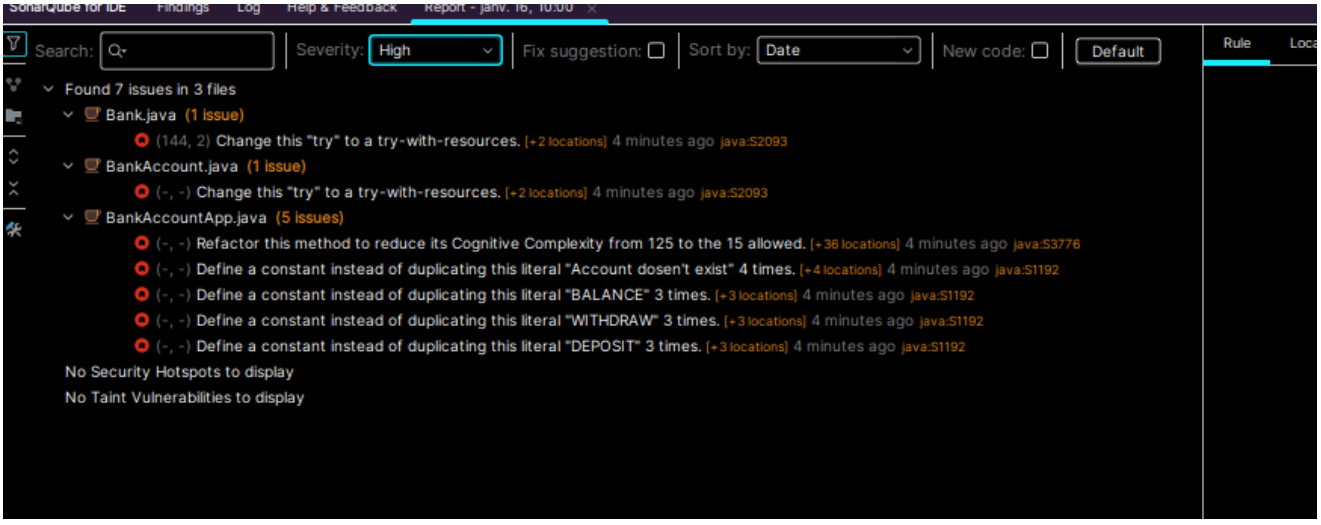
Exercise 4 (Repo 1): SonarQube: Static Analysis & Quick Fixes

I filtered the errors to display HIGH severitiy ones :



I identified three important issues

Issue 1

- **Rule name:** Refactor this method to reduce its Cognitive Complexity
- **File and line:** `BankAccountApp.java` (main method)
- **Explanation:**
  The method has a cognitive complexity of **125**, far above the allowed **15**.
  This indicates deeply nested conditions and too many responsibilities in a single method, making the code hard to read, understand, and maintain.

Issue 2

- **Rule name:** Replace duplicated string literals with constants
- **File and line:** `BankAccountApp.java` (multiple lines – literals like `"DEPOSIT"`, `"WITHDRAW"`, `"BALANCE"`)
- **Explanation:**
  Repeating string literals increases the risk of inconsistencies and typos.
  If a value needs to change, it must be updated in several places instead of one wich is a risk of bug.

Issue 3

- **Rule name:** Change this `try` to a try-with-resources
- **File and line:** `Bank.java:144` and `BankAccount.java`
- **Explanation:**
  Using a classic `try` block with resources can lead to mem leak if they are not properly closed.

Do SonarLint issues appear more often in the classes with higher WMC / CBO you saw earlier, or not really?

No, as **Person** was identified before and the number of errors and their severity seems unrelated to previous WMC / CBO analysis