Realtime tracking van voetbalwedstrijden

Versie: 0.0.1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Projectteam

|  |  |  |  |
| --- | --- | --- | --- |
| **Naam en voornaam** | Volgt BE | Volgt  FE | Functie binnen het team |
| Martijn Loth | OK | OK | BE en FE |
| Bart Callant | NOK | OK | FE en ontwikkeling applicatie |
| Kristof Colpaert | OK | OK | BE en FE |

# Deployment parameters

|  |  |
| --- | --- |
| **TFS of GITHUB?** | Github |
| **URL** | <https://github.com/KristofColpaert/NMCTBackFront> |
| **LOGIN** (indien toepasselijk) | N.V.T. |
| **ADMIN info** | Login: [docent@nmct.be](mailto:docent@nmct.be)  Password: -Password1 |

# Gebruikerseisen

## Algemene beschrijving

We streven ernaar om via onze applicatie realtime de posities van spelers en bal in een voetbalspel weer te geven. Deze data wordt live gepresenteerd via een webinterface en opgeslagen in een database voor latere analyse. Op basis van de opgeslagen data worden bijvoorbeeld ook heat maps opgesteld.

## Meerwaarde voor de gebruiker

Wij zien de volgende voordelen voor de gebruiker:

1. De gebruiker beschikt over realtime-informatie gedurende de match.
2. Tijdens of na de wedstrijd kunnen bepaalde fases herbekeken worden vanuit een ander standpunt. De opgeslagen data kan bovendien aangewend worden om bijkomende analyses, zoals heat maps, te genereren.
3. Het systeem laat de gebruiker toe om wedstrijden vanuit een ander perspectief te bekijken. Hij kan het ook als second screen-applicatie gebruiken tijdens de wedstrijd.
4. Het systeem beperkt zich niet tot voetbalwedstrijden: het kan ook gebruikt worden voor basketbal, rugby, atletiek, tennis of zelfs autoraces.

# Implementatie, methodiek & technologie

## Hardware

De data die we in onze applicatie gebruiken, wordt gegenereerd door een wisselwerking van beacons (verzenden Bluetooth-signalen) en iPhones (ontvangen Bluetooth-signalen). Wij maken gebruik van beacons van het merk [Estimote](http://estimote.com/indoor/), omdat deze standaard komen met een goede SDK voor het berekenen van posities en locaties van objecten.

Wanneer de iPhone via de Estimote Indoor Location SDK informatie ontvangt over zijn eigen positie, zendt hij die via sockets door naar een server. Deze server slaat de locatiedata op in een database (MongoDB) en stuurt ze tegelijkertijd via sockets verder naar aangesloten clients.

## Functionaliteit frontend

Voor de frontend verwachten wij de volgende functionaliteit te implementeren:

* Liveweergave van een voetbalmatch wanneer deze bezig is. Veld, spelers en bal zijn op voldoende wijze van elkaar te onderscheiden.
* Detectie van een aantal situaties, waaronder buitenspelval en goals.
* Achteraf kunnen wedstrijden opnieuw bekeken worden. Er kan genavigeerd worden naar een bepaald tijdstip uit de wedstrijden.
* Er kunnen rapporten opgesteld worden over een wedstrijd: heat maps en overzichten van cruciale fases (bijvoorbeeld de passen die leiden tot een goal).
* Bovenstaande rapporten kunnen in een PDF gegoten worden.
* Er is een managementinterface voorzien waarin kan aangegeven worden welke speler gekoppeld is aan welke iPhone.
* Wedstrijden kunnen lokaal opgeslagen worden om offline bekeken te worden.

## Functionaliteit backend

Voor de backend verwachten wij de volgende functionaliteit te implementeren:

* Registreren van locatiebepalingen en opslag in een database.
* Het genereren van rapporten (bijvoorbeeld heat maps) op basis van de opgeslagen data van een match (zie frontend).
* Authenticatie en beveiliging.

## Technologie

* De **iPhones** draaien een native applicatie. Er wordt gebruik gemaakt van de [Estimote Indoor Location SDK](http://estimote.com/indoor/) voor locatiebepalingen. Deze SDK is enkel beschikbaar voor **Swift**, wat meteen de keuze voor een iPhone-applicatie verklaart.
* De **backend** is een eenvoudige REST-API die is opgebouwd in **Node.js** (4.2.4. LTS) en die gebruik maakt van het **Express**-framework (4). Authenticatie wordt voorzien via [**JSON Web Tokens**](https://github.com/auth0/node-jsonwebtoken)**.** De onderliggende datalaag berust op **MongoDB** in combinatie met **MongoSkin** als driver.
* De **frontend** draait op **React** en gebruikt **Flux** als data pattern**.** De keuze voor React en Flux mag, gezien de nadruk op Angular, dan wel vreemd klinken, maar heeft toch duidelijke gronden:
  + React is door de implementatie van een virtuele DOM een stuk sneller dan andere frontend frameworks. Concreet worden alle wijzigingen in de layout onderbracht in een virtuele DOM. Vervolgens wordt die afgespiegeld aan de reële DOM en worden slechts de verschillen geadopteerd in de werkelijke DOM. Deze techniek zorgt ervoor dat er minder reflows plaatsvinden.
  + React voorziet enkel een view (geen controller of datalaag), en laat daardoor meer bewegingsruimte aan de ontwikkelaar.

## Frontend

|  |  |
| --- | --- |
| Technologie | Waarom |
| React en Flux | React is sneller dan andere frontend frameworks door de implementatie van een virtuele DOM. Bovendien wordt er meer bewegingsruimte aan de ontwikkelaar gelaten.  Aangezien React enkel voorziet in een view, hebben wij het uitgebreid met Flux om de dataflow te beheren en te controleren. |
| Sass | Sass is de meest gebruikte CSS preprocessor. |
| Gulp | Gulp is een van de meeste gebruikte taskrunners. |
| Jest | Jest is een testing framework op basis van Jasmine. Het werd ontwikkeld door Facebook om te gebruiken in combinatie met het eveneens door Facebook ontwikkelde React en Flux. |

## Backend

|  |  |
| --- | --- |
| Technologie | Waarom |
| Node.js en Express | De backend draait op Node.js (vereiste) in combinatie met het Express-framework. Er werd gekozen voor Express wegens de eenvoudige afhandeling van de request-response-cycle en de mogelijkheid tot het implementeren van middleware. |
| MongoDB | De datalaag berust op MongoDB in combinatie met MongoSkin als driver in Node.js. |
| Socket.io | De realtimecommunicatie tussen client en server gebeurt via sockets. Sockets hebben als voordeel dat ze minder overhead hebben dan HTTP-requests via Ajax en dat ze bijgevolg sneller zijn. |
| Mocha | Mocha is het meestgebruikte testing framework voor Node.js. |
| JSON Web Tokens | JSON Web Tokens maken het mogelijk om de backend te voorzien van autorisatie en authenticatie. |

# Taken

Onderstaande lijst geeft een overzicht van alle taken die moeten worden uitgevoerd. Er wordt een onderscheid gemaakt tussen frontend, backend, debugging en deployment.

## Frontend

* Ontwerp:
  + Wireframes.
  + Huisstijl (look and feel, kleurenpalet).
  + Logo.
  + UI.
* Functionaliteit – managementinterface:
  + Teams aanmaken (en kleuren toewijzen aan teams).
  + Overzicht erkende iPhones binnen veld.
  + Beacons toewijzen aan een team en een speler.
  + Het opnemen/registreren van een match starten.
  + Het opnemen/registreren van een belangrijke fase starten en stoppen.
* Functionaliteit – gebruikersinterface:
  + Liveweergave van een match.
  + Detectie buitenspelval.
  + Detectie goal.
  + Herbekijken match.
  + Ontwikkelen heat map match.
  + Ontwikkelen mogelijkheid om match (achteraf, niet live) terug te spoelen.
  + Ontwikkelen mogelijkheid om match (achteraf, niet live) vertraagd te bekijken.
  + Ontwikkelen mogelijkheid om match (achteraf, niet live) versneld te bekijken.
  + Ontwikkelen mogelijkheid om match (achteraf, niet live) als een aaneenschakeling van belangrijke fases te bekijken.
  + Ontwikkelen mogelijkheid om match offline te bekijken.

## Backend

* Beacons:
  + Klaarmaken hardware (beacons).
  + Detectie van beacons.
  + Verzamelen locatiebepalingen en doorsturen naar server.
* API:
  + Ontwikkeling van API inclusief routing.
  + Authenticatie en autorisatie.
* Locatiebepalingen:
  + Opslaan van locatiebepalingen in de database (meerdere keren per seconde).
* Rapporten:
  + Het genereren van een heat map van een match.
  + Het genereren van een PDF met de heat map van een match.
  + Het genereren van een rapport met belangrijke fases van een match (voor belangrijke fases: zie managementinterfaces bij frontend).
  + Het genereren van een PDF-rapport met belangrijke fases van een match.

## Testing, debugging, commenting en deployment

* Testen en debuggen van de applicatie.
* Documenteren van de applicatie.
* Deployment op een externe service.

# Planning

## Frontend

|  |  |  |
| --- | --- | --- |
| Taak | Personeel | Deadline |
| **Ontwerp** |  |  |
| Wireframes. | Martijn | 29/10 |
| Huisstijl. | Martijn | 29/10 |
| Logo. | Martijn | 06/11 |
| UI. | Martijn | 09/11 |
| **Functionaliteit – managementinterface** |  |  |
| Teams aanmaken (kleuren). | Kristof | 09/11 |
| Overzicht erkende beacons in veld. | Kristof | 09/11 |
| Beacon toewijzen aan team en speler. | Kristof | 09/11 |
| Start/stop registratie van een match. | Bart | 20/11 |
| Start/stop registratie van een belangrijke fase. | Kristof | 20/11 |
| **Functionaliteit – gebruikersinterface** |  |  |
| Liveweergave van een match. | Martijn | 27/11 |
| Detectie buitenspelval. | Bart | 27/11 |
| Detectie goal. | Bart | 27/11 |
| Herbekijken match. | Kristof | 11/12 |
| Ontwikkelen heat map match. | Kristof | 11/12 |
| Ontwikkelen mogelijkheid om match (achteraf, niet live) terug te spoelen. | Bart | 11/12 |
| Ontwikkelen mogelijkheid om match (achteraf, niet live) vertraagd te bekijken. | Bart | 11/12 |
| Ontwikkelen mogelijkheid om match (achteraf, niet live) versneld te bekijken. | Bart | 11/12 |
| Ontwikkelen mogelijkheid om match (achteraf, niet live) als een aaneenschakeling van belangrijke fases te bekijken. | Bart | 11/12 |
| Ontwikkelen mogelijkheid om een match offline te herbekijken. | Kristof | 18/12 |

## Backend

|  |  |  |
| --- | --- | --- |
| Taak | Personeel | Deadline |
| **Beacons** |  |  |
| Klaarmaken hardware (beacons). | Bart | 09/11 |
| Detectie van beacons. | Bart | 09/11 |
| Verzamelen locatiebepalingen en doorsturen naar server. | Bart | 09/11 |
| **API** |  |  |
| Ontwikkeling van API inclusief routing. | Kristof | 27/11 |
| Authenticatie en autorisatie. | Kristof | 27/11 |
| **Locatiebepalingen** |  |  |
| Opslaan van locatiebepalingen in de database (meerdere keren per seconde). | Kristof | 27/11 |
| **Rapporten** |  |  |
| Het genereren van een heat map van een match. | Kristof | 27/11 |
| Het genereren van een PDF met de heat map van een match. | Kristof | 27/11 |
| Het genereren van een rapport met belangrijke fases van een match (voor belangrijke fases: zie managementinterfaces bij frontend). | Kristof | 27/11 |
| Het genereren van een PDF-rapport met belangrijke fases van een match. | Kristof | 27/11 |

## Testing, debugging, commenting en deployment

|  |  |  |
| --- | --- | --- |
| Taak | Personeel | Deadline |
| Testen en debuggen van de applicatie. | Bart / Kristof | 18/12 |
| Documenteren van de applicatie. | Bart / Kristof / Martijn | 18/12 |
| Deployment op een externe service. | Kristof | 18/12 |

# Realisatie volgens planning

De realisatie is verlopen volgens planning. Door het experimentele karakter van het project en de vele hindernissen die we onderweg tegenkwamen, is het ons niet gelukt om alle vooropgestelde taken correct uit te voeren. Vooral het bestuderen van de harware (beacons) en React (nieuw platform voor ons) bleek meer tijd te vragen dan we hadden vooropgesteld.

Uiteindelijk is de volgende functionaliteit **niet geïmplementeerd**:

* Het aanduiden van belangrijke fases in een match.
  + Ontwikkelen mogelijkheid om een match als een aaneenschakeling van belangrijke fases te bekijken.
  + Het genereren van PDF-rapporten met een overzicht van alle belangrijke fases van een match.
  + Het genereren van PDF-rapporten van de heat maps van een match.
* Detectie buitenspelval (technisch niet mogelijk door onnauwkeurigheid van de hardware).
* Detectie goals (technisch niet mogelijk door onnauwkeurigheid van de hardware).

# Toekomstplannen of mogelijke uitbreidingen

Als belangrijkste toekomstplannen noteren we de zaken waarvan we niet geslaagd zijn ze te implementeren tijdens de looptijd van het project. In een volgende fase zou het mogelijk moeten worden om de applicatie uit te breiden naar andere sportwedstrijden zoals atletiek, tennis, basketbal en zelfs autoraces.

# Succesfactoren

Het succes van het project hangt in grote mate af van de implementatie van de vooropgestelde taken. Enkele functionaliteiten die zeker aanwezig moeten zijn:

* De liveweergave van een match.
* Het herbekijken van een match.
* Het genereren van statistieken (heat maps) van een match.

Bovenstaande zaken werden stuk voor stuk geïmplementeerd in de applicatie.

# Conclusie

Het project heeft voor ons een grote experimentele waarde. We hebben voor het eerst gewerkt met beacons en kozen voor een afwijkend frontend framework (React), waardoor we niet zeker waren dat we het vooropgestelde resultaat zouden behalen. Niettemin slaagden we erin het overgrote deel van de functionaliteit te implementeren en zo het project toch tot een goed einde te brengen.