hogeschool Windesheim
Dichter bij jou

dinsdag 26 maart 2024

Technisch Ontwerp Showcase

Versie 0.4

Martijn Schuman
Client & Server & Security
S1186586
ICTWDf0
H. Bosman & R. Hulsing & J. Brouwers

Algemene informatie

Versiebeheer

Versie	Datum	Omschrijving
0.1	6-2-2024	Initiële opzet
0.2	3-3-2024	C4 model level 1 & 2 uitgewerkt
0.3	4-3-2024	C4 model level 3 uitgewerkt
0.4	26-3-2024	C4 model bijwerken

Distributie

Ontvanger	Datum	Versie
H. Bosman & R. Hulsing & J. Brouwers	04-03-2024	0.3
H. Bosman & R. Hulsing & J. Brouwers	26-03-2024	0.4

Inhoudsopgave

1 Inleiding

In dit technisch ontwerp wordt een systeem beschreven dat zich richt op het ontwerp en de implementatie van de use cases die zijn beschreven in het Functioneel Ontwerp. Dit systeem stelt gebruikers in staat om een beeld te vormen van de skills van een developer, de Showcase. Het ontwerp omvat een gedetailleerde analyse van de systeemcontext, container- en componentdiagrammen, authenticatie mechanismen, deployment aspecten en de mailfunctionaliteit.

De systeemcontext omvat de koppeling met externe systemen en gebruikersinterfaces. Het systeem maakt gebruik van een externe e-mailserver om e-mails te verzenden en ontvangen.

Het containerdiagram biedt een overzicht van de verschillende containers waaruit het systeem is opgebouwd. De belangrijkste containers omvatten de applicaties voor de webinterface en de database. Deze containers werken samen om de functionaliteit van het systeem te leveren.

Het componentdiagram biedt een gedetailleerdere weergave van de interne structuur van het systeem. De belangrijkste componenten omvatten gebruiker beheer, informatieverstrekking en e-mailverwerking. Elk component heeft specifieke verantwoordelijkheden en interacties met andere componenten om de gewenste functionaliteit te bereiken.

Veiligheid is van het grootste belang voor het systeem, met name bij het verifiëren van de identiteit van gebruikers. Ook is uitgewerkt hoe de beide applicaties met elkaar communiceren op een beveiligde manier.

De deployment van het systeem omvat de configuratie en implementatie van de verschillende componenten op de juiste infrastructuur. Het systeem wordt ingezet op Cloudflare en Skylab.

Dit technisch ontwerp biedt een uitgebreide beschrijving van het systeem, inclusief de systeemcontext, container- en componentdiagrammen, authenticatiemechanismen, deploymentaspecten en de mailfunctionaliteit. Het vormt een solide basis voor de ontwikkeling en implementatie van een efficiënte en veilige toepassing.

2 Systeem Context

Dit hoofdstuk geeft een overzicht van het systeem.

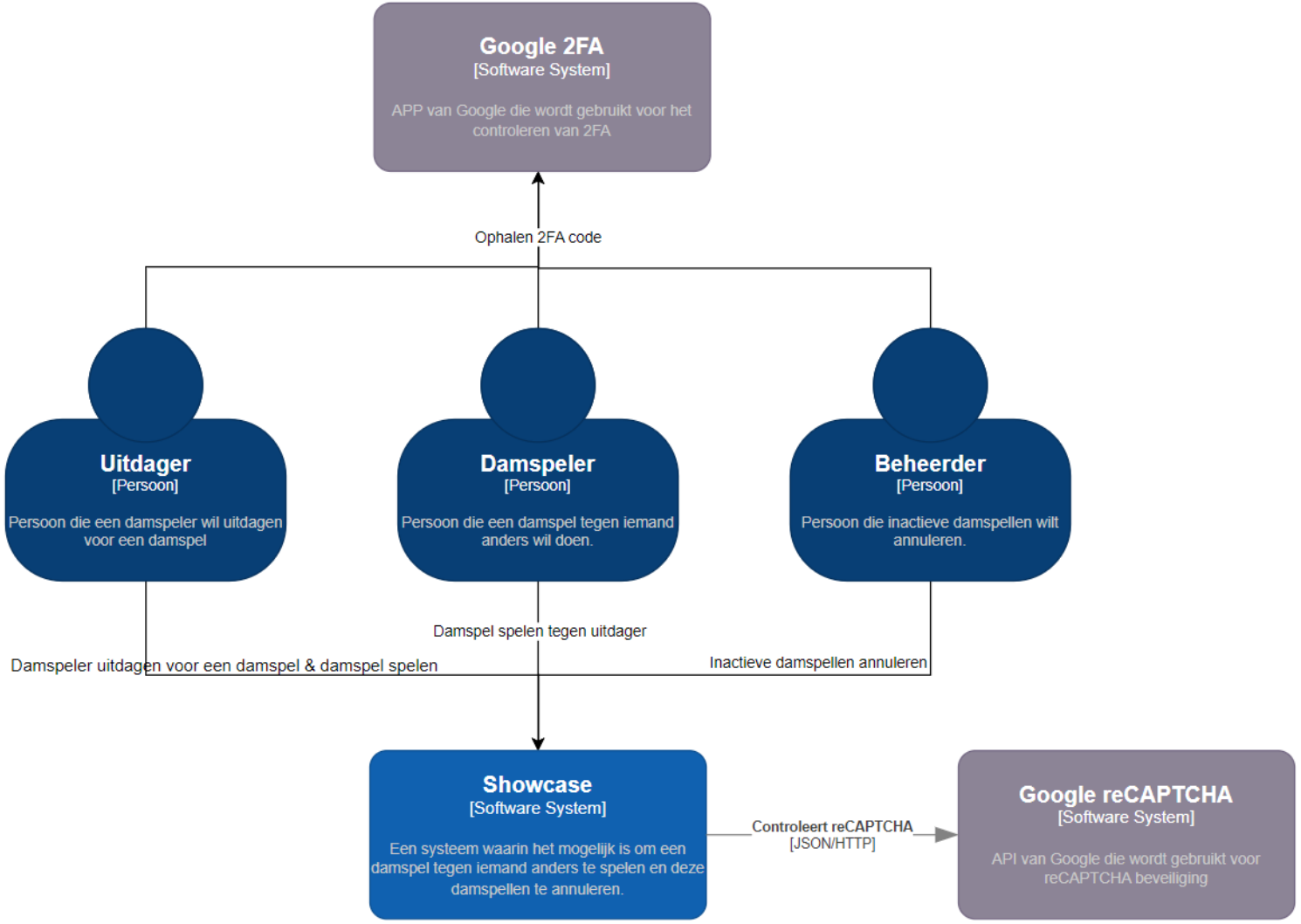
2.1 Primaire actoren

In de onderstaande tabel is een overzicht van de verschillende actoren te zien.

Actor	Omschrijving
Uitdager	Deze actor kan zich registreren bij het platform en een damspel aanmaken. Als het spel is gemaakt kan hij een code met een damspeler delen, waarna hij tegen deze speler gaat spelen.
Damspeler	Deze actor kan zich registeren bij het platform en via een code zich aansluiten bij een spel en tegen de uitdager spelen.
Beheerder	Deze actor kan damspellen annuleren als deze een lange tijd geen activiteit hebben gehad.

2.2 Externe systemen

In het onderstaande diagram is te herleiden de verschillende actoren omgaan met het systeem. Verder is ook te zien welke externe systemen er worden gebruikt.



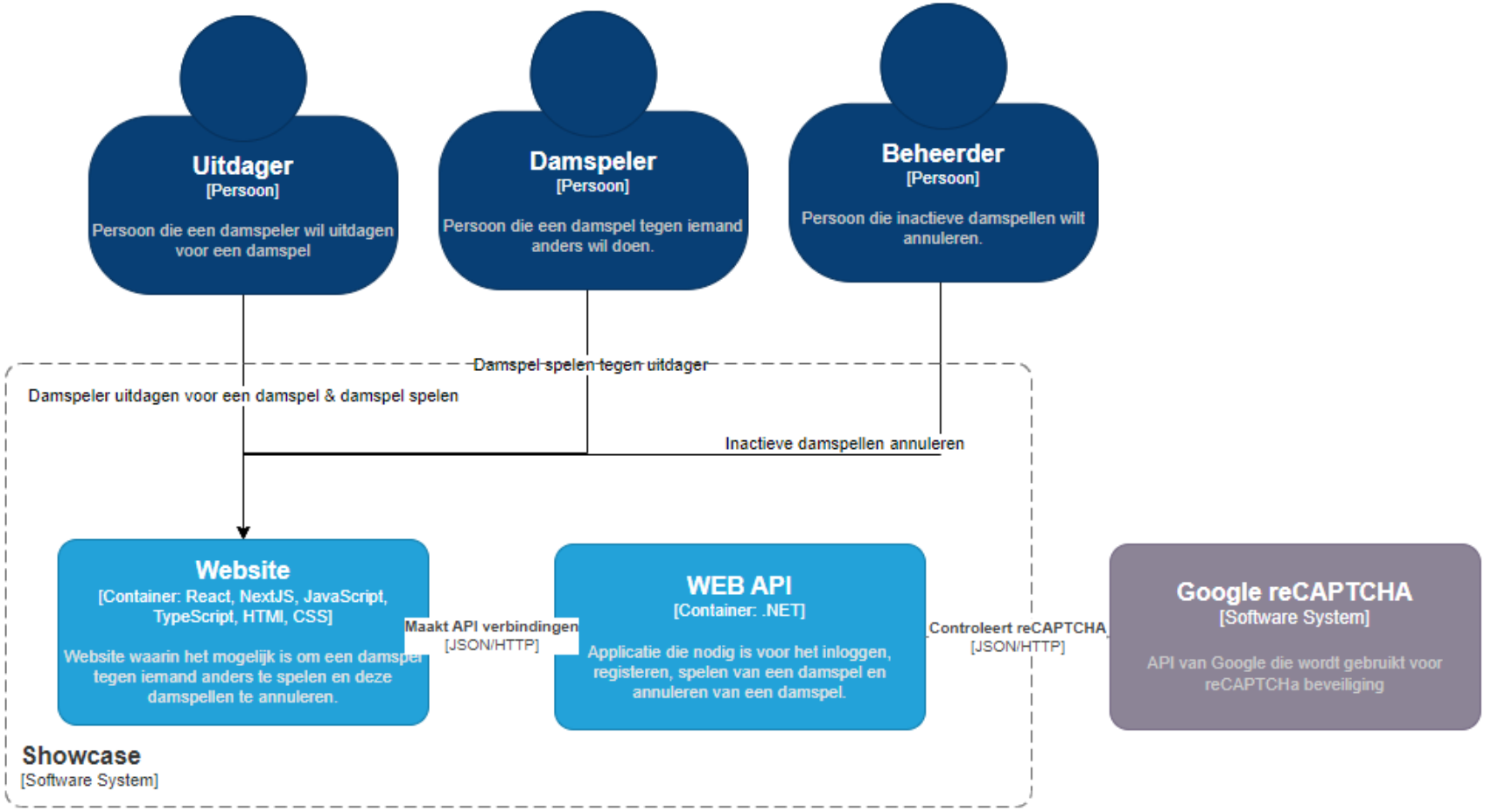
Figuur 1 - globaal overzicht van de showcase

3 Containers

Dit hoofdstuk beschrijft de applicaties waaruit het systeem bestaat. Ook is de communicatie tussen de applicaties en de database beschreven.

Op basis van dit hoofdstuk is een Threat Model ontwikkeld die zijn vastgelegd in het rapport Threat Model Showcase Report. De bedreigingen en de gekozen maatregelen zijn vastgelegd in het document Threat List.xlsx en moeten nog worden doorgevoerd in het Risk Assessment document.

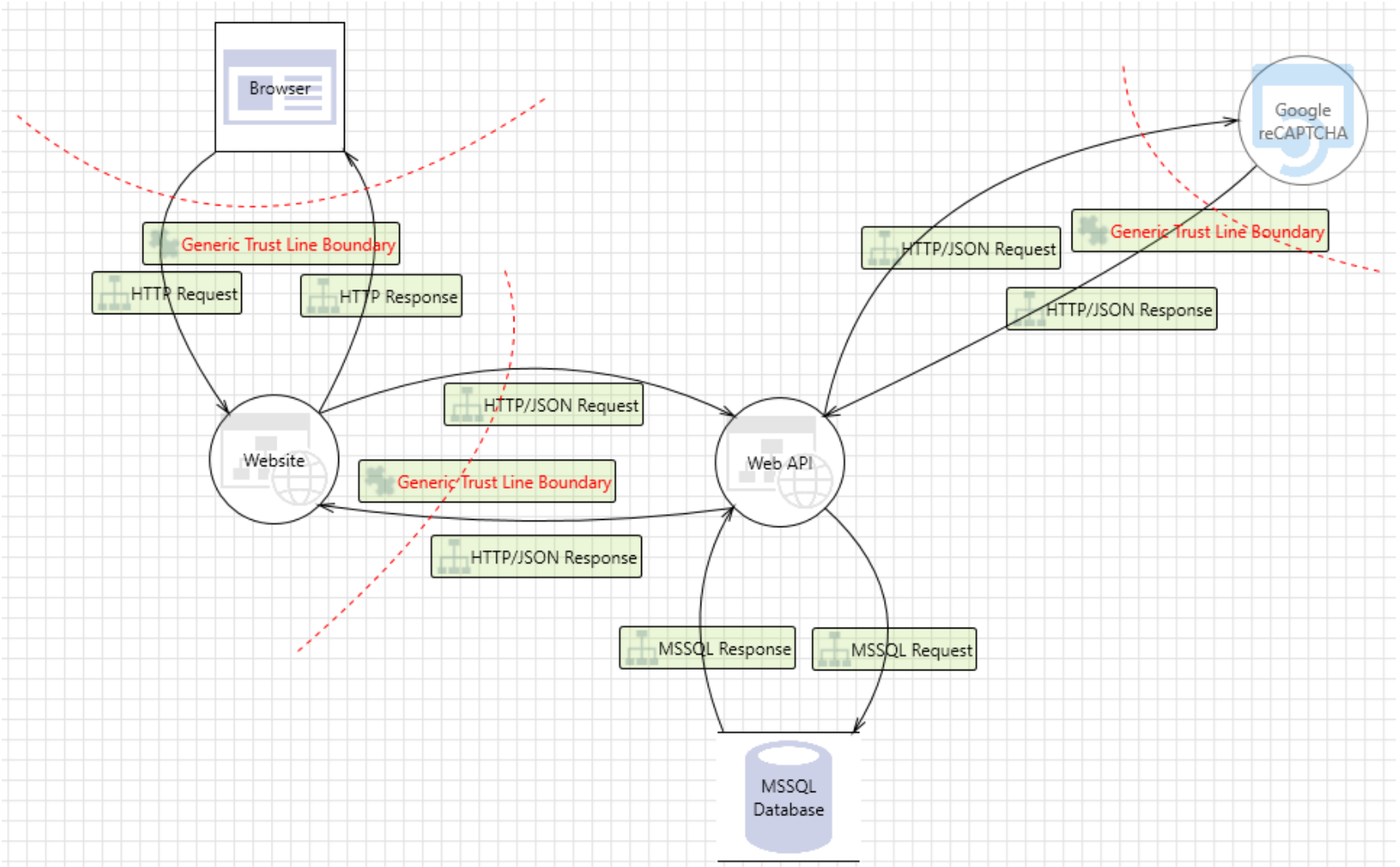
3.1 Applicaties



Figuur 2 - container overzicht van de showcase

Threat modeling op Container level

Op basis van de containers is het onderstaande threat model gemaakt. In het model is te zien hoe de systemen onderling met elkaar communiceren en waar de trust line boundaries zitten.



Figuur 3 - threat model gebaseerd op de containers

3.2 Security Maatregelen

In eerste instantie is bepaald welke items van toepassing zijn door per communicatie stap voor het versturen van een contactverzoek nagegaan welke threat aanwezig is welke securitymaatregelen noodzakelijk is.

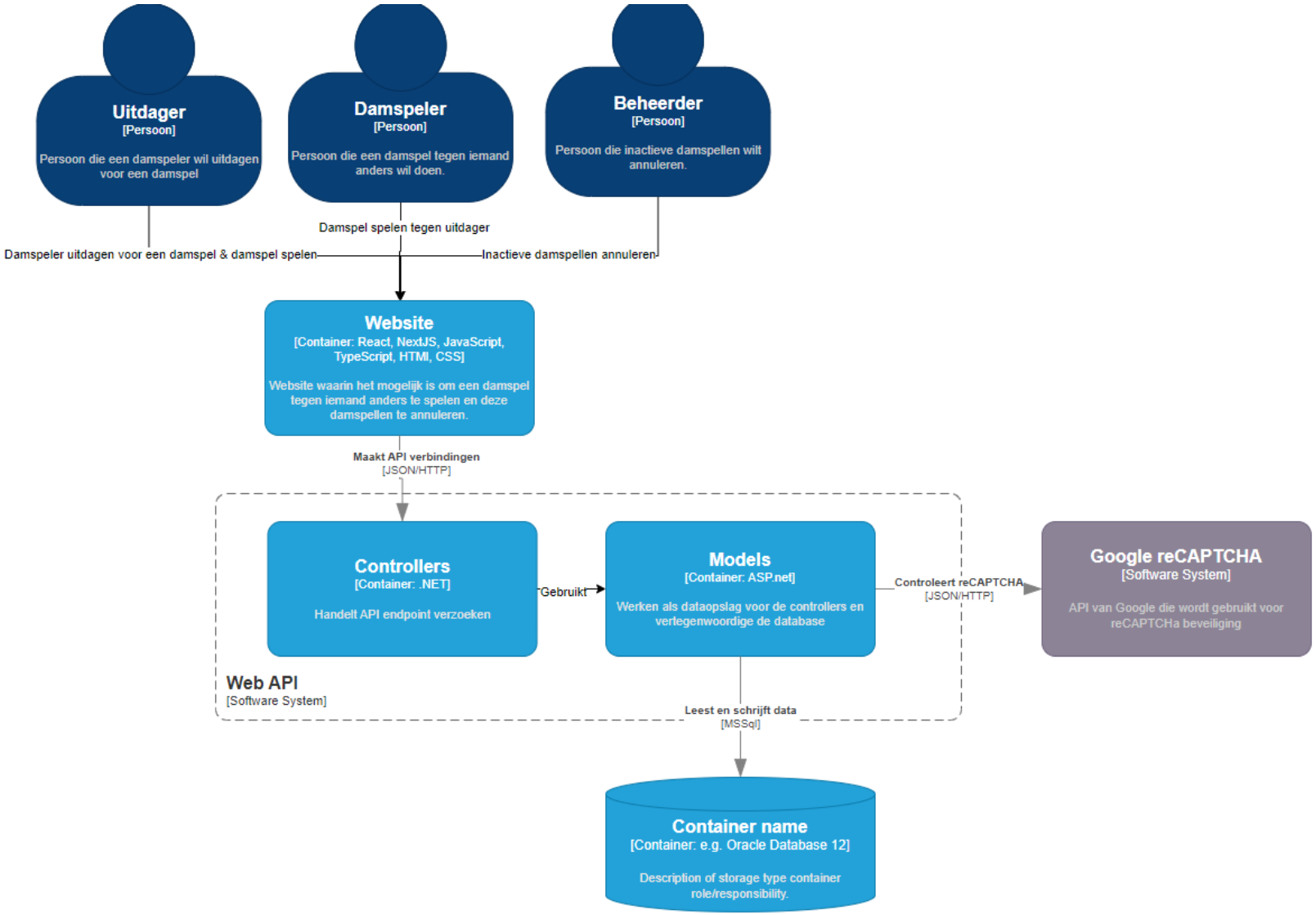
Een aantal algemene opmerkingen. Bij de implementatie van de eerste twee user stories is nog geen sprake van authenticatie, autorisatie en sessie. De items uit het Threat Model die hierop betrekking hebben zijn dus (nog) niet toegepast. Ook is in dit ontwerp uitgegaan van een situatie waarbij geen packages worden opgehaald bij externe partijen.

Stap	Threat	SM
Ophalen pagina (HTTP)	13 An adversary can spoof the target web application due to insecure TLS certificate configuration. 16 An adversary can create a fake website and launch phishing attacks. 21 An adversary can gain access to sensitive data stored in Web App's config files	# SSL / HTTPS verplichten # # Config bestanden verbergen
Ophalen Javascript en CSS	18 An adversary can deface the target web application by injecting malicious code or uploading dangerous files	# Inline javascript uitschakelen
Invoergegevens	19 An adversary can gain access to sensitive data by performing SQL injection through Web App 28. An adversary may inject malicious inputs into an API and affect downstream processes 29. An adversary can gain access to sensitive data by performing SQL injection through Web API	# Input validatie # Prepared statements # Input validatie # Prepared statements # Input validatie # Prepared statements
Validatie gegevens	18 An adversary can deface the target web application by injecting malicious code or uploading dangerous files	# Input validatie # Tekst als string weergeven en niet *ruwe HTML.*
Versturen gegevens	8 An adversary may gain access to sensitive data from uncleared browser cache.	# Generieke foutmeldingen

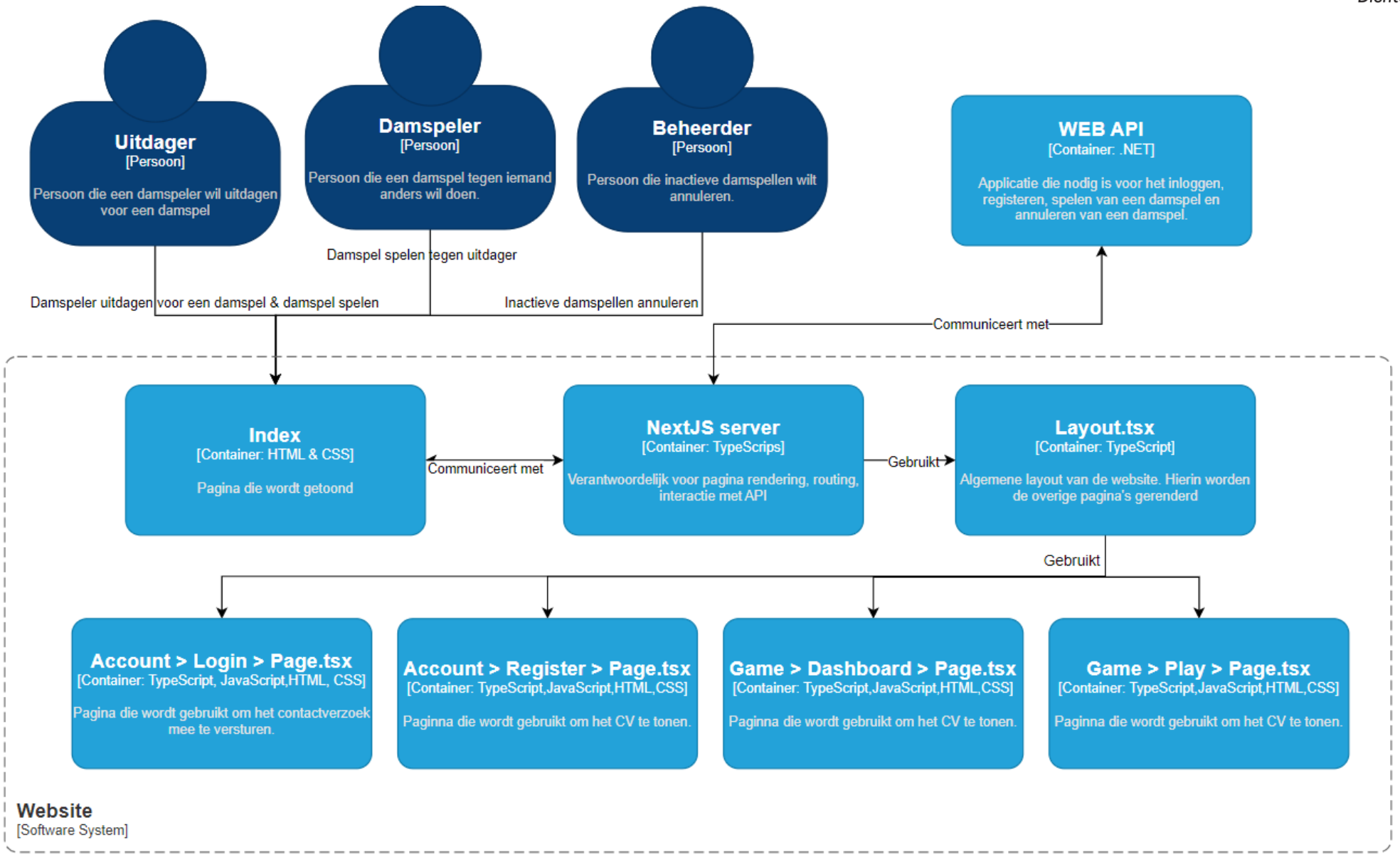
	24 An adversary can gain access to sensitive information through error messages. 25 Attacker can deny the malicious act and remove the attack footprints leading to repudiation issues. 26 Attacker can deny a malicious act on an API leading to repudiation issues. 27. An adversary may spoof Web API and gain access to Web AP 30. An adversary can reverse weakly encrypted or hashed content	# Geavanceerde logging toevoegen # Errors en warnings binnen de WebAPI worden gelogd naar de runtime console en een log bestand. # WebAPI calls moeten worden gevalideerd met een JWT-security token. Admin endpoints moeten worden bevestigd met een 2FA code # Zorgen voor sterke hashing en encryptie algoritmes

4 Componenten

De website bestaat uit twee onderdelen. In het onderstaande diagram is gemodelleerd hoe de website met de Web API communiceert.



Figuur 4 - component overzicht van de Web API



Figuur 5 – component overzicht van de website

5 Structuur

In de vierde laag wordt er dieper ingegaan op de systeemarchitectuur. Dit wordt gedaan aan de hand van klassendiagrammen en sequentiediagrammen. Onderstaand staan meerdere klassendiagrammen die elk een belangrijk onderdeel van het systeem toelichten. Enkel de belangrijkste klassen, methodes en properties zijn uitgewerkt. Vervolgens worden er sequentiediagrammen getoond. In de sequentiediagrammen is de flow van de applicatie duidelijk te zien. Direct na de sequentiediagrammen wordt er over het diagram een toelichting gegeven en eventuele belangrijke ontwerpbeslissingen toegelicht.

Gezamenlijk tonen deze diagrammen een uitgebreid overzicht van de systeemarchitectuur, waarbij zowel de statische structuur als de dynamische interacties van het systeem worden weergegeven.

Code

Voor de client is er gekozen om te werken met Next.JS v14.1.0. Next.JS is een framework dat is gebouwd op React. Dit framework biedt een eenvoudige ontwikkeling van server-side rendering (SSR), static site generation (SSG), en client-side rendering (CSR) voor React applicaties. Hierdoor kan gebruik gemaakt worden van de voordelen van zowel SSR als SSG, wat resulteert in een snelle en schaalbare front-end ervaring.

Voor de webapi is gekozen om gebruik te maken van .NET 8.0 vanwege de snelheid, betrouwbaarheid en uitgebreide functionaliteit die het biedt. Bovendien is .NET Core cross-platform, wat betekent dat onze webapi kan worden gehost op verschillende besturingssystemen, waaronder Windows, Linux en macOS.

Database

Voor de database is gekozen voor Microsoft SQL Server. Deze keuze is gemaakt vanwege de betrouwbaarheid, schaalbaarheid en uitgebreide functionaliteit die SQL Server biedt. SQL Server is een krachtig relationeel databasebeheersysteem. Bovendien integreert SQL Server zonder problemen met andere Microsoft-producten en -technologieën, waardoor het in combinatie met .NET een ideale keuze is.

Client Packages

In de Next.JS client applicatie wordt er gebruik gemaakt van enkele packages. De packages zijn gekozen op basis van een aantal criteria. De package moet recent updates hebben ontvangen moet genoeg maandelijkse downloads hebben.

Package	Versie	Toelichting
fontawesome	6.5.1	Deze package wordt gebruikt om eenvoudig iconen in te laden.
tailwindcss	3.4.1	Deze package wordt gebruikt om de website van eenvoudige en duidelijke opmaak te voorzien.
js-cookie	3.0.6	Deze package wordt gebruikt om eenvoudig Cookies in te stellen.
flowbite	2.3.0	Deze package wordt gebruikt om bepaalde TailwindCSS componenten correct te laten werken.
flowbite-react	0.7.2	Deze package wordt gebruikt om specifieke React Flowbite componenten te laten werken.

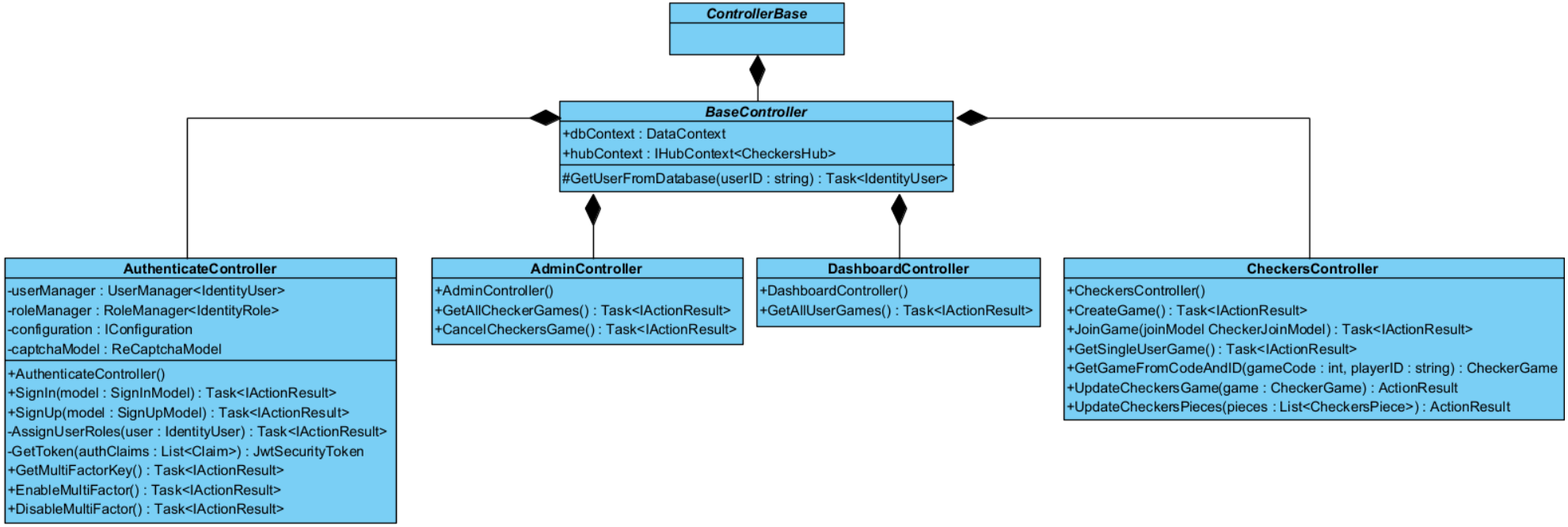
Server Packages

In de .NET API-applicatie wordt er gebruik gemaakt van een aantal packages. De packages zijn gekozen op basis van een aantal criteria. De package moet recent updates hebben ontvangen moet genoeg maandelijkse downloads hebben.

Package	Versie	Toelichting
DotNetEnv	3.0.0	Deze package wordt gebruikt om eenvoudig environment variabelen zoals inloggegevens of API-sleutels op te slaan.
NewtonSoft.Json	13.0.3	Deze package wordt gebruikt gemaakt om eenvoudige JSON-response body's mee op te zetten.
Serilog	3.1.1	Deze package wordt gebruikt als basis voor het loggen van API-toegang.
Serilog.Sinks.Console	5.0.1	Deze package wordt gebruikt om log acties te printen naar de console.
Serilog.Sinks.File	5.0.0	Deze package wordt gebruikt om log acties op te slaan in een log bestand.
Microsoft.AspNetCore.Authentication.JwtBearer	8.0.2	Deze package maakt het mogelijk om binnen de applicatie JWT Bearer tokens te kunnen gebruiken.
Microsoft.EntityFrameworkCore.Design	8.0.2	Deze package maakt het mogelijk om door middel van een DbContext met de database te kunnen verbinden.
Microsoft.EntityFrameworkCore.Tools	8.0.2	Deze package maakt het mogelijk om de database te onderhouden en op te bouwen.
Microsoft.EntityFrameworkCore.SqlServer	8.0.2	Deze package maakt het mogelijk om met een SqlServer verbinding te maken
Swashbuckle.AspNetCore	6.5.0	Deze package maakt het mogelijk om doormiddel van Swagger een API-documentatie op te zetten.
TwoFactorAuth.Net	1.4	Deze package maakt het mogelijk om 2FA validatie toe te voegen

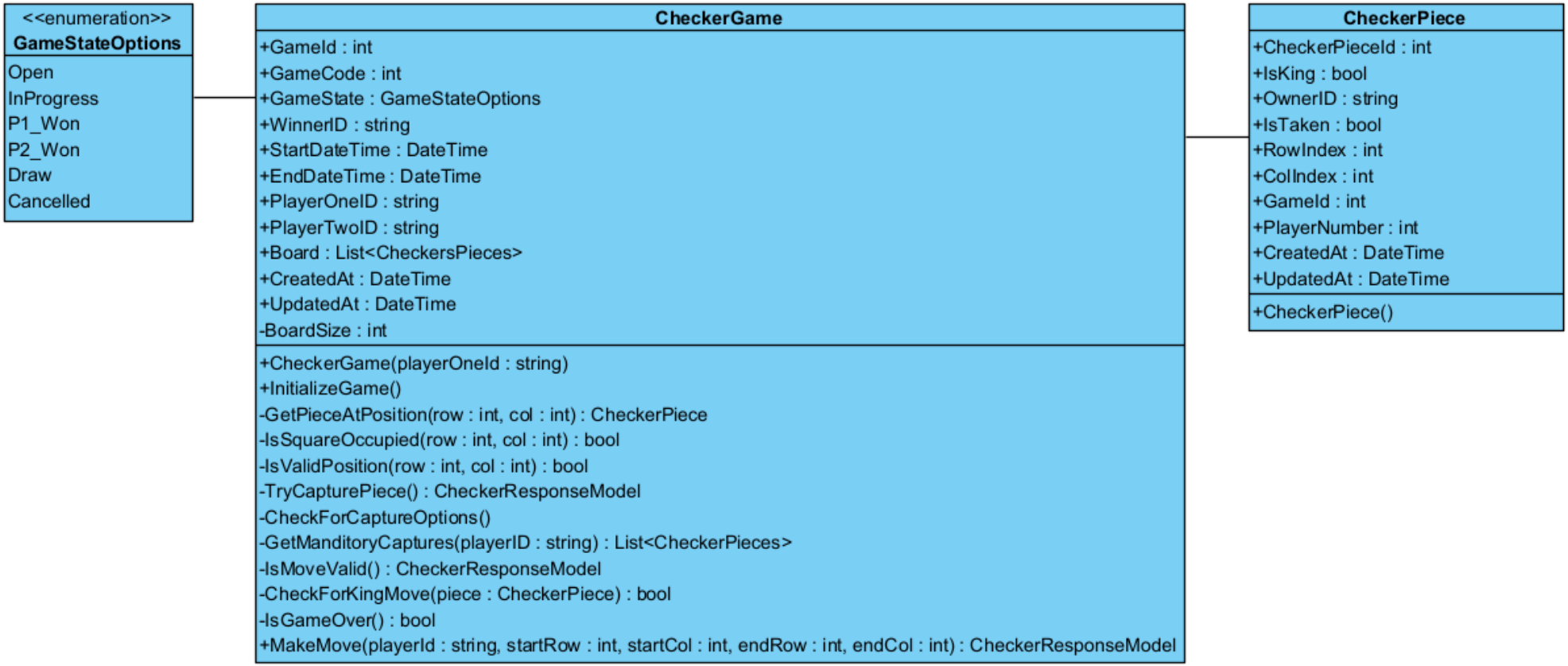
5.1 Klassendiagrammen

In het onderstaande diagram is een globaal overzicht van de .NET webapi weergegeven. In het diagram zijn de verschillende controllers met hub properties en methodes uitgewerkt. Verder is de DataContext; de verbinding met de database uitgewerkt en is de SignalR verbinding hub uitgewerkt.



Figuur 6 - klassendiagram van de .NET webapi

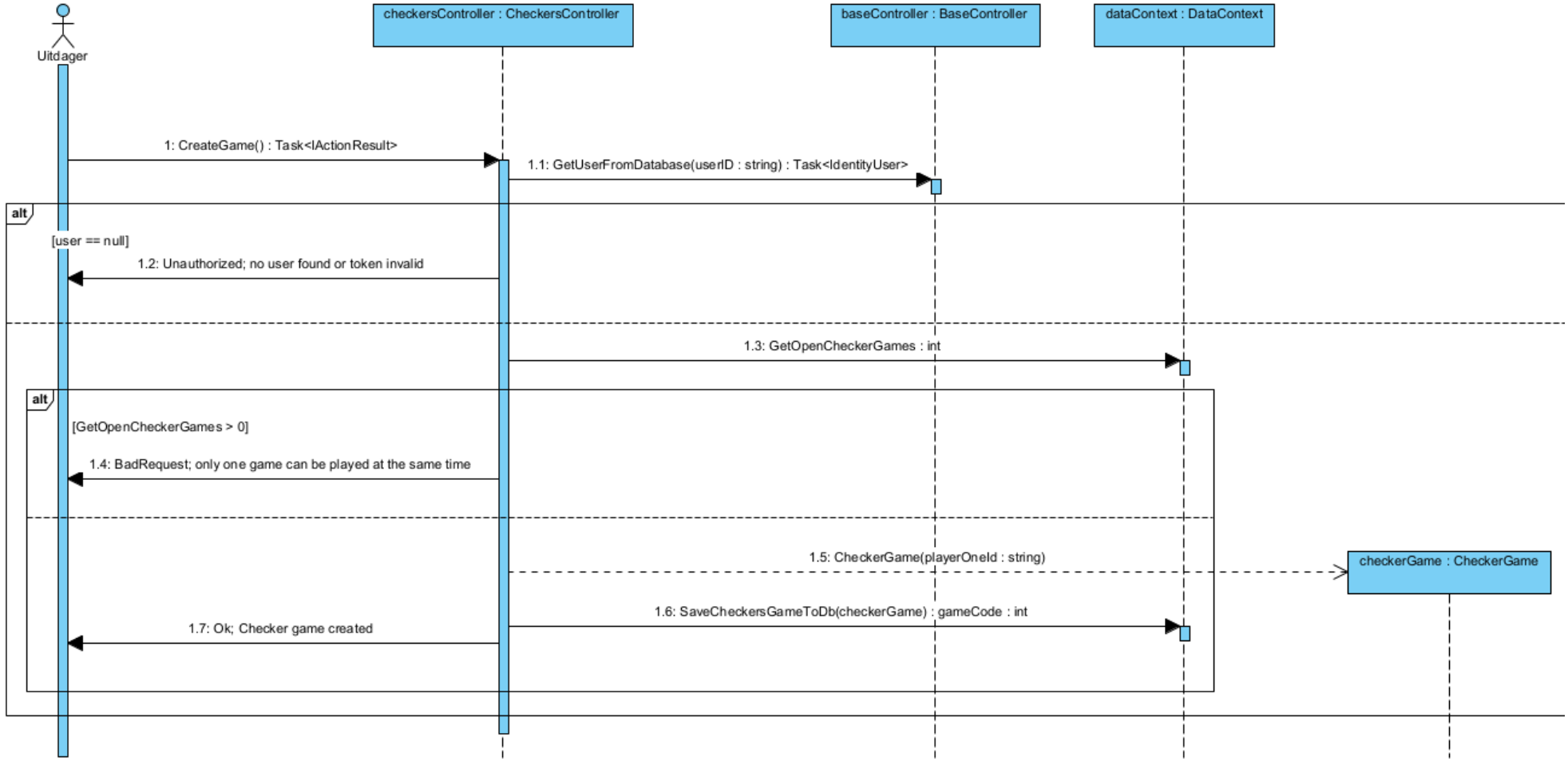
In het onderstaande diagram zijn de benodigde klassen voor de gamelogica uitgewerkt.



Figuur 7 - klassendiagram van de gamelogica

5.2 Aanmaken van een damspel

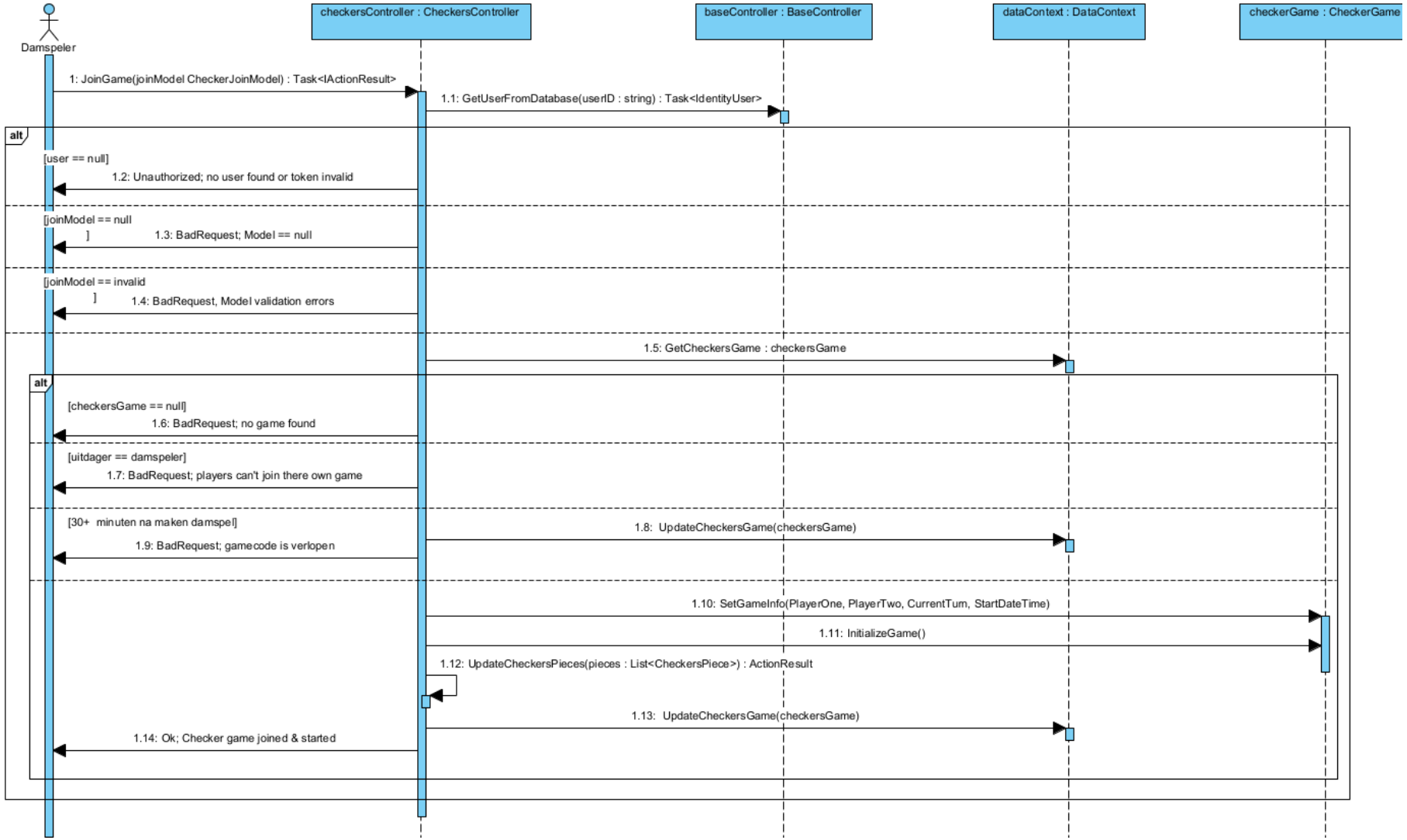
In het onderstaande diagram is de sequentie flow van het aanmaken van een damspel op de .NET webapi kant uitgewerkt.



Figuur 8 - sequentiediagram voor het aanmaken van een damspel

5.3 Deelnemen aan een damspel

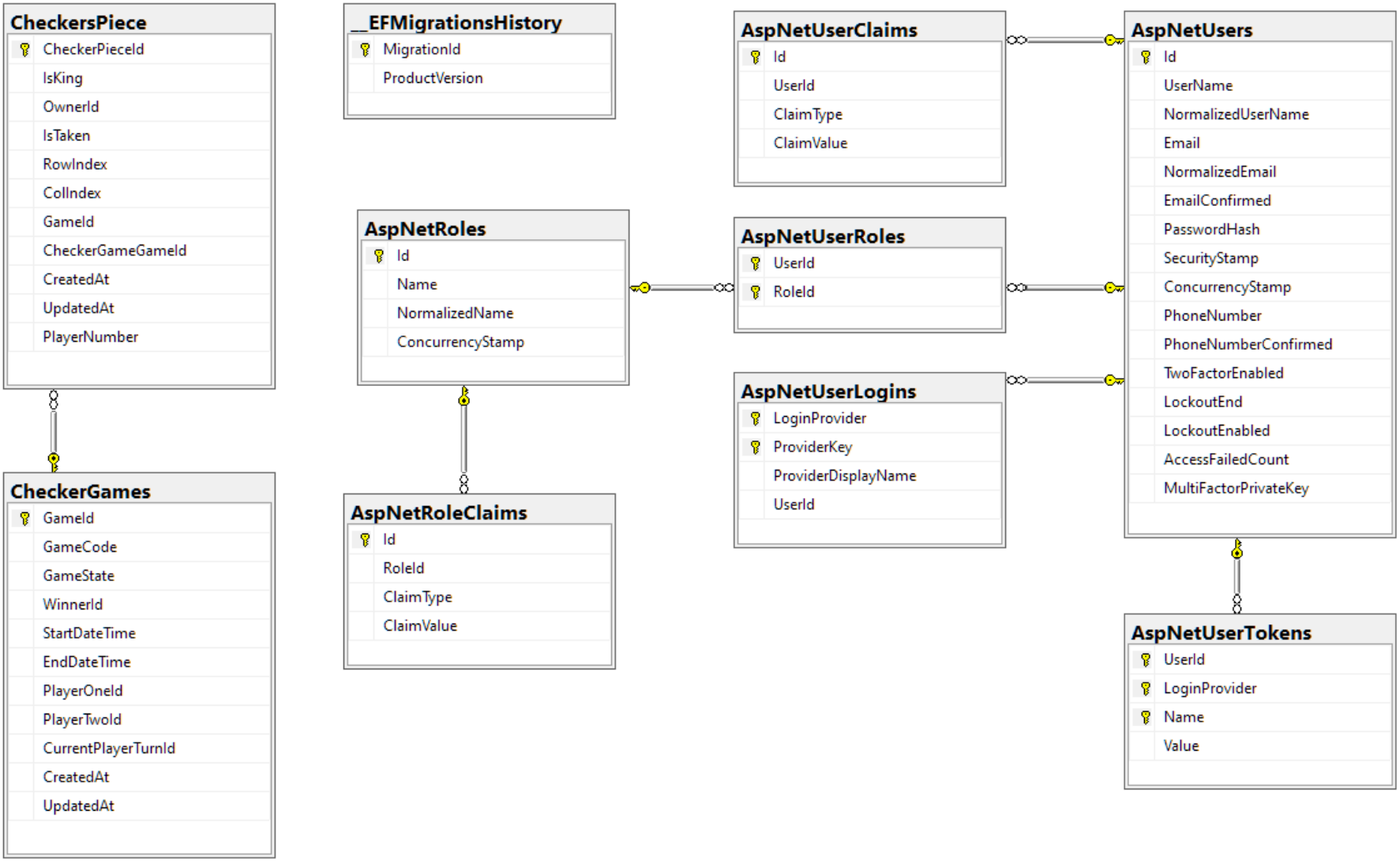
In het onderstaande diagram is de sequentieflow van het deelnemen aan een damspel uitgewerkt.



Figuur 9 - sequentiediagram voor het deelnemen aan een damspel

5.4 Database ontwerp

In het onderstaande diagram is het ontwerp van de database te zien.



Figuur 10 - ontwerp van de database