

Static Analysis Report  
Abass Sesay  
CS 576

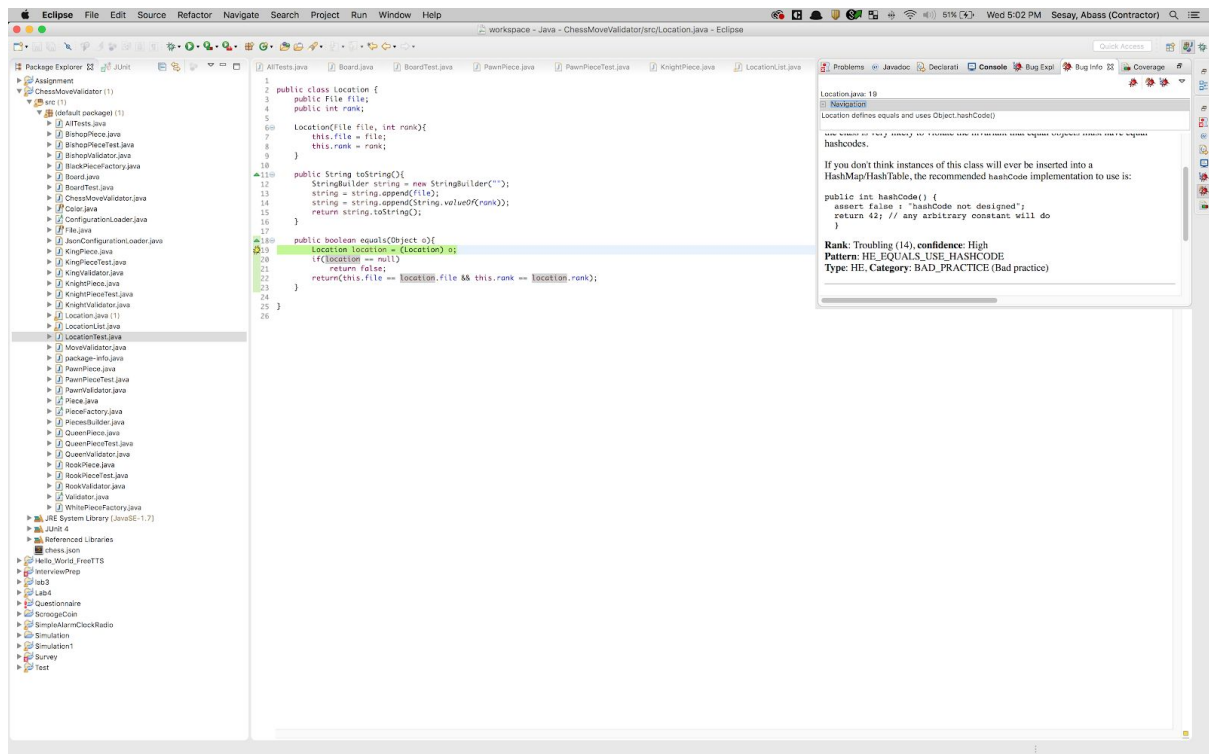
**Tools used**

FindBugs - Used to identifies bugs within the codebase.

EclEmma - JaCoCo Code coverage tool.

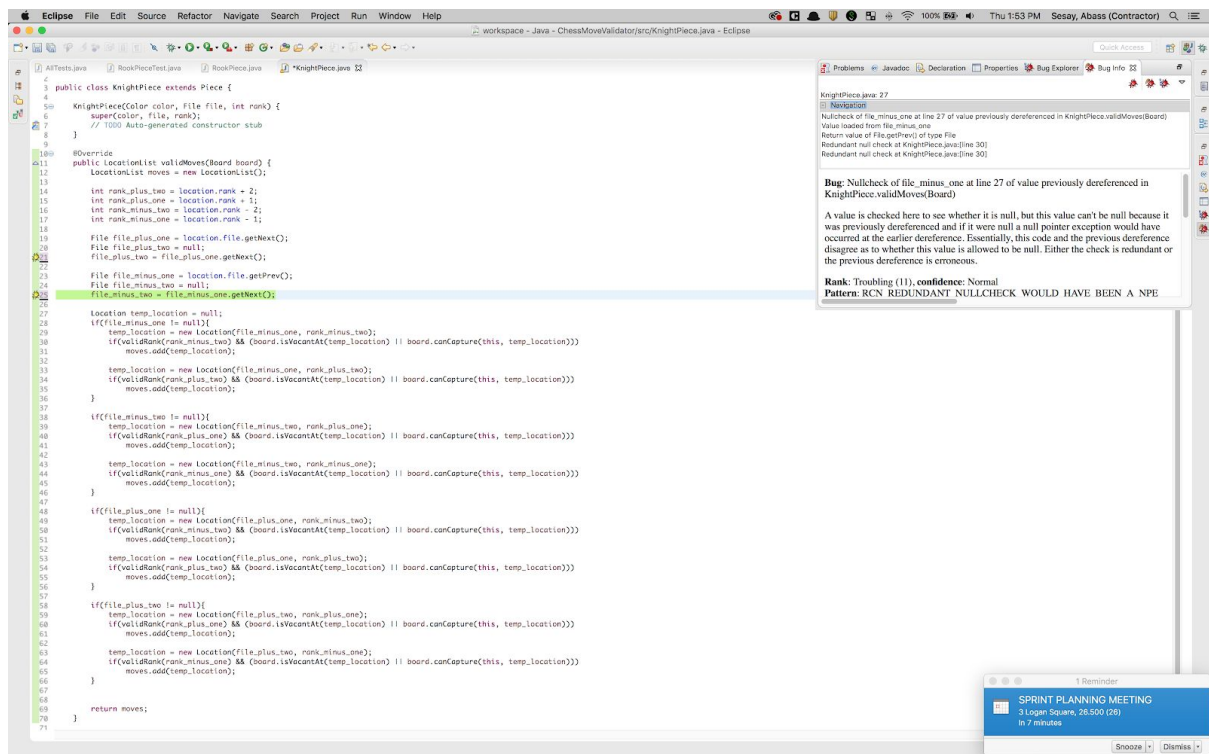
Codacy - automated code analysis/quality tool that helped with cyclomatic complexity, duplication and code unit test coverage changes in every commit.

## FindBugs - Used to identifies bugs within the codebase.

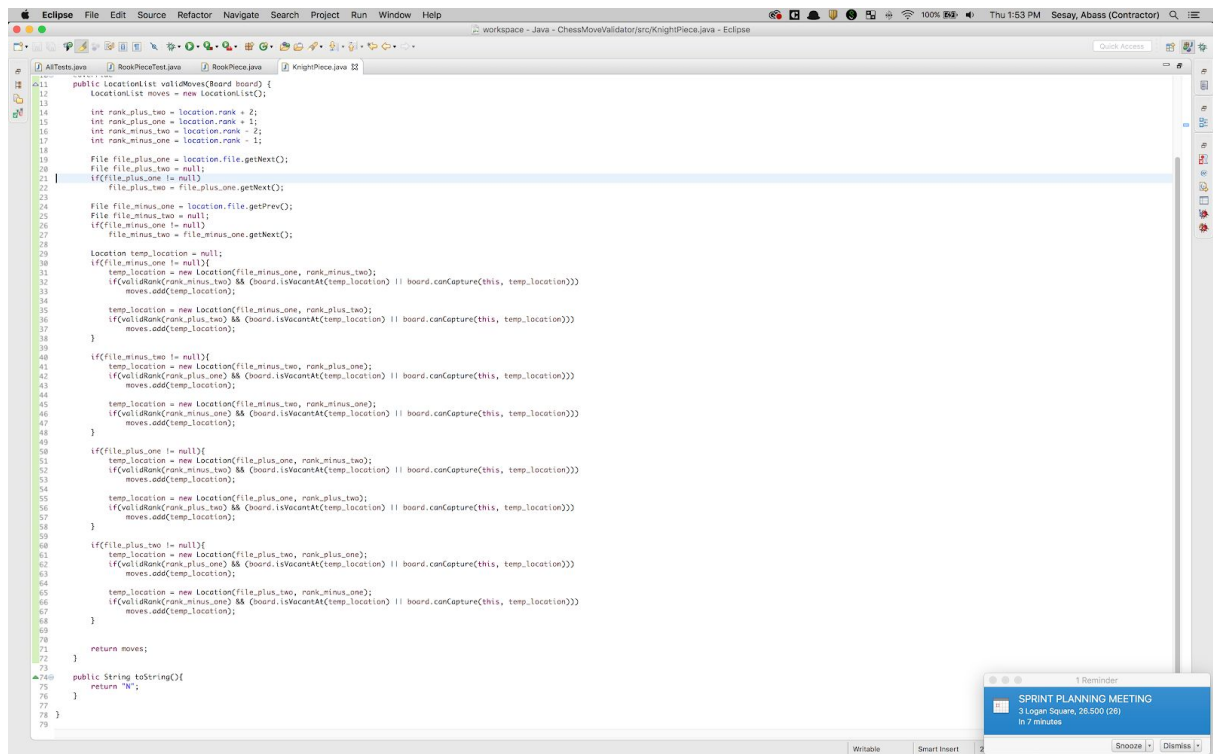


Above is an example of an error found by FindBugs. I override the equals method of the Location class but did not override the hashCode method. This is a bug that would create issue if I was to use Location in a HashMap or HashTable. This issue was fixed by overriding the HashCode functions.

## Another example of a bug found by FindBug is Nullcheck



This issue is fixed by checking the value before using it.

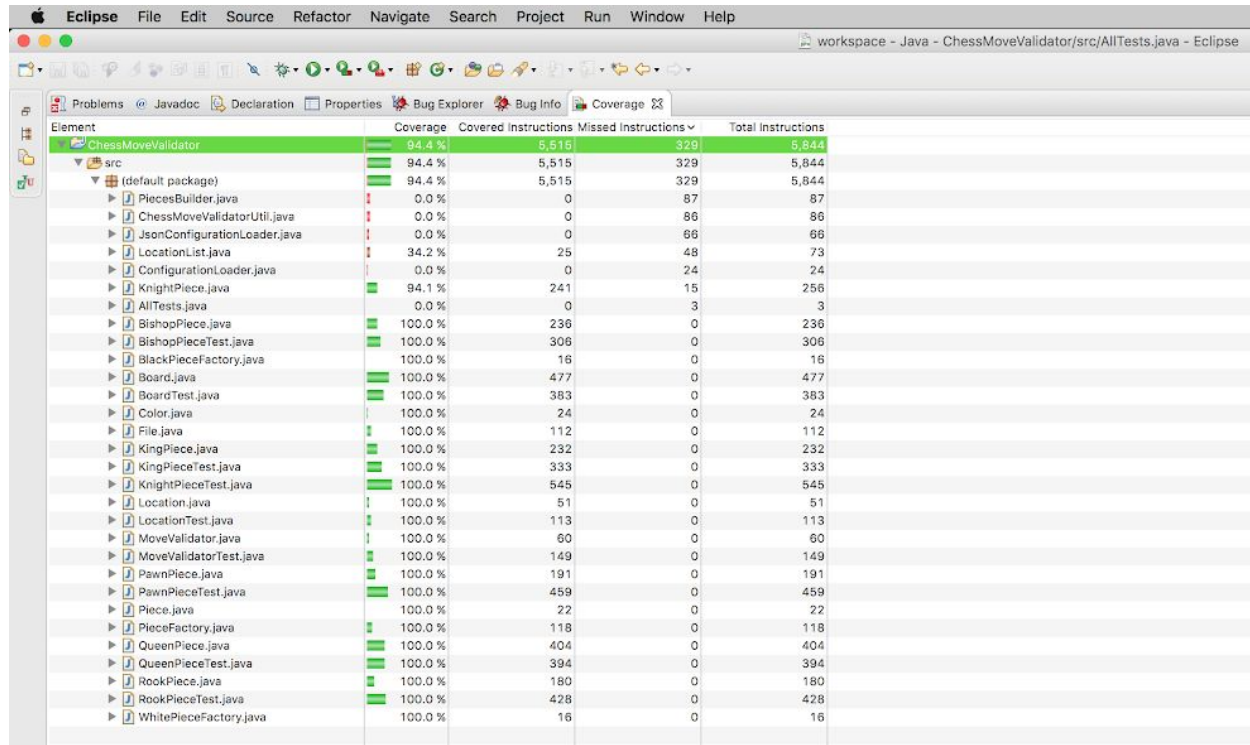


```
11 public LocationList validMoves(Board board) {
12     LocationList moves = new LocationList();
13
14     int rank_plus_two = location.rank + 2;
15     int rank_plus_one = location.rank + 1;
16     int rank_minus_two = location.rank - 2;
17     int rank_minus_one = location.rank - 1;
18
19     File file_plus_one = location.file.getNext();
20     File file_plus_two = null;
21     if(file_plus_one != null)
22         file_plus_two = file_plus_one.getNext();
23
24     File file_minus_one = location.file.getPrev();
25     File file_minus_two = null;
26     if(file_minus_one != null)
27         file_minus_two = file_minus_one.getPrev();
28
29     Location temp_location = null;
30     if(file_minus_one != null){
31         temp_location = new Location(file_minus_one, rank_minus_two);
32         if(validRank(rank_minus_two) && (board.isVacantAt(temp_location) || board.canCapture(this, temp_location)))
33             moves.add(temp_location);
34
35         temp_location = new Location(file_minus_one, rank_plus_two);
36         if(validRank(rank_plus_two) && (board.isVacantAt(temp_location) || board.canCapture(this, temp_location)))
37             moves.add(temp_location);
38     }
39
40     if(file_minus_two != null){
41         temp_location = new Location(file_minus_two, rank_plus_one);
42         if(validRank(rank_plus_one) && (board.isVacantAt(temp_location) || board.canCapture(this, temp_location)))
43             moves.add(temp_location);
44
45         temp_location = new Location(file_minus_two, rank_minus_one);
46         if(validRank(rank_minus_one) && (board.isVacantAt(temp_location) || board.canCapture(this, temp_location)))
47             moves.add(temp_location);
48     }
49
50     if(file_plus_one != null){
51         temp_location = new Location(file_plus_one, rank_minus_two);
52         if(validRank(rank_minus_two) && (board.isVacantAt(temp_location) || board.canCapture(this, temp_location)))
53             moves.add(temp_location);
54
55         temp_location = new Location(file_plus_one, rank_plus_two);
56         if(validRank(rank_plus_two) && (board.isVacantAt(temp_location) || board.canCapture(this, temp_location)))
57             moves.add(temp_location);
58     }
59
60     if(file_plus_two != null){
61         temp_location = new Location(file_plus_two, rank_plus_one);
62         if(validRank(rank_plus_one) && (board.isVacantAt(temp_location) || board.canCapture(this, temp_location)))
63             moves.add(temp_location);
64
65         temp_location = new Location(file_plus_two, rank_minus_one);
66         if(validRank(rank_minus_one) && (board.isVacantAt(temp_location) || board.canCapture(this, temp_location)))
67             moves.add(temp_location);
68     }
69
70     return moves;
71 }
72
73 public String toString(){
74     return "K";
75 }
76 }
77
78 }
```

## EclEmma - JaCoCo Code coverage tool.

This tool was used to determine the extent to which the code base is tested. I would admit here that the test cases were written after the code was written. This was primarily because the design was changed multiple times -- attributed to the fact that I don't know how to play chess.

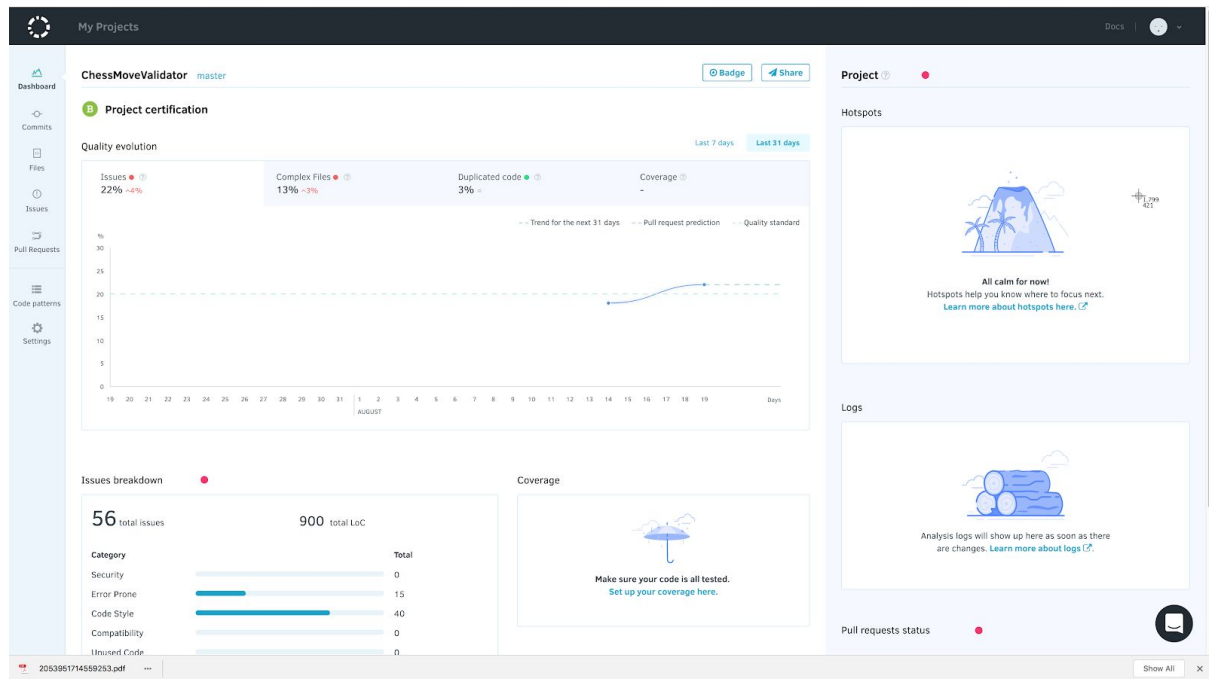
At the time of writing this report code coverage is at 94.4%. The pieces that are currently remaining are mainly utility components.



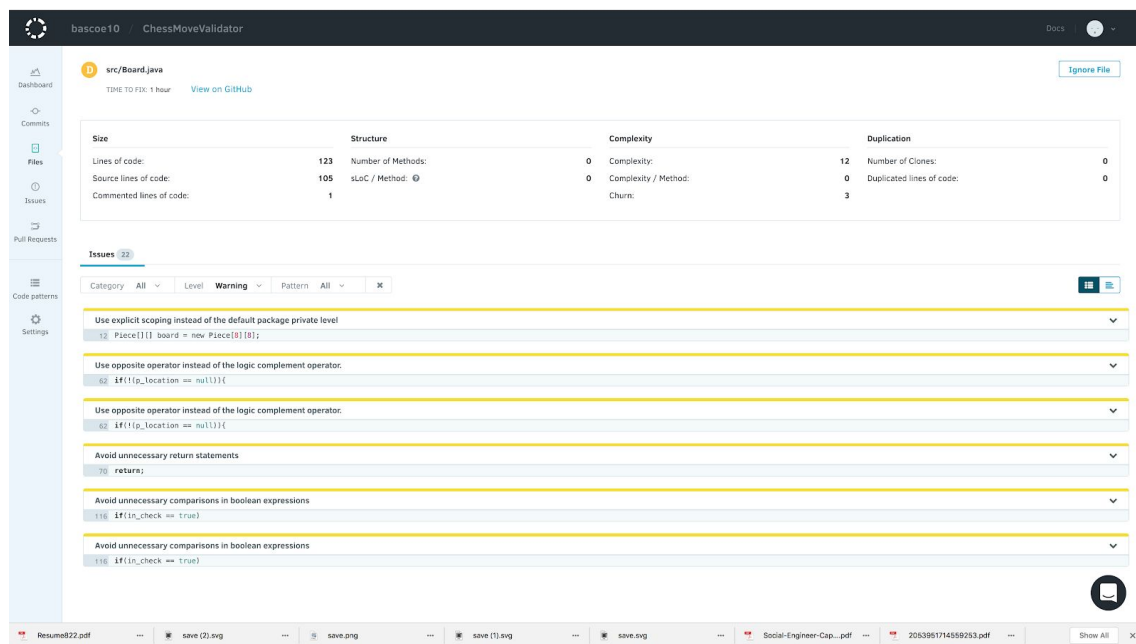
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
ChessMoveValidator	94.4 %	5,515	329	5,844
src	94.4 %	5,515	329	5,844
(default package)	94.4 %	5,515	329	5,844
PiecesBuilder.java	0.0 %	0	87	87
ChessMoveValidatorUtil.java	0.0 %	0	86	86
JsonConfigurationLoader.java	0.0 %	0	66	66
LocationList.java	34.2 %	25	48	73
ConfigurationLoader.java	0.0 %	0	24	24
KnightPiece.java	94.1 %	241	15	256
AllTests.java	0.0 %	0	3	3
BishopPiece.java	100.0 %	236	0	236
BishopPieceTest.java	100.0 %	306	0	306
BlackPieceFactory.java	100.0 %	16	0	16
Board.java	100.0 %	477	0	477
BoardTest.java	100.0 %	383	0	383
Color.java	100.0 %	24	0	24
File.java	100.0 %	112	0	112
KingPiece.java	100.0 %	232	0	232
KingPieceTest.java	100.0 %	333	0	333
KnightPieceTest.java	100.0 %	545	0	545
Location.java	100.0 %	51	0	51
LocationTest.java	100.0 %	113	0	113
MoveValidator.java	100.0 %	60	0	60
MoveValidatorTest.java	100.0 %	149	0	149
PawnPiece.java	100.0 %	191	0	191
PawnPieceTest.java	100.0 %	459	0	459
Piece.java	100.0 %	22	0	22
PieceFactory.java	100.0 %	118	0	118
QueenPiece.java	100.0 %	404	0	404
QueenPieceTest.java	100.0 %	394	0	394
RookPiece.java	100.0 %	180	0	180
RookPieceTest.java	100.0 %	428	0	428
WhitePieceFactory.java	100.0 %	16	0	16

**Codacy** - automated code analysis/quality tool that helped with cyclomatic complexity, duplication and code unit test coverage changes in every commit.

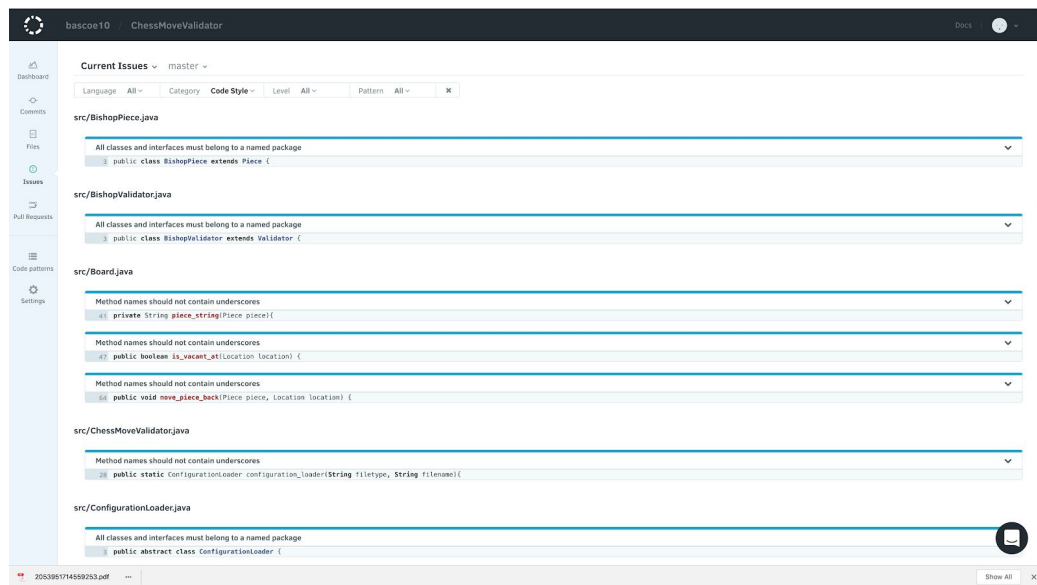
Codacy provided a more comprehensive view of the the project. Codacy gives warning not only on potential runtime bugs but also flag things that violate design principles.



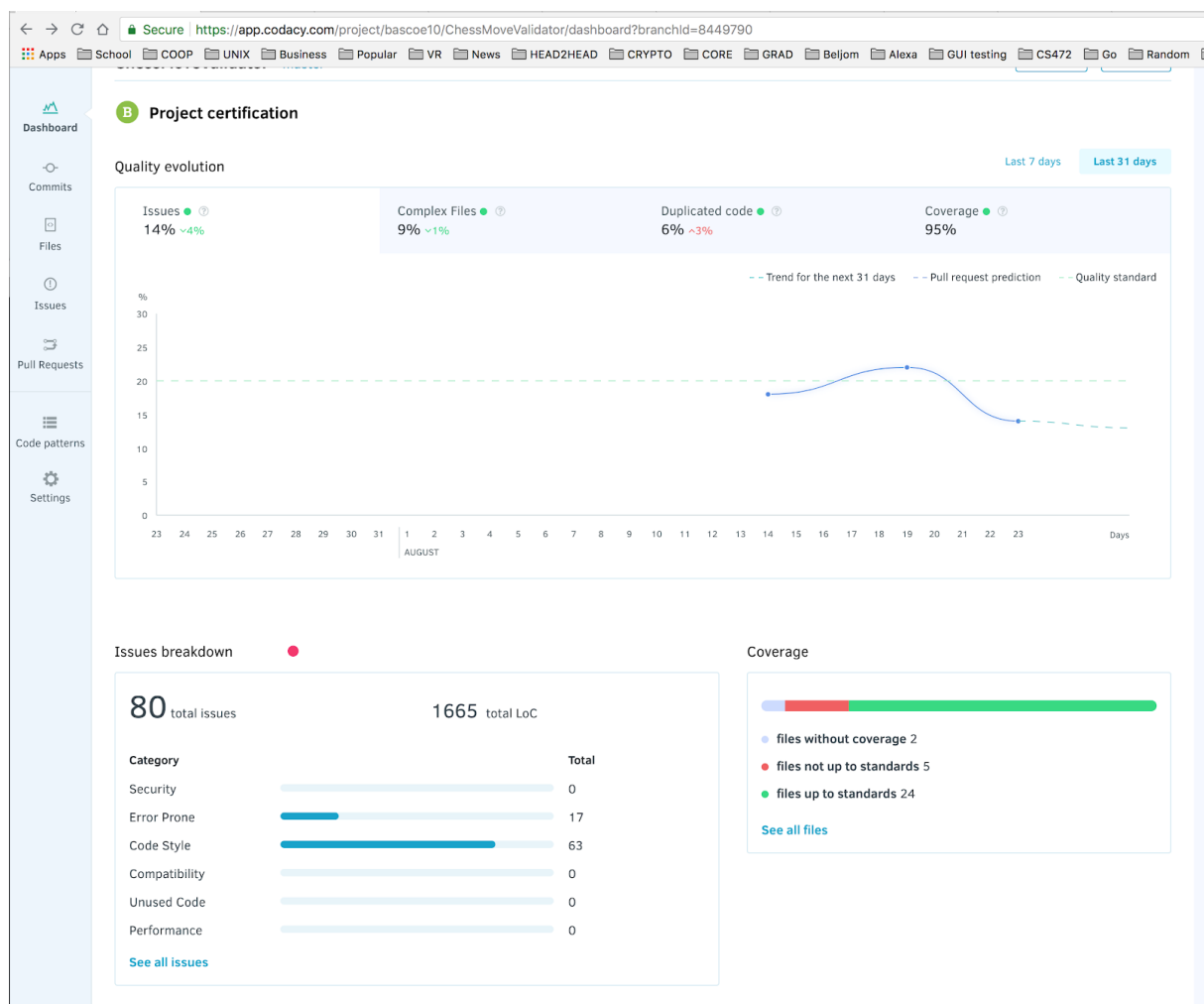
## Codacy Dashboard



## Warning errors



Styling error



After adding test and refactoring some of the reported errors.

# Final Codacy Report - after refactoring

