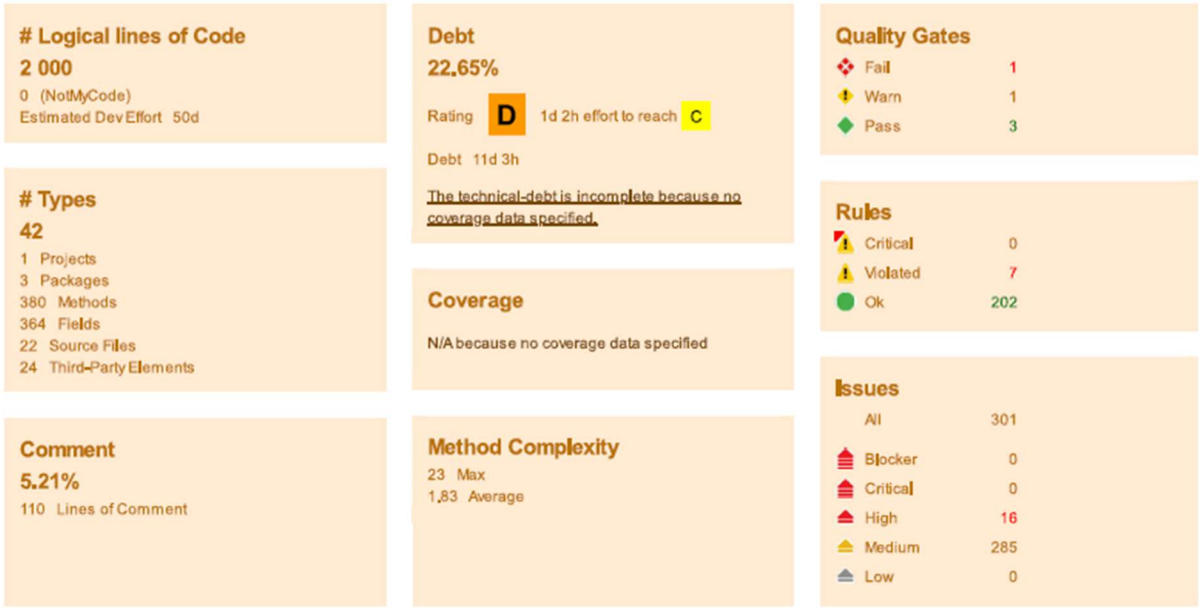

Figure 5. JArchitect Java Static Analysis Tool Application Metrics[95]

Rules Summary

The following rules were violated

Name	# Issues	Elements	Group
Avoid types with too many fields	6	types	Project Rules \ Code Smell
Avoid methods with too many local variables	8	methods	Project Rules \ Code Smell
Avoid methods potentially poorly commented	14	methods	Project Rules \ Code Smell
Avoid types with poor cohesion	14	types	Project Rules \ Code Smell
Instance fields should begin with a lower character	108	fields	Project Rules \ Naming Conventions
Methods name should begin with an lower character	130	methods	Project Rules \ Naming Conventions
Avoid having different types with same name	21	types	Project Rules \ Naming Conventions

Figure 6. JArchitect Java Static Analysis Tool Rules Summary[96]

[94] Jarchitect.com [2019] *Jarchitect.com Discover amazing details about your Java projects* https://www.jarchitect.com/ [Accessed 11th December 2019].

[95] Taaffe, Jonathon [2019] *Figure 5. JArchitect Java Static Analysis Tool Application Metrics* [Created 11th December 2019].

[96] Taaffe, Jonathon [2019] *Figure 6. JArchitect Java Static Analysis Tool Rules Summary* [Created 11th December 2019].

Kiuwan Static Code Analysis[97]

Using the Kiuwan Java Static Code Analysis Tool the following results were reported for the Java project

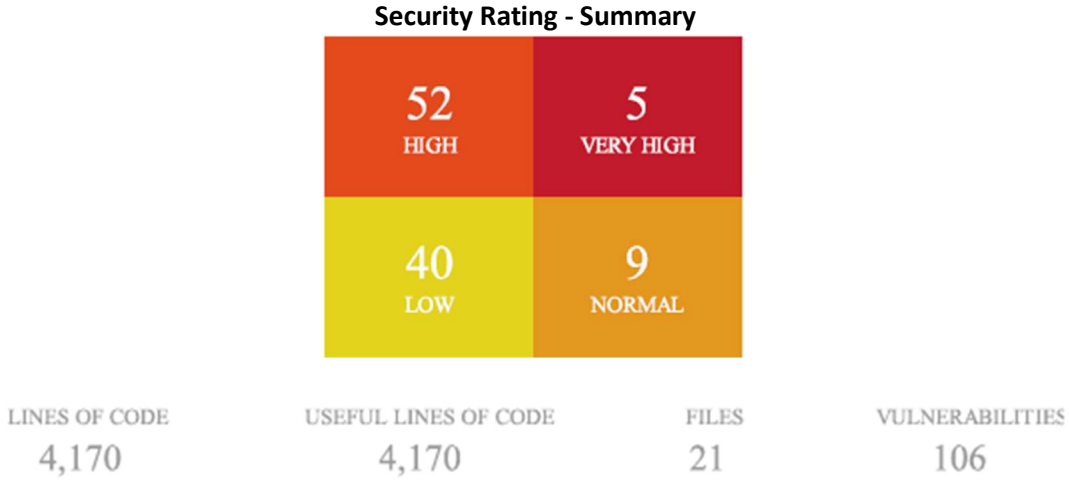

Figure 7. Kiuwan Static Code Analysis Security Rating - Summary[98]

Figure 8. Kiuwan Static Code Analysis Top 3 Vulnerabilities by Type[99]

Figure 9. Kiuwan Static Code Analysis Top 3 Worst Files[100]

[97] Kiuwan.com [2019] *Kiuwan Security Solutions for your DevOps Process* https://www.kiuwan.com/ [Accessed 11th December 2019].

[98] Taaffe, Jonathon [2019] *Figure 7. Kiuwan Static Code Analysis Security Rating - Summary* [Created 11th December 2019].

[99] Taaffe, Jonathon [2019] *Figure 8. Kiuwan Static Code Analysis Top 3 Vulnerabilities by Type* [Created 11th December 2019].

[100] Taaffe, Jonathon [2019] *Figure 9. Kiuwan Static Code Analysis Top 3 Worst Files* [Created 11th December 2019].

PMD Source Code Analyser[101]

PMD Source Code Analyser identified 668 lines of code that there were issues with. A summary of the results are as follows

Priority	Rule set	Rule
4	Code Style	UnnecessaryFullyQualifiedName
4	Best Practices	UnusedImports
4	Best Practices	OneDeclarationPerLine
3	Best Practices	UnusedFormalParameter
3	Code Style	IdenticalCatchBranches
3	Design	SingularField
3	Error Prone	CloseResource
3	Code Style	UselessQualifiedThis
3	Best Practices	UnusedPrivateField
3	Design	UseUtilityClass
3	Best Practices	CheckResultSet
3	Error Prone	EmptyIfStmt
3	Error Prone	EmptyStatementNotInLoop
3	Best Practices	UnusedPrivateMethod
3	Best Practices	UnusedLocalVariable
3	Error Prone	UseEqualsToCompareStrings
1	Code Style	LocalVariableNamingConventions
1	Code Style	MethodNamingConventions
1	Code Style	ClassNamingConventions
1	Code Style	FormalParameterNamingConventions

Table 9. PDM Summary Results[102]

[101] PMD.github.io [2019] *PMD Source Code Analyser* https://pmd.github.io/ [Accessed 11th December 2019].
[102] Taaffe, Jonathon [2019] *Table 9. PDM Summary Results* [Created 11th December 2019].

Xanitizer Static Code Analysis Tool[103]

Xanitizer Static Code Analysis Tool identified issues with the mysql-connector-java-5.1.45-bin.jar file as follows:

Finding ID:	1	**Problem Type**	OWASP Dependency Check: Libraries with known vulnerabilities (OWASP Dependency Check)
Classification:	Warning	**First detected at:**	2019-11-11 13:04:49 - new
Rating:	6.50	**Reviewed State:**	Not Reviewed
Location:	mysql-connector-java-5.1.45-bin.jar (C:/Users/jonat/OneDrive/Cyber Security/NCI - Post Graduate Diploma/Course/S3.2 AppSecProg/Project/securityProject)		
Tags:	no tags assigned		
Comment:	*no comment*		
Description:	A finding identified by the external 'OWASP Dependency Check' plugin: mysql-connector-java-5.1.45-bin.jar: CVE-2018-3258 National Vulnerability Database of NIST reports: Vulnerability in the MySQL Connectors component of Oracle MySQL (subcomponent: Connector/J). Supported versions that are affected are 8.0.12 and prior. Easily exploitable vulnerability allows low privileged attacker with network access via multiple protocols to compromise MySQL Connectors. Successful attacks of this vulnerability can result in takeover of MySQL Connectors. CVSS 3.0 Base Score 8.8 (Confidentiality, Integrity and Availability impacts). CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H). The vulnerability in the library is assigned by the National Vulnerability Database of NIST to the CWE: NVD-CWE-noinfo The confidence level 'HIGH' based on the following CPE: cpe:2.3:a:oracle:connector\/j:* up to version 8.0.12		
Problem Type Description:	Jar file has been checked by OWASP Dependency Check tool See CWE number 1035 for details.		

Figure 10. Xanitizer Static Code Analysis Tool Finding 1[104]

Finding ID:	2	**Problem Type**	OWASP Dependency Check: Libraries with known vulnerabilities (OWASP Dependency Check)
Classification:	Warning	**First detected at:**	2019-11-11 13:04:49 - new
Rating:	3.50	**Reviewed State:**	Not Reviewed
Location:	mysql-connector-java-5.1.45-bin.jar (C:/Users/jonat/OneDrive/Cyber Security/NCI - Post Graduate Diploma/Course/S3.2 AppSecProg/Project/securityProject)		
Tags:	no tags assigned		
Comment:	*no comment*		
Description:	A finding identified by the external 'OWASP Dependency Check' plugin: mysql-connector-java-5.1.45-bin.jar: CVE-2019-2692 National Vulnerability Database of NIST reports: Vulnerability in the MySQL Connectors component of Oracle MySQL (subcomponent: Connector/J). Supported versions that are affected are 8.0.15 and prior. Difficult to exploit vulnerability allows high privileged attacker with logon to the infrastructure where MySQL Connectors executes to compromise MySQL Connectors. Successful attacks require human interaction from a person other than the attacker. Successful attacks of this vulnerability can result in takeover of MySQL Connectors. CVSS 3.0 Base Score 6.3 (Confidentiality, Integrity and Availability impacts). CVSS Vector: (CVSS:3.0/AV:L/AC:H/PR:H/UI:R/S:U/C:H/I:H/A:H). The vulnerability in the library is assigned by the National Vulnerability Database of NIST to the CWE: CWE-20 The confidence level 'HIGH' based on the following CPE: cpe:2.3:a:oracle:mysql_connector\/j:* up to version 8.0.15		
Problem Type Description:	Jar file has been checked by OWASP Dependency Check tool See CWE number 1035 for details.		

Figure 11. Xanitizer Static Code Analysis Tool Finding 2[105]

[103] Xanitizer.com [2019] *Xanitizer Static Code Analysis Tool* https://www.xanitizer.com/xanitizer/ [Accessed 11th December 2019].

[104] Taaffe, Jonathon [2019] *Figure 10. Xanitizer Static Code Analysis Tool Finding 1* [Created 11th December 2019].

[105] Taaffe, Jonathon [2019] *Figure 11. Xanitizer Static Code Analysis Tool Finding 2* [Created 11th December 2019].

Secure Application Development – Proposed Solutions

Java Code Security Issues Proposed Solutions

Issue: Public Class Not Declared as Final

If classes are not declared as final, those classes can be extended which can change the behaviour of the class.[106]

The following table lists the Java files, associated line number and code in which a public class is not declared as final.

File Name	Line	Code
About.java	12	public class About extends javax.swing.JFrame {
AllStudent.java	23	public class AllStudent extends javax.swing.JFrame {
BookDao.java	7	public class BookDao {
BookForm.java	17	public class BookForm extends javax.swing.JFrame {
DB.java	17	public class DB {
DeleteBook.java	14	public class DeleteBook extends javax.swing.JFrame {
IssueBookForm.java	17	public class IssueBookForm extends javax.swing.JFrame {
LibrarianDao.java	5	public class LibrarianDao {
LibrarianLogin.java	14	public class LibrarianLogin extends javax.swing.JFrame {
LibrarianSuccess.java	18	public class LibrarianSuccess extends javax.swing.JFrame {
MainLibrary.java	12	public class MainLibrary extends javax.swing.JFrame {
NewView.java	26	public class NewView extends javax.swing.JFrame {
ReturnBookForm.java	15	public class ReturnBookForm extends javax.swing.JFrame {
TransBookDao.java	6	public class TransBookDao {
UserForm.java	15	public class UserForm extends javax.swing.JFrame {
UserLogin.java	14	public class UserLogin extends javax.swing.JFrame {
UserLoginSuccess.java	16	public class UserLoginSuccess extends javax.swing.JFrame {
UsersDao.java	17	public class UsersDao {
UserView.java	25	public class UserView extends javax.swing.JFrame {
UserViewBook.java	26	public class UserViewBook extends javax.swing.JFrame {
ViewBook.java	26	public class ViewBook extends javax.swing.JFrame {

Table 10. Java Files and Associated Code with Public Class Not Declared as Final[107]

[106] Oracle.com [2019] *Writing Final Classes and Methods* https://docs.oracle.com/javase/tutorial/java/IandI/final.html [Accessed 11th December 2019].

[107] Taaffe, Jonathon [2019] *Table 10. Java Files and Associated Code with Public Class Not Declared as Final* [Created 11th December 2019].

Solution: Declare the Public Class as Final

Modify the following lines of code to declare the public class as final as follows

File Name	Line	Code
About.java	12	public **final** class About extends javax.swing.JFrame {
AllStudent.java	23	public **final** class AllStudent extends javax.swing.JFrame {
BookDao.java	7	public **final** class BookDao {
BookForm.java	17	public **final** class BookForm extends javax.swing.JFrame {
DB.java	17	public **final** class DB {
DeleteBook.java	14	public **final** class DeleteBook extends javax.swing.JFrame {
IssueBookForm.java	17	public **final** class IssueBookForm extends javax.swing.JFrame {
LibrarianDao.java	5	public **final** class LibrarianDao {
LibrarianLogin.java	14	public **final** class LibrarianLogin extends javax.swing.JFrame {
LibrarianSuccess.java	18	public **final** class LibrarianSuccess extends javax.swing.JFrame {
MainLibrary.java	12	public **final** class MainLibrary extends javax.swing.JFrame {
NewView.java	26	public **final** class NewView extends javax.swing.JFrame {
ReturnBookForm.java	15	public **final** class ReturnBookForm extends javax.swing.JFrame {
TransBookDao.java	6	public **final** class TransBookDao {
UserForm.java	15	public **final** class UserForm extends javax.swing.JFrame {
UserLogin.java	14	public **final** class UserLogin extends javax.swing.JFrame {
UserLoginSuccess.java	16	public **final** class UserLoginSuccess extends javax.swing.JFrame {
UsersDao.java	17	public **final** class UsersDao {
UserView.java	25	public **final** class UserView extends javax.swing.JFrame {
UserViewBook.java	26	public **final** class UserViewBook extends javax.swing.JFrame {
ViewBook.java	26	public **final** class ViewBook extends javax.swing.JFrame {

Table 11. Java Files and Associated Code Updated with Public Class Declared as Final[108]

[108] Taaffe, Jonathon [2019] *Table 11. Java Files and Associated Code Updated with Public Class Declared as Final* [Created 11th December 2019].

Issue: Operation on Primitive Data Type

The code appears to be carrying out a mathematical operation on a primitive data type. In some situations, this can result in an overflow and unexpected behaviour.

In Java all variables must be declared before they can be used. The data type assigned to a variable will determine the data or values it can hold as well as the operations that can be performed against it. In total Java has 8 Primitive Data Types[109]:

- byte: byte data type used for memory conservation and for limit clarifications
- short: short data type used for used for memory conservation
- int: int data type used for integers
- long: long data type used for value ranges longer than int can support
- float: float data type for memory conservation
- double: double data type used for decimal values
- boolean: boolean data type used for true and false statements
- char: char data type is used for characters

The following table lists the Java files, associated line number and code in which the code appears to be carrying out a mathematical operation on a primitive data type.

File Name	Line	Code
AllStudent.java	48	for (int i = 1; i <= colnum; i++) {
AllStudent.java	49	Row[i - 1] = rs.getString(i);
AllStudent.java	264	for (int i = 1; i <= colnum; i++) {
AllStudent.java	265	Row[i - 1] = rs.getString(i);
AllStudent.java	305	for (int i = 1; i <= colnum; i++) {
AllStudent.java	306	Row[i - 1] = rs.getString(i);
AllStudent.java	364	for (int i = 1; i <= colnum; i++) {
AllStudent.java	365	Row[i - 1] = rs.getString(i);
NewView.java	64	for (int i = 1; i <= colnum; i++) {
NewView.java	65	Row[i - 1] = rs.getString(i);
NewView.java	267	for (int i = 1; i <= colnum; i++) {
NewView.java	268	Row[i - 1] = rs.getString(i);
NewView.java	313	for (int i = 1; i <= colnum; i++) {
NewView.java	314	Row[i - 1] = rs.getString(i);
NewView.java	359	for (int i = 1; i <= colnum; i++) {
NewView.java	360	Row[i - 1] = rs.getString(i);
NewView.java	437	for (int i = 1; i <= colnum; i++) {
NewView.java	438	Row[i - 1] = rs.getString(i);

Table 12. Java Files and Associated Code with Operation on Primitive Data Type[110]

[109] Oracle.com [2019] *Primitive Data Types* https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html [Accessed 11th December 2019].
[110] Taaffe, Jonathon [2019] *Table 12. Java Files and Associated Code with Operation on Primitive Data Type* [Created 11th December 2019].

File Name	Line	Code
TransBookDao.java	67	ps2.setInt(1, quantity - 1);
UserView.java	68	for (int i = 1; i <= colnum; i++) {
UserView.java	69	Row[i - 1] = rs.getString(i);
UserViewBook.java	65	for (int i = 1; i <= colnum; i++) {
UserViewBook.java	68	Row[i - 1] = "Not Issued";
UserViewBook.java	70	Row[i - 1] = "Issued";
UserViewBook.java	74	Row[i - 1] = rs.getString(i);
UserViewBook.java	294	for (int i = 1; i <= colnum; i++) {
UserViewBook.java	297	Row[i - 1] = "Not Issued";
UserViewBook.java	303	Row[i - 1] = "Issued";
UserViewBook.java	310	Row[i - 1] = rs.getString(i);
UserViewBook.java	355	for (int i = 1; i <= colnum; i++) {
UserViewBook.java	358	Row[i - 1] = "Not Issued";
UserViewBook.java	364	Row[i - 1] = "Issued";
UserViewBook.java	369	Row[i - 1] = rs.getString(i);
UserViewBook.java	470	for (int i = 1; i <= colnum; i++) {
UserViewBook.java	473	Row[i - 1] = "Not Issued";
UserViewBook.java	479	Row[i - 1] = "Issued";
UserViewBook.java	484	Row[i - 1] = rs.getString(i);
ViewBook.java	65	for (int i = 1; i <= colnum; i++) {
ViewBook.java	66	Row[i - 1] = rs.getString(i);
ViewBook.java	282	for (int i = 1; i <= colnum; i++) {
ViewBook.java	283	Row[i - 1] = rs.getString(i);
ViewBook.java	326	for (int i = 1; i <= colnum; i++) {
ViewBook.java	327	Row[i - 1] = rs.getString(i);
ViewBook.java	413	for (int i = 1; i <= colnum; i++) {
ViewBook.java	414	Row[i - 1] = rs.getString(i);

Table 12. Java Files and Associated Code with Operation on Primitive Data Type[110]

Solution: Manually review the code to determine the risk.

Java ARchive File (JAR) Security Issues Proposed Solutions

Vulnerability in the MySQL Connectors component of Oracle MySQL

File Name: mysql-connector-java-5.1.45-bin.jar

As per CVE-2018-3258[111], MySQL J connector components are vulnerable to MySQL connector takeover.

CVSS Scores and Vulnerability Types
Below is the CVSS Scores and Vulnerability Types for this vulnerability

– CVSS Scores & Vulnerability Types

CVSS Score	6.5
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Single system (The vulnerability requires an attacker to be logged into the system (such as at a command line or via a desktop session or web interface).)
Gained Access	None
Vulnerability Type(s)	
CWE ID	CWE id is not defined for this vulnerability

Figure 12. MySQL J Connector CVSS Scores and Vulnerability Types[112]

Solution: As per Oracle Critical Patch Update Advisory – October 2018[113], ensure this critical patch update is applied to remediate this vulnerability.

[111] CVEDetails.com [2019] *Vulnerability Details : CVE-2018-3258* https://www.cvedetails.com/cve/CVE-2018-3258/ [Accessed 11th December 2019].
[112] Taaffe, Jonathon [2019] *Figure 12. MySQL J Connector CVSS Scores and Vulnerability Types* [Created 11th December 2019].
[113] Oracle.com [2019] *Oracle Critical Patch Update Advisory - October 2018* https://www.oracle.com/security-alerts/cpuoct2018.html [Accessed 11th December 2019].

MySQL Security Related Issues

The following MySQL security related issues were also found with this project

Hard Coding of Passwords

For both the Librarian and User accounts the associated account passwords are stored in plaintext within the MySQL database.

Solution: All passwords should be securely stored using hashing and salting encryption

SQL Injection

The Java application login screen is prone to SQL injection attack

Simply entering ' or 1=1# for the username field in the librarian login, allows you to login as a Librarian.

Solution: Use parameterised queries

Java Static Code Analysis

Summary

After completing a file by file, and line by analysis of the src\mainlibrary*.java files, multiple coding issues were identified. In total 199 classes with code containing issues were identified. The following table is a summary of these coding issues and their proposed solutions:

Type	Description	Freq
Issue	Empty IF statement	1
Solution	If statements should never be left without a value	
Issue	Unused local variable	5
Solution	Never declare a local variable if it is not going to be used	
Issue	Unused Private Method	2
Solution	Never declare a Private Method if it is not going to be used	
Issue	Unused imported classes	17
Solution	Only import classes that you are going to use	
Issue	Parameters passed to methods must be referenced in the method body	89
Solution	Do not declare a private field or assign a value to the private field if it is not being used	
Issue	Do not use '==' or '!=' to compare strings	1
Solution	Recommend using equals () method	
Issue	Resources not closed	83
Solution	Resources should always be closed after use	

Table 13. Summary of Java Static Code Analysis[114]

[114] Taaffe, Jonathon [2019] *Table 13. Summary of Java Static Code Analysis* [Created 11th December 2019].

Detailed Analysis

The following section includes the results of a file-by-file and line-by-line detailed analysis. The analysis results include the following data

- **.java File:** Name of the relevant .java file
- **Line:** Line number of reference code
- **Class:** Class in which the referenced code is encapsulated
- **Code:** Line of code
- **Issue:** Summary description of coding issue

.java File	Line	Class	Code	Issue
About	89	public class About extends javax.swing.JFrame {	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
AllStudent	38	public class AllStudent extends javax.swing.JFrame {	PreparedStatement ps = Con.prepareStatement("select * from Users", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
AllStudent	39	public class AllStudent extends javax.swing.JFrame {	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
AllStudent	224	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
AllStudent	229	private void SearchFieldActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchFieldActionPerformed	private void SearchFieldActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchFieldActionPerformed	Parameters passed to methods must be referenced in the method body
AllStudent	233	private void NameRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_NameRadioActionPerformed	private void NameRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_NameRadioActionPerformed	Parameters passed to methods must be referenced in the method body
AllStudent	239	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	Parameters passed to methods must be referenced in the method body
AllStudent	252	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	PreparedStatement ps = Con.prepareStatement("select * from Users where UserName like ?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use

AllStudent	254	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
AllStudent	293	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	PreparedStatement ps = Con.prepareStatement("select * from Users where Email like ?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
AllStudent	295	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
AllStudent	336	private void AuthorRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_AuthorRadioActionPerformed	private void AuthorRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_AuthorRadioActionPerformed	Parameters passed to methods must be referenced in the method body
AllStudent	342	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	Parameters passed to methods must be referenced in the method body
AllStudent	354	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	PreparedStatement ps = Con.prepareStatement("select * from Users", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
AllStudent	355	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
AllStudent	422	public static void main(String args[]) {	private javax.swing.ButtonGroup buttonGroup3;	Do not declare a private field or assign a value to the private field if it is not being used
BookDao	11	public static int save(String callno,String name,String author,String publisher,int quantity){	Connection con=DB.getConnection();	Resources should always be closed after use
BookDao	12	public static int save(String callno,String name,String author,String publisher,int quantity){	PreparedStatement ps=con.prepareStatement("insert into books(callno,name,author,publisher,quantity) values(?,?,?,?,?)");	Resources should always be closed after use
BookDao	33	public static boolean PublisherValidate(String Publisher)	PreparedStatement ps = con.prepareStatement("select * from Publisher where PublisherName = ?");	Resources should always be closed after use
BookDao	35	public static boolean PublisherValidate(String Publisher)	ResultSet rs=ps.executeQuery();	Resources should always be closed after use
BookDao	46	public static int AddPublisher(String Publisher)	PreparedStatement ps=con.prepareStatement("insert into Publisher(PublisherName) values(?)");	Resources should always be closed after use

BookDao	59	public static int SaveBook(String BookN, String AuthorN, String PublisherN, String ShelfN, String RowN, String GenreN) {	PreparedStatement ps=con.prepareStatement("insert into Books(BookName,Author,Genre,Publisher,Shelf, Row) values(?,?,?,?,?,?)");	Resources should always be closed after use
BookDao	75	public static int Delete(int BookID)	PreparedStatement ps=con.prepareStatement("DELETE FROM Books where BookID=?");	Resources should always be closed after use
BookForm	211	private void BookNameActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BookNameActionPerformed	private void BookNameActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BookNameActionPerformed	Parameters passed to methods must be referenced in the method body
BookForm	215	private void PublisherActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_PublisherActionPerformed	private void PublisherActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_PublisherActionPerformed	Parameters passed to methods must be referenced in the method body
BookForm	219	private void AuthorActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_AuthorActionPerformed	private void AuthorActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_AuthorActionPerformed	Parameters passed to methods must be referenced in the method body
BookForm	223	private void RowActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_RowActionPerformed	private void RowActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_RowActionPerformed	Parameters passed to methods must be referenced in the method body
BookForm	227	private void ShelfActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ShelfActionPerformed	private void ShelfActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ShelfActionPerformed	Parameters passed to methods must be referenced in the method body
BookForm	231	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	Parameters passed to methods must be referenced in the method body
BookForm	237	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
BookForm	246	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	if (BookDao.PublisherValidate(PublisherN)) {	If statements should never be left without a value
BookForm	267	private void GenreActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_GenreActionPerformed	private void GenreActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_GenreActionPerformed	Parameters passed to methods must be referenced in the method body
BookForm	8	import java.awt.Frame;	import java.awt.Frame;	Only import classes that you are going to use

BookForm	9	import javax.swing.JFrame;	import javax.swing.JFrame;	Only import classes that you are going to use
DB	10	import java.sql.SQLException;	import java.sql.SQLException;	Only import classes that you are going to use
DeleteBook	122	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
DeleteBook	143	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	Parameters passed to methods must be referenced in the method body
DeleteBook	149	private void UserNameActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_UserNameActionPerformed	private void UserNameActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_UserNameActionPerformed	Parameters passed to methods must be referenced in the method body
IssueBookForm	241	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
IssueBookForm	282	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	Parameters passed to methods must be referenced in the method body
IssueBookForm	288	private void UserIDActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_UserIDActionPerformed	private void UserIDActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_UserIDActionPerformed	Parameters passed to methods must be referenced in the method body
IssueBookForm	292	private void IDateActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_IDateActionPerformed	private void IDateActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_IDateActionPerformed	Parameters passed to methods must be referenced in the method body
IssueBookForm	296	private void IMonthActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_IMonthActionPerformed	private void IMonthActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_IMonthActionPerformed	Parameters passed to methods must be referenced in the method body
IssueBookForm	300	private void RYearActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_RYearActionPerformed	private void RYearActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_RYearActionPerformed	Parameters passed to methods must be referenced in the method body
IssueBookForm	304	private void BookIDActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BookIDActionPerformed	private void BookIDActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BookIDActionPerformed	Parameters passed to methods must be referenced in the method body
IssueBookForm	9	import java.util.Date;	import java.util.Date;	Only import classes that you are going to use

LibrarianDao	11	public static int save(String name, String password, String email, String address, String city, String contact) {	Connection con = DB.getConnection();	Resources should always be closed after use
LibrarianDao	12	public static int save(String name, String password, String email, String address, String city, String contact) {	PreparedStatement ps = con.prepareStatement("insert into librarian(name,password,email,address,city,contact) values(?,?,?,?,?,?)");	Resources should always be closed after use
LibrarianDao	30	public static int delete(int id) {	Connection con = DB.getConnection();	Resources should always be closed after use
LibrarianDao	31	public static int delete(int id) {	PreparedStatement ps = con.prepareStatement("delete from Librarian where id=?");	Resources should always be closed after use
LibrarianDao	44	public static boolean validate(String name, String password) {	Connection con = DB.getConnection();	Resources should always be closed after use
LibrarianDao	46	public static boolean validate(String name, String password) {	Statement selectStatement = con.createStatement();	Resources should always be closed after use
LibrarianDao	47	public static boolean validate(String name, String password) {	ResultSet rs = selectStatement.executeQuery(select);	Resources should always be closed after use
LibrarianLogin	16	private static void setvisible(boolean b) {	private static void setvisible(boolean b) {	Parameters passed to methods must be referenced in the method body
LibrarianLogin	16	private static void setvisible(boolean b) {	private static void setvisible(boolean b) {	Never declare a Private Method if it is not going to be used
LibrarianLogin	157	private void usernameActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_usernameActionPerformed	private void usernameActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_usernameActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianLogin	161	private void passwordActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_passwordActionPerformed	private void passwordActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_passwordActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianLogin	165	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianLogin	185	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianSuccess	274	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body

LibrarianSuccess	280	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianSuccess	286	private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton3ActionPerformed	private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton3ActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianSuccess	292	private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton4ActionPerformed	private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton4ActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianSuccess	298	private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton5ActionPerformed	private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton5ActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianSuccess	304	private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton7ActionPerformed	private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton7ActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianSuccess	310	private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton8ActionPerformed	private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton8ActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianSuccess	316	private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jTextField1ActionPerformed	private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jTextField1ActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianSuccess	320	private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jTextField2ActionPerformed	private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jTextField2ActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianSuccess	324	private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jTextField3ActionPerformed	private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jTextField3ActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianSuccess	328	private void jTextField1MousePressed(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_jTextField1MousePressed	private void jTextField1MousePressed(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_jTextField1MousePressed	Parameters passed to methods must be referenced in the method body
LibrarianSuccess	332	private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton9ActionPerformed	private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton9ActionPerformed	Parameters passed to methods must be referenced in the method body

LibrarianSuccess	337	private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton6ActionPerformed	private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton6ActionPerformed	Parameters passed to methods must be referenced in the method body
LibrarianSuccess	380	public void run() {	Connection Con;	Resources should always be closed after use
LibrarianSuccess	382	public void run() {	PreparedStatement ps;	Resources should always be closed after use
LibrarianSuccess	386	public void run() {	ResultSet rs;	Resources should always be closed after use
LibrarianSuccess	388	public void run() {	boolean status = rs.next();	Never declare a local variable if it is not going to be used
MainLibrary	112	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
MainLibrary	119	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	Parameters passed to methods must be referenced in the method body
MainLibrary	125	private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton3ActionPerformed	private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton3ActionPerformed	Parameters passed to methods must be referenced in the method body
MainLibrary	130	private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton4ActionPerformed	private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton4ActionPerformed	Parameters passed to methods must be referenced in the method body
NewView	42	public NewView() throws SQLException {	PreparedStatement ps = Con.prepareStatement("select IssuedBook.BookID,IssuedBook.UserID,Books.BookName , IssuedBook.IssueDate, IssuedBook.ReturnDate from Books,IssuedBook where Books.BookID=IssuedBook.BookID; ", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
NewView	43	public NewView() throws SQLException {	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
NewView	232	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
NewView	238	private void SearchFieldActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchFieldActionPerformed	private void SearchFieldActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchFieldActionPerformed	Parameters passed to methods must be referenced in the method body

NewView	242	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	Parameters passed to methods must be referenced in the method body
NewView	255	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	PreparedStatement ps = Con.prepareStatement("select IssuedBook.BookID,IssuedBook.UserID,Books.BookName , IssuedBook.IssueDate, IssuedBook.ReturnDate from Books,IssuedBook where Books.BookID=IssuedBook.BookID and Books.BookName like ?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
NewView	257	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
NewView	301	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	PreparedStatement ps = Con.prepareStatement("select IssuedBook.BookID,IssuedBook.UserID,Books.BookName , IssuedBook.IssueDate, IssuedBook.ReturnDate from Books,IssuedBook where Books.BookID=IssuedBook.BookID and IssuedBook.BookID=?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
NewView	303	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
NewView	347	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	PreparedStatement ps = Con.prepareStatement("select IssuedBook.BookID,IssuedBook.UserID,Books.BookName , IssuedBook.IssueDate, IssuedBook.ReturnDate from Books,IssuedBook where Books.BookID=IssuedBook.BookID and IssuedBook.UserID=?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
NewView	349	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use

NewView	391	private void BookIDRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BookIDRadioActionPerformed	private void BookIDRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BookIDRadioActionPerformed	Parameters passed to methods must be referenced in the method body
NewView	398	private void NameRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_NameRadioActionPerformed	private void NameRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_NameRadioActionPerformed	Parameters passed to methods must be referenced in the method body
NewView	406	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	Parameters passed to methods must be referenced in the method body
NewView	415	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	PreparedStatement ps = Con.prepareStatement("select IssuedBook.BookID,IssuedBook.UserID,Books.BookName , IssuedBook.IssueDate, IssuedBook.ReturnDate from Books,IssuedBook where Books.BookID=IssuedBook.BookID; ", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
NewView	416	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
NewView	450	private void UserIDRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_UserIDRadioActionPerformed	private void UserIDRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_UserIDRadioActionPerformed	Parameters passed to methods must be referenced in the method body
NewView	8	import java.awt.HeadlessException;	import java.awt.HeadlessException;	Only import classes that you are going to use
NewView	18	import javax.swing.JScrollPane;	import javax.swing.JScrollPane;	Only import classes that you are going to use
NewView	19	import javax.swing.JTable;	import javax.swing.JTable;	Only import classes that you are going to use
ReturnBookForm	183	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
ReturnBookForm	221	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	Parameters passed to methods must be referenced in the method body
ReturnBookForm	227	private void UserIDActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_UserIDActionPerformed	private void UserIDActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_UserIDActionPerformed	Parameters passed to methods must be referenced in the method body

ReturnBookForm	231	private void IYearActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_IYearActionPerformed	private void IYearActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_IYearActionPerformed	Parameters passed to methods must be referenced in the method body
ReturnBookForm	235	private void BookIDActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BookIDActionPerformed	private void BookIDActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BookIDActionPerformed	Parameters passed to methods must be referenced in the method body
ReturnBookForm	239	private void IDateActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_IDateActionPerformed	private void IDateActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_IDateActionPerformed	Parameters passed to methods must be referenced in the method body
TransBookDao	11	public static boolean checkBook(String bookcallno) {	Connection con = DB.getConnection();	Resources should always be closed after use
TransBookDao	12	public static boolean checkBook(String bookcallno) {	PreparedStatement ps = con.prepareStatement("select * from Books where BookID=?");	Resources should always be closed after use
TransBookDao	14	public static boolean checkBook(String bookcallno) {	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
TransBookDao	26	public static boolean BookValidate(String BookID) {	PreparedStatement ps = con.prepareStatement("select * from Books where BookID = ?");	Resources should always be closed after use
TransBookDao	28	public static boolean BookValidate(String BookID) {	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
TransBookDao	40	public static boolean UserValidate(String UserID) {	PreparedStatement ps = con.prepareStatement("select * from Users where UserID = ?");	Resources should always be closed after use
TransBookDao	42	public static boolean UserValidate(String UserID) {	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
TransBookDao	55	public static int updatebook(String bookcallno) {		Resources should always be closed after use
TransBookDao	57	public static int updatebook(String bookcallno) {	PreparedStatement ps = con.prepareStatement("select quantity,issued from books where callno=?");	Resources should always be closed after use
TransBookDao	59	public static int updatebook(String bookcallno) {	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
TransBookDao	66	public static int updatebook(String bookcallno) {	PreparedStatement ps2 = con.prepareStatement("update books set quantity=?,issued=? where callno=?");	Resources should always be closed after use
TransBookDao	84	public static int IssueBook(int BookID, int UserID, String IDate, String RDate) {	Connection con = DB.getConnection();	Resources should always be closed after use
TransBookDao	85	public static int IssueBook(int BookID, int UserID, String IDate, String RDate) {	PreparedStatement ps = con.prepareStatement("insert into IssuedBook values(?,?,?,?)");	Resources should always be closed after use

TransBookDao	102	public static int ReturnBook(int BookID, int UserID) {	Connection con = DB.getConnection();	Resources should always be closed after use
TransBookDao	103	public static int ReturnBook(int BookID, int UserID) {	PreparedStatement ps = con.prepareStatement("delete from IssuedBook where BookID=? and UserID=?");	Resources should always be closed after use
TransBookDao	117	public static boolean CheckIssuedBook(int BookID) {	PreparedStatement ps = con.prepareStatement("select * from IssuedBook where BookID=?");	Resources should always be closed after use
TransBookDao	119	public static boolean CheckIssuedBook(int BookID) {	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
TransBookDao	129	boolean status = false;	boolean status = false;	Never declare a local variable if it is not going to be used
TransBookDao	132	public static int Check(int UserID) {	PreparedStatement ps = con.prepareStatement("select * from Book_Count UserID=?");	Resources should always be closed after use
TransBookDao	134	public static int Check(int UserID) {	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
TransBookDao	4	import javax.swing.JTextField;	import javax.swing.JTextField;	Only import classes that you are going to use
UserForm	203	private void UserNameActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_UserNameActionPerformed	private void UserNameActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_UserNameActionPerformed	Parameters passed to methods must be referenced in the method body
UserForm	207	private void PositionActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_PositionActionPerformed	private void PositionActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_PositionActionPerformed	Parameters passed to methods must be referenced in the method body
UserForm	211	private void YearActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_YearActionPerformed	private void YearActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_YearActionPerformed	Parameters passed to methods must be referenced in the method body
UserForm	215	private void ProgramActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ProgramActionPerformed	private void ProgramActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ProgramActionPerformed	Parameters passed to methods must be referenced in the method body
UserForm	219	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	Parameters passed to methods must be referenced in the method body
UserForm	225	private void EmailActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_EmailActionPerformed	private void EmailActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_EmailActionPerformed	Parameters passed to methods must be referenced in the method body

UserForm	229	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
UserForm	314	private void setVisible() {	private void setVisible() {	Never declare a Private Method if it is not going to be used
UserLogin	153	private void usernameActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_usernameActionPerformed	private void usernameActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_usernameActionPerformed	Parameters passed to methods must be referenced in the method body
UserLogin	157	private void passwordActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_passwordActionPerformed	private void passwordActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_passwordActionPerformed	Parameters passed to methods must be referenced in the method body
UserLogin	161	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
UserLogin	180	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	Parameters passed to methods must be referenced in the method body
UserLoginSuccess	197	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
UserLoginSuccess	202	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed	Parameters passed to methods must be referenced in the method body
UserLoginSuccess	207	private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton7ActionPerformed	private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton7ActionPerformed	Parameters passed to methods must be referenced in the method body
UserLoginSuccess	213	private void EmailActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_EmailActionPerformed	private void EmailActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_EmailActionPerformed	Parameters passed to methods must be referenced in the method body
UserLoginSuccess	255	public static void main(String args[]) {	Connection Con;	Resources should always be closed after use
UserLoginSuccess	257	public static void main(String args[]) {	PreparedStatement ps;	Resources should always be closed after use
UserLoginSuccess	261	public static void main(String args[]) {	ResultSet rs;	Resources should always be closed after use
UserLoginSuccess	263	public static void main(String args[]) {	boolean status = rs.next();	Never declare a local variable if it is not going to be used

UsersDao	22	public static boolean validate(String name, String password) {	Connection con = DB.getConnection();	Resources should always be closed after use
UsersDao	24	public static boolean validate(String name, String password) {	Statement selectStatement = con.createStatement();	Resources should always be closed after use
UsersDao	25	public static boolean validate(String name, String password) {	ResultSet rs = selectStatement.executeQuery(select);	Resources should always be closed after use
UsersDao	37	public static boolean CheckIfAlready(String UserName) {	Connection con = DB.getConnection();	Resources should always be closed after use
UsersDao	39	public static boolean CheckIfAlready(String UserName) {	Statement selectStatement = con.createStatement();	Resources should always be closed after use
UsersDao	40	public static boolean CheckIfAlready(String UserName) {	ResultSet rs = selectStatement.executeQuery(select);	Resources should always be closed after use
UsersDao	56	public static int AddUser(String User, String UserPass, String UserEmail, String Date) {	Connection con = DB.getConnection();	Resources should always be closed after use
UsersDao	57	public static int AddUser(String User, String UserPass, String UserEmail, String Date) {	PreparedStatement ps = con.prepareStatement("insert into Users(UserPass,RegDate,UserName,Email) values(?,?,?,?)");	Resources should always be closed after use
UserView	45	public UserView() throws SQLException {	try (Connection Con = DB.getConnection()) {	Resources should always be closed after use
UserView	47	public UserView() throws SQLException {	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
UserView	163	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
UserView	8	import java.awt.HeadlessException;	import java.awt.HeadlessException;	Only import classes that you are going to use
UserView	17	import javax.swing.JScrollPane;	import javax.swing.JScrollPane;	Only import classes that you are going to use
UserView	18	import javax.swing.JTable;	import javax.swing.JTable;	Only import classes that you are going to use
UserViewBook	42	public UserViewBook() throws SQLException {	PreparedStatement ps = Con.prepareStatement("select Books.BookID, Books.BookName,Books.Genre,Books.Author,Books.Publisher, Books.Row,Books.Shelf, IssuedBook.UserID from Books left outer join IssuedBook on Books.BookID= IssuedBook.BookID;", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
UserViewBook	43	public UserViewBook() throws SQLException {	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
UserViewBook	62	public UserViewBook() throws SQLException {	String Check = "";	Never declare a local variable if it is not going to be used

UserViewBook	248	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
UserViewBook	253	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	Parameters passed to methods must be referenced in the method body
UserViewBook	255	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	if (SearchField.getText() == "") {	Recommend using equals () method
UserViewBook	282	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	PreparedStatement ps = Con.prepareStatement("select A.BookID, A.BookName,A.Genre,A.Author,A.Publisher, A.Row,A.Shelf, IssuedBook.UserID from (select * from Books where BookName like ?) as A left outer join IssuedBook on A.BookID= IssuedBook.BookID", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
UserViewBook	284	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
UserViewBook	343	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	try (Connection Con = DB.getConnection()) {	Resources should always be closed after use
UserViewBook	345	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
UserViewBook	401	private void AuthorRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_AuthorRadioActionPerformed	private void AuthorRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_AuthorRadioActionPerformed	Parameters passed to methods must be referenced in the method body
UserViewBook	406	private void NameRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_NameRadioActionPerformed	private void NameRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_NameRadioActionPerformed	Parameters passed to methods must be referenced in the method body
UserViewBook	411	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	Parameters passed to methods must be referenced in the method body

UserViewBook	417	private void NotIssuedActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_NotIssuedActionPerformed	private void NotIssuedActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_NotIssuedActionPerformed	Parameters passed to methods must be referenced in the method body
UserViewBook	423	private void ShowALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ShowALLActionPerformed	private void ShowALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ShowALLActionPerformed	Parameters passed to methods must be referenced in the method body
UserViewBook	447	private void ShowALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ShowALLActionPerformed	PreparedStatement ps = Con.prepareStatement("select Books.BookID, Books.BookName,Books.Genre,Books.Author,Books.Publisher, Books.Row,Books.Shelf, IssuedBook.UserID from Books left outer join IssuedBook on Books.BookID= IssuedBook.BookID;", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
UserViewBook	448	private void ShowALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ShowALLActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
UserViewBook	467	private void ShowALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ShowALLActionPerformed	String Check = "";	Never declare a local variable if it is not going to be used
UserViewBook	8	import java.awt.HeadlessException;	import java.awt.HeadlessException;	Only import classes that you are going to use
UserViewBook	18	import javax.swing.JScrollPane;	import javax.swing.JScrollPane;	Only import classes that you are going to use
UserViewBook	19	import javax.swing.JTable;	import javax.swing.JTable;	Only import classes that you are going to use
ViewBook	41	public ViewBook() throws SQLException {	PreparedStatement ps = Con.prepareStatement("select * from Books", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
ViewBook	42	public ViewBook() throws SQLException {	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
ViewBook	242	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed	Parameters passed to methods must be referenced in the method body
ViewBook	247	private void SearchFieldActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchFieldActionPerformed	private void SearchFieldActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchFieldActionPerformed	Parameters passed to methods must be referenced in the method body

ViewBook	251	private void NameRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_NameRadioActionPerformed	private void NameRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_NameRadioActionPerformed	Parameters passed to methods must be referenced in the method body
ViewBook	257	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	Parameters passed to methods must be referenced in the method body
ViewBook	270	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	PreparedStatement ps = Con.prepareStatement("select * from Books where BookName like ?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
ViewBook	272	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
ViewBook	314	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	PreparedStatement ps = Con.prepareStatement("select * from Books where Author like ?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
ViewBook	316	private void SearchActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SearchActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
ViewBook	372	private void AuthorRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_AuthorRadioActionPerformed	private void AuthorRadioActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_AuthorRadioActionPerformed	Parameters passed to methods must be referenced in the method body
ViewBook	378	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	Parameters passed to methods must be referenced in the method body
ViewBook	391	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	PreparedStatement ps = Con.prepareStatement("select * from Books", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);	Resources should always be closed after use
ViewBook	392	private void ALLActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ALLActionPerformed	ResultSet rs = ps.executeQuery();	Resources should always be closed after use
ViewBook	8	import java.awt.HeadlessException;	import java.awt.HeadlessException;	Only import classes that you are going to use
ViewBook	18	import javax.swing.JScrollPane;	import javax.swing.JScrollPane;	Only import classes that you are going to use
ViewBook	19	import javax.swing.JTable;	import javax.swing.JTable;	Only import classes that you are going to use

Secure Application Development – Report Summary

This report began with a detailed review of the security features available in both Java and MySQL including

State of the Art - Latest Programming Paradigms

1. Security of Java
2. Java Object Orientated Programming (OOP)
3. Java Virtual Machine (JVM) Architecture
4. Java Language Security
5. Secure Coding in Java
6. Java Security – Recent Updates
7. Security of MySQL

State of the Art – Security Testing

The following section reviewed current Security Testing processes, tools and technologies including

1. Software Development Life Cycle (SDLC)
2. Software Assurance Maturity Model (SAMM)
3. Application Security Testing
4. Application Security Testing Framework
5. Application Security Testing Processes
6. Application Security Testing Types

Secure Application Development – Java Application Testing

In the following section different application testing tools and techniques were used to analyse the Java Project. This section included

1. Project Configuration
2. Java Application Functional Test
3. Java Code Analysis

Secure Application Development – Proposed Solutions

This report was concluded by identifying solutions to the vulnerabilities identified during Java Application Testing as follows

1. Java Code Security Issues Proposed Solutions
2. Java ARchive File (JAR) Security Issues Proposed Solutions
3. MySQL Security Related Issues

References

1	TIOBE.com	[2019]	TIOBE (the software quality company)	https://www.tiobe.com/	[Accessed 11th Dec 2019]
2	TIOBE.com	[2019]	The Java Programming Language	https://www.tiobe.com/tiobe-index/java/	[Accessed 11th Dec 2019]
3	ComputerWeekly.com	[2002]	Write once, run anywhere?	https://www.computerweekly.com/feature/Write-once-run-anywhere	[Accessed 11th Dec 2019]
4	Wikipedia.org	[2019]	Object-Oriented Programming	https://en.wikipedia.org/wiki/Object-oriented_programming	[Accessed 11th Dec 2019]
5	Java.com	[2019]	Java Downloads for All Operating Systems	https://www.java.com/en/download/manual.jsp	[Accessed 11th Dec 2019]
6	Oracle.com	[2019]	Java SE Security	https://www.oracle.com/java/technologies/index-jsp.html	[Accessed 11th Dec 2019]
7	Wikipedia.org	[2019]	Object-oriented programming	https://en.wikipedia.org/wiki/Object-oriented_programming	[Accessed 11th Dec 2019]
8	Taaffe, Jonathon	[2019]	Figure 1. Java Runtime Environment (JRE)		[Created 11th Dec 2019]
9	Guru99.com	[2019]	What is JVM (Java Virtual Machine)? with Architecture: JAVA Programming Tutorial	https://www.youtube.com/watch?v=G1ubVOl9IBw	[Accessed 11th Dec 2019]
10	Oracle.com	[2019]	Class SecurityManager	https://docs.oracle.com/javase/7/docs/api/java/lang/SecurityManager.html	[Accessed 11th Dec 2019]
11	Oracle.com	[2019]	Java™ Security Overview	https://docs.oracle.com/javase/8/docs/technotes/guides/security/overview/jsoverview.html	[Accessed 11th Dec 2019]
12	Oracle.com	[2019]	The Java Language Environment	https://www.oracle.com/technetwork/java/security-136118.html	[Accessed 11th Dec 2019]
13	Oracle.com	[2019]	What Is an Interface?	https://docs.oracle.com/javase/tutorial/java/concepts/interface.html	[Accessed 11th Dec 2019]
14	Taaffe, Jonathon	[2019]	Figure 2. Bytecode Verification Process		[Created 11th Dec 2019]
15	Oracle.com	[2019]	Package java.security	https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/security/package-summary.html	[Accessed 11th Dec 2019]
16	Oracle.com	[2019]	Java SE Development Kit 11 Downloads	https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html	[Accessed 11th Dec 2019]
17	Oracle.com	[2019]	Introduction to JDK Providers	https://docs.oracle.com/en/java/javase/11/security/oracle-providers.html	[Accessed 11th Dec 2019]
18	Oracle.com	[2019]	JDK Security API: java.base	https://docs.oracle.com/en/java/javase/11/docs/api/java.base/module-summary.html	[Accessed 11th Dec 2019]
19	Oracle.com	[2019]	JDK Security API: java.security.jgss	https://docs.oracle.com/en/java/javase/11/docs/api/java.security.jgss/module-summary.html	[Accessed 11th Dec 2019]
20	Oracle.com	[2019]	JDK Security API: java.security.sasl	https://docs.oracle.com/en/java/javase/11/docs/api/java.security.sasl/module-summary.html	[Accessed 11th Dec 2019]
21	Oracle.com	[2019]	JDK Security API: java.smartcardio	https://docs.oracle.com/en/java/javase/11/docs/api/java.smartcardio/module-summary.html	[Accessed 11th Dec 2019]
22	Oracle.com	[2019]	JDK Security API: java.xml.crypto	https://docs.oracle.com/en/java/javase/11/docs/api/java.xml.crypto/module-summary.html	[Accessed 11th Dec 2019]
23	Oracle.com	[2019]	JDK Security API: jdk.jartool	https://docs.oracle.com/en/java/javase/11/docs/api/jdk.jartool/jdk/security/jarsigner/package-summary.html	[Accessed 11th Dec 2019]
24	Oracle.com	[2019]	JDK Security API: jdk.security.auth	https://docs.oracle.com/en/java/javase/11/docs/api/jdk.security.auth/module-summary.html	[Accessed 11th Dec 2019]
25	Oracle.com	[2019]	JDK Security API: jdk.security.jgss	https://docs.oracle.com/en/java/javase/11/docs/api/jdk.security.jgss/module-summary.html	[Accessed 11th Dec 2019]
26	Taaffe, Jonathon	[2019]	Table 1. JDK Modules including Security API		[Created 11th Dec 2019]
27	Oracle.com	[2019]	Module java.base, Package java.security, Class KeyFactory	https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/security/KeyFactory.html	[Accessed 11th Dec 2019]
28	Oracle.com	[2019]	Java Cryptography	https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-C6D250FC-F147-4284-A6BF-8384DFD39DA6	[Accessed 11th Dec 2019]
29	Oracle.com	[2019]	Java Public Key Infrastructure	https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-054AD71D-D449-47FF-B6F7-F416DA821D46	[Accessed 11th Dec 2019]
30	Techopedia.com	[2019]	Public Key Infrastructure (PKI)	https://www.techopedia.com/definition/4071/public-key-infrastructure-pki	[Accessed 11th Dec 2019]
31	Techopedia.com	[2019]	Certificate Authority (CA)	https://www.techopedia.com/definition/29742/certificate-authority-ca	[Accessed 11th Dec 2019]
32	Oracle.com	[2019]	Java Archive (JAR) Files	https://docs.oracle.com/javase/8/docs/technotes/guides/jar/index.html	[Accessed 11th Dec 2019]
33	Oracle.com	[2019]	Java Authentication	https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-F8BE6C49-3506-4D1A-8E4E-053CA439D1E2	[Accessed 11th Dec 2019]

34	Oracle.com	[2019]	Figure 3. Java Application, Authentication Framework and Authentication Mechanism Independence	https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-F8BE6C49-3506-4D1A-8E4E-053CA439D1E2	[Accessed 11th Dec 2019]
35	Oracle.com	[2019]	Java Secure Communication	https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-4E5FEEF5-A541-4222-AD18-31AE184F38E4	[Accessed 11th Dec 2019]
36	SSL.com	[2019]	What is SSL?	https://www.ssl.com/faqs/faq-what-is-ssl/	[Accessed 11th Dec 2019]
37	Techopedia.com	[2019]	Transport Layer Security (TLS)	https://www.techopedia.com/definition/4143/transport-layer-security-tls	[Accessed 11th Dec 2019]
38	Wikipedia.org	[2019]	Datagram Transport Layer Security	https://en.wikipedia.org/wiki/Datagram_Transport_Layer_Security	[Accessed 11th Dec 2019]
39	Oracle.com	[2019]	Simple Authentication and Security Layer (SASL)	https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-7C74ECA4-3645-4756-B8FB-63D84480F4AD	[Accessed 11th Dec 2019]
40	Oracle.com	[2019]	Java Access Control	https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-BBEC2DC8-BA00-42B1-B52A-A49488FCF8FE	[Accessed 11th Dec 2019]
41	Oracle.com	[2019]	Java Security Classes	https://docs.oracle.com/en/java/javase/11/security/java-security-overview1.html#GUID-CF323502-F719-4618-91FE-4D37CB57FF24	[Accessed 11th Dec 2019]
42	Taaffe, Jonathon	[2019]	Table 2. Java Security Classes		[Created 11th Dec 2019]
43	Oracle.com	[2019]	Secure Coding Guidelines for Java SE	https://www.oracle.com/technetwork/java/seccodeguide-139067.html	[Accessed 11th Dec 2019]
44	Oracle.com	[2019]	Secure Coding Guidelines – 0. Fundamentals	https://www.oracle.com/technetwork/java/seccodeguide-139067.html#0	[Accessed 11th Dec 2019]
45	Oracle.com	[2019]	Secure Coding Guidelines – 1. Denial of Service	https://www.oracle.com/technetwork/java/seccodeguide-139067.html#1	[Accessed 11th Dec 2019]
46	Oracle.com	[2019]	Secure Coding Guidelines – 2. Confidential Information	https://www.oracle.com/technetwork/java/seccodeguide-139067.html#2	[Accessed 11th Dec 2019]
47	Oracle.com	[2019]	Secure Coding Guidelines - 3. Injection and Inclusion	https://www.oracle.com/technetwork/java/seccodeguide-139067.html#3	[Accessed 11th Dec 2019]
48	Oracle.com	[2019]	Secure Coding Guidelines - 4. Accessibility and Extensibility	https://www.oracle.com/technetwork/java/seccodeguide-139067.html#4	[Accessed 11th Dec 2019]
49	Oracle.com	[2019]	Secure Coding Guidelines - 5. Mutability	https://www.oracle.com/technetwork/java/seccodeguide-139067.html#6	[Accessed 11th Dec 2019]
50	Oracle.com	[2019]	Immutable Objects	https://docs.oracle.com/javase/tutorial/essential/concurrency/immutable.html	[Accessed 11th Dec 2019]
51	Oracle.com	[2019]	Secure Coding Guidelines - 6. Object Construction	https://www.oracle.com/technetwork/java/seccodeguide-139067.html#7	[Accessed 11th Dec 2019]
52	Oracle.com	[2019]	Secure Coding Guidelines - 7. Serialization and Deserialization	https://www.oracle.com/technetwork/java/seccodeguide-139067.html#8	[Accessed 11th Dec 2019]
53	StackOverflow.com	[2019]	What is serialization?	https://stackoverflow.com/questions/633402/what-is-serialization	[Accessed 11th Dec 2019]
54	Oracle.com	[2019]	Secure Coding Guidelines - 8. Access Control	https://www.oracle.com/technetwork/java/seccodeguide-139067.html#9	[Accessed 11th Dec 2019]
55	Oracle.com	[2019]	JDK 11 Release Notes	https://www.oracle.com/technetwork/java/javase/11-relnote-issues-5012449.html	[Accessed 11th Dec 2019]
56	Taaffe, Jonathon	[2019]	Table 3. Java 11 SE and JDK Security Updates		[Created 11th Dec 2019]
57	Oracle.com	[2019]	JDK 12 Release Notes	https://www.oracle.com/technetwork/java/javase/12-relnote-issues-5211422.html	[Accessed 11th Dec 2019]
58	Taaffe, Jonathon	[2019]	Table 4. Java 12 SE and JDK Security Updates		[Created 11th Dec 2019]
59	Oracle.com	[2019]	JDK 13 Release Notes	https://www.oracle.com/technetwork/java/13-relnote-issues-5460548.html	[Accessed 11th Dec 2019]
60	Taaffe, Jonathon	[2019]	Table 5. Java 13 SE and JDK Security Updates		[Created 11th Dec 2019]
61	StackOverflow.com	[2019]	StackOverflow Developer Survey Results 2019	https://insights.stackoverflow.com/survey/2019#technology-_-databases	[Accessed 11th Dec 2019]
62	DB-Engines.com	[2019]	DB-Engines Ranking	https://db-engines.com/en/ranking	[Accessed 11th Dec 2019]
63	JavaPoint.com	[2019]	MySQL Features	https://www.javatpoint.com/mysql-features	[Accessed 11th Dec 2019]
64	MySQL.com	[2019]	MySQL Technical Specifications	https://www.mysql.com/products/enterprise/techspec.html	[Accessed 11th Dec 2019]
65	Taaffe, Jonathon	[2019]	Table 6. MySQL Technical Specifications Summary		[Created 11th Dec 2019]
66	MySQL.com	[2019]	Chapter 1 Security	https://dev.mysql.com/doc/mysql-security-excerpt/8.0/en/security.html	[Accessed 11th Dec 2019]

67	MySQL.com	[2019]	Security Guidelines	https://dev.mysql.com/doc/refman/8.0/en/security-guidelines.html	[Accessed 11th Dec 2019]
68	MySQL.com	[2019]	Keeping Passwords Secure	https://dev.mysql.com/doc/refman/8.0/en/password-security.html	[Accessed 11th Dec 2019]
69	MySQL.com	[2019]	Making MySQL Secure Against Attackers	https://dev.mysql.com/doc/refman/8.0/en/security-against-attack.html	[Accessed 11th Dec 2019]
70	MySQL.com	[2019]	Security-Related mysqld Options and Variables	https://dev.mysql.com/doc/refman/8.0/en/security-options.html	[Accessed 11th Dec 2019]
71	MySQL.com	[2019]	How to Run MySQL as a Normal User	https://dev.mysql.com/doc/refman/8.0/en/changing-mysql-user.html	[Accessed 11th Dec 2019]
72	MySQL.com	[2019]	Access Control and Account Management	https://dev.mysql.com/doc/refman/8.0/en/access-control.html	[Accessed 11th Dec 2019]
73	MySQL.com	[2019]	Using Encrypted Connections	https://dev.mysql.com/doc/refman/8.0/en/encrypted-connections.html	[Accessed 11th Dec 2019]
74	MySQL.com	[2019]	Security Components and Plugins	https://dev.mysql.com/doc/refman/8.0/en/security-plugins.html	[Accessed 11th Dec 2019]
75	Stackify.com	[2019]	What is SDLC?	https://stackify.com/what-is-sdlc/	[Accessed 11th Dec 2019]
76	OWASP.org	[2019]	OWASP™ Foundation. The free and open software security community	https://www.owasp.org/	[Accessed 11th Dec 2019]
77	OWASP.org	[2019]	OWASP SAMM Project	https://www.owasp.org/index.php/OWASP_SAMM_Project	[Accessed 11th Dec 2019]
78	OWASP.org	[2019]	Figure 4. OWASP SAMM Foundation Model	https://github.com/OWASP/samm/raw/master/Supporting%20Resources/v1.5/Final/SAMM_Core_V1-5_FINAL.pdf	[Accessed 11th Dec 2019]
79	MicroFocus.com	[2019]	Gartner 2019 Magic Quadrant for Application Security Testing	https://www.microfocus.com/en-us/assets/security/magic-quadrant-for-application-security-testing	[Accessed 11th Dec 2019]
80	OWASP.org	[2019]	OWASP Testing Guide	https://www.owasp.org/index.php/OWASP_Testing_Guide_v3_Table_of_Contents	[Accessed 11th Dec 2019]
81	Taaffe, Jonathon	[2019]	Table 7. OWASP Application Security Testing for Java and MySQL		[Created 11th Dec 2019]
82	ISECOM.org	[2019]	The Open Source Security Testing Methodology Manual	https://www.isecom.org/OSSTMM.3.pdf	[Accessed 11th Dec 2019]
83	OWASP.org	[2019]	Buffer overflow attack	https://www.owasp.org/index.php/Buffer_overflow_attack	[Accessed 11th Dec 2019]
84	Nmap.org	[2019]	Nmap Network Mapper	https://nmap.org/	[Accessed 11th Dec 2019]
85	Microsoft.com	[2019]	netstat	https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/netstat	[Accessed 11th Dec 2019]
86	Wikipedia.org	[2019]	Transmission Control Protocol	https://en.wikipedia.org/wiki/Transmission_Control_Protocol	[Accessed 11th Dec 2019]
87	Tenable.com	[2019]	Tenable Nessus	https://www.tenable.com/products/nessus	[Accessed 11th Dec 2019]
88	Tenable.com	[2019]	Tenable.com	https://www.tenable.com/	[Accessed 11th Dec 2019]
89	Oracle.com	[2019]	Java SE Development Kit 11 Downloads	https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html	[Accessed 11th Dec 2019]
90	Eclipse.org	[2019]	Eclipse Foundation Download Eclipse Technology that is right for you	https://www.eclipse.org/downloads/	[Accessed 11th Dec 2019]
91	MySQL.com	[2019]	MySQL Community Downloads	https://dev.mysql.com/downloads/windows/installer/8.0.html	[Accessed 11th Dec 2019]
92	SourceForge.net	[2019]	Visual Code Grepper (VCG)	https://sourceforge.net/projects/visualcodegrepp/	[Accessed 11th Dec 2019]
93	Taaffe, Jonathon	[2019]	Table 8. Visual Code Grepper Summary Results		[Created 11th Dec 2019]
94	Jarchitect.com	[2019]	Jarchitect.com Discover amazing details about your Java projects	https://www.jarchitect.com/	[Accessed 11th Dec 2019]
95	Taaffe, Jonathon	[2019]	Figure 5. JArchitect Java Static Analysis Tool Application Metrics		[Created 11th Dec 2019]
96	Taaffe, Jonathon	[2019]	Figure 6. JArchitect Java Static Analysis Tool Rules Summary		[Created 11th Dec 2019]
97	Kiuwan.com	[2019]	Kiuwan Security Solutions for your DevOps Process	https://www.kiuwan.com/	[Accessed 11th Dec 2019]
98	Taaffe, Jonathon	[2019]	Figure 7. Kiuwan Static Code Analysis Security Rating - Summary		[Created 11th Dec 2019]
99	Taaffe, Jonathon	[2019]	Figure 8. Kiuwan Static Code Analysis Top 3 Vulnerabilities by Type		[Created 11th Dec 2019]
100	Taaffe, Jonathon	[2019]	Figure 9. Kiuwan Static Code Analysis Top 3 Worst Files		[Created 11th Dec 2019]
101	PMD.github.io	[2019]	PMD Source Code Analyser	https://pmd.github.io/	[Accessed 11th Dec 2019]

102	Taaffe, Jonathon	[2019]	Table 9. PDM Summary Results		[Created 11th Dec 2019]
103	Xanitizer.com	[2019]	Xanitizer Static Code Analysis Tool	https://www.xanitizer.com/xanitizer/	[Accessed 11th Dec 2019]
104	Taaffe, Jonathon	[2019]	Figure 10. Xanitizer Static Code Analysis Tool Finding 1		[Created 11th Dec 2019]
105	Taaffe, Jonathon	[2019]	Figure 11. Xanitizer Static Code Analysis Tool Finding 2		[Created 11th Dec 2019]
106	Oracle.com	[2019]	Writing Final Classes and Methods	https://docs.oracle.com/javase/tutorial/java/IandI/final.html	[Accessed 11th Dec 2019]
107	Taaffe, Jonathon	[2019]	Table 10. Java Files and Associated Code with Public Class Not Declared as Final		[Created 11th Dec 2019]
108	Taaffe, Jonathon	[2019]	Table 11. Java Files and Associated Code Updated with Public Class Declared as Final		[Created 11th Dec 2019]
109	Oracle.com	[2019]	Primitive Data Types	https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html	[Accessed 11th Dec 2019]
110	Taaffe, Jonathon	[2019]	Table 12. Java Files and Associated Code with Operation on Primitive Data Type		[Created 11th Dec 2019]
111	CVEDetails.com	[2019]	Vulnerability Details : CVE-2018-3258	https://www.cvedetails.com/cve/CVE-2018-3258/	[Accessed 11th Dec 2019]
112	Taaffe, Jonathon	[2019]	Figure 12. MySQL J Connector CVSS Scores and Vulnerability Types		[Created 11th Dec 2019]
113	Oracle.com	[2019]	Oracle Critical Patch Update Advisory - October 2018	https://www.oracle.com/security-alerts/cpuoct2018.html	[Accessed 11th Dec 2019]
114	Taaffe, Jonathon	[2019]	Table 13. Summary of Java Static Code Analysis		[Created 11th Dec 2019]