Mezelf voorbereiden voor mobiele apps voor dummys:  
  
Welkom idiootjes!

Bij het vak mobiele apps gaan we apps bouwen met behulp van Xamarin.  
  
Wat is dat? Xamerin is microsoft hun gratis en open source platform dat je kan gebruiken om met behulp C# en het .NET Framework apps te bouwen speciaal voor IOS, Android of Universal Windows Platform(UWP).

Cool! Is dat wat wij gaan gebruiken? Hahaha Nee!  
Heb je die “of” soms gemist?  
Als we gewoon standaard Xamarin zouden gebruiken kunnen we de logica van onze app wel universeel maken maar moeten we nog steeds een aparte UI maken voor elk platform en dat is dubbel werk.

Wat gebruiken wij dan wel?  
Wij gebruiken Xamarin Forms. Dit is een universele UI bibliotheek van Microsoft. Hiermee kunnen we een keer onze UI maken. Als we dan gaan compileren zal Xamarin deze UI automatisch aanpassen aan de verschillende huisstijlen van elk platform.  
  
Dit klinkt geweldig hoe begin ik hieraan?  
  
Hieronder staat beschreven hoe je Xamarin moet installeren en doorlopen we een demo projectje om te laten zien hoe alles werkt.

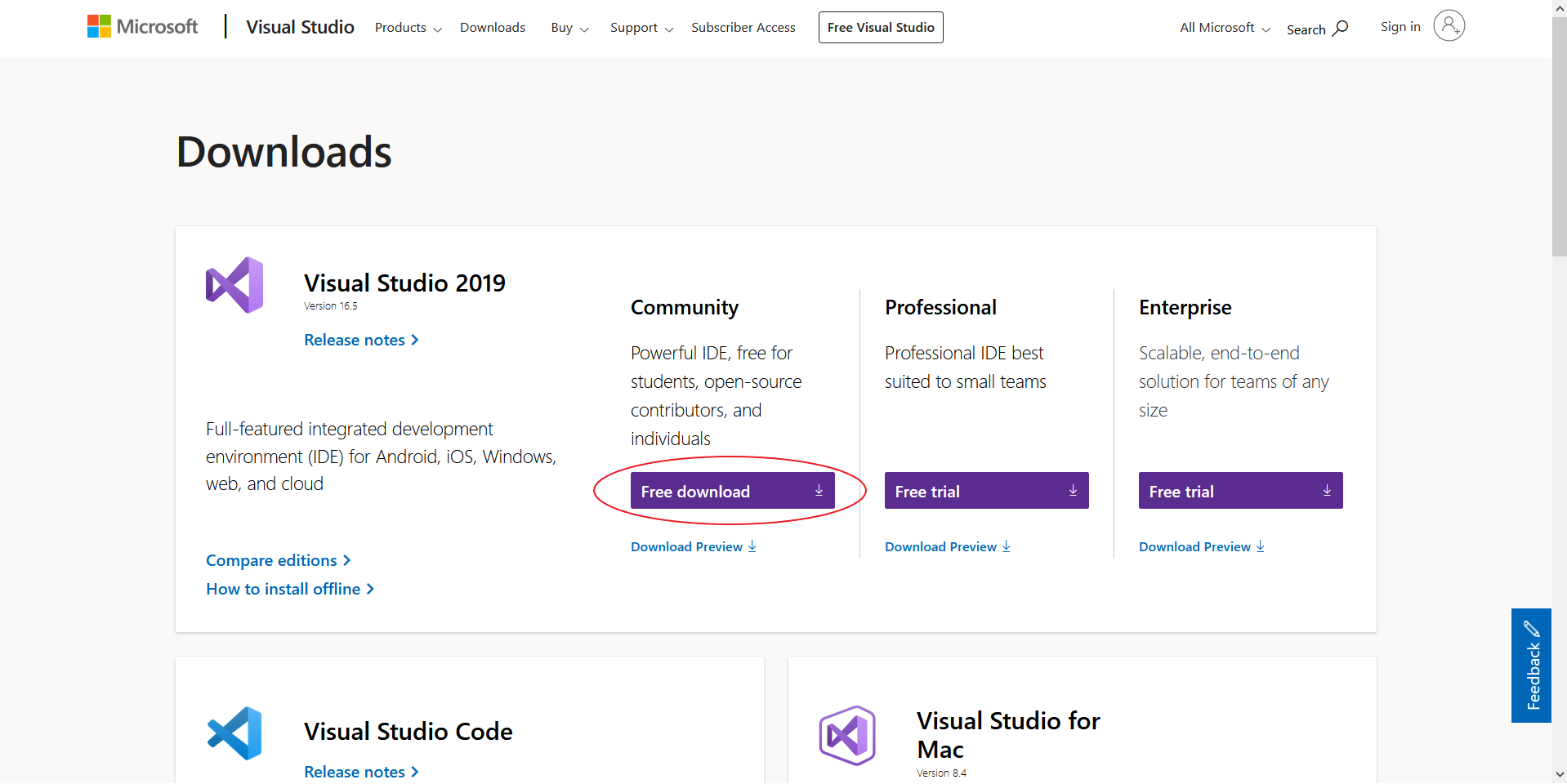
Volg gewoon deze stappen:

# Stap 1: Bestuuring systeem

Zorg dat je een computer gebruikt met Windows of Mac OS.  
Sorry Linux mensen.  
 

# Stap 2: IDE

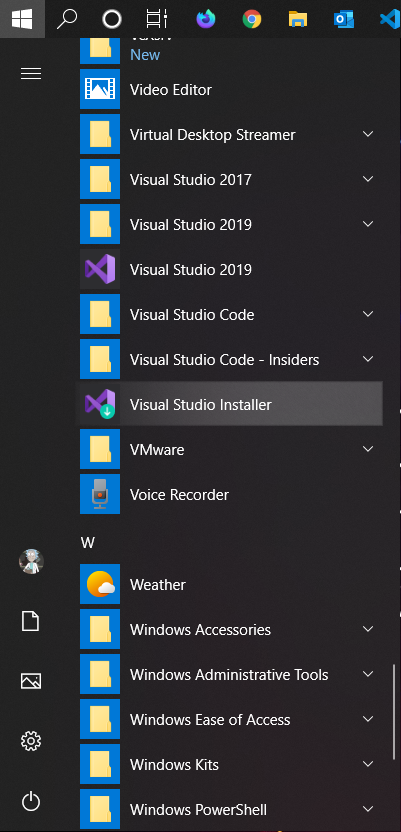
Hoewel er meerdere editors zijn voor C# is er maar eentje echt geschikt voor Xamarin en dat is Visual Studio.  
  
Je gebruikt best de meest recente Community versie, momenteel 2019  
Link: <https://visualstudio.microsoft.com/downloads/>

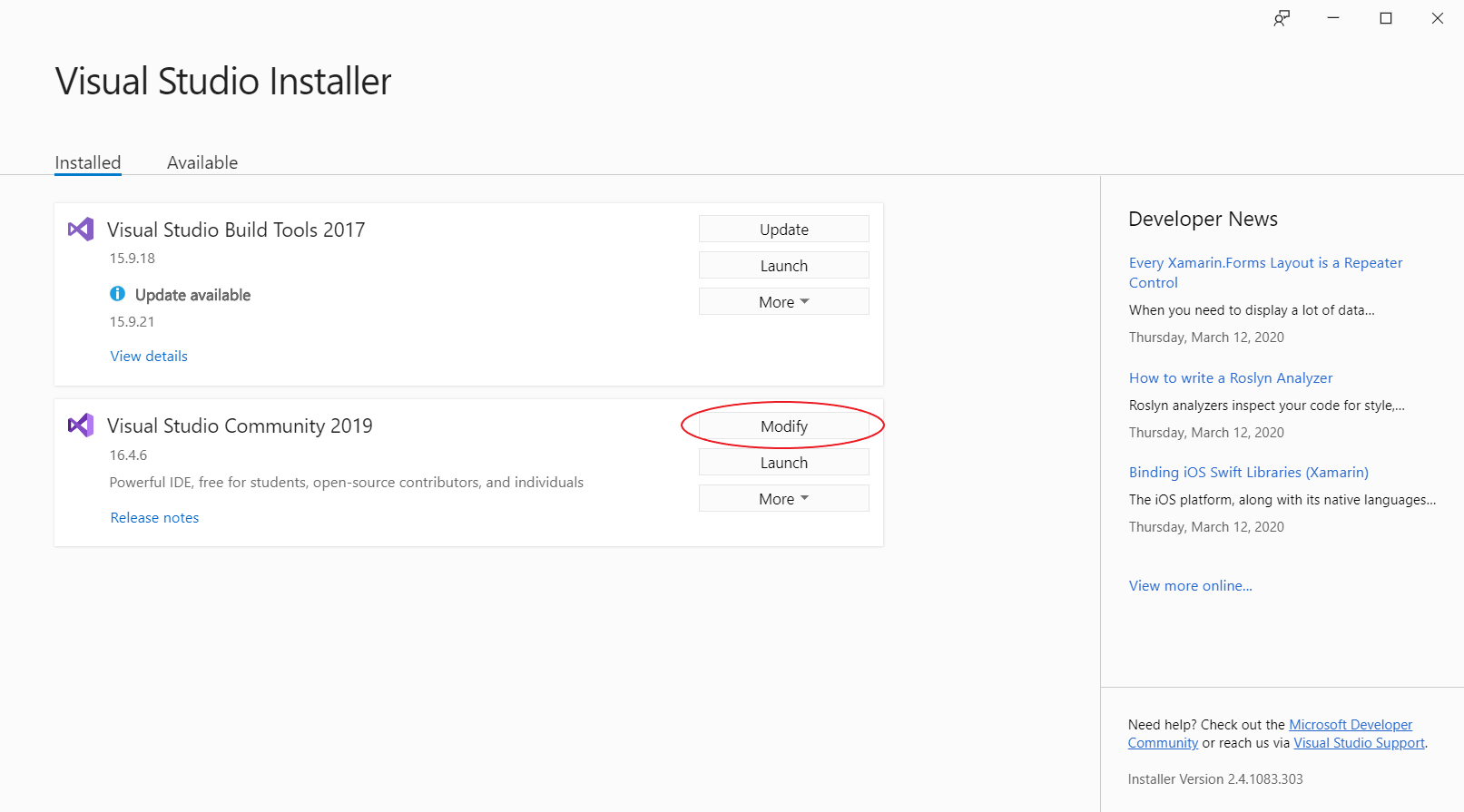


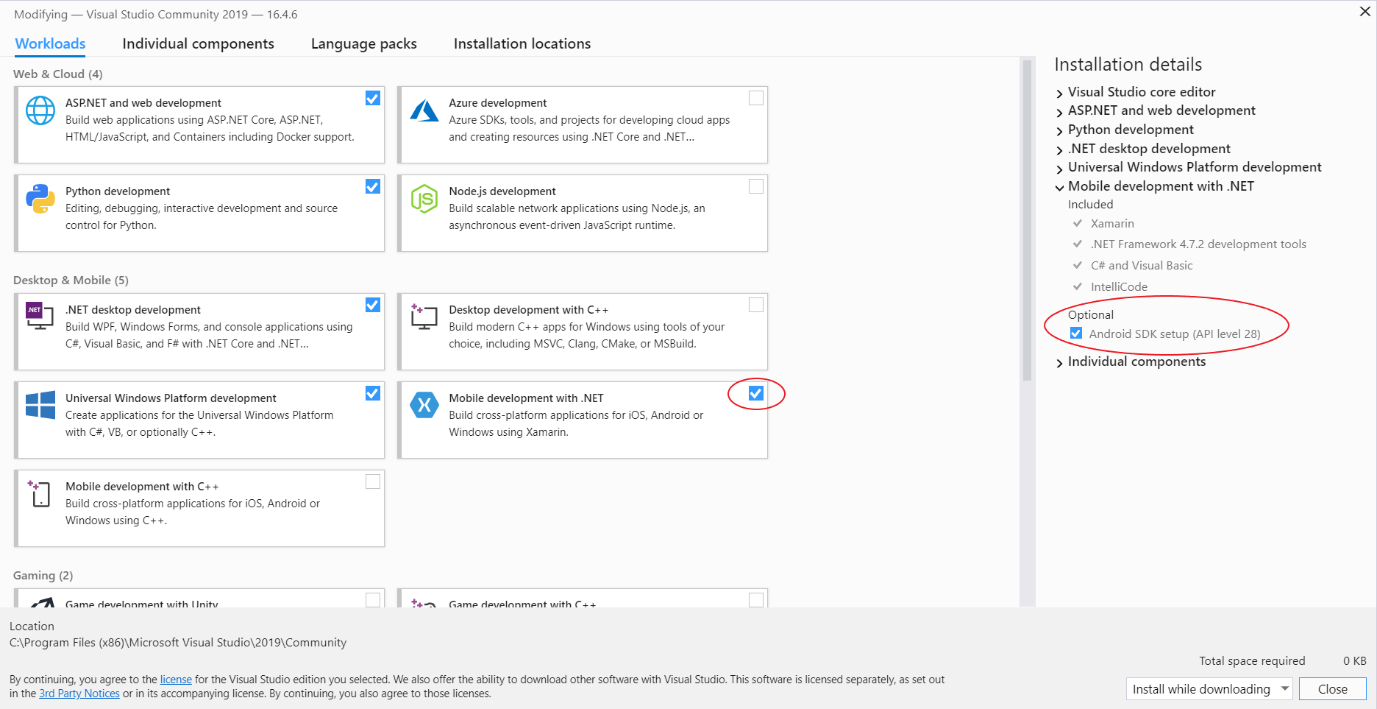
Stap 3: Xamarin support toevoegen

Visual studio heeft standaard geen Xamarin Support dit moet je toevoegen bij het installeren of achteraf met behulp van de Visual Studio Installer.  
  
3.1 Visual studio installer

Ga naar je start menu en zoek naar de Visual Studio Installer. (Als je deze niet ziet raadpleeg google.)

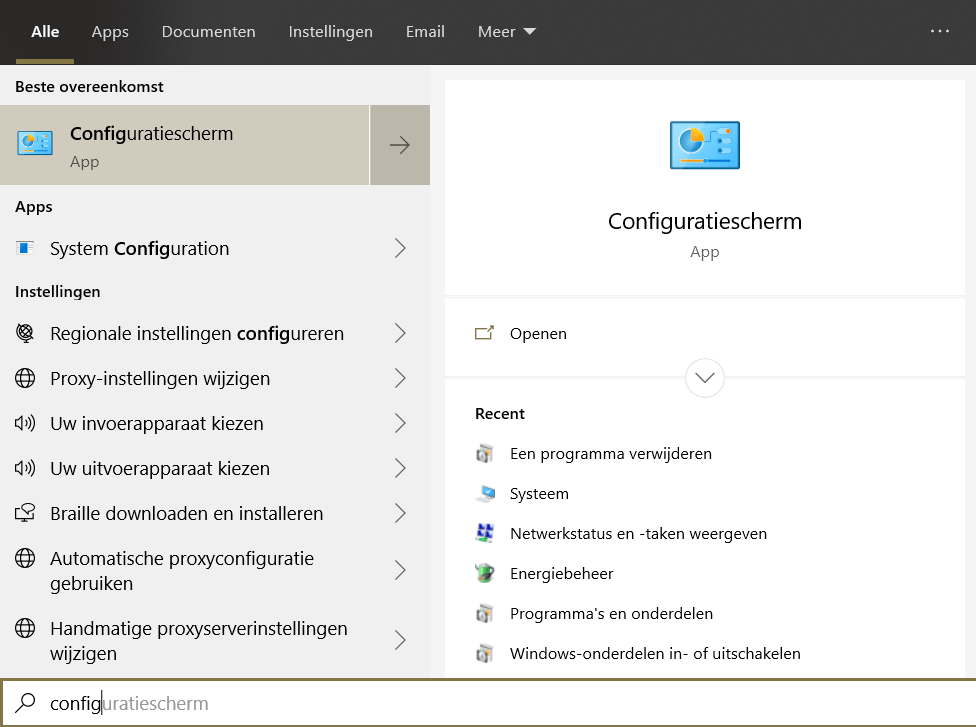


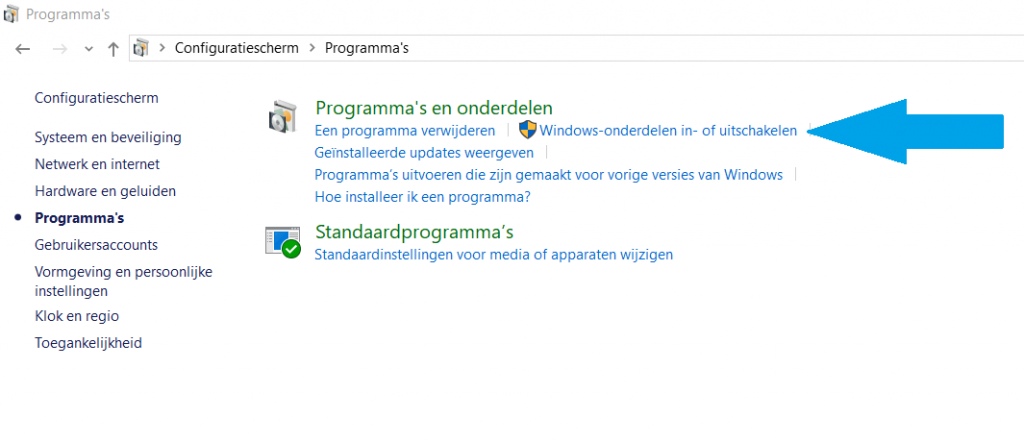
Als je deze geopend hebt gaat deze waarschijnlijk eerst vragen om de installer zelf te updaten. Als je deze geopend hebt kan je de verschillende versies van Visual Studio zien die je geinstalleerd hebt. Wij zijn geintresseerd in   
“Visual Studio Community 2019”  
  
Om Xamarin support toe te voegen moeten we Visual Studio aanpassen. Dit doen we door op “Modify” te klikken. (Als er een update beschikbaar is zal je Visual Studio eerst moeten updaten voordat je kan modifyen)

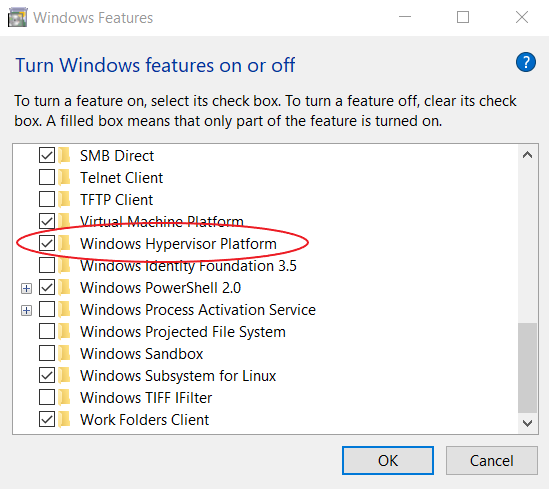
In de installer moet je nu “Mobile App development with .NET” aanvinken.  
Zorg dat rechts ook de “Android SDK Setup” aangevinkt is.  
  
 Klik vervolgens rechtsonder op “Modify” om de installatie te starten.

## 3.2 Windows Hyper Visor Platform Installeren

Dit onderdeel is enkel voor Windows gebruikers.

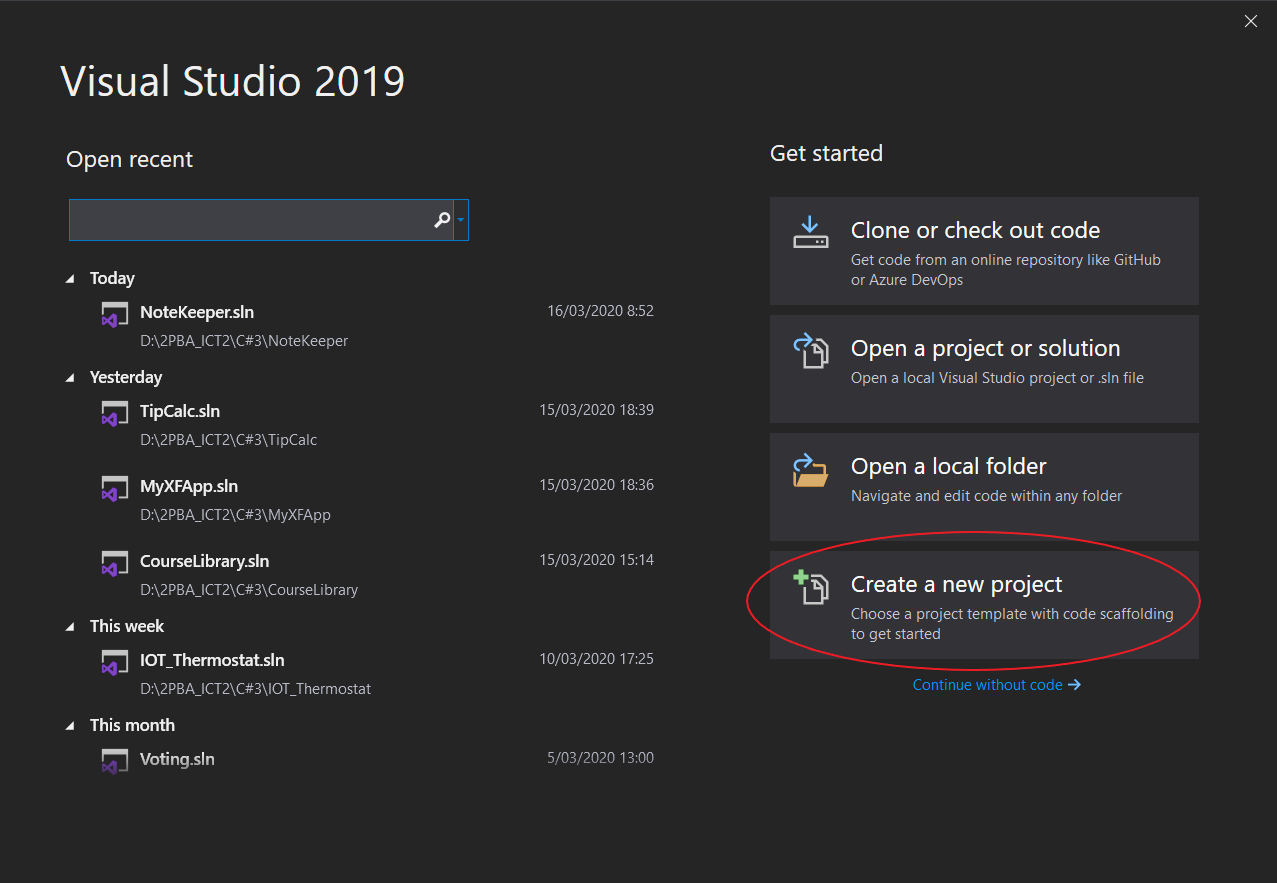
Om de Android versies van onze App te kunnen testen en debuggen gaat Visual Studio voor ons een Android smartphone emuleren. Hiervoor maakt het gebruik van het “Windows Hyper Visor Platform”. Een virtualiserings platform dat je kan vergelijken met VM Ware of Virtual Box. Dit is een optioneel Windows onderdeel dat we nog even moeten installeren.  
  
Ga naar start en zoek naar “Configuratiescherm”, open dit.  
  
  
Klik vervolgens op “Programma’s”

Klik vervolgens op “Windows Onderdelen In- of uitschakelen”  


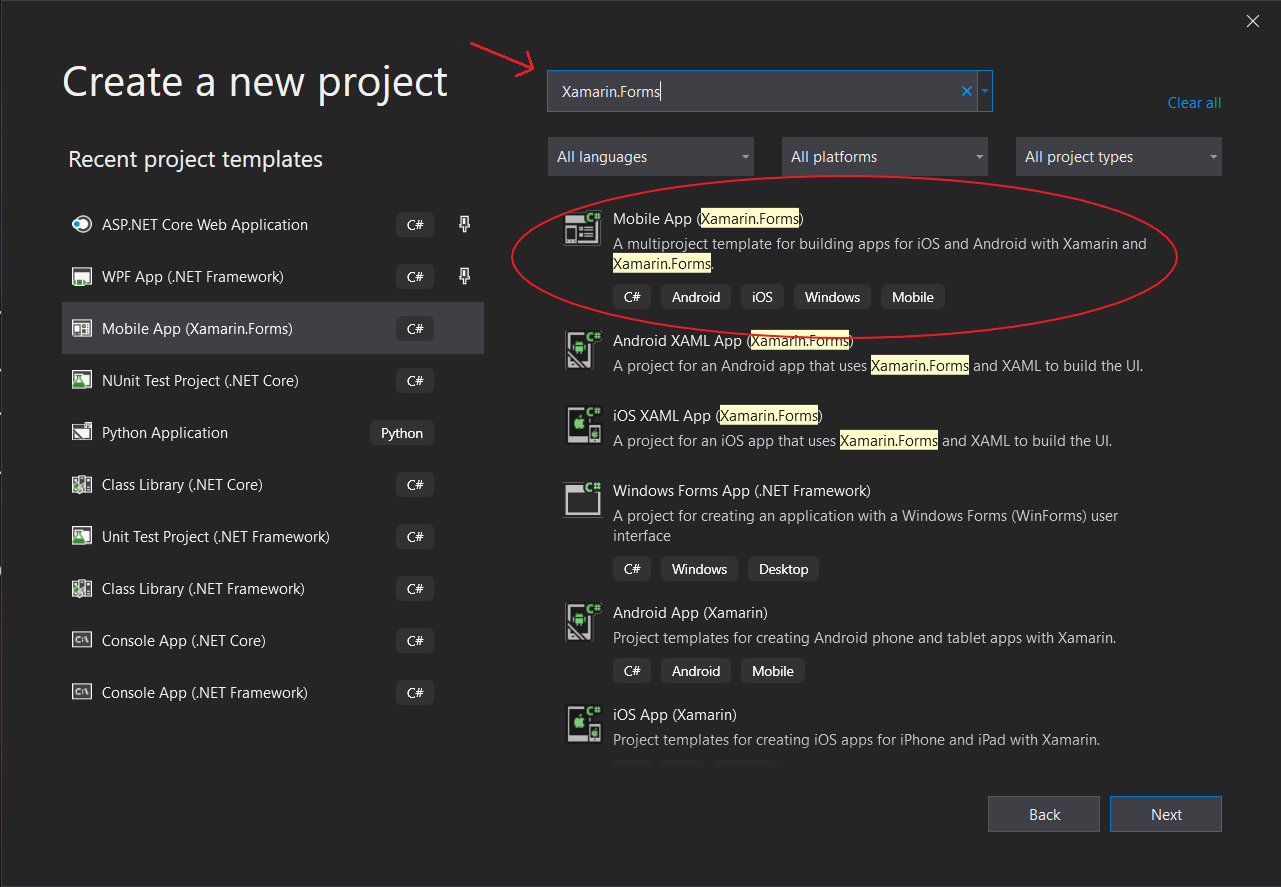
Vink hier het “Windows Hyper Visor Platform” aan.  
  
  
Klik op OK en je PC zal de installatie op de achtergrond beginnen.  
Wanneer de installatie klaar zal Windows je waarschijnlijk vertellen dat alles klaar is, maar om het te laten werken moet je toch eerst je PC herstarten.

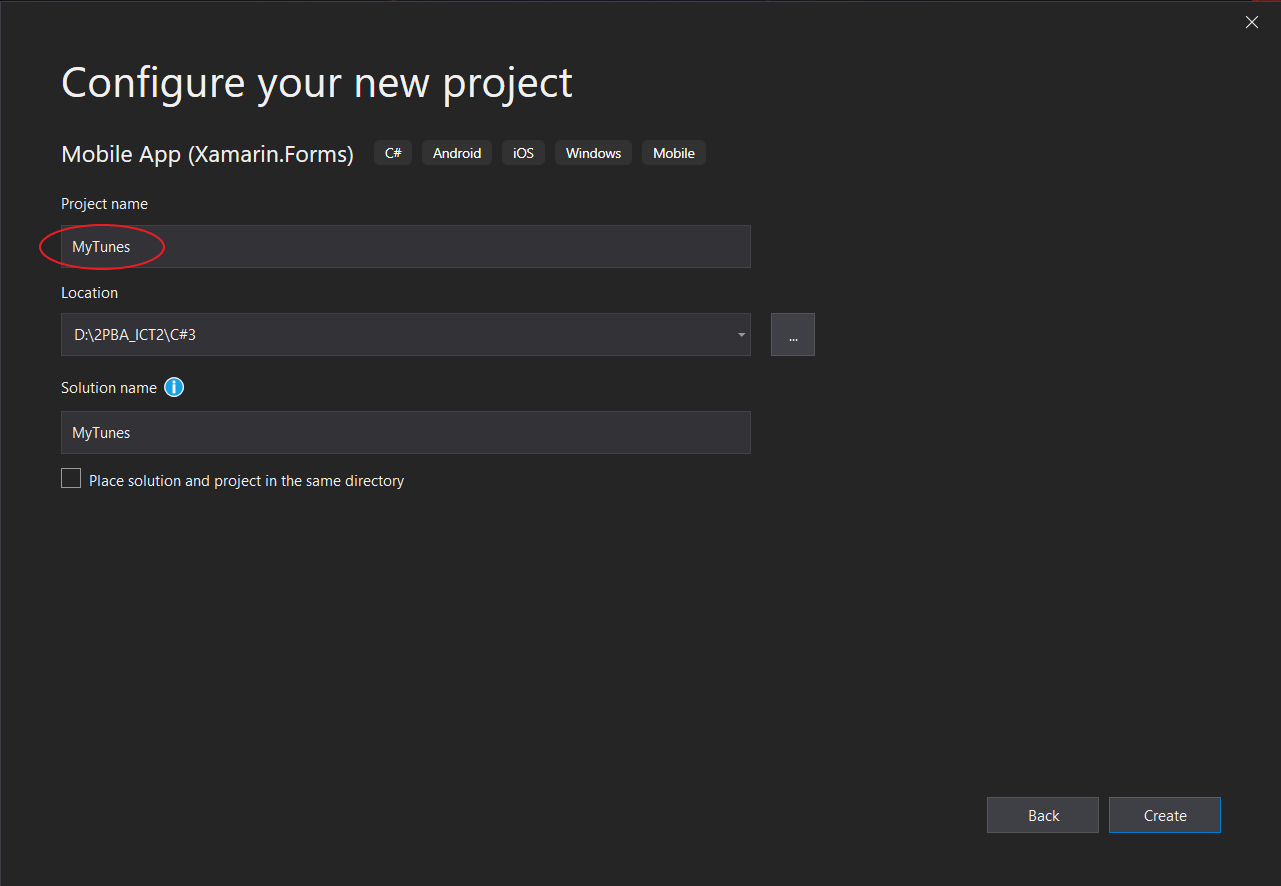
Stap 4: Het voorbereiden van de Android Emulator

Hoewel je nu alle nodige onderdelen geinstalleerd hebt en klaar bent voor de les mobiele apps, is het toch een goed idee om een eerste Xamarin Project aan te maken. Hierdoor kan je bekend kan worden met Xamarin en kan je de Android Emulator al initialiseren aangezien dit de eerste keer behoorlijk wat tijd in beslag neemt.  
  
4.1 Xamarin project aanmaken

Ga naar je start menu en open Visual Studio. Begin een nieuw project.  


Selecteer nu een “Mobile App (Xamarin.Forms)” project door bovenaan naar “Xamarin.Forms” te zoeken. En klik op “Next”.

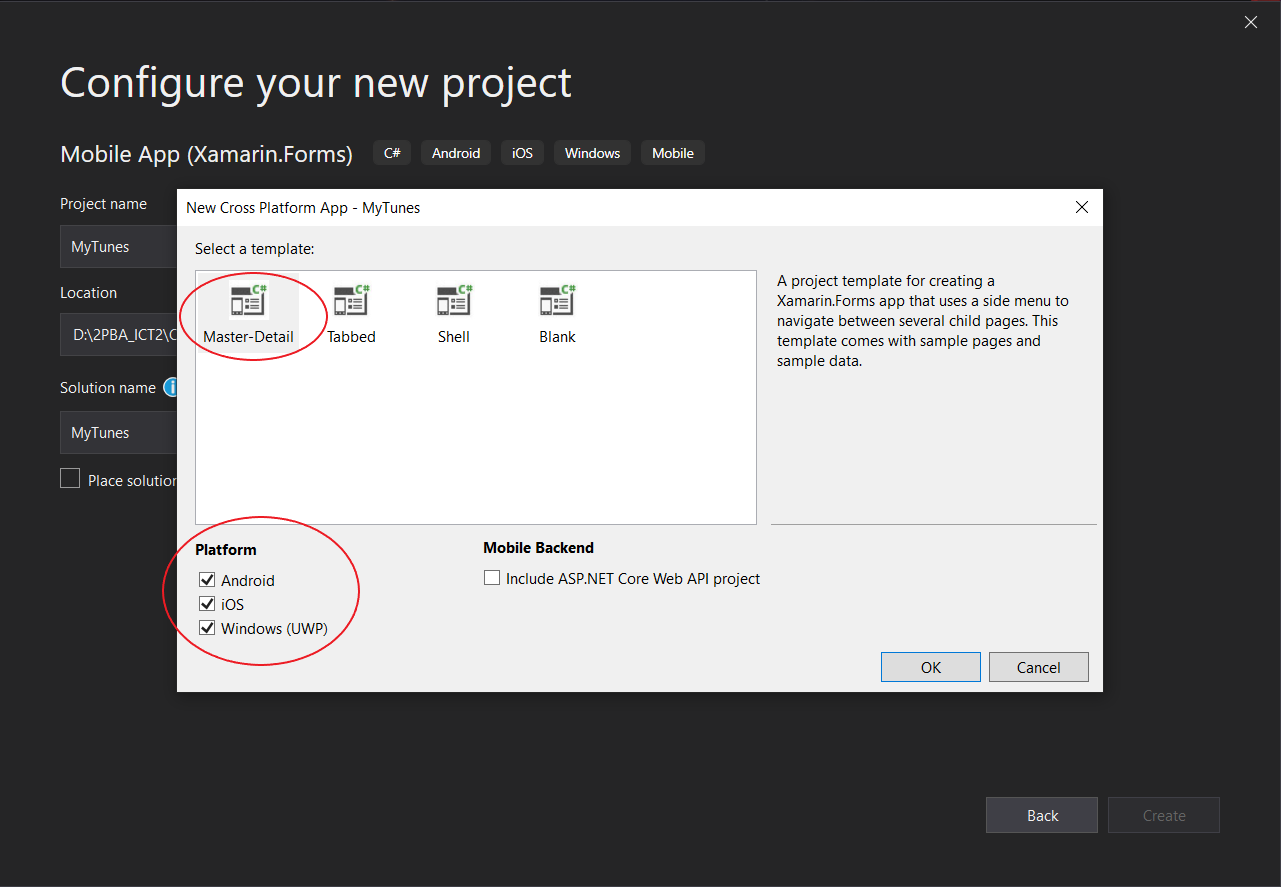


Vervolgens geven we onze app een originele naam. En klikken we op “Create”.  


Vervolgens gaat Visual Studio ons vragen welke App template we willen gebruiken en welke platformen we willen ondersteunen.

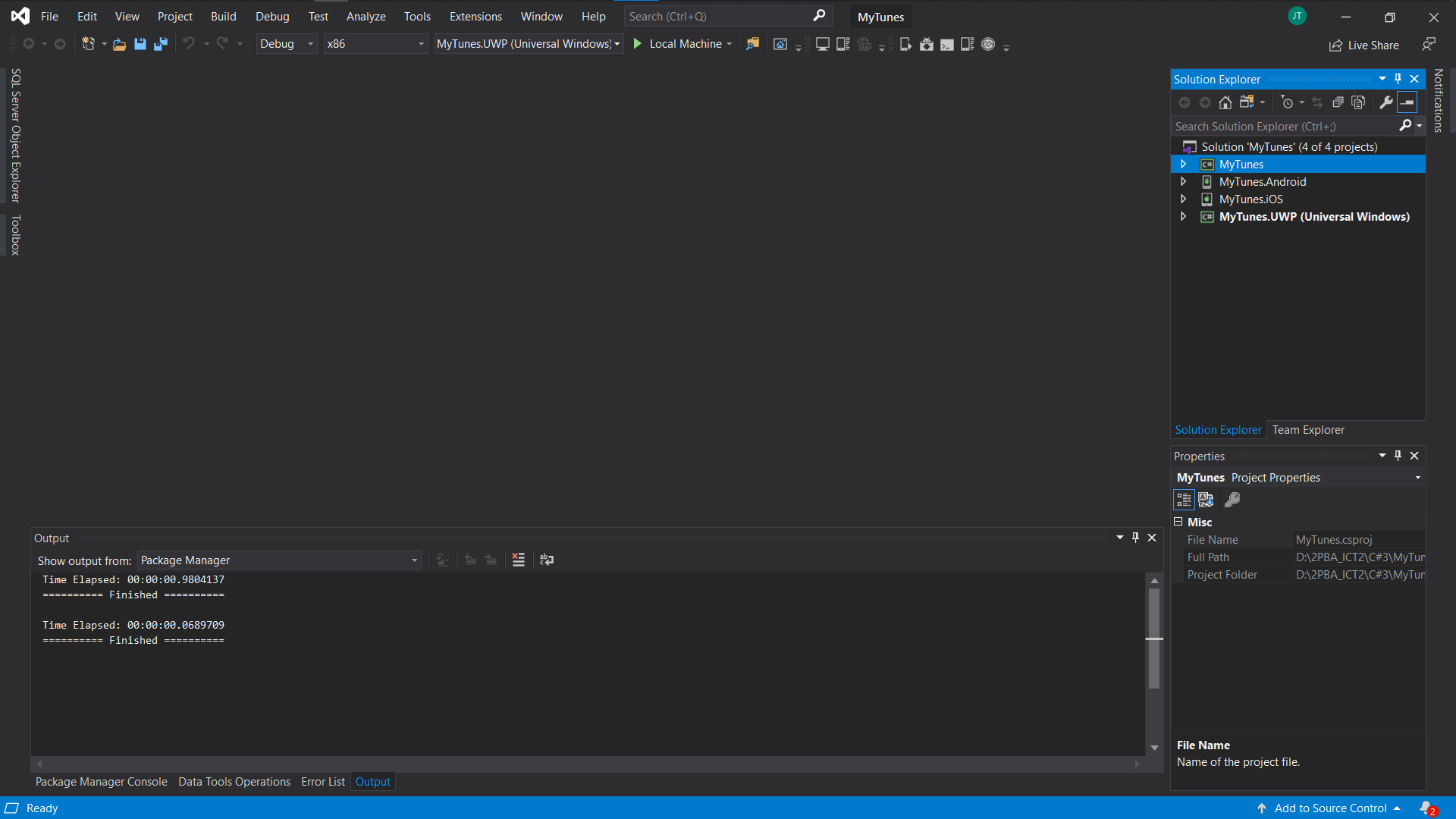
In de les beginnen we meestal met een “Blank” template dat leeg is maar voor de demo kiezen wij nu even voor een “Master detail” template. Hierdoor krijgen we een basic boodschappenlijstjes app cadeau.

Vink onderaan ook de platformen aan die je wilt ondersteunen. In de les werken we meestal enkel met Android maar voor de demo kiezen wij nu even voor allemaal.



4.2 Project overlopen

Als Visual Studio ons project aangemaakt heeft zien we het volgende:

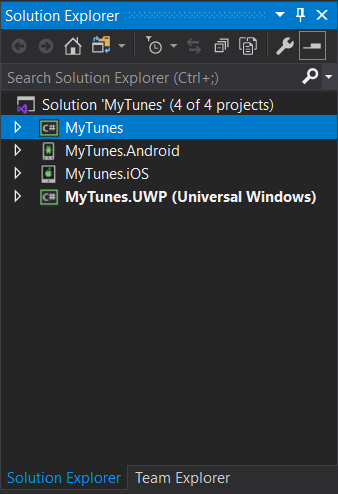


In onze “Solution Explorer” zien we dat Visual Studio een “Solution” heeft aangemaakt met daarin 4 projecten.

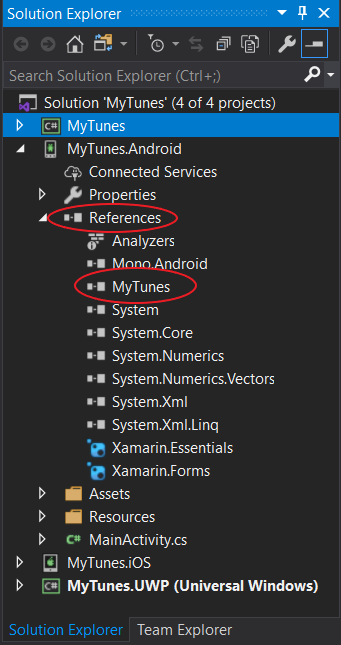
Het project “MyTunes” bevat onze App Code en UI (Xamarin Forms).

Het project “Mytunes.Android” bevat de Android implementatie van onze app.

Hetzelfde voor “MyTunes.IOS” en “Mytunes.UWP”.



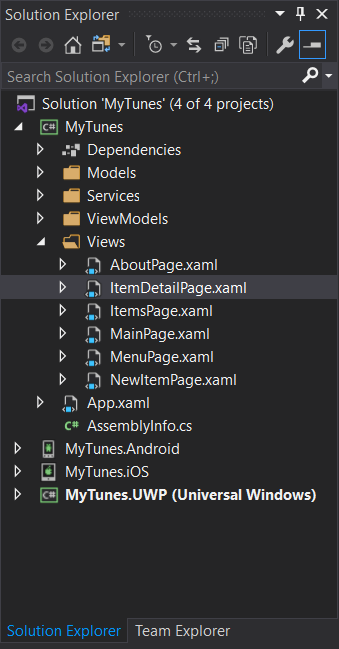
Als we ons Android project openvouwen vinden we bij de “References” ons “MyTunes” project terug. In het project vinden we ook een “MainActivity.cs” file terug. Dit is de Android implementatie van onze app die gewoon ons “MyTunes” project oproept. Hetzelfde voor ons IOS en UWP project.



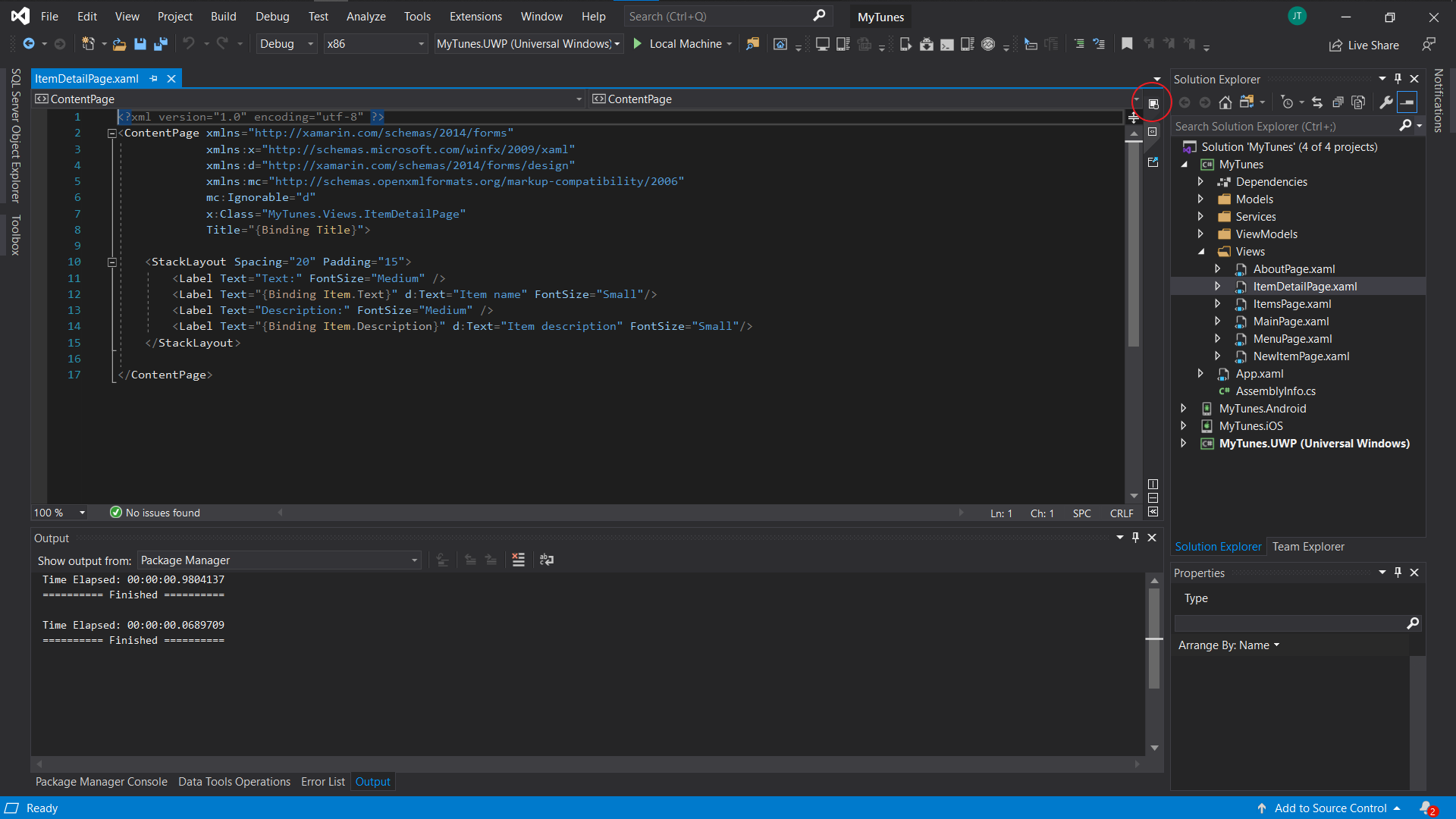
Als we terugkeren naar ons “MyTunes” project zien we dat er een hoofd xaml pagina is genaamd “App.xaml”. Hieruit vertrekt onze app. Er is ook een mapje “Views” aanwezig waarop de verschillende pagina’s van onze app staan.

Vanwege onze “Master-Detail” template die we geselecteerd hadden hebben we een basic app gekregen die lijkt op een boodschappenlijstje.

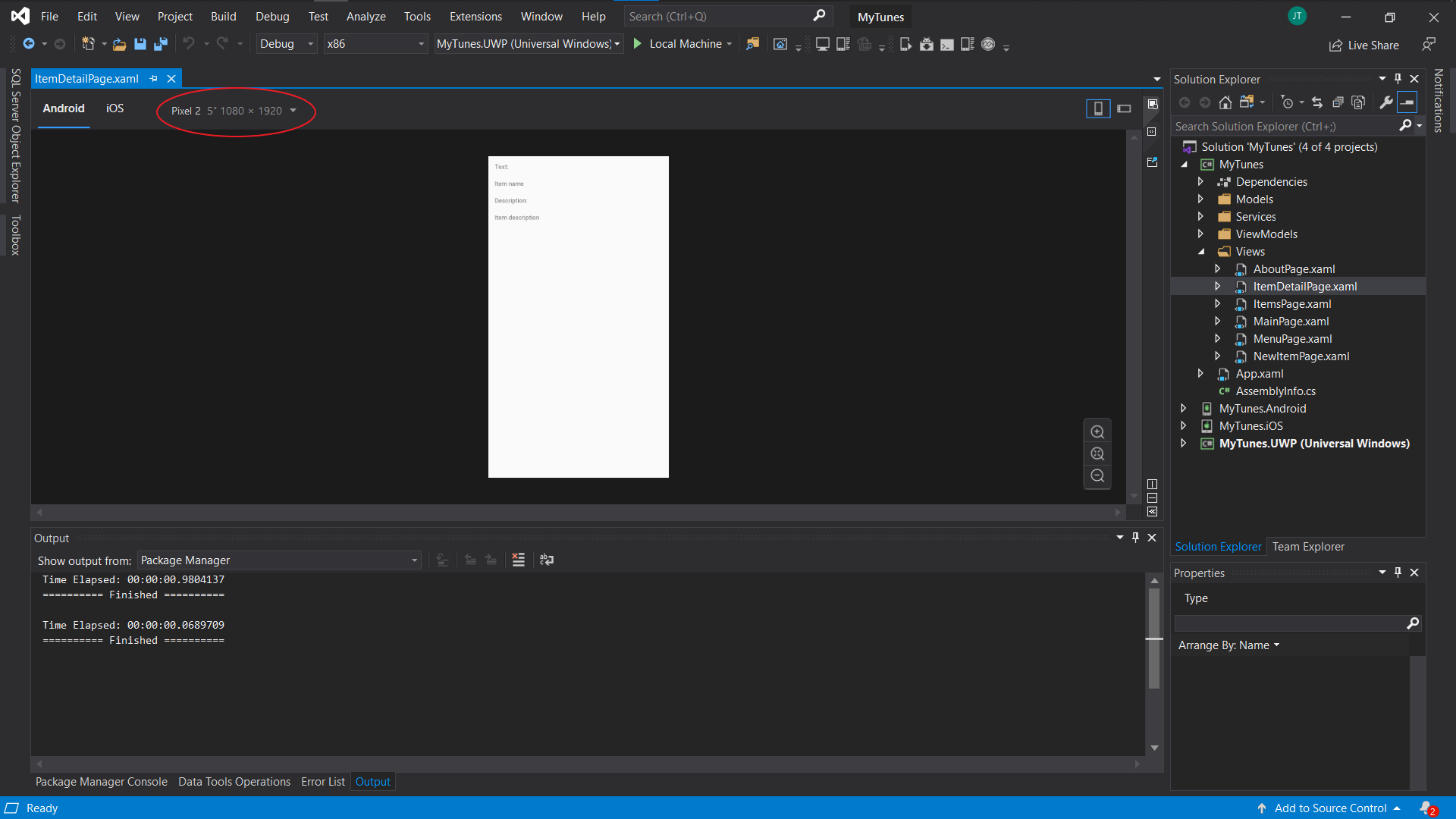
Open de “ItemDetailPage” even als voorbeeld.

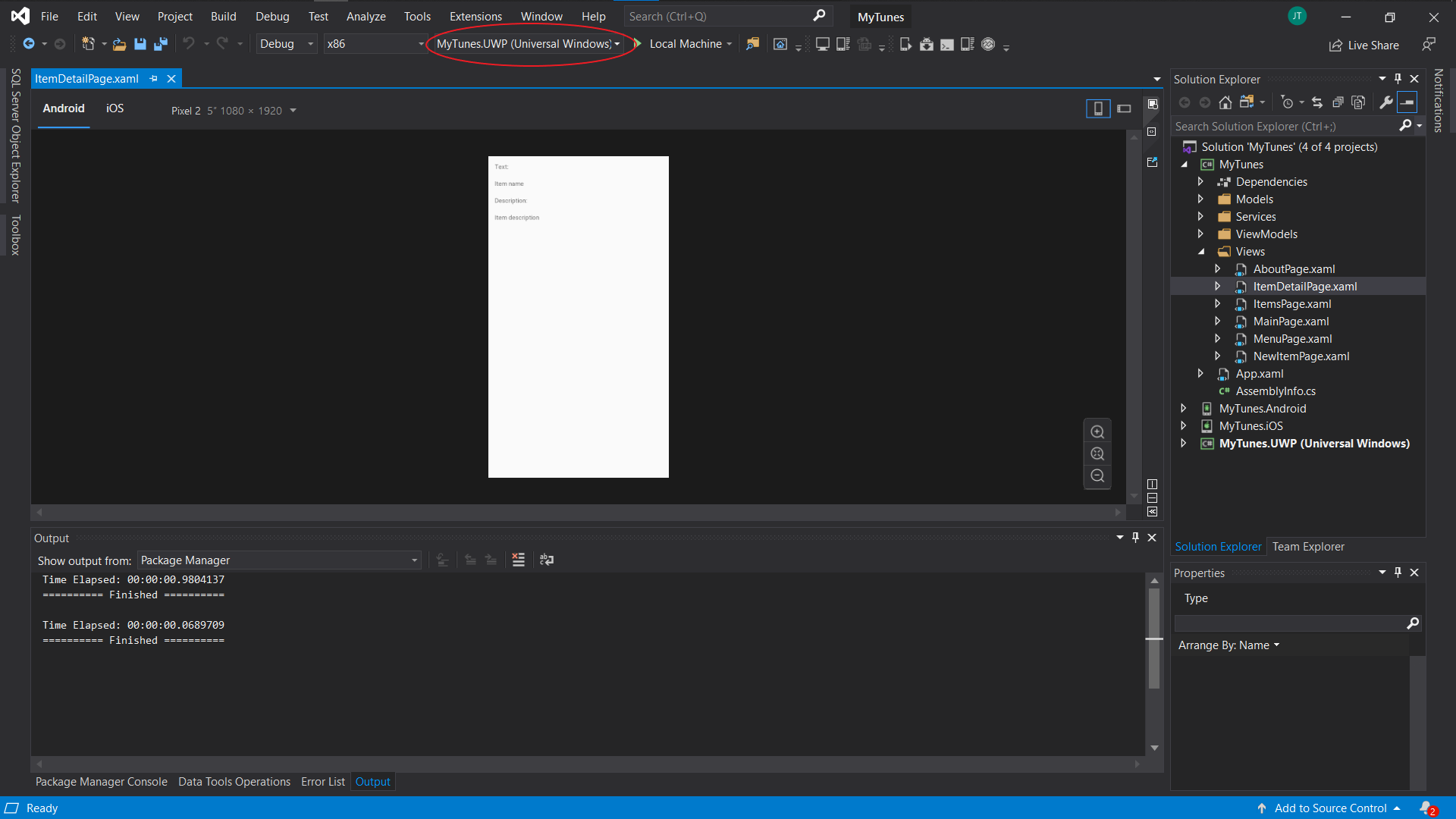


Als je bekend met C# en Xaml vraag je je waarschijnlijk af waar dat handige design venster vanboven naartoe is? Dit wordt bij Xamarin standaard apart getoond. Klik hiervoor rechtsbovenaan je xaml venster op het designer icoontje (Het initialiseren duurt even).

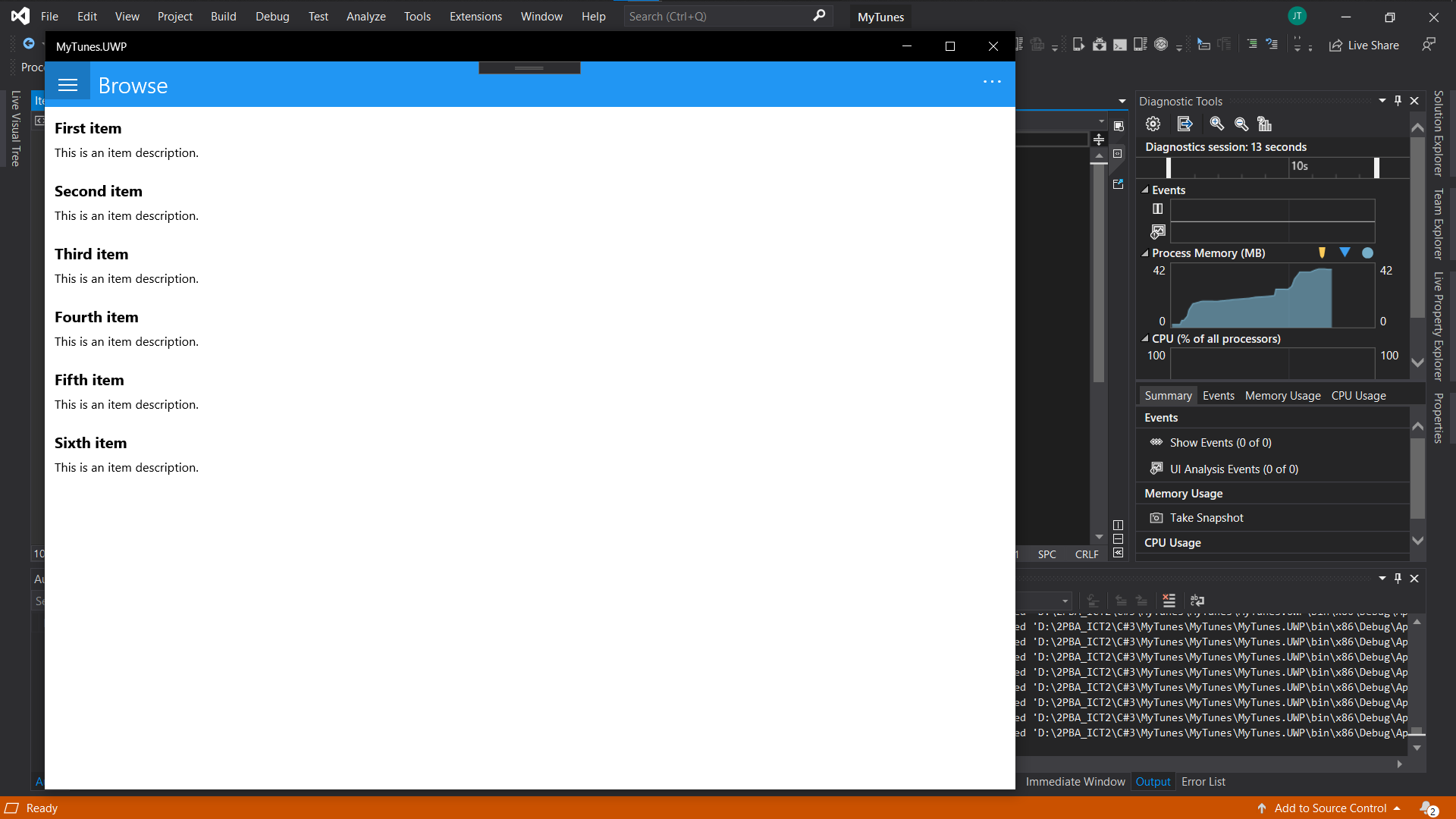


Hier kan je de previews van Android (en IOS op Mac) bekijken voor verschillende apparaten.

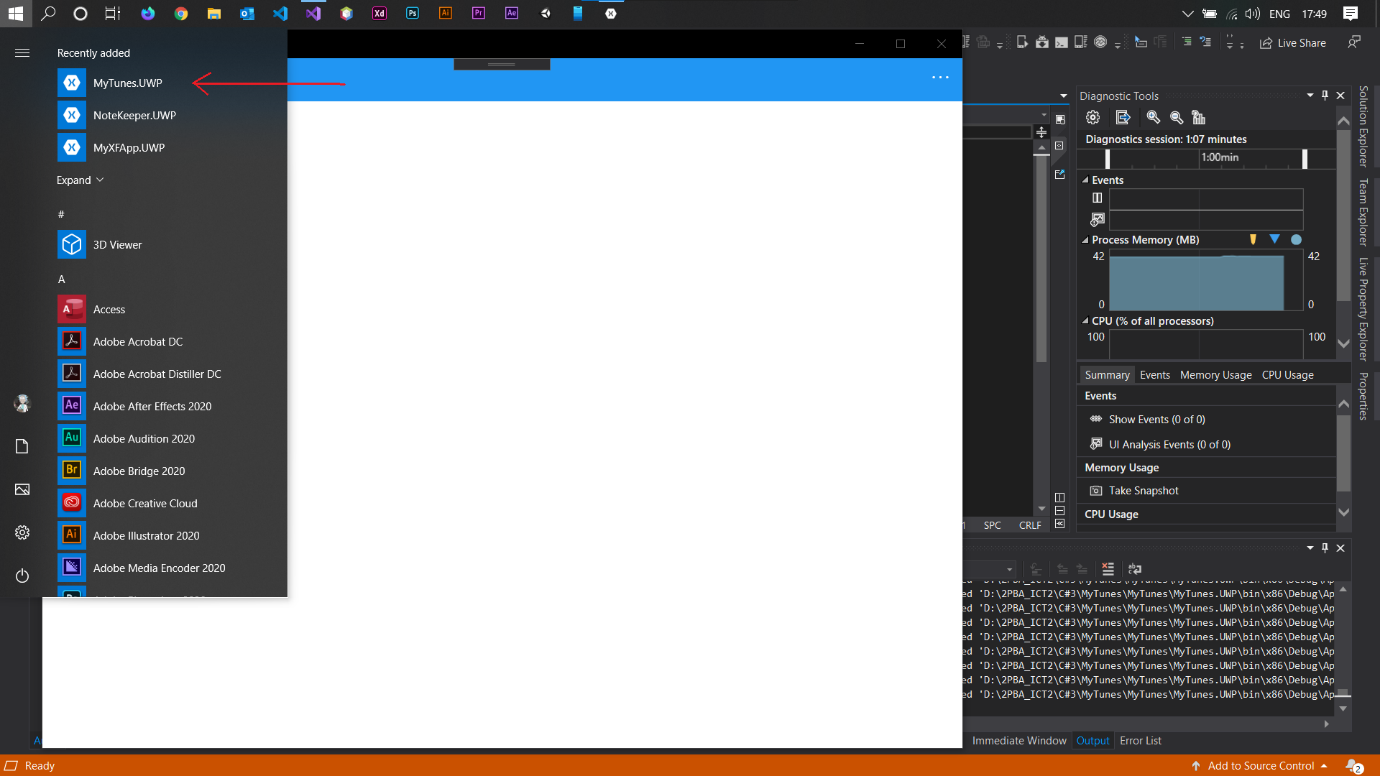


Bovenaan in Visual Studio kan je het huidige “Startup project” selecteren dat je wil bouwen en testen. In het begin staat dit op UWP.  


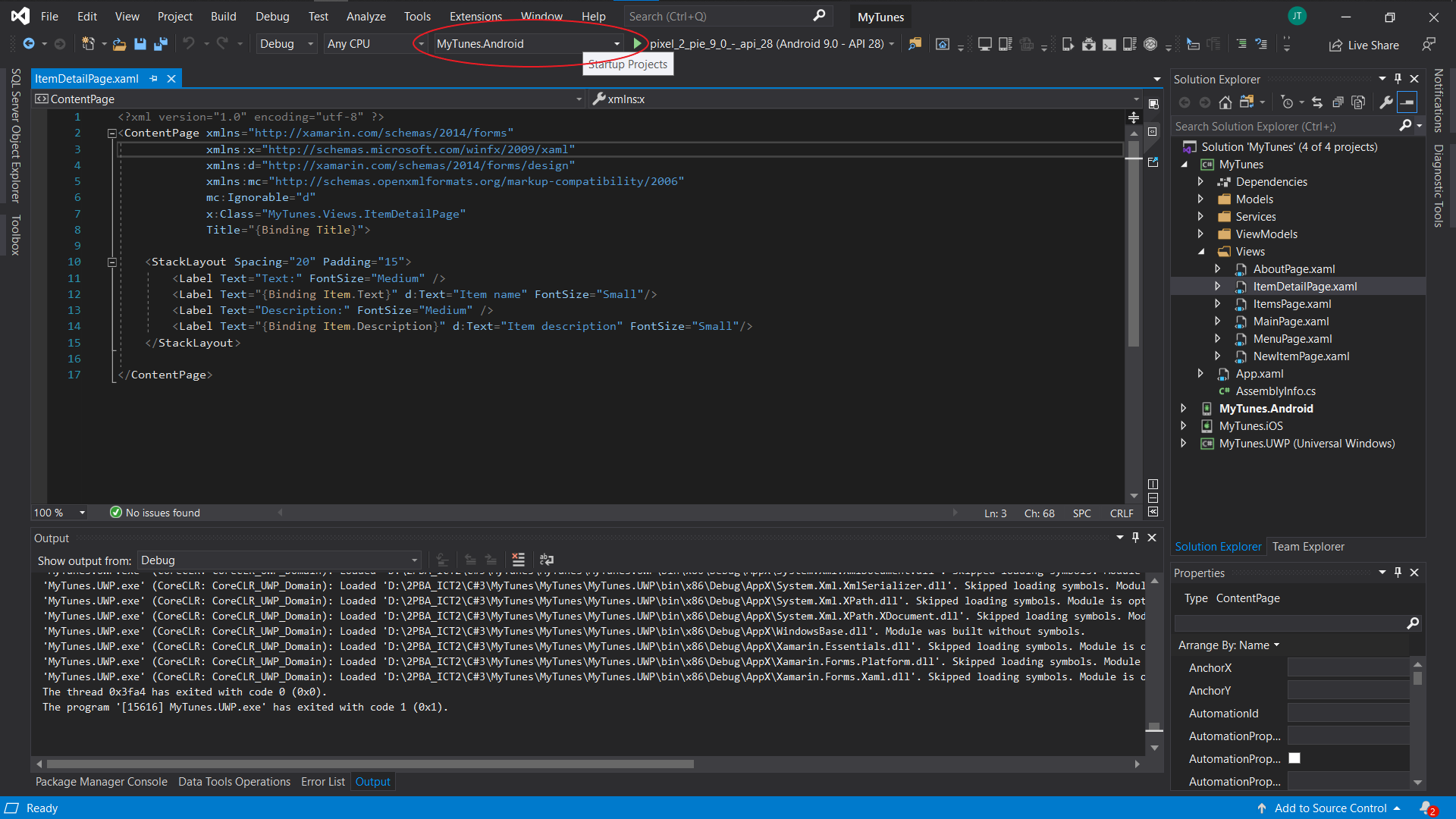
Als we deze runnen opent de Windows versie van onze app zich.

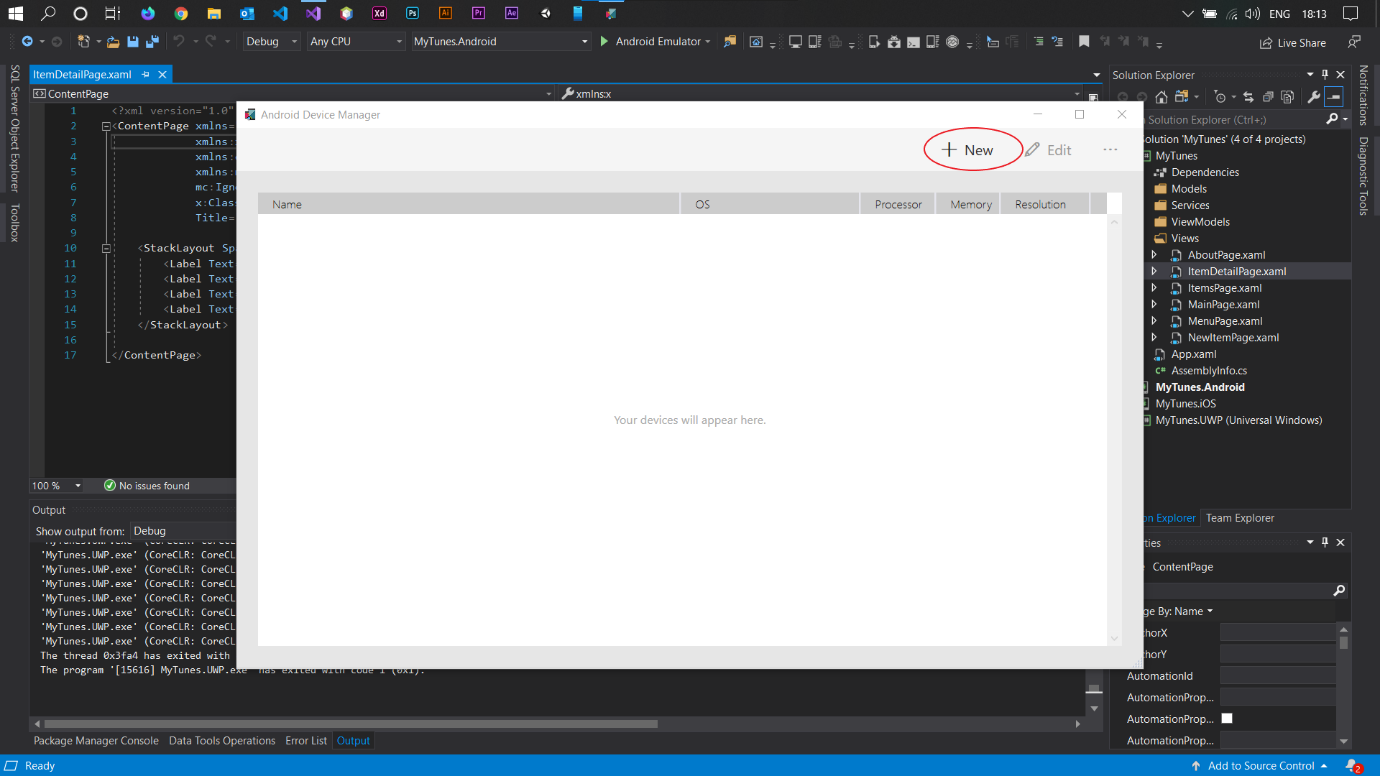


Let op: Visual Studio gaat de Windows versie van onze app ook automatisch toevoegen aan het start menu, waar deze altijd beschikbaar blijft. Als je dit niet wil kan je deze gewoon verwijderen.

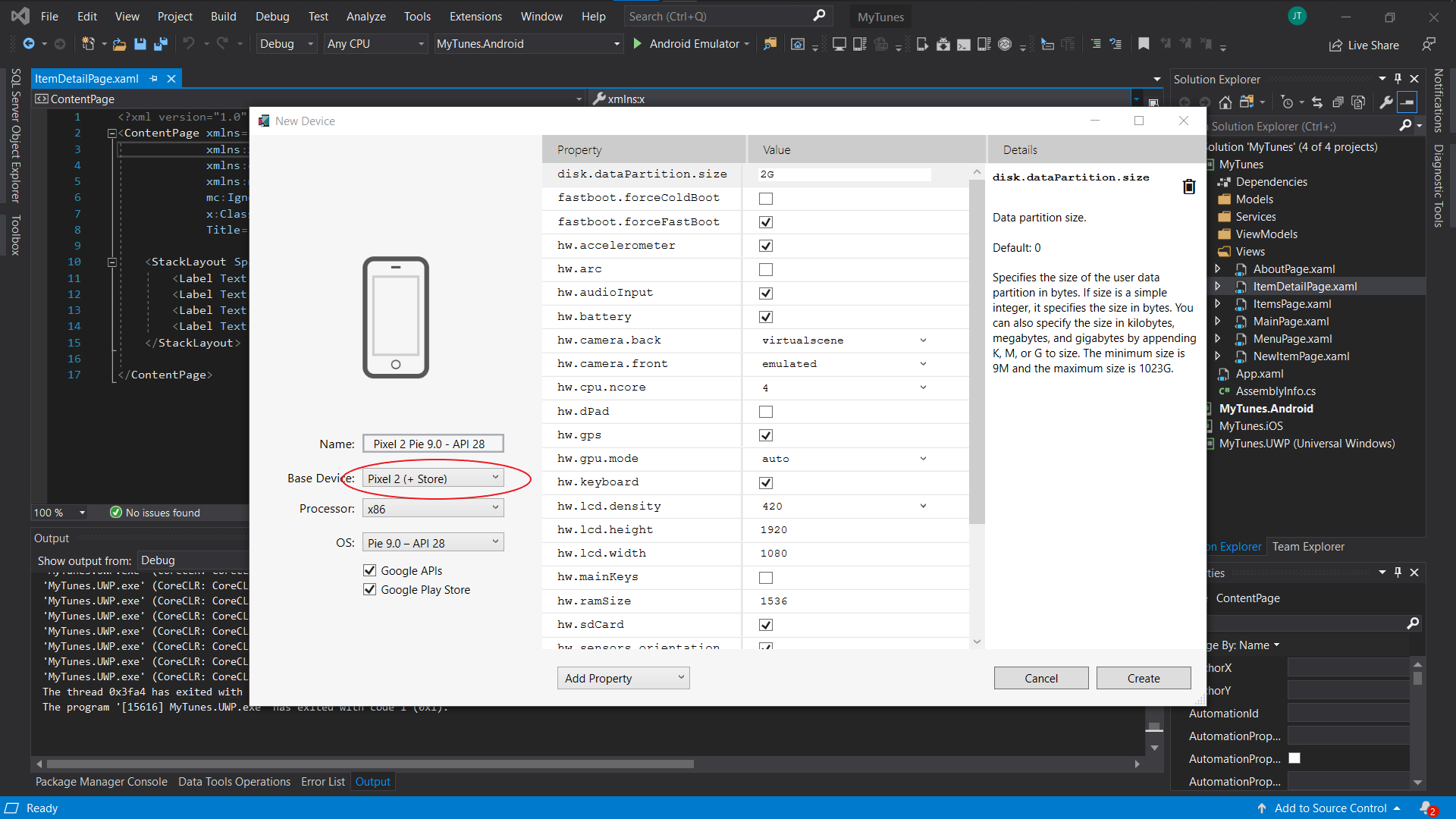


4.3 De Android emulator

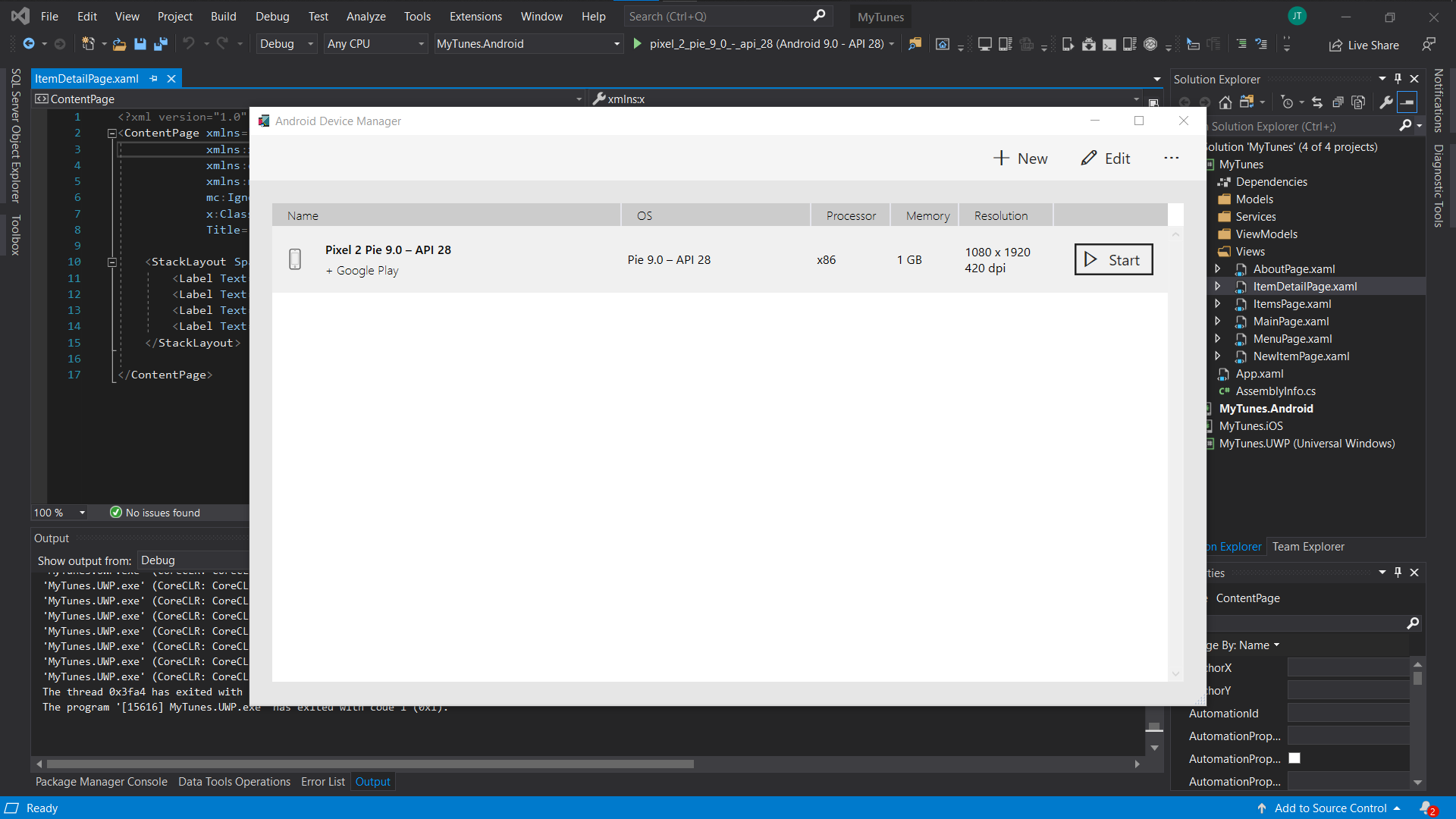
Als we de app sluiten en bij “Startup project” ons Android project selecteren zal Visual Studio eerst de Android Device manager starten.  


Hierin beheren we de virtuele Android smartphones waarmee we kunnen testen (Incl. smartwatches en smart TV’s). In het begin is dit venster leeg. Klik op new om een nieuwe virtuele smartphone aan te maken.  


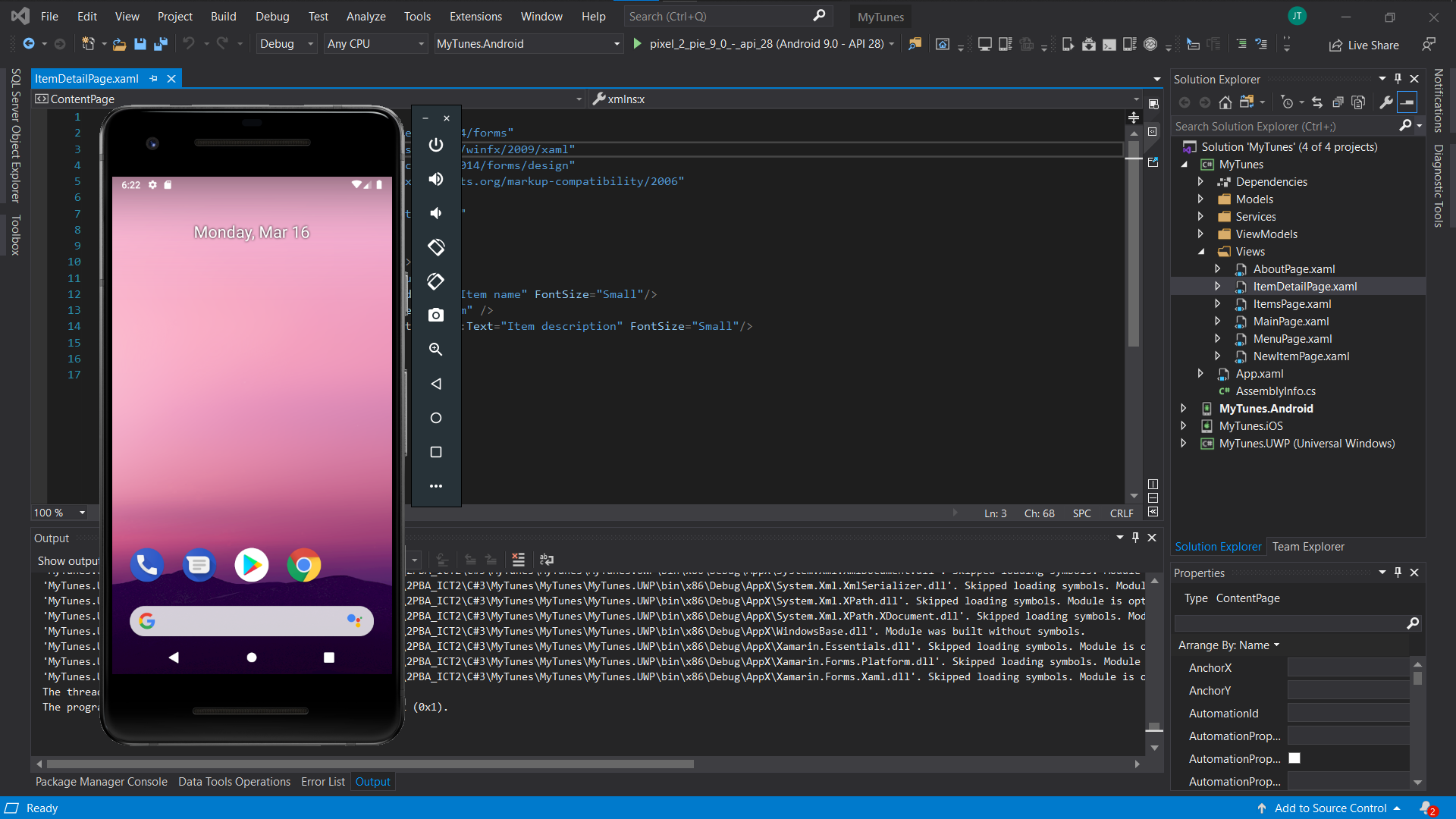
Hier kan je kiezen welke smartphone je wil emuleren plus een aantal geavanceerde opties zoals virtuele sensors, etc. Het is aangeraden om hier gewoon de standaard instellingen te kiezen. Klik vervolgens op “Create”.



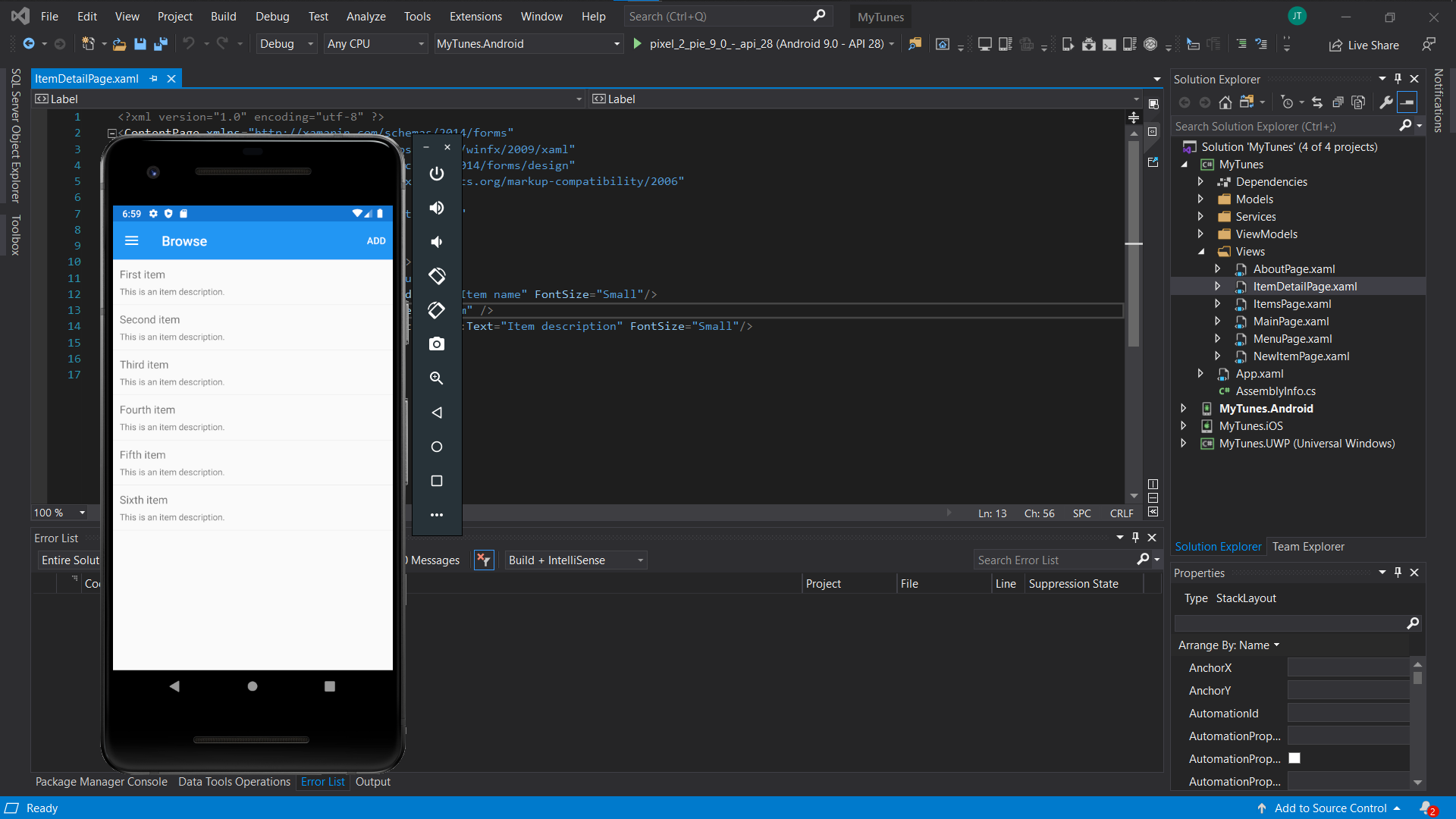
Dit kan even duren en het is mogelijk dat je bij het downloaden 2 smartphones te zien krijgt (dit is normaal). Als het klaar is ziet het er als volgt uit.  
Normaal gezien start je project ook automatisch. Als dit niet het geval is kan je dit venster gewoon sluiten. En de Android app starten in Visual Studio.



Je krijgt nu een virtuele smartphone te zien. In het begin is het scherm zwart, en het opstarten van de virtuele smartphone duurt de eerste keer lang. Als je het home-screen ziet gewoon even wachten, jouw app zal automatisch geopend worden.



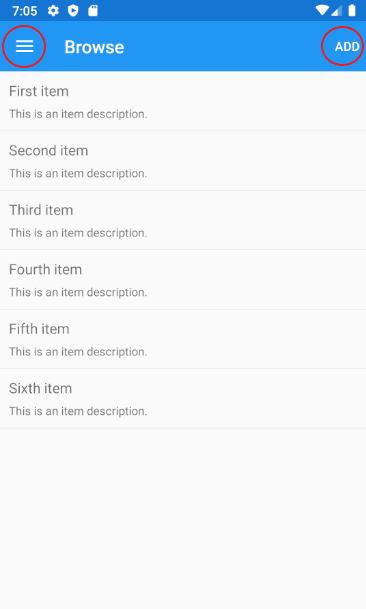
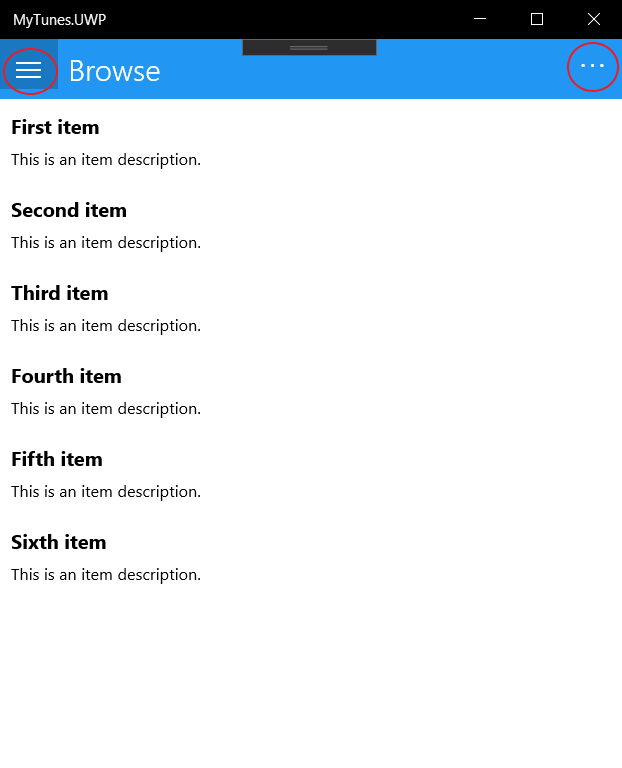
Nu de app geopend is kan je deze uitproberen.



Let op: Deze virtuele smartphone wordt gedeeld over jouw verschillende Xamarin projecten. Dat betekent dat bij je volgende Xamarin project de app wordt toegevoegd aan dezelfde smartphone. Als je dit niet wil kan je in de device manager altijd meerdere virtuele smartphones aanmaken.

4.4 De power van Xamarin

Zoals je ziet werken de verschillende implementaties van de app hetzelfde maar zitten er subtiele verschillen in de UI. Die ervoor zorgen dat de app de huisstijlen van het platform volgt. Dit is de kracht van “Xamarin.Forms”.

Android: Windows:  
 

En als we dit wensen kunnen we in de verschillende platform implementaties altijd overiddes voor onze UI toevoegen of platform specifieke functionaliteit in code implementeren.