Samenvatting

Dit document bevat het functioneel ontwerp, hierin wordt duidelijk hoe de functies van de C# applicatie en web app gerealiseerd gaan worden

Functioneel ontwerp

Gemaakt door groep 3

Inhoud

[1. Functioneel ontwerp van de webapp 2](#_Toc480380520)

[1.1 Hoe werkt de web app 2](#_Toc480380521)

[1.2 Login scherm 2](#_Toc480380522)

[1.3 Het aanmaken van teams 3](#_Toc480380523)

[1.4 Het verwijderen van teams 4](#_Toc480380524)

[1.5 Het maken van een wedstrijd schema 4](#_Toc480380525)

[1.6 Het updaten van de scores van een wedstrijd 5](#_Toc480380526)

[1.7 Het aanmaken van speler 5](#_Toc480380527)

[1.8 Het aanpassen van teams 6](#_Toc480380528)

[1.9 Het aanpassen van spelers 6](#_Toc480380529)

[2. Functioneel ontwerp van de C# applicatie 7](#_Toc480380530)

[1.1 Hoe werkt de C# applicatie 7](#_Toc480380531)

[1.2 De functies 7](#_Toc480380532)

[Verplicht 7](#_Toc480380533)

[Tijd over 7](#_Toc480380534)

[Komt er sowieso niet in 7](#_Toc480380535)

[1.3 Wedden op de uitslagen 8](#_Toc480380536)

[1.4 Wedstrijden importeren 8](#_Toc480380537)

[1.5 Per wedstrijd kunnen bieden 9](#_Toc480380538)

[1.6 Uitslagen importeren 10](#_Toc480380539)

[1.7 Naast punten krijgen ontvang je ook geld 10](#_Toc480380540)

[3. Wireframe 11](#_Toc480380541)

# 1. Functioneel ontwerp van de webapp

Hierin komt alle informatie conform de functies van de webapp

## 1.1 Hoe werkt de web app

De web app staat in samenhang met een database. Door de database connectie wordt het mogelijk gemaakt om data op te halen en er ook in te verwerken. Het data verwerken wordt gedaan via forms, deze dienen ingevuld te worden(door de gebruiker) en in de achtergrond worden deze dan weer geüpload in de database.

## 1.2 Login scherm

Dit wordt gedaan door een eerder genoemde form. Hier achter draait een PHP script die in connectie staat met een database. Het form dat gebruikt wordt voor het inloggen ziet er zo uit



Het moment dat een gebruiker op login klikt dan zal op de achtergrond via PHP script gekeken of de gebruiker bekend is in de database zo ja wordt hij geredirect naar de beheerpage en vanuit krijgt hij toegang tot de onderstaande functies

* Aanmaken team / Verwijderen (Done)
* Wedstrijd schema maken (Done)
* Per wedstrijd een score kunnen bijwerken
* Speler aanmaken + linken aan een team

Dingen die mogelijk worden toegevoegd aan de webapp zijn

* Team data aanpassen
* Speler data aanpassen / speler data verwijderen

1.3 Het aanmaken van teams

Dit wordt gedaan door een eerder genoemde form. Hier achter draait een PHP script die in connectie staat met een database. Het form dat gebruikt wordt voor het aanmaken van een team ziet er zo uit



Het moment dat de gebruiker op creëer team klikt zal deze in de database worden geüpload en deze is dan gelijk klaar om gebruikt te worden om spelers bijvoorbeeld aan een team te linken.

1.4 Het verwijderen van teams

Dit wordt gedaan door een foreach loop met hidden form mits je eroverheen hovert. Hier achter draait een PHP script die in connectie staat met een database. Het form dat gebruikt wordt voor het verwijder van een team ziet er zo uit



Het moment dat de gebruiker op delete klikt zal het team worden verwijderd uit de database

## 1.5 Het maken van een wedstrijd schema

Dit wordt gedaan door een eerder genoemde form. Hier achter draait een PHP script die in connectie staat met een database. Het form dat gebruikt wordt voor het aanmaken van een wedstrijd schema ziet er zo uit.



Op het moment dat hij geneer klikt zal het bestand worden geëxporteerd naar een png bestand, die zal dan direct worden gedownload.

## 1.6 Het updaten van de scores van een wedstrijd

Als je schema gecreëerd hebt dan kun je vervolgens een poule overzicht krijgen door op poule overzicht te klikken dan krijg je het volgende overzicht te zien



Hier kun je de scores bijwerken en als je dan op de update knop klikt dan zullen de scores in de achtergrond door een PHP script worden geüpload naar de database.

1.7 Het aanmaken van speler

Dit wordt gedaan door een eerder genoemde form. Hier achter draait een PHP script die in connectie staat met een database. Er zit ook een foreach loop in om te zien welke nummer bij welk team hoort. Zo weet je welke speler je aan welk team kunt toevoegen. Het form dat gebruikt wordt voor het aanmaken van een team ziet er zo uit



Het moment dat de gebruiker op creëer speler klikt zal deze in de database worden geüpload.

1.8 Het aanpassen van teams

Dit wordt gedaan door een eerder genoemde form. Hier achter draait een PHP script die in connectie staat met een database. Het form dat gebruikt wordt voor het wijzigen van team-data ziet er zo uit



Het moment dat de gebruiker op update team klikt zal deze in de database worden geüpload en deze is dan gelijk klaar om gebruikt te worden.

## 1.9 Het aanpassen van spelers

Dit wordt gedaan door een eerder genoemde form. Hier achter draait een PHP script die in connectie staat met een database. Het form dat gebruikt wordt voor het wijzigen van speler-data ziet er zo uit



Het moment dat de gebruiker op update speler klikt zal deze in de database worden geüpload en deze is dan gelijk klaar om gebruikt te worden.

# 2. Functioneel ontwerp van de C# applicatie

Hierin komt alle informatie conform de functies van de C# applicatie.

## 1.1 Hoe werkt de C# applicatie

De C# applicatie staat net als de web-app in samenhang met een database, hieruit haalt de applicatie alle data die hij nodig heeft voor de functionalitieten. Via de C# applicatie kan je de data van de database ophalen en ook aanpassen om zo de uitslagen bekend te maken en zo de punten uit te geven aan de gebruikers die geboden hebben.

## 1.2 De functies

Hiervoor moet onze applicatie ook een aantal functies hebben, die zijn ook te vinden in het MoSCoW ontwerp.

### Verplicht

* Wedden op de uitslagen
* Wedstrijden importeren
* Per wedstrijd kunnen wedden
* Uitslagen importeren

### Tijd over

* Naast punten krijgen ontvang je ook geld

### Komt er sowieso niet in

* Naast punten krijgen ontvang je ook geld

## 1.3 Wedden op de uitslagen

Om te kunnen wedden op de uitslagen moet er een functie komen die opslaat op wie je bied en of hoeveel jij denk dat de score gaat worden. Dit moet dan ook weer opgeslagen worden in de database door middel van een mysql query.

Dit komt er als volgt uit te zien.



Als de gebruiker op “submit” klikt dan word de data doorgestuurd naar de database doormiddel van de query.

## 1.4 Wedstrijden importeren

Vergeleken met de functie in 1.3 word deze functie alleen op de backend uitgevoerd.

Er word naar de database een query gestuurd hier uit haalt de applicatie de benodigde data en laat dan de wedstrijden zien aan de gebruiker.

## 1.5 Per wedstrijd kunnen bieden

Om per wedstrijd te kunnen bieden moet er ook een functie komen die kijkt op welke wedstrijd je van plan bent om te gaan bieden. Hierin moet je dan aangeven op welk team je bied en hoeveel goals je denkt dat ze gaan maken. Of je kan gokken wie er gaat winnen in de wedstrijd hier voor krijg je dan 1 punt en als je de score ook goed had dan krijg je 3 punten.

Dit ziet er als volgt uit.



Als de gebruiker op “submit” klikt word de data naar een database toe gestuurd om daar te worden opgeslagen.

## 1.6 Uitslagen importeren

Net als bij functie 1.4 word deze functie alleen op de achtergrond uitgevoerd en ziet de gebruiker hier niets van. Deze functie die importeert alle uitslagen uit de database om hiermee te bepalen of de gebruiker goed heeft geboden of juist niet.

## 1.7 Naast punten krijgen ontvang je ook geld

Deze functie is niet verplicht en maken we ook alleen als we tijd over hebben.

Als je goed geboden hebt op een team dan krijg je naast je 1 of 3 punten ook nog het geld 1 of 3 keer zo veel geld terug als je hebt ingezet op het genoemde team.

# 3. Wireframe