

# **Verslag PROJ1**

## **Agile & OO programming**

Groep: H-1-SE.8-D

Deelnemers: [Naam, Studentnummer]

Anthony Delgado - 22102167

David van der Harst - 22140875

Daan Polet - 22106162

Luuk van den Maagdenberg - 22149449

Luke van der Knaap - 20135599

Seb Geers - 22098941

<b>Versie</b>	<b>Datum</b>	<b>Aangepast door</b>	<b>Beschrijving</b>
0.1	13/04/2023	Anthony Delgado	Initiële versie
0.2	13/04/2023	Anthony Delgado	Moet nog nagekeken worden door groep

*Tabel 1: Versiebeheer*

## Inhoudsopgave

1	Ontwerp.....	4
1.1	UML Class diagram.....	4
1.2	Uitleg.....	4
2	Realisatie.....	5
3	Beheersbaar laten verlopen van ontwikkelprocessen.....	6

# **1    Ontwerp**

## **1.1   UML Class diagram**

## 1.2 Uitleg

Class naam	Functionaliteit
Commands	In commands en de geneste classes hierin aanwezig zijn de static Strings opgeslagen waarmee user input wordt vergeleken om een functie uit te voeren in Runtime.
TextColors	Hierin zijn static Strings opgeslagen die aangeroepen kunnen worden om Strings te maken dat in kleur wordt geprint in de console.
Optie	Van deze class worden objecten aangemaakt met informatie van een optie, zoals naam en prijs.
OptieLijst	Deze class is verantwoordelijk voor het uitlezen van en schrijven naar het .csv bestand 'opties.csv', en aan de hand daarvan een ArrayList te vullen met Opties.
Klant	Van deze class worden objecten aangemaakt met informatie over een klant, zoals de naam van de klant en wat voor soort klant het is.
KlantSoort	Van deze class worden objecten aangemaakt met informatie over een soort klant, zoals de naam van de soort en het bijbehorende kortingspercentage.
OfferteNieuw	Nieuwe class die de offerte regelt. Aan een object van deze klas kan een klant en opties toegevoegd worden, en doormiddel van de Runtime class kan een methode aangeroepen worden dat de voorafgenoemde informatie uitprint met een prijsberekening. Met een andere input vanuit Runtime wordt een 2e, vergelijkbare methode aangeroepen dat een folder en een .txt bestand maakt van de offerte.
Runtime	De hoofdklasse. Hierin wordt user input geregistreerd en vergeleken met de static Strings in Commands, en aan de hand daarvan worden methodes aangeroepen.
Package CSV_Files	In deze package worden de csv bestanden bewaard.

Associatiennaam	Toegevoegde waarde
"Heeft"	De [naam]-Lijst klassen hebben een lijst gevuld met objecten. Runtime heeft 1 centrale klantenlijst en optielijst.
Vergelijkt input met Command Strings	Als een gebruiker iets invoert in de console, wordt het vergeleken met alle bekende Command Strings om zo de juiste methode aan te roepen.
Gebruikt Color Strings om gekleurde tekst te printen	Classes kunnen de statische Strings in TextColor gebruiken om een string te maken dat in kleur geprint zal worden.
Implementeert	Deze class implementeert een interface.

Als er ergens in het UML [-attribute] of [+operation()] staat : Dit zijn placeholders, zodat elk afzonderlijk vak voor variabelen en methodes in een class diagram zichtbaar is. Zonder deze placeholders zouden de diagrammen en vak missen.

## 2 Realisatie

Link naar Github Repo (zorg dat je de OPT-docent daarvoor eerder al hebt uitgenodigd).

<https://github.com/DKnightAnon/OPT-project-1>

In onderstaande tabel geef je een kort overzicht van de functionaliteit en de tests die door de individuele leden van het team zijn gerealiseerd (uiteraard kunnen we dat zo 1:1 terugvinden in GitHub, maar dit maakt het zoeken naar ieders bijdrage in GitHub mogelijk):

Naam teamlid	Bijdrage aan functionaliteit en tests
Daan Polet	Eerste versie van de software(User input, standaard opties, offerte)
David van der Harst	TestClass.java(voegAlleOptiesToeAanLijst,testmaakOptie, testVoegOptieToe), concept Offerte
Anthony Delgado	SystemInterface voor user input, data lezen vanuit CSV bestanden voor KlantLijst en OptieLijst
Luuk van den Maagdenberg	Comments in code
Luke van der Knaap	

### 3 Beheersbaar laten verlopen van ontwikkelprocessen

Bij dit leerdoel worden jullie beoordeeld op drie criteria en twee daarvan vinden we terug in dit verslag. Geef in dit verslag het volgende aan:

- 3.1 Een verslagje van tenminste twee retrospectives met hierin een reflectie op de afgelopen sprint en de verbeterpunten die jullie de volgende sprint gaan doorvoeren.
- 3.2 Geef per sprint aan wat jullie planning was en in hoeverre deze planning gehaald is. Als een punt niet behaald is, geef aan waarom niet.

#### **Sprint 1 :**

1. UML user story 1,3,4
2. Code op basis van UML 1,3,4

Tijdens deze sprint zijn de code en UMLs van User Story 1 en 3 gemaakt. De taken voor User Story 4 hebben we niet afgemaakt, maar verplaatst naar de volgende sprint omdat het nodig was om andere User Stories uitgewerkt te hebben.

#### **Sprint 2 :**

1. UML User story 2, 4, 6
2. Code op basis van User Story 2,4,6
3. UMLs samenvoegen User Story 1 t/m 4
4. Tests schrijven bij code

Tijdens deze sprint hebben we al onze taken afgekregen. Richting het einde van de sprint kwamen wij erachter dat een van de teamleden vooruit had gewerkt en user story 5 en 9 gedeeltelijk had uitgewerkt.

#### **Sprint 3 :**

1. UML uitbreiden
2. User inputs in aparte class afhandelen
3. Opties aanmaken en kiezen versimpelen
4. Inleverdocumenten

Tijdens deze sprint hebben wij onze taken op tijd afgekregen.