2-7-2023

Functioneel Ontwerp Infrastructuur

Rick Goos

Rick Goos | 512714
COMPETENTIE ONTWERPEN

Inhoudsopgave

1. Inleiding

Dit functioneel ontwerp dient ervoor om de functionele en niet-functionele vereisten van dit project in kaart te brengen en de bijbehorende implementatie van deze vereisten duidelijk te maken. Er wordt aangenomen dat de lezer al inzicht heeft in de overkoepelende opdracht, welke te lezen is in het document ‘requirements analyse.

Eerst worden de functionele en niet-functionele vereisten beschreven. Dit zijn vereisten waaraan de oplossing moet voldoen. Vervolgens worden deze vereisten met behulp van de methode MoSCoW geprioriteerd. Hierna wordt een visualisatie gegeven voor het gehele proces en de specifieke vereisten. Tot slot komen de consequenties van dit project voor de organisatie en haar medewerkers aan bod.

Doordat het project een technische achtergrond heeft, hebben de bijbehorende processen ook een technische insteek. Dit zorgt ervoor dat in dit ontwerp ook de techniek van het project wordt behandeld. Het is wel verder uitgewerkt in het technisch ontwerp.

2. Huidige Infrastructuur

In dit hoofdstuk gaan we het hebben over de huidige rollen, processen en wat de huidige werkwijze of werkwijze(s) zijn, door dit uit te gaan zoeken weten we wat het proces is en hoe ervoor kan worden gezorgd dat deze omgeving kan worden gedigitaliseerd.

2.1 Informatieverwerking

Momenteel maakt het bedrijf gebruik van een fysieke methode voor informatieverwerking, namelijk een magneetbord (zie **Fout! Verwijzingsbron niet gevonden.**). Dit magneetbord wordt al geruime tijd gebruikt als een methode om het plezier van de zwemleerlingen na de les vast te leggen. Het doel van het magneetbord is om zwemleerlingen in staat te stellen aan te geven hoe leuk ze de les vonden door een van de drie beschikbare smileys bij hun dierenavatar te plaatsen.

De verzamelde gegevens, komen voort uit het plaatsen van de smileys op het whiteboard, dit wordt dan ook wel de pleziermeting genoemd. Deze meting biedt zwemleraren waardevolle inzichten in de ervaringen van leerlingen en stelt hen in staat om te beoordelen hoe leuk de les was en of er sprake is van progressieve of conservatieve voortgang in de lessen, met betrekking tot het plezier. Door deze informatie te gebruiken, kunnen zwemleraren hun onderwijsaanpak aanpassen en afstemmen op de behoeften en voorkeuren van de leerlingen, wat uiteindelijk kan leiden tot een verhoogd plezier en een verbeterde leerervaring voor de zwemleerlingen.

2.2 Huidige applicatie

Het bedrijf maakt momenteel nog geen gebruik van een digitale applicatie, ondanks dat het bedrijf hier wel baart bij heeft is het er nog niet van gekomen. Het bedrijf maakt op het moment wel gebruik van een fysieke manier van het opslaan van gegevens, dit is momenteel het magneetbord (zie figuur 1), er is meer informatie te lezen in de **Fout! Verwijzingsbron niet gevonden.** over hoe dit werkt.

Figuur 1: huidige infrastructuur

3. Systeembeschrijving

Voor het bedrijf InnoSportLab is het belangrijk om een goede infrastructuur op te zetten, die de nodige continuïteit biedt voor de ontworpen applicatie. Deze nieuwe infrastructuur dient niet alleen goed te functioneren, maar ook betrouwbaar en veilig te zijn volgens de AVG-wetgeving.

Om dit te bereiken, is het noodzakelijk om een webserver te implementeren die gekoppeld is aan een database met de gegevens van de zwemklassen. Tussen deze twee moet er een firewall worden opgezet om zowel de nodige veiligheid te waarborgen als de veilige overdracht van gegevens te faciliteren. Bovendien is het van groot belang dat de database automatisch wordt geback-upt, zodat er geen kans bestaat dat gegevens verloren gaan.

Het is belangrijk dat medewerkers eenvoudig toegang hebben tot de applicatie en de webserver, zodat ze hun dagelijkse taken eenvoudig kunnen uitvoeren. Daarom moet er worden gezorgd dat het systeem toegankelijk is tijdens de openingstijden van het bedrijf, zodat medewerkers gemakkelijk gebruik kunnen maken van de applicatie.

Eveneens dient er aandacht te worden besteed aan het waarborgen van de integriteit en beschikbaarheid van de infrastructuur door passende maatregelen te treffen. Dit betekent dat de database regelmatig automatisch moet worden geback-upt. Bovendien dient er een functionerend logging systeem te worden opgezet om de infrastructuur te monitoren, zodat problemen of ongeoorloofde toegang eenvoudig kunnen worden opgespoord.

Al met al is heel belangrijk dat InnoSportLab een goede en vooral betrouwbare infrastructuur heeft die voldoet aan de vereisten van de applicatie en het AVG-onderzoek dat is opgezet.

4 Proces voor een nieuwe infrastructuur

Om een compleet beeld te krijgen van dit project en de bijbehorende processen zijn er visualisaties gemaakt. Eerst is een voorbeeld visualisatie van een CI/CD implementatie te zien en vervolgens wordt per requirement uit de `requirementsanalyse' het bijbehorende gedeelte uitvergroot.

4.1 MoSCoW prioritering

Met behulp van de MoSCoW methode is een prioritering gemaakt voor de eerdergenoemde vereisten. Per vereiste wordt een prioriteit gegeven met de bijbehorende toelichting.

Requirement nummer	**Prioritering**	**Toelichting**
SR01	M	Cloudopslag is doorgaans schaalbaar, waardoor InnoSportLab grote hoeveelheid van data kan opvangen zonder zich zorgen te hoeven maken over fysieke servers. Ook is het ontwikkelproces gemakkelijk voor de stichting, dit wordt verder beschreven op de volgende pagina.
SR03, SR04	S	Met een testomgeving kan het bugs of problemen identificeren en oplossen voordat het wordt vrijgegeven aan de wereld. Daarmee moet ook de GitLab repo verbonden zijn aan de AWS.
SR05, SR07	M	Voor InnoSportLab moet de webapplicatie in een EC2-instance zitten, dankzij de flexibiliteit van EC2-instances kan InnoSportLab gemakkelijk hun applicatie vanuit hier hosten.
SR02, SR06, SR09	M	Met een relationele database zorgt het ervoor dat de opslag en het ophalen van gegevens zo efficiënt mogelijk verloopt.
SR11, SR16	M	InnoSportLab moet hieraan voldoen zonder dat ze zich in de toekomst zorgen hoeven te maken als ze hier wel aan voldoen.
SR10, SR15	M	Hetzelfde gaat voor deze requirement, het is namelijk ook goed voor het vertrouwen van de gebruiker
SR08, SR12, SR17	M	Voor InnoSportLab moeten er regelmatig back-ups worden gemaakt van belangrijke gegevens zodat er geen gegevensverlies is.
SR18	C	Een monitoring tool is goed om bedreigingen snel te identificeren. Dit is echter geen prioriteit, omdat in testing fase dit nog niet gebruikt gaat worden door het ontwikkeling team. Pas wanneer de applicatie live is, is het van toepassing.

Tabel 1

Alle requirements met de prioriteit ‘Must’ moeten geïmplementeerd worden. Dit is belangrijk, omdat tijdens dit project minstens 3 applicaties worden ontwikkeld door de twee Saxion teams. Om dit te bereiken zijn deze vereisten nodig.

SR03 & SR17 is wenselijk om te implementeren, maar geen vereiste om het doel van dit project te bereiken.

4.2 Overige overwegingen

Tijdens dit project zijn verschillende overwegingen gemaakt over mogelijke andere oplossingen. In dit hoofdstuk wordt de belangrijkste overweging beschreven.

4.2.1 RDS aurora Database in de AWS-omgeving

Tijdens dit project wordt alleen een Database ec2 instance gebruikt. Geen RDS-aurora Database, hoewel dit eerst wel de bedoeling leek te zijn, wees hun netwerkbeheerder ons aan dat de kosten voor dit onnodig waren. De configuratie op deze server wordt zo beperkt mogelijk, zodat het beheer makkelijker wordt. Door deze maatregelen wordt de infrastructuur zo minimalistisch gemaakt en geeft dat een makkelijk beheerproces.

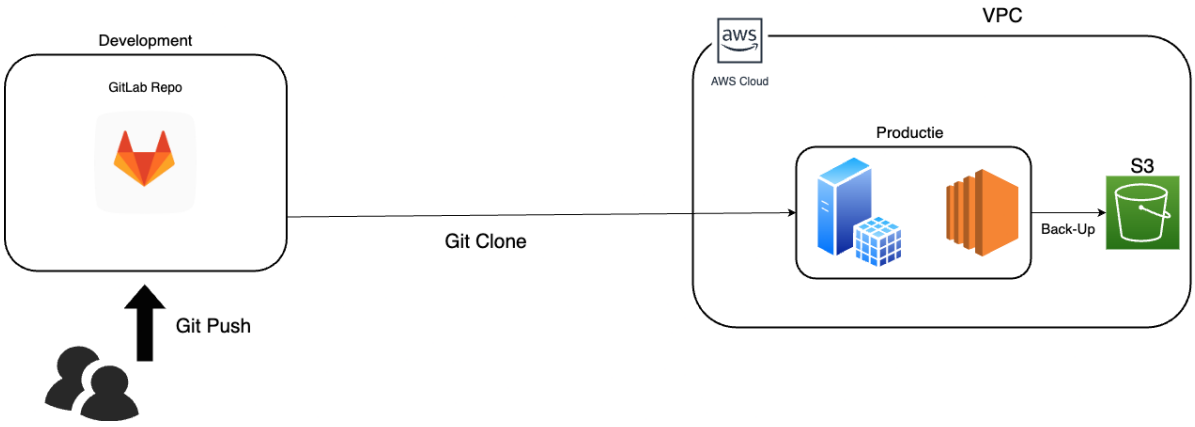
4.2.2 GitLab CI/CD implementatie naar AWS

Er werd onderzoek gedaan naar een goede manier om de GitLab Repo met AWS te verbinden. De eerste overweging om dit te doen was door een CloudFormation template. Met Cloudforation krijg je de bronnen automatisch in AWS zodat je minder tijd hoeft te besteden aan het beheren van die bronnen en meer tijd kunt besteden aan bouwen van applicaties. Maar dit is voor nu geen Must want voor onze groep is de GitLab repo volledig en wordt er niet verder mee gewerkt. Daarom is gekozen voor alleen slechts een GitLab clone naar de server. CloudFormation is goed als InnoSportLab in de toekomst verder blijft ontwikkelen aan de applicaties.

4.3 Onderbouwing per requirement

Per vereiste wordt het bijbehorende proces weergegeven.

4.3.1 SR01: De app heeft een cloudgebasseerde server nodig om aanvragen af te handelen en gegevens te beheren. (AWS)



Figuur 2

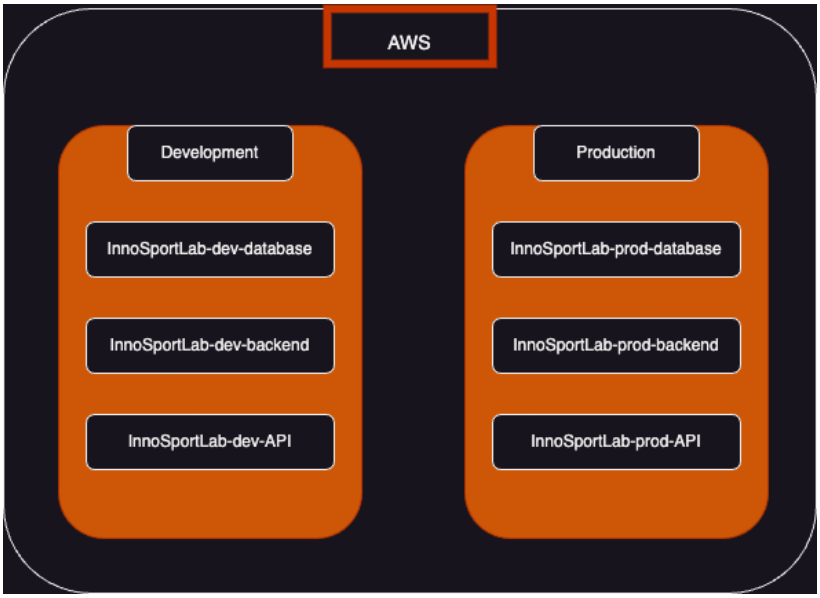
In figuur 2 is een voorbeeld hoe een ontwikkelproces eruitziet in de Cloud (AWS). Hieronder worden de stappen beschreven die ondernomen worden voor een goede CI/CD.

1. Allereerst duwt een ontwikkelaar de verbeterde code naar de GitHub-repository. Vervolgens wordt deze gecontroleerd door een andere persoon en samengevoegd met de master branch.
2. GitHub Actions detecteert een wijziging in de master branch en activeert de workflow.
3. Verbinding tussen GitHub en AWS.
4. De EC2 instances halen de code op en passen deze toe op de virtual servers.
5. Back-ups worden geplaatst in de S3-opslag.

4.3.2 SR03: Er moet een testomgeving zijn om de app te testen voordat deze openbaar wordt gemaakt & SR04: Er moet een verbinding zijn tussen GitLab en AWS

In de nieuwe infrastructuur worden twee afzonderlijke groepen gecreëerd voor ontwikkeling en productie, wat zal resulteren in een geoptimaliseerd overzicht van de omgeving en een duidelijke scheiding tussen de twee omgevingen.
De development omgeving is de gitlab repo van de software engineers, wanneer zij een code hebben gemaakt en die code is getest. Normaal gesproken testen de software engineers zelf de door hun gemaakte code, vaak met Unit tests of automatische testen. Kan dit gecloned worden naar productie in AWS.

Door het gebruik van de commando hieronder in de EC2 server kan de code opgehaald worden in AWS, en staat er een verbinding tussen GitLab en AWS.
commando git clone (https link uit gitlab).

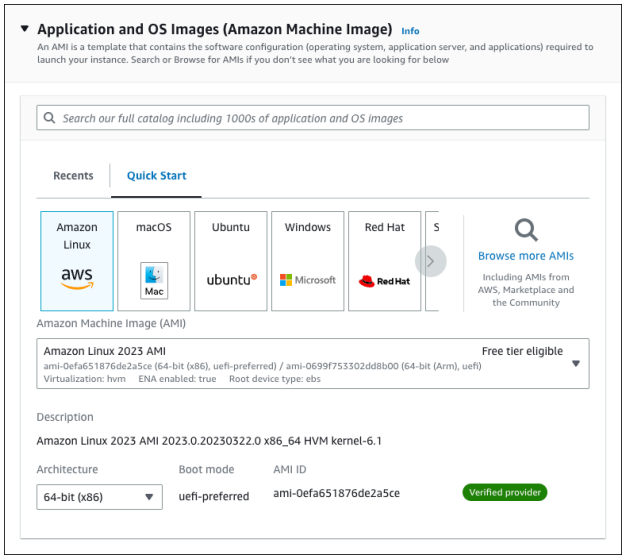


Figuur 3

4.3.3 SR05: De applicatie moet komen te draaien in AWS EC2 instances & SR07: Voor de Webserver moet er een Linux EC2 instance zijn met Apache.

Een Amazon EC2 instance in de volledige vorm is Amazon Elastic Compute Cloud. Het is een van de meest gebruikte en meest essentiële services op Amazon, dus is het gebruik hiervan het meest voor de hand liggende. EC2 is simpel gezegd een machine met een besturingssysteem en hardwarecomponenten naar keuze. Maar het verschil is dat het volledig gevirtualiseerd is. Er kunnen dus meerdere virtuele computers op één fysieke hardware worden uitgevoerd.
Amazon EC2-instances elimineren de investering vooraf voor hardware, hiermee kunnen applicaties dus sneller ontwikkeld worden. En op een EC2 instance betaal je alleen voor de service die ja daadwerkelijk gebruikt er zitten dus geen extra kosten vooraf aan vast. (Thakur & Thakur, 2023)

In de figuur hieronder zie je hoe dat er uit ziet:



Figuur 4

4.3.4 SR02: De app heeft een database nodig om informatie op te slaan, zoals gebruikersprofielen en statistieken.
SR06: Voor de database moet er een DB instance zijn met een Linux MySQL server &
SR09: De database in de infrastructuur moet een relationeel databasebeheersysteem zijn

Na overleg met de software-engineers in ons groepje is gekozen voor een relationeel databasebeheersysteem. Het geeft een gestructureerde manier om de gegevens op te slaan, te organiseren en op te halen, want het maakt namelijk gebruik van tabellen, rijen en kolommen.
De DB instance die wij gaan gebruiken is met MySQL en dat is een relationeel databasesysteem waarmee de requirement voldoet aan de oplossing.

Binnen AWS wordt er via EC2 een DB instance aangemaakt met daarop MySQL geïnstalleerd. Dit gebeurd dus op dezelfde manier als de webserver.
Om MySQL te installeren gaan deze commands gebruikt worden:
sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022
wget http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm
sudo yum localinstall -y mysql57-community-release-el7-8.noarch.rpm
sudo yum install -y mysql-community-server

Voor de data die nodig is, is dit het meest kosteneffectief want als je het laat runnen door Aurora betaal je €40/€50,- extra kosten vooraf.

4.3.5 SR10: De infrastructuur moet voldoen aan de ISO/IEC 27001:2017 norm &
SR15: De applicatie moet voldoen aan de eisen en verplichtingen van de AVG-Wet

Om ervoor te zorgen dat InnoSportLab zich houdt aan de ISO/IEC Norm en algemene verordening gegevensbescherming (AVG) is AWS een uitstekende keuze voor het beheren van de applicatie. Het naleven van deze norm zorgt ervoor dat er passende maatregelen genomen worden om de bescherming, beveiliging en naleving van hun gegevens te waarborgen. Want InnoSportLab werkt bovendien wel met gegevens van kinderen en

wanneer het niet voldoet aan deze norm en wet kan het worden geconfronteerd met verschillende consequenties.
Dit kan zijn:

- Boetes en sancties
- Juridische aansprakelijkheid
- Reputatieschade

In het bestand 'AVG-onderzoek' geschreven door Nick is meer over de AVG-wetgeving te lezen. (*GDPR - Amazon Web Services (AWS)*, z.d.)

4.3.6 SR08: Er moeten automatische back-ups gemaakt worden die de data in de S3 wegschrijft &
SR11: Om de nieuwste versie van de software op te slaan moet er gebruik gemaakt worden van Amazon S3 bucket &
SR13: Er moet een encryptiemethode -/hashing methode aanwezig zijn &
SR16: De database moet regelmatig worden geback-upt om gegevensverlies te voorkomen

De AVG vereist dat persoonsgegevens, dus ook die van kinderen, moeten worden beschermd en dat maatregelen worden genomen om gegevensverlies te voorkomen. Daarom is gekozen samen met de beheerder van InnoSportLab om back-ups automatisch te laten gebeuren. Zodat deze kunnen worden hersteld in geval van storing, menselijke fouten, cyberaanvallen of andere noodsituaties. Een back-up plan omvat:

- Back-upfrequentie: Het bepalen van de juiste tijden voor het maken van back-ups. Tijdens dit project stellen we het in op 1 keer in de week een back-up voor InnoSportLab. Als ze dit in de toekomst vaker willen, dan kunnen ze dat zelf gemakkelijk aanpassen.
- RPO (Recovery Point Objective): Een waarde moet worden bepaald wat het maximale hoeveelheid gegevensverlies acceptabel is tijdens een herstelproces.
- Retentieperiode: Het bepalen van de tijd hoelang een back-up moet worden bewaard. In de S3 kun je gegevens bewaren voor een onbepaalde tijd.
- Dataherstel: Er moeten algemene procedures opgesteld worden binnen de Stichting voor herstellen van de data. Dit is buiten onze opdracht en in de toekomst moet de organisatie hier zelf afspraken over maken.
- Veiligheid en versleuteling: Een back-up moeten ook versleuteld zijn zodat geen mensen erbij kunnen. Amazon versleuteld altijd back-ups, zelfs als de S3 buckets niet zijn versleuteld. Zodat er geen misbruik kan worden gemaakt met gegevens.

In ons back-up plan is gekozen voor periodieke back-ups waarmee InnoSportLab gegevens kan bewaren voor de opgegeven duur, en ook voor onbepaalde tijd.
De keuze om de data naar de S3 te sturen is gemaakt omdat een Amazon S3 bucket een flexibele en simpele Cloud opslag biedt. Je kunt daar alle soorten en maten van objecten in kwijt. Denk daarbij aan internetapplicaties, foto's en back-ups.

Al met al biedt het regelmatig back-uppen van de data een gevoel van rust. Het beschermt de stichting van gegevensverlies en zorgt het voor naleving van wettelijke verplichtingen. (*Working with backups - Amazon Relational Database Service*, z.d.)

4.3.7 SR17: Een logging-methode om verdachte activiteiten te detecteren om inbreuk te voorkomen

Om de veiligheid van de AWS-infrastructuur beter te monitoren kan een logging methode een uitstekende oplossing zijn. Dit biedt InnoSportLab de mogelijkheid om snel te reageren op bedreigingen en verdachte activiteiten.
Een manier om dit te doen binnen AWS is het gebruik van OpenSearch. Binnen deze service ziet de admin een dashboard waarop hij alle activiteiten in real-time kan zien. Dit dashboard kun je aanpassen naar je eigen eisen en wensen. Je kunt bijvoorbeeld zien:

- Event tijdlijn
- Alerts samenvattingen
- Geografische map: als InnoSportLab wil uitbreiden naar meerdere locaties kan een logging visualisatie als dit erg goed helpen.
- Trendanalyse
- Risico score

Een uitgebreid dashboard inrichten voor InnoSportLab zou in de toekomst een mooie stap zijn, maar in het tijdvak van de opdracht wordt dit moeilijk om te realiseren voor de stichting. (*Centralized Logging on AWS | AWS Solutions*, z.d.)

4 Consequenties

Na de implementatie van de cloudoplossing AWS zijn er gevolgen voor de organisatie en de medewerkers. Per onderdeel wordt kort omschreven wat de gevolgen zijn van dit project.

4.1 Organisatie

AWS is gericht op het verbeteren van werkzaamheden van medewerkers. Problemen kunnen wel sneller opgespoord worden met behulp van AWS, wat betekent dat de klanten van InnoSportLab, ofwel de gebruikers van de softwareapplicaties, met minder problemen gebruik kunnen maken van de applicaties.

4.2 Medewerker

Voor de IT-medewerkers verandert het meest. Op dit moment maken zij geen gebruik van een monitoringoplossing zoals AWS, maar halen de informatie handmatig op door middel van een whiteboard wat gebruikt wordt tijdens de zwemles. Wanneer dit vervangen wordt met een dashboard, wordt het proces voor het oplossen van problemen efficiënter, doordat problemen sneller te zien zijn in een dashboard.

5 Diagrammen

5.1 Entity Relationship Diagram

Dit is een diagram van de tabellen in de database. Dit is gemaakt door Emanuel, en wordt meegenomen in dit functioneel ontwerp.

De AspNet tabellen worden automatisch in de database gezet door het framework omdat ze bij het ingebouwde Identity systeem horen. Dit systeem maakt accounts, authenticatie en autorisatie voor ons makkelijker. AspNetUserRoles is een koppeltabel tussen de accounts en rollen. Hiermee kan een account meer dan 1 rol hebben.

__EFMigrationsHistory houd bij welke migraties er op de database zijn toegepast. Migraties komen van het Entity Framework Core systeem dat klassen in de backend omzet naar een database structuur. Elke keer als er iets in de data klassen van de code veranderd, maken we een nieuwe migratie die deze aanpassing in de database verwerkt.

AdminLogs worden gemaakt als een admin een belangrijke actie doet. Zoals een account een rol geven, of een accounts wachtwoord aanpassen. Zo kan er ingezien worden wie wat wanneer heeft gedaan bij fouten en misbruik.

Het Avatar systeem geeft elk kind een avatar die hij/zij kan personaliseren met accessoires en body parts (ogen, neus, mond, etc.). Accessoires kunnen in de winkel met punten gekocht worden. Deze punten worden aan de kinderen gegeven tijdens beoordelingen van de leraren. Helaas was er niet genoeg tijd om het systeem af te maken. Maar de benodigde tabellen staan al wel in de database.

Keuze onderbouwing

De StudentImages staan apart van de Students (kinderen die zwemles volgen) omdat niet elk kind een uniek plaatje heeft. Dit zou te veel plek kosten en te veel werk zijn. In plaats hiervan werken we met dierenplaatjes. Er wordt wel gezorgd dat kinderen in dezelfde zwemklas allemaal een ander plaatje krijgen.

5.2 Klassendiagram

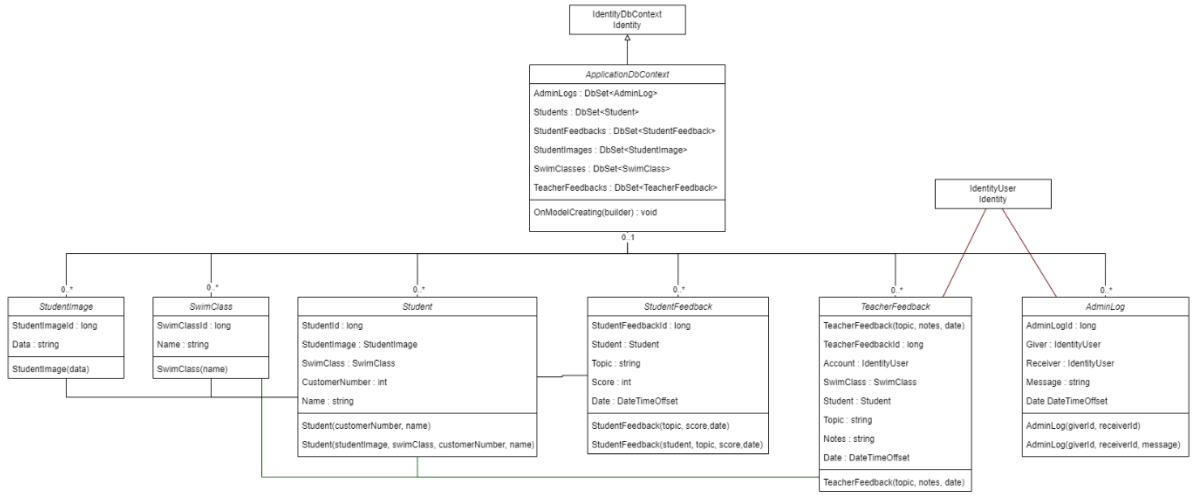
Deze diagrammen laten zien welke klassen er in de codebase zijn en welke data en functionaliteiten ze hebben. Ook laten ze zien welke relatie klassen tussen elkaar hebben.

Als een klas een relatie heeft met een klas van een ander diagram, wordt deze simpel getoond in het formaat:

Klas naam Diagram naam

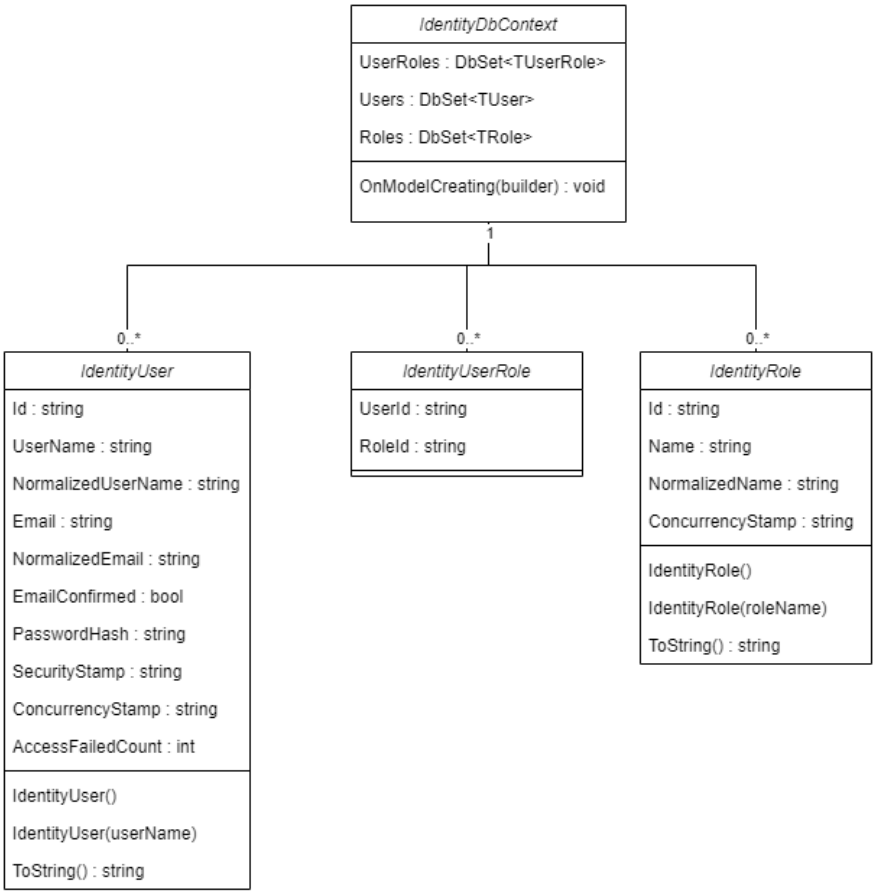
Klassen, velden en functionaliteiten die niet meer worden gebruikt zijn weg gelaten. Bijvoorbeeld het zelf gemaakte account systeem die is vervangen en het avatar, accessoires, body parts en shop systeem die we hebben laten vallen door tijdsdruk.

5.2.1 Data



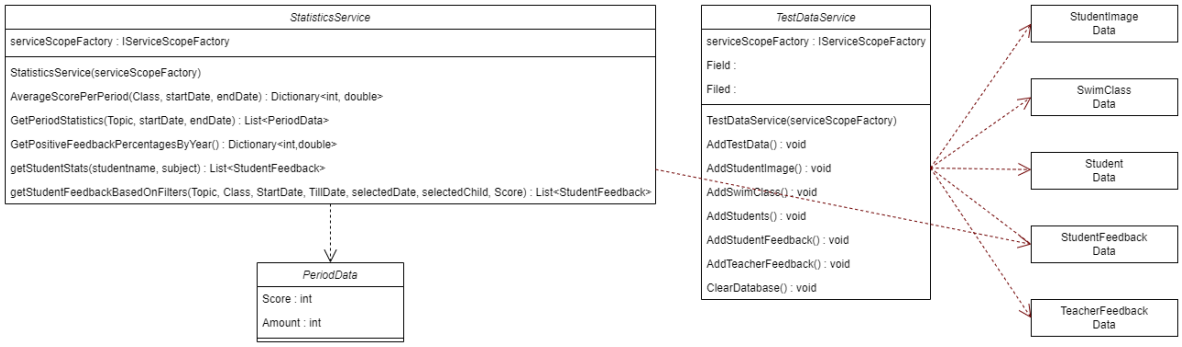
Dit zijn de klassen die de data van de database representeren. We gebruiken Entity Framework Core om code first de database met deze klassen aan te maken. Als het programma draait kunnen we dan via ApplicationDbContext zonder SQL met de database communiceren. ApplicationDbContext erft van IdentityDbContext. Identity is het ingebouwde authenticatie systeem. In de andere diagrammen wordt hier meer over verteld.

5.2.2 Identity



Deze klassen zijn te vergelijken met die van het Data diagram. Met als verschil dat ze in IdentityDbContext staan. Deze klassen hebben wij zelf niet gemaakt. Ze horen bij het Identity systeem.

5.2.3 Services

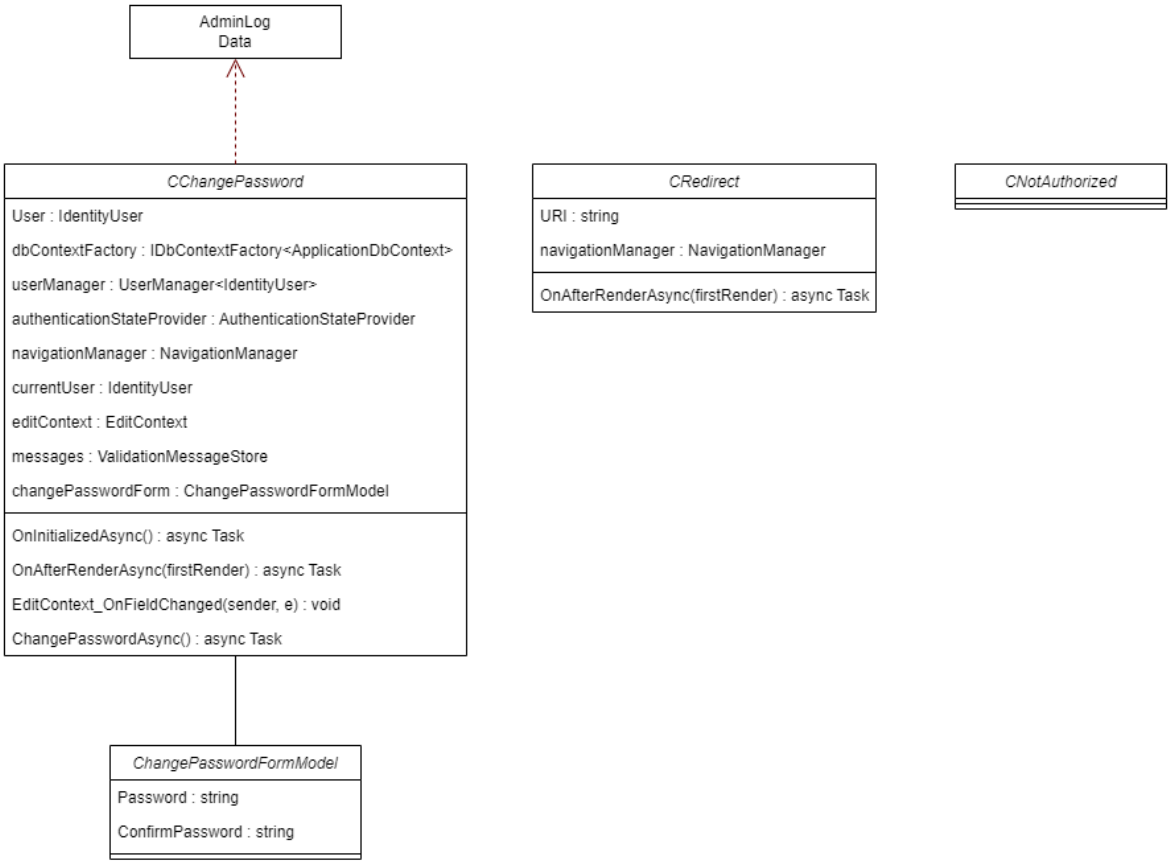


Services zijn het globale logic gedeelte van ons systeem. Het zijn singletons die op elke pagina gebruikt kunnen worden. StatisticsService gebruikt de StudentFeedback uit de database om statestieken te vormen. Deze kunnen dan in de frontend gevisualiseert worden. Met TestDataService kunnen we een nieuwe database makkelijk vullen met test data. Het is niet bedoeld voor productie

Keuze onderbouwing

Het grootste deel van de code logic zit in de pagina's zelf omdat het specifiek voor de pagina is. Ook zijn een componenten met frontend en logic als er iets op meerdere pagina's moet staan. De logic in StatisticsService is hier een uitzondering. Het heeft geen frontend nodig en we hebben meerdere pagina's die statestieken nodig hebben. Daarom is er hier gekozen voor een aparte service als tussenweg. Het is generiek geschreven zodat elke pagina er iets aan heeft.

5.2.4 Components



Components kunnen in een pagina worden gezet om een functionaliteit toe te voegen. Het zijn een soort services met een frontend gedeelte en een nieuwe instantie voor elke pagina en sessie. CChangePassword heeft een wachtwoord en bevestig wachtwoord veld om een user een nieuw wachtwoord te geven. CRedirect en CNotAuthorized staan in elke pagina met authorization. Als de gebruiker niet ingelogd is, word hij/zij met CRedirect terug gestuurd naar de login pagina. Als de gebruiker niet de juiste rol heeft, word alleen CNotAuthorized laten zien, waarop uitleg staat wat er mis is en hoe dit op te lossen is.

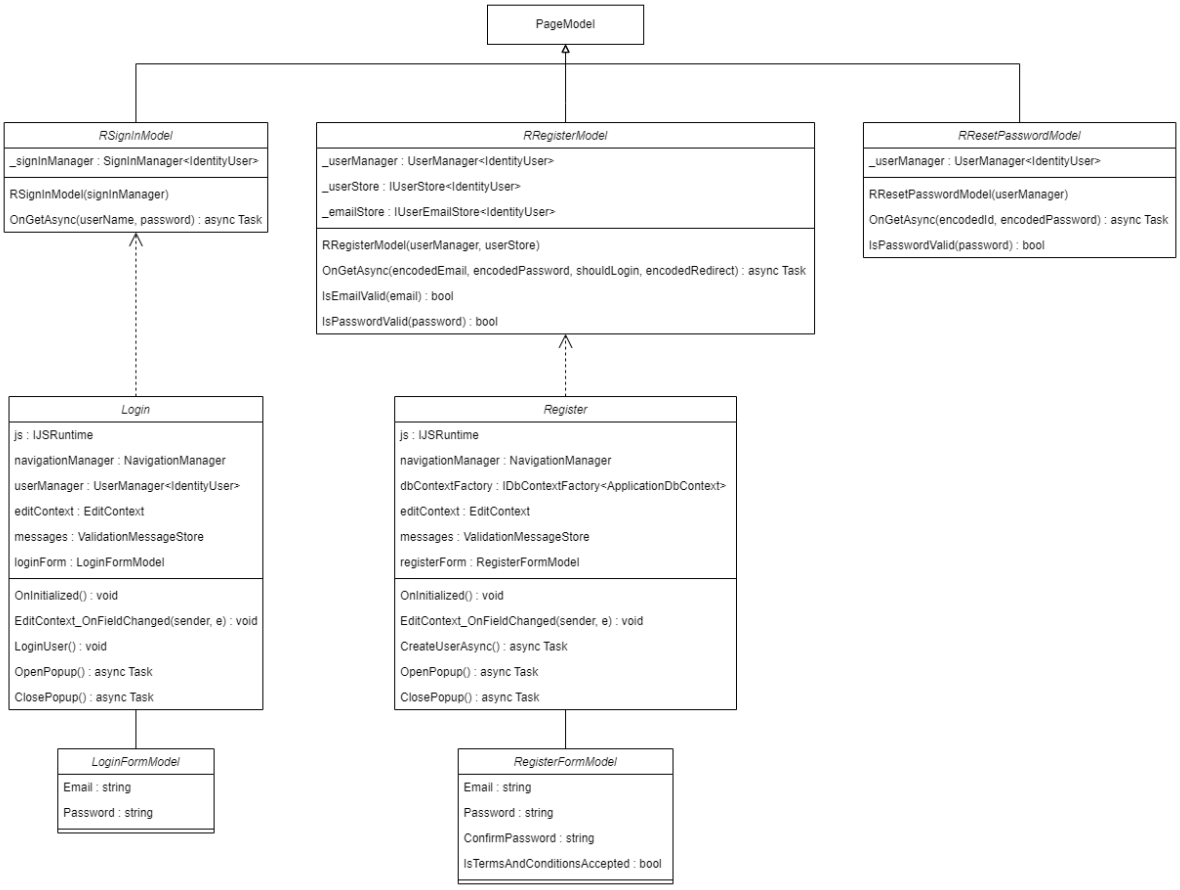
Keuze onderbouwing

CChangePassword geeft admins de mogelijkheid om een accounts wachtwoord aan te passen. Bij veel systemen doet de gebruiker van het account dit zelf. Echter is hiervoor een email bevestiging nodig om dit veilig te kunnen doen. De klant heeft duidelijk gemaakt dat hij kosten wil minimaliseren en voor de bevestigingsemail zou een emailservice nodig zijn. Dit zou extra kosten opleveren.

Een nadeel hiervan is dat het veel werk voor admins kan worden als de applicatie door veel docenten word gebruikt. Maar de klant is het met ons eens dat het voor de komende tijd goed

beheersbaar zal blijven vanwege het lage aantal docenten en hoe weinig het voorkomt dat een wachtwoord aanpassing nodig is.

5.2.5 Authentication



Dit zijn de pagina's voor het inloggen, registreren en aanpassen van een wachtwoord. De klassen met een "R" voor de naam gebruiken het oude framework pagina systeem. Dit is omdat het Identity systeem eigenlijk niet compatibel is met het nieuwe systeem en daarom sommige functionaliteiten op een oude pagina moeten gebeuren. Als voorbeeld: De Login klas heeft de frontend en zo veel mogelijk functionaliteit van de login pagina. Hier worden de gegevens in gevuld en gevalideerd. Maar om echt in te loggen moet er aan het einde even omgeleid worden naar RSignInModel.

De pagina's van het oude systeem erven van PageModel.

LoginFormModel en RegisterFormModel houden de data die door de gebruiker is ingevuld. Met annotaties worden deze gegevens gevalideerd.

Het rollen systeem werk op de volgende wijze:

Een gast kan zonder restrictie een account aanmaken. Alleen heeft een nieuw account nog geen rollen. Een account zonder rol kan net zo weinig als een gast.

Een admin kan dan het nieuwe account een van 2 rollen geven:

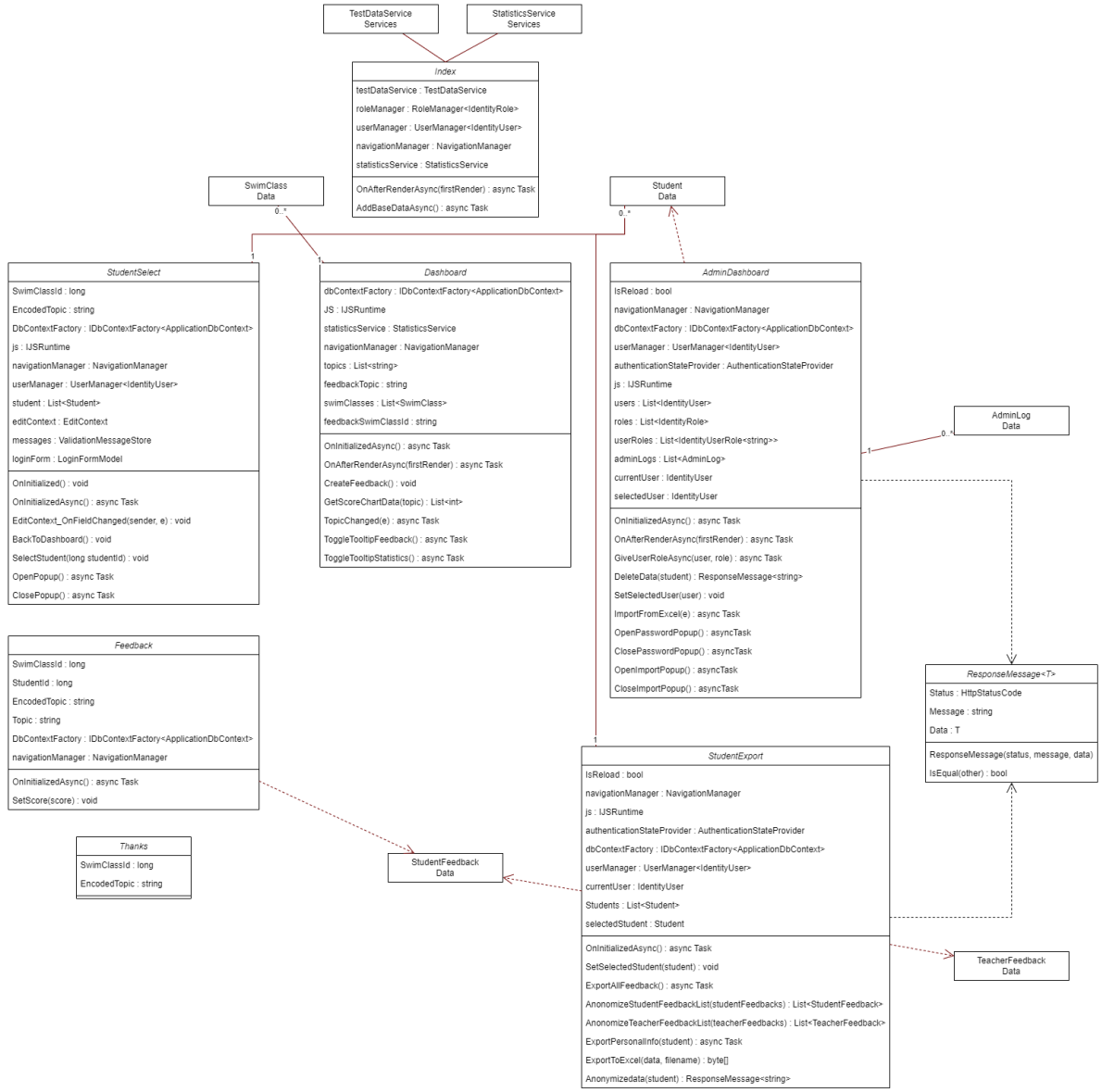
- Teacher: Kan statestieken inzien en feedback aan kinderen vragen.
- Admin: Kan alles wat teacher kan en kan bij het admin dashboard. In het admin dashboard worden de admin logs getoond, kan een accounts rollen en wachtwoord aangepast worden, kunnen kinderen verwijderd worden en kan kinder en feedback data geimporteet en geexporteert worden.

Verificatie van een nieuw account kan daarom worden gezien als het geven van een rol door een admin.

Keuze onderbouwing

Er is voor deze manier van account verificatie gekozen zodat niet zomaar iedereen zich als een leraar kan laten voordoen. De klant heeft zelf besloten dat het veiliger is als alleen een admin dit kan doen. Ook is er net zoals met het aanpassen van een wachtwoord, met dit systeem geen email verificatie en daarmee geen email service nodig.

5.2.6 Pages



Hier staan alle pagina's die niet tot de authenticatie horen. Dit is kern van de frontend en backend van het programma. Veel van de pagina's hebben een OnInitialized en/of OnAfterRender functie. Deze worden automatisch door het framework aangeroepen op bepaalde stadia van het bezoeken van een pagina.

Keuze onderbouwing

Als je naar de site gaat, kom je als eerste langs Index. Deze heeft geen frontend maar wordt gebruikt om wat dingen te regelen en de gebruiker door te sturen naar de juiste begin pagina voor hem/haar. Dit doen we hier omdat het zekerheid bied dat het framework alles heeft geinitialiseert en we dus alles zonder problemen kunnen doen/gebruiken.

ResponseMessage word op sommige plekken gebruikt om na het roepen van een functie te zien of er iets fout is gegaan en wat er fout is gegaan. De T is een Template die kan worden vervangen door een datatype zodat er ook data terug gegeven kan worden. Het word niet veel gebruikt omdat het niet nodig is met hoe het systeem werkt. Het is gekomen uit een misverstand van Jochem.

Bibliografie

InnoSportLab de Tongel Reep. (n.d.). *About - InnoSportLab*. Retrieved juni 9, 2023, from ISLT.nl: https://islt.nl/about/

PSV - Dutch Dolphin Swimming Club. (sd). *Over Ons*. Opgeroepen op juni 9, 2023, van Dutch Dolphins wimmingclub: https://www.dutchdolphinswimmingclub.com/over-ons

Centralized Logging on AWS | AWS Solutions. (z.d.). Amazon Web Services, Inc. https://aws.amazon.com/solutions/implementations/centralized-logging/

Working with backups - Amazon Relational Database Service. (z.d.). https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithAutomatedBackups.html

GDPR - Amazon Web Services (AWS). (z.d.). Amazon Web Services, Inc. https://aws.amazon.com/compliance/gdpr-center/

Thakur, A., & Thakur, A. (2023). What is AWS EC2 (Elastic Compute Cloud)? *Intellipaat Blog*. https://intellipaat.com/blog/what-is-amazon-ec2-in-aws/?US