

[Ucll beacons]

Project dossier Mobiele app

Dit project dossier bevat alle informatie van het UCLL BEACONS MOBIELE APP project.

Project omschrijving, opbouw van de app, documentatie.

Sven Liekens, Jo Claes en Jens Molenaers.

Project dossier Mobiele app

[Ucll beacons]

1. Inhoud

[2 Project omschrijving 5](#_Toc483661995)

[2.1 iBeacons 5](#_Toc483661996)

[2.2 Onze opdracht 6](#_Toc483661997)

[3 Opbouw app 7](#_Toc483661998)

[4 Screen mockups app 8](#_Toc483661999)

[4.1 Splash screen 8](#_Toc483662000)

[4.2 Front screen 8](#_Toc483662001)

[4.3 Route screen 9](#_Toc483662002)

[4.4 Content screens 10](#_Toc483662003)

[4.4.1 Html 10](#_Toc483662004)

[4.4.2 YouTube 10](#_Toc483662005)

[4.4.3 Foto 10](#_Toc483662006)

[4.4.4 Audio 10](#_Toc483662007)

[4.4.5 Video 10](#_Toc483662008)

[5 Linken 11](#_Toc483662009)

Afbeelding 1: IBeacons Apple 5

Afbeelding 2: Screen mockup geen internet 8

Afbeelding 3: Screen mockup front screen 8

Afbeelding 4: Screen mockup route screen 9

Afbeelding 5: Screen mockup geen bluetooth 10

# Project omschrijving

## iBeacons

IBeacons zijn kleine zendertjes (bluetooth), die gericht informatie naar mobiele telefoons kunnen sturen als je in de buurt bent.

Apple introduceerde deze iBeacon-technologie in juni 2013 tijdens de ontwikkelaarsconferentie WWDC 2013. Sinds iOS 7 is ondersteuning voor iBeacons aanwezig, waarbij gebruikt wordt gemaakt van Bluetooth 4.0 voor de draadloze verbinding. IBeacons hebben afhankelijk van het type een bereik tot wel 50 meter. iBeacons werken ook met smartphones van andere merken.

[](http://iculture.textopus.nl/wp-content/uploads/2014/07/ibeacons-schoenen.jpg)Via Bluetooth kan een app de nabijheid van een iBeacon inschatten, bijvoorbeeld wanneer deze in een winkel is geplaatst. Apple gebruikt zelf iBeacons in de Apple Store: als je bij een bepaald product staat kun je via de Apple Store-app meer informatie opvragen. De iBeacon maakt een schatting van jouw locatie en stuurt informatie over het product waar je dichtbij staat.

Afbeelding 1: IBeacons Apple

IBeacons vormen een alternatief voor NFC, de draadloze technologie die ook in de iPhone voorkomt. Het verschil is dat NFC alleen op korte afstand werkt, namelijk enkele centimeters. Bij iBeacons mag de afstand veel groter zijn, tot wel 50 meter. iBeacons worden daarom vooral gebruikt om informatie te verstrekken en niet zoals NFC om betalingen te doen.

**Toepassingen van iBeacons**

iBeacons bieden eigenlijk tal van mogelijkheden. De toepassingen zijn in het algemeen onder te brengen in twee grote groepen.

* Apps die jou, op basis van je locatie, informatie geven betreffende deze locatie (musea, winkels, openbare plaatsen, muziekfestivals…).
* Apps die op basis van je locatie jou navigeert naar het punt waar je moet zijn. (Musea, hospitaal, luchthaven, muziekfestivals…)

## Onze opdracht

Onze opdracht is om een mobiele app te maken om een bepaalde navigatieroute te volgen en onderweg op je route gerichte informatie te krijgen op bepaalde locaties.

Deze opdracht is op te splitsen in drie deelopdrachten.

* Een REST api om informatie te leveren aan de mobiele client (json formaat)
* Een webapplicatie om de informatie en de routes op te slaan.
* De mobiele applicatie voor de toepassing.

Binnen het opleidingsonderdeel Mobiele apps 1 maken we de mobiele applicatie. Deze staat in dit dossier beschreven met de opbouw van de app en documentatie.

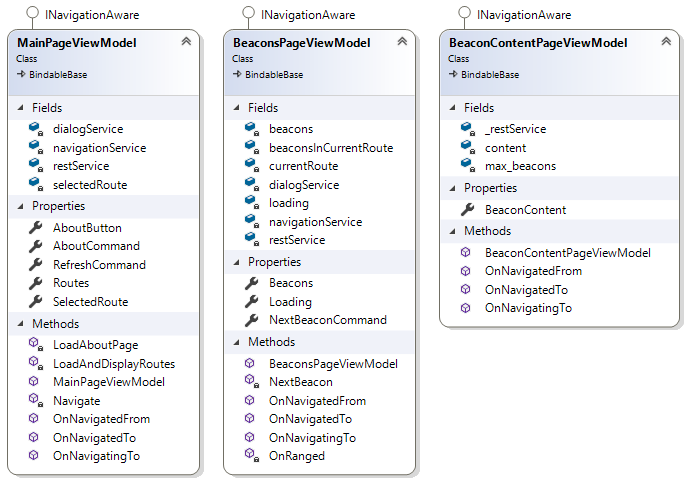
# Opbouw app

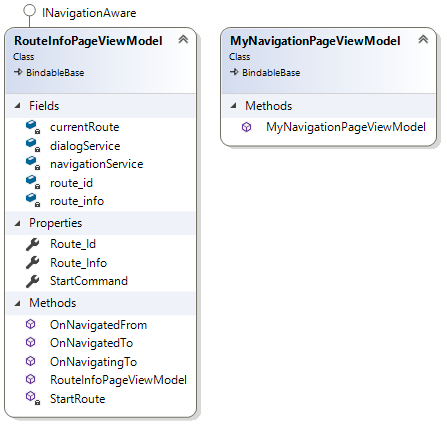
Hier komen stukken code met meer uitleg van de app.

## Models



## ViewModels





**in MainPageViewModel.cs**

private Route selectedRoute;

public Route SelectedRoute

{

// vervangen door event ItemSelected & checken of SelectedItem = null

//(itemselected wordt 2x uitgevoerd: bij indrukken en bij loslaten

get { return selectedRoute; }

set

{

if (SetProperty(ref selectedRoute, value) && selectedRoute != null)

{

// Wanneer er een route geselecteerd wordt navigeren we naar de RouteInfoPage

// en geven we deze route mee met de navigatie parameters

var navParams = new NavigationParameters();

navParams.Add("selectedRoute", selectedRoute);

Navigate("RouteInfoPage", navParams);

SelectedRoute = null;

}

}

De geselecteerde route in property steken.

Het inladen van alle routes uit de restservice als er internetconnectie is en tonen van de routes op de MainPage.

private async Task LoadAndDisplayRoutes()

{

try

{

// Routes ophalen via de restService

ObservableCollection<Route> routes = await restService.GetRoutesAsync();

//if (routes != Routes)

//{

// Als de routes die ingeladen werden verschillend zijn van degenen die   
 weergegeven worden refreshen we het lijstje, anders niet.

Routes.Clear();

foreach (Route r in routes)

{

Routes.Add(r);

}

//}

}

catch (HttpRequestException)

{

await dialogService.DisplayAlertAsync("Geen internetverbinding.", "Sluit de   
 app volledig af en zorg voor een werkende internetverbinding.", "OK");

}

catch (WebException)

{

await dialogService.DisplayAlertAsync("Geen internetverbinding.", "Sluit de  
 app volledig af en zorg voor een werkende internetverbinding.", "OK");

}

catch (Exception e)

{

await dialogService.DisplayAlertAsync("Error", e.Message, "OK");

}

}

**in BeaconsPageViewModel.cs**

private void NextBeacon()

{

Beacons = "";

if (App.currentSequenceNumber < beaconsInCurrentRoute.Count - 1)

{

App.currentSequenceNumber++;

Beacons = "Ga naar: " + beaconsInCurrentRoute[App.currentSequenceNumber].Location\_Ln;

}

else

{

Beacons = "last beacon reached";

}

}

De volgende beacon die moet gezocht worden zoeken en de locatie ervan tonen.

private void OnRanged(object sender, IEnumerable<IBeacon> foundBeacons)

{

foreach (var foundBeacon in foundBeacons)

{

if (beaconsInCurrentRoute[App.currentSequenceNumber].Beacon\_Id == foundBeacon.Major)

{

var navParams = new NavigationParameters();

navParams.Add("route\_id", currentRoute.Route\_Id);

navParams.Add("beacon\_id",  
 beaconsInCurrentRoute[App.currentSequenceNumber].Beacon\_Id);

navParams.Add("max\_beacons", (beaconsInCurrentRoute.Count - 1));

navigationService.NavigateAsync("BeaconContentPage", navParams);

}

}

}

Het ordenen van de beacons van de geselecteerde route op hun volgnummer.

**in BeaconContentPageViewModel.cs**

public async void OnNavigatingTo(NavigationParameters parameters)

{

int route\_id = -1;

int beacon\_id = -1;

if (parameters.ContainsKey("route\_id")) route\_id = (int)parameters["route\_id"];

if (parameters.ContainsKey("beacon\_id")) beacon\_id = (int)parameters["beacon\_id"];

if (parameters.ContainsKey("max\_beacons")) max\_beacons = (int)parameters["max\_beacons"];

ObservableCollection<Content> content = await  
 \_restService.GetContentForBeaconInRoute(route\_id, beacon\_id);

SourceUrl = content[0].Content\_Txt;

}

De content van de beacon tonen.

**in** **RouteInfoPageViewModel.cs**

private async void StartRoute()

{

NavigationParameters navParams = new NavigationParameters();

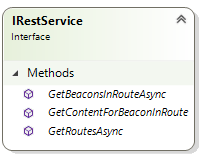
navParams.Add("selectedRoute", currentRoute);

await navigationService.NavigateAsync("BeaconsPage", navParams);

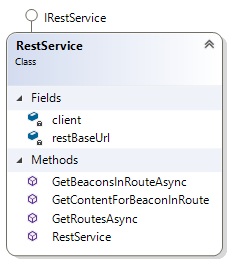
}

De route starten door naar de BeaconsPage te navigeren en de geselcteerde route mee te sturen.

## Interfaces



## Data



**in** **Restservice.cs**

public async Task<ObservableCollection<Beacon>> GetBeaconsInRouteAsync(int route\_id)

{

var uri = new Uri(restBaseUrl + "/route/" + route\_id + "/beacon");

ObservableCollection<Beacon> beaconsInRoute = null;

var response = await client.GetAsync(uri);

if (response.IsSuccessStatusCode)

{

var content = await response.Content.ReadAsStringAsync();

beaconsInRoute = JsonConvert.DeserializeObject<ObservableCollection<Beacon>>(content);

}

return beaconsInRoute;

}

Beacons van meegestuurde route binnenhalen van restservice.   
(URL: http://api.beacons.ucll.be/v1/route/”route\_id”/beacon)

public async Task<ObservableCollection<Route>> GetRoutesAsync()

{

var uri = new Uri(restBaseUrl + "/route");

ObservableCollection<Route> routes = null;

var response = await client.GetAsync(uri);

if (response.IsSuccessStatusCode)

{

var content = await response.Content.ReadAsStringAsync();

routes = JsonConvert.DeserializeObject<ObservableCollection<Route>>(content);

}

return routes;

}

Routes binnenhalen van restservice.   
(URL: http://api.beacons.ucll.be/v1/route)

public async Task<ObservableCollection<Content>> GetContentForBeaconInRoute(int route\_id,

int beacon\_id)

{

// Voorbeeld request: http://api.beacons.ucll.be/v1/beacon/4/route/1/dynamicData

var uri = new Uri(restBaseUrl + "/beacon/" + beacon\_id.ToString() + "/route/" +

route\_id.ToString() + "/dynamicData");

ObservableCollection<Content> content = null;

var response = await client.GetAsync(uri);

if (response.IsSuccessStatusCode)

{

var responseContent = await response.Content.ReadAsStringAsync();

content= JsonConvert.DeserializeObject<ObservableCollection<Content>>(responseContent);

}

return content;

}

Content van meegestuurde beacon binnen meegsestuurde route binnenhalen van restservice.   
(URL: http://api.beacons.ucll.be/v1/beacon/”beacon\_id”/route/”route\_id”/ dynamicData)

# Screen mockups app

## Splash screen

Als men de app opstart krijgt men eerst een splash screen dat het logo is van de organisatie.

## Front screen

Op het front screen krijgt men een opsomming van de verschillende routes die zich in de database bevinden.

Per route krijgt men een afbeelding en het Id van de route te zien, ook bevind zich hier de naam.

Rechts in de bovenhoek bevindt zich een refresh knop. Deze knop gaat de pagina refreshen. Is er een update geweest in de database gaat deze de info opvragen en de pagina updaten als men de refresh knop heeft ingedrukt.

Als men een route selecteert gaat men verder naar het volgende scherm waar een knop getoond word.

Extra functie die er in zit is als men vergeet de wifi op te zetten van zijn apparaat. Hierbij krijgt de gebruiker een melding om de app af te sluiten en de wifi op te zetten.

Afbeelding 2: Screen mockup geen internet

Afbeelding 3: Screen mockup front screen

## Route screen

Op het route scherm krijgt de gebruiker in de titel te zien in welke route hij zit.

In het midden van het scherm bevind zich een knop waar de gebruiker de route kan starten die hij heeft aangeduid vanuit het vorige menu.

Boven de knop komt nog te staan: “Welkom bij route-naam”.

Als de gebruiker de route wilt starten klikt hij op de knop en gaat de applicatie verder naar het volgende scherm.

Voordat hij volledig verder gaat krijgt de gebruiker een wacht icoon op het scherm.

Afbeelding 4: Screen mockup route screen

## C:\Users\jensie\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20170527-150324.pngContent screens

Als een gebruiker een bepaalde route gestart heeft, gaat de applicatie opzoek naar de beacon die het kortst bij de gebruiker bevindt. Eenmaal als de gebruiker kort bij een beacon komt wordt dit opgevangen door de applicatie en geeft deze bepaalde content naar verwant het soort media.

Als de gebruiker zijn bluetooth niet aanheeft krijgt men een melding dat deze de bluetooth moet aanzetten en gaat terug naar de vorige pagina.

Afbeelding 5: Screen mockup geen bluetooth

### Html

Als de media van het soort html is wordt er een html pagina getoond.

### YouTube

Is de media van het YouTube formaat (YouTube link) openen we een video van YouTube.

### Foto

Als we een foto aankrijgen als media tonen we de foto op het apparaat.

### Audio

Met het formaat audio spelen we de muziek af op het apparaat.

### Video

Als we van media een video hebben spelen we de video af op het apparaat van de gebruiker.

# Linken

* Developer android: <https://developer.android.com/index.html>
* Material design: <https://material.io/guidelines/style/color.html>
* Developer Xamarin: <https://developer.xamarin.com/>
* Material icons: <https://material.io/icons/>
* Icon & splash screen generator: <http://www.zelf-een-app-maken.nl/app-icon-splashscreen-generator/>
* Github: <https://github.com/jensMole/MobieleApp>
* GithubGist voor info over bepaalde code: <https://gist.github.com/> <https://gist.github.com/keannan5390/863c2072d1244fc3fde1>