

Phapp
De app voor N.S.R.V. Phocas Nijmegen

Wietze Mulder (s4557557) David Korsman (s4619579)
Caja Sieben (s4587642) Patrick Hilhorst (s4577434)
Groepsnaam: ()xxxxxx{:.....>

17 juni 2016

Inhoudsopgave

1	Voorwoord	3
2	Beschrijving	4
2.1	Inleiding	4
2.2	Productverantwoording	4
2.3	Specificaties	4
2.3.1	Loginpagina en hoofdpagina	4
2.3.2	Hoofdmenu	4
2.3.3	Foto's	4
2.3.4	Weer	5
2.3.5	Smoelenboek	5
2.3.6	Contact	5
3	Ontwerp	6
3.1	Globaal ontwerp	6
3.2	Detailontwerp	6
3.2.1	Fragments	6
3.2.2	Networking	6
3.2.3	Package structure	6
3.3	Ontwerpverantwoording	6
4	Reflectie	7
4.1	Proces	7
4.2	Ervaringen	8
4.3	Conclusie	8

1 Voorwoord

Dit document is bedoeld ter toelichting van het ontwikkelingsproces van de app ontworpen voor Research and Development. We zullen het product beschrijven, een beeld schetsen van hoe het product tot stand gekomen is, en enkele ontwerpbeslissingen bespreken. Ook wordt de werkwijze gereflecteerd. We zullen beginnen met de beschrijving van de app, daarna doorgaan met het ontwerp, en tot slotte een reflectie bespreken.

2 Beschrijving

2.1 Inleiding

Onze app is bedoeld voor leden van Phocas. Phocas heeft een website, maar gezien er interesse was in een duidelijke app die dezelfde informatie bevat, hebben we besloten om de Phapp te ontwikkelen. De app bestaat uit een inlog- en een hoofdscherm. Het hoofdscherm kan wisselen tussen de verschillende onderdelen.

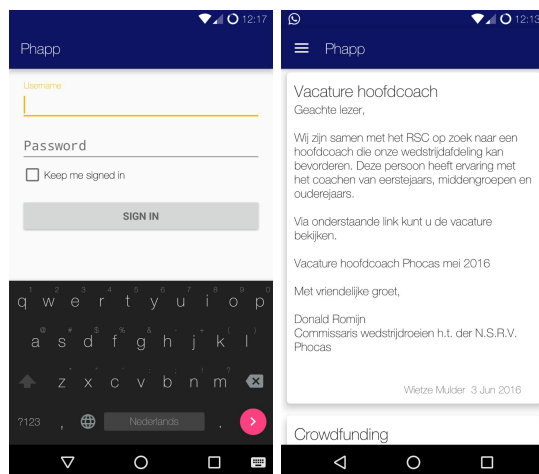
2.2 Productverantwoording

Wij hebben deze app ontwikkeld omdat er nog geen dergelijke app beschikbaar was voor Phocas. Andere verenigingen hebben wel al een dergelijke app maar daartoe hebben anderen geen toegang. Tot nu toe was de enige manier om via een smartphone van de Phocas-website gebruik te kunnen maken, via een browser. Daarom was er behoefte aan een app. Het bestuur van Phocas had ons gevraagd om een app te maken.

2.3 Specificaties

Onze app bestaat uit de volgende pagina's:

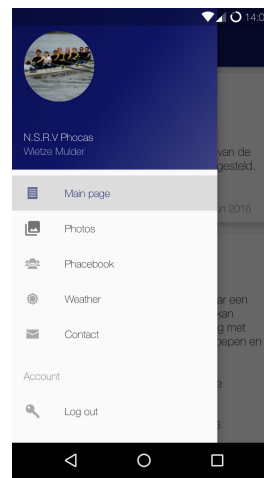
2.3.1 Loginpagina en hoofdpagina



Dit zijn in essentie de twee activiteiten van de app. Bij het eerste gebruik krijgt de gebruiker het inlogscherm te zien zodat hij kan inloggen en eventueel zijn inloggegevens kan opslaan. Als de gebruiker correcte inloggegevens heeft ingevuld komt hij bij de Main activity. Deze bestaat voornamelijk uit een fragment die het nieuws laadt. Daarna is er ook nog een titel en een meknop voor het hoofdmenu. De gebruiker kan

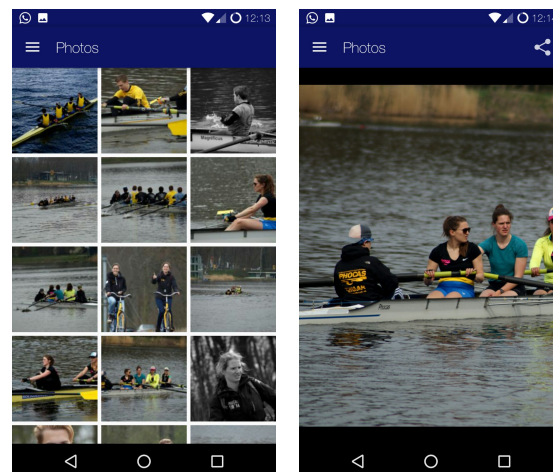
ook het nieuws verversen als hij naar beneden swipet.

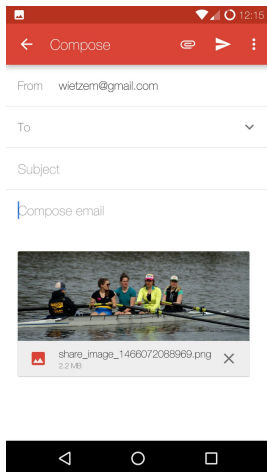
2.3.2 Hoofdmenu



Via het hoofdmenu kan de gebruiker naar alle delen van de app navigeren. het menu bestaat uit een lijst met iconen en tekst van de verschillende onderdelen. Ook staat bovenaan de avatar van de ingelogde gebruiker en zijn naam en e-mail. Ten slot is er in het menu ook de mogelijkheid om uit te loggen.

2.3.3 Foto's





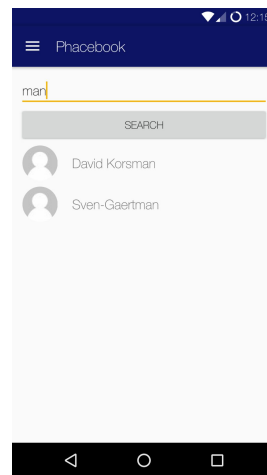
Het Foto's-onderdeel is een galerij van foto's die geladen worden van de site. Men kan door de foto's heen scrollen en men kan de foto's opnieuw verversen op dezelfde manier als bij het nieuws. Wanneer men op een foto klikt, komt de foto in een vergrote weergave waar de achtergrond zwart wordt gemaakt en de foto op volledige resolutie wordt geladen. In deze modus kan de gebruiker de afbeelding in- en uitzoomen. Ook is het mogelijk om naar de andere foto's te navigeren door te swipen. Ook kan de foto dan gedeeld worden met andere apps.

2.3.4 Weer



Het Weer-onderdeel is een pagina waar de gebruiker het huidige weer van Nijmegen te zien krijgt door middel van bondige textuele informatie en een afbeelding. De weergegevens worden geladen van buienradar.nl.

2.3.5 Smoelenboek



In het Smoelenboek kan de gebruiker zoeken naar gebruiker in de database van de site. De gebruiker zoekt met een regel tekst en kan op enter drukken of de zoekknop gebruiken om te zoeken. De resultaten worden weergegeven in een lijst met voor elk resultaat een avatar en zijn of haar naam.

2.3.6 Contact

Op de contactpagina wordt de pagina Contact van de site geladen en weergegeven. Daar staan gegevens zoals het adres en contactgegevens van het bestuur van Phocas.

3 Ontwerp

3.1 Globaal ontwerp

Men begint in het inlogscheren en wordt daarna doorverwezen naar het hoofdscheren. Via het hoofdscheren kan men wisselen tussen de verschillende onderdelen van de app.

3.2 Detailontwerp

In de onderliggende Java-code is elk onderdeel gesplitst in klassen. klassen hebben vaak een Adapter voor het weergeven van de geladen data.

3.2.1 Fragments

Onze app is wat activiteiten betreft opgebouwd uit een `LoginActivity` en een `MainActivity`. Voor de verschillende pagina's in de app maken we gebruik van Fragments die we in de `MainActivity` plaatsen wanneer ze in het menu aangeropen worden.

De `MainActivity` is gemaakt van een `Navigation Drawer Activity`. Hierin staat een `<include>` die dynamisch kan worden aangepast. Er is een `HashMap` die alle Fragments bevat (in de vorm van `FragmentHolder`-objecten, die zowel een `Fragment` als een titel onthouden), en wanneer je een pagina voor de eerste keer na het opstarten van de app opent, wordt een nieuw `Fragment` gemaakt, en in een nieuwe `FragmentHolder` toegevoegd aan deze `HashMap`. De volgende keer dat je deze pagina opent, wordt dezelfde `Fragment` opnieuw gebruikt.

3.2.2 Networking

Voor netwerk-gerelateerde zaken hebben we een API-klasse gemaakt. Deze klasse bestaat uit verschillende methoden om specifieke gegevens op te vragen. Deze methoden maken bijna allemaal gebruik van

de `private void getRequest(String url, Response.Listener<String> listener)`.

Hierbij is `url` een API-request (zoals "users/me") en `listener` is een response listener die uitgevoerd wordt zodra er een response is van de server.

3.2.3 Package structure

Alle klassen zijn zo ingedeeld dat er onderscheid wordt gemaakt tussen de UI, netwerkcommunicatie en de communicatie tussen de klassen. De `fragments`-package is onderverdeeld in de verschillende fragments. de Activities zijn gescheiden van de Fragments zodat duidelijk is welke Activities er zijn.

- haakjeopenen.phapp
 - activities
 - fragments
 - * contact
 - * news
 - * phacebook
 - * photos
 - models
 - net
 - util
 - widgets

3.3 Ontwerpverantwoording

Wij hebben voor deze beschreven structuur gekozen omdat dit ook gebruikt wordt in andere bekende apps. Deze structuur wordt bijvoorbeeld ook gebruikt in veel Google-apps, zoals Google Play en Gmail. Omdat Android gebruikers vaak al bekend zijn met deze structuur, hebben wij die in grote lijnen aangehouden. Er is ook zoveel mogelijk Material Design toegepast aan het ontwerp van de Phocas-app. Dit is terug te zien in de swipe-mogelijkheden, de `RecyclerView` en de `Cardview` die in de app zijn verwerkt.

4 Reflectie

4.1 Proces

Proces	Resultaten	Ondertussen
Welke app		Communicatie Phocas
Welke onderdelen	Idee app	
Opbouw	Outline app	
Wordpress Inlogscher DebugActivity Development site	Netwerk van app	
Nieuwspagina/Main Album/Contact Weer	Communicatie Phocas-site	
Verbeterd	Eerste versie app	
	Phapp is af	

We hebben tijdens dit vak en tijdens het maken van het project veel geleerd. We hebben met name geleerd om apps voor Android te ontwikkelen. Ook hebben we iets beter leren plannen, en meer ervaring gekregen met samenwerken in een groep. Ook hebben we geleerd te werken met Git, wat we tijdens het project vaak hebben gebruikt. Verder hebben we leren werken met Wordpress en de Wordpress Rest-API. We weten nu hoe het ontwikkelingsproces van een Android-app eruit ziet. Ten slotte hebben we geleerd gebruik te maken van libraries, met name Volley en Picasso.

De technische problemen en moeilijkheden waar we tegenaan zijn gelopen zijn als volgt:

- Authenticatie met de Wordpress-website (was eerst moeilijk om in te bouwen, door slechte documentatie)
- Refreshen door te swipen
- Het internet is niet perfect; daardoor zijn response listeners nodig om te zorgen dat de main thread niet vastloopt
- Geneste response listeners waren nodig, omdat alle posts geladen moesten worden, en voor elke post apart de gebruikersnaam van wie het bericht geplaatst had. Hierdoor werden de posts in een onbepaalde volgorde geladen

- Bij het delen moest de foto lokaal opgeslagen worden. Hiervoor zijn Android-permissies vereist.
- Uitvinden hoe Picasso gebruikt kon worden om een afbeelding in een Bitmap o.i.d. te laden, en niet meteen in een ImageView.
- Fragments werden steeds opnieuw gecreëerd
- Bij trage verbinding kon je niet zien dat foto's geladen werden (nu is er een laadicoontje)
- Ingebouwde code uit Android Studio bij vooral de standaard-LoginActivity, met emails, code om emails automatisch aan te vullen vanuit de contactenlijst, waardoor er best veel code stond die we niet nodig hadden, het kostte best wat tijd om dat allemaal weg te halen.

Problemen die nog steeds aanwezig zijn:

- Schermrotatie werkt niet volledig
- De afbeelding op de weerpagina, het is niet makkelijk om nearest-neighbor te doen hiermee, dus we kregen telkens een wazig zonnetje. Ook is niet bekend welke afbeeldingen allemaal gebruikt worden bij Buienradar, dus je kunt ook niet zomaar je eigen afbeeldingen gebruiken. Verder heeft

de afbeelding in een VirtualBox-VM ineens zijn transparantie verloren, waardoor het een zwarte achtergrond krijgt en het er dus niet uitziet.

- De app werkt alleen nog op Android Marshmallow, waarschijnlijk door de share-knop.
- Er kan een soort van parameter-injectie gedaan worden in het Smoelenboek/Phacebook, door in het zoekveld bijvoorbeeld in te typen: `&per_page=2`. Dit is verder niet gevaarlijk.
- Het Smoelenboek zoekt ook op email-adres, dus als je `com` invult vind je veel mensen. Deze fout ligt echter met name bij de Wordpress Rest-API.

4.2 Ervaringen

Het was over het algemeen een positieve ervaring om aan dit project te werken. Dat komt doordat het in interessante uitdaging was, omdat we nog niet wisten hoe Android-apps geprogrammeerd moesten worden. Ook hadden we nog niet zoveel ervaring met het in een groep werken aan een programmeerproject. We zijn erachter gekomen dat de tweede verdieping van

het Mercator-gebouw een prima plek is om in stilte aan het project te werken.

Waar we minder tevreden over zijn is dat Patrick Hilhorst niet genoeg inzet heeft getoond om goed bij te dragen aan het project, terwijl wij zijn kracht wel goed hadden kunnen gebruiken. Ook was hij totaal afwezig bij het presenteren, het usability-onderzoek en het maken van het verslag.

Het draagt ook niet bij dat we redelijk laat aan het programmeren zijn begonnen, wat opleverde dat we in korte tijd veel tijd moesten besteden aan de app. Maar toen we erachter kwamen dat de rubric van de app niet te veel verwachtte, waren wij opgelucht dat we op de goede weg zaten.

4.3 Conclusie

We kunnen concluderen dat we veel hebben geleerd met over het algemeen een positieve ervaring. Bij een volgend project zullen we onszelf adviseren bijtijds een planning met deadlines op te zetten en natuurlijk eerder te beginnen. Wij zullen ook eerder mensen erop aanspreken als ze te weinig leveren.