SOFTWARE ARCHITECTUUR DOCUMENT

Luuk van den Maagdenberg

PROJECT DoodleGuesser

Inhoud

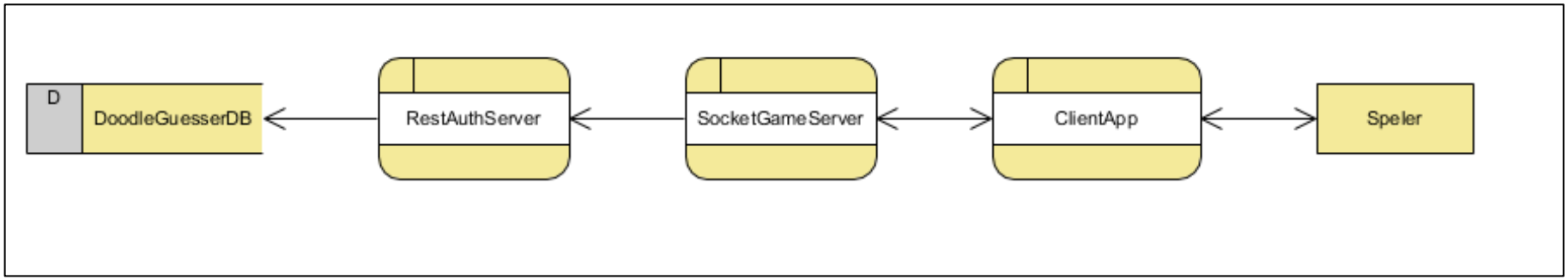
1. Inleiding

Dit document gaat over het spel DoodleGuesser. Het doel van het spel is dat 1 speler een woord krijgt, hij moet vervolgens een tekening maken die het woord uitbeeldt. De andere spelers moeten vervolgens proberen te raden welk woord speler 1 probeert te tekenen.

Wanneer iedereen het woord heeft geraden of de timer is afgelopen, start de volgende ronde en moet een andere speler een woord tekenen.

In dit document staat technische beschrijving van het ontwerp van de DoodleGuesser applicatie.

2. Systeem Context (C1)



3. Containers en Technologiekeuzes (C2)

Rest Authentication Server

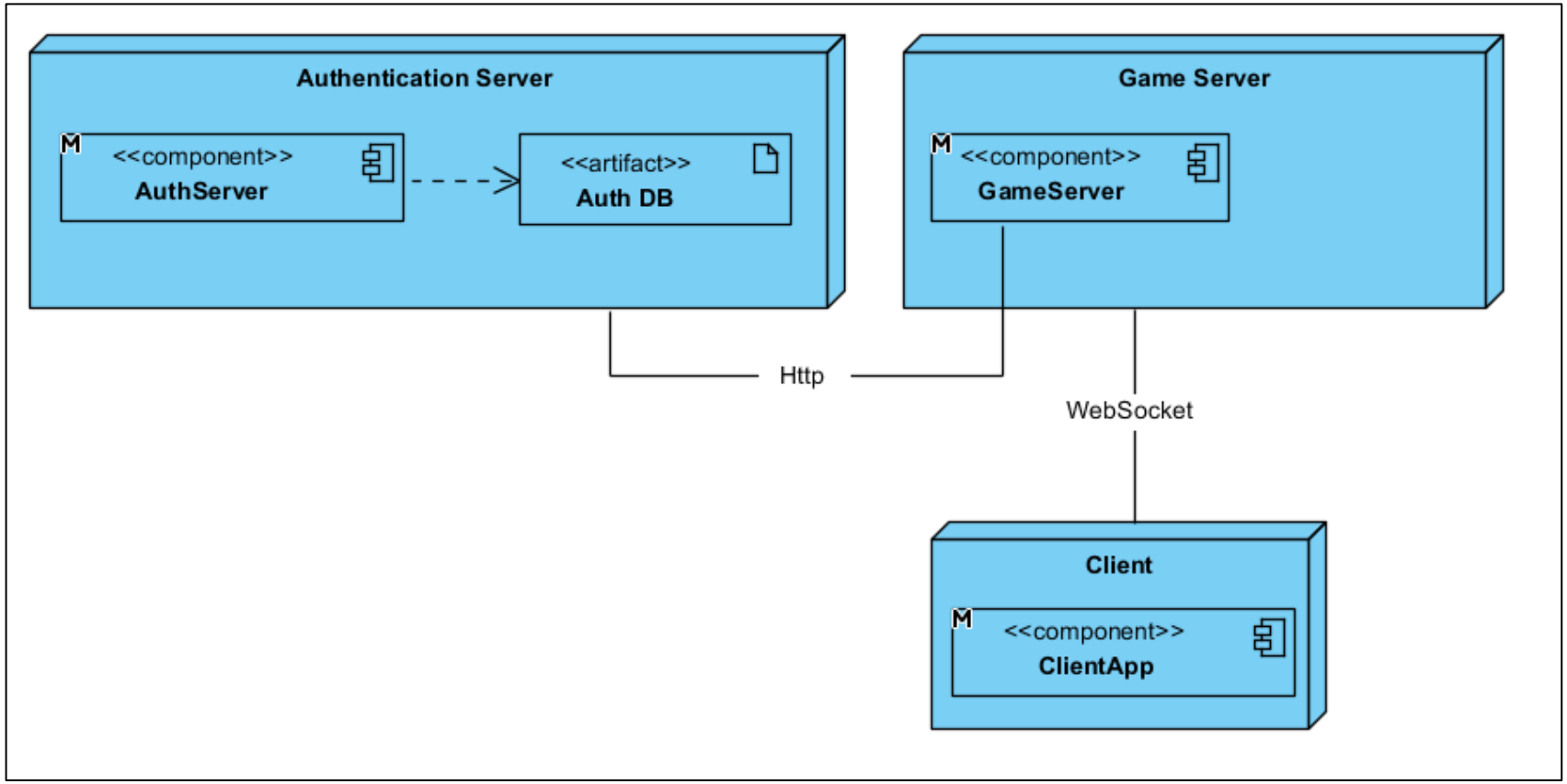
De authentication server is een Jetty server met een REST interface. De authentication server is verantwoordelijk voor authenticatie binnen het systeem. Andere componenten kunnen via de REST interface gebruikers inloggen en registreren. Het REST authentication server component is ook het enige component wat met de authenticatie database verbonden is.

Socket Game Server

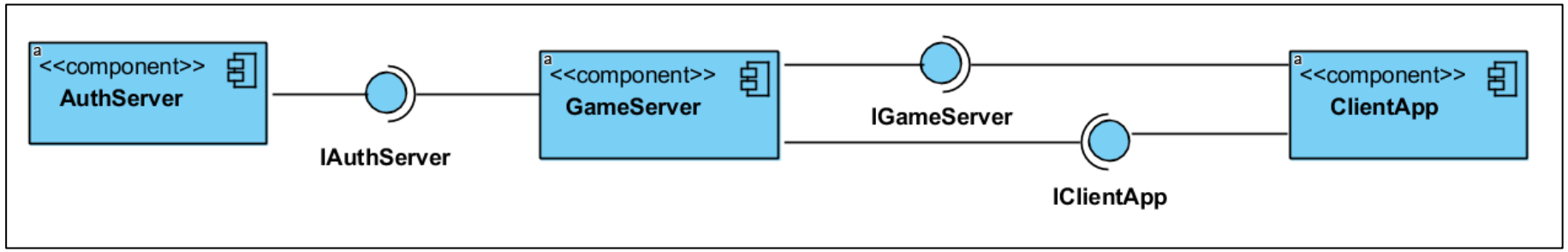
Dit component is het component waar de client apps mee verbinden. Het component voert de game logica uit. Dit component gebruikt de rest authentication server om spelers in te loggen en te registreren. Het component heeft een websocket interface.

Client App

De client app is een JavaFx applicatie. De applicatie is de grafische interface voor de spelers. De client app gebruikt websockets om te communiceren met de game server. De game server heeft alle logica, de client app kan alleen niet functioneren.


Authentication Server
<<component>>
AuthServer
<<artifact>>
Auth DB
Game Server
<<component>>
GameServer
Http
WebSocket
Client
<<component>>
ClientApp

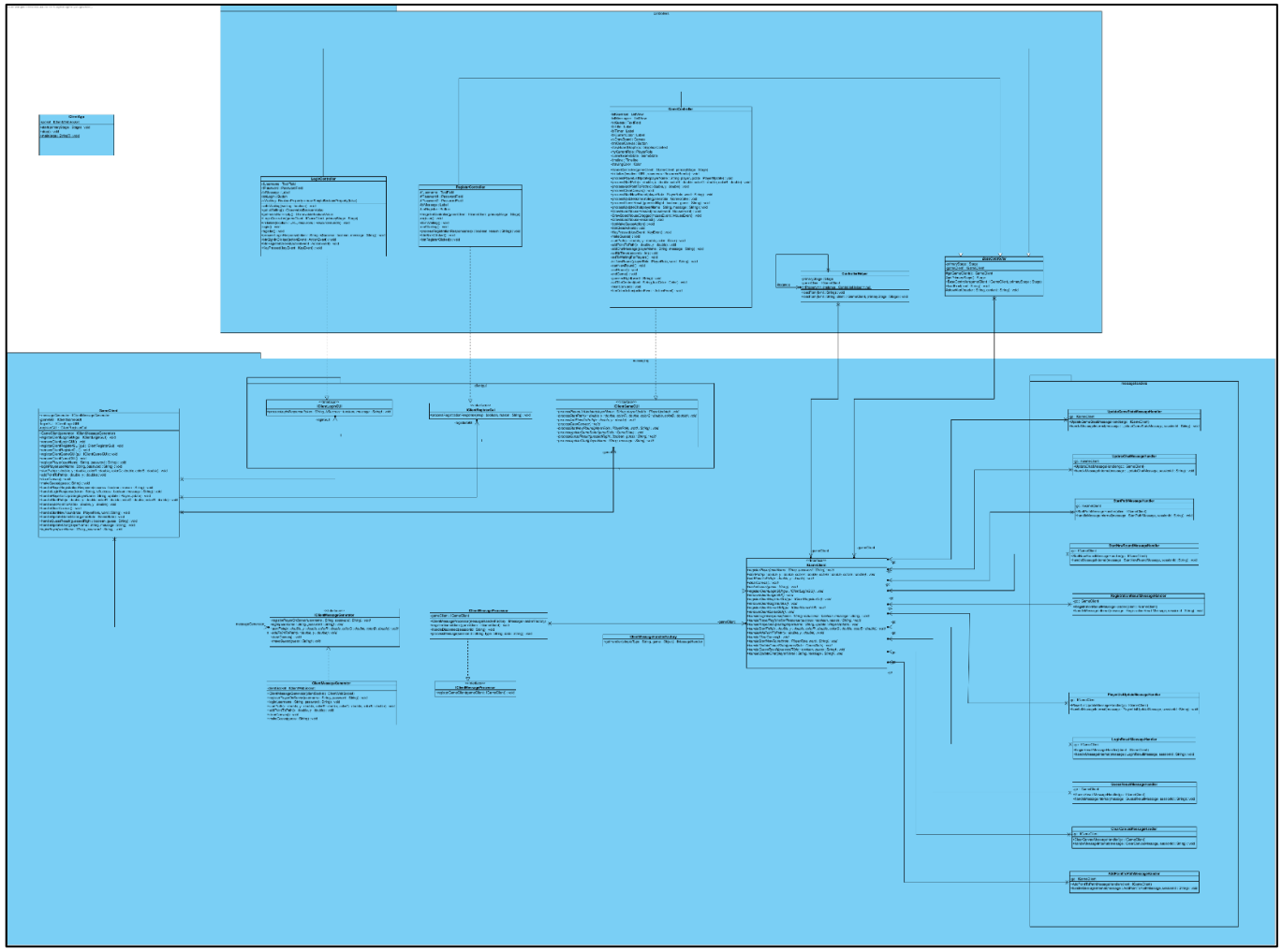
4. Componenten (C3)



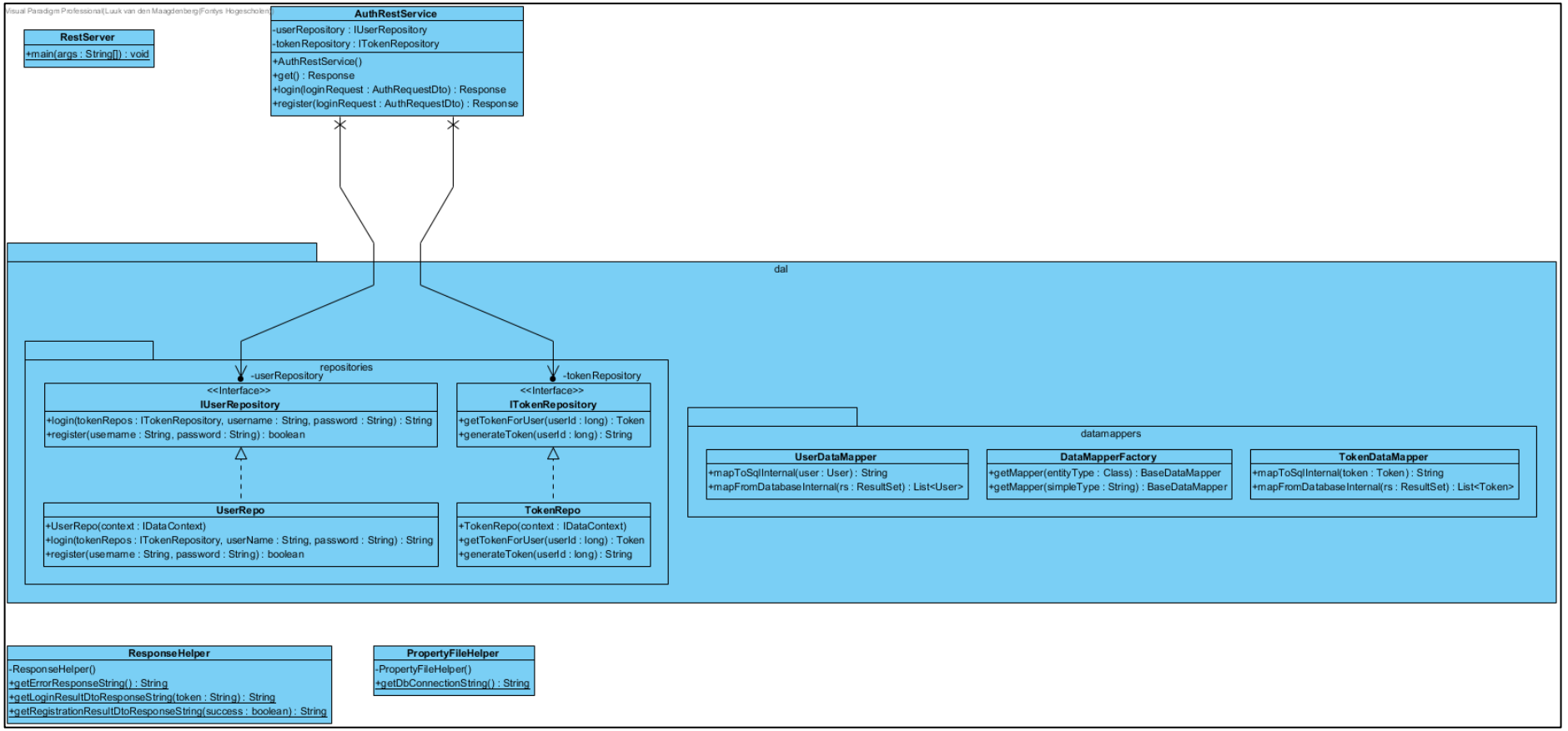
5. Klassen diagrammen en Sequence diagrammen (C4)

1. Klassen diagrammen

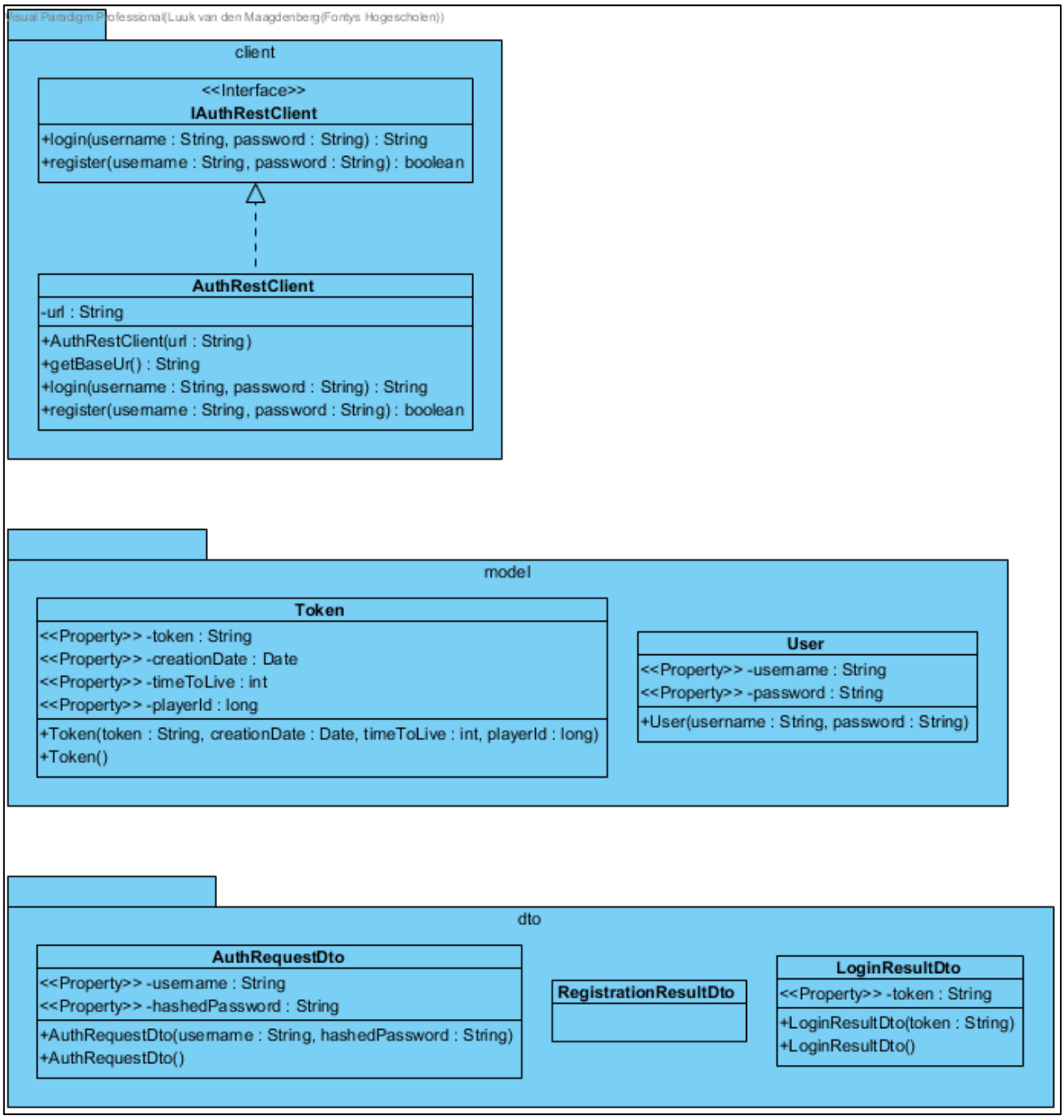
Client app



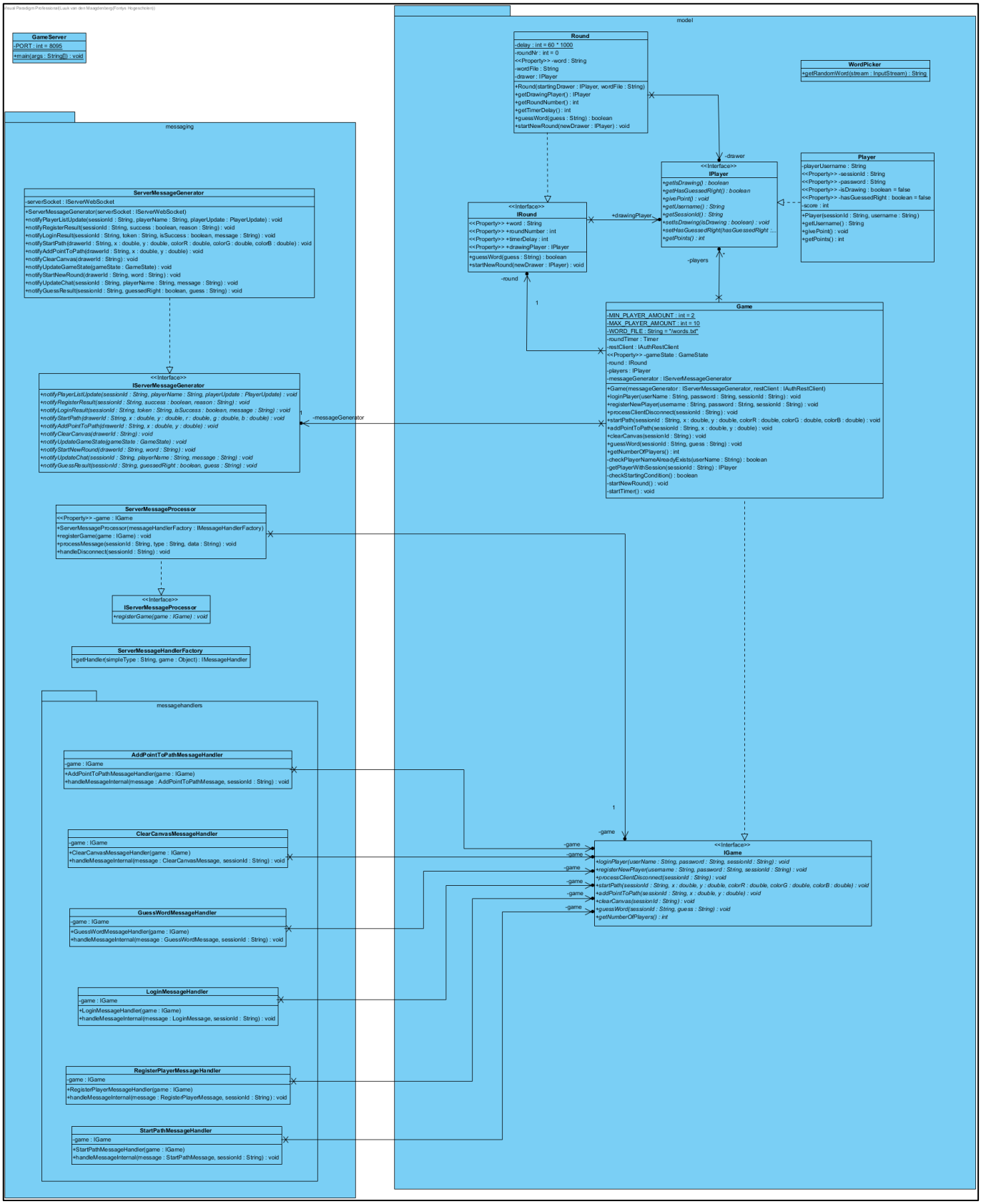
REST auth server



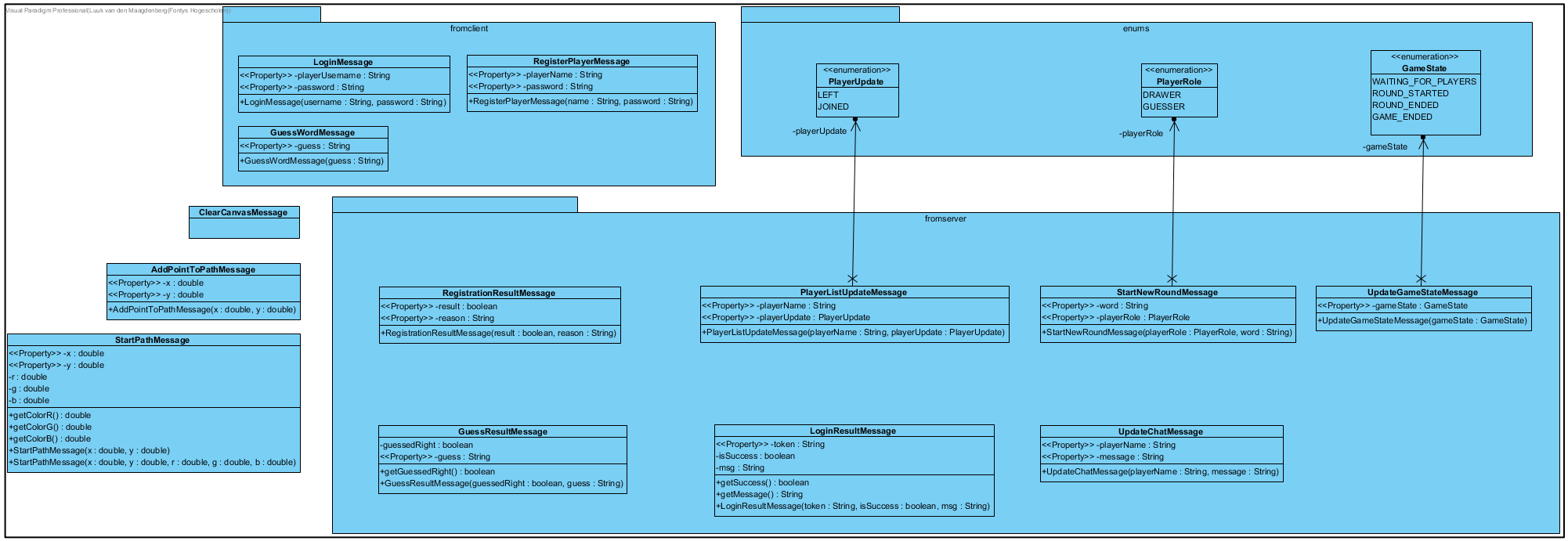
REST auth shared



Socket game server

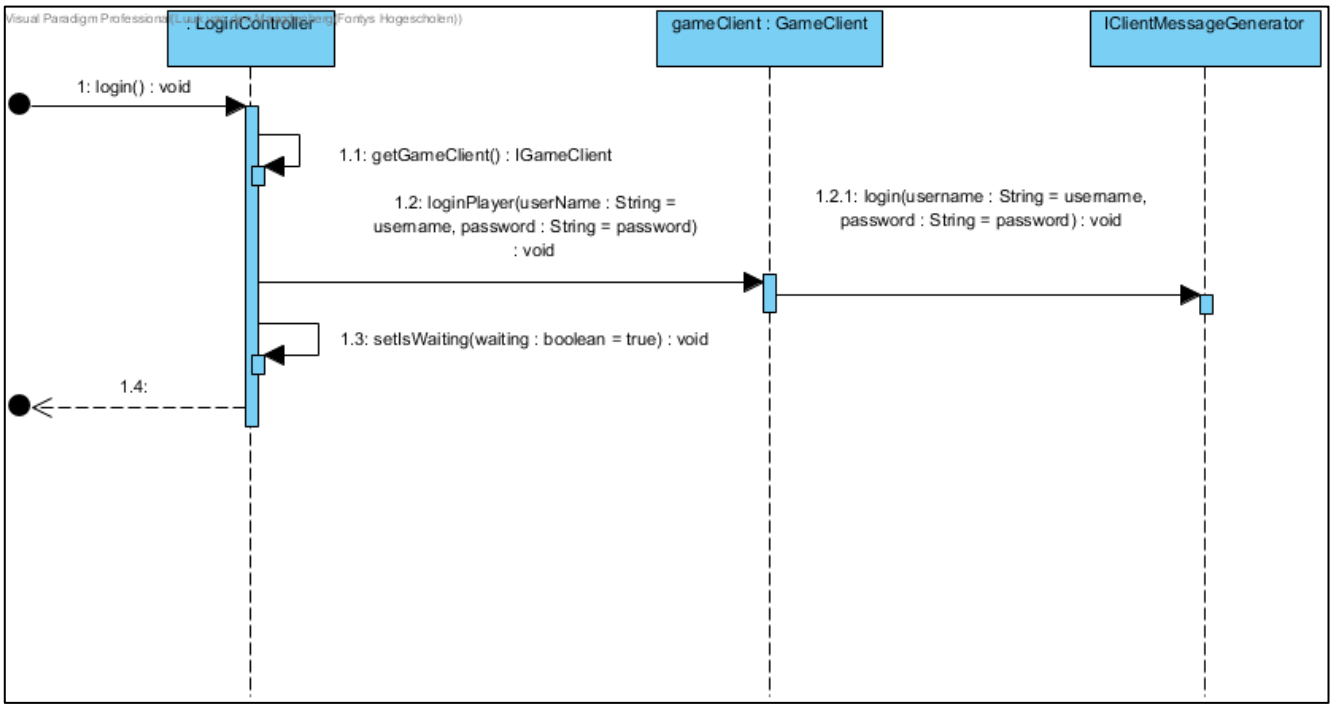


Socket game server shared



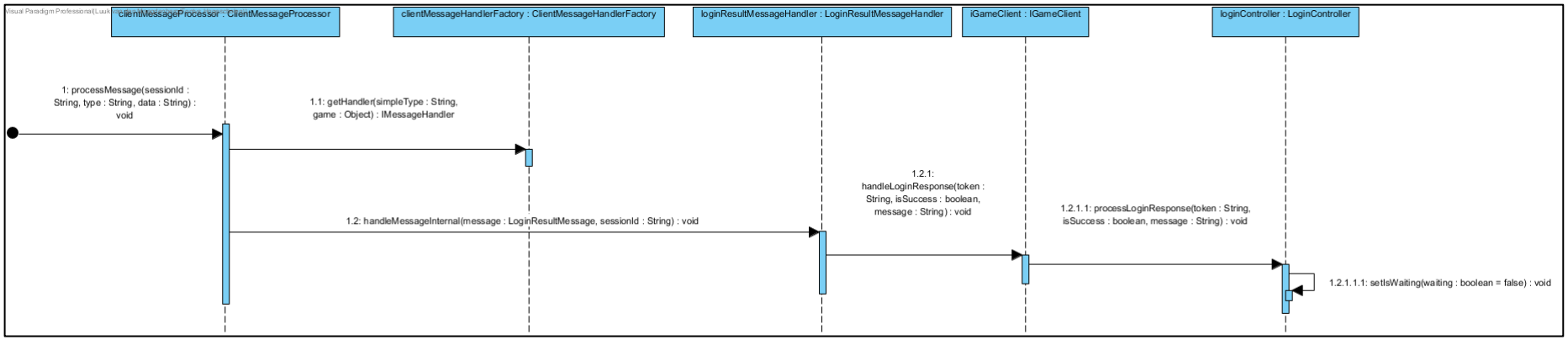
2. Sequence diagrammen

ClientApp Login Message Generator



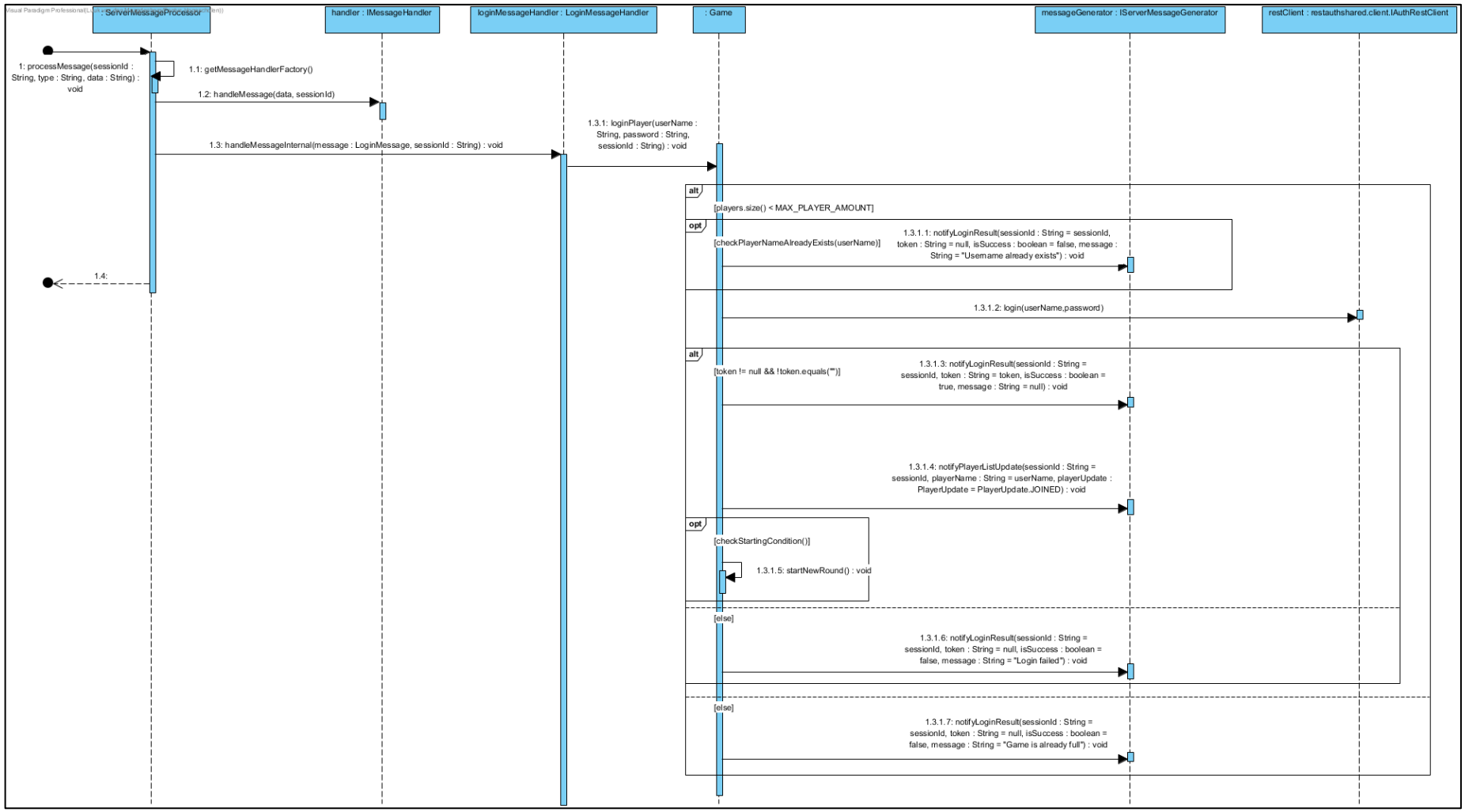
Het versturen van berichten werkt voor elk bericht ongeveer hetzelfde. De controller gebruikt de super klasse BaseController om de GameClient op te halen. De controller kan de GameClient gebruiken om een bericht te sturen. De GameClient roept vervolgens de juiste methode in de MessageGenerator aan, en die verstuurd het bericht met de ClientSocket uit de FontysMultiPurposeLibrary.

ClientApp Login response



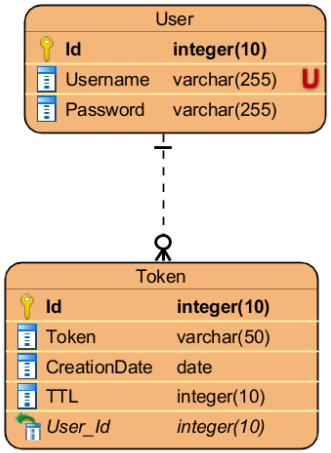
Response van andere berichten werkt in ongeveer dezelfde manier. Het bericht begint bij de MessageProcessor die de MessageHandlerFactory gebruikt om de MessageHandler die bij het bericht past te instantieren. Vervolgens gebruikt de MessageProcessor een generieke methode in de MessageHandler om een methode aan te roepen met de juist parameters in de IGameClient. De GameClient stuurt het vervolgens door naar de juiste controller.

SocketGameServer Player Login



6. Persistentie per component

Accounts en tokens worden in een MSSQL Server opgeslagen. Wanneer een gebruiker inlogt wordt er eerst gekeken of er een vorig token is en of die al verlopen is. Wanneer het token nog niet verlopen is wordt het oude token gebruikt, anders wordt er een nieuw token aangemaakt. De gebruiker kan ook een token gebruiken om in te loggen, dit kan handig zijn wanneer een gebruiker zijn applicatie opnieuw op wil starten zonder opnieuw te moten inloggen.



7. Specificatie van interfaces

De specificaties van de REST API staan in de vorm van Swagger op:

{server address}:8096/docs/

De websocket interface bestaat uit een aantal Message klassen die kunnen worden geserialiseerd naar JSon. Vervolgens kunnen ze naar de server worden verstuurd.

Authorisatie messages

Message	Response message
socketgameshared.messaging.messages.fromclient.LoginMessage	socketgameshared.messaging.messages.fromserver.LoginResultMessage
socketgameshared.messaging.messages.fromclient.RegisterPlayerMessage	socketgameshared.messaging.messages.fromserver.RegistrationResultMessage

In game messages

From client

Message
socketgameshared.messaging.messages.fromclient.GuessWordMessage
socketgameshared.messaging.messages.AddPointToPathMessage
socketgameshared.messaging.messages.ClearCanvasMessage
socketgameshared.messaging.messages.StartPathMessage

From server

Message
socketgameshared.messaging.messages.AddPointToPathMessage
socketgameshared.messaging.messages.ClearCanvasMessage
socketgameshared.messaging.messages.StartPathMessage
socketgameshared.messaging.messages.fromserver.GuessResultMessage
socketgameshared.messaging.messages.fromserver.PlayerListUpdateMessage
socketgameshared.messaging.messages.fromserver.StartNewRoundMessage
socketgameshared.messaging.messages.fromserver.UpdateChatMessage
socketgameshared.messaging.messages.fromserver.UpdateGameStateMessage