HET WL-SYSTEEM
(Wachtlijst-Systeem)

Plan van Aanpak

Datum: 12-05-2019
Versie: 1.0
Auteur: Redouan el Hidraoui

Revisiehistorie

Datum	Versie	Omschrijving

Inhoud

1. Inleiding

Bij vele voetbalverenigingen kan er weleens sprake zijn van talloze inschrijvingen op een bepaald moment, deze inschrijvingen moeten allemaal behandeld kunnen worden. Toch zullen de teams echter op een gegeven moment vol raken, hierop heeft voetbalclub VV de Meern en idee op bedacht; een wachtlijstgroep. De inschrijvingen die niet behandeld kunnen worden, worden in de wachtlijstgroep geplaatst zodat deze personen toch nog de kans krijgen om te voetballen tijdens het wachten. Echter kost het enorm wat papierwerk om ij te houden van wie de inschrijvingen zich momenteel bevind in de wachtlijstgroep, daarom moet er een wachtlijst-systeem applicatie geïmplementeerd worden om al het papierwerk terug te dringen;
het WL-Systeem (het WachtLijst-Systeem).

2. Project beschrijving

Persoonlijk heb ik een aantal jaar terug ook een tijd in zo'n wachtlijstgroep gezeten. Het was destijds vrij ingewikkeld om goed te achterhalen wanneer er nou eindelijk een plek vrij kwam in een officieel team, daarom wil ik door middel van het WL-Systeem ervoor zorgen dat dit gehele systeem soepeler en overzichtelijker verloopt. Ik wil er o.a. voor zorgen dat de desbetreffende persoon zijn/haar huidige team kan inzien, een melding ontvangt zodra er een lege plek is ontstaan in een bepaald team en tot slot wil proberen te realiseren dat de persoon een verzoek kan indienen bij de systeembeheerder en dat deze dan de notificatie kan accepteren en dus zo de speler van team kan wijzigen. Wat ik ook wil proberen te realiseren is dat als de systeembeheerder het verzoek niet wilt accepteren, er een reden moet worden meegegeven die de speler dan zou moeten ontvangen.

2.1 Afbakening

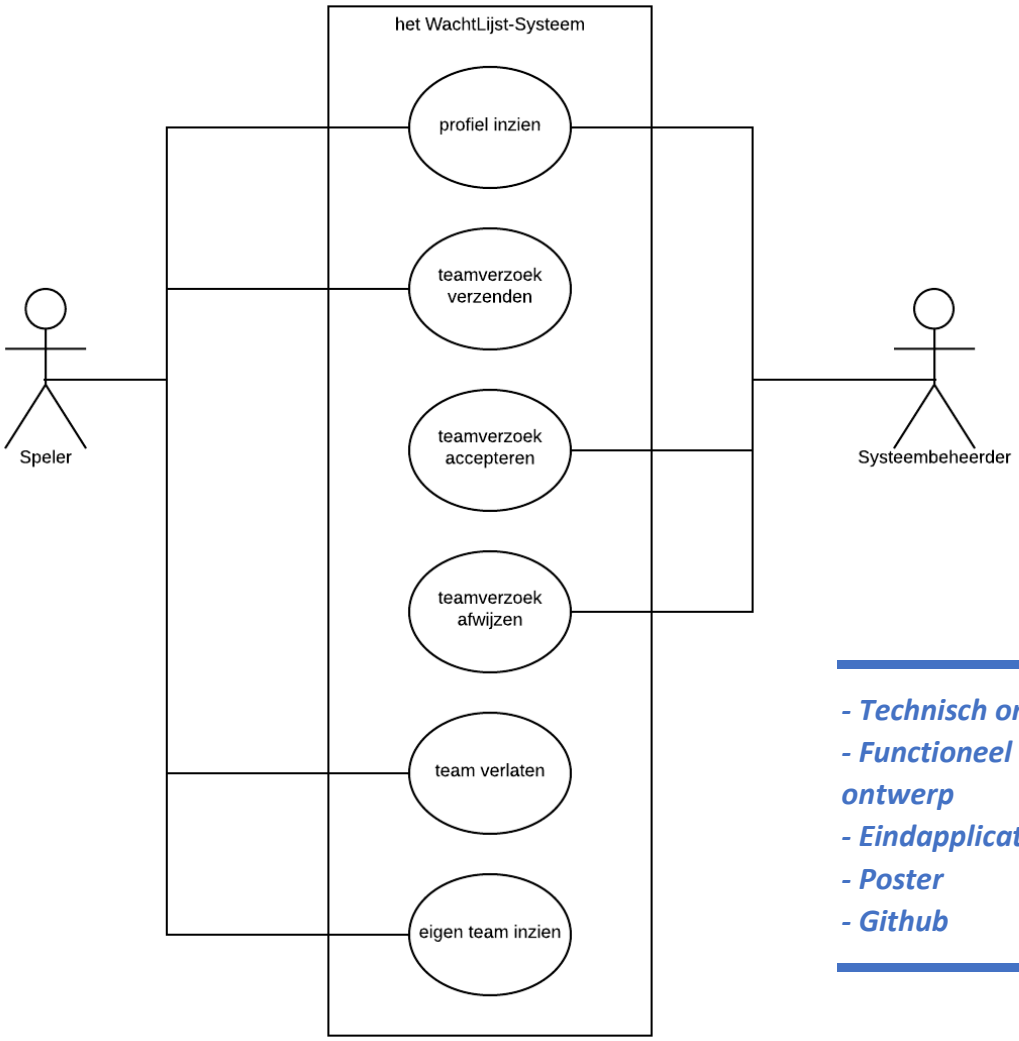
Voor een heldere toelichting maak ik hier gebruik van de MoSCoW-methode;

Must-have (Vereist)	Should-have (Hoge prioriteit, niet vereist)	Could-have (Alleen als er tijd over is)	Won't-have (Eventueel toekomstig, geen prioriteit)
- Inloggen **- Van team wisselen** **- Profiel inzien**	**- Als speler van team wisselen door middel van het accepteren van de melding.** **- Als systeembeheerder meldingen/verzoeken kunnen accepteren** **- Systeembeheerder heeft een unieke homepagina**	**- Account kunnen verwijderen als gebruiker.** **- Vragen versturen aan de systeembeheerder.** **- Andermans profielpagina kunnen vinden/bezoeken**	**- Trainingstijden/wedstrijdtijden per team tonen.** **- Wachtwoord kunnen wijzigen.**

2.2 Op te leveren eindresultaat

Er zal een applicatie worden opgeleverd genaamd het WL-Systeem (zie '**Project beschrijving'**).
Mijn datamodel in het "kort":
De minimale entiteiten in de database zijn "speler", "team" en tot slot "systeembeheerder".
Een speler heeft altijd één team, de wachtlijstgroep wordt ook als één groot team gezien. Een team kan 0 of meerdere spelers bevatten (Stel dat ineens een heel team besluit te stoppen, dan bevat deze team op dat moment 0 spelers.

Er zal een functioneel- en een technisch ontwerp worden opgesteld en opgeleverd.
In het functioneel ontwerp zal ieder Use-case in de onderstaande Use-case Diagram worden voorzien van een samenvatting, met daarbij de Use-case Template. Het technisch ontwerp zal de hierboven genoemde relationeel databasemodel bevatten, en de domeinklasse.



- ***Technisch ontwerp***
- ***Functioneel ontwerp***
- ***Eindapplicatie***
- ***Poster***
- ***Github***

Ook zal er een demo van het WL-Systeem (eindapplicatie) worden getoond, waarbij de werking van de eindapplicatie word getoond. Daarnaast word ook een poster getoond met daarin een korte beschrijving van de opdracht. Ook zal via een GitHub repository link de source code terug te vinden zijn.

2.3 Uitgangspunten, randvoorwaarden en aannames

- De eindapplicatie moet kunnen draaien op Heroku.
- De eindapplicatie moet onafhankelijk van andere projecten testbaar kunnen zijn.
- HTML/CSS front-end
- Java backend
- Database connectie

2.4 Relaties met andere projecten

N.v.t.

3.1 Projectaanpak/fasering

Fase	Tussenresultaat/deelproduct
Plan van aanpak	Een volledig ingevulde bijlage “Plan van Aanpak”. (Word)
Functioneel ontwerp	Een volledig ingevulde bijlage “Functioneel ontwerp”. (Word)
Technisch ontwerp	Een volledig ingevulde bijlage “Technisch ontwerp”. (Word)
Realisatie applicatie Persoonlijk denk ik dat ik hier vrij lang mee bezig zal zijn, aangezien mijn kennis op dit gebied nog redelijk beperkt is. Ik zal hier genoeg tijd voor vrij moeten maken, en op tijd mee moeten beginnen.	Een volledig werkend eindapplicatie die lokaal draait. (HTML, CSS, Javascript, Java, SQL etc..)
Deployment naar de cloud Mijn kennis op dit gebied is ook redelijk beperkt, dus zal ik goed op tijd moeten beginnen om e.v. tegenslagen te voorkomen.	Een volledig werkend eindapplicatie die draait op de cloud. (Heroku)
Poster	Een poster op A3-formaat met daarop een korte beschrijving van de opdracht, de aanpak, de resultaten en een aantal korte conclusies.

3.2 Planning

Deadlines	Week 25 (17-21 juni)	Week 26 (24-28 juni)	Week 27 (1-3 juli)
Plan van aanpak **Zo. 12 mei 23:59**	Front-end HTML 5h	Alles wat de week hiervoor niet is gelukt, wordt in deze week aan doorgewerkt	Voorbereiden voor de demo presentatie van donderdag 4 juli
Functioneel ontwerp **Zo. 16 jun 23:59**	Front-end CSS 5h	Connectie backend met front-end 10h	
Technisch ontwerp **Zo. 16 jun 23:59**	Database creëren 10h	Heroku 10h	
Programma code **Zo. 30 jun 23:59**	Javascript 10h		
Poster **Vr. 5 juli 23:59**	Java klassen schrijven 10h		
Presentatie **Do. 4 juli**	Connectie Java met database 10h		

Ik wil eerst de HTML/CSS creëren, daarna de database. Dan wil ik de Java klassen schrijven en de connectie tussen Java en de database creëren. Vervolgens wil ik m.b.v. de database en javascript een login creëren.

Tot slot wil ik alles werkend krijgen op Heroku.

4. Risico's

Risico	(Eventuele) maatregel
Gebrek aan kennis	IPASS spreekuren bezoeken, medestudenten om hulp vragen.
Laptop crasht/gegevens raken verloren	Altijd naar Github blijven pushen, zodat alles waar je gebleven was niet ineens verloren raakt.
Te hoog gegrepen ideeën	Probeer het simpel te houden. Als er tijd over is, kun je altijd nog één van deze ideeën proberen uit te werken.
Planning word niet goed bijgehouden	Vraag aan je SLB'er of hij/zij je kan steunen bij het aanhouden van je planning/deadlines.
Een bepaald iets duurt te lang om te programmeren wegens gebrek aan kennis	Wanneer je merkt dat het echt te lang duurt, probeer dan een simpeler alternatief te overwegen zodat je altijd resultaat hebt. Het geen wat eerst niet lukte kan altijd nog naar gekeken worden op een later moment.
Tijdsgebrek	Prioriteiten stellen, zaken met hoogste prioriteit eerst afronden.
Een bepaalde fase (zie 3.1) lukt niet wegens gebrek aan kennis	Spreekuren, niet op het laatste moment pas aan een fase beginnen.

5. Referenties

1. (2019). MoSCoW-methode. Geraadpleegd op 12 mei 2019, https://www.marketingtermen.nl/begrip/moscow-methode