Verslag PROJ2   
Agile & OO programming

Groep: Klas 3, F

Deelnemers: Joren Prkic (22111131), Laurens Frensen (22106189)

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Datum | Aangepast door | Beschrijving |
| 0.1 | 22-05-2023 | Joren en Laurens | De eerste versie was klaar aan het einde van de eerste sprint, hier hadden we een login window met een login systeem gemaakt. Verder was er een basis structuur gemaakt waar we later op zullen bouwen. |
| 0.2 | 05-06-2023 | Joren en Laurens | De tweede versie had alle functionaliteit er bijna helemaal in verwerkt. Je kon chatten, van chats veranderen, uitloggen en inloggen. |
| 0.3 | 11-06-2023 | Joren en Laurens | In deze versie is ons eindproduct voor dit project afgerond, alle user stories zijn er in verwerkt en het gehele project (alles wat binnen scope lag) is compleet. |

Tabel 1: Versiebeheer

Inhoudsopgave

[1 Ontwerp 4](#_Toc135732385)

[1.1 UML Class diagram 4](#_Toc135732386)

[1.2 Uitleg classes 4](#_Toc135732387)

[1.3 Uitleg SOLID principles en code smells 4](#_Toc135732388)

[1.3.1 Screenshot BetterCodeHub 4](#_Toc135732389)

[1.4 Uitleg design patterns 4](#_Toc135732390)

[2 Testontwerp 5](#_Toc135732391)

[2.1 Beslissingstabellen voor modified condition/decision coverage 5](#_Toc135732392)

[2.2 Decision, condition, multiple condition coverage testen 5](#_Toc135732393)

[3 SCRUM 6](#_Toc135732394)

[3.1 Definition of Ready 6](#_Toc135732395)

[3.2 Definition of Done 6](#_Toc135732396)

[3.3 Retrospective 6](#_Toc135732397)

[Appendix A: Java code 7](#_Toc135732398)

# Ontwerp

## UML Class diagram

UML class diagram. Let op dat het leesbaar is.

## Uitleg classes

Niet nodig vlgns OPT docent.

## Uitleg SOLID principles en code smells

Ik (joren) ben een voor een door de code gegaan en heb de checklist van code smells erbij gehad:  
Long Method, Large Class, Temporary Field, Long Parameter List, Duplicate Code, Switch-statements, Primitive Obsession, Divergent Change, Shotgun Surgery.

In de laatste versie zijn ze er allemaal uitgehaald (kon er maar 2 vinden). Daarna heb ik ChatGPT alles nog een keer laten nakijken, die kwam tot de conclusie dat er nergens een code smell te vinden was.

Verder kun je aan de screenshot van Sigrid zien dat de SOLID princles ook goed zijn toegepast, aangezien die daar gedeeltelijk zn cijfer op baseerd, daarnaast, indien mogelijk zou ik de code kunnen aanwijzen waar de SOLID’s zijn toegepast.

### Screenshot BetterCodeHub (Sigrid)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

## Uitleg design patterns

We maken gebruik van de **Adapter pattern** en de **template method pattern**. We hebben de template method pattern toegepast in de adapter pattern, dus ze staan beide op dezelfde plek.

De adapter pattern is toegepast doordat in plaats van AI Manager.ask te callen, call je de adapter, en die called AIManager.ask intern nadat de data verwerkt is op de manier waarop het nodig is.

Klik op [deze](https://github.com/joren-dev/AI42/tree/main/src/main/java/nl/ai42/managers/AIAdapter) link om naar de map te gaan waar de adapter gedefinieerd is. Je zult zien dat er een interface (AIAdapter) is en een uitwerking (AIAdapterManager), de uitleg daarover kun je [hier](https://github.com/joren-dev/AI42/blob/0c58051a7fd7c6b429f1c9d8bb617874fcf7b367/src/main/java/nl/ai42/managers/AIAdapter/AIAdapter.java#L4) terugvinden. Hier leg ik namelijk uit hoe de template method pattern perfect werkt met de adapter pattern die we hebben.

# Testontwerp

We hebben een lijst van nuttige test toegevoegd volgens de volgende standaard java junit naming convention: FunctionName\_Status\_ExceptedReturnValue

A screenshot of a computer program

Description automatically generated with medium confidence

## Beslissingstabellen voor modified condition/decision coverage

Waren niet nodig, we moesten enkele “nuttige” tests laten zien.

## Decision, condition, multiple condition coverage testen

Waren niet nodig, we moesten enkele “nuttige” tests laten zien.

# SCRUM

We voldoen aan Onderhoudbaarheid, betrouwbaarheid en efficiëntie uit ISO 25010:

Onderhoudbaarheid doordat we gebruik hebben gemaakt van de SOLID principles (kun je in sigrid terugzien), deze SOLID’s zorgen voor onderhoudbaarheid. Daarnaast voldoet de efficientie ook aangezien er tijdens het gebruik geen moment van hapering of slome prestaties te zien en/of voelen zijn. Betrouwbaarheid komt terug uit de SOLID’s en het feit dat het programma op elke systeem werkt, daarnaast werkt het ook offline.

## Definition of Ready

Beschrijving van Definition of Ready…

## Definition of Done

Beschrijving van Definition of Done…

## Retrospective

In scrumwise zijn de retrospectives te vinden.

**Verbeteringen toegepast na sprint 1**

**Verbeteringen toegepast na sprint 2**

**Verbeteringen toegepast na sprint 3**

Geef hier aan per sprint welke verbeterpunten jullie doorgevoerd hebben naar aanleiding van de uitkomsten van de retrospective.

# Appendix A: Java code

Documentation link: <https://joren-dev.github.io/AI42/>

Github link: <https://github.com/joren-dev/AI42>