**security tool spot bugs**

below I’m showing me using different code previous with spot bugs and how it highlights what errors in the code, and I show how to fix one on them.

Text

Description automatically generated



Fig 1.0

* Fig 1.0 SQL code is to receive the user with the correct email and password. This is fine but with spot bugs it is telling you to use a prepared statement instead as its more secure and less prone to SQL injections.

Text

Description automatically generated



Fig 1.1

* Fig 1.1 is SQL code as well to update stock, but it used two or three times in the code and SonalLint suggest making a variable that is called so if updating the string it only needs updated in one place instead of other places which is less prone to mistakes. Sonar lint doesn’t want string literal to be duplicated.

**Analysis of bug 1 spot bugs**

Identification of issue – what is the bug that was found

* The method invokes the execute method on an SQL statement with a String that seems to be dynamically generated. The issue is the SQL statement its more secure to use prepared statement than the way used with the execute update way.

Solution to this bug

* using a prepared statement instead. It is more efficient and less vulnerable to SQL injection attacks.

code fix – show and explain code changes



Graphical user interface, text

Description automatically generated

Fig 2 - spot bugs prepared statement

As seen in Fig 2 I have commented out the statement and the execute Update statement so that I can use the prepared statement which is more efficient and less vulnerable to SQL injections. For the query string you are insert into the user table and the values we use questions marks. Then below we put in the values by mystmt.setString(1,user) for the first question mark and pass for the second question mark. Then with the result set we execute the mystmt prepared statement.