Testrapport GamingLads

# Versiebeheer

|  |  |  |
| --- | --- | --- |
| **Auteur** | **Datum** | **Toevoeging/verandering** |
| S. Braam | 6-1-2021 | Aanmaken document & introductie geschreven |
| S.Braam | 12-1-2021 | Unit tests en component testen uitgelegd en toegevoegd |

# Introductie

## Context

Dit testrapport heeft betrekking op het project GamingLads. GamingLads is een applicatie die gebasseerd is op Tinder, maar dan op zoek gaat naar de ideale gamepartner gemaakt in semester drie van de software opleiding te Fontys Hogescholen ICT in Eindhoven. Dit testrapport is opgesteld met als doel de kwaliteit van het project aan te tonen en inzicht te scheppen over de code-base van het desbetreffende project.

## Afkortingen en definities

### Defenities

|  |  |
| --- | --- |
| **Term** | **Definitie** |
| Tester | Sharony Braam |
| Teamleden | Sharony Braam |
| Product Owner | Sharony Braam |

### Afkortingen

|  |  |
| --- | --- |
| **Afkorting** | **Definitie** |
| US | User Story |
| CT | Component Test |
| UT | Unit Test |
| DSS | Distributed Software System |

### Referenties

|  |  |
| --- | --- |
| **Referentie** | **Link/verwijzing** |
| Front-end User | https://github.com/Sharr15/GamingLads-Frontend-User |
| Front-end Admin | https://github.com/Sharr15/GamingLads-Frontend-Admin |
| Api Gateway | https://github.com/Sharr15/gaminglads-gateway |
| Discovery-Server | https://github.com/Sharr15/GamingLads-discovery-server |
| Chat-Service | https://github.com/Sharr15/GamingLads-chat-service |
| User-Service | https://github.com/Sharr15/GamingLads-user-service |
| Profile-Service | https://github.com/Sharr15/GamingLads-profile-service |

## Samenvatting

GamingLads is gemaakt volgens de Distributed Software System architectuur. Dit manifesteert zich in de onderverdeling van het gehele project in meerdere projecten die onderling samenwerken. De front-end applicaties communiceren via een API Gateway met de verschillende services. De discovery service zorgt ervoor dat de API Gateway weet naar welke service het request moet worden verstuurd.

De gebruiker maakt een account aan en logt daarna ermee in. Daarna komt de gebruiker op het hoofdscherm waarbij hij/zij een mening kan geven over een andere gamer. Als ze gematch zijn met elkaar, kan er gechat worden netzoals bij Tinder. Verder kan de eigen profiel nog worden aangepast, denk aan welke games gespeeld worden en op welk level.

Contents

[Overzicht 5](#_Toc61944706)

[Unit tests 5](#_Toc61944707)

[Front-end User 5](#_Toc61944708)

[Front-end Admin 5](#_Toc61944709)

[Api Gateway 5](#_Toc61944710)

[Discovery-Server 5](#_Toc61944711)

[Chat-Service 5](#_Toc61944712)

[User-Service 5](#_Toc61944713)

[Profile-Service 5](#_Toc61944714)

[Component tests 6](#_Toc61944715)

[Api Gateway 6](#_Toc61944716)

[Discovery-Server 6](#_Toc61944717)

[Chat-Service 6](#_Toc61944718)

[User-Service 6](#_Toc61944719)

[Profile-Service 6](#_Toc61944720)

[Integration Tests 6](#_Toc61944721)

[Front-end User 6](#_Toc61944722)

[Front-end Admin 7](#_Toc61944723)

[Api Gateway 7](#_Toc61944724)

[Discovery-Server 7](#_Toc61944725)

[Chat-Service 7](#_Toc61944726)

[User-Service 7](#_Toc61944727)

[Profile-Service 7](#_Toc61944728)

[Conclusie 7](#_Toc61944729)

# Overzicht

Binnen GamingLads zijn er door de tester meerdere soorten tests geschreven, namelijk unit tests, component tests en enkele integration tests. Deze worden automatisch getest in de pipelines die bij elke service is toegevoegd. Verder worden de resultaten online gezet in SonarCloud. Maar de front-end functionaliteiten zijn handmatig getest.

## Unit tests

Front-end User

Front-end User heeft geen unit tests. Alle testen zijn handmatig uitgevoerd i.v.m. de tijd om front-end testing te leren, ook is er nog niet veel functionaliteit en daarom is het dus onnodig.

De user stories die op dit moment getest zijn, zijn “Als gebruiker wil ik een account aan kunnen maken”, “Als gebruiker wil ik in kunnen loggen” en “Als gebruiker wil ik een profiel hebben”. Dit zijn de grootste functionaliteiten op dit moment in GamingLads.

US: “Als gebruiker wil ik een account aan kunnen maken” en “Als gebruiker wil ik een profiel hebben”. Deze worden samen gepakt omdat als iemand een account aanmaakt, zij standaard een profiel krijgen. Dit is dan ook het resultaat.

US: “Als gebruiker wil ik in kunnen loggen” heeft als resultaat dat de gebruiker in kan loggen en een jwt meegestuurd krijgt.

Front-end Admin

Front-end Admin heeft geen testen omdat de front-end nog niet van toepassing is gekomen.

Api Gateway

Api Gateway heeft geen testen omdat daar niks te testen is.

Discovery-Server

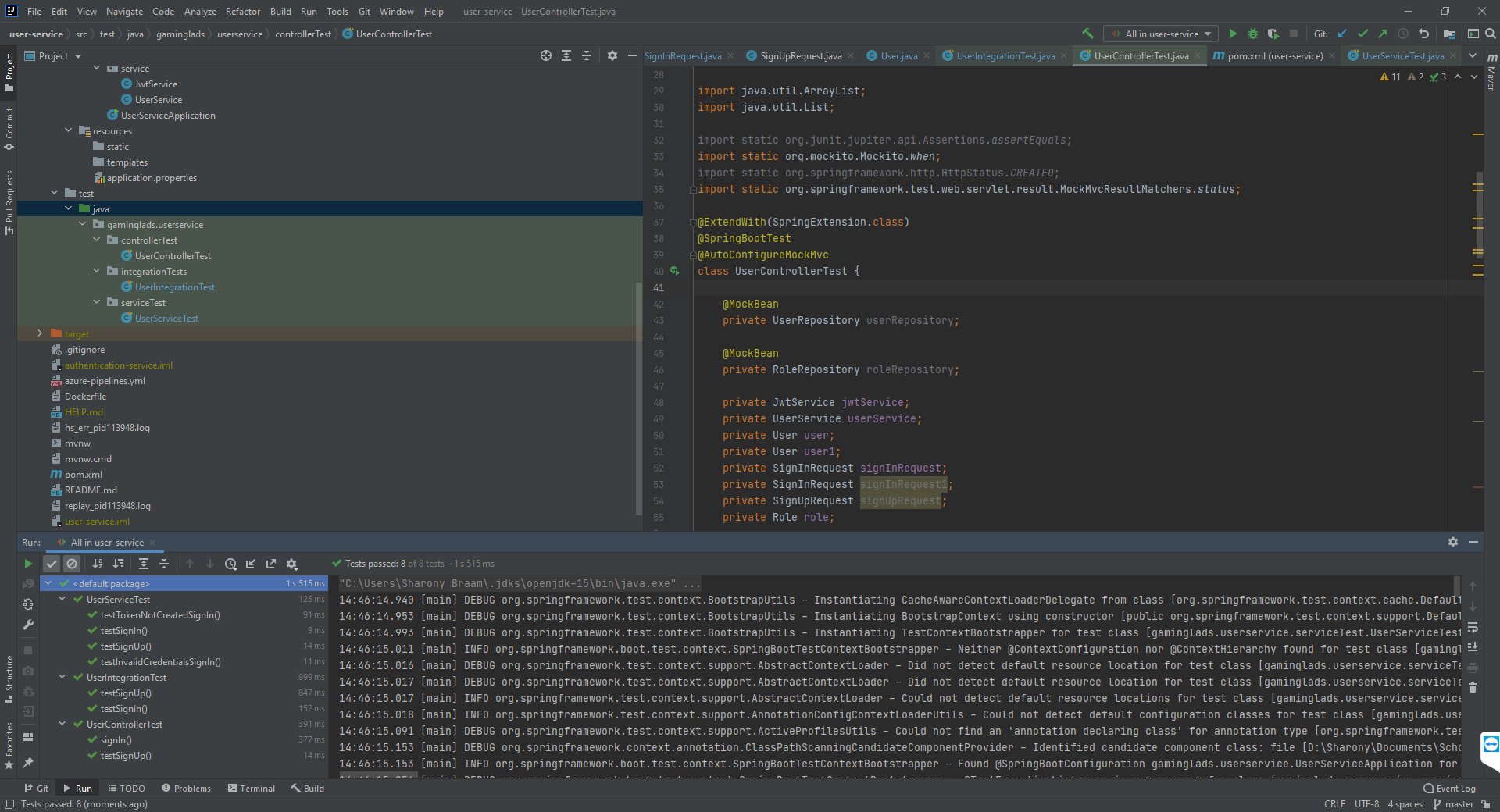
Discovery-Server heeft geen testen omdat daar niks te testen is.

Chat-Service

Chat-Service heeft geen testen omdat Chat-Service nog niet van toepassing is gekomen.

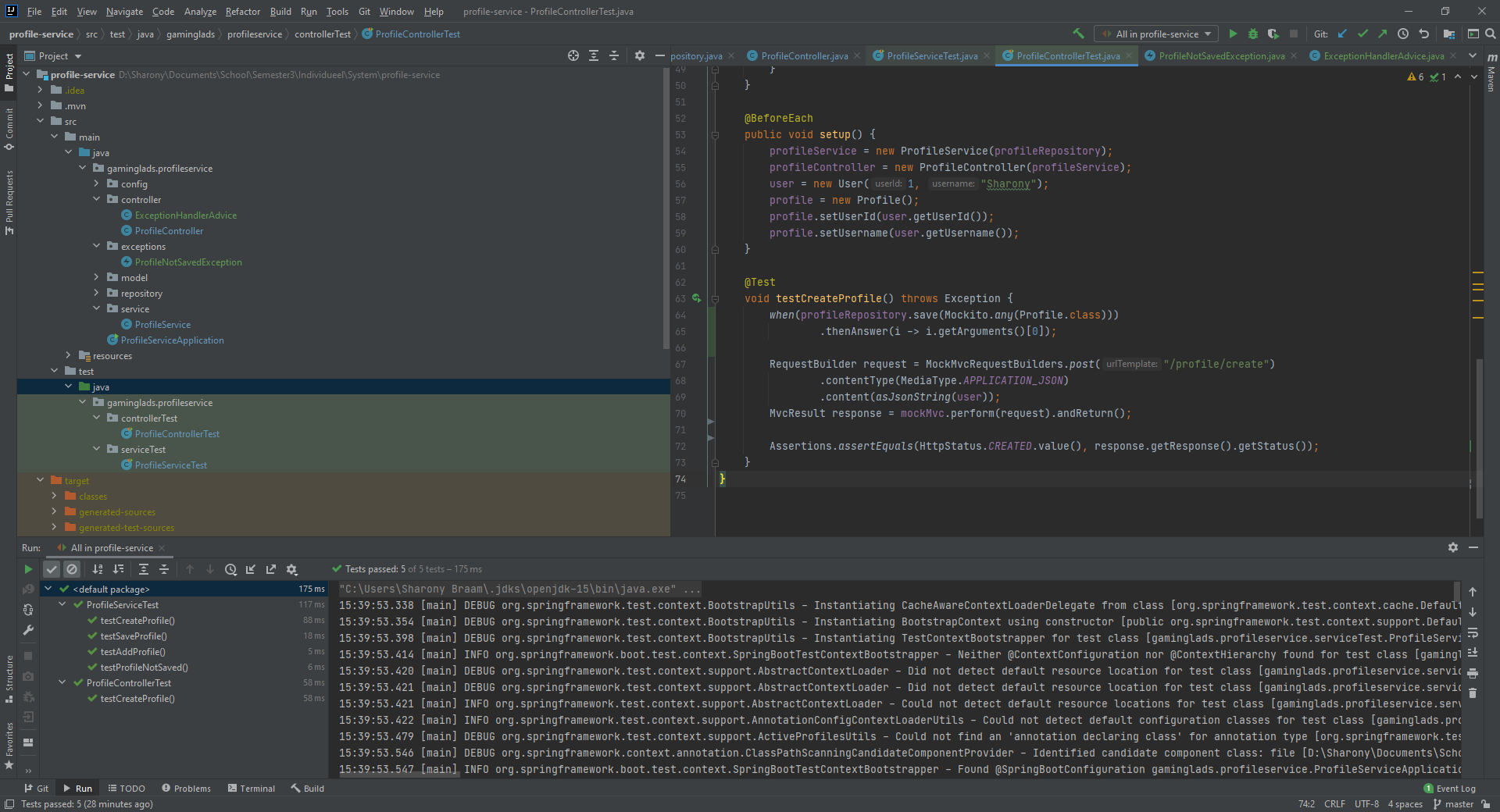
### User-Service

User-service heeft een aantal unit testen. Deze zijn van toepassing op de us: “Als gebruiker wil ik een account aan kunnen maken” en “Als gebruiker wil ik in kunnen loggen”. Hier wordt de happy flow en de unhappy flow getest. Elke method die in gebruik is door het project, wordt getest.



Profile-Service

Profile-Service heeft een aantal unit testen. Deze zijn van toepassing op de us: “Als gebruiker wil ik een profiel hebben. De happy en unhappy flow worden getest. Die hebben dus als resultaat dat een gebruiker een profiel heeft, als hij/zij een account heeft aangemaakt.



## Component tests

Api Gateway

Api Gateway heeft geen testen omdat daar niks te testen is.

Discovery-Server

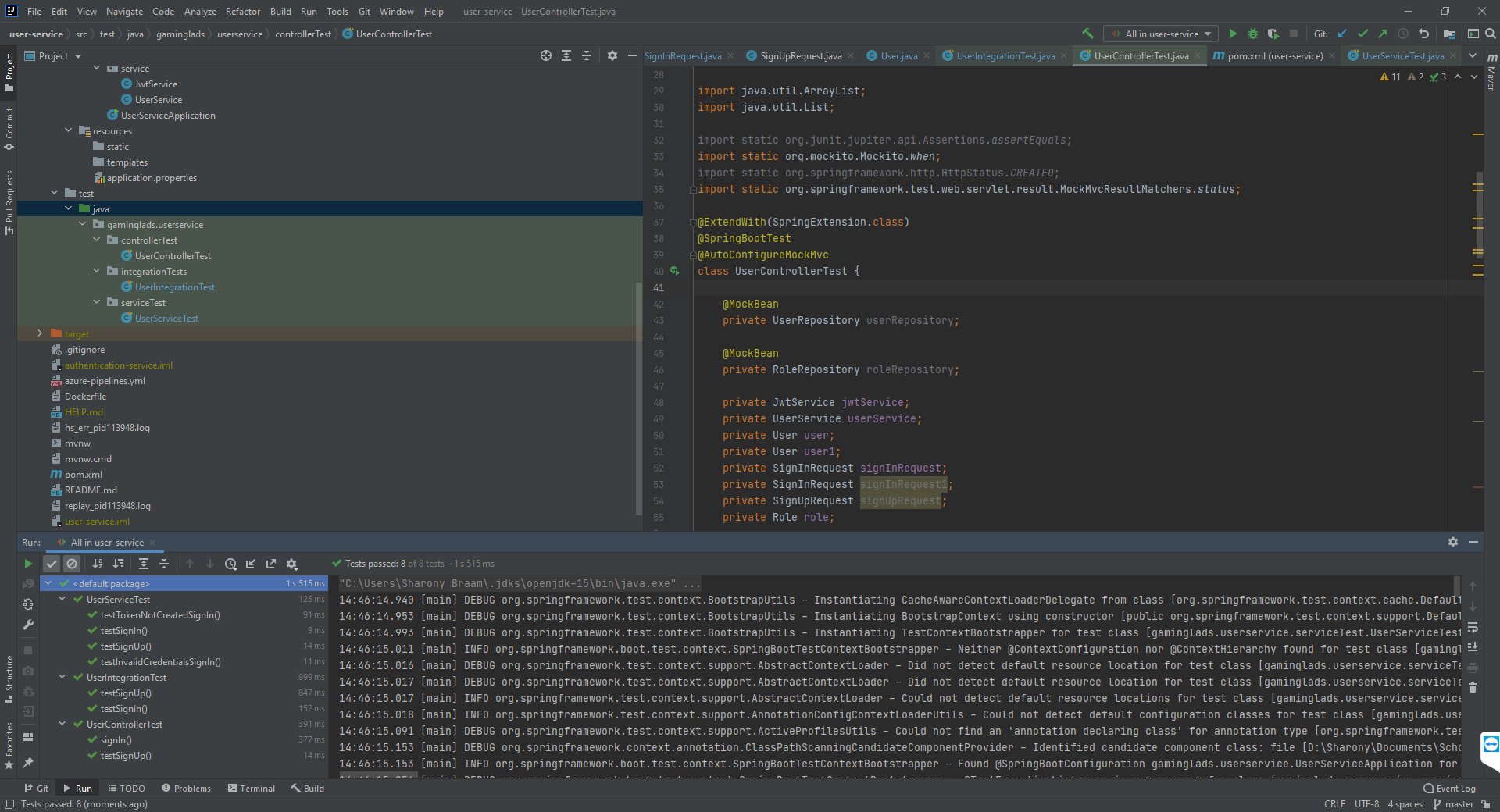
Discovery-Server heeft geen testen omdat daar niks te testen is.

Chat-Service

Chat-Service heeft geen testen omdat Chat-Service nog niet van toepassing is gekomen.

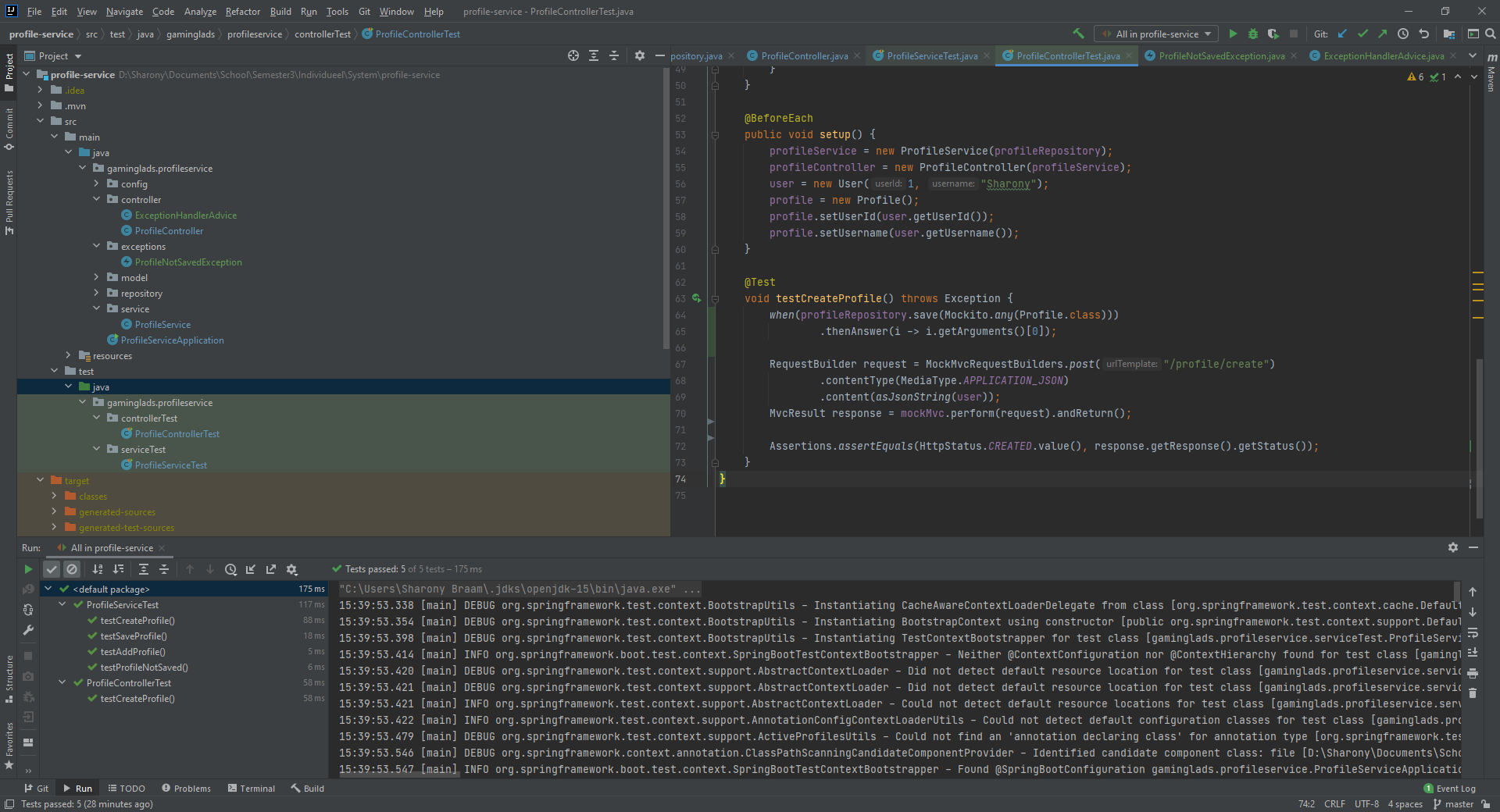
User-Service

User-Service heeft twee component test. De eerste api-endpoint is signUp. Dit heeft te maken met de US: “Als gebruiker wil ik een account aan kunnen maken.” Hier test hij de controller en heeft als resultaat dat er een account is aangemaakt. De tweede is het api-endpoint signIn. Dit heeft te maken met de US: “Als gebruiker wil ik in kunnen loggen”. Dit heeft als resultaat dat er ingelogd is en een jwt is aangemaakt. De component tests testen alleen de happy flow omdat de unhappy flow al getest is in unit tests.



### Profile-Service

Profile-Service heeft één component test. Het test het api-endpoint create. Dit heeft te maken met de US: “Als gebruiker wil ik een profiel hebben.” Hier test hij de controller. Dit is ook alleen de happy flow omdat de unhappy flow al getest is in unit tests en heeft dus als resultaat dat er een profiel is aangemaakt.



## Integration Tests

Front-end User

Front-end User heeft geen integration tests. Alle testen zijn handmatig uitgevoerd i.v.m. de tijd om front-end testing te leren, ook is er nog niet veel functionaliteit en daarom is het dus onnodig.

De user stories die op dit moment getest zijn, zijn “Als gebruiker wil ik een account aan kunnen maken”, “Als gebruiker wil ik in kunnen loggen” en “Als gebruiker wil ik een profiel hebben”. Dit zijn de grootste functionaliteiten op dit moment in GamingLads.

US: “Als gebruiker wil ik een account aan kunnen maken” en “Als gebruiker wil ik een profiel hebben”. Deze worden samen gepakt omdat als iemand een account aanmaakt, zij standaard een profiel krijgen. Dit is dan ook het resultaat.

US: “Als gebruiker wil ik in kunnen loggen” heeft als resultaat dat de gebruiker in kan loggen en een jwt meegestuurd krijgt.

Front-end Admin

Front-end Admin heeft geen testen omdat de front-end nog niet van toepassing is gekomen.

Api Gateway

Api Gateway heeft geen testen omdat daar niks te testen is.

Discovery-Server

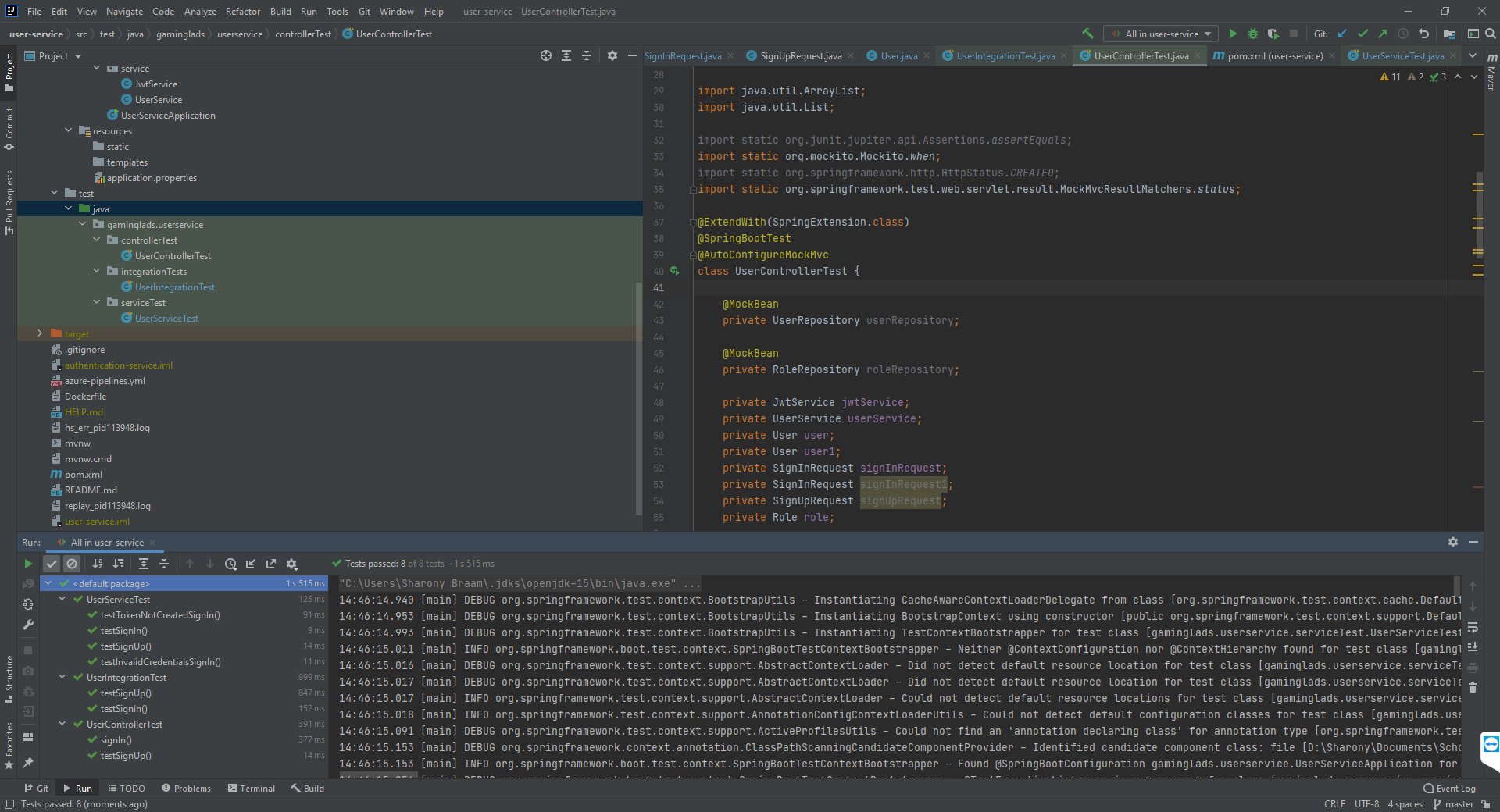
Discovery-Server heeft geen testen omdat daar niks te testen is.

Chat-Service

Chat-Service heeft geen testen omdat Chat-Service nog niet van toepassing is gekomen.

### User-Service

User-service heeft twee integration tests. Deze zijn van toepassing op de us: “Als gebruiker wil ik een account aan kunnen maken” en“Als gebruiker wil ik in kunnen loggen”. Hier wordt alleen de happy flow getest met dezelfde reden als bij de component tests. Als er een account aangemaakt wordt, wordt er ook een profiel voor diegene aangemaakt. Dit houdt in dat in de integratie test van signUp. De profile-service’s create endpoint ook getest wordt. Om die reden is er geen integratie test bij de profile-service.



Profile-Service

Profile-Service heeft geen integratie testen, zie user-service voor de uitleg.

# Conclusie

Voor de functionaliteiten die er zijn in het project zijn deze testen genoeg. De functionaliteiten worden op meerdere manieren getest. Het enige wat matig is, is de front-end testing omdat daar niet automatisch getest is. Als daar tijd voor was geweest, was de kwaliteit iets beter.