

PROJECT PALÈSTRA

Een terugblik op onze maaltijd

WeetWatJeEet eindverslag

Thijs klein Baltink, Jorrick Boshove en Mick Koomen

26 juni, 2015

Voorwoord

Het verslag wat wij hier presenteren dient als documentatie over het ontwikkelingsproces van onze applicatie WeetWatJeEet. Verder blikken wij terug op het verloop van dit ontwikkelingsproces en laten wij de app zien zoals hij uiteindelijk geworden is. Het verslag bestaat uit dit voorwoord, een inleiding, de productverantwoording, specificaties, ons globaal en gedetailleerd ontwerp, de ontwerpverantwoording en als laatste de reflectie. De 'inleiding' beschrijft onze app en de belangrijkste eigenschappen. In de 'productverantwoording' geven we aan wat onze app zo uniek maakt onder andere vergelijkbare applicaties die er al zijn. In de specificaties beschrijven we alle eigenschappen van de applicatie. Verder geven wij in het 'globaal en gedetailleerd ontwerp' een kijk in ons ontwerp toe en leggen we verantwoording af voor ons ontwerp in de 'ontwerpverantwoording'. Als laatste reflecteren we op onze applicatie en het bijbehorend ontwikkelingsproces.

Inleiding

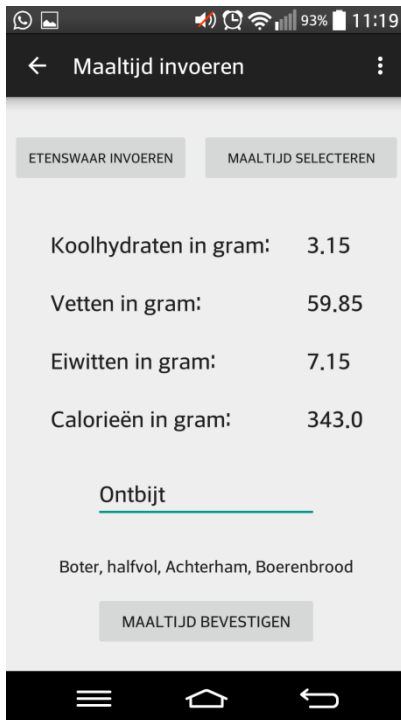
De applicatie WeetWatJeEet is een hulpmiddel bij de administratie van je voeding. Met deze applicatie kun je maaltijden maken die bestaan uit een gegeven lijst van etenswaren. Je kunt vervolgens in een grafiek zien hoeveel calorieën, koolhydraten, eiwitten en vetten je per dag binnen hebt gekregen over een tijdsperiode van dertig dagen. Ook is er een lijst met alle maaltijden die je tot dan toe gegeten hebt.

Nu eerst wat over de verscheidene activiteiten en hun functies.

Als je voor de eerste keer de applicatie opstart heb je nog geen maaltijden kunnen toevoegen en wordt er ook nog niks weergegeven in de grafiek, of lijst van maaltijden.



Dit scherm wordt weergegeven als je de applicatie opstart. Vervolgens zijn er 3 opties. Het is handig om allereerst een maaltijd in te voeren door op *Maaltijd invoeren* te drukken.



Dit scherm wordt getoond nadat je op *Maaltijd invoeren* hebt gedrukt. Hiernaast zie je het overzicht van de maaltijd die je aan het samenstellen bent. De voedingsinformatie wordt als een lijstje weergegeven en daaronder zie je de naam van de maaltijd. Onder de naam van de maaltijd (“Ontbijt”) zie je een reeks etenswaren die in deze maaltijd zitten.

Stel, je wilt nog een etenswaar toevoegen, dan druk je linksboven op *etenswaar invoeren*.



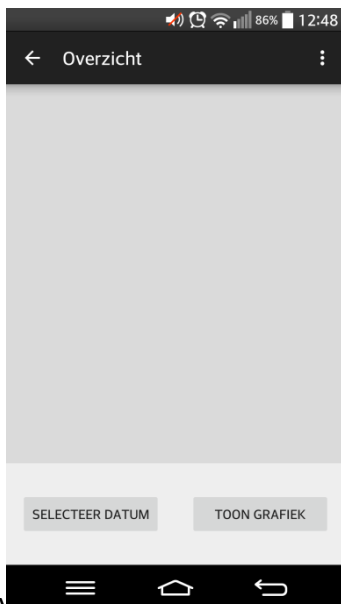
Hiernaast zie je de lijst met etenswaren waar je uit kunt kiezen. Je kunt elk item uit de lijst selecteren, waarna een Toast bericht weergegeven wordt met daarin de voedingsinformatie per 100 gram van het geselecteerde etenswaar. Vervolgens kun je bij *Vul hoeveelheid in* een gewenste hoeveelheid gram van het geselecteerde item invoeren.

Behalve het invoeren van etenswaren kun je ook een al bestaande maaltijd opnieuw selecteren en toevoegen.



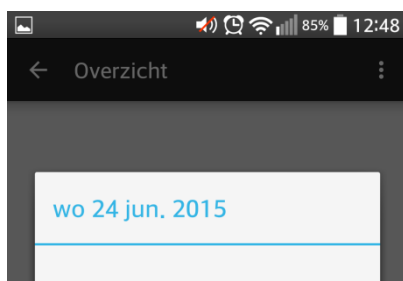
De lijst maaltijden die wordt getoond is geordend op dag van invoeren, en twee maaltijden, die ingevoerd zijn op dezelfde dag, zijn geordend op alfabet. Per item in de lijst wordt linksboven de datum van invoeren weergegeven en onder de titel zie je de etenswaren waaruit deze maaltijd bestaat. Zoals je ook kunt zien wordt er bij het selecteren van een maaltijd in een Toast bericht voedingsinformatie getoond. Na het selecteren van een maaltijd kun je deze bevestigen door op de knop *Bevestigen* te drukken.

Na het invoeren van maaltijden kun je via *Overzicht* een overzicht krijgen van de ingevoerde maaltijden.

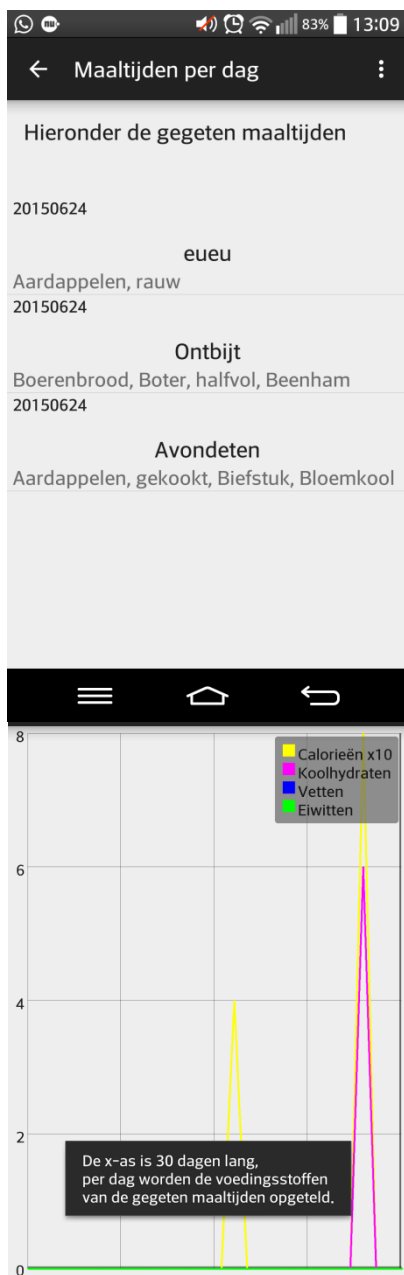


Na het selecteren van *Overzicht* zijn er twee opties. Door op *selecteer datum* te drukken kun je een datum selecteren, waarna er vervolgens een lijst met maaltijden wordt weergegeven met de gegeten maaltijden op die datum. Bovendien wordt na het drukken op *toon grafiek* een grafiek getoond met daarin de hoeveelheid voedingsstoffen op de y-as en de tijd op de x-as.

Als je op *selecteer datum* drukt, kun je een dag selecteren, daarna worden de maaltijden getoond die zijn gegeten op die specifieke dag.



Je selecteert een datum en drukt vervolgens op *gereed*.



Net als in de lijst met maaltijden, bij het invoeren van een maaltijd, zijn de maaltijden geordend op dag van invoeren en per dag geordend op alfabet. Ook hier wordt, bij het selecteren van een maaltijd, een Toast bericht weergegeven met voedingsinformatie over de desbetreffende maaltijd.

De grafiek heeft als functie een overzicht weer te geven van je eetgeschiedenis. In één oogopslag is het gemakkelijk om bepaalde patronen te zien in je voeding. Omdat calorieën in maaltijden meestal in grote hoeveelheden voorkomen hebben we deze getallen gedeeld door tien, zodat de grafiek van de calorieën gemakkelijker past bij de andere drie grafieken.

Verder kun je, naast voedingsstoffen in een grafiek, ook een lijst met alle ingevoerde maaltijden tonen.



Net als in de lijst met maaltijden, bij het invoeren van een maaltijd, zijn de maaltijden geordend op dag van invoeren en per dag geordend op alfabet. Ook hier verschijnt er, bij het selecteren van een maaltijd, een Toast bericht met voedingsinformatie over de desbetreffende maaltijd.

Productverantwoording

Het idee voor deze applicatie begon bij een klacht van teamgenoot Jorrick. Hij had zelf altijd moeite om gemakkelijk bij te houden wat hij gegeten had, maar wilde dit wel graag bijhouden in verband met sportoverwegingen. Een applicatie waarmee je bij zou kunnen houden welke voedingsstoffen je allemaal binnen had gekregen zou perfect zijn om zijn probleem op te lossen.

Onze applicatie heeft zo een functionaliteit waar veel mensen baat bij zouden kunnen hebben. In deze wereld is men steeds drukker en zijn er tal van lekkernijen beschikbaar. Als men toch een gezond dieet gemakkelijk wilt behouden, is ons product een mogelijk hulpmiddel. Om namelijk zelf de voedingswaardes op te zoeken van alles dat je eet, deze vervolgens op te tellen, en zo een kalender bij te gaan houden vinden de meeste mensen veel teveel werk.

Het voordeel van onze applicatie ten opzichte van andere vergelijkbare applicaties is dat onze erg licht, en dus, efficiënt is. Onze applicatie richt zich puur op het bijhouden van wat je binnenkrijgt, terwijl andere applicaties op de markt dit meestal hebben als een onderdeel van een grotere applicatie welke bedoeld is voor fitness. Dit zorgt ervoor dat wij voor het dieetaspect het snelst zijn. Een ander voordeel van onze applicatie is dat je er geen extra hardware voor hoeft aan te schaffen, veel van de fitness applicaties met dergelijke functionaliteit hebben een eigen 'wearable' product dat aangeschaft hoort te worden in combinatie met hun applicatie, dit zijn dan apparaten zoals stappentellers, of hartslag monitoren. Onze applicatie heeft geen dergelijke verplichtingen.

Specificatie

De applicatie WeetWatJeEet heeft als functie het bijhouden van de voedingswaarden die je inneemt. De applicatie heeft een database met tabel etenswaren, met bijbehorende voedingswaarden per 100 gram. Het heeft de optie om hieruit etenswaren te selecteren, en een hoeveelheid gegeten gram van zo'n etenswaar in te voeren. Een combinatie van etenswaren en inname hoeveelheden kan omgezet worden naar een maaltijd. Een maaltijd is een combinatie van etenswaren en ingenomen hoeveelheden, plus de datum waarop deze is ingevoerd. De maaltijden worden in een tabel voor maaltijden ingevoerd met een door de gebruiker ingevulde naam en een tekst waarin kort staat wat voor etenswaren er in de maaltijd zitten.

De applicatie biedt verder de mogelijkheid om via overzicht -> grafiek de ingenomen voedingswaarden over de afgelopen 30 dagen te bekijken. Dit gebeurt door per betreffende dag de voedingswaarden van de maaltijden bij elkaar op te tellen. De applicatie heeft tevens bij overzicht de mogelijkheid om voor een geselecteerde dag de maaltijden gegeten weer te geven. Extra functionaliteit stelt je bij maaltijd invoeren in staat om een eerder gegeten maaltijd direct nogmaals in te voeren, en bij etenswaren selecteren kun je zoeken door een string in te voeren waarmee alleen etenswaren met de gegeven substring worden weergegeven.

Ontwerp

Globaal

Onze app bestaat uit de volgende componenten. De database met etenswaren, de invoer van maaltijden, de grafische representatie van de gegeten maaltijden. De database met etenswaren slaat alle etenswaren op met hun bijbehorende voedingsstoffen. De invoer van maaltijden maakt het mogelijk om maaltijden aan te geven welke maaltijden je hebt gegeten. De grafische representatie van de gegeten maaltijden maakt het mogelijk om in een grafiek te zien hoeveel voedingsstoffen je hebt binnengekregen de afgelopen 30 dagen of in een kalender te zien op welke dag je welke voedingsstoffen hebt binnengekregen.

De hierboven beschreven componenten opereren niet in hun eentje. Zo maakt 'de invoer van maaltijden' gebruik van de component 'database' om daar zijn maaltijden in op te slaan. Tevens haalt de 'grafische representatie' info over de maaltijden op uit de database. Dit simpele design maakt het mogelijk om snel en efficiënt informatie binnen de applicatie uit te wisselen.

Detail

Bij ons ontwerp begonnen we met de database. Voor de database hebben we twee klassen ontworpen, namelijk DatabaseContract en DatabaseManager. Hier specificceert DatabaseContract de tabel namen en namen van de kolommen. Wij hebben voor de namen van de kolommen en tabellen 'public static final' Strings aangemaakt. Public zodat andere klassen, met name DatabaseManager, deze Strings kunnen gebruiken en final zodat ze niet veranderd kunnen worden. De DatabaseContract heeft verder geen methodes.

In DatabaseManager wordt de database gecreëerd en zijn er functies om de database te veranderen door bijvoorbeeld etenswaren toe te voegen. Alle andere klassen in onze applicatie maken gebruik van de DatabaseManager klasse om een instantiatie van de database te krijgen en hier informatie aan toe te voegen dan wel informatie uit de database op te vragen. In deze klasse hebben we vier final strings die zich vertalen naar SQL statements om de tabellen etenswaren en maaltijden te maken in de database en om eventueel deze te deleten. Deze klasse heeft verder methodes om een etenswaar of een maaltijd aan de database toe te voegen en om informatie uit de database te halen. De uit de database te halen informatie kan bijvoorbeeld een maaltijd zijn met een bepaalde datum, of een bepaald etenswaar.

De MainActivity klasse is de 'toegangspoort' tot onze app. Deze klasse toont het menu dat de gebruiker ziet als hij de app opstart. Vanuit deze klasse kunnen de activiteiten maaltijdInvoeren, OverzichtKalender en mijnMaaltijden worden gestart. MainActivity heeft één attribuut, namelijk een databasemanager object. Verder heeft deze klasse een aantal methodes. Uiteraard hebben we drie methodes welke een Intent gebruiken om een nieuwe activity te starten. Deze methodes corresponderen natuurlijk met de opties die de gebruiker in het menu heeft. Verder hebben we een methode setCalendar in deze klasse die een kalender object aanmaakt die we gebruiken in de OverzichtKalender klasse.

Zoals hierboven aangegeven kun je vanuit de mainActivity drie andere activity's starten. MaaltijdInvoeren is een van deze activiteiten. In maaltijdInvoeren kun je een maaltijd invoeren. Dit kun je doen door een etenswaar te selecteren waarvan de voedingswaardes worden opgeslagen. Je kunt zoveel etenswaren opslaan als je wenst, vervolgens kun je je maaltijd een naam geven en wordt deze opgeslagen. Tevens kun je in maaltijdInvoeren een maaltijd die je in het verleden al gebruikt hebt opnieuw selecteren. Het selecteren van een etenswaar wordt geregeld door de klasse EtenswaarInvoer, het opnieuw selecteren van een maaltijd wordt geregeld door de mijnMaaltijden klasse. In de maaltijdInvoeren klasse zelf hebben we een aantal attributen. De belangrijkste zijn de TextViews en een DatabaseManager object. De TextViews worden gebruikt om de voedselwaardes van de etenswaren bij te houden en aan te passen wanneer er een etenswaar wordt toegevoegd. Dit aanpassen van de voedingswaardes wordt gedaan door de methode setValues. Het DatabaseManager object gebruiken we om de aangegeven maaltijden op te slaan in de database. Dit opslaan van de

maaltijd gebeurt in de methode maaltijdBevestigen. Verder hebben we nog twee methodes om de activiteiten EtenswaarInvoer en maaltijdenLijst te starten. Bij EtenswaarInvoer geven we nog data mee, te weten de huidige voedselwaardes van de maaltijd op dat moment.

Een andere activity die je kunt starten vanuit de mainActivity is OverzichtKalender. In OverzichtKalender kun je een overzicht krijgen van jouw gegeten maaltijden. Dit overzicht kan getoond worden door een datum te selecteren waarbij vervolgens van die datum de gegeten maaltijden en bijbehorende voedselwaardes worden getoond. Tevens kun je in een grafiek een overzicht krijgen van de voedselwaardes van de afgelopen 30 dagen. De klasse zelf heeft geen attributen. Wel is er een functie showDatePickerDialog die het mogelijk maakt om een datum te selecteren via een Fragment. Dit fragment wordt beheerd in de Klasse DatePickerFragment. Hier wordt het daadwerkelijk kunnen selecteren van een datum gerealiseerd. Verder hebben we nog een functie toonGrafiek in de OverzichtKalender klasse die de Klasse GraphHelper start.

De laatste activity die we kunnen starten vanuit de MainActivity is mijnMaaltijden. In mijnMaaltijden wordt in een lijst alle maaltijden weergegeven die je hebt gegeten. Deze klasse heeft een aantal attributen. De belangrijkste zijn ook hier een DatabaseManager object en een ListView voor de lijst van maaltijden. De listView wordt gevuld door alle maaltijden uit de database te halen met een query. Dit doen we in de functie populateListView. Verder heeft deze klasse geen functies.

Ontwerpverantwoording

Wij hebben er in ons ontwerp voor gekozen om onze applicatie simpel te houden in gebruik en voor de gebruiker altijd toegankelijk. Dit is het dan ook het voornaamste voordeel van ons ontwerp. Twee belangrijke ontwerpbeslissingen die we hebben genomen zijn het gebruik van een lokale SQLite database, en het bijhouden van alleen calorieën, koolhydraten, vetten en eiwitten.

Het lokaal opslaan van de database heeft grote voordelen. Zo hoef je met een lokaal opgeslagen database geen connectie met internet te hebben om toegang te hebben tot de database. Dit maakt het gemakkelijker voor de gebruiker om ten alle tijden de applicatie te kunnen gebruiken. Verder heeft het als voordeel dat wij geen server hoeven te hosten, wat liggend aan aantal gebruikers duur kan worden, en dat we geen gebruikers-registratie- en inlogsysteem bij hoeven te houden. Dit maakt het makkelijker om security-risico's te ontwijken en zorgt ervoor dat een gebruiker minder stappen hoeft te ondernemen voor hij de applicatie kan gebruiken.

Daarnaast hebben wij in ons ontwerp gekozen om calorieën, koolhydraten, vetten en eiwitten te implementeren. Dit omdat men deze in het algemeen beschouwd als de belangrijkste voedingsstoffen om bij te houden en op te kunnen letten. Tevens is er goede informatie beschikbaar over calorieën, koolhydraten, vetten en eiwitten dat zich per 100 gram in verscheidene voedingsmiddelen bevindt. Hierdoor is het voor ons makkelijk gebleken om betrouwbare informatie over de voedingswaarden te verkrijgen. Waarden van vitaminen zijn erg onbetrouwbaar en veranderlijk op basis van bereidingsmethoden, daarom hebben wij ervoor gekozen deze in ons ontwerp niet te includeren.¹ Er zijn extreem veel verschillende soorten mineralen die in voedingsmiddelen voor kunnen komen. Het toevoegen van deze mineralen zou het geheel erg onoverzichtelijk maken. Daarom hebben wij ervoor gekozen om mineralen buiten ons ontwerp te laten.

1

¹ <https://www.puurgezond.nl>

Reflectie

Onze applicatie heeft de basis-functionaliteit die we wilde hebben, en hier zijn we dan ook zeker tevreden mee. Het maken van de applicatie heeft ons veel geleerd over de communicatie tussen Java en xml binnen de Android IDE, we hebben geleerd om de vertaalstap te maken van een java programma dat op console uitgevoerd kan worden, naar een programma dat gebruik maakt van de Android libraries, en dat uitvoerbaar is op je mobiele telefoon.

In het begin hadden we erg veel moeite met het schrijven van code, dit kwam voornamelijk doordat wij niet bekend waren met android studio, en dat wij niet wisten wat we nou precies met die .xml-bestanden

Als we terugkijken naar het proces zijn er een aantal verbeter-punten. We hebben, voornamelijk in de eerste en tweede fases slecht met elkaar gecommuniceerd, en overleg en planningen slecht bijgehouden. Hierdoor was het zo dat we vaak erg kort van tevoren afspraken om aan het project te gaan werken, waardoor we voor onszelf minder bruikbare tijd overlieten. Dit heeft ons in het begin van de tweede fase een redelijke achterstand opgeleverd, en deze achterstand heeft er aan het einde van de tweede fase weer voor wat problemen gezorgd.

Wanneer we kijken naar het eindproduct dan zijn we niet volledig tevreden, onze afwerking is niet mooi, en we hadden enkele functies het liefst toch op een net andere manier aangepakt. De basis functionaliteit van onze applicatie is er, maar als wij in het begin van de project-periode meteen goed waren begonnen, en betere afspraken hadden gemaakt, dan hadden we voor ons gevoel een mooier eindproduct kunnen krijgen. Voornamelijk esthetisch gezien zal dit veel verschil gemaakt hebben, maar ook bij onze implementatie van de tabel etenswaren hadden wij dan een betere kans gehad om een automatisch importeer-systeem te hebben kunnen implementeren.

Als we een dergelijk project in de toekomst zullen doen dan zal het er in de uitvoering voor een groot deel hetzelfde uitzien, maar zullen we vooral strikter zijn in het maken en opvolgen van een planning. En tevens zullen wij een dergelijke planning dan vanaf dag 1 maken.