Industriële Wetenschappen en

Technologie

**Verslag Mobiele Apps 1 :**

Run2View

Tom Boelen

**Professionele Bachleor ICT**

Academiejaar 2014-2015

# Inhoudstabel

Inhoud

[Inhoudstabel 2](#_Toc420635770)

[Inleiding. 3](#_Toc420635771)

[Productomschrijving: 3](#_Toc420635772)

[Probleemstelling: 3](#_Toc420635773)

[Algemene opbouw van de applicatie: 4](#_Toc420635774)

[Probleem 1 : PushPin : MVVM 5](#_Toc420635775)

[1.1 Het toevoegen van PushPins 6](#_Toc420635776)

[1.2 Het laden van de pushpins 7](#_Toc420635777)

[Probleem 2 : GPS-markeringen 8](#_Toc420635778)

[Probleem 3: Scores bijhouden 9](#_Toc420635779)

[Probleem 4 : Run In The Background 10](#_Toc420635780)

[4.1 WMAppManifest.xml 10](#_Toc420635781)

[4.2 Windows Phone Application LifeCycle 10](#_Toc420635782)

[4.3 Oplossing 11](#_Toc420635783)

# Inleiding.

Een wereld zonder apps is tegenwoordig ondenkbaar. Van bijna elk merk , website en professionele bedrijven kan men tegenwoordig wel een applicatie via de play-store afhalen / kopen. Tijdens deze lessen kregen we de mogelijkheid om meer te weten te komen over WP8-apps. Om dan uiteindelijk zelf een applicatie te bouwen en in te dienen als project.

Omdat dit zo hedendaags is , is het natuurlijk voor de studenten zelf heel erg fijn om hiermee bezig te zijn en zal ik met veel vreugde mijn applicatie in dit verslag aan jullie voorstellen.

# Productomschrijving:

Run2View is een simpele run-tracking application. Men kan door te gaan lopen, meevolgen op je eigen gsm hoever en hoelang je al gelopen hebt. Bovendien was het voor mij enorm belangrijk dat wij ook een soort scorebord zouden toevoegen. Het werkt meestal enorm stimulerend om te kijken hoelang je gedaan hebt over de vorige run. Door dit scorebord bij je te hebben , ga je keer op keer je record willen verbruiken en dus ook steeds meer gaan lopen.

Een 3de toevoeging aan deze simpele app is de toevoeging van PushPins. Op elk moment dat je ergens aan het lopen bent en merkt dat dit een leuke locatie is om later terug te komen , kan je een pushpin toevoegen. Zo weet je later excact waar je deze locatie makkelijk kan terugvinden.

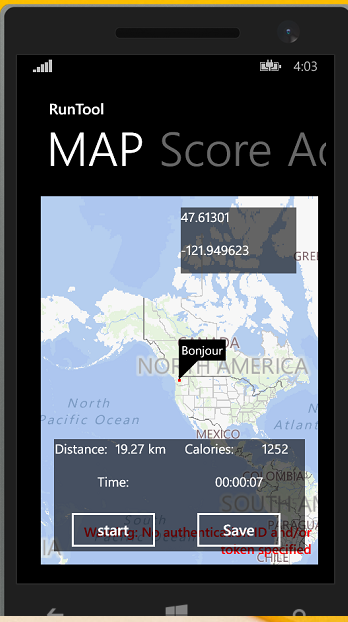
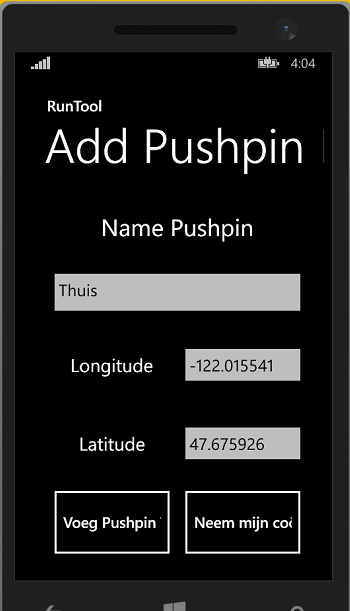
# Probleemstelling:

Voor ik ben gaan coderen , heb ik mezelf afgevraagd wat nodig was om dit project zeker tot een goed einde te brengen. Wat zijn de mogelijke problemen die zich kunnen voordoen tijdens het ontwikkelen van de applicatie ? En vooral , welke mogelijkheid hebben we om dit op te lossen?

* Hoe de app laten runnen op de background?
* Hoe scores van de loopsessies bijhouden?
* Hoe kunnen we pushpins toevoegen , gebruik makend van de MVVM-structuur?
* Hoe GPS-markeringen op een map aanbrengen?

# Algemene opbouw van de applicatie:

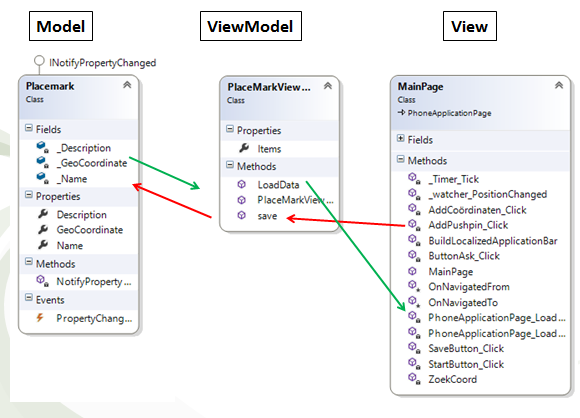
Voor de algemene opbouw van de hoofdpagina heb ik gekozen voor een PIVOT-structuur. Door de Pivot kan je heel makkelijk door je pagina heen sliden zonder een ander scherm te openen. Bovendien zal de tijd & het lokaliseren op de map gewoon doorgaan omdat je eigenlijk maar op één pagina aan het werken bent. Je moet dus niet alles globaal gaan maken wat veel handiger is om mee te werken.

# Probleem 1 : PushPin : MVVM

Bij het maken en produceren van de pushpins , hebben wij gebruik gemaakt van het MVVM - model. Dit model is van uiterste belang om een programma op een zeer overzichtelijke wijze te laten werken. Ondanks er in het begin van dit project een URL is rond gegaan om pushpins puur in de hoofdcode te coderen , leek het mij juist belangrijk om dit naar een MVVM-structuur om te gooien.

Hieronder zie je heel simpel het proces wat er gebeurt als je een PushPin wil toevoegen (De rode pijlen) of wil ophalen (De Groene pijlen). Door bijvoorbeeld te klikken op de button AddPushPin zal je een Placemark met de eigenschappen Name, Description & GeoCoordinate toevoegen aan de observableCollection die opgeslaan wordt in het PlaceMarkViewModel.

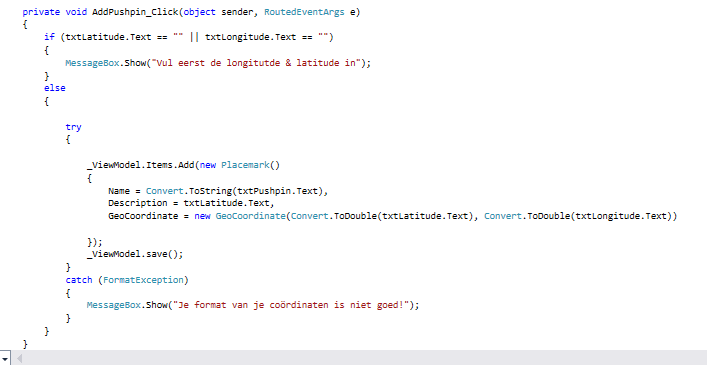
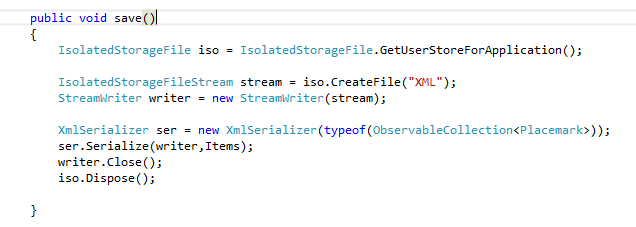


## 1.1 Het toevoegen van PushPins

Op het moment dat de textboxen zijn ingevuld en men klikt op de button Add PushPin , zal direct het event AddPushPin\_Click worden afgevuurd.

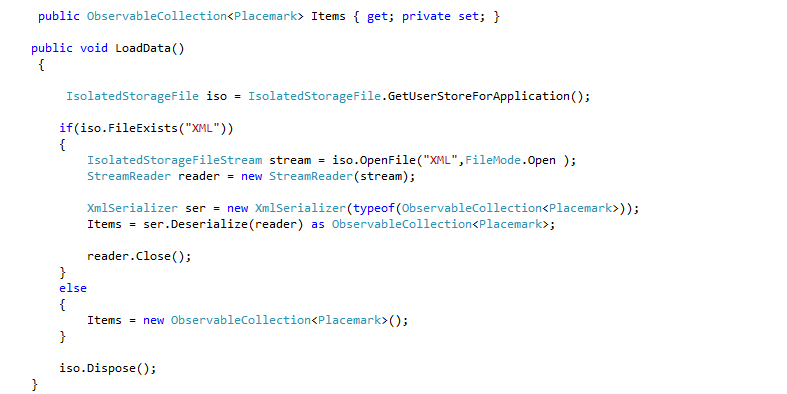
Zoals je hieronder ook op de afbeelding ziet , zal op dat moment een nieuw object Placemark worden aangemaakt. In dit object worden de 3 benodigde eigenschappen (de 3 textboxen die je moet invullen) gestoken om een volwaardig object van het type Placemark te verkrijgen.

Heel belangrijk om te weten is eigenlijk éénmaal als je dit object hebt aangemaakt , dat dit object onmiddellijk in ObservableCollection gestoken wordt. Deze ObservableCollection (een lijst met allemaal objecten van Placemark) gebruikt men in het ViewModel om alle placemarks tesamen op te slaan in isolated storage. (figuur 2). Daar zie je heel duidelijk dat de ObservableCollection<Placemark> opgeslaan wordt als een XML-bestand om later opnieuw in te laden en te tonen op de map.



## 1.2 Het laden van de pushpins

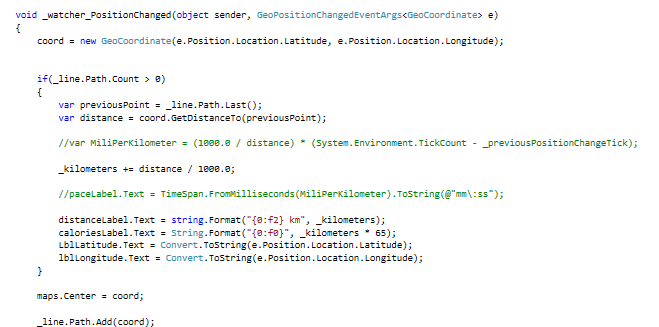
Dit deel is eigenlijk ongeveer hetzelfde als het vorige. Met het kleine verschil dat men nu de observableCollection van placemarks niet gaat opslaan , maar gaat ophalen uit een XML-bestand.



Eens als het XML – bestand is ingeladen en is omgezet naar een ObservableCollection van Placemarks. Zal deze ObservableCollection gebruikt worden om Placemarks op te kaart te zetten.  
Hier onder zie je dat ze dus eerst de data inlaaden en daarna de observableCollection (items) gebruiken om toe te voegen aan de .ItemsSource van de map. Hierdoor zullen de ingelezen PushPins zichtbaar worden op de map.

# Probleem 2 : GPS-markeringen

Eenmaal als we de map hadden toegevoegd , moesten we bekijken hoe we GPS-markeringen zichtbaar konden maken op de applicatie zelf. En niet alleen zichtbaar , de GSM moet je volgen als je aan het lopen bent en bovendien ook nog eens de afstand berekenen.

Na lang zoeken , ben ik eigenlijk op een relatief makkelijk oplossing uitgekomen.

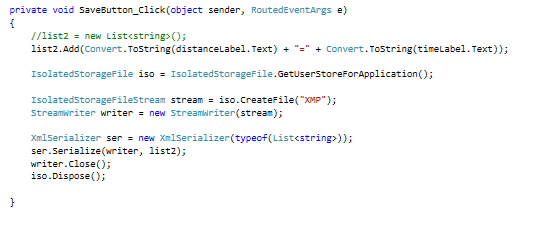
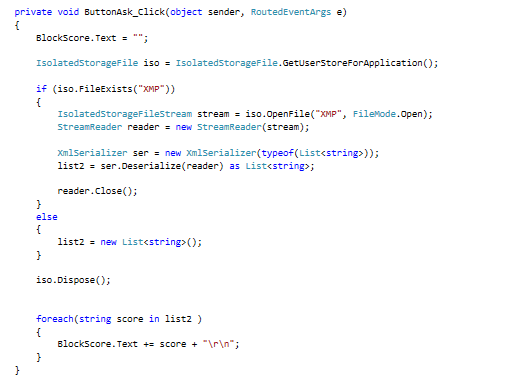
In dit stukje code maakt men eigenlijk gebruik van het event PositionChanged. Telkens als de positie veranderd , gaat men bij de \_line een puntje toevoegen. Dit puntje staat natuurlijk voor de coördinaten waar je op dat moment staat. Als je al deze puntjes verbind met mekaar , dan krijg je uiteindelijk een lijn die je gevolgd heeft van waar je gestart bent , tot waar je geëindigd bent.

Het zinnetje erboven (Maps.center = coord) lost dan weer een ander probleem op. Door bij elke ‘PositionChange’ dit regeltje uit te voeren , gaat de kaart zich centreren op de plaats waar jij dan juist staat. Loop je 10 kilometer verder? De map zal je nog steeds volgend tot en met daar.

Als allerlaatste wil ik het even hebben over het bereken van de gegevens \_kilometers & \_calories. Op het moment dat je beweegt , gaat er in de IF functie van \_line.Path.Count > 0 vanalles berekend worden. Zo gaat men eerst het laatste punt van de \_line eruit halen. En met dat laatste punt gaan ze dan de coördinaten die je nu heb vergelijken. Aan de hand van die verschillenen kan men de afstand berekenen die men heeft afgelegd. En dus uiteindelijk ook de calorieën die voor een erg groot deel samenhangen.

# Probleem 3: Scores bijhouden

Voor ik begon met het coderen , leek dit een probleem te zijn.Het opslaan moet immers gebeuren met isolated-storage , en ik was nog niet helemaal mee hoe dit werkte. Omdat er reeds bij de PushPins gebruik werd gemaakt van Isolated Storage , moest ik hier ongeveer hetzelfde doen als bij dat stukje code. Dit zorgde dus niet meer voor grote problemen omdat het dan vooral overnemen was van een vorig stuk codetaal.

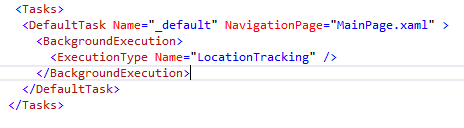


# Probleem 4 : Run In The Background

Een running – app zou eigenlijk nooit mogen stoppen. Op het moment dat je Applicatie nog aanstaat en je GSM gaat op standby , moet nog steeds de locatie en dus afgelegde weg nog worden gemeten. Er zijn 2 omslachtige manieren om er voor te zorgen dat dit ook kan gebeuren.

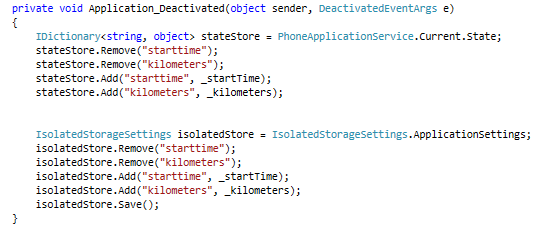
## 4.1 WMAppManifest.xml

Door onderstaande code aan het WMAppManifest toe te voegen , ga je ervoor zorgen de applicatie op de achtergrond blijft runnen. In TombStone gaat de applicatie door deze regels ook niet komen , juist omdat het moet blijven draaien (zeer belangrijk!)



## 4.2 Windows Phone Application LifeCycle

De 2de manier om dit te doen werken , is door gebruik te maken van WP Application LifeCycle.  
Door deze manier te gebruiken , ga je niet de applicatie op de achtergrond laten runnen , maar ga je de afgelegde weg en tijd opslaan juist voor je GSM in DORMANT of TOMBSTONED – modus komt.

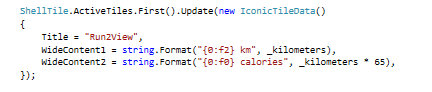
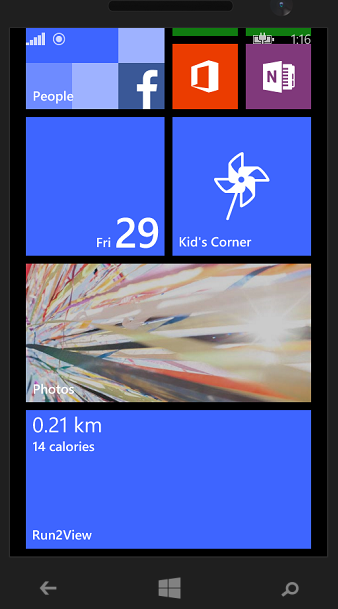
Op het moment dat je je applicatie terug opstart , gebruik je de OnNavigatedTo – method & het Application Activated Event om het laatste opgeslagen tijdstip en afgelegde weg terug op te halen. Deze worden dan vergeleken met het huidige tijdstip / locatie waardoor een nieuwe tijd / afgelegde berekend kan worden.[[1]](#footnote-1)

## 4.3 Oplossing

Als oplossing heb ik gekozen voor een combinatie van de 2. Het is erg belangrijk dat de applicatie gewoon op de achtergrond kan blijven draaien. Maar moest het op één of andere manier toch in Dormant of Tombstoned komen , dan kan het nieuwe meetgegevens berekenen aan de hand van de oude (die opgeslaan zitten in isolated storage).

Ik ben er van bewust dat beide technieken nadelen hebben. De eerste techniek gaat natuurlijk enorm veel energie in beslag nemen bij je GSM. Terwijl de 2de techniek niet zo accuraat is en momenteel in de huidige applicatie de MapPolyLine dan nog niet onthoudt.

Door gebruik te maken van beide , kan je heel gemakkelijk gebruik makken van een Actieve Tegel op je GSM-pagina. Zo zal , omdat het programma op de achtergrond aan het draaien is , de afstand en de calories mee veranderen op jeGSMpagina als het event PositionChanged afgeschoten is.

**

1. [↑](#footnote-ref-1)