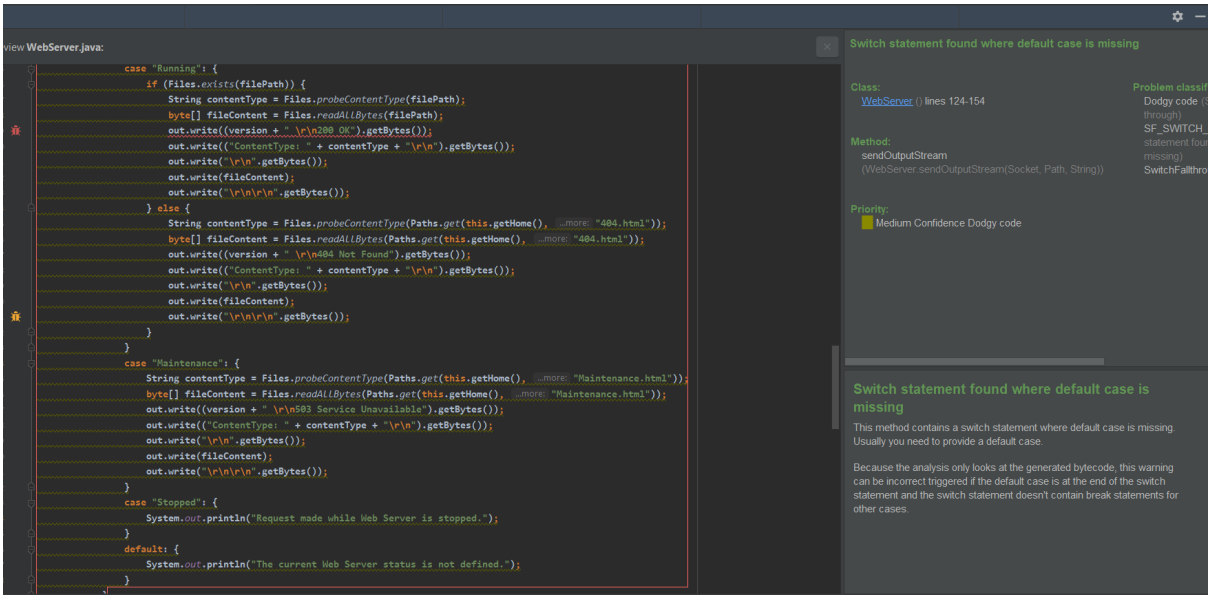
Cărare Raluca-Teodora
IV CTI RO - *VVS Project*

Analiza statică și dinamică
- Web Server -

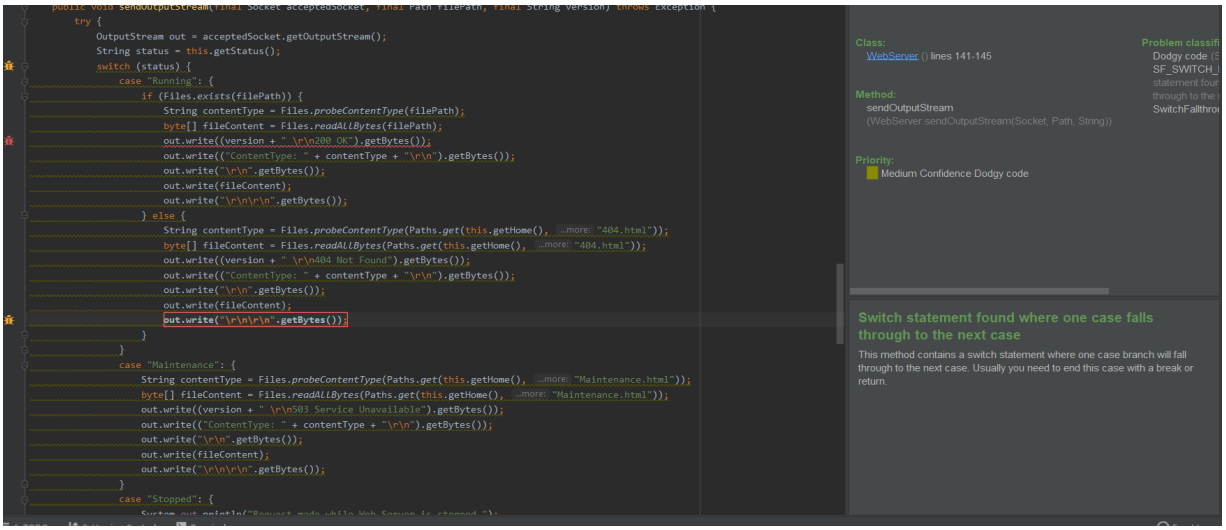
1. Analiză statică

Pentru analiza statică a proiectului, am folosit pluginul SpotBugs din IntelliJ IDEA. După rulare, pluginul a depistat 6 bug-uri în 3 clase diferite, 4 dintre ele fiind Dodgy code, iar 2 Internationalization.

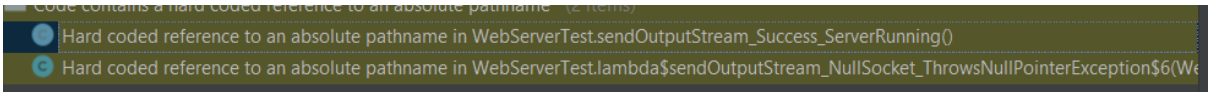
a) Dodgy code



Cum aceast statement are default case la sfârșitul switch-ului, acesta a fost un incorrent warning și nu a avut nevoie de schimbări.

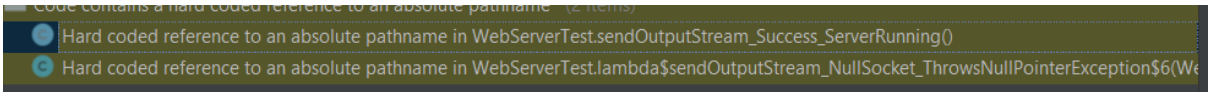


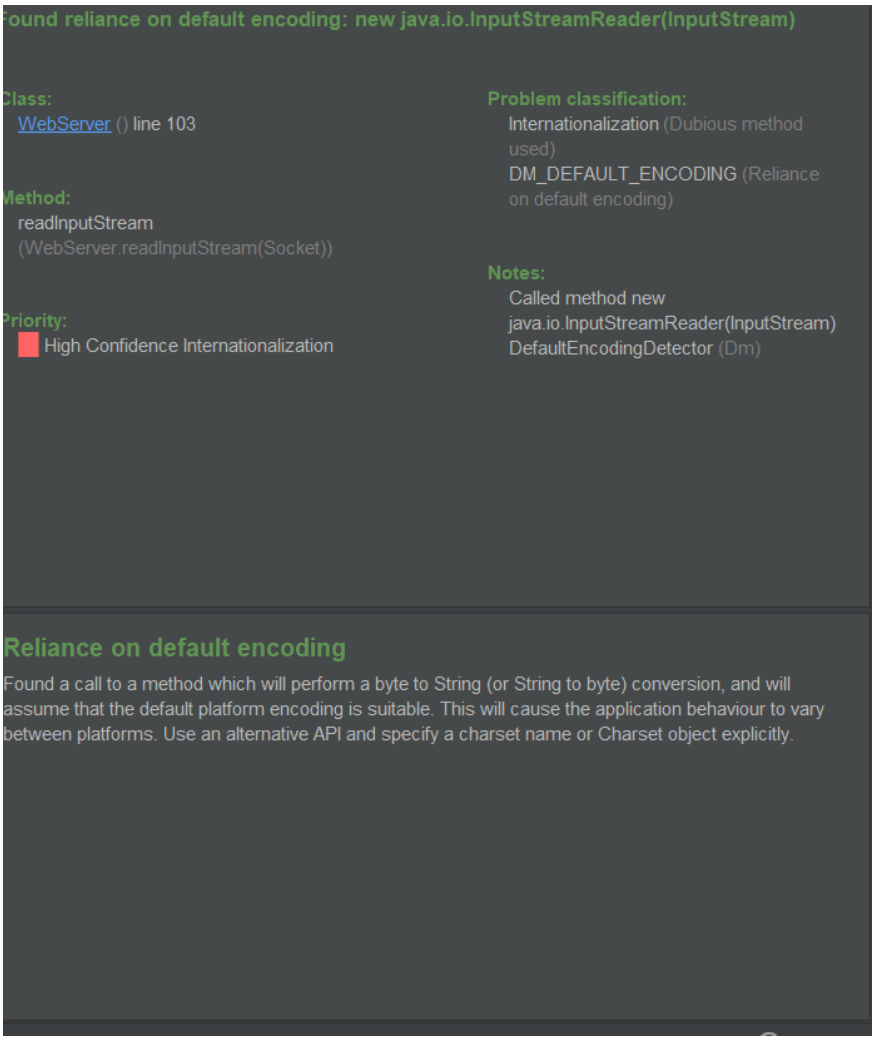
În acest caz problema nu există, căci nu e nevoie de un break sau return în cadrul case-urilor menționate.



Următoarele 2 bug-uri fac referire la valori harcodate ale unor path-uri pe care le-am folosit. Rezolvare este mutare acestora în fișiere de tip .env pentru a nu fi vizibile pe GitHub, de exemplu și a nu fi harcodate.

b) Internationalization





Bug-urile descoperite aici au apărut din cauza conversiei byte to String sau String to byte și nu a fost nevoie de rezolvarea lor.

De asemenea, pentru o mai clară analiză dinamică, am folosit tool-ul CheckStyle din IntelliJ IDEA.

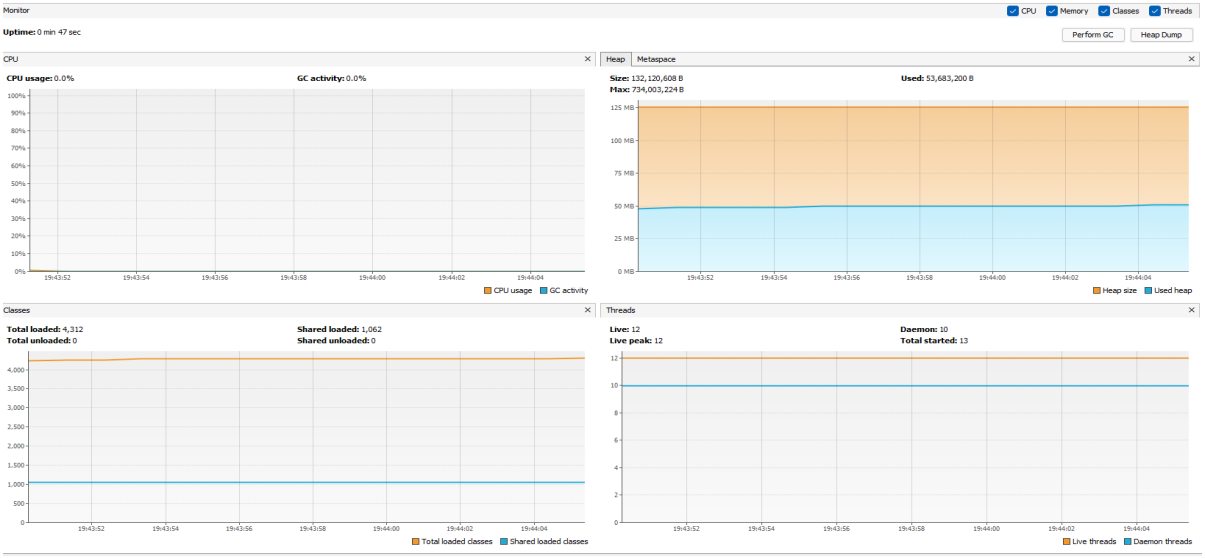
Cu ajutorul acestui plugin am depistat 67 de posibile greșeli la proiect.

Cum majoritatea acestor bug-uri nu afectau în vreun mod modul de funcționare sau de înțelegere a proiectului, nu a fost nevoie de multe ajutări.

2. Analiză dinamică

Pentru analiza dinamică am folosit tool-ul menționat la laboratorul de VVS, VisualVM, care are un plug-in pentru IntelliJ IDEA.

În modul Stopped, cu uptime de aproximativ un minut, aceasta este starea serverului:



În funcție de timp, memoria heap crește, dar are și mici scăderi la anumite intervale.

În modul Running, la aproximativ un mint uptime, serverul arată așa, având mai multe clase încărcate:

Deși memoria heap pare că scare de la un punct, după scăderea bruscă aceasta a crescut progresiv la același nivel ca anterior.

La rularea testelor, aceasta este activitatea monitorizată: