Technisch Ontwerp

Scouting System

|  |  |
| --- | --- |
| Versie | V.2 |
| Datum | 21-02-23 |
| Naam | Robert Breno |
| Studentnummer | 160336 |

## Inleiding

Dit technisch ontwerp is opgesteld voor Scouting LIMS System een project waar ik binnen de bedrijf STARLIMS stage heb gelopen. LIMS is een Software waar je Specifiek applicaties kan bouwen van Laboratoriums.

Binnen het system van LIMS heb ik een project gemaakt voor een Scouting System waar de doel is om nieuwe jongeren talent kunnen ontdekt worden.

Deze opdracht heb ik van Starlims gekregen om hun system te leren kennen in de praktijk daarnast gebruik ik hem voor mijn portfolio.

1. Planning en Voorbereiding:

- Definieer het doel en de doelgroep van de app.

* een platform waarmee voetbalscouts talentvolle spelers kunnen ontdekken, met profielen, Post’s en scoutingsrapporten.

*- Maak een projectplan met deadlines.*

2. Ontwerp:

- Ontwerp de lay-out van de profielpagina's voor scouts en spelers, inclusief de benodigde informatie zoals naam, leeftijd, positie, statistieken, etc.

- Maak mock-ups en prototypes om het ontwerp te valideren voordat u begint met de ontwikkeling.

3. Ontwikkeling

- Bouw de registratie- en inlogfunctionaliteit voor scouts en spelers.

- Implementeer de zoekfunctionaliteit op basis van verschillende criteria.

- Maak een database waarin profielinformatie van spelers wordt opgeslagen, inclusief links naar hun social media-accounts en video's van wedstrijden.

- Ontwikkel de mogelijkheid voor scouts om rapporten over spelers te maken en scoutingactiviteiten bij te houden.

- Implementeer de functie voor spelers om video's van hun wedstrijden te uploaden naar hun profiel.

4. Testen:

- Voer uitgebreide tests uit om ervoor te zorgen dat alle functies correct werken en dat de app bugvrij is.

- Test de app op verschillende apparaten en besturingssystemen om compatibiliteit te garanderen.

- Verzamel feedback van testgebruikers en voer indien nodig aanpassingen door.

1. Registratie en profielen voor scouts en spelers:

Ik gebruik het login systeem van LIMS. Dit systeem is al gebouwd door het Amerikaanse kantoor. Mijn taak is om gebruikers aan te maken en rollen toe te kennen, zoals 'speler' of 'scout'. Er is ook een hoofdbeheerder.

Gebruikers kunnen hun eigen account aanmaken. De gegevens worden verzonden naar de Starlims server, die vervolgens de inloggegevens terugstuurt via e-mail. De gebruiker ontvangt zijn gebruikersnaam en een willekeurig wachtwoord.

Bij het inloggen wordt de gebruiker gevraagd om zijn wachtwoord te wijzigen. Alleen de gebruiker heeft toegang tot dit wachtwoord.

Alleen de beheerder kan het wachtwoord wijzigen en dit wordt opnieuw naar de gebruiker gemaild.

Hetzelfde geldt voor de scouts.

2. functionaliteit:

Ik heb functionaliteit toegevoegd aan het systeem, zodat gebruikers Posts kunnen maken, een profiel kunnen aanmaken, profielen kunnen bewerken, gegevens kunnen bekijken en spelers kunnen opzoeken.

Voor het maken van berichten kunnen gebruikers berichten maken en op basis van een URL-veld kunnen scouters de tekst selecteren en plakken in het tabblad, en vervolgens de link volgen die verwijst naar een social media video of een foto. Ik overwoog ook om afbeeldingen op te slaan in de starlims database, maar ik vond dat niet zo handig omdat het systeem traag werd, dus koos ik ervoor om een URL te gebruiken.

**HOE MAAK IK EEN POST IN LIMS ?**

Om te beginnen verzamel je alle informatie die je in een tabel wilt plaatsen. Vervolgens maak je een database aan. Maak een koppeling in het lims systeem en leg een relatie met de primaire sleutel (PK) in je tabel, wat betekent dat je tabel een primaire sleutel heeft.

Als dat is gedaan, moet je alleen nog een HTML formulier hebben. Vul de waardes van elk veld in en verbind vervolgens met MySQL met invoegcode. Voer de verbinding uit en voeg de informatie meteen toe aan jouw database.

Op deze manier heb ik een Post gemaakt.

**HOE MAAK IK EEN PROFIEL AAN IN LIMS ?**

Hier heb ik precies dezelfde basisstructuur, net zoals het voorbeeld dat ik hierboven heb gegeven. Alleen bij de spelergegevens heb ik unieke informatie, en vervolgens leg ik een relatie met de speler-ID omdat elke speler extra informatie heeft.

Bij het toevoegen van een speler maak ik een functie aan die aangeeft welke velden er bestaan. Ik maak een verbinding met de database om de informatie op te slaan en roep deze vervolgens weer op bij het updateformulier. Elke gebruiker die inlogt en geen profiel heeft, krijgt een pop-up te zien met de panelmanagerfunctie. Deze functie controleert of je een profiel hebt. Als dat het geval is, ga je naar het bewerkingsformulier; zo niet, dan ga je naar het toevoegingsformulier.

**HOE KAN IK EEN PROFIEL OPZOEKEN IN LIMS ?**

Hier hebben we een zoekfunctionaliteit voor nodig.

EERST maak ik een formulier in HTML MET LIMS dat alleen visueel is, want aan de achterkant wordt niets meer gedaan, dan moet ik gaan programmeren in LIMS.

Ik selecteer eerst een datagrid, dat is een kolom waar LIMS database tabellen kan laten zien, en een kolomformulier omdat dit veel wordt gebruikt in laboratoria.

Ik maak een functie aan die een verbinding gaat maken met een SSL-script, dus een server script. Het script praat eerst met de server en voert dan de functie uit.

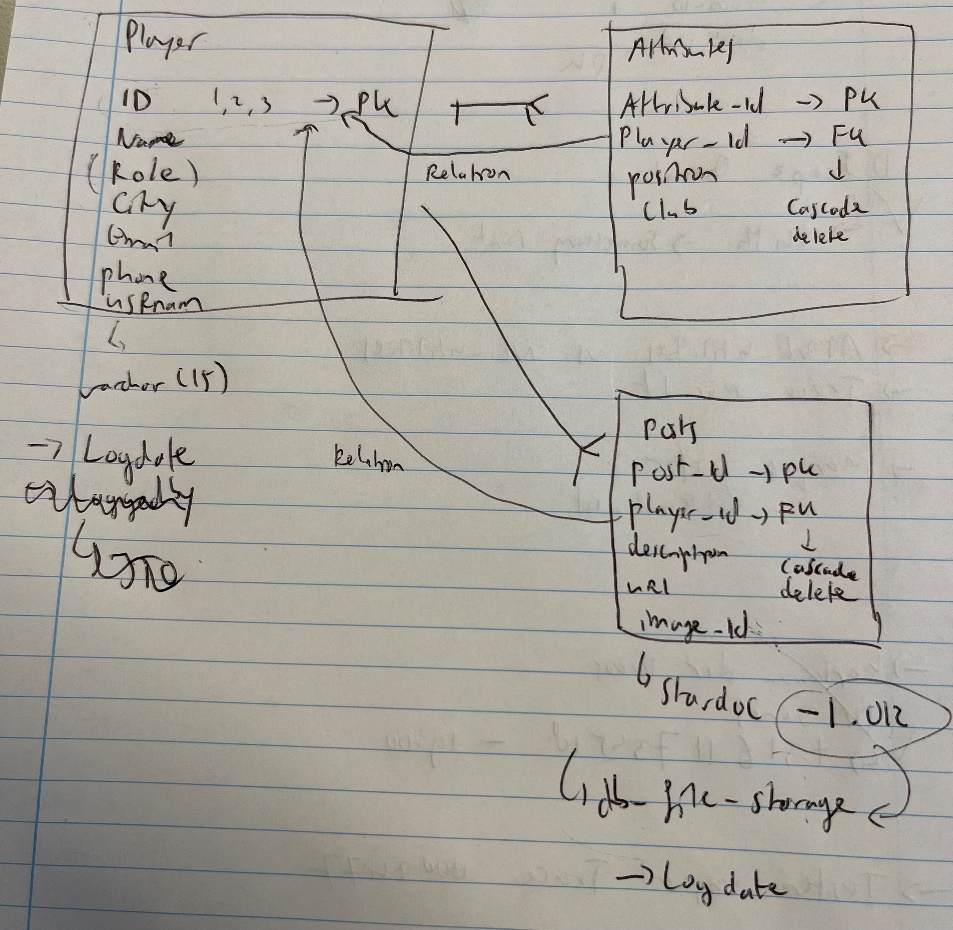
In het script maak ik een verbinding via roots bij de datagrid en selecteer ik welke database ik wil. Daarnaast selecteer ik de tabellen en voeg ik in de MySQL een JOIN in omdat ik natuurlijk alleen de gegevens van een bepaald veld wil, zoals Positie of Club. Anders zie ik alle informatie van alle spelers, wat verwarrend kan zijn, dus ik gebruik de JOIN als een filter.

Daarnaast is er alleen de mogelijkheid om te zoeken. Dan krijg je het resultaat terug dat je zoekt in het zoekveld.

3. database:

Ik heb samen met een ingenieur de structuur van de database op papier gezet. Hij legde me goed uit over de relaties en vroeg me om bij elke relatie een reden te hebben. Dus begon ik na te denken over wat ik kon doen.

Dit is mijn eerste concept van mijn ERD:



Dus, ik heb vastgesteld dat 3 tabellen nodig waren. De reden hiervoor is dat de eerste tabel, NL\_PLAYER, alleen de unieke informatie van de speler bewaarde. Deze tabel bevat ook een Primary Key, Player\_ID genaamd, en deze sleutel heeft een relatie met de andere tabellen.

Daarnaast is er de NL\_Attribute Tabel. Hierin worden de attributen van de speler bewaard, zoals CLUB en POSITIE. Omdat een speler op een bepaald moment voor een andere club kan gaan spelen, kan hij niet uniek zijn voor een club, en hetzelfde geldt voor de positie.

NL\_POST is gemaakt om de berichten van de speler op te slaan. Elke speler kan meerdere berichten plaatsen, en wanneer hij zijn profiel bekijkt, ziet hij alleen zijn eigen berichten terug.s ik ben uit gekomen dat 3 Tabellen nodig waren en de reden is

# ERD

Dit helpt ons en anderen om snel te begrijpen hoe onze database werkt.

De tabellen die Ik heb opgezet zijn: **NL\_Player, NL\_Attributers , NL\_Posts ,NL\_HEALTH** Elke tabel heeft zijn eigen unieke velden en relaties met andere tabellen.

De reden waarom ik "NL\_" aan het begin van de database gebruik, is omdat StarLIMS meerdere internationale projecten heeft. Op deze manier is het makkelijker te herkennen welke tabel specifiek voor Nederland is aangemaakt, enzovoort.

De hoofdtabel is verbonden met een CASCADE UPDATE DELETE in het systeem. Dit betekent dat als iets wordt verwijderd uit een tabel die gerelateerd is aan NL\_PLAYER op basis van de PLAYER\_ID, dan worden ook alle attributen en berichten van die speler verwijderd. Dit voorkomt problemen zoals fouten in de gegevens en opslagproblemen in de toekomst.

**NL\_PLAYER : Deze tabel bevat Alle informatie van de profile van de Players dus De Naam,Woonplaats,Clubs,Positie,Email,Phone,UsrNaam en DataLog.**

**Dat zijn de Uniek velden dus daarom staan ze in de Player Tabel.**

**NL\_Attribute: Deze tabel bevat onder de Attributen\_ID , Player\_ID FK dus hier wordt een relatie gemaakt met de EERSTE tabel en Positie en Clubs ze staan onder de attributs omdat er kan ook**

**zijn.de die speler bij meerdere clubs kan spelen dus op een club is is de Player een verderdiger en bij een andere een aanvaller.**

**Nl\_post : Deze tabel bevat onder de Posts\_ID PK dat is de Primary Key van de tabel daarnast zien we de Player\_ID FK weer terug dus die heeft ook een relatie met de Player Profile tabel.**

**Onderaan zien we de Descpription , Url en IMAGE ID.**

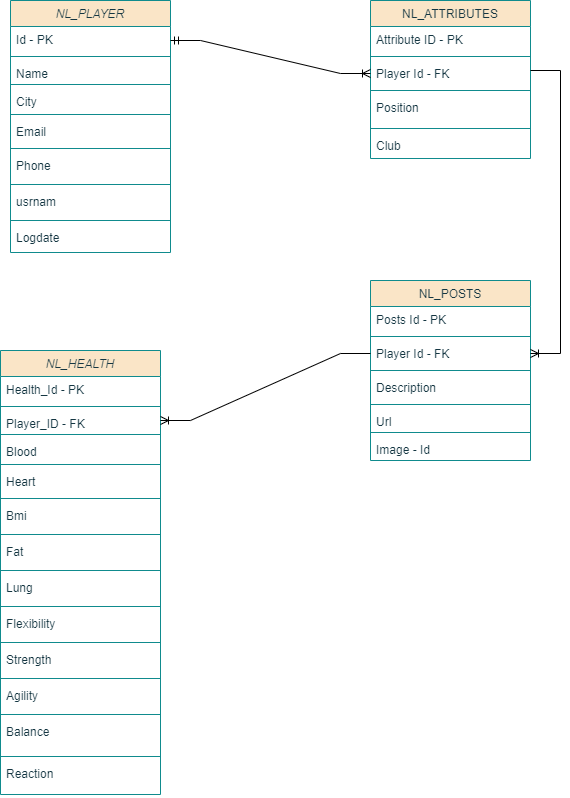
**Die 3 zorgen dat de gegevens van de posts van de players worden opgeslagen dus een descriptie daarnast een url veldje waar de players een url van een filmpje bijvoorbeeld kunnen toevoegen of een image.**

**NL\_HEALTH:** Deze tabel slaat alle informatie op die is verkregen uit de gezondheidstesten. Er bestaat ook een Foreign Key (FK) van PLAYER\_ID, wat betekent dat er een relatie bestaat met de nl\_player tabel. Hierdoor kan een speler worden geselecteerd in een dropdown menu op het formulier, waarna de ID van die speler automatisch wordt toegevoegd aan de nl\_health tabel. Later, op de pagina met gezondheidsinformatie, kan een speler worden geselecteerd en worden de bijbehorende testgegevens weergegeven in een andere tabel. Het uiteindelijke doel is dat een scout deze gegevens kan uitprinten in de vorm van een rapport.

A screenshot of a computer

Description automatically generated

Figuur 1 ERD Eerst Concept



#### 1. Server

*- Starlims heeft een eigen appaart serve dus ik kan alleen er bij met toesteming van de Engineers.*

#### 2. Databaseserver:

*- Voor Database StarLIMS heft een eigen DataServer en ze maken ook gebruik van de SQL.*

3. Webserver: *- De webserver is ingesteld op basis LIMS Software.*

#### 4. Ontwikkelplatform:

*- Ik heb dit project gebouwd in de LIMS PLATFORM en Lims progammertaal gebruikt en die is gemaakt met .NET en Javascript. Dus ik bouw niet met C# OF JAVA SCRIPT maar met LIMS er is een library vol met Functies die zijn gebouwd vanuit de Kantoor in USA en daar maak ik gebruik van op basis van de logica van LIMS.*

#### 5. Framework:

*- LIMS gebruik een HTML5 en JAVASCRIPT Versie en hele oude system in de achterkant.*

*- Maar wij hebben geen toegang tot de HTML FORM EN DE JAVASCRIPT CODE.*