|  |  |  |
| --- | --- | --- |
|  |  | QPower  Course: Object Oriented Analysis and Design |

Software Design Description



**Student: Studentnr.:**

Mick Tuit 574374

Wessel Barten 522518

**Docent:**

Herman Telman

**Datum:**

06-04-2017

**Versie:**

1.0

# Inhoud

[Inhoud 2](#_Toc479177433)

[1 Introduction 3](#_Toc479177434)

[1.1 Overall description 3](#_Toc479177435)

[1.1 Purpose of this document 3](#_Toc479177436)

[2 Detailed Design Description 4](#_Toc479177437)

[2.1 Design QPower Single Player 4](#_Toc479177438)

[2.1.1 Design Class Diagram 4](#_Toc479177439)

[2.1.2 Activity Diagram 5](#_Toc479177440)

[2.1.3 Design decisions made for the sub-system 6](#_Toc479177441)

[2.2 Overall Design QPower 7](#_Toc479177442)

[2.2.1 Design Class Diagram 7](#_Toc479177443)

[2.2.2 Activity Diagram 8](#_Toc479177444)

[2.2.3 Design decisions made for the sub-system 9](#_Toc479177445)

# Introduction

## Overall description

Dit document bevat de software design description voor QPower, een applicatie waarmee eenvoudig een pubquiz gespeeld kan worden door meerdere teams of een single player quiz door één speler. De applicatie is ontworpen voor activiteitenbureau A-Team, een bedrijf dat pubquizzen organiseert op diverse locaties. QPower gaat ondersteuning bieden in de processen rondom het aanmaken, spelen en beheren van de verschillende quizzen.

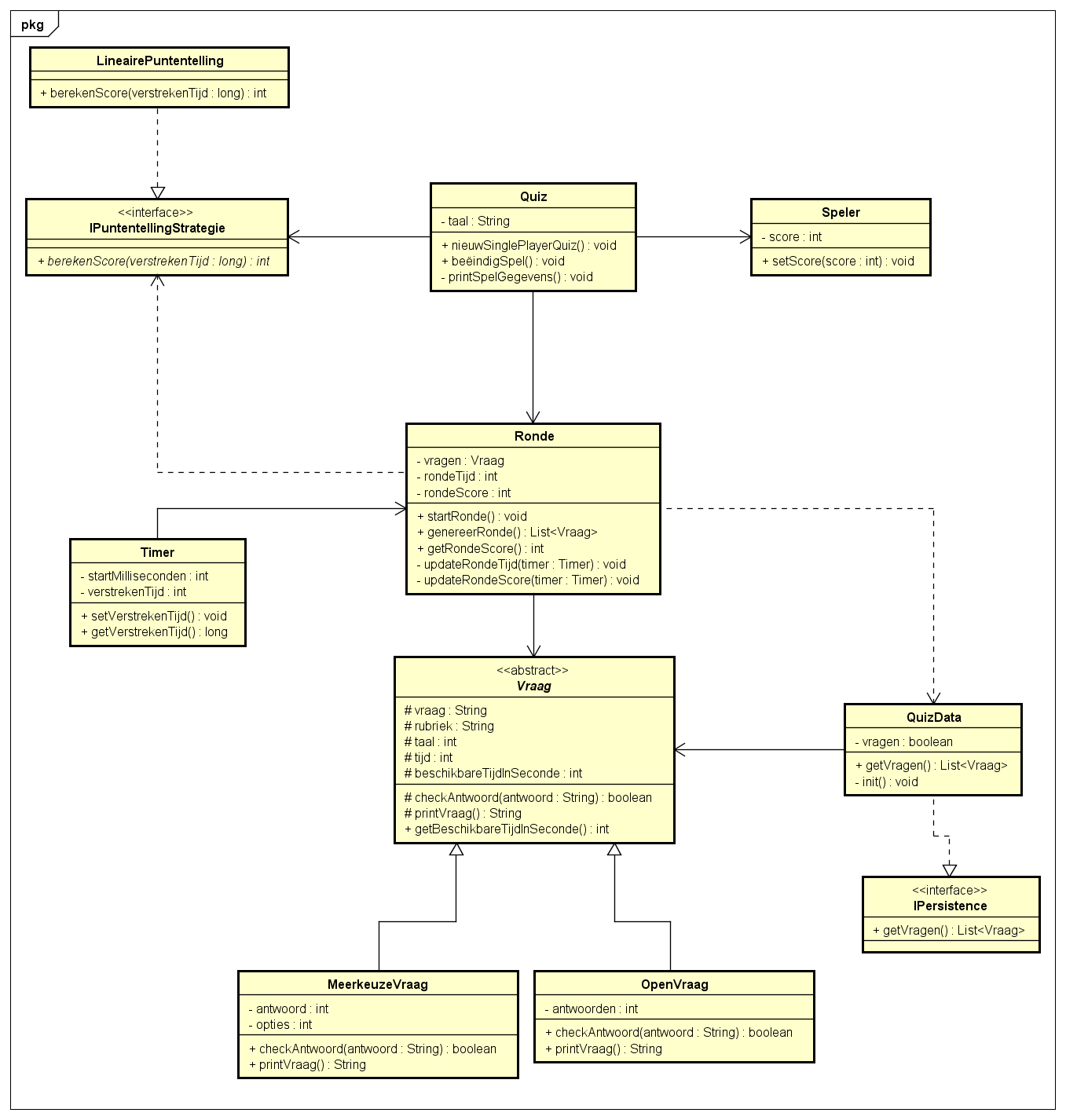
## Purpose of this document

Doel van dit document is inzicht geven in het ontwerp van de QPower applicatie door middel van UML design class diagrammen voor de Single Player puiz en de gehele Pubquiz.

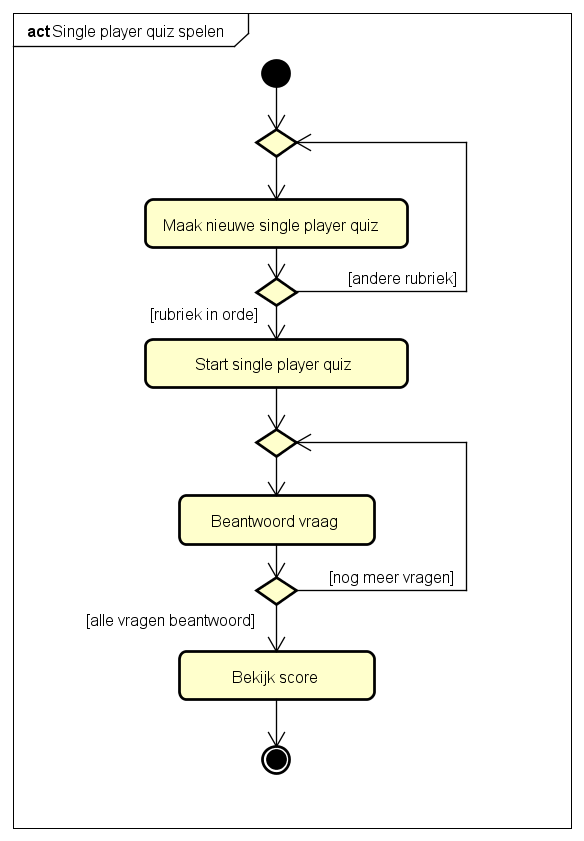
# Detailed Design Description

## Design QPower Single Player

### Design Class Diagram



### Activity Diagram



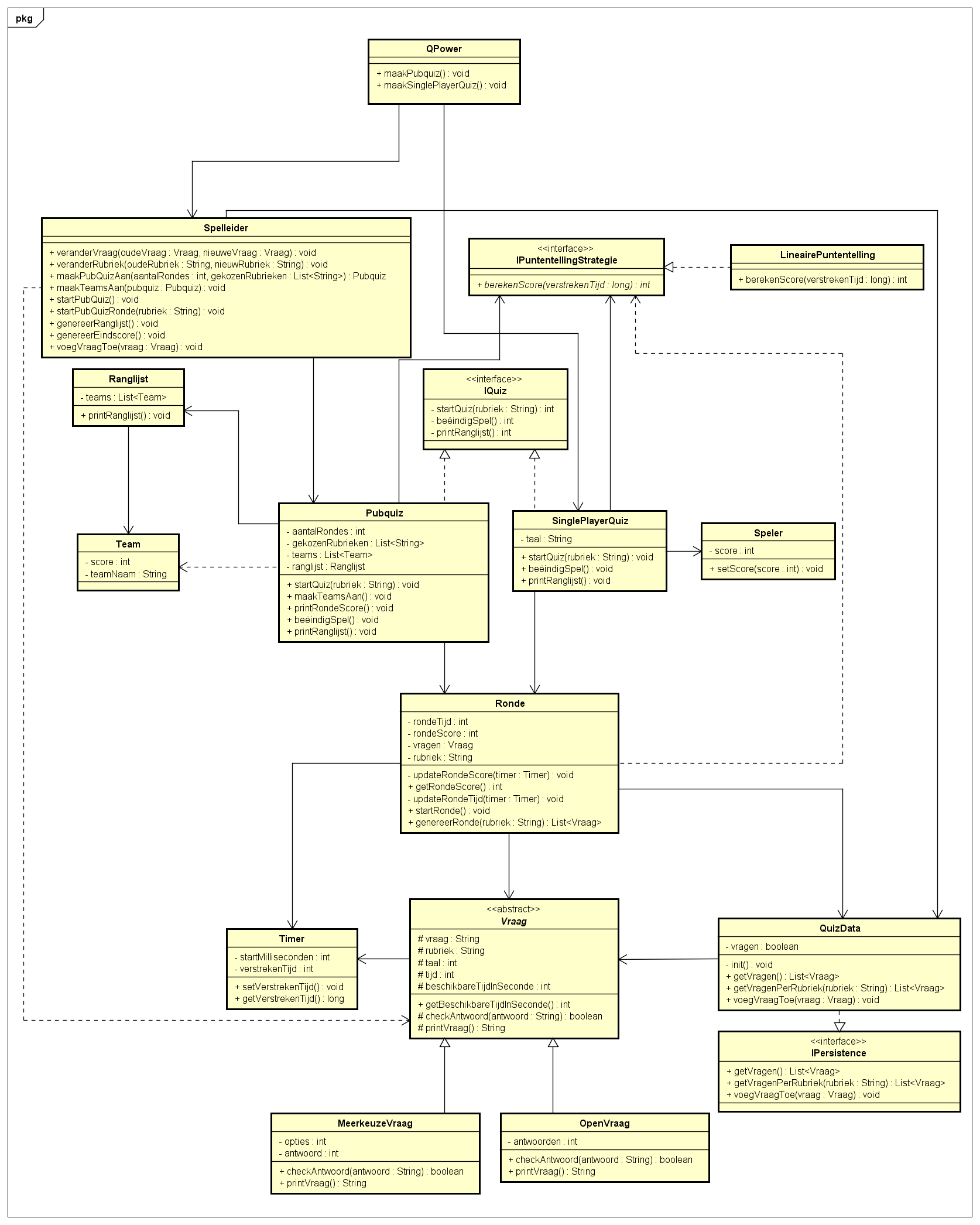
### Design decisions made for the sub-system

In het subsysteem QPower Singleplayer is geen gebruik gemaakt van frameworks, libraries of andere technologieën. Alleen de functionaliteiten die beschikbaar zijn binnen Java zijn gebruikt. Er is wel een design pattern (behaviour) toegepast, namelijk het Strategy Pattern. Dit is toegepast op de puntentelling, aangezien er verschillende strategieën geïmplementeerd kunnen worden voor het bepalen van de score van de speler(s).

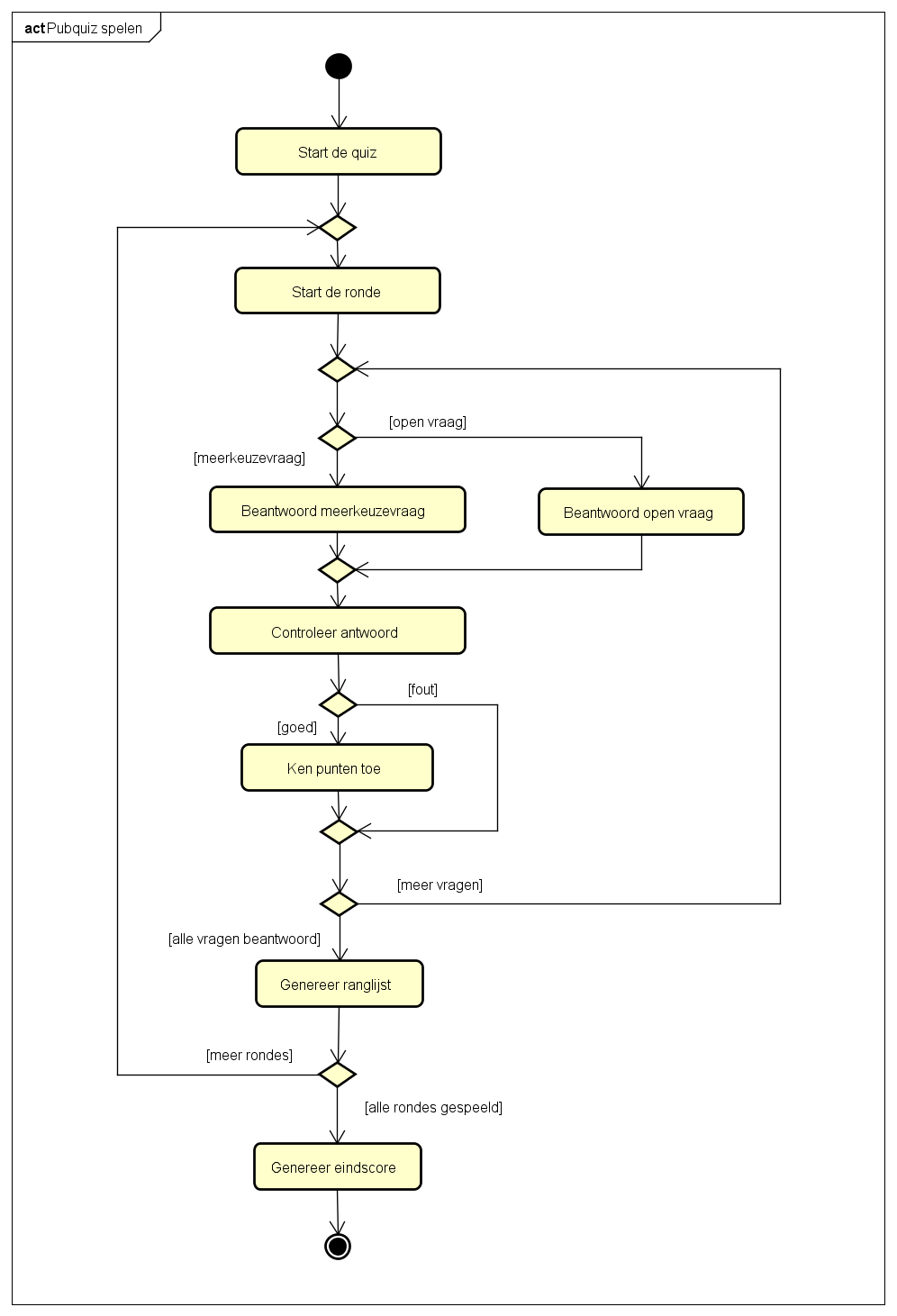
Een aantal geadapteerde principes van GRASP zijn ‘high cohesion’, ‘low coupling’, ‘information expert’ en ‘polymorphism’. ‘High cohesion’ aangezien klasses één verantwoordelijk hebben en niet meer dan dat. De verschillende klasses werken onderling goed samen. Echter is er een ‘low coupling’, omdat zo te zien is in het design class diagram er maar twee dependencies zijn. ‘Information expert’ omdat er veel gedelegeert wordt in onze applicatie. Zo delegeert Quiz naar Ronde om een ronde te spelen en Ronde naar Vraag om het antwoord te checken. Als laatste passen we ‘polymorphism’ toe aangezien we een abstracte klasse Vraag hebben.

## Overall Design QPower

### Design Class Diagram



### Activity Diagram



### Design decisions made for the sub-system

Voor het ontwerp van de gehele QPower applicatie is het ontwerp voor de single player quiz als basis gebruikt en uitgebreid met de functionaliteiten voor het spelen van de pubquiz met teams. Ook hier is het Strategy Pattern toegepast voor de puntentelling en is er gebruik gemaakt van de GRASP principes *high cohesion*, *low coupling*, *information expert* en *polymorphism*. Er wordt geen gebruik gemaakt van API’s, libraries of frameworks.

Een van de verschillen tussen de twee diagrammen is dat de class Quiz is opgesplitst in twee classes *Pubquiz* en *SinglePlayerQuiz* die beide de interface *IQuiz* implementeren. Hierdoor wordt het mogelijk om de applicatie uit te breiden met meerdere soorten quizzen. Dit sluit aan bij het GRASP principe *polymorphism*.