

Research & Development

Structuur & Fases

Dit document dient als een poging om structuur aan te brengen in de ontwikkeling van Hot Meals app. Ik stel voor de ontwikkeling in fases te doen van één week ieder, waarin incrementeel de app groeit in functionaliteit en afwerking.

Fase 1

1. De app bevat de volgende functionaliteit:
 1. Vier *views* volgens de huidige storyboard:
 1. Home
 2. Search
 3. List
 4. Recipe
 2. Vanuit elke *view* kan de gebruiker naar elke *view* gaan (volgens storyboard) gebruik makende van knoppen in de interface (gestures kunnen later worden toegevoegd).
 3. Een eenvoudige search functie op basis van:
 1. Bereidingstijd
 2. Budget
 3. Ingrediënten
 4. Een eenvoudige Recepten-*view*
2. XML data accessing. Dat wil zeggen:
 1. XML data bevat recepten volgens standaardlayout:
 1. Bereidingstijd
 2. Budget
 3. Ingrediënten
 2. Maar ook de user-data, waarvan in ieder geval nu nodig is:
 1. Favorite (boolean)

En misschien later:

2. Rating
3. Extra benodigdheden

In ieder geval goed om in het achterhoofd te houden dat de XML layout niet vast ligt, maar kan veranderen naarmate we meer functionaliteit toevoegen. Softcoden dus!

3. Er moet van deze lijst gelezen én geschreven kunnen worden.

Taakverdeling

Om elkaar niet in de weg te zitten en (deels) onafhankelijk van elkaar te kunnen werken stel ik voor enige taakverdeling in te brengen. Sander gaf mij ook de indruk hier achter te staan. Een voorlopige taakverdeling (kan veranderd worden, holla at me):

1. Sander zet de *back-end* van een Android app op. Dat wil zeggen:
 1. Vier *views* die onafhankelijk van elkaar bestaan en kunnen wisselen in elkaar. Het lijkt me dat dit een zeer standaard functionaliteit is van Android, en hier zou weinig dubbelzinnigheid in moeten zitten, hoop ik (zoals er wel is bij het creëren van een *game-view*).
 2. Denk ook na over dingen als I/O, verschillende klassen die verschillende dingen doen (zoekalgoritme komt niet in de klasse voor search waarschijnlijk) etc. Wat wordt de structuur van onze code?
2. Timo maakt een voorlopig zoekalgoritme. De eerste versie van een zoekalgoritme, zo lijkt me, hoeft nog niet in de app worden ontwikkeld. Het lijkt me relatief eenvoudig om een standaard Java project op te zetten en een textuele interface om door een voorbeeldpopulatie van recepten te zoeken en resultaten te leveren.
3. Jaco onderzoekt XML:
 1. *Read*: Hoe wordt vanuit Java de toegang tot een XML database geregeld?
 2. *Write*: Hoe werkt schrijven naar XML? Doen we dit on te fly of bij het afsluiten van de app? Wat als de app crasht tijdens schrijven? Is dan de database corrupt?
 3. Specifiek in Java, welke klassen/functions/libraries/etc zijn nodig om met XML te werken? Hoe ziet standaardcode er uit om naar een standaard XML file te lezen/schrijven?
 4. Lezen we bij het opstarten van de app de hele database in of kunnen we on te fly XML lezen?
 1. Denk hierbij aan de search functie: moet deze door XML zoeken of door een ArrayList (bijvoorbeeld) van instanties van een Recept klasse?
 5. Hoe zorgen we dat het uitbreidbaar blijft?
 6. Hoe gaan we de XML onderhouden?
4. Ik maak functionaliteit en interaction, ofwel *front-end*:
 1. Knoppen, zoekvelden, weergave van recepten etc.
 2. User input: on-screen keyboards, favoriting, *radio buttons*, *check boxes*, *sliders*, etc.
 3. Hoe werkt dit normaliter in Android? EventListeners? Hoe delegeren klassen/knoppen/*views* taken en data naar elkaar?
 4. Welke Android libraries dienen hiervoor?
 5. Wat zijn standaard design conventies op Android? Hoe implementeren we deze? Denk bijvoorbeeld aan: een *scrollbar* die moet verschijnen wanneer een Recept niet in zijn geheel getoond kan worden.

Verder heb ik de storyboard, flow van het programma, en taakverdeling bedacht (en schrijf ik dit document, meta yo).

Samenwerking

Er moet natuurlijk veel overleg plaatsvinden, in het specifiek bij de volgende onderdelen:

- Sander en Jaco: XML en het inlezen en schrijven hiervan.
- Daniel en Sander: Wat behoort tot het front end van de app en wat tot de back end? Hoe interacteren deze met elkaar?
- Timo en Sander: Hoe implementeren jullie de zoekfunctie?
- Allemaal: data-representatie. Dit is een van de belangrijkste punten: hoe wordt alles gerepresenteerd? De strings die uit een input komen, hoe moeten die er uit zien zodat er mee gezocht kan worden? De recepten, hoe worden deze gepresenteerd zodat ze gedisplayed kunnen worden?
- Timo en Jaco: wat wordt een handig format voor XML, met betrekking tot zoeken?

Voortgang en workload verdeling

Omdat deze verdeling niet perfect is kan het zijn dat één onderdeel meer werk is dan een andere. Houd even bij hoeveel werk je er in steekt en zorg dat het eerlijk blijft!

- Sander: 0 uur.
- Jaco: 0 uur.
- Timo: 0 uur.
- Daniel: 2 uur.

Feedback

Just to be clear, I'm not a professional 'taakverdeling maker'. This being said, I am open to any and all criticism.

Bron: [Know Your Meme: In This Moment I Am Euphoric.](#)

Vragen, klachten of commentaar? Email: daniel@roeven.com, subject "Haatmail".

Zonder grappen, holla at a boy.

Terminologie:

- **View:** alles wat in één scherm te zien is. Bij iOS komt dit ook precies overeen met één object. Met behulp van standaardlibraries is er te wisselen tussen views.
- **Radio button:** radio buttons of keuzerondjes zijn keuzemogelijkheden in een GUI, waarbij maar één keuze tegelijkertijd geselecteerd kan zijn. Keuzerondjes die bij elkaar horen, noemt men een groepsvak. Bron: [Wikipedia: Keuzerondje](#).
- **Check box:** Een check box of selectievakje, ook wel aankruisvakje, wordt in veel GUI's gebruikt om de gebruiker iets aan te laten kruisen of aan te vinken. Dit aanvinken kan bij alle selectievakjes in een groep, dit in tegenstelling tot een groepsvak keuzerondjes. Bron: [Wikipedia: Selectievakje](#).
- **Slider:** A slider, also known as track bar in Microsoft literature, is an object in a GUI with which a user may set a value by moving an indicator, usually in a horizontal fashion. In some cases the user may also click on a point on the slider to change the setting. It is different from a scrollbar in that it is typically used to adjust a value without changing the format of the display or the other information on the screen. Bron: [Wikipedia: Slider](#).
- **Scrollbar:** Een scrollbar of schuifbalk is een verticale balk die zich meestal bevindt aan de rechterkant van een GUI. Een schuifbalk verschijnt als er in een venster meer gegevens moeten staan dan de ruimte in het venster toelaat. Bron: [Wikipedia: Schuifbalk](#).
- **Front and back ends:** In computer science, the front end is responsible for collecting input in various forms from the user and processing it to conform to a specification the back end can use. Bron: [Wikipedia: Back end](#).