Systeemtestplan



**Opdrachtgever**:  Fontys **Project**: Big idea - Twitter

**Versie**:       1.0 **Project** **Team** : Tweeta

**Versiedatum**:      13 november 2019 **Teamleden**: Joshua Keulers

**Status**:        Concept

# Documenthistorie

|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Wijzigingen** | **Auteur** | **Datum** |
| 1.0 | Opzet document | Joshua Keulers | 13 november 2019 |
| 1.1 | Aanpassingen … | Joshua Keulers | 2 januari 2020 |

# H1 Inleiding

Voor mijn big Idea wil ik Twitter gaan maken, hierbij zullen de gebruikers elkaar kunnen volgen en dus up-to-date blijven met elkaars activiteiten( zoals bijvoorbeeld: tweets, retweets en comments).

Twitter moet gedistribueerd zijn en dit ga ik doen d.m.v. web sockets en REST api. In dit document ga ik vaststellen waar de risico’s in mijn project zitten en op basis van dit risico en hoe belangrijk de klasse is ga ik hem testen. Dit doe ik zodat ik niet voor onverwachte verassingen kom te staan.

Inhoud

[Documenthistorie 2](#_Toc24527275)

[H1 Inleiding 2](#_Toc24527276)

[H2 Uitgangssituatie 4](#_Toc24527277)

[H3 Productrisicoanalyse 4](#_Toc24527278)

[H4 Teststrategie 4](#_Toc24527279)

[H5 Logische testcases 4](#_Toc24527280)

[H6 Fysieke testcases 4](#_Toc24527281)

[H7 Testcoverage 4](#_Toc24527282)

[H8 Unittesten en code coverage 4](#_Toc24527283)

[H9 Statische code analyse 4](#_Toc24527284)

[H10 Conclusie 4](#_Toc24527285)

# H2 Uitgangssituatie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versie | test data | Soort test | Resultaat | Datum |
| 1.0 |  |  |  |  |

# H3 Productrisicoanalyse

De risicoklasse (RK) is vervolgens bepalend voor de zwaarte van de test. Hierbij is risicoklasse A de hoogste risicoklasse en C de laagste. De teststrategie is er op gericht om de risico’s met de hoogste risicoklasse zo vroeg mogelijk in het testtraject af te dekken. Productrisicoanalyse (PRA) wordt bepaald door per testdoel te bepalen: faalkans, schade, risicoklasse en ernstklasse. De risicoklasse is gelijk aan schade \* faalkans. De ernstklasse is gebaseerd op de risicoklasse. Geef aan hoe de ernstklasse wordt bepaald. Bijvoorbeeld:

*Schade: 1 = Laag, 2 = Midden, 3 = Hoog*

*Faalkans: 1 = Laag, 2 = Midden, 3 = Hoog*

*Risicoklasse = Schade \* Faalkans*

*Ernstklasse C: risicoklasse ≤ 2*

*Ernstklasse B: 2 &lt; risicoklasse ≤ 6*

*Ernstklasse A: risicoklasse &gt; 6*

Alle acties en regels kunt u terugvinden in het analyse document.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirements ID** | **Schade** | **Faalkans** | **Risicoklasse** | **Ernstklasse** |
| 1.A | 2 | 1 | 2 | C |
| 2.A | 2 | 2 | 4 | B |
| 3.A | 1 | 1 | 1 | C |
| 1.U | 2 | 1 | 2 | C |
| 2.U | 2 | 1 | 2 | C |
| 3.U | 3 | 1 | 3 | B |
| 4.U | 2 | 2 | 4 | B |
| 5.U | 3 | 1 | 3 | B |
| 1.T | 3 | 3 | 9 | A |
| 2.T | 3 | 2 | 6 | B |
| 3.T | 2 | 1 | 2 | C |
| 4.T | 3 | 3 | 9 | A |
| 5.T | 2 | 2 | 4 | B |

# H4 Teststrategie

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testdoel | PRA | UT | CT | IT | ST handm. | ST auto |
| 1.A  Registreren | C | ⬤ |  |  | ⬤ |  |
| 2.A  Inloggen | B | ⬤ |  |  | ⬤ ⬤ |  |
| 3.A  Account verwijderen | C |  |  |  | ⬤ |  |
| 1.U  Profiel aanmaken | C | ⬤ |  |  | ⬤ |  |
| 2.U  Profiel  Aanpassen | C | ⬤ |  |  | ⬤ |  |
| 3.U  User volgen | B | ⬤ ⬤ |  |  |  | ⬤ |
| 4.U  User opzoeken | B | ⬤ ⬤ |  |  |  | ⬤ |
| 5.U  Chatten | B | ⬤ ⬤ |  |  |  | ⬤ |
| 1.T  Tweeten | A | ⬤ ⬤ ⬤ |  |  |  | ⬤ |
| 2.T  Reageer op tweet | B | ⬤ ⬤ |  |  |  | ⬤ |
| 3.T  Tweet verwijderen | C | ⬤ |  |  | ⬤ |  |
| 4.T  Tweet aanpassen | A | ⬤ ⬤ ⬤ |  |  |  |  |
| 5.T  Tweet liken | B |  |  |  | ⬤ ⬤ |  |

# 

# H5 Logische testcases

|  |  |
| --- | --- |
| ID | Omschrijving |
| TC 1.1 | Een gebruiker registreert zich |
| TC 1.2 | Een gebruiker wil zich registreren, maar vult niet alles in |
| TC 2.1 | Een gebruiker logt zich in |
| TC 2.2 | Een gebruiker probeert zich in te loggen, maar vult de verkeerde gegevens in |
| TC 2.3 | Een gebruiker wil inloggen, maar vergeet één veld in te vullen |
| TC 3.1 | Een gebruiker volgt een andere gebruiker |
| TC 3.2 | Een gebruiker stuurt één tweet |
| TC 3.3 | Een gebruiker reageert op een tweet |
| TC 3.4 | Een gebruiker stuurt een tweet met verboden inhoud |
| TC 3.5 | Een gebruiker tagt een van zijn volgende gebruikers |
| TC 3.6 | Een gebruiker Liked een tweet |
| TC 4.1 | Een gebruiker stuurt een bericht via een chat |

# H6 Fysieke testcases

|  |  |
| --- | --- |
|  | De test is geslaagd en er is niks op aan te merken. |
|  | De test lijkt goed te gaan, maar er is iets op het behaalde resultaat aan te merken. |
|  | De test is gefaald. |
|  | Deze test is niet van toepassing op de huidige versie van de applicatie en is ook niet uitgevoerd. |
|  | Dit is geen test maar een actie die ondernomen moet worden om daar op volgende tests uit te kunnen voeren (initialisatie actie). |

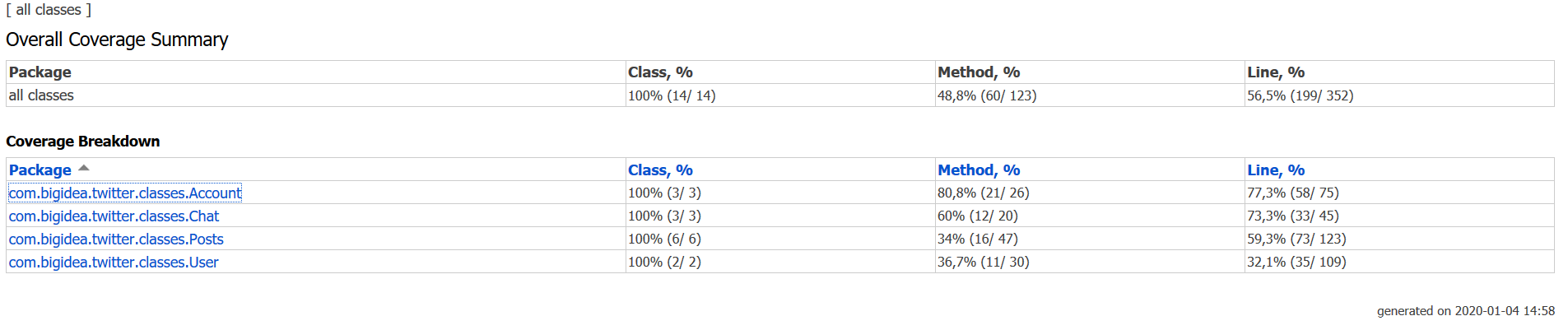
|  |  |  |  |
| --- | --- | --- | --- |
| ID | Aanname | Hypothese | Resultaat |
| TC 1.1 | Een gebruiker klikt op registreren en vult alle velden in | Een account wordt aangemaakt en de gebruiker heeft de mogelijkheid om in te loggen |  |
| TC 1.2 | Een gebruiker klikt op registreren, maar vergeet één veld in te vullen | De front-end checkt of alles Velden zijn ingevuld anders krijgt hij een melding |  |
| TC 2.1 | Een gebruiker vult zijn inloggegevens en klikt op inloggen | De gebruiker zal inloggen nar het startscherm gestuurd worden |  |
| TC 2.2 | Een gebruiker probeert zich in te loggen, maar vult de verkeerde gegevens in | Het systeem geeft een foutmelding terug dat de credentials niet kloppen |  |
| TC 2.3 | Een gebruiker wil inloggen, maar vergeet één veld in te vullen | De front-end checkt of alles Velden zijn ingevuld anders krijgt hij een melding |  |
| TC 3.1 | Een volger is ingelogd en zoekt een andere gebruiker op. | De gebruiker klikt op volgen en hierdoor wordt zijn tijdlijn t en zijn Tweetlijst geupdate. |  |
| TC 3.2 | Een gebruiker stuurt een tweet | Het systeem checkt de inhoud en keurt dit goed waardoor het bericht gestuurd word |  |
| TC 3.3 | Een gebruiker stuurt een tweet met verboden inhoud | Het systeem ziet dat de inhoud niet voldoet aan de eisen en geeft een melding aan de gebruiker dat hij op zijn taal moet letten |  |
| TC 3.4 | Een gebruiker reageert op een tweet | Het systeem checkt op de inhoud en update vervolgens zijn eigen tijdlijn en stuurt een notificatie naar de andere gebruiker |  |
| TC 3.5 | Een gebruiker tagt een van zijn volgende gebruikers | Hierdoor wordt de andere gebruiker op de hoogte gesteld en kan hij zien waar het over gaat |  |
| TC 3.6 | Een gebruiker liked een tweet | De tweet wordt geupdate en de gebruiker van de tweet wordt ook op de hoogte gesteld |  |
| TC 4.1 | Een gebruiker stuurt een bericht via een chat | Er wordt een chatbox aangemaakt |  |

# H7 Testcoverage

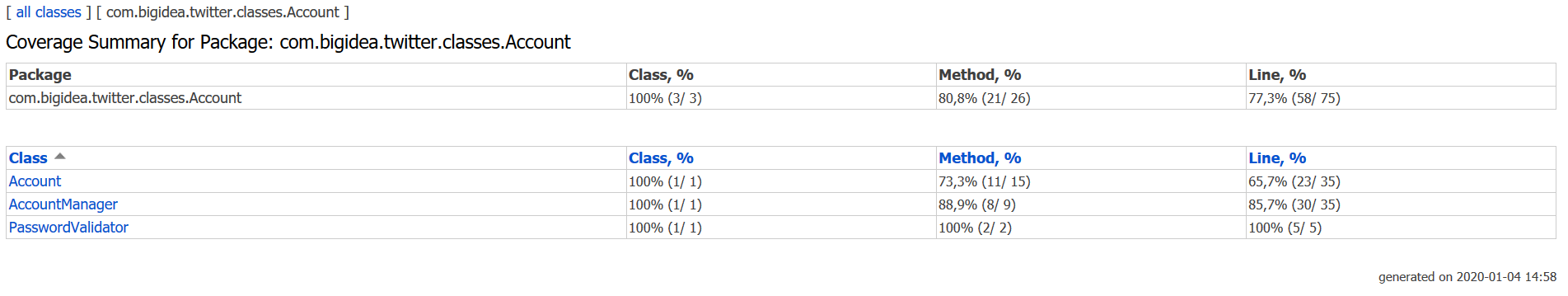
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TC  1.1 | TC  1.2 | TC  2.1 | TC  2.2 | TC  2.3 | TC  3.1 | TC  3.2 | TC  3.3 | TC  3.4 | TC  3.5 | TC  3.6 | TC  4.1 |
| 1.A | X | X |  |  |  |  |  |  |  |  |  |  |
| 2.A |  |  | X | X | X |  |  |  |  |  |  |  |
| 3.A |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.U | X | X |  |  |  |  |  |  |  |  |  |  |
| 2.U |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.U |  |  |  |  |  | X |  |  |  |  |  |  |
| 4.U |  |  |  |  |  | X |  |  |  |  |  |  |
| 5.U |  |  |  |  |  |  |  |  |  |  |  | X |
| 1.T |  |  |  |  |  |  | X | X |  | X |  |  |
| 2.T |  |  |  |  |  |  |  | X | X | X |  |  |
| 3.T |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.T |  |  |  |  |  |  | X | X |  |  |  |  |
| 5.T |  |  |  |  |  |  |  |  |  |  | X |  |

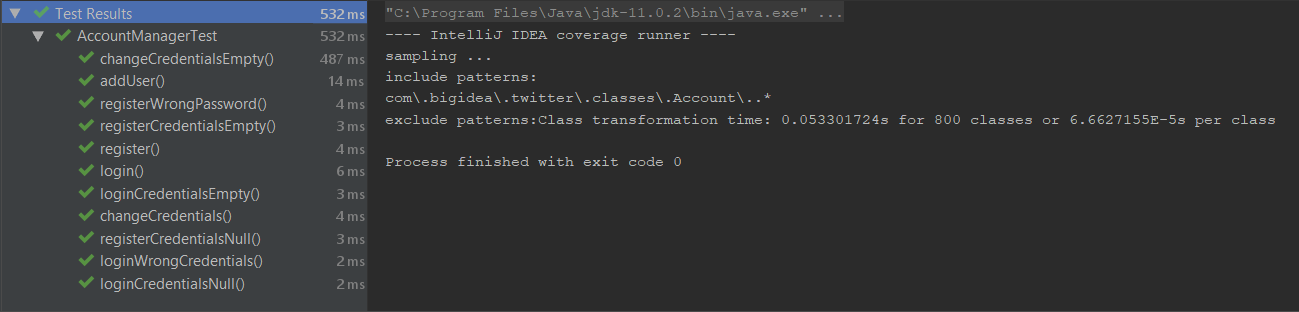
# H8 Unittesten en code coverage

De unittesten zullen zich richten op de manager classes. Dit zijn Account-, Chat-, User- en Post Manager. De classes bevatten elke voor hun eigen onderwerp de logica en bestaan uit vrijwel alleen maar methodes en vandaar dat ik hun gekozen heb.

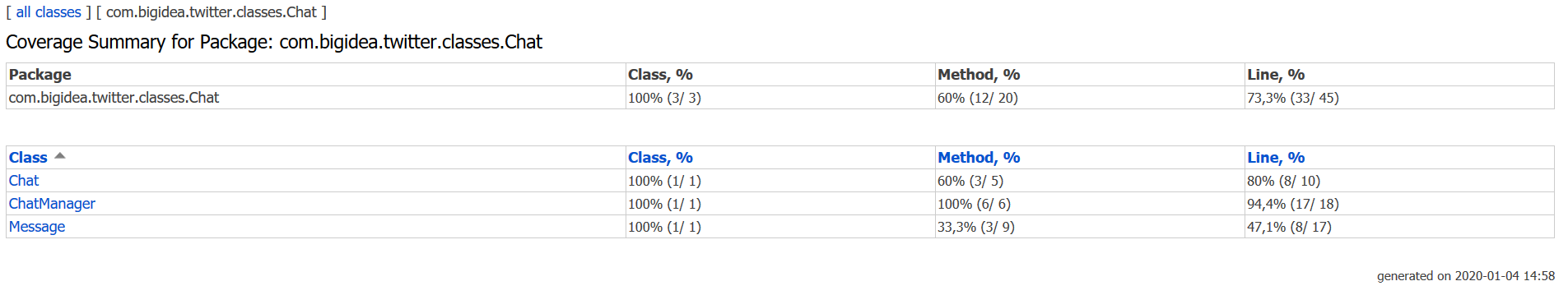


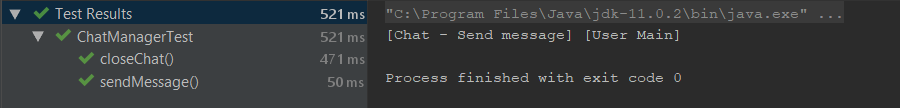
## Account



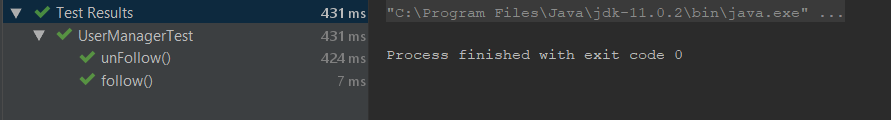
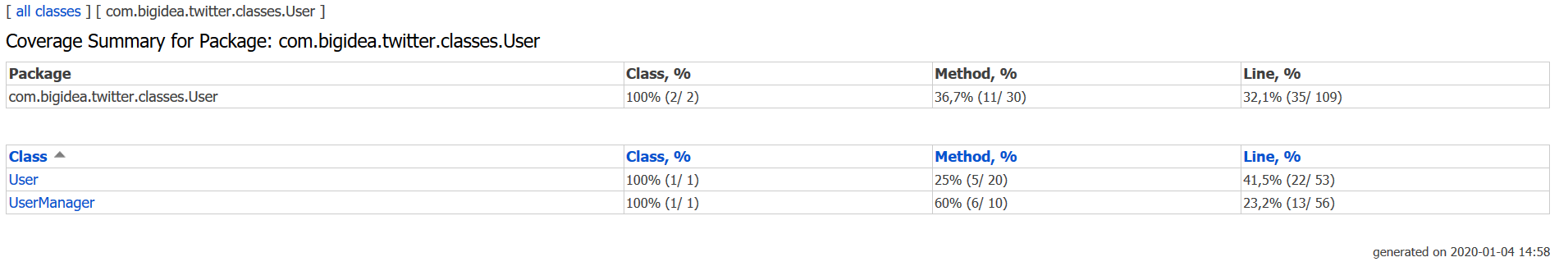


## Chat

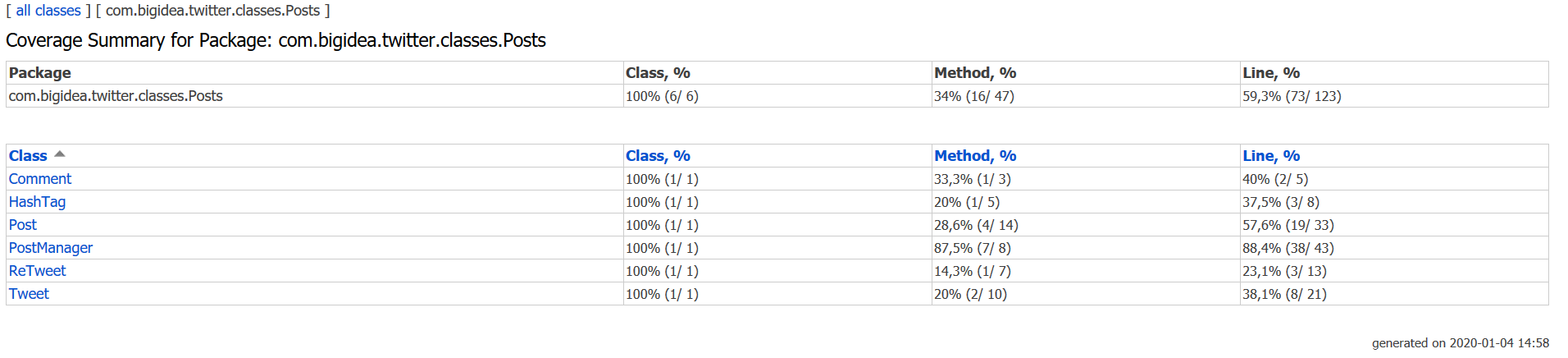


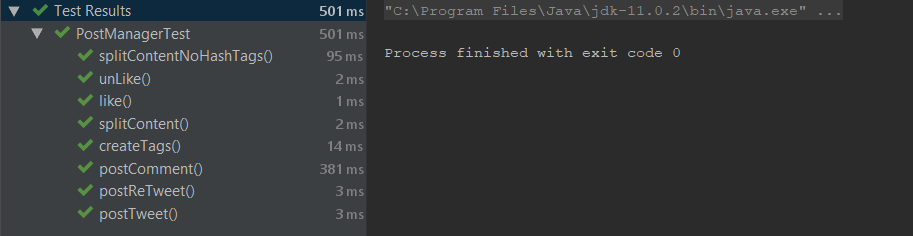


## User



## Post





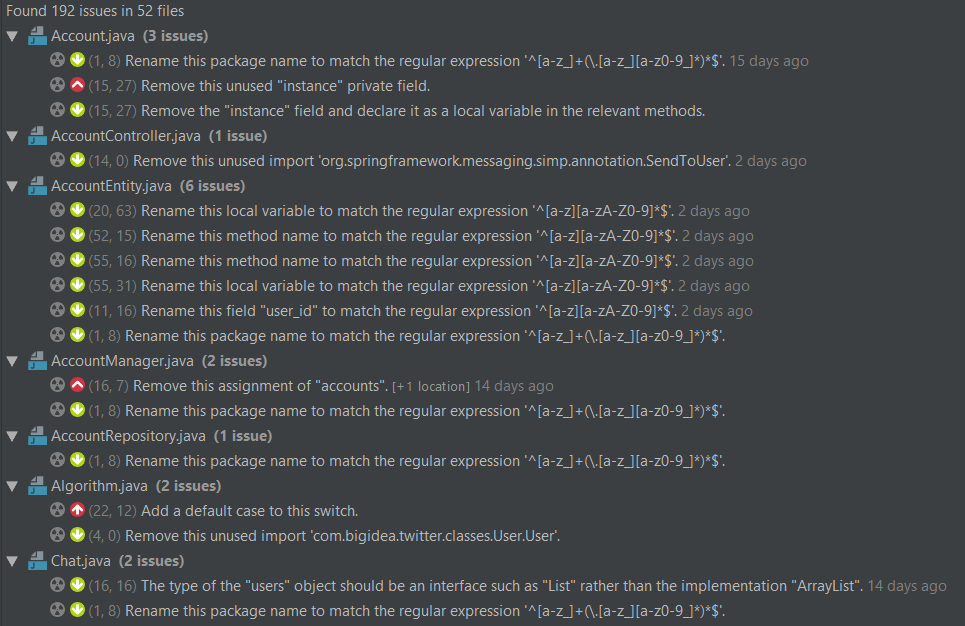
## Conclusie

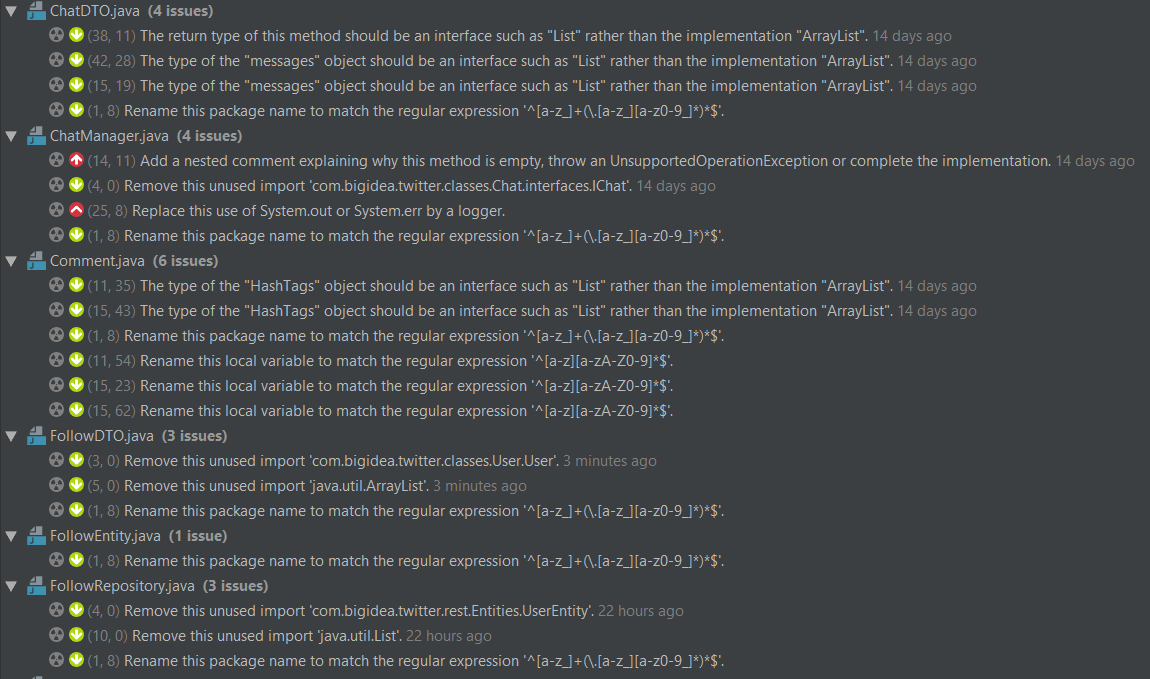
Op dit moment heb ik een al een aantal unittesten en heb ik daarmee al een hoop kunnen testen. Tijdens het programmeren heb ik nog een aantal andere functies toegevoegd aan de manager classes. Hiervoor heb ik alleen nog niet de tijd gehad om deze te kunnen testen, maar dit zal ik bij de volgende iteratie doen.

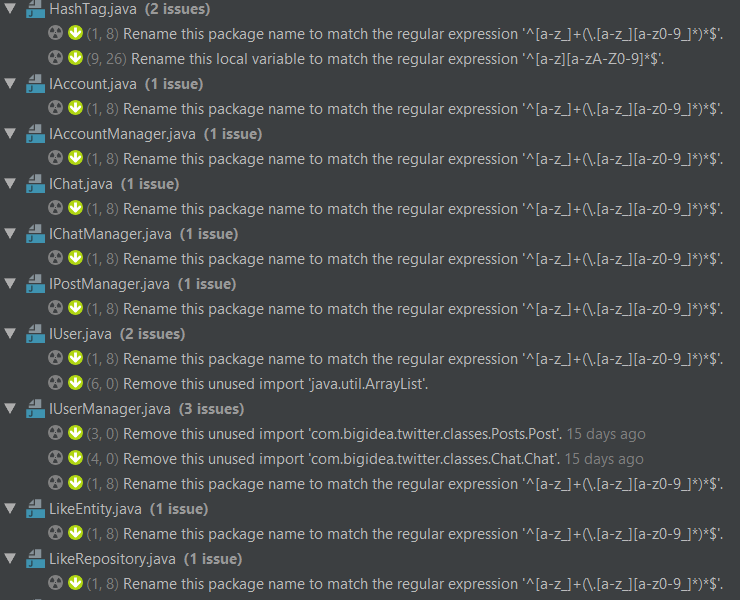
Als u kijkt naar mijn code coverage ziet u dat alle andere classes weinig getest zijn, maar dit zijn over het algemeen de properties van het object en dit hoeft niet getest te worden.

# H9 Statische code analyse

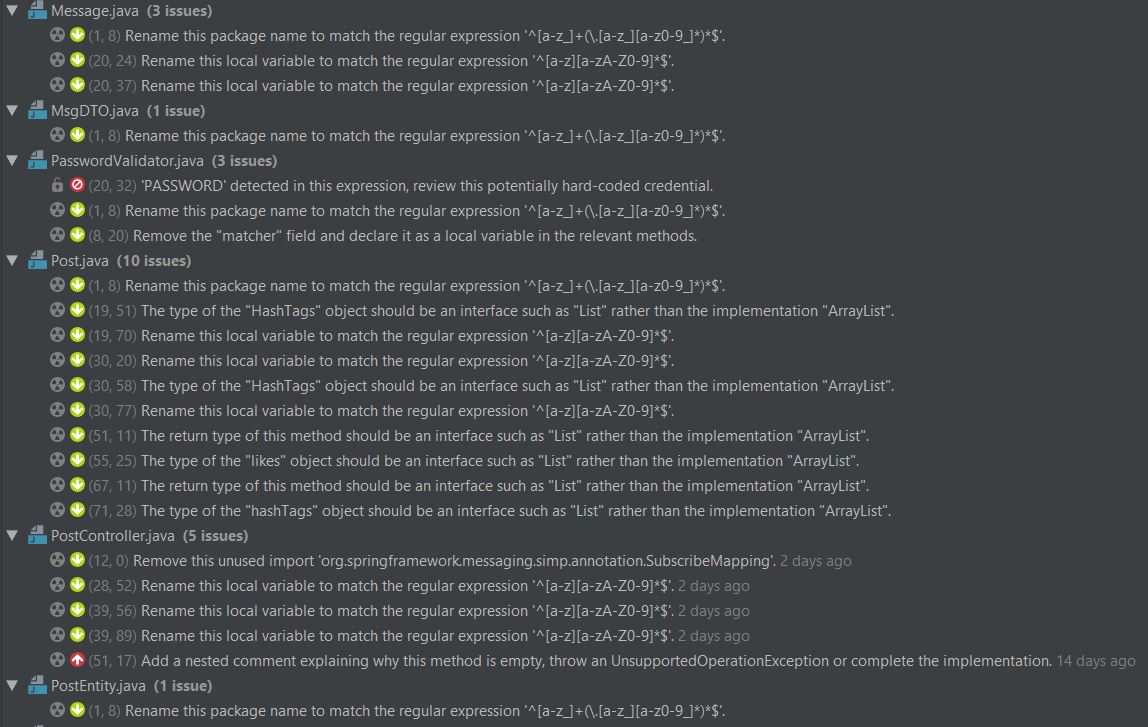
Het is mij nog niet gelukt mijn spring boot project te koppelen aan sonarqube vanwege problemen met het werkend krijgen van sonarqube server. Vandaar dat ik voor nu gekozen heb om met solarLint te werken om zo alsnog een code analyse te kunnen doen. Voor de volgende iteratie zal ik proberen sonarqube te koppelen.

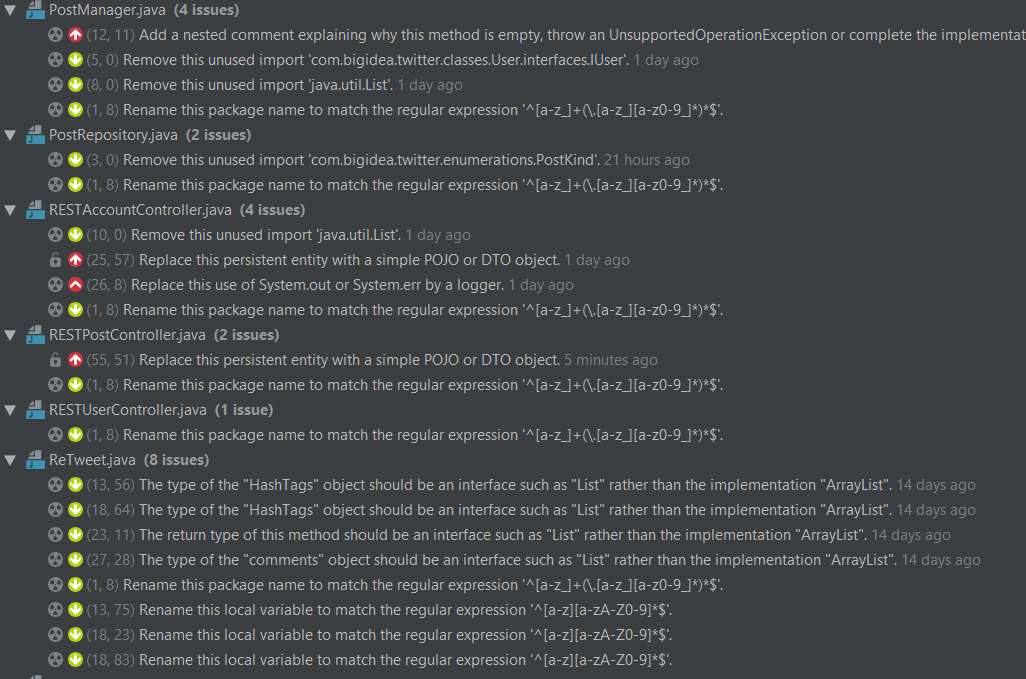


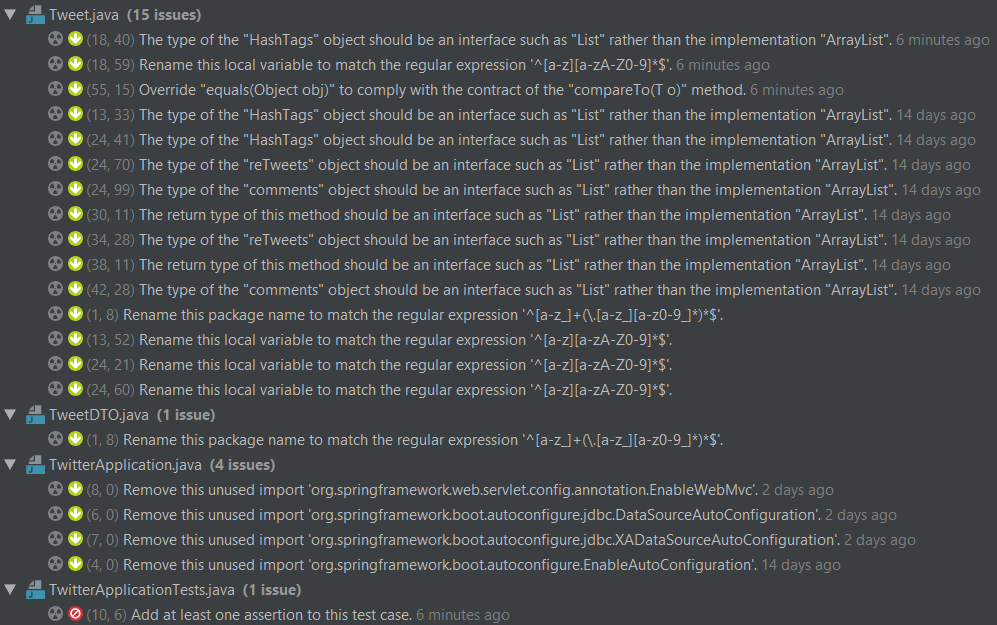


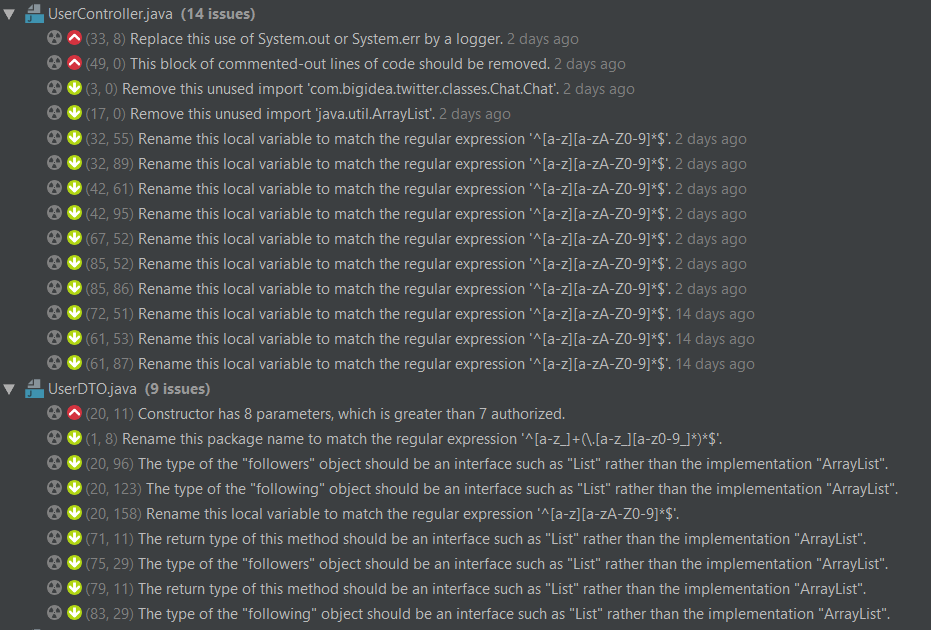












Zoals te zien is op de bovenstaande foto’s, zijn de meeste fouten in de benamingen of dat ik een List i.p.v. een Array List moet gebruiken. Bij mijn volgende iteratie zal de fouten eruit gaan halen. Deze keer heb ik dit niet gedaan omdat er nog veel veranderd kan worden.

# H10 Conclusie