Technische documentatie

HomePlan

Versie: 1.0

Datum: 21-01-2015

Groep: 3

Projectleden: Niek Kruse,

Edwin Lambregts,

Mike Rooijackers,

Andrew Waijers,

Ruud de Peijper

Inhoudsopgave

Software informatie 3

Gebruikte tools 3

Software versies 3

Gebruikte library 3

Backend 4

Front-End 4

Callback / delegation 5

Google GSON 5

# Software informatie

Hieronder worden tools en software versies beschreven die zijn gebruikt voor het ontwikkelen van de back- en frontend. Tools die zijn gebruikt om te ontwerpen worden niet opgenomen.

## Gebruikte tools

* Visual Studio 2012/2013
* Android Studio 1.0.2
* SQLExpress 2014

## Software versies

De frontend (Android app) is ontwikkeld om te werken met onderstaande API versies:

* Minimale vereiste Android versie: API 18 (Android 4.3 (Jelly Bean))
* Compile Android versie: API 21 (Android 5.0 (Lollipop))

## Gebruikte library

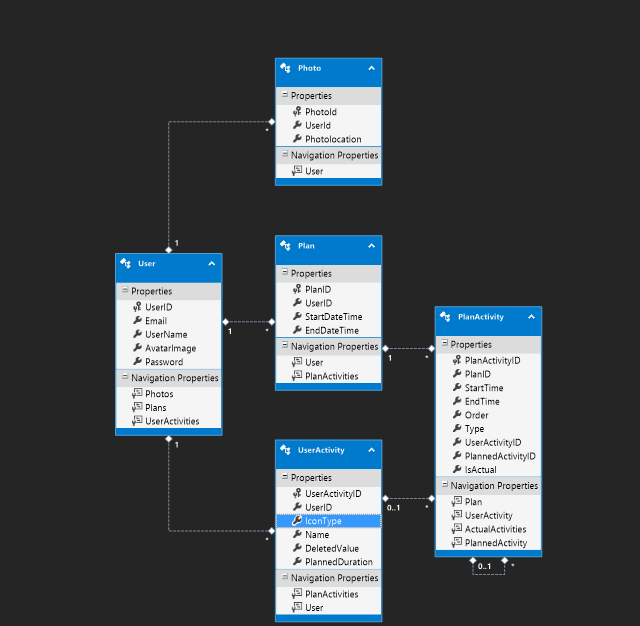
* mpandroidchartlibrary-1-7-4.jar (voor de pie chart)

# Backend

De backend van onze applicatie is waar alle data van alle gebruikers wordt opgeslagen. De applicatie heeft een verbinding met de backend nodig om de benodigde data op te vragen en naar de server te versturen.

De backend server is een ASP.NET C# applicatie, die gebruikt maakt van JSON Rest WCF services om de verzoeken vanuit de applicatie te kunnen afhandelen. Omdat er voor de ontwikkeling van onze applicatie ervoor gekozen is om deze op een zo snel mogelijke manier in elkaar te kunnen zetten, is er gekozen voor de ASP.NET oplossing.

De backend server schrijft en leest uit de database middels Entity Framework, door gebruik te maken van Entity Framework hoeft er weinig tot geen aandacht besteed te worden aan de database, waardoor veel tijd bespaard is. Windows Communication Foundation (WCF) is een manier om de communicatie tussen client (app) en server (backend) middels JSON te kunnen afhandelen.



*Fig 1. Het datamodel voor de benodigde data*

# Front-End

## Callback / delegation

Veel van de functionaliteiten in onze applicatie moeten communiceren met onze webserver om de taak succesvol te kunnen uitvoeren. Om de gebruiker zo min mogelijk last te geven van deze taken kunnen deze niet op het main thread van de applicatie worden uitgevoerd, het scherm zou in dat geval bevriezen en de gebruiker kan op zo’n moment niets met het Android device doen. Om dit tegen te gaan bevat de Android SDK “AsyncTask”. Hiermee is het mogelijk om bepaalde taken asynchroon uit te voeren. Wanneer een asynchrone taak uitgevoerd is moet het resultaat van de asynchrone taak echter doorgegeven worden aan de activiteit die de taak aanriep.

Om dit te kunnen realiseren hebben we gebruik gemaakt voor het Callback pattern. De activiteiten die bepaalde asynchrone taken aanroepen, implementeren een interface die methoden specificeren die aangeroepen worden na het uitvoeren van een bepaalde taak. De AsyncTask klasse heeft weet van deze interface, en roept de correcte interface methoden aan op de implementatie van de interface wanneer dit nodig is.

Op deze manier is het mogelijk om verschillende activiteiten gebruik te laten van de zelfde *AsyncTask,* door de verschillende activiteiten enkel een andere implementatie van de callback interface te geven.

## Google GSON

Onze Android applicatie communiceert middels een web service met de backend server. Om data naar en van deze server te kunnen versturen en ontvangen wordt er gebruik gemaakt van Json. In onze android applicatie hebben wij ervoor gekozen om gebruik te maken van Google Gson, Gson is een Json library ontwikkeld door Google, die het mogelijk maakt om Java objecten (Pojo’s) om te zetten naar een Json string en vice versa. Door gebruik te maken van Google Gson, kunnen de verschillende objecten in de C# en Java backend 1-op-1 gemapped worden, op deze manier kunnen we ondanks dat de programmeertalen anders zijn toch gebruik maken van objecten.