

TOPWoord

Alex Hlushenok, Dion van de Vooren, Martin Huyben

28-06-2013

1 Voorwoord

Dit document is bedoeld om de lezer inzicht te geven in het ontwikkeling van TOPwoord, de achterliggende ontwerpbeslissingen en de keuzes die ons team heeft gemaakt bij het maken en verantwoorden van de app. In dit document zal eerst een omschrijving van de app komen met globale doelen die we hadden en die zijn bereikt, naast de specificaties die we voor ogen hadden. Daarna wordt het ontwerp van de app toegelicht en als laatste is er een reflectie op het verloop van het project en enkele ontwerpbeslissingen.

2 Beschrijving

2.1 Inleiding

TOPWoord is een Android applicatie die helpt woordjes leren. Het is gebaseerd op het feit dat de beste manier van leren herhaling over een lange periode is, i.p.v. alles in een keer leren. Het laat je verschillende keren per dag over een langere periode woorden herhalen door middel van pushberichten, die leiden naar een korte overhoring van een paar woorden. Door dit regelmatig te doen, wordt de informatie beter geleerd en is het minder saai en belastend voor de gebruiker. De app ondersteunt meerdere manieren van leren, heeft opties om de gebruiker op gezette tijden niet te storen en is gestroomlijnd en simpel vormgegeven. Het toevoegen van woordjes kan handmatig door ze in te typen, of een lijst op te halen van je telefoon. Nadat woordlijsten zijn toegevoegd, zijn ze eenvoudig te wijzigen of te verwijderen om de orde in de app te behouden.

2.2 Productverantwoording

Er bestaan vergelijkbare apps en sites, ook met het doel woordjes te leren. Voorbeelden zijn Wtrs.nl, teach.nl, memorize.nl, Wrts mobile en Anki, maar geen van deze diensten biedt het leren verdeeld over een langere periode zoals wij dat doen. Bij de meeste diensten stop je pas als je alle woordjes uit een lijst geleerd hebt. Teach.nl biedt wel leren over een langere periode aan, maar aangezien het een site is, kan je niet altijd en overal leren en is het niet zo flexibel als een smartphone. Ook is onze app veel doelgerichter en heeft het geen overbodige functies die in de weg zitten.

- Actief leren met pushberichten, de app gaat naar de gebruiker toe in plaats van andersom.
- Simpel en overzichtelijk ten opzichte van alle genoemde voorbeelden

2.3 Specificaties

2.3.1 Functionele eisen

- Push berichten sturen op basis van datum toets en het aantal woorden. De app gebruikt een algoritme om de frequentie van de pushberichten te regelen, zodat ze niet in de weg zitten als het niet nodig

is. De gebruiker krijgt korte en duidelijke berichtjes met enkele woorden om te leren. Is dat gedaan, verdwijnt de app weer tot het nodig is om nog een bericht te sturen.

- Overhoren (Strikt en los): TOPWoord biedt de mogelijkheid om zowel op de traditionele manier te overhoren, door het juiste woord in te typen, alsook de functie om de woordjes te leren zonder te typen: de gebruiker geeft aan of hij het antwoord in gedachten had en de app reageert hierop.
- Lijst toevoegen/aanpassen: Lijsten kunnen op elk gewenst moment toegevoegd, aangepast, uitgebreid, ingekort of verwijderd worden door middel van een simpele interface.
- Niet storen tijd: De gebruiker kan een tijd instellen waarin de app niet zal storen met pushberichten, bijvoorbeeld slaaptijd. Bij het berekenen van de frequentie van de pushberichten houdt TOPWoord ook rekening met deze tijd.
- Vooruitgang: de vooruitgang van de gebruiker in de betreffende lijst van woord of de huidige overhoring wordt nauwkeurig en duidelijk bijgehouden door middel van visuele feedback. De gebruiker ziet gelijk hoeveel woorden hij of zij goed heeft en hoe lang de overhoring nog duurt.

2.3.2 Niet-functionele eisen

- De app moet snel zijn(opstarten): Dit is gerealiseerd, app start erg snel op.
- De app moet ook in latere android versies nog werken: Niet uitvoerig kunnen testen, maar op Android 2.3 en latere versies werkt TOPWoord bij ons goed.
- De app moet eenvoudig te bedienen zijn: Geen overbodige functies, simpele interface, weinig knoppen, alleen noodzakelijke data en duidelijke instructies dragen allemaal bij aan de simpele bediening.

2.3.3 Use Case uitwerking: UC04

Deze use case is niet de langste of de meest uitgebreide, maar het idee erachter en de functie die het vervult speelt een sleutelrol in onze app. Notificaties zijn het speerpunt van TOPWoord, dus is gekozen om deze use case uit te werken.

- UC04: Krijg melding

Omschrijving: De actor krijgt een melding om een set woorden te leren.

Primaire Actor: TOPWoord. TOPWoord bepaalt wanneer de gebruiker een melding krijgt.

Trigger: Push-bericht van TOPwoord

Basic flow of Events:

1. TOPWoord stuurt een bericht naar gebruiker Anne dat ze nog een deel van een lijst moet leren voor Frans voor overmorgen. Anne heeft de afgelopen dagen normaal het schema van de app gevolgd en hoeft dus nog maar een paar woordjes te leren.
2. Anne opent het bericht, want ze heeft nog wel een paar minuutjes om de woordjes te leren.
3. TOPWoord start UC03 Overhoren en verwijdert de notificatie.

Alternate paths:

(VARIATION)

1. Anne heeft geen tijd om de woordjes te leren, dus verwijdert ze de notificatie om later te leren. TOPWoord zal de set woordjes die hoorden bij de notificatie gewoon opnieuw inplannen voor het proefwerk.

Pre-conditions:

- Er is ten minste 1 lijst geselecteerd bij de 'te leren lijsten'.
- De gebruiker heeft niet aangegeven niet gestoord te willen worden.
- Als er een datum voor de toets is ingevuld, is deze nog niet verstreken.

3 Ontwerp

3.1 Globaal

- Lijst: Een vector van woordenparen en extra informatie erover, o.a. de toetsdatum, de leermethode, de leerrichting en de woordfrequentie.
- Lijsten: Een verzameling van alle lijsten. Dit is een aparte klasse die in een Singleton architectuur is gemodeleerd.
- Globale Instellingen: Globale instellingen van de applicatie.
- Lijstspecifieke Instellingen: Hier kunnen per lijst o.a. de toetsdatum, de leermethode, de leerrichting en de woordfrequentie gewijzigd worden.
- Overhoren: Voor een bepaalde set woorden wordt het vraagwoord gevraagd en moet de gebruiker een antwoord geven.
- Lijst Invoeren: Hier kan een lijst ingevoerd worden, handmatig of via de file browser van de telefoon. Ook wordt hier de toetsdatum, de leermethode, de leerrichting en de woordfrequentie gekozen.
- Klok: Klok bepaalt met de gebruikte informatie uit de lijsten wanneer er push-berichten worden verstuurd.

De samenhang van de componenten is als volgt: Lijst is vrij duidelijk: het is een verzameling van woordenparen met extra informatie zoals toetsdatum. Lijsten is een verzameling van een aantal Lijst objecten met de lijst en extra informatie erin. Dit dient als opslag voor de informatie. Lijsten wordt gebruikt door Overhoren om de juiste informatie op te halen en wordt aangeroepen door Lijst Invoeren om een nieuw ingevoerde lijst op te slaan. Verder wordt er informatie aangeroepen uit een specifiek Lijstobject in Lijsten en uit Globale Instellingen door de Klok om te bepalen wanneer pushberichten worden verstuurd. Ten slotte kan met Lijstspecifieke Instellingen de extra informatie in de Lijst objecten worden gewijzigd, waarbij Lijsten wordt aangeroepen om het juiste Lijst object door te geven.

3.2 Detail

Bij het uitvoeren van UC04 worden verschillende klassen aangeroepen. Klok is de hoofdklasse, dat met een verzoek aan Lijsten een specifieke lijst wil ophalen, waarvan is aangegeven dat deze lijst moet worden overhoord. Deze waarde wordt opgeslagen als extra informatie. Als dit het geval is, vraagt Klok aan het Lijst object wat de datum is tot aan de toets en hoeveel woorden nog niet zijn overhoord. Met deze informatie wordt berekend dat het handig is om nu een pushbericht aan te maken en te overhoren en dat de huidige tijd niet in de 'niet storen' zone ligt. Dit laatste is snel geregeld met een verzoek aan de Globale Instellingen met de vraag wanneer de rustperiode is. Klokt vergelijkt dit met de huidige tijd en constateert dat het toegestaan is om een pushbericht te sturen. Klok notificeert Overhoren om een set woorden aan te maken om te overhoren en maakt zelf een pushbericht aan om te sturen als de set woorden klaar is om overhoord te worden. Als de gebruiker klikt op de notificatie, wordt hij of zij herleid naar de Overhoren klasse voor UC03. Mocht de gebruiker kiezen om het bericht te verwijderen, zal Klok dit meenemen en een nieuwe planning maken voor de pushberichten.

3.3 Verantwoording

Het ontwerp van TOPWoord is gebaseerd om simpelheid en snelheid. Het is goed omdat het niet te ingewikkeld is. Een vrij groot hieraan gerelateerd technisch probleem was het opslaan van de lijsten die net waren ingevoerd. Android heeft hele andere opvattingen over het opslaan van gegevens dan Java. Hierdoor was het een tijd erg moeilijk om het goed te krijgen, tot we wat hulp kregen van Bas met het probleem. Het probleem wat we hadden was op te lossen met een speciale klasse, een Singleton. Hiervan kan maar 1 instantie worden gemaakt, die ook altijd wordt aangeroepen zonder te kopiëren, zoals meeste Android objecten doen als er intents worden doorgestuurd. Met een Singleton was dat probleem dus weg en konden we gewoon de lijst van lijsten opslaan en bij de volgende start van de app weer aanroepen. Een alternatief hiervoor was bijvoorbeeld SQLite, maar deze oplossing is niet gekozen omdat een SQL database alleen primitieve datatypes kan opslaan, waar ons doel het opslaan van een lijst van objecten was. Dit was niet makkelijk op te lossen met SQLite, dus is er gekozen voor de Singleton.

Synchronisatie was ook een onderwerp van discussie bij het team. Het inbouwen van een link met bekende sites om automatisch lijsten op te halen van die sites was een moeilijke en uitdagende opgave. Na een aantal weken zijn we toch van het idee afgestapt omdat het bijvoorbeeld te veel werk was om realistisch te kunnen oplossen. Ook was het feit dat we erg veel moeite hadden met het wennen aan de Android structuur bijgedragen aan het feit om de functie te laten vallen. Het positieve aspect van het ontbreken van de feature is dat de app minder bug-gevoelig is en het geen afhankelijkheid heeft van externe bronnen die kunnen uitvallen.

Het usabilityonderzoek heeft niet veel bijgedragen aan de verdere ontwikkeling van de app omdat het vrij laat in de ontwikkelcyclus naar voren kwam, maar ook vooral omdat het probleem van TOPWoord vooral ligt in de crashes en bugs die optreden. De interface en gebruiksvriendelijkheid waren degelijk, zoals het usability onderzoek van dev5 laat zien, maar de app liet erg veel crashes zien die niet voorkwamen op onze eigen systemen en de emulator. Dit is het grootste en veruit het belangrijkste probleem van de app. Maar door tijdsgebrek en de planning van het usabilityonderzoek was er verder geen tijd om de app crashproof te maken.

4 Reflectie

4.1 Positief

- De app doet wat het moet doen: woordjes overhoren werkt, notificaties zijn aanwezig en alle aanwezige functies in de applicatie werken.
- Onze planning voor het maken van de app was wel goed gemaakt, maar er werden enkele zaken onderschat: complexiteit Android voor onervaren ontwikkelaars, enkele problemen met de developerssoftware Eclipse en verdere moeilijkheden aan de technische kant. Gelukkig waren ook deze problemen geen ramp voor de ontwikkeling. We hebben namelijk heel veel extra zaken geschrapt en gekozen om te focussen op de kernaspecten zoals het overhoren en de notificaties. De planning liet dit toe, omdat er tijd was ingepland om die extra zaken in te bouwen. We zijn niet over de planning heen gegaan.
- Het oplossen van de opslag probleem met de Singleton was een goede oplossing omdat het ons precies bracht wat wij wilden hebben: opslag van objecten. Dit is een technisch probleem dat ons veel tijd heeft gekost.
- Notificaties waren verrassend snel opgelost vergeleken met bijvoorbeeld het opslagprobleem. Na wat documentatie en veel trial en error is het gelukt om het in orde te krijgen, waarmee onze app af was wat betreft de kernzaken.

4.2 Verbeterpunten

- Het is wel duidelijk dat om een dergelijk product te maken we beschikking moeten hebben over meer Android apparatuur om het op te testen, wat het usabilityonderzoek van onze app duidelijk liet zien. Wij hadden de beschikking over de emulator en enkele androidversies waar de app goed op werkte, maar toen anderen (dev5) ermee aan de slag gingen, bleek het veel te crashen en bugs te bevatten die wij nog nooit gezien hadden.
- We kunnen in het vervolg beter vaker vragen hoe iets moet, want zonder de expertise van een ontwikkelaar in dit gebied(Bas) was het opslagprobleem misschien wel minder mooi of helemaal niet opgelost. Het is dus handig om vaker vragen te stellen.
- Minder idealistisch nadenken wat betreft functies. Veel van de functies die we voor ogen hadden zijn niet in de app gekomen door tijdsgebrek. Dit is iets waar aan gewerkt kan worden. Dit is ook voor een groot deel veroorzaakt door ons gebrek aan Android-ervaring, wat extra tijd kostte om in te halen.

4.3 Negatief

Echt negatieve punten zijn er niet bij dit project. Misschien dat we in het vervolg meer functies kunnen inbouwen in een kortere tijd, geholpen door onze Android ervaring. De crashgevoeligheid van de app is een verbeterpunt, maar kan voor deze app als negatief aspect worden beschouwd.

4.4 Conclusie

De meeste verbeterpunten staan boven, maar over het algemeen kan het worden samengevat als de volgende keer sneller de app maken, minder tijd besteden aan onrealistische functies en vooral veel testen op veel verschillende apparaten. Aan het concept mankeert verder weinig, dat was goed uitgevoerd. Samenwerking zit ook prima in het team, net zoals de onderlinge communicatie. Meer uiterlijke verzorging kan natuurlijk ook geen kwaad, al bestaat de kans dat dit gaat afleiden van de app zelf.

5 Opmerkingen

Door aanhoudende problemen met LaTeX en plaatjes zijn de laatste niet aanwezig in het verslag.