

Ontwerp document

Jaron Poel



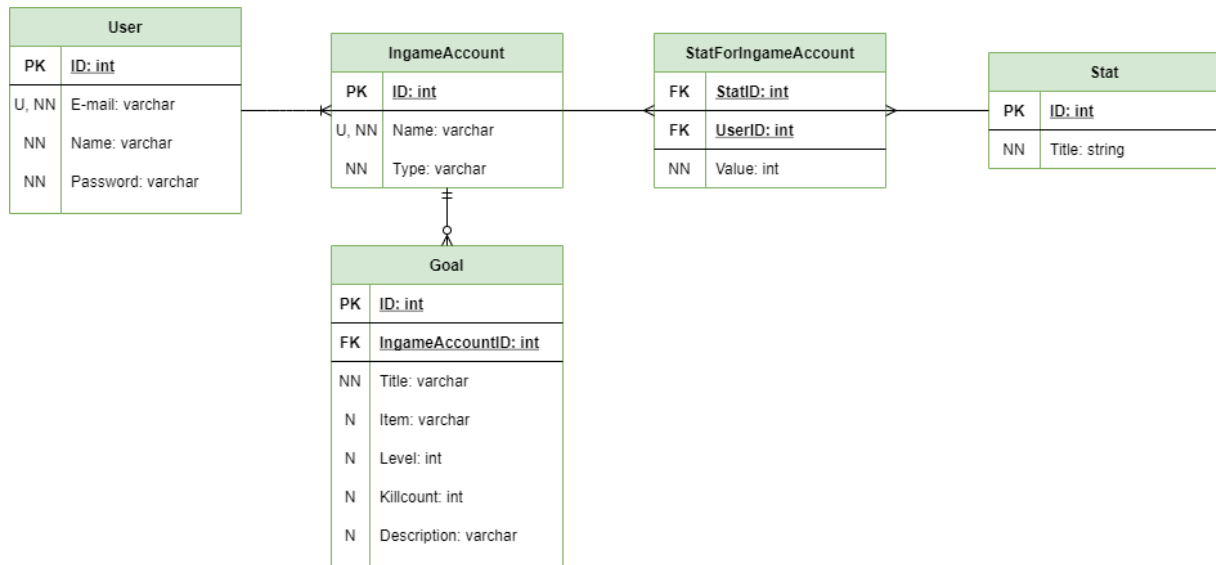
Inhoudsopgave

Inleiding:	2
Database model.....	3
Uitleg Database model	3
Klassendiagram V1	0
Klassendiagram V2	0
Uitleg klassendiagram	0
Meerlaagse architectuur	1
Uitwerking voor 1 entiteit	2

Inleiding:

In dit document staat alles in voor de voorbereiding voor het implementeren van het project. Zodat dit gemakkelijk gaat en ik een overzicht hebt over alle musts in het project.

Database model



Uitleg Database model

U	Unique
NN	Not Null
N	Null

User: Om te kunnen inloggen op de website en Ingame accounts te kunnen koppelen aan een user.

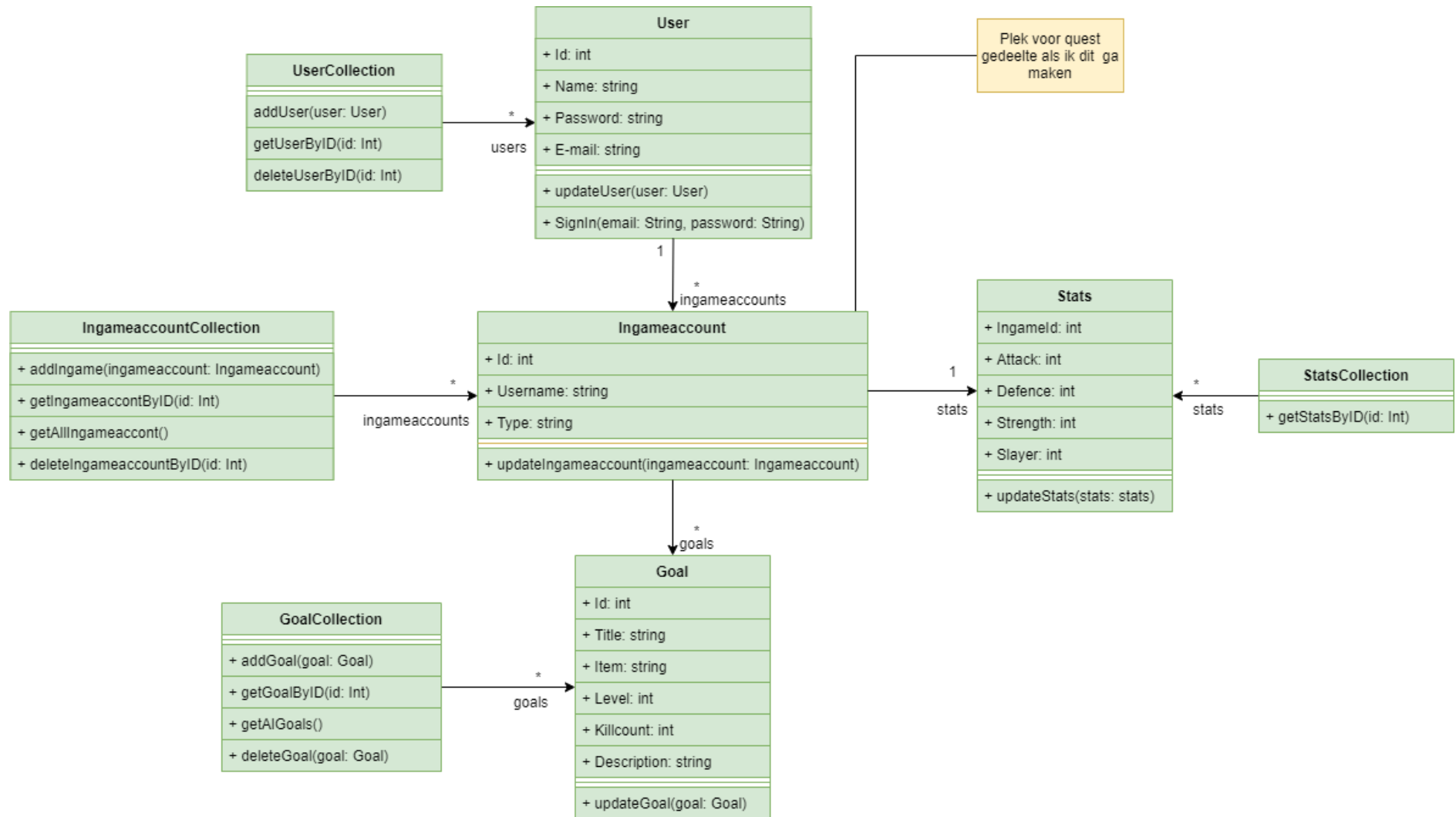
IngameAccount: Dit is vooral om aan te geven wat voor soort account het is ingame en de naam ervan te weten

StatsForIngameAccount: Aan elke ingame account zitten stats verbonden en om te weten welke stat bij wie horen en wat de value hiervan is heb ik deze tabel nodig

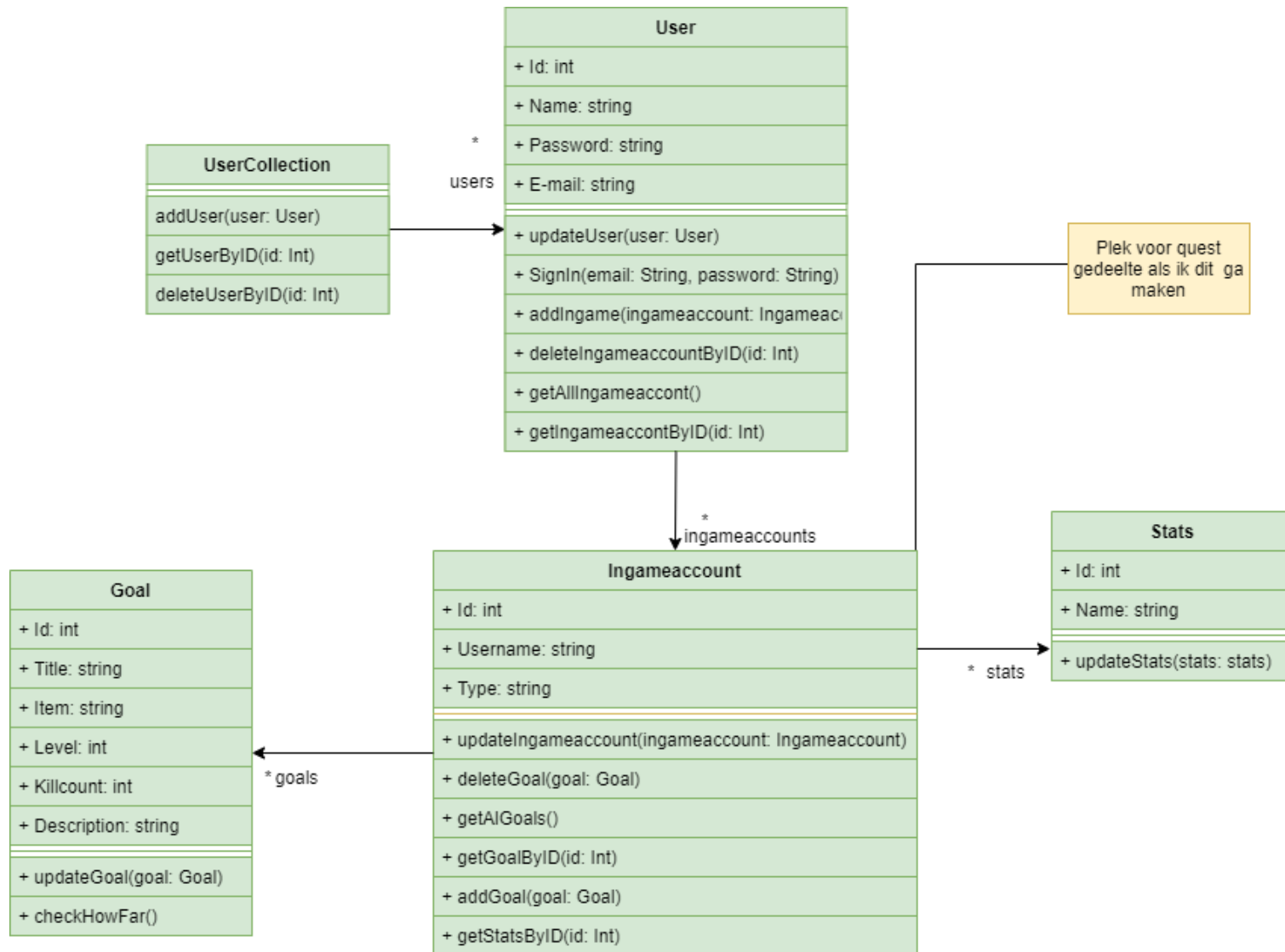
Stat: Om te bepalen welke stat het is.

Goal: Hierin staan alle goals en bij welke ingame account het hoort zodat ik dit makkelijk in een overzicht kan verwerken op de site

Klassendiagram V1



Klassendiagram V2



Uitleg klassendiagram

Groen = must have

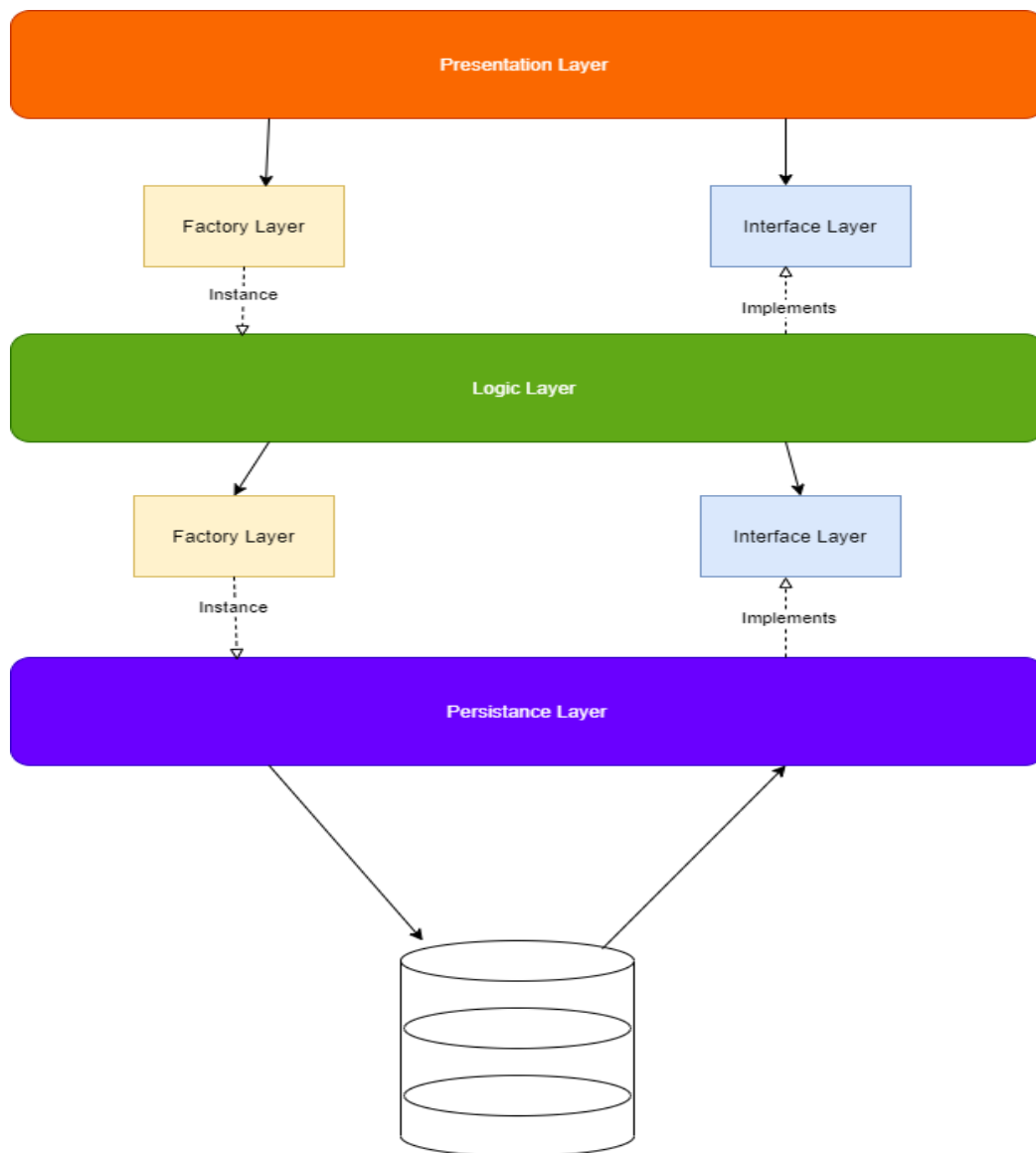
Geel = could have, dit komt als de must have klaar zijn.

V1 dit is hoe het er nu uitziet en V2 moet de uiteindelijke versie worden.

Omdat ik in het begin overal hetzelfde trucje gebruikte had ik voor elk entiteit en eigen collection gemaakt. Maar ik heb in een feedback gesprek erover gehad dat dit niet klopte. Want er kunnen geen goals zijn zonder dat er een ingameaccount bestaat dus moeten de functies die eigen in de GoalCollection staan in ingameaccount staan want deze is ervoor bedoelt. Hetzelfde voor IngameaccountCollection. Zonder een user kan en geen accounts gemaakt worden dus de IngameaccountCollection zou eigenlijk in User moeten horen.

Meerlaagse architectuur

De applicatie zal een Multi-tier lagen structuur hanteren die er als volgend eruit ziet: De UI gedeelte bestaat uit specifiek het MVC structuur die communiceert met de Logica laag waarin alle logica wordt gehanteerd van het berekenen van dingen tot aan het controleren van input en output. De logica laag communiceert met de UI nadat de logica is uitgevoerd om de UI bij te werken. De logica laag communiceert ook met de DAL laag om data te sturen naar en op te vragen vanuit de (lokale) database



Uitwerking voor 1 entiteit

