* Dia 1: Hallo allemaal (Hallo meneer), vandaag gaan wij (Elkaar voorstellen) jullie/u onze applicatie wat toelichten.
* Dia 2: Waar gaan we het vandaag over hebben.

- We gaan het eerst hebben over wat de bedoeling is van de app.

- Daarna hebben we het over de opbouw applicatie met de belangrijkste elementen aangehaald in de code.

- En als laatste gaan we een demo geven van de werking van de app.

* Dia 3: We beginnen eerst met de werking van de applicatie.
* Dia 4: Splash screen: Bij het opstarten van de app krijgt de gebruiker een splash screen dat het logo van de organisatie is.
* Dia 5: Hoofd scherm: Op het front screen krijgt men een opsomming van de verschillende routes die zich in de database bevinden.

Refresh knop om de pagina te refreshen als er een update is geweest in de database.

Ook is er een about knop op de voorpagina die een aparte pagina opent.

* Dia 6: Over scherm: Het over scherm waar meer info opstaat hoe je de app moet gebruiken zodat men het niet zelf hoeft uit te leggen aan de gebruiker.
* Dia 7: Route scherm: In het midden van het scherm bevindt zich de route naam en een knop waar de gebruiker de route kan starten.

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

Klik;

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

* Dia 8: Content Screens: Als een gebruiker een bepaalde route gestart heeft, gaat de applicatie opzoek naar de beacon die zich het kortst bij de gebruiker bevindt.

Eenmaal als de gebruiker kort bij een beacon komt wordt dit opgevangen door de applicatie en wordt de media getoond.

De media die getoond kan worden is: Audio, Html, Foto, YouTube en Video.

In een korte video laten we de verschillende inhoud zien.

Als eerste heb je Audio die kan worden afgespeeld op het apparaat, Html wordt getoont zoals Html getoond moet worden, ook foto’s worden getoont op het apperaat en als laatste kunnen ook video’s / YouTube getoond wordt die ook mee draaien met scherm rotatie.

* Dia 9: Als volgende gaan we het hebben over de opbouw van de Applicatie.
* Dia 10: App.xaml.cs: We gaan gebruik maken van dependency injection waarbij we voor zorgen dat de verschillende klassen met elkaar verbonden zijn.

Hierdoor kunnen de verschillende klassen data uitwisselen zonder dat de relatie is vastgelegd in de broncode.

Als we de REST service zouden willen herschrijven kunnen we bijvoorbeeld een klasse bijmaken die RestService2 noemt en ook gebaseerd is op IRestService.

Indien we gebruik willen maken van RestService2 moeten we maar één ding veranderen in onze andere code.

Namelijk in de App.xaml.cs Container.RegisterType<IRestService, RestService2>() schrijven i.p.v. RestService.

* Dia 11: RestService: Met de klasse RestService gaan we de URL ophalen met de eventueel meegestuurde data.

De basis URL die we meegeven verwijst naar de URL van de API die in verbinding staat met de database.

GetRoutesAsync gaat al de routes opvragen uit de database.

Met GetBeaconsInRouteAsync verkrijgen we de data van al de beacons die zich bevinden in de route.

Met behulp van GetContentForBeaconInRoute kunnen we de data verkrijgen van een bepaalde beacon in een bepaalde route.

* Dia 12: GetRouteAsync: GetRoutesAsync gaat al de routes opvragen uit de database.

De link die gevormd wordt zal uiteindelijk deze worden: <http://api.beacons.ucll.be/v1/route>.

Als de gegevens correct zijn aangekomen gaan we deze omvormen naar json formaat zodat we deze kunnen terugsturen en hiermee bewerkingen kunnen doen.

De gegevens bevatten het route\_id en de naam van de route. Deze staan gedefinieerd in het Route model.

* Dia 13: GetBeaconsInRouteAsync: Met GetBeaconsInRouteAsync verkrijgen we de data van al de beacons die zich bevinden in de route.

Deze gegevens worden opgeroepen vanuit de database met het meegestuurde route id.

Als de gegevens correct zijn aangekomen gaan we deze omvormen naar json formaat zodat we deze kunnen terugsturen en hiermee bewerkingen kunnen doen.

De link die gevorm wordt zal uiteindelijk deze worden: [http://api.beacons.ucll.be/v1/route/"beacon\_id"/beacon](http://api.beacons.ucll.be/v1/route/1/beacon).

* Dia 14: GetContentForBeaconInRoute: Met behulp van GetContentForBeaconInRoute kunnen we de data verkrijgen van een bepaalde beacon in een bepaalde route.

Hierbij gaan we uit de database de content ophalen van de beacon (Html, YouTube, Audio, enz..).

De link die gevormd wordt zal uiteindelijk deze worden: [http://api.beacons.ucll.be/v1/beacon/"beacon\_id"/route/"route\_id"/dynamicData](http://api.beacons.ucll.be/v1/beacon/%22beacon_id%22/route/%22route_id%22/dynamicData).

* Dia 15: We gaan het nu hebben over de verschillende viewmodels.

Hieruit hebben we de belangrijkste delen uitgehaald die we dieper gaan bespreken.

De eerste die we gaan bespreken is de model van de hoofdpagina.

Vervolgens kijken we naar de Route info page view model. Deze geeft informatie weer over de route die je wilt gaan starten.

Als derde bekijken we de Beacon page view model. Deze geeft meer info waar je naar toe moet gaan en eenmaal als er een beacon kort genoeg is gaat deze over naar de beacon content page view model.

* Dia 16: De selectedRoute wordt opgeroepen eenmaal als een gebruiker op een route heeft geklikt. We verkrijgen de parameters van deze route (Route\_id en route naam).

Aan de parameters wordt het id van de geselecteerde route extra meegegeven.

Vervolgens roepen we Navigate op. Deze gaat al de parameters meegeven, inclusief de naam van de view die hij moet openen.

( await navigationService.NavigateAsync(page, navParams); ).

De gegevens worden in dit geval doorgestuurd naar het RouteInfoPageViewModel.

* Dia 17: De volgende methode gaat al de routes verkrijgen van de RestService.

Deze wordt ook opgeroepen als je de refresh knop gebruikt.

* Dia 18: Als een gebruiker navigeert van de hoofdpagina naar een route die men heeft gekozen roept men via navigationService de pagina RouteInfoPage op.

Door deze oproeping wordt de OnNavigatingTo opgeroepen.

De gegevens die worden meegegeven via de parameters van selectedRoute worden hier in Route\_id en Route\_info geplaatst.

* Dia 19: De gebruiker die een route wil starten drukt op de start knop.

Hierbij wordt de methode StartRoute opgeroepen.

Deze gaat de parameters ophalen van de route en doorgeven aan het BeaconsPageViewModel.

* Dia 20: Als een gebruiker op de start knop heeft gedrukt worden de parameters doorgegeven en in currentRoute gestoken.

Deze gaat een route\_id opvragen.

Vervolgens kijken we na als de gebruiker toestemmingen heeft gegeven of de applicatie zijn Bluetooth mag gebruiken.

* Dia 21: Heeft de gebruiker toestemming gegeven, kijken we na of bluetooth aanstaat.

Indien niet verschijnt er een error met de vraag om bluetooth aan te zetten en navigeren we terug naar de vorige pagina.

Eenmaal als de route gestart is krijgt de gebruiker op het scherm te zien waar hij naar toe moet gaan en start de app met het scannen van beacons.

De methode Onranged wordt gekoppeld aan het event Ranged van de estimote API.

* Dia 22: De methode wordt doorlopen zolang er beacons zijn.

Als de beacon gevonden is gaat de applicatie de gegevens meegeven via de parameters.

Deze parameters worden meegegeven aan de BeaconContentPageViewModel.

* Dia 23: De methode OnNavigatedTo wordt opgeroepen en gaat met behulp van de parameters de gegevens ophalen.

Als de gegevens opgehaald zijn gaan we nakijken wat voor soort metatype het gegeven is.

Aan de hand van het soort type geven we mee welke zichtbaar mag gemaakt worden op de view.

* Dia 24: Demo van de app geven. Via gsm of Via video als iets fout loopt.
* Dia 25: We zijn op het einde gekomen van onze presentatie. Bedankt voor jullie aandacht.
* Dia 26: Lege eind dia.