

Projecttitel: Subporter

Projectbeschrijving

Wat willen wij realiseren? Tijdens de brainstorm voor ons SBP kwamen enkele ideeën naar boven, die we uiteindelijk niet voor SBP zelf konden realiseren, maar die zeker bruikbaar waren voor de andere modules.

De probleemstelling waar we ons op richten is de volgende: een persoon "Rudy" heeft een abonnement bij zijn favoriete voetbalclub, maar komende zaterdag kan Rudy de match niet bijwonen. Rudy heeft echter een vriend "Tim", tevens fanatiek supporter, maar niet in het bezit van een abonnement.

Het platform dat wij willen realiseren staat Rudy toe om zijn abonnement op een legale wijze uit te lenen aan Tim, die hiervoor een klein bedrag betaalt. Rudy ontvangt op zijn beurt een kleine compensatie vanwege de voetbalploeg.

Hierdoor worden een aantal problemen opgelost en creëren we een win-win situatie voor alle partijen:

1. De voetbalploeg in kwestie vult de reeds lege zitplaats van Rudy terug op, waardoor een leeg stadion geen issue meer vormt. Daarenboven wordt extra omzet gemaakt, aangezien Tim een klein bedrag betaalt om het abonnement van Rudy te lenen. Tim zal hoogstwaarschijnlijk ook wel wat honger en een droge keel krijgen, waardoor hij aan de bar zal consumeren.
2. Rudy krijgt een klein bedrag terug van zijn abonnement, dat hij die avond toch niet kan gebruiken, in de vorm van een compensatie als drank- en eetbonnen of eventuele korting op artikelen in de fanshop.
3. Tevens kan Tim nu ook eens de sfeer gaan opsnuiven tijdens een voetbalmatch, zonder aan het begin van het seizoen een abonnement te hoeven kopen en dat voor een zeer goedkope prijs.

Onze doelgroep is duidelijk: personen die in het bezit zijn van een abonnement voor zijn/haar club en voetbalclubs zelf. Er zal contact met clubs moeten worden genomen om te zien of dit al dan niet aantrekkelijk is voor hun. In principe is het zeker mogelijk hen warm te maken voor dit project, aangezien dergelijke uitleenpraktijken nu ook al gebeuren. Deze personen geraken dan door de iets minder goede controles nu ook al het stadion binnen, maar zijn dus vooraf niet gekend. Dat risico zal ons platform dan ook meteen wegnemen, aangezien profclubs lijsten zouden kunnen opvragen met uitgeleende abonnementen voor de komende match.

Subporter

Kort samengevat zorgt Subporter voor een platform waarop abonnementen uitgeleend worden tussen drie betrokken partijen: de voetbalploeg, de persoon met het abonnement, genaamd Rudy, en de persoon zonder, genaamd Tim. Rudy biedt zijn abonnement te leen aan bij de voetbalploeg en krijgt daarvoor een compensatie. Tim kan dan op zijn beurt dit abonnement "lenen" en ontvangt in ruil voor een klein bedrag een ticket voor de wedstrijd. De voetbalploeg ontvangt dit bedrag en creëert zo, naast extra omzet, ook meer controle over aanwezige supporters tijdens de wedstrijd.

De naam Subporter is redelijk eenvoudig te ontleden: “substituting the supporter”.

Doelgroep

Onze doelgroep bestaat personen die in het bezit zijn van een abonnement voor hun favoriete voetbalploeg. Deze personen kunnen hun abonnement uitlenen voor een zeer kleine prijs, wanneer ze niet naar een bepaalde wedstrijd kunnen gaan kijken, en ontvangen een compensatie. Daarnaast zijn er natuurlijk de andere personen, die graag eens een wedstrijd willen bijwonen in het stadion en ons platform gebruiken om een abonnement te lenen.

Tevens kan het platform rechtstreeks door de voetbalploegen of eventueel de supportersverenigingen gebruikt worden: zij kunnen overgebleven tickets voor de wedstrijd ook aanbieden.

Teamleden

Het team dat Subporter wil uitwerken bestaat uit volgende vier leden:

- Bril Niels (project manager, full stack developer)
- Gemmel Simon (full stack developer)
- Leplae Alexander (front end developer)
- Vande Cappelle Arno (back end developer)

De toegewezen rollen zijn wel een indicatie voor de verantwoordelijkheden van ieder lid, maar tevens zal heel dynamisch gewerkt worden. We hebben elk onze sterktes en zwaktes, maar zullen elkaar helpen waar nodig. Wanneer de één niet verder kan, kan de hulp van een 2^e teamlid wonderen doen.

Technische analyse

Daar deze opdracht gecombineerd zou worden voor zowel front- als backend zullen heel wat technologieën gecombineerd kunnen worden. Kortom betekent dit dat gewerkt zal worden met de MEAN stack: MongoDB, Express, AngularJS, Node.js.

Allereerst het gebruik van AngularJS in combinatie met SASS en GULP voor de frontend. Als backend zal er gewerkt worden met node.js, aangezien dit ook de stof is die ons tijdens de lessen wordt toegereikt.

De data zal opgeslagen worden in een NoSQL database, wat wellicht MongoDB zal worden, en deployment zal hoogstwaarschijnlijk gebeuren op Azure, indien er opnieuw vanuit Howest credits voorzien worden voor leerlingen.

Dit kan natuurlijk nog aangevuld worden met andere libraries of technologieën, afhankelijk van de noden van het project.

Ons project zal gebruik maken van source control met behulp van GitHub. De repository kan gevonden worden op volgende link:

<https://github.com/Subporter>

Functionele analyse

Het project bevat qua functionaliteit heel wat kleine details. Zo zal er voorzien moeten worden in de basisfunctionaliteit om abonnement te lenen en ontlenden, maar moet ook de mogelijkheid voorzien worden om vanwege voetbalploegen lijsten op te vragen met uitgeleende abonnementen. Daarenboven moet gezorgd worden voor een betaalmethode en moet bij het registreren van een abonnement de nodige controle gebeuren of dit effectief een geldig abonnement is voor deze gebruiker.

Planning

Onze planning zal bestaan uit de deadlines die we voor onszelf opstellen. In de loop van komende week (week van 17 oktober) zullen we als team samenzitten om deze data te bespreken.

Het zal belangrijk zijn om enkele cruciale feiten in te calculeren. Er zal rekening moeten gehouden worden met bepaalde vertragingen die kunnen optreden e.d., maar meer hierover in de loop van komende week.

Meer informatie kan ook gevonden worden op Trello:

<https://trello.com/subporter>

Een Work Breakdown Structure is alvast hieronder te vinden, met vermelding van de verantwoordelijke en een eerste richtlijn qua deadline. Deze zaken zullen echter wellicht nog gewijzigd worden in de loop van het project.

Taak	Verantwoordelijke	Deadline
Project management		
WBS	Niels	16/11/2016
Planning	Niels	16/11/2016
Functionele analyse	Alexander	20/11/2016
Branding		
Logo	Simon	20/11/2016
Huisstijl	Simon	20/11/2016
Naam		16/11/2016
Setup		
GitHub repository	Niels	16/11/2016
Slack	Niels	16/11/2016
Trello	Niels	16/11/2016
Frontend		
Project structuur	Niels	20/11/2016
UX Design	Simon + Alexander	22/11/2016
...		
Backend		
Project structuur	Niels	20/11/2016
Models	Arno	22/11/2016
Database	Niels	20/11/2016
Authenticatie	Niels	20/11/2016
Unit-testing		
...		
Testing		
Debugging & fixing		31/12/2016
Deployment		
Webserver opzetten		2/1/2017
Documentatie		2/1/2017

Implementatie

Na de goedkeuring van ons idee door J. Vannieuwenhuyze en K. Derudder begon de eerste fase met name de taakverdeling. Binnen het team werd al snel geconcludeerd dat Niels en Arno zouden instaan voor de backend, terwijl Simon en Alexander zich op de frontend zouden richten. Het eerste struikelblok ontstond meteen, niemand van ons had ergens een idee hoe aan de opdracht te beginnen. Een teammeeting hielp ons dan toch op weg naar de realisatie van een functionaliteitenlijst.

In de eerste stadia van de projectrealisatie werd de authenticatie tussen front – en backend door Niels gerealiseerd. Arno spitste zich toe op het gebruik van sockets zodat de mogelijkheid bestond een realtime applicatie te ontwikkelen. Simon ontwikkelde de huidige landingspage en enkele overige webpagina's. Alexander ging op zoek naar een huisstijl en ontwikkelde het logo in samenwerking met Simon.

Subporter maakt gebruik van een noSQL database (MongoDB) waardoor het gebruik van sql statements en inner joins niet toepasbaar was. Niels spitste zich vervolgens toe op het maken van onze eigen api, hierdoor kon data wel worden opgevraagd m.b.v api-calls en meerdere queries. Het schrijven van de api en deze ook werkende te krijgen was de grootste taak die moest gebeuren.

Voor gebruikers van Subporter is het belangrijk dat ze de wedstrijden, gegevens, data van hun favoriete ploegen duidelijk kunnen terugvinden op de homepage. Een fan van K.A.A. Gent zal niet geïnteresseerd zijn in wedstrijden van Anderlecht. Om deze reden is het correct gebruik van api-calls cruciaal.

Een service waar abonnementen worden aangeboden moet altijd up-to-date zijn, vanaf het moment er een nieuw abonnement wordt aangeboden op de website moeten geïnteresseerde gebruikers hiervan op de hoogte gebracht worden. Dit gebeurt met een popup-melding die na enkele seconden terug verdwijnt, of tot wanneer de gebruiker ze wegvinkt. Om dit realtime mogelijk te maken wordt er gebruik gemaakt van sockets. Een bijkomend voordeel van sockets is dat wanneer de webpagina geüpdatet wordt, tijdens het herladen ervan, deze ervoor zorgen dat de website minder belast wordt. De data die door de sockets wordt bijgehouden is realtime en moet dus niet herladen worden.

Vele gebruikers zullen de service via hun mobiele toestel raadplegen, hiervoor was een responsive design van de webpagina's een noodzaak. Alexander nam dit deel voor zijn rekening.

Tijdens de realisatie werd ons meer en meer duidelijk dat we dit project na onze studies gerust verder zouden kunnen uitwerken. De markt is er, de vraag naar een platform als Subporter bestaat hoogstwaarschijnlijk ook. Dit was voor ons dan ook de drijfveer om tijd te investeren in de realisatie van het platform. Het draaide niet meer enkel om het (school)resultaat, dit konden voor ons de eerste stappen zijn in ons verdere leven na onze studies.

Op een platform als dit wordt vaak gewerkt met gevoelige, persoonlijke informatie, we denken hierbij niet alleen aan wachtwoorden, maar ook aan abonnementsgegevens e.d. Niels zocht een oplossing om zo secure mogelijk te werken qua authenticatie en autorisatie en is hier ook in geslaagd. Https werd voorzien om end-to-end encryptie te ondersteunen.

Realisatie

Werkverdeling was voor dit project een moeilijke opgave. Het is altijd zo dat bij een groepswork de werkuren sterk verdeeld zijn, de ene persoon kan meer realiseren in een kortere tijdsperiode dan een andere, de ene heeft een programmeeraanleg, de ander niet. Ondanks deze verschillen werd toch door iedereen samengewerkt en droeg ieder zijn steentje bij.

Niels heeft een enorme aanleg voor programmeren, dus nam hij de taak van de backend grotendeels op zich. Het maken van de api, de database, de server en ga zo maar verder. Arno spitste zich toe op het gebruik van sockets wat ook nieuwe materie was en dus zelfstudie met zich meebracht.

Simon is technisch zeer sterk op vlak van design en front-end programmatie. Hij nam het maken van de webpagina's voor zijn rekening, mede ook de profielpagina van gebruikers maakte hij en enkele functionaliteiten die met de front-end gepaard gingen.

Alexander toonde zijn inzet, heeft niet meteen de intentie om programmeur te worden, maar hield zich dan bezig met de tijdrovende, minder technische zaken. Het responsive maken van de webpagina's. creëren van het logo, de footer, de werkdocumenten bijhouden.

Elk van ons heeft zo zijn sterktes en zwaktes en helpt waar nodig, dit maakt ons een sterk gevarieerd team dat op elkaar kan inspelen. Taakverdelingen werden hermaakt en mogelijke frustraties werden uitgepraat. Ook met het oog op de toekomst en wat er met Subporter kan gebeuren is al heel wat besproken geweest. Ieder heeft zijn rol.

We zijn als team vooral trots op de backend van ons project, dit was voor ieder van ons een volledig nieuw onderwerp en deze staat voorlopig voor 95% op poten. Uitgewerkte manieren voor authorisatie en caching zijn aanwezig terwijl dit onderwerpen waren die in de lessen nog niet aan bod kwamen. De debian webserver waar voorlopig op gehost wordt werd volledig zelf geconfigureerd door Niels.

Als we terugkijken op dit project zijn er toch wel enkele zaken die anders of beter konden. We waren één van de laatste groepen die aan hun project was gestart, dus de eerste opmerking is het sneller afronden van de taakverdeling en wel degelijk aan het project beginnen werken. Duidelijke afspraken maken over wie wat maakt zodat dubbel werk vermeden kan worden en wanneer we niet weten hoe te starten zou de beste oplossing zijn om samen te zitten en hier als team over te brainstormen vooraleer aan de slag te gaan.

Overzicht

Onderstaand schema geeft u een samenvatting van wat met dit project gerealiseerd werd en welke pakketten gebruikt werden.

De Applicatie	Subporter	
De server-taal	Javascript	
Framework	Express	
Foutcontrole	Eigen + logging van uncaughtExceptions met winston	
Persistent stockeren	MongoDB met mongoose Mongoose-increment voor oplopende id's	
Realtime websockets	Socket.io	
Unit-testing	Mocha	
Registratie (+authorizatie)	jwt tokens passport library	
Deployment	Eigen server (Debian)	
Source Control	Github	
Caching	Redis cache	