

Veluwenkamp, Bart B.

chess bracket System

Ontwerpdocument

6 februari 2023

Inhoudsopgave

[Klassen diagram 3](#_Toc135397165)

[Databasemodel 4](#_Toc135397166)

[Architectuur diagram 5](#_Toc135397167)

[Presentatie laag 6](#_Toc135397168)

[Logica laag 6](#_Toc135397169)

[Dataopslag laag (Dal) 6](#_Toc135397170)

# Klassen diagram

In dit hoofdstuk is het klassen diagram te vinden. Ook is er uitleg te vinden over de keuzes die gemaakt zijn.

Om te beginnen zijn er de klassen Players en Matches te vinden. De responsabiliteit van deze klassen had ik eerst bij mijn Toernooi en Bracket klassen gelegd. Door dit uit elkaar te halen zorg ik ervoor dat iedere klassen maar een ding hoeft te doen “Single responsibility”. Ook heb ik ervoor gekozen om enums te gebruiken in mijn applicatie. Deze worden gebruikt om ervoor te zorgen dat er niet de hele tijd strings gecontroleerd te hoeven worden of ze wel juist zijn.

Ook implementeer ik interfaces in mijn applicatie. Dit is gedaan omdat er verwacht wordt dat ik drie lagen gebruik in mijn applicatie. Een drie laags applicatie bestaat uit een view laag, core laag en een data acces laag. Om te zorgen dat de afhankelijkheden van de lagen juist zijn moet er een interface gebruikt worden. Een andere reden waarom er voor interfaces is gekozen is zodat de dal/view laag makkelijk te veranderen zijn zonder dat ik dingen hoef aan te passen in de core laag.

Afbeelding met tekst, diagram, Parallel, Lettertype

Automatisch gegenereerde beschrijving

# Databasemodel

In dit hoofdstuk is het database model te vinden. Ook is er uitleg te vinden over de keuzes die gemaakt zijn.

In het database model staan eigenlijk alleen alle eigenschappen van de klassen samen met een identiteit. Het ID is er zodat het te vinden is in de database. Er zijn ook een extra tabellen te vinden namelijk PlayerList. Dit is een soort koppel tabel. Omdat er meerdere spelers in een spelerslijst zitten kan ik niet alle ID’s van de Players in de kolom van PlayerListID zetten. Als dit het enige was kon ik hetzelfde doen als bij de Match tabel. Hier zit een match maar bij een bracket en dus wordt het ID van het bracket opgeslagen in de match tabel. Voor spelers kan dit niet omdat spelers bij meerdere spelerslijsten kunnen horen. Vandaar dat een koppeltabel noodzakelijk is.

Afbeelding met tekst, diagram, schermopname, Plan

Automatisch gegenereerde beschrijving

# Architectuur diagram

In dit hoofdstuk is een diagram te vinden dat de architectuur van het project wat duidelijker maakt. Ook worden de keuzes die gemaakt zijn in het bedenken van de architectuur uitgelegd. Daarna wordt er kort uitgelegd wat de functie is van iedere laag in de applicatie.

Om te beginnen is er gekozen voor een drie laags architectuur. Dit houdt in dat de presentatie laag, logica laag en dal allemaal aparte projecten zijn. Hier is voor gekozen om het project modulair te houden. In het project is dit terug te vinden in het gebruik van testen. Hiervoor wordt er gebruik gemaakt van aparte dal laag zodat er geen nep data tussen de echte data komt te staan. Ook zorgt dit ervoor dat er een scheiding is van verantwoordelijkheden. Dit zorgt ervoor dat het gemakkelijk is om veranderingen aan een project te maken zonder dat er ook meteen aanpassingen gemaakt hoeven te worden aan de rest van de projecten. Afbeelding met tekst, schermopname, diagram, lijn

Automatisch gegenereerde beschrijving

## Presentatie laag

De presentatie laag is verantwoordelijk voor het laten zien van informatie aan de gebruiker. Ook is deze laag verantwoordelijk voor het verzamelen van de input van de gebruiker. De presentatie laag is afhankelijk van de logica laag voor het krijgen van alle informatie. Ook is de presentatie laag afhankelijk van de logica laag om de verzamelde informatie hiernaartoe te sturen zodat het op een juiste manier verwerkt kan worden.

## Logica laag

In de logica laag is alle logica van de applicatie te vinden. Het is verantwoordelijk voor het verwerken van de gegevens die van de presentatie laag komen. Deze gegevens worden dan door de logica laag op de juiste manier bewerkt door het implementeren van alle processen die in deze laag gedaan worden. De logica laag zorgt ervoor dat de data dat naar de presentatie laag op de juiste manier is voorbereid om te laten zien. Ook kan de logica laag communiceren met de dal laag om daar gegevens naar toe te sturen of te krijgen.

## Dataopslag laag (Dal)

De dataopslag laag is verantwoordelijk voor de communicatie met de data base. Het zorgt ervoor dat data opgevraagd vanuit de database kan worden. En dat er data naar de database geschreven kan worden. De dataopslag laag heeft een directe verbinding met een database. De Dal zorgt er ook voor dat de gegevens die in de database staan op een veilige manier opgeslagen en opgehaald worden.