

GetPekt: Een eenvoudige gedeelde paklijst

Auteurs: Thomas Ilmer, Bart van den Boom, Maarten Carsjens en Ischa Stork

Voorwoord

De afgelopen periode zijn wij, AppTastic, bezig geweest met het maken van een Android App voor de cursus Research & Development. Deze cursus heeft meerdere doelen: leren hoe je een applicatie dient te ontwerpen, leren hoe je een applicatie moet realiseren en evalueren, samenwerken, presenteren en als laatste documenteren. In dit verslag willen we het laatste doel bereiken: we gaan onze ontwerpkeuzes verantwoorden, vertellen hoe onze app tot stand is gekomen en de resultaten van ons project laten zien. We gaan proberen de lezer systematisch door ons ontwerpproces te laten lopen. We beginnen met een globale beschrijving van de app en de belangrijkste eisen/eigenschappen die we hebben bedacht voordat we begonnen met het ontwerpen van de app. Daarna gaan we iets vertellen over verwante apps en gaan we beargumenteren waarom het de moeite waard is om onze app te ontwikkelen. Vervolgens werken we enkele gekozen eigenschappen uit door middel van use cases. Verder gaan we nog wat vertellen over het ontwerp van de app, waarbij we wat vertellen over de programmatuur en we dieper ingaan op beslissingen die we hebben genomen ten opzichte van de verschillende eisen. Ook kijken we naar alternatieve ontwerpkeuzes en de voor- en nadelen van die keuzes. Als laatste blikken we terug op de afgelopen periode en analyseren we ons ontwerpproces en de bereikte doelen. Uiteindelijk kunnen we dan een concluderen of het ons project geslaagd is en kunnen we tips of suggesties geven voor het volgende project.

Inhoudsopgave

Voorwoord	1
Beschrijving	4
Inleiding	4
Productverantwoording	5
Specificatie	5
Functionele eigenschappen	5
Niet-functionele eigenschappen	5
Use-cases	6
Ontwerp	7
Globaal ontwerp	7
Inloggen	7
Evenementen	8
Evenement aanmaken	9
Evenement bekijken	10
Evenement informatie	11
Settings	11
Detailontwerp	12
LoginActivity	12
RegisterActivity	13
AppController	13
SessionManager	13
SQLiteHandler	13
MainActivity	14
PhoneActivity	14
CreateActivity	15
CropActivity	15
AddContactsActivity	15
EventActivity	15

InfoActivity.....	16
SettingsActivity	16
PHP.....	16
Ontwerpverantwoording	17
Inloggen	17
Telefoonnummers	17
Evenement Informatie.....	17
Reflectie	18
Ons ontwerpproces.....	18
Bereikte doelen	18
Conclusie en tips voor de volgende keer	19

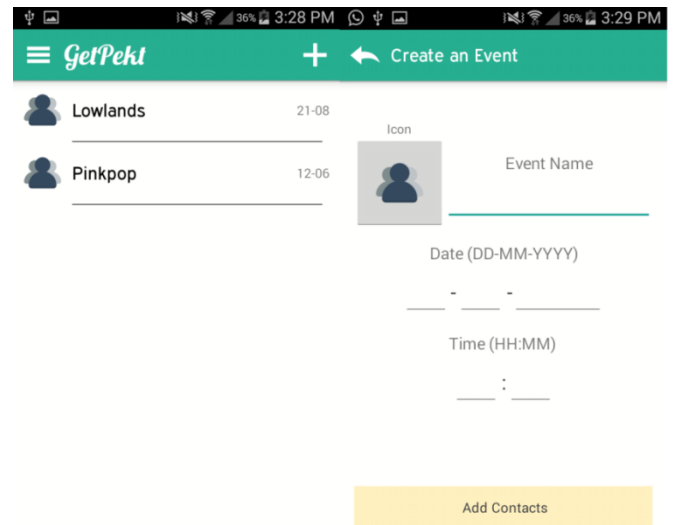
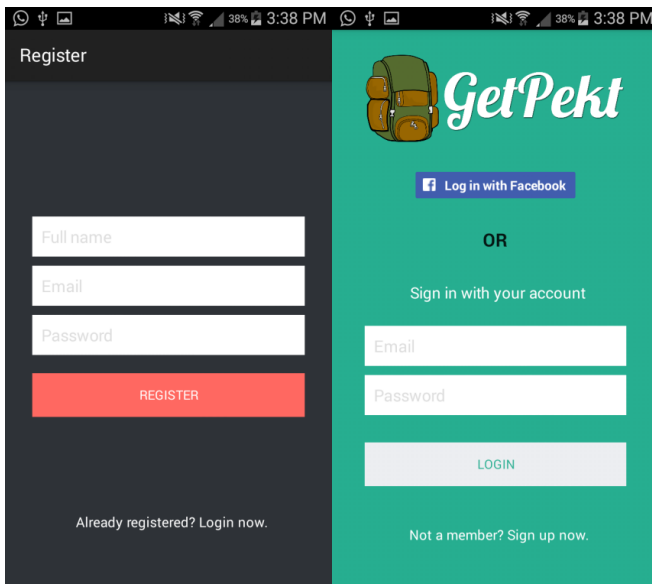
Beschrijving

Inleiding

Een probleem wat veel festivalgangers wellicht tegenkomen; er moeten spullen meegenomen worden, maar het is niet duidelijk wie wat meeneemt. Wij wilden dit probleem graag oplossen met een android app.

Het idee van onze app is het kunnen maken en delen van checklists, specifiek voor evenementen zoals festivals, campingvakanties, etc. Het idee hierachter is dat iedereen een gedeelde lijst heeft met spullen die meegenomen moeten worden, waarin iedereen kan selecteren wat hij/zij meeneemt. Op deze manier is het makkelijk en overzichtelijk wie nou wat meeneemt. Dit voorkomt niet alleen het vergeten van spullen, maar ook het dubbel meenemen van spullen.

Belangrijk bij het maken van deze app was dat het eenvoudig en intuïtief in gebruik moest zijn, zodat we een grote doelgroep bereiken.



Productverantwoording

Belangrijke motivatie voor het maken van deze app was dat de concurrentie geen directe oplossing bood voor het door ons gestelde probleem: een overzicht krijgen van alle spullen die meegenomen moeten worden en welke al meegenomen worden.

Een groot deel van de concurrentie bestaat uit persoonlijke checklist apps. Met deze apps kun je wel checklists bijhouden, maar kun je niet anderen personen toevoegen aan deze checklist om synchroon erin te werken. Een voorbeeld hiervan is 'Simplest Checklist(check list)'. Deze app is erg simpel en heeft een goedwerkende gebruikersinterface, maar de checklists in deze app zijn niet te delen met anderen.

De enige app die deze functionaliteit wel bood was 'Checklist.com', een vrij populaire app met ongeveer 10.000 downloads. Ons probleem met deze app was dat het erg complex was in gebruik. De gebruikersinterface was erg onduidelijk en kwam neer op trial-and-error.

Om deze reden denken wij dat onze app meerwaarde biedt. Het heeft de functionaliteit die velen niet hebben, maar heeft naar onze mening een simpelere, intuïtieve, gebruikersinterface.

Specificatie

Functionele eigenschappen

- Het moet mogelijk zijn om te registreren en in te loggen.
- Het maken/aanpassen/verwijderen van evenementen
- Het toevoegen, selecteren en verwijderen van voorwerpen bij een evenement.
- Toevoegen van contacten aan evenementen.

Niet-functionele eigenschappen

- De app moet intuïtief werken, dat betekent dat het voor de gebruiker meteen duidelijk moet zijn wat er van hem of haar wordt gevraagd.
- Het moet altijd duidelijk zijn wat er gebeurt; de app moet feedback geven als een gebruiker een fout maakt of er informatie wordt geladen vanuit de database.
- De app moet eenvoudig zijn, dus geen overbodige opties die niemand begrijpt of gebruikt.

Use-cases

Use case 1

Registeren	
Beschrijving	Een gebruiker registreert een nieuw account.
Gebruiker	Een gebruiker van de app
Preconditie	De gebruiker heeft verbinding met het internet.
Basic Course of Events	<ol style="list-style-type: none">1. De gebruiker opent de app2. De gebruiker klikt op "Not a member? Sign up now."3. De gebruiker vult zijn gegevens in.4. De gebruiker klikt op sign up5. De app voegt het account toe aan de database.6. De gebruiker wordt doorgestuurd naar het login scherm.
Alternate flow	<ol style="list-style-type: none">3a. De gebruiker klikt op 'Already registered? Login now'3b. De gebruiker wordt doorgestuurd naar het login scherm
Postconditie	De gebruiker heeft een nieuwe account.

Use case 2

Een evenement aanmaken	
Beschrijving	Een nieuw evenement aanmaken
Gebruiker	Een gebruiker van de app
Preconditie	De gebruiker heeft verbinding met het internet en is ingelogd.
Basic Course of Events	<ol style="list-style-type: none">1. De gebruiker klikt rechtsboven op het "+" icoon.2. De gebruiker kiest een icoontje .3. De gebruiker vult de gegevens van het evenement in.4. De gebruiker klikt op 'Add Contacts'.5. De app laat op het scherm de lijst met contacten zien.6. De gebruiker selecteert nu een of meerdere contacten7. De gebruiker klikt op 'Create Event'.8. De app maakt het evenement aan.9. De app stuurt de gebruiker door naar het scherm met alle evenementen.
Alternate flow	<ol style="list-style-type: none">4a. De gebruiker klikt op het pijltje linksboven4b. De gebruiker wordt teruggestuurd naar het scherm met alle evenementen7a. De gebruiker klikt op het pijltje linksboven7b. De gebruiker wordt teruggestuurd naar het vorige scherm, waar hij de informatie van zijn evenement heeft ingevuld.
Postconditie	Er is een nieuw evenement aangemaakt.

Ontwerp

Globaal ontwerp

Onze app is op te delen in een aantal componenten, deze zullen wij hieronder een voor een bespreken in de volgorde waarop je door deze componenten heen gaat als je voor de eerste keer de applicatie opstart.

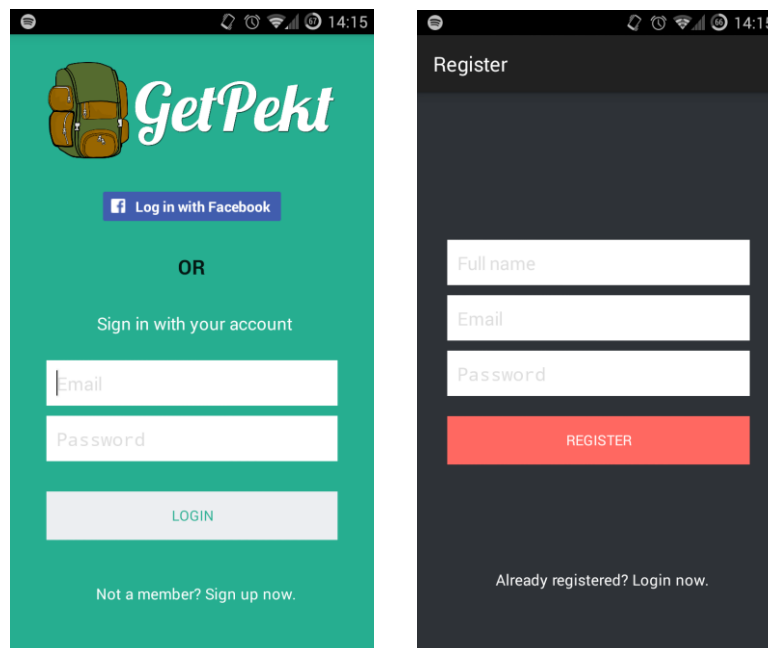
Inloggen

Bij het opstarten van de applicatie komt de gebruiker in onze login activity (zie figuur 1) terecht. Hier heeft de gebruiker twee opties:

1. Inloggen met Facebook.
2. Een account aanmaken om vervolgens hiermee in te loggen.

Optie 1 spreekt voor zich, de gebruiker klikt op de ‘log in with Facebook’ knop en zal hierna doorgestuurd worden naar een aparte activity van Facebook waar de gebruiker moet aangeven of hij inderdaad deze app toegang wil geven tot zijn/haar Facebook.

Optie 2 is opgedeeld in 2 verschillende activities: eerst gaat de gebruiker naar de registratie activity en maakt hij een account aan. Vervolgens wordt de gebruiker weer teruggestuurd naar de login activity waar hij met het zojuist aangemaakte account kan inloggen. Als de gebruiker succesvol is ingelogd wordt er een sessie opgestart en wordt de gebruiker doorgestuurd naar de event activity.



Figuur 1: login en register activity

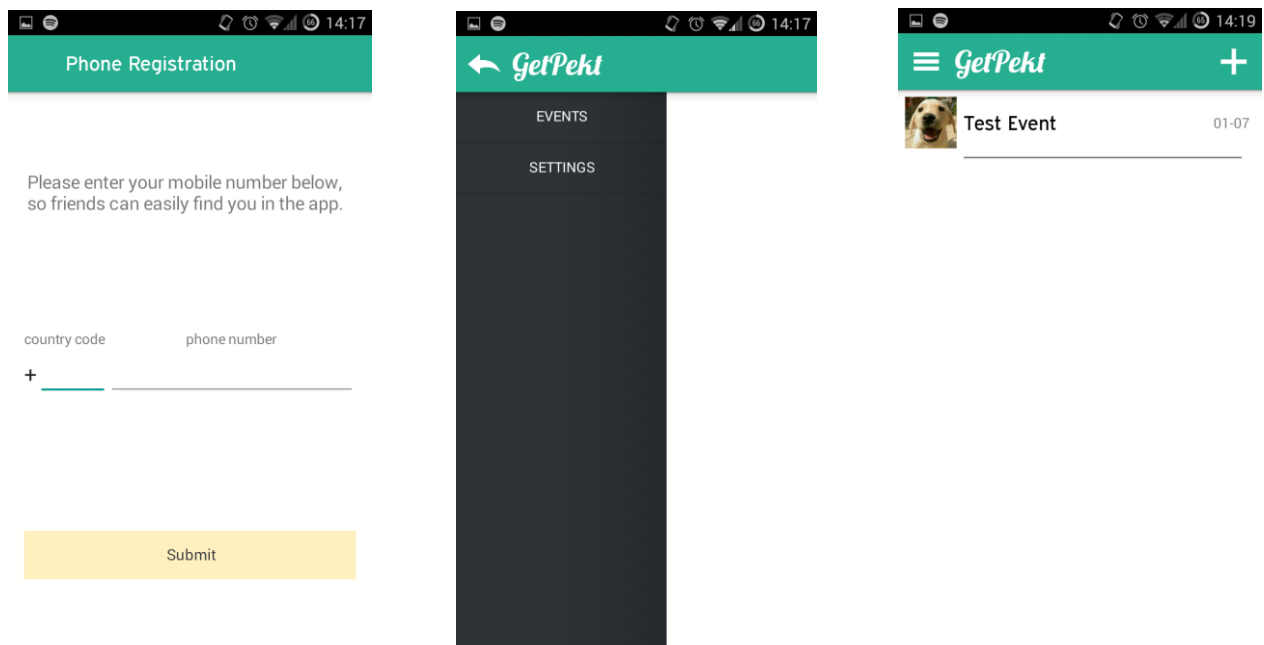
Evenementen

De main activity is het scherm waarin de gebruiker terecht komt na het inloggen. De eerste keer dat de gebruiker inlogt zal hij gevraagd worden om zijn telefoonnummer in te voeren, zodat er in onze database komt te staan dat hij een gebruiker van onze app is. Op deze manier kunnen wij door de contactlijst van de gebruiker lopen en kijken of er telefoonnummers zijn die in onze database staan, als dat het geval is dan weten we dat de gebruikers vrienden van elkaar zijn en krijgt de gebruiker de mogelijkheid om deze persoon uit te nodigen voor evenementen.

De main activity bestaat uit 3 hoofdelementen:

- de lijst met evenementen
- de '+'-knop
- het menu

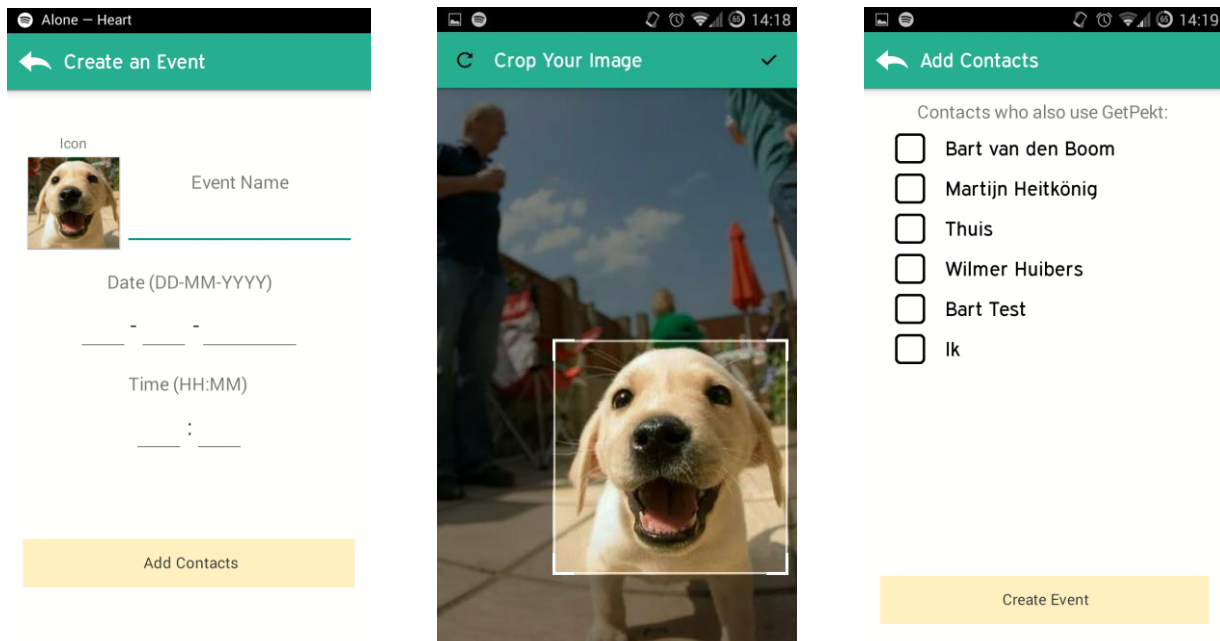
Al deze elementen spreken redelijk voor zich; het grootste deel van het scherm is gevuld met een lijst van evenementen, waar je het icoontje, de naam van het evenement en de datum ziet. Als er op een evenement wordt geklikt zal de gebruiker worden doorgestuurd naar de event activity, als de gebruiker op de '+'-knop drukt zal hij doorgestuurd worden naar een activity waar hij een evenement kan aanmaken, en de menuknop opent het slidingmenu waar de gebruiker naar de settings kan gaan.



Figuur 2: de main activity

Evenement aanmaken

Dit is de activity waar de gebruiker een evenement aan kan maken. Er zijn verschillende invoervelden voor de naam van het evenement, de datum en de tijd. Daarnaast kan de gebruiker ook nog het icoontje van het evenement aanpassen, waarbij hij wordt doorgestuurd naar zijn gallery en daarna (als hij een afbeelding heeft gekozen) naar de cropper activity. Als de gebruiker tevreden is met de ingevoerde gegevens kan hij op de ‘add contacts’ knop drukken en zal hij op een scherm komen waar een lijst staat met de contacten uit zijn contactenlijst die ook gebruik maken van de app. Hier kan hij de gewenste contacten aan het evenement toevoegen en het evenement aanmaken.



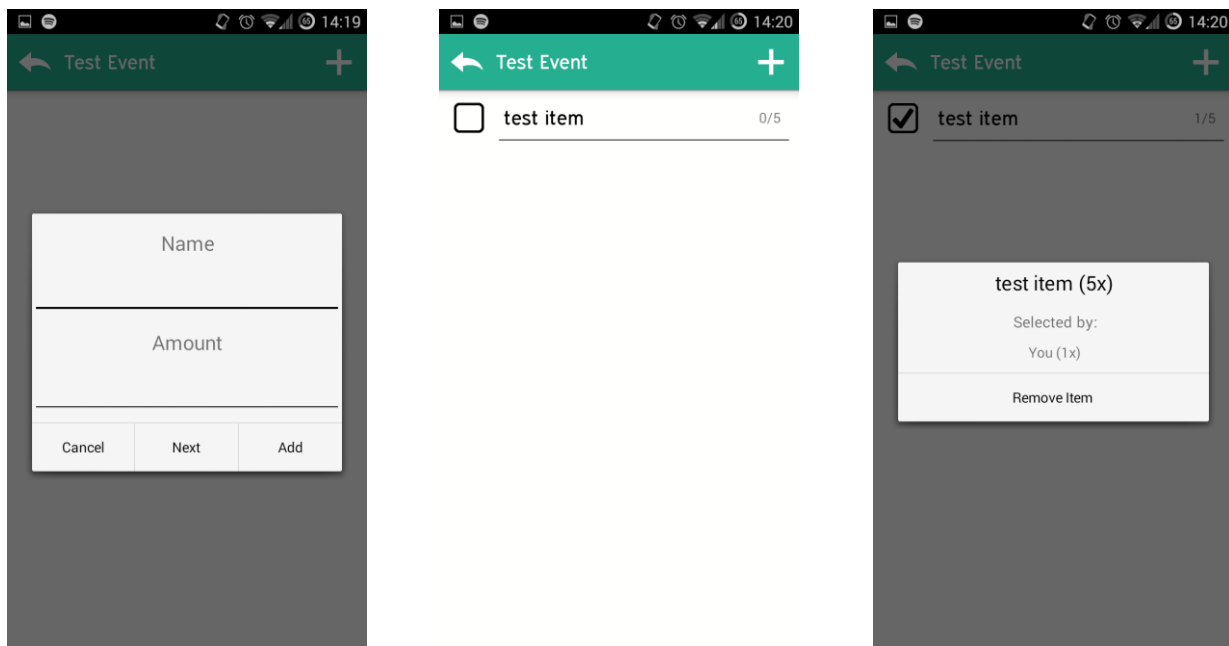
Figuur 3: de verschillende activiteiten voor het aanmaken van een evenement

Evenement bekijken

Als de gebruiker een evenement heeft geselecteerd komt hij op het scherm behorende bij dit evenement terecht. Bovenin, op de ActionBar, staat de naam van het evenement, als hierop wordt geklikt dan wordt de gebruiker doorgestuurd naar de informatie van het evenement. Verder bestaat deze activity weer uit 3 hoofdonderdelen:

- de lijst met voorwerpen
- de '+'-knop
- de terug-knop

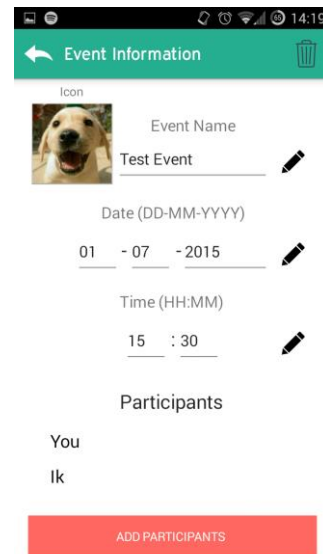
De '+'-knop geeft een dialog waarin de gebruiker een nieuw item kan toevoegen, met een bijbehorende hoeveelheid. Het grootste deel van het scherm wordt gevuld door de lijst met voorwerpen; elk voorwerp wordt hier getoond met een checkbox ervoor, deze is leeg als het item door niemand is geselecteerd, heeft een grijs vinkje als het door iemand anders is geselecteerd en een zwart vinkje als het door de gebruiker zelf is geselecteerd. Aan de andere kant van het scherm staat de hoeveelheid behorende bij het voorwerp en hoeveel er al zijn geselecteerd. Verder kan de gebruiker op de naam van het voorwerp drukken om te tonen wie het voorwerp hebben geselecteerd en om het item te verwijderen.



Figuur 4: het bekijken van een evenement

Evenement informatie

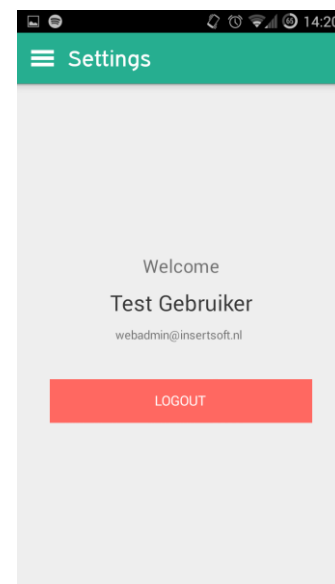
Dit scherm toont de informatie van het evenement. Als de gebruiker degene is die de groep heeft aangemaakt heeft hij hier de mogelijkheid om het evenement te verwijderen, nieuwe contacten toe te voegen en de informatie aan te passen. Als de gebruiker de groep niet heeft aangemaakt kan hij alleen de informatie aanpassen en de groep verlaten.



Figuur 5: de evenement informatie voor de gebruiker die de groep heeft aangemaakt

Settings

Als laatste heb je het settings scherm, bereikbaar vanuit de main activity via het menu. Op het moment staat er op dit scherm alleen maar wat informatie van de gebruiker en een knop om uit te loggen. Ook heb je weer het slidingmenu waarmee de gebruiker terug kan gaan naar de main activity.



Figuur 6: Het settings scherm

Detailontwerp

Ons product bestaat uit 13 verschillende klassen:

- AddContactsActivity
- AppController
- CreateActivity
- CropActivity
- EventActivity
- InfoActivity
- LoginActivity
- MainActivity
- PhoneActivity
- RegisterActivity
- SettingsActivity
- SessionManager
- SQLiteHandler

Verder hebben wij nog honderden regels aan php code draaien op een server om de communicatie met de MySQL database te onderhouden, hier zullen wij later ook het een en ander over vertellen.

Hieronder staat per klasse een korte samenvatting wat de klasse doet en wat de belangrijkste functies en attributen van deze klasse zijn.

LoginActivity

Deze activity regelt het login scherm, bijna alles in deze activity gebeurt in de onCreate functie. Als eerste worden de Facebook SDK en de SQLiteHandler geïnitieerd. De rest van de onCreate functie is op te delen in twee delen: het Facebook gedeelte en het account gedeelte.

Het Facebook gedeelte controleert eerst of de gebruiker al is ingelogd, als dat het geval is dan start deze de MainActivity. Als dit niet het geval is dan maakt hij een CallbackManager aan voor de Facebook knop, zodat de gebruikersdata van de gebruiker aan de lokale database wordt toegevoegd en de MainActivity wordt gestart als er succesvol wordt ingelogd op Facebook.

Ook het account gedeelte controleert eerst of de gebruiker al is ingelogd, als dat het geval is dan wordt ook deze de gebruiker doorgestuurd naar de MainActivity. Als dit niet het geval is worden er 2 onClickListeners gestart. Eén voor de login knop die de checkLogin functie aanroept als er iets is ingevoerd. De andere onClickListener is voor de register knop, die de gebruiker doorstuurt naar de RegisterActivity.

De checkLogin functie stuurt de inloggegevens van de gebruiker naar de webserver door middel van de volley library, als deze geen error terugstuurt dan worden de gebruikersgegevens toegevoegd aan de lokale database en wordt de gebruiker doorgestuurd naar de MainActivity.

RegisterActivity

De RegisterActivity is een relatief kleine klasse; in de onCreate wordt eerst gecontroleerd of de gebruiker niet al is ingelogd en als dit op de een of andere manier toch wel het geval is dan wordt de gebruiker doorgestuurd naar de MainActivity. In alle normale scenario's is dit echter niet het geval en zullen er weer 2 onClickListeners gestart worden; één voor de knop die je terugbrengt naar de LoginActivity en één die de registerUser functie aanroept als alle gegevens zijn ingevoerd en het wachtwoord minstens 8 tekens lang is.

De registerUser functie lijkt heel erg op de checkLogin functie; ook hier worden de (door de gebruiker ingevoerde) gegevens naar de server doorgestuurd met behulp van de volley library. Als dit geen error oplevert dan zal de gebruiker weer worden teruggestuurd naar de LoginActivity waar hij met het zojuist aangemaakte account kan inloggen.

AppController

De AppController klasse regelt alle communicatie via de volley library. Het is een vrij kleine klasse die een paar korte functies heeft om het versturen en ontvangen van data via volley iets makkelijker te maken.

SessionManager

De SessionManager is een hele korte klasse die in de SharedPreferences aangeeft of de gebruiker is ingelogd. Elke keer als er een klasse wordt opgestart waar de gebruiker ingelogd voor moet zijn (dat zijn alle klassen behalve de LoginActivity en de RegisterActivity) wordt er gecontroleerd of de gegevens van de gebruiker in de lokale database staan en als de gebruiker is ingelogd met Facebook wordt er gecontroleerd of er een Facebook sessie is geïnitieerd.

SQLiteHandler

De SQLiteHandler regelt de database die lokaal op de telefoon draait. Deze database bestaat uit 3 tables:

- de login table
- de contacts table
- de icons table

De functies in deze klasse zijn allemaal gericht op het toevoegen en ophalen van gegevens uit deze 3 tables. De login table wordt gebruikt om de gegevens van de huidige gebruiker op te slaan, de contacts table is een table waarin alle contacten uit de contactenlijst van de gebruiker staan die ook gebruik maken van de app en de icons table is een table waarin alle icoontjes van

de verschillende evenementen als BLOB (Binary Large Object) worden opgeslagen zodat de app niet elke keer deze icoontjes van de online database moet halen.

MainActivity

De MainActivity is het hoofdscherm van de hele applicatie. Als de gebruiker is ingelogd of teruggaat vanuit een evenement komt hij in dit scherm terecht. De MainActivity is een van de grootste klassen in onze applicatie en bestaat uit een paar belangrijke functies. In de onCreate functie wordt eerst gecheckt of dit de eerste keer is dat de gebruiker inlogt door te kijken of hij in de online database met telefoonnummers staat, als dit niet het geval is dan wordt de PhoneActivity aangeroepen. Verder maakt de onCreate de ActionBar en het slidingmenu aan en wordt de getCountry functie aangeroepen. De ActionBar heeft 2 knoppen, de '+'-knop die de CreateActivity opstart en de menu-knop die het slidingmenu toont.

De getCountry functie haalt het land van de gebruiker op uit de telefoondatabase en roept vervolgens de getContacts functie aan. Deze functie leest de contactenlijst van de gebruiker uit en controleert per telefoonnummer of het nummer in de online database staat. Als het telefoonnummer met een 0 begint wordt dit vervangen door +(landcode). Alle telefoonnummers die ook in de online database staan worden vervolgens opgeslagen in de lokale contacts table, zodat de app weet welke contacten de gebruiker kan toevoegen aan zijn evenementen. Hierna worden respectievelijk de getEvents en de drawEvents functies aangeroepen. Deze 2 functies spreken redelijk voor zichzelf, de eerste haalt de evenementen van de gebruiker op uit de online database en de tweede tekent vervolgens deze opgehaalde evenementen. De drawEvents functie controleert ook of de versienummers van de icoontjes hetzelfde zijn als het versienummers uit de lokale database, als de versienummers overeenkomen dan worden de icoontjes direct getekent, anders worden ze eerst opgeslagen voordat ze worden getekent. De getekende evenementen krijgen allemaal een onClickListener mee die de EventActivity start en de gegevens van het aangeklikte evenement meegeeft.

PhoneActivity

Dit is de klasse die wordt aangeroepen als de gebruiker voor het eerst inlogt. Het is een erg simpele activity met 2 EditText velden en een button die de gegevens naar de online database verstuurd. De code in deze klasse is dan ook voornamelijk te vinden in de onCreate functie en bestaat eigenlijk alleen uit een onClickListener voor de knop en een ProgressDialog die wordt getoond als de button wordt ingedrukt. Verder override deze klasse de onBackPressed functie, net als in de meeste klassen in onze applicatie. De functie is zo ingesteld dat de er een Toast wordt getoond als de back knop wordt ingedrukt en als er binnen 2 seconden nog een keer op de knop wordt gedrukt dan wordt de applicatie afgesloten.

CreateActivity

Deze klasse wordt gebruikt om een nieuw evenement aan te maken. Zoals bij alle andere klassen wordt eerst in de onCreate de ActionBar getekend en wordt de vorige knop aangemaakt. De rest van de functionaliteit binnen deze klasse vindt plaats in de buttonEvents functie.

De buttonEvents functie maakt 2 onClickListeners aan, de eerste luistert naar de knop om het icoontje aan te passen, als deze wordt ingedrukt wordt er een dialog getoond waarin de gebruiker kan kiezen uit het standaard icoontje of een eigen icoontje. Als de gebruiker kiest voor een eigen icoontje wordt de CropActivity gestart. De andere onClickListener is voor de 'add contacts' knop, als deze knop wordt ingedrukt zet de app eerst de datum en tijd om naar het door de database ondersteunde formaat en controleert hij vervolgens of de ingevoerde gegevens correct zijn. Als dit het geval is start hij de AddContactsActivity en geeft hij al deze gegevens mee.

CropActivity

Op het moment dat de CropActivity wordt aangeroepen maakt deze meteen weer een nieuwe intent aan om de galerij van de gebruiker te openen zodat deze een icoontje kan selecteren. Vervolgens krijgt de CropActivity via de onActivityResult dit plaatje binnen, en wordt de ActionBar met de bijbehorende buttons aangemaakt. De rest van deze Activity wordt voornamelijk geregeld door een externe library, hier valt dus niet veel over te vertellen.

AddContactsActivity

Deze activity wordt aangeroepen op het moment dat de maker van een groep contacten aan zijn groep wil toevoegen, of dit nou bij het aanmaken van de groep of bij het aanpassen van de groepsinformatie.

In de drawContacts functie worden de contacten uit de lokale contacts table in de database getekend, mits deze niet al in de groep zitten. Als de gebruiker op de knop drukt zullen alle gebruikers die aangevinkt zijn aan de groep worden toegevoegd door middel van een communicatie met de online database die via Volley tot stand is gebracht. Als dit tijdens het aanmaken van de groep gebeurt dan wordt ook de gebruiker zelf als creator in de online database toegevoegd.

EventActivity

De EventActivity is de hoofdfunctionaliteit van onze applicatie. In de onCreate worden weer de ActionBar en de "+"-knop getekend. de "+"-knop toont een dialog waarin de gebruiker een naam en bijbehorende hoeveelheid van een voorwerp invoeren dat hij wil toevoegen. De gebruiker kan dan op drie knoppen drukken: hij kan op 'next' drukken om het voorwerp toe te voegen, vervolgens krijgt hij nog een keer hetzelfde dialog om nog een voorwerp toe te voegen.

Als hij op 'add' drukt wordt het voorwerp ook toegevoegd maar het dialog wordt gesloten, de gebruiker kan ook op cancel drukken als hij de actie wil annuleren. Bij de ActionBar wordt als titel de naam van het evenement gebruikt, dit is ook een knop die, als erop gedrukt wordt, de InfoActivity start en de gegevens van de groep meegeeft.

De getItem functionie wordt aangeropen als de gebruiker de groep opent, een voorwerp aanvinkt, een voorwerp toevoegt en als de applicatie wordt hervat. Deze functie haalt met behulp van volley de voorwerpen van de huidige groep op uit de online database en roept vervolgens drawItems aan.

De drawItems functie tekent alle items met bijbehorende checkbox: leeg als het bijbehorende voorwerp door niemand is geselecteerd, met een grijs vinkje als dit voorwerp door iemand anders is geselecteerd en met een zwart vinkje als het door de gebruiker zelf is geselecteerd. Ook worden in deze functie voor elk voorwerp 2 onClickListeners aangemaakt, één voor de checkbox en één voor de naam van het item.

InfoActivity

De InfoActivity lijkt heel erg op de CreateActivity, maar er zijn ook enkele verschillen. Zo staan er hier naast alle EditText velden potloodjes waar de gebruiker op kan drukken om de bijbehorende gegevens van de groep aan te passen. Als de gebruiker degene is die de groep heeft aangemaakt staat er in de ActionBar een icoontje om de groep te verwijderen, anders staat er een icoontje om de groep te verlaten. Ook heeft degene die de groep heeft aangemaakt de mogelijkheid om nieuwe contacten toe te voegen aan de groep (dit wordt allemaal geregeld door een heel simpele if-else statement die gebruik maakt van het feit dat in de online database is opgeslagen of iemand de creator van een groep is).

SettingsActivity

De SettingsActivity bestaat op het moment praktisch alleen uit de onCreate functie waar de ActionBar en het menu worden aangemaakt en waar de uitlog knop wordt getoond, de layout hiervan hangt af van of de gebruiker is ingelogd met een account of met Facebook. Verder toont hij nog de naam en het e-mailadres/ID-nummer van de gebruiker uit de lokale database.

PHP

Achter onze app draait een webserver waarop enkele honderden regels aan PHP code staan om de communicatie met onze MySQL database mogelijk te maken. Deze php code is onderverdeeld in 5 secties, 1 voor elke table in de database. Dit zijn een login table, een events table, een icons table, een items table, en een details table (waar alle telefoonnummers van de gebruikers worden opgeslagen). Al deze tables spreken redelijk voor zich en aangezien de PHP code vrij repetitief is gaan wij hier nu niet te diep op in. Het principe achter deze code is dat er binnen elke sectie eerst wordt gekeken naar wat de meegegeven tag(create, remove, list, etc.)

is en aan de hand hiervan, mits alle benodigde velden ook in de POST request zijn meegegeven, een bepaalde database query uitvoert.

Ontwerpverantwoording

Wij hebben voor ons grafische ontwerp van de app veel inspiratie opgedaan uit veel gebruikte apps zoals WhatsApp. We hebben er voor gekozen om een minimalistische, maar ook zo overzichtelijk mogelijke, layout te creëren om de app zo gebruiksvriendelijk mogelijk te maken. Het ontwerp van de verschillende klassen is zo intuïtief mogelijk gedaan, alle verschillende activiteiten hebben natuurlijk een eigen klasse en er zijn een paar overkoepelende klassen om veelgebruikte functionaliteiten zo eenvoudig mogelijk te implementeren. Hieronder zullen wij 3 van onze ontwerp keuzes uitwerken.

Inloggen

Binnen onze applicatie zijn er 2 verschillende manieren waarop een gebruiker kan inloggen; met Facebook en met een account. Wij hebben hiervoor gekozen omdat het inloggen met Facebook voor veel mensen gemakkelijk is, zo hoeven ze niet weer een nieuw account aan te maken en kunnen ze gewoon zo aan de slag met de applicatie. Er zijn echter gebruikers die geen Facebook hebben of niet willen inloggen met hun Facebook, daarom hebben wij ervoor gekozen om ook nog een tweede manier van inloggen te maken, namelijk het inloggen met een gebruikersaccount. Op deze manier kan in principe iedereen gebruik maken van de app en heeft de gebruiker vrije keuze hoe hij of zij inlogt, dat is iets wat de mensen onder wie wij de app hebben verspreid een erg op prijs stellen.

Telefoonnummers

Wij hebben ervoor gekozen om gebruikers hun telefoonnummer in te laten voeren bij hun eerste login en de contacten te gebruiken uit de contactenlijst van de gebruiker. We hebben hier erg over zitten twifelen en we hebben dan ook verschillende manieren van uitnodigen van contacten uitgetoetst voordat we op deze oplossing zijn uitkwamen. Dit omdat we willen dat gebruikers in principe iedereen die ze kennen kunnen uitnodigen. de andere manier, via Facebook contacten en mailadressen, is een stuk minder gebruiksvriendelijk omdat iedereen dan moet doorgeven welk e-mailadres ze hebben gebruikt voor de app aan anderen. Bovendien zou iemand die met een gebruikersaccounts inlogt geen gebruik kunnen maken van het uitnodigen van contacten via Facebook. Dit heeft er uiteindelijk voor gezorgd dat wij op zoek zijn gegaan naar een universele manier van het uitnodigen van contacten, die voor zowel gebruikers die zijn ingelogd met Facebook als voor gebruikers die zijn ingelogd met een account hetzelfde is. Uiteindelijk kwamen we uit op deze oplossing.

Evenement Informatie

De InfoActivity wordt gestart als je klikt op de naam van de groep binnen de EventActivity. Dit hebben wij zo geïmplementeerd omdat dit ons een vrij intuïtieve plek, zeker aangezien dit ook

door WhatsApp wordt gebruikt. Toen wij na het inleveren van het product echter de applicatie onder wat mensen hadden verspreid kwamen wij erachter dat dit mogelijk een verkeerde ontwerp keuze is geweest, aangezien iedereen veel moeite had om dit scherm te vinden.

Reflectie

Ons ontwerpproces

Het maken van de app begon allemaal met het verzinnen van een (goed) idee. Het duurde best lang voordat we op een goed idee kwamen omdat we iets wilden ontwerpen wat enigszins uniek is, maar tegelijkertijd moet het wel in één periode kunnen worden gerealiseerd. Uiteindelijk kwam iemand op het idee om een gedeelde paklijst app te ontwerpen, dit leek ons allemaal een goed idee en dat werd bevestigd toen we gingen kijken naar verwante apps en het bleek dat de er weinig (goede) concurrentie is. Het idee wat we hadden verzonnen was één van de positieve aspecten van het project.

Vervolgens moesten we het gaan uitwerken, we gingen bepalen welke eigenschappen en functies de app dient te hebben. We kwamen op een lijstje met noodzakelijke eigenschappen en optionele eigenschappen. We hadden er wel vertrouwen in dat we dit allemaal konden realiseren aangezien het niet hele ingewikkelde opties waren (het scheelt enorm dat we niet met de android grafische library hoeven te werken). Een gevolg daarvan was dat we weinig technische problemen hadden.

Het ging wel een beetje mis toen we daadwerkelijk de app moesten maken, aangezien we geen goede taakverdeling hadden gemaakt. Daardoor heeft Maarten Carsjens het grootste deel van de app geprogrammeerd. Dit is dan ook de negatieve ervaring van ons project. Over het eindresultaat zijn we redelijk tevreden, de noodzakelijke eigenschappen zijn in ieder geval allemaal uitgewerkt. Ook hebben we er voor gezorgd dat de app intuïtief en gebruiksvriendelijk is en dat is erg belangrijk aangezien de enige directe concurrent slecht scoort op die eigenschappen. De reden dat we niet helemaal tevreden zijn is dat er nog enkele bugs in de app zitten en dat we de optionele eigenschappen (die niet heel ingewikkeld zijn om te maken) niet hebben kunnen implementeren.

Bereikte doelen

De doelen van deze cursus zijn zeker wel bereikt. We hebben veel geleerd over het ontwerpen van een app, door van tevoren te bepalen waaraan de app moet voldoen en te kijken naar verwante apps. Ook het realiseren van een app was interessant aangezien niemand van ons eerder met Android had gewerkt, we hebben dus veel geleerd over Android en het verschil tussen Android en gewone JAVA. Het evalueren van een applicatie op een systematische manier was compleet nieuw voor ons. Verder is presenteren natuurlijk een vaardigheid die erg

belangrijk is, hier is altijd wel iets op aan te merken en de feedback die we hebben gekregen is dus ook erg handig.

Conclusie en tips voor de volgende keer

We kunnen dus concluderen dat het eindresultaat niet slecht is maar dat het veel beter had kunnen zijn als we in het begin van de periode een betere taakverdeling hadden gemaakt. Dat is dan ook een tip voor de volgende keer: maak vooraf al een goede taakverdeling. Als je eenmaal een taakverdeling hebt gemaakt dan is het ook makkelijk om deadlines te maken voor wanneer bepaalde delen van het project af moeten zijn. Dat heeft als gevolg dat je vervolgens niet op de laatste dag nog bezig bent met het implementeren van functies.