

Static Analysis with SpotBugs

SpotBugs was configured using version 4.0.0 of spotbugs-maven-plugin, which uses version 4.0.1 of SpotBugs. Here is a list of issues detected by SpotBugs that remain unfixed, and our justification for leaving them so.

1.

[ERROR] edu.baylor.csi3471.netime_planner.models.domain_objects.User.getPasswordHash()
may expose internal representation by returning User.passwordHash

[edu.baylor.csi3471.netime_planner.models.domain_objects.User] At User.java:[line 39]

EI_EXPOSE_REP

[ERROR]

edu.baylor.csi3471.netime_planner.models.domain_objects.User.setPasswordHash(char[]) may
expose internal representation by storing an externally mutable object into User.passwordHash

[edu.baylor.csi3471.netime_planner.models.domain_objects.User] At User.java:[line 43]

EI_EXPOSE_REP2

Justification:

The reason the password hashes are stored as character arrays rather than Strings is to provide more security, since Strings are immutable and cause copies to be made transparently in the background. To prevent this from exposing the internal object, we would have to copy the array containing the password hash, which defeats the purpose of avoiding String.

2.

[ERROR]

edu.baylor.csi3471.netime_planner.models.persistence.impl.ActivityDbDAO.delete(Activity)
may fail to clean up java.sql.Statement

And many other similar errors

This is a false positive. SpotBugs doesn't take try-with-resources statements into account. See this issue: <https://github.com/spotbugs/spotbugs/issues/493>

3.

[ERROR]

edu.baylor.csi3471.netime_planner.models.persistence.impl.ActivityDbDAO.findById(int)
may fail to clean up java.sql.ResultSet

This is also a false positive. Per the Java documentation (<https://docs.oracle.com/javase/9/docs/api/java/sql/Statement.html#close>) the result sets are automatically closed when the statements are closed.

4.

[ERROR] Possible doublecheck on
edu.baylor.csi3471.netime_planner.services.ServiceManager.instance

This is intentional. The first check is unsynchronized for performance reasons. If no instance exists, one is created in a synchronous block, but it must be checked again inside the synchronous block.