

Gebroeders de Smetstraat 1, 9000 Gent België



Smart Fooseball

Hardware en software voor tafelvoetbalgoalregistratie

Siebe Van de Voorde, Jarno Van Osselaer, Ruben Van Poucke, Eli Van Stichelen

Professionele Batchelor Electronica-ICT /   
Fase 2

2022-2023

Mentor: Serge Fabre, Sabine Martens

Opdrachtgevers: Sven Sanders

Smart Fooseball

Siebe Van de Voorde, Jarno Van Osselaer, Ruben Van Poucke, Eli Van Stichelen

The goal of this project is to make a smart foosball table.

When frequent foosball players go to pubs, they should be able to keep track of the game and scores in a legit and safe manner. It’s also difficult to check whether people are telling the truth, solving this is one of the main purposes of this project.

In the first chapter you’ll read which languages and technologies we have used and why we have done so.

By building sensors into an existing table and having it communicate with a server, a table can be made smart. The server then communicates with a mobile-first website that the user visits. People can create an account on this website. After they have created an account, they can enter a code that is on the foosball table and register for a match. There will also be a plethora of different data such as total goals, games won and lost, and a lot more.INHOUDSOPGAVE

[CODEFRAGMENTENLIJST 5](#_Toc130472953)

[FIGURENLIJST 6](#_Toc130472954)

[TABELLENLIJST 7](#_Toc130472955)

[AFKORTINGENLIJST 8](#_Toc130472956)

[INLEIDING 9](#_Toc130472957)

[1 Mogelijke en gekozen hardware 10](#_Toc130472958)

[1.1 Spelbediening 10](#_Toc130472959)

[1.2 Goaldetectie 10](#_Toc130472960)

[1.3 Display met game-informatie 10](#_Toc130472961)

[1.4 Ledelementen voor verlichting en versiering 10](#_Toc130472962)

[1.5 Luidsprekers 10](#_Toc130472963)

[2 Mogelijke en gekozen Software 11](#_Toc130472964)

[2.1 Frontend 11](#_Toc130472965)

[2.2 Backend 11](#_Toc130472966)

[2.3 Communicatie tussen front- en backend 11](#_Toc130472967)

[2.4 Database 11](#_Toc130472968)

[2.5 Communicatie tussen backend en database 12](#_Toc130472969)

[3 Technische uitwerking hardware 13](#_Toc130472970)

[3.1 Spelbediening 13](#_Toc130472971)

[3.2 Goaldetectie 13](#_Toc130472972)

[3.4 Display met game-informatie 13](#_Toc130472973)

[3.5 Ledelementen voor verlichting en versiering 13](#_Toc130472974)

[3.6 Luidsprekers 13](#_Toc130472975)

[4 Technische uitwerking software 14](#_Toc130472976)

[4.1 Frontend 14](#_Toc130472977)

[4.2 Backend 14](#_Toc130472978)

[4.3 Communicatie tussen front- en backend 14](#_Toc130472979)

[4.4 Database 14](#_Toc130472980)

[4.5 Communicatie tussen backend en database 14](#_Toc130472981)

[5 Risicoanalyse 15](#_Toc130472982)

[6 Kostenraming 16](#_Toc130472983)

[Conclusie 17](#_Toc130472984)

[Handleiding 18](#_Toc130472985)

[Literatuurlijst 19](#_Toc130472986)

[Bijlagenoverzicht 20](#_Toc130472987)

[Bijlage 1: Kopieën datasheets 20](#_Toc130472988)

[Bijlage 2: Vergaderverslagen 21](#_Toc130472989)

[Bijlage 3: Logboek rapporteren 22](#_Toc130472990)

# CODEFRAGMENTENLIJST

# 

# FIGURENLIJST

# 

# TABELLENLIJST

# AFKORTINGENLIJST

|  |  |
| --- | --- |
| API | Application Programming Interface |
| URL | Uniform Resource Locator |
| RDMS | Relational Database Management Systems |
| GPL | General Public License |
|  |  |
|  |  |
|  |  |
|  |  |

# INLEIDING

SmartFooseball is een gewone – analoge – voetbaltafel die wordt voorzien van hardware waardoor deze kan communiceren met een webapp.

Er is dus hardware voorzien die zowel de communicatie behandelt met de webapp en hardware die de goals registreerd. Alsook is er een webapp ontwikkeld waarop spelers een account kunnen aanmaken. In de webapplicatie worden alle data van voorgaande spellen en spelers bijgehouden en overzichtelijk weergegeven.

De beginsituatie is dus een analoge voetbaltafel. Aan deze starttoestand worden diverse bijzondere functies en opties gekoppeld die het spelgevoel een andere dimensie moeten geven.

Het eindproduct kan aangesloten op een normale voetbaltafel zodat deze vervolgens kan worden gebruikt in bv. café’s, jeugdhuizen …. Dit alles moet verwezenlijkt worden voor 3 juni en mag niet meer dan €150 kosten.

Doorheen het project wordt er opzoekwerk gedaan en worden schema’s ontworpen. Zo wordt gezocht hoe de hardware werkt en hoe deze kan communiceren. Alsook wordt gekeken welke programeertalen mogenlijks gebruikt kunnen worden en wat de voor- en nadelen hiervan zijn. Er moet ook gekeken worden hoe de databank er zal uitzien en of er beter een rationale of een NoSQL databank wordt gebruikt.

In dit rapport worden de mogenlijke oplossingen onderzocht en wordt één van deze mogenlijk heden gekozen in de eerste twee hoodstukken. In een derde volgt de technische uitwerking van de hardware. Met het hoofdstuk dat erop volgt een uitwerking van de software. Er wordt in de laatste hoodstukken nog een risicoanalyse uitgevoerd en een kostenraming gemaakt.

# 1 Mogelijke en gekozen hardware

## 1.1 Spelbediening

## 1.2 Goaldetectie

## 1.3 Display met game-informatie

## 1.4 Ledelementen voor verlichting en versiering

## 1.5 Luidsprekers

# 2 Mogelijke en gekozen Software

## 2.1 Frontend

## 2.2 Backend

Voor de backend zijn er twee opties: .Net 7 (laatste versie die uit is in 2023) of Php 8 (eveneens de laatste versie, een framework of standaard). Er kan ook gewerkt worden met Ecmascript voor de backend maar aangezien deze taal vooral bedoelt is voor frontend en niet voor backend is dit niet aan te raden.

De keuze is dus tussen Php en .Net. Beide zijn goed voor het maken van web applicaties door middel van Application Programming Interfaces (API) en hebben een sterke en actieve communities. Er is dus veel documentatie en er zijn veel voorbeelden beschikbaar. Ze kunnen allebij grote hoeveelheden verkeer en data aan. Zelfs voor grote zakelijkse activiteiten.

Het grote verschil ligt echter bij de leercurve. Php is makkelijker om onder de knie te krijgen. Dit zorgt ervoor dat het populairder is bij developers en ons eigen team. Twee personen kunnen .Net, vier personen kunnen Php. Dus gaat de keuze uit naar php.

Dan rest enkel nog de vraag of we standaard Php gaan gebruiken met enkele library’s of een full fledged framework. Hoewel er altijd meer developers te vinden zijn voor een standaard taal dan een specifiek framework, gaan we wel voor een framework gaan aangezien er veel handige features en functies zijn in frameworks. Waarom het wiel heruitvinden als iemand het al gedaan heeft?

We kiezen hierbij voor Laravel. Het is het populairste framework voor php en het werkt goed met verschillende databases. Het is zeer goed voor het maken van API’s en het heeft een template engine genaamt Blade die makkelijk te leren is.

## 2.3 Communicatie tussen front- en backend

De communicatie tussen de front- en de backend zal verlopen via een API. De frontend stuurt requests naar de backend via een bepaalde Uniform Resource Locator (URL). En deze geeft dan een antwoord terug. We kiezen ervoor om deze antwoorden in Json te doen aangezien dit goed samenwerkt met Laravel en de meest standaard is.

## 2.4 Database

Voor de database is de eerste grote keuze die we ons moeten afvragen of we een relationele of een irrelationele database gaan gebruiken. Aangezien we wel wat relaties hebben tussen onze tabellen, gestructuureerde data hebben en we niet met big data werken. Klinkt een relationele database ons de beste optie.

Dan hebben we nog de keuze tussen welke Relational Database Management Systems (RDMS) we gebruiken. De drie populairste RDMS zijn Oracle, MySQL, Microsoft SQL Server. Aangezien we voor de backend Php gebruiken en Microsoft SQL Server vooral gemaakt is voor met .Net. Lijkt het ons beter om deze al uit te sluiten. Dan hebben we nog de keuze tussen Oracle en MySQL. Hoewel Oracle een grotere market share heeft dan MySQL, gaan we toch voor MySQL gaan. MySQl heeft een General Public License (GPL), is gratis en open-source. Oracle daarintegen vraagt een licensing fee.

## 2.5 Communicatie tussen backend en database

Doordat we gekozen hebben voor MySQL als RDMS gebeurt de communicatie via SQL-queries. Dit gebeurt echter achter de schermen. Door Laravel te gebruiken moeten we zelf geen queries schrijven en zal deze dat voor ons doen door gebruik te maken van ingebouwde functies in Laravel.

# 3 Technische uitwerking hardware

## 3.1 Spelbediening

## 3.2 Goaldetectie

## 3.4 Display met game-informatie

## 3.5 Ledelementen voor verlichting en versiering

## 3.6 Luidsprekers

# 4 Technische uitwerking software

## 4.1 Frontend

## 4.2 Backend

## 4.3 Communicatie tussen front- en backend

## 4.4 Database

## 4.5 Communicatie tussen backend en database

# 5 Risicoanalyse

# 6 Kostenraming

# Conclusie

# Handleiding

# Literatuurlijst

# Bijlagenoverzicht

## Bijlage 1: Kopieën datasheets

## Bijlage 2: Vergaderverslagen

Smart Fooseball

Vergaderverslag 1

Datum: 14/02/2023   
Locatie: Teams

Aanwezig: Serge Fabre, Jarno Van Osselaer, Ruben Van Poucke, Eli Van Stichelen, Siebe Van de Voorde

Afwezig: /

1 Notulen

2 Agendapunten

2.1. Agendapunt 1: uitleg project aan mentor

Alle spelers hebben een account, loggen zich in de app en selecteren de juiste kickertafel, en starten de match. Alle goals worden geregistreerd in de app en worden aan het account van de spelers gelinkt.

2.2. Agendapunt 2: kiezen componenten

-draadloos of kabel : eens extra info vragen aan collega’s electronica

2.3. Agendapunt 3: project management tools

-onderling bepalen welke tools we gebruiken.

3 Actieplan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| naam | omschrijving | deadline | opvolging | Voltooid |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Bijlage 3: Logboek rapporteren

|  |  |  |
| --- | --- | --- |
| Eli Van Stichelen | 2, 9 | Vertaling concreet + inleiding + titels |
| Ruben Van Poucke |  |  |
| Jarno Van Osselaer | 9 | Inleiding + overlezen |
| Siebe Van de Voorde | 1, 2, 9, 20 | Voorblad + concreet + herwerking inleiding + herwerking titels + agenda + hoofdstuk 2.2, 2.3, 2.3, 2.4, 2.5 geschreven |