|  |
| --- |
|  |
| Software Design Description |
| Versienummer: 1.1 |

|  |
| --- |
| Namen: Sem Hones & Emiel Dreef  Studentnummers: 659955 & 603443  Naam docent: Ben Nieuwhof  Klas: ITA-OOSE-A  Course/semestergegevens: OOAD  Datum: 3-13-2022 |

Inhoudsopgave

[1 Inleiding 2](#_Toc98076141)

[1.1 Algemene omschrijving 2](#_Toc98076142)

[1.2 Doel 2](#_Toc98076143)

[2 Architectuur 3](#_Toc98076144)

[3 Gedetailleerd design omschrijving 4](#_Toc98076145)

[3.1 Deployment diagram 4](#_Toc98076146)

[3.1.1 Design keuzes 4](#_Toc98076147)

[3.2 Design Provo 5](#_Toc98076148)

[3.2.1 Design class diagram 5](#_Toc98076149)

[3.2.2 Sequence diagrammen 5](#_Toc98076150)

[3.3 Design keuzes 7](#_Toc98076151)

[3.3.1 Toetstimer 7](#_Toc98076152)

[3.3.2 Puntentelling 7](#_Toc98076153)

# **Inleiding**

## Algemene omschrijving

In dit document wordt een beschrijving gegeven van de interne werking van het programma Provo.

In dit hoofdstuk wordt een introductie gedaan voor het document. In het volgende hoofdstuk wordt de architectuur van het programma beschreven. In het laatste hoofdstuk wordt de structuur van het programma in diagrammen getoond. Ook worden er hier verschillende subsystemen beschreven.

## Doel

Het doel van dit document is om een duidelijk beeld te geven van de interne structuur en werking van dit systeem. Hierbij moet een willekeurig persoon dit document kunnen lezen en het gelijk kunnen begrijpen en bouwen.

# Architectuur

Het project wordt opgedeeld in 3 lagen: presentatie, domein en technische service.

De presentatie zal het uiterlijk van programma hebben. In het domein laag komt de logica en in de technische service laag zal de data komen en opgeslagen worden.

Het idee van de lagen is dat de presentatie laag alleen gebruik maakt van het domein laag en het domein laag alleen gebruik maakt van de data. De reden dat de lagen gescheiden gehouden worden is zodat het makkelijk vervangbaar is.

# Gedetailleerd design omschrijving

In dit hoofdstuk worden alle softwarecomponenten gedocumenteerd. In 3.1 wordt het deployment diagram getoond. Deze wordt hierna toegelicht.

Verder worden alle subsystemen en de database beschreven.

## Deployment diagram

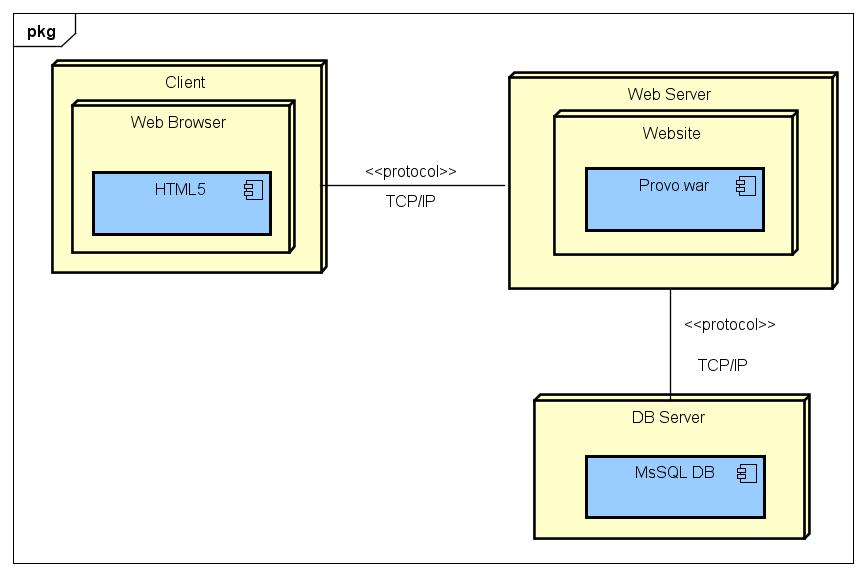


Diagram 2 – Deployment diagram

### Design keuzes

Voor het deployen van de webapplicatie is gekozen voor een WAR file. Hiervoor is gekozen omdat er sprake is van een webapplicatie.

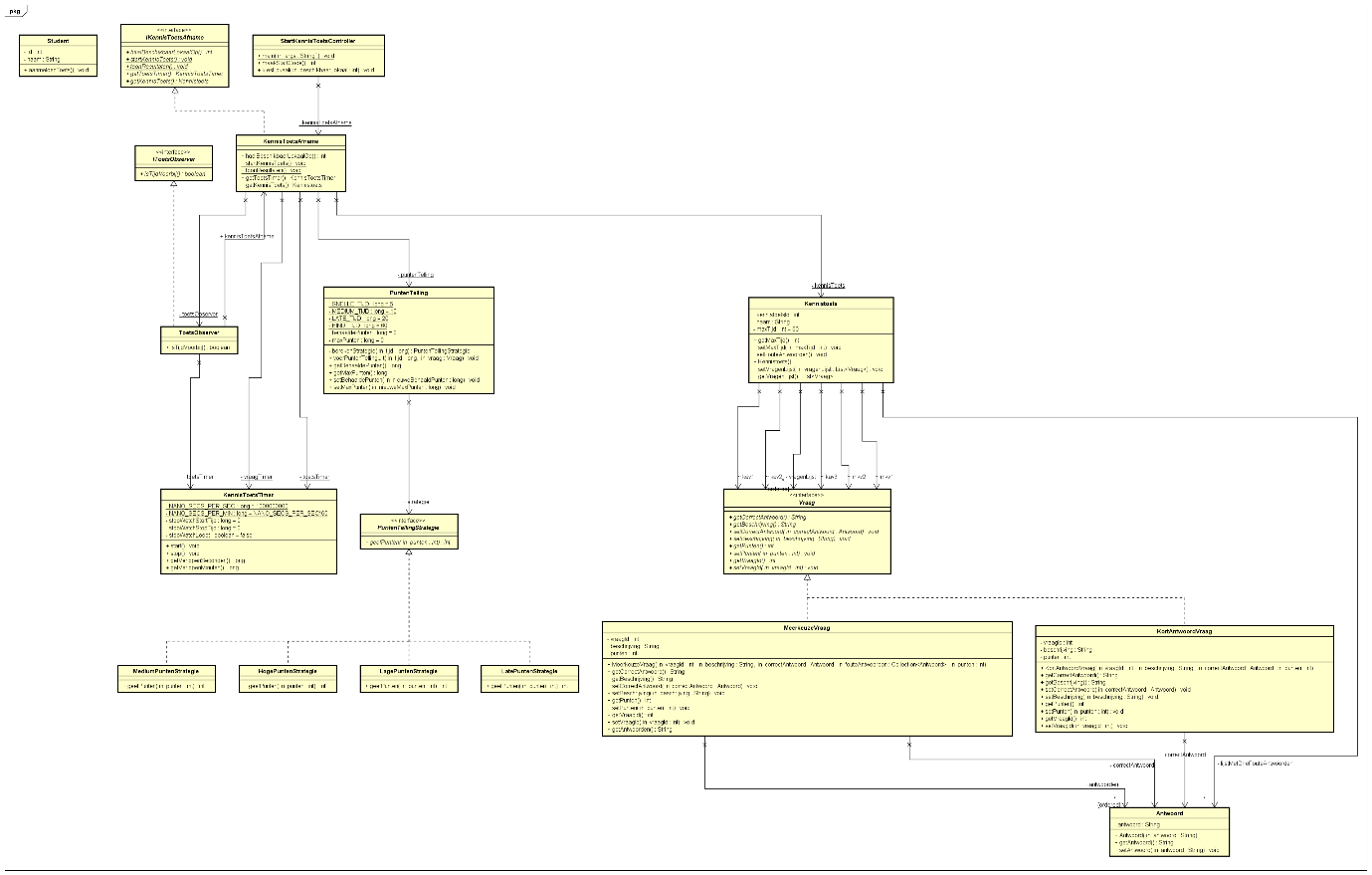
Er is gekozen om een MsSQL database te gebruiken de reden hiervoor is omdat, er meer ervaring is met de MsSQL database.

Als protocol wordt gebruik gemaakt van TCP/IP. Dit is het standaard protocol voor communicatie tussen de cliënt en de servers.

## Design Provo

Alle diagrammen zijn als bijlage geleverd om ze beter te kunnen bekijken

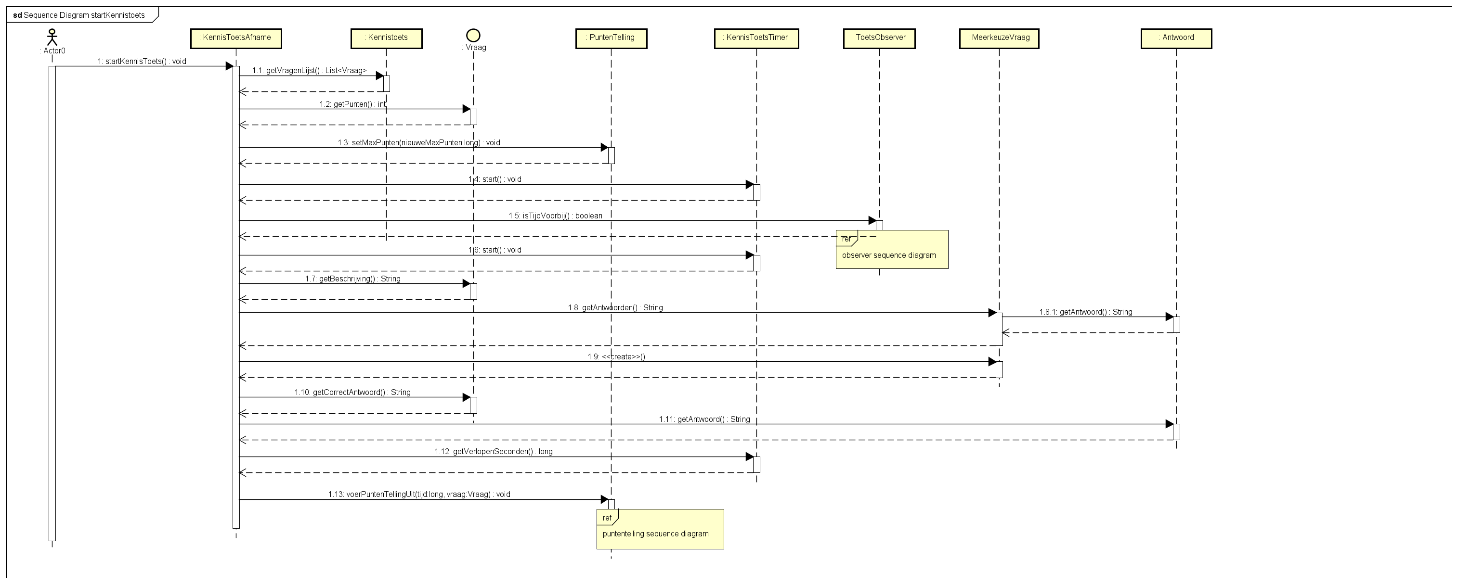
### Design class diagram



*Design class diagram 1 – volledige applicatie*

### Sequence diagrammen

#### Start kennistoets



*Sequence diagram 1 – start kennistoets*

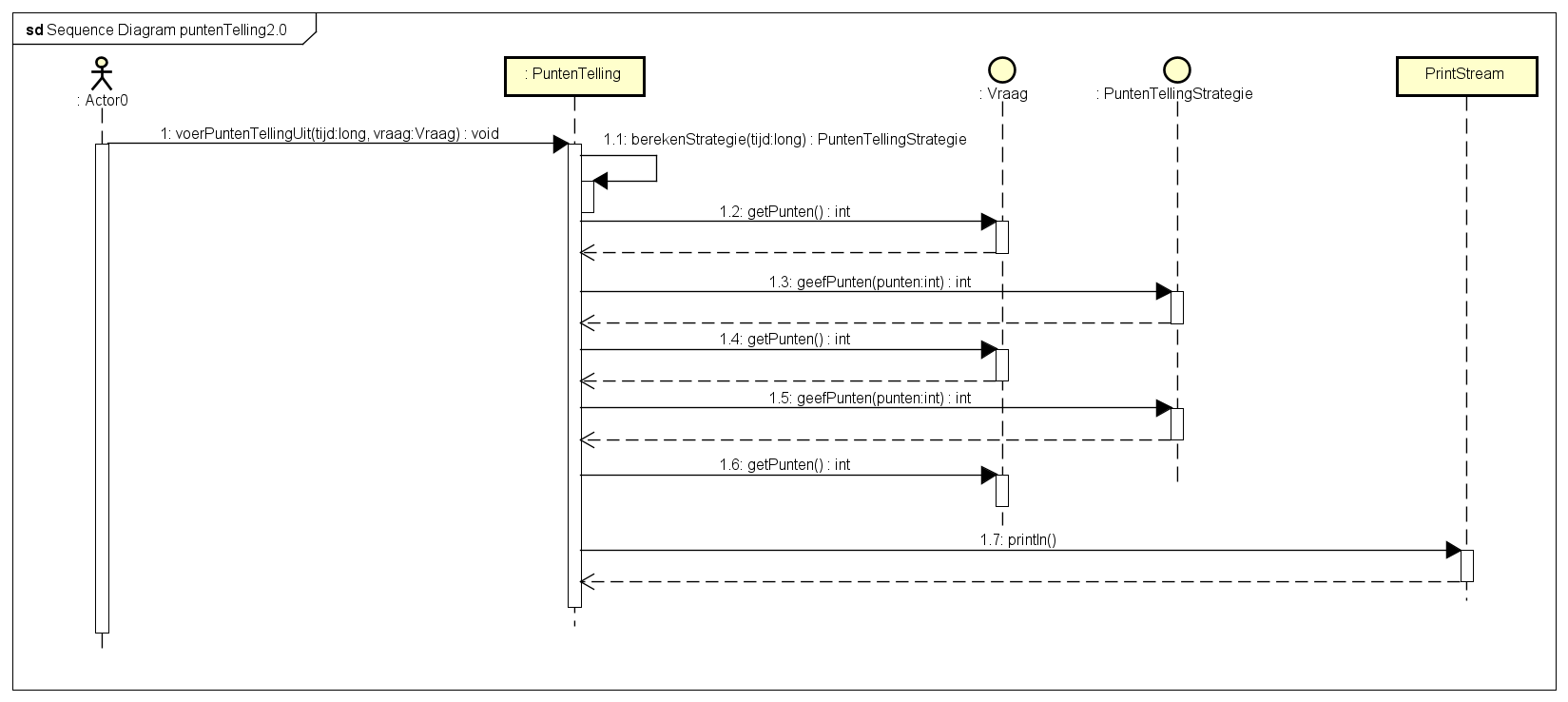
#### Observer toetstimer

Afbeelding met tafel

Automatisch gegenereerde beschrijving

*Sequence diagram 2 – Observer toetstimer*

#### Puntentelling



*Sequence diagram 3 – puntentelling*

## Design keuzes

### 3.3.1 Toetstimer

Voor de toetstimer is een observer gebruikt of ook wel een listener pattern. Het idee hiervan is als de gebruiker over de maximale gegeven tijd heen gaat voor een toets zal de observer de hele toets stopzetten. Ook wel de state aanpassen.

### 3.3.2 Puntentelling

Voor de puntentelling is er een strategy pattern gebruikt. De strategy pattern is gebruikt omdat de puntentelling op meerdere manieren uitgevoerd moet kunnen worden. bijvoorbeeld elke vraag krijgt een op een manier zijn punten dit is gebaseerd op hoelang een persoon doet over zijn vraag beantwoorden. Tussen mogelijke tijdspannen komt er dan ook een formule voor het aantal gegeven punten.