

Een app als vaste begeleider

Eindverslag Research & Development

Michael Jansen, Jeremy Timessen, Jonathan Seesink

27 juni 2013

1 Voorwoord

In dit document zullen we spreken over de Android applicatie die we in het kader van het vak "Research & Development" gemaakt hebben. We verantwoorden ons idee, de specificaties, onze ontwerpbeslissingen en het eindproduct. Ten slotte zullen we in de reflectie terugblikken op de samenwerking binnen de projectgroep, gemiste kansen aanstippen en zullen we onze visie op de toekomst van de verdere ontwikkeling van de app uitspreken.

Inhoudsopgave

1	Voorwoord	1
2	Beschrijving	1
2.1	Inleiding	1
2.2	Productverantwoording	2
2.2.1	Boodschappenlijst-app	2
2.2.2	Time-Tracking voor studenten	2
2.2.3	Voorlichtings-app	3
2.2.4	Beslissing	3
2.3	Specificaties	4
2.3.1	Integrated Use-Case Diagram	4
2.3.2	Use Case: Geef feedback	4
2.3.3	Niet-functionele eisen	5

3	Ontwerp	6
3.1	Globaal ontwerp	6
3.1.1	Codeinvoer	6
3.1.2	Lijst van activiteiten	6
3.1.3	Beschrijving van een activiteit	6
3.1.4	Gebruikersgegevens	6
3.1.5	Samenvatting	6
3.1.6	Locatiewijzer	7
3.2	Detailontwerp	7
3.3	Ontwerpverantwoording	8
3.3.1	Google Maps	8
3.3.2	Downloaden van gegevens	9
4	Reflectie	10
4.1	Samenwerking	10
4.2	Gaten	10
4.2.1	Locatiewijzer	11
4.2.2	Samenvatting	11
4.2.3	Ondersteuning schermgroottes	11
4.2.4	Accenten	11
4.2.5	Buttons	12
4.2.6	Foto's	12
4.3	Kansen	12
4.3.1	Gadgets binnen de app	12
4.3.2	iOS-versie	13
4.3.3	Webinterfaces	14
4.3.4	Communicatie universiteit	14

2 Beschrijving

2.1 Inleiding

De Nijmeegse Tweedaagse is een voorlichtingsactiviteit van de Radboud Universiteit te Nijmegen voor VWO-leerlingen van klas 5 of 6. De activiteit vindt altijd plaats in de herfstvakantie (regio zuid). Het evenement is verdeeld over twee dagen. De student maakt niet alleen kennis met de toekomstige studie maar ook met de universiteit en het studentenleven. De deelnemers volgen

proefcolleges, practica, eten in studentenhuizen en daarnaast hebben ze de mogelijkheid om te sporten en te feesten. Er worden veel verschillende activiteiten aangeboden. Het grootste deel van het programma wordt centraal door de universiteit verzorgd. Het studie-gerelateerde programma wordt georganiseerd per opleiding.

Met de RU Tweedaagse app bieden wij deelnemers ondersteuning voor de voorlichtingsdagen. Met behulp van onze app kunnen toekomstige studenten zich wegwijs maken op de universiteit, overzichtelijk zien welke activiteiten ze de komende dagen gaan doen of hebben gedaan en beschikken zij over alle informatie om de tweedaagse tot een succes te kunnen maken. Daarnaast nodigen we de gebruikers op een aantrekkelijke manier uit om feedback te geven op elk onderdeel.

2.2 Productverantwoording

We hebben vanaf het begin van dit vak nagedacht over wat voor een app we zouden kunnen ontwikkelen. Iedereen uit de groep had ideeën (boodschappenlijst, dagboek, time-tracking voor studenten, voedings-app, sms-filter, lyricsviewer, broodjes-app, etc.). Ons viel op dat er al voor alles wel iets bestaat. Na aanleiding van een cursusbijeenkomst hebben we dieper nagedacht over de drie app-ideeën die ons het leukst leken. We hebben eigenschappen in kaart gebracht (daarbij onderscheid gemaakt wat wij als noodzakelijk en wat als gewenst achten), vastgelegd wat wij met zo een app zouden willen bereiken (het doel), wat onze inspiratiebron voor dit idee was en waarin wij een meerwaarde van de app zien t.o.v. alles wat er al op de markt bestaat.

2.2.1 Boodschappenlijst-app

Er zijn diverse apps beschikbaar die een gebruiker helpen bij het maken van boodschappenlijsten (enkele voorbeelden: “Grocery Gadget”, “Mighty Grocery”, “ZipList”, “iShopNCook”). Bij elkaar hadden ze alle bijzondere functies die wij in onze mogelijke app wilden integreren al.

2.2.2 Time-Tracking voor studenten

Een groepslid merkte op dat mensen, maar vooral ook studenten, vaak niet in de gaten hebben waar de tijd blijft. Een app die helpt om een tijdrooster in te vullen zou de gebruiker kunnen helpen om meer inzicht te krijgen

in zijn dagbesteding. Hierbij zouden bijvoorbeeld ook statistieken te pas kunnen komen. Op de markt bestaan diverse apps t.b.v. Time-tracking. Het gaat daarbij voornamelijk om het vasthouden hoeveel werkuren iemand aan een activiteit besteed heeft. De tijden worden dan ook direct gekoppeld aan geldbedragen. Het wordt duidelijk dat deze apps de nadruk leggen op werknemers en/of projectleiders (enkele voorbeelden: “Xpert-Timer Time Tracking”, “Timesheet – time tracking”, “Timesheet Time Meter”).

2.2.3 Voorlichtings-app

In het college uitte een docent de vraag naar een app die deelnemers begeleid tijdens een voorlichtingsactiviteit van de universiteit. Er bestaan reeds apps die te maken hebben met voorlichtingsactiviteiten voor de introductie en de open dag. Het bedrijf dat deze apps gemaakt heeft blijkt typische massa-software te ontwikkelen (de open dag apps van andere universiteiten zoals de Universiteit Utrecht, de universiteit Leiden, Avans hogeschool en apps van diverse andere onderwijsinstellingen zijn van dezelfde ontwikkelaar)¹ en wij nemen aan dat het bedrijf dus slechts het uiterlijk per universiteit aanpast. De apps voor de Radboud Universiteit hebben wij gedownload en getest. Het bleek dat de apps slechts basisfunctionaliteiten bieden. De apps voorzien wél in de behoefte aan informatie, maar de locatiewijzer is minimaal en er is daarentegen geen mogelijkheid om feedback te geven. Er is tevens geen mogelijkheid om het eigen dagprogramma automatisch te laten genereren. Wij zijn van mening dat zo een app niet bijdraagt aan het doel om deelnemers van een voorlichtingsactiviteit voor de universiteit te enthousiasmeren.

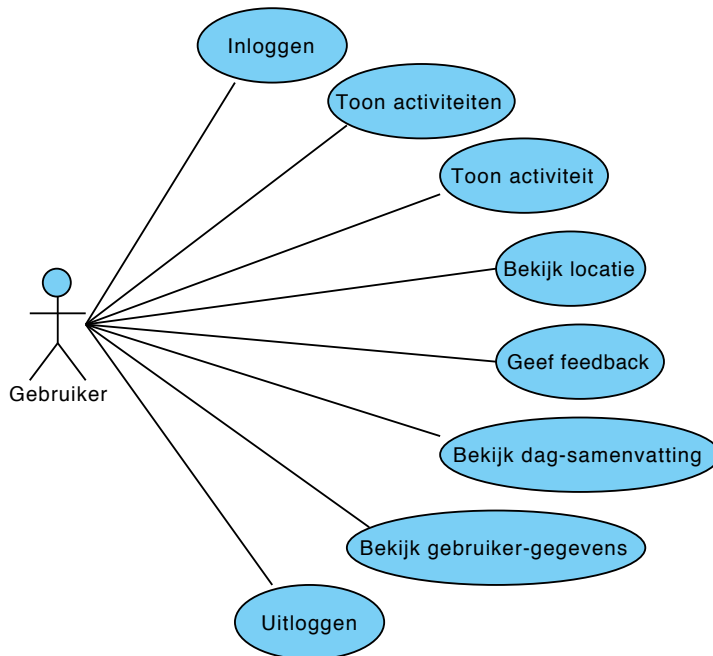
2.2.4 Beslissing

Het viel op dat groepsleden heel erg enthousiast waren over hun eigen idee. Dat belemmerde een gefundeerde beslissing. Het idee van de voorlichtingsactiviteit-app werd niet door de groepsleden verzonnen, maar kwam van een persoon buiten de groep (een docent). Dit hielp ons om dit idee objectief te beoordelen. Hoewel we op dat tijdstip het meest enthousiast waren over de eerste twee ideeën, bleek dat het laatste idee ons de meeste vrijheid zou bieden om de app een eigen invulling te geven. Hier zagen wij de mogelijkheid om een meerwaarde te creëren door een eigen app te ontwikkelen.

¹<https://play.google.com/store/apps/developer?id=MissMarvel%27s&zstart=0&num=12>

2.3 Specificaties

2.3.1 Integrated Use-Case Diagram



2.3.2 Use Case: Geef feedback

Description De deelnemer geeft een waardering van een bepaalde activiteit.

Actors Deelnemer

Triggers Deelnemer selecteert een activiteit.

- BCoE
1. Systeem toont de volgende basisinformatie: Titel activiteit, locatie, naam van spreker/organisatie, functie van spreker/organisatie, beschrijving van de activiteit en feedback-elementen
 2. Deelnemer geeft aan welke categorie hij wil voorzien van feedback.
 3. Het systeem vraagt om een waardering van de desbetreffende categorie.
 4. Deelnemer geeft een waardering aan.

5. Systeem selecteert de volgende categorie. Herhaling vanaf stap 3 totdat elke categorie van feedback is voorzien.
6. Deelnemer geeft aan een schriftelijke opmerking over de activiteit te willen maken.
7. Het systeem vraagt om de tekst in te typen.
8. Deelnemer typt tekst in.
9. Deelnemer geeft aan terug te willen keren naar het vorige scherm.
10. Het systeem keert terug op het vorige scherm.

Alternative paths 5. Deelnemer keert terug naar het vorige scherm zonder de feedback volledig ingevuld te hebben.

5.1 Het systeem slaat gegeven feedback alsnog op.

Exception paths /

Precondition De activiteit is al begonnen of geëindigd.

Postcondition De deelnemer heeft zijn waardering van de activiteit per categorie aangegeven.

Related Business Rules /

2.3.3 Niet-functionele eisen

- App nodigt ertoe uit om feedback te geven
 - door live-feedback / live-ranking
 - verloting
- Overzichtelijkheid
 - één centraal lijstje met alle activiteiten
 - per activiteit een eigen scherm met alle informatie
 - altijd makkelijk terugkomen naar het centrale lijstje
- Onderhoudbaarheid
 - backend voor informatie invoer gepland

- alle gegevens zijn online opgeslagen en worden van de app gedownload
- zodra nieuwe gegevens beschikbaar zijn, update de app zichzelf

3 Ontwerp

3.1 Globaal ontwerp

3.1.1 Codeinvoer

De gebruiker krijgt per post of per mail een code toegestuurd die hij bij de eerste opstart van de app in moet tikken. Door middel van deze code kan de juiste programma samenstelling en persoonlijke informatie (bijvoorbeeld gegevens over RU-guest-account, de voor- en achternaam) gedownload worden.

3.1.2 Lijst van activiteiten

Een lijst van het specifieke programma van de deelnemer wordt getoond. Het is vanaf hier mogelijk om de volledige beschrijving voor een enkele activiteit tevoorschijn te halen.

3.1.3 Beschrijving van een activiteit

Het scherm toont informatie over de activiteit en biedt een mogelijkheid om de locatiewijzer naar de locatie van de geselecteerde activiteit te openen.

3.1.4 Gebruikersgegevens

Het scherm toont informatie over de gebruiker (voor- en achternaam, opleiding waarvoor hij zich heeft ingeschreven, RU-Guest-Account-gegevens).

3.1.5 Samenvatting

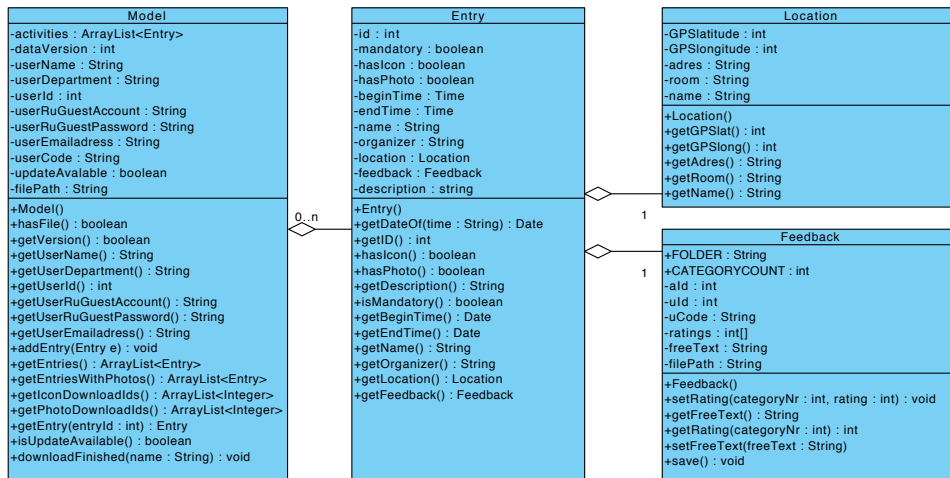
Van elke activiteit wordt één foto ter plekke opgenomen door de host van de activiteit. Hierbij is het mogelijk alle deelnemers, of een groot deel van de deelnemers, op de foto te krijgen. De samenvatting toont een lijst van alle activiteiten waaraan de gebruiker heeft deelgenomen, waarbij elke record van de lijst o.a. de opgenomen foto laat zien. De samenvatting herinnert

de gebruiker aan wat hij heeft gedaan tijdens de Tweedaagse. Doordat hij zichzelf, of andere deelnemers, op de foto herkent kan hij extra enthousiast worden over de organisatie van het evenement. De lijst kan hem stimuleren om naar de beschrijving van een specifieke activiteit te gaan en vervolgens, als dit nog niet gebeurd is, feedback te geven.

3.1.6 Locatiewijzer

De locatiewijzer is het onderdeel dat de gebruiker overzichtelijk laat zien waar hij zich op dit moment bevindt en waar hij naartoe moet lopen.

3.2 Detailontwerp



Het model wordt volgens het MVC-pattern op grond van een XML-bestand gegenereerd. Door de constructor op te roepen van de klasse "model" kan het model altijd opnieuw gegenereerd worden, mits het XML-bestand en bijhorende plaatjes naar behoren gedownload zijn. Het beginscherm ("Main-Activity.java") heeft de opdracht deze belangrijke bestanden te downloaden op de smartphone. Om dit te bereiken wordt op internet een bestand aangevraagd dat alle activiteiten voor een gebruiker samenstelt. Als de code niet bestaat, heeft het bestand geen activiteiten en de dataversie wordt als -1 aangegeven. De klasse "model" kan met behulp van zo een bestand ook een object aanmaken, maar dit object heeft dan juist geen activiteiten en een dataversie van -1. MainActivity downloadt dus het XML-bestand voor de ingevoerde code. Vervolgens vraagt hij naar de dataversie. Als die -1

is, antwoordt MainActivity met “Je code blijkt niet goed te zijn.”. Anders vraagt hij aan het model welke icoontjes en plaatjes hij moet downloaden. Hierbij hebben wij ervoor gekozen om de bestanden gewoon zo te noemen als de activiteiten-id (dus “1.jpg”, “2.jpg”, enz.). Icons zijn in de map `icons` en foto’s in de map `photos` te vinden. Het model kijkt dus welke activiteiten een icon hebben en geeft een lijst van nummers van deze activiteiten terug. Hetzelfde gebeurt bij de foto’s. Nadat de plaatjes allemaal volledig zijn gedownload, start de klasse “MainActivity” de programmalijst. Deze toont dan alle activiteiten met – als er eentje is – het bijhorende icon.

3.3 Ontwerpverantwoording

3.3.1 Google Maps

Voor het implementeren van de locatiewijzer hebben we gekeken naar bestaande methodes en oplossingen om kaarten te gebruiken en routebeschrijvingen te creëren. We zouden dit bijvoorbeeld via de webbrowser kunnen doen of zelf kunnen tekenen. Al gauw kwamen we terecht bij de Google Maps API. Deze voldeed aan de criteria die wij opgesteld hadden. Te weten:

- Snel te implementeren
- Dienst is van hoge kwaliteit, accuraat en kan efficiënt omgaan met resources²
- Veel documentatie beschikbaar³

Daarnaast is zowel Android als de Google Maps Android API door Google ontwikkeld, waardoor ze goed (kunnen) samenwerken.

Toch blijven we sceptisch. Er zijn namelijk ook tegenargumenten aan te voeren voor het gebruik van de Google Maps Android API. We hadden bijvoorbeeld een dienst van de concurrent kunnen implementeren. Enkele nadelen die kleven aan het gebruik van deze API zijn:

- Veiligheid: We hebben geen controle over de code die uitgevoerd wordt binnen de API
- De publisher van de app moet betalen vanaf 25.000 kaartverzoeken⁴

²<http://android-developers.blogspot.com.au/2011/03/android-30-hardware-acceleration.html>

³<https://developers.google.com/maps/documentation/android/>

⁴<https://developers.google.com/maps/faq?hl=nl#usagelimits>

per dag

Aangezien onze app niet daadwerkelijk gepubliceerd gaat worden, zijn bovenstaande negatieve punten wel belangrijk, maar wegen de voordelen op tegen de nadelen. Door het tijdsbestek hebben we tevens geen empirisch onderzoek uit kunnen voeren, waardoor we een puur theoretische werkwijze hebben gehanteerd om een goede bestaande methode/oplossing te selecteren.

3.3.2 Downloaden van gegevens

Er zijn verschillende manieren om bestanden van het internet te downloaden: Android biedt bijvoorbeeld een Async-Task die door generics aan de eigen applicatie kan worden aangepast. Hierdoor kan een download worden gestart die dan door het systeem wordt uitgevoerd. Om te weten hoe ver de download is en wanneer hij klaar is, kunnen de bijhorende functies worden overschreven. Bovendien wordt de voortgang in het notificatie-center van Android getoond. Wij hebben tegen deze Async-Task gekozen, omdat wij in de achtergrond bestanden die voor onze app belangrijk zijn willen downloaden. De gebruiker hoeft niet te weten hoe deze bestanden heten en waar ze worden opgeslagen. Een andere mogelijkheid is een bestand van het internet “handmatig” te downloaden, dus een verbinding op te bouwen en byte voor byte downloaden, in een ByteArray te cachen en vervolgens in een lokaal bestand schrijven. Dit bleek de goede oplossing voor ons te zijn, omdat het niet de bedoeling is dat de gebruiker de gedownloade bestanden (zoals het XML-bestand met de verschillende activiteiten en de bijhorende icoontjes) buiten de app kan zien. Een belangrijk punt bij het implementeren van de tweede mogelijkheid is dat er een nieuwe thread moet worden aangemaakt om onze handmatige download te laten draaien, want Android staat de GUI-thread geen netwerkactiviteit toe. Tijdens onze app-review viel ons op hoe vervelend het is als je niet weet waar het systeem mee bezig is. Wij hebben daarom gekozen om tijdens het downloaden van de plaatjes te laten weten hoever de voortgang is

4 Reflectie

4.1 Samenwerking

De samenwerking tussen de groepsleden is niet altijd even vlekkeloos gegaan. Hoewel we dit al wisten, kwam dit expliciet naar voren tijdens de workshop documentatie en verantwoording. Met name persoonlijke omstandigheden van één van de groepsleden heeft het diegene een aantal keer onmogelijk gemaakt om zich te houden aan de planning. Ook de communicatie is een aantal keer gebrekkig geweest. Gelukkig wisten de overige groepsleden dit op te vangen. We kunnen concluderen dat als er niet goed gecommuniceerd wordt, het werk ook niet goed verdeeld wordt. Dit heeft effect op de harmonie binnen de groep. Concrete actiepunten voor de toekomst zijn:

- Altijd mobiel bereikbaar
- Op tijd laten weten dat je niet bij afspraken kan zijn
- Eerder vertellen wat er speelt

De samenwerking aangaande voorbereiding van presentaties en verslaglegging was bovengemiddeld goed. We wisten op een systematische manier Real Time⁵ samen te werken aan documenten, waarbij we constant elkaars werk konden controleren. Problemen werden gelijk besproken en konden ook direct verbeterd worden. Door alle feedback die we elkaar ieder moment konden geven, waren we in staat problemen snel op te lossen en inhoud zonodig te verbeteren.

Tevens hebben we interviews gehouden met personen die zich bezighouden met voorlichtingsactiviteiten binnen de Radboud Universiteit. Dit heeft ons enorm geholpen om de requirements van ons project in kaart te brengen.

4.2 Gaten

Op het tijdstip van inleveren is ons opgevallen dat we niet alles konden implementeren zoals voorgesteld. We hebben elk onderdeel in rust gelaten zodra we merkten dat het werkte en begonnen aan het volgende onderdeel. Aan het einde hebben we een app die werkt en vele mogelijkheden laat zien. Wij kunnen echter kleine onvolledigheden in onze app zien. Wij noemen ze

⁵<http://www.writelatex.com>

“gaten” omdat de problemen technisch gezien makkelijk op te lossen zijn met de kennis en ervaring die we nu hebben. Wij willen enkele van zulk soort gaten hier nader toelichten.

4.2.1 Locatiewijzer

De “Google Maps API v2” biedt ons de mogelijkheid om een typische Google Maps-kaart in onze app te laten verschijnen. De implementatie (en administratie) van deze API was moeizamer dan gedacht en nam veel tijd in beslag. Tijdens de implementatie bleek dat er geen kant en klare functie is om een route-lijn tussen twee punten te tekenen. Dit zou programmatisch getekend moeten worden met behulp van matrices die wél gemakkelijk te verkrijgen zijn via de Google Maps API. Voor het programmatisch tekenen hadden we echter geen tijd meer. Wat blijft is slechts een kaart met punten van begin- en eindlocatie. De route moet de gebruiker zelf uitzoeken.

4.2.2 Samenvatting

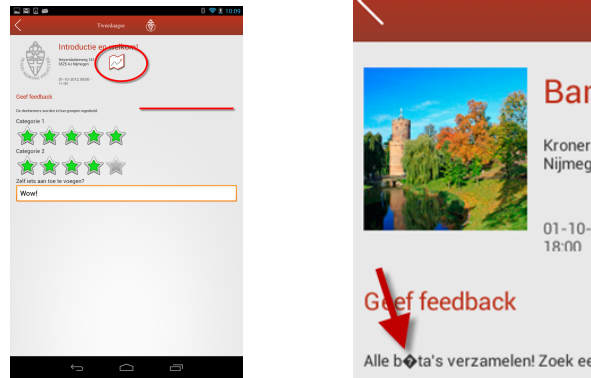
De activity voor de samenvatting ondersteunt slechts dag 1 van de Tweedaagse. Dit komt doordat de betreffende activity de “ListActivity”-klasse uitbreidt die met één object van het type ListView werkt. Voor een tweede lijst binnen een “ListActivity” hadden we voor een andere en ingewikkeldere structuur moeten kiezen.

4.2.3 Ondersteuning schermgroottes

Aan het begin hebben we de GUI ontworpen met behulp van de emulator van het Android SDK. Deze emulator was ingesteld op de schermgrootte van een kleine smartphone. Draait de app op een tablet kun je zien dat de schermgrootte niet optimaal wordt benut (zie figuur 1a).

4.2.4 Accenten

Accenten worden in tekstelementen niet goed weergegeven (zie figuur 1b). We kunnen ons voorstellen dat dit hardware-afhankelijk is, maar wat het probleem hier precies is hebben wij niet meer uitgezocht.



(a) Schermruimte en map (b) Accenten in teksten

Figuur 1: Schermafbeeldingen

4.2.5 Buttons

Soms reageert de digitale back-button boven links niet. Dit is met name het geval bij kleine schermen. Daarnaast twijfelen wij over de duidelijkheid van het map-icoontje om de locatiewijzer te openen (zie rode cirkel in figuur 1a). We kunnen ons voorstellen dat gebruikers niet direct begrijpen dat het icoontje een button is.

4.2.6 Foto's

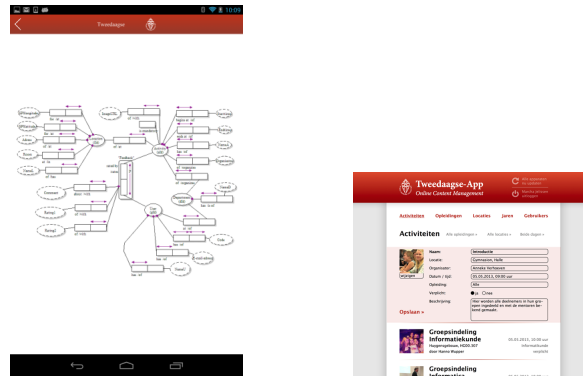
We hebben vaak gezien dat mensen die onze app gebruikten probeerden op de thumbnails van plaatjes te drukken met de verwachting dat de afbeelding wordt vergroot. Deze functie hebben we echter niet geïmplementeerd.

4.3 Kansen

We zien een hoop kansen in onze app. Hadden we meer kennis en ervaring met Android gehad, dan hadden we ze het liefst gelijk geïmplementeerd. Graag willen we op enkele mogelijke functieuitbreidingen ingaan.

4.3.1 Gadgets binnen de app

Elke opleiding heeft zo zijn bijzonderheden. Volg je de studie informatica (of informatiekunde), dan zijn tentamens vaak oplossingsgericht. Er is een



(a) Doemeinmodel-editor (b) Webinterface voor de administratie als gadget

Figuur 2: Kansen

probleem en je moet een werkende oplossing bedenken. De producten van de oplossing zijn vaak geen losse tekst, maar vaak in de vorm van modellen (bijv. eindige automaten), genormde tekeningen (bijv. ORM, UML) of tekst met een bepaalde syntax (bijv. wiskundige redeneringen of instructies in een bepaalde programmeertaal). Deze speciale vorm van producten zouden aan bod kunnen komen tijdens een hoorcollege op de RU Tweedaagse. De deelnemers zouden met de materie gelijk kunnen oefenen op onze app! Hun producten zouden ze kunnen delen via sociale media. De editoren binnen onze app die dat mogelijk maken noemen wij gadgets. Ze zijn te zien als kleine subprogramma's binnen onze app. Een voorbeeld van een gadget voor het tekenen van een domeinmodel volgens ORM is te zien in figuur 2a.

4.3.2 iOS-versie

De doelgroep van de app heeft voor een groot deel een iPhone en geen Android-toestel. Onze app ondersteunt slechts Android-toestellen. Hierdoor wordt een groot gedeelte van onze doelgroep uitgesloten. Bij een mogelijke publicatie van onze app is het dus noodzakelijk ook voor een iOS-variant te zorgen.

4.3.3 Webinterfaces

De administratie van de Tweedaagse bestaat waarschijnlijk uit kantoorklerken die niet bekend zijn met het directe wijzigen van SQL-tabellen zonder een bepaald front-end (wij bedoelen een GUI die als tussencommunicatie dient tussen gebruiker en de database). Om het wijzigen en toevoegen van gegevens zo laagdrempelig mogelijk te houden zou een webinterface, zoals in figuur 2b is te zien, gemaakt kunnen worden. Dit zorgt ervoor dat gegevens actueel blijven (en alleen dan kan onze app slagen). Een webinterface voor de deelnemer zou het mogelijk maken om gemakkelijk aan te geven dat hij toch geen gebruik zal maken van de app en alle relevante informatie liever in papiervorm wil ontvangen. Daarnaast zou het daarmee mogelijk zijn persoonlijke gegevens of de gekozen opleiding te wijzigen.

4.3.4 Communicatie universiteit

De Radboud Universiteit kent een aantal bijzonderheden. Sommige gebouwen zouden ook digitaal extra benadrukt kunnen worden omdat ze er bijzonder mooi uitzien, omdat ze een van Nederlands grootste magneetvelden kunnen creëren of omdat de ramen elke drie maanden met kranen gezeemd moeten worden. Niet alleen gebouwen, maar ook docenten, het ziekenhuis of bepaald onderzoek zijn zeker een vermelding waard in onze app om deelnemers extra te enthousiasmeren voor de Radboud Universiteit.