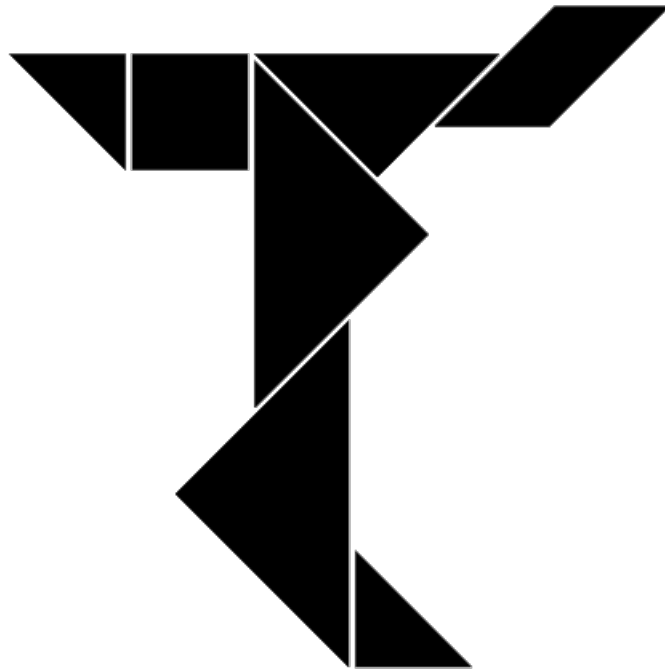


R&D1  
De inzet waaruit succes volgde

**Appsinth**  
Tim Janssen  
Jip Dekker  
Mats Ouborg

30 juni 2012



# Inhoudsopgave

<b>1</b>	<b>Beschrijving</b>	<b>4</b>
1.1	Inleiding . . . . .	4
1.2	Productverantwoording . . . . .	4
1.3	Specificaties . . . . .	5
1.3.1	Functionele eisen . . . . .	5
1.3.2	Niet-functionele eisen . . . . .	5
<b>2</b>	<b>Ontwerp</b>	<b>8</b>
2.1	Globaal ontwerp . . . . .	8
2.1.1	Systeemarchitectuur . . . . .	8
2.1.2	Thalia-app . . . . .	9
2.1.3	Online database . . . . .	9
2.1.4	Pizza-API . . . . .	9
2.1.5	Thaliawebsite & database . . . . .	9
2.1.6	Communicatie . . . . .	9
2.2	Detailontwerp . . . . .	10
2.3	Ontwerpverantwoording . . . . .	11
<b>3</b>	<b>Evaluatie</b>	<b>12</b>
3.1	Testdoelen . . . . .	12
3.2	Gebruikers(sub)groepen . . . . .	13
3.3	Scenario's . . . . .	14
3.4	Methoden voor gegevensverzameling . . . . .	15
3.5	Procedure . . . . .	16
3.6	Resultaten . . . . .	17
3.6.1	Werving proefpersonen . . . . .	17
3.6.2	Resultaten interviews . . . . .	17
3.7	Conclusies . . . . .	18
<b>4</b>	<b>Reflectie</b>	<b>19</b>
4.1	Groep . . . . .	19
4.2	Reflectie Mats . . . . .	20
4.3	Reflectie Tim . . . . .	21
4.4	Reflectie Jip . . . . .	22
	...	

Bijlage: verwijzing naar code & ruwe interview gegevens

# Voorwoord

Dit document is het eindverslag van ons app-bouwproject in het kader van de cursus Research en Development 1. Voor deze cursus hebben wij de Thalia-app gemaakt. Dit document zal onze resultaten en bevindingen tonen die we het afgelopen kwartaal hebben opgedaan.

Allereerst zal er een beschrijving worden gegeven van onze app, waarin wij uitleggen wat de eigenschappen van de app zijn en waarom wij het beter vinden dan wat er nu is. Daarna zal een wat technischere beschrijving gegeven worden over de interactie van de mens met de machine.

Ten tweede zal er uitgelegd worden hoe de architectuur van de app eruit ziet. Welke onderdelen daar bij komen kijken en voornamelijk hoe dit technisch uiteindelijk gerealiseerd is.

Ten derde zal er een usability onderzoek worden getoond waarin proefpersonen hun mening op onze app geven, waarnaar wij concluderen wat er eventueel nog aan gedaan dient te worden.

Als laatste in dit document is er een stuk reflectie van de groep en daarna een individuele evaluatie per persoon. Hierin geven wij als onderzoekers maar ook als ontwikkelaars aan wat wij in dit traject hebben geleerd.

# 1 Beschrijving

## 1.1 Inleiding

Wat kan onze app nou eigenlijk?

Onze app is een pizzabestel-app voor de studievereniging Thalia aan de Radboud Universiteit te Nijmegen. De app is bedoeld voor het gebruik bij de Thalia-borrels.

Als er op een Thalia-borrel pizza's besteld moet worden kan de gebruiker op de app, na het inloggen met zijn Thalia-ID, kiezen uit een lijst met pizza's. Vervolgens wordt er naar specifieke wensen voor de pizza gevraagd. Bij keuze geeft de app aan waar er betaald kan worden.

## 1.2 Productverantwoording

De Thalia-app is een vervanging van de bestaande papieren pizzalijst. De klacht van die pizzalijst was dat het erg lang duurde, soms wel een uur, voordat de lijst helemaal rond was gegaan. In die zin is onze app al een hele verbetering want het is op de app mogelijk binnen 1 minuut je bestelling in te voeren.

Ten tweede is onze app ook nog goed te gebruiken als je veel alcohol hebt genuttigd. Iets wat vaak voorkomt op een borrel. Bij de pizzalijst had je iemand die de lijst rondbracht en die tevens controleerde of het allemaal wel goed was ingevuld. Bij een app op een mobiele telefoon heb je deze controle niet meer en daarom is het van uiterst belang dat de app makkelijk en overzichtelijk in gebruik is. We hebben dit bereikt door veel te werken met plaatjes in plaats van tekst en duidelijke grote iconen te gebruiken. Daardoor is de app ook meteen makkelijk te begrijpen en te leren.

Ten derde motiveert onze app het steeds weer opnieuw te gebruiken. Het klikken op een plaatje van je pizza met voorkeur is vele malen gemakkelijker dan het in tekstvorm op te schrijven. Ten opzichte van de huidige pizzalijst is dit dus een goede verbetering.

Ten vierde is onze app simpel. Dit realiseren we door niet meer schermen en menu's te gebruiken dan nodig is. Je kan bij gebruik in één flow door de app heen manoeuvreren, zonder terug te hoeven gaan of twee keer door het zelfde menu te gaan.

Ten vijfde geeft onze app feedback over wat hij aan het doen is. Op die manier weet de gebruiker meteen of de app doet wat hij of zij wil dat de app moet doen.

Op het moment is er niet een vergelijkbare app op de markt. Er bestaan wel pizzabestel-apps van grote bedrijven, maar die zien wij niet als concurrentie omdat onze app voor specifieke één doel is gemaakt, namelijk het vervangen van de pizzalijst bij de Thalia-borrels. Bij onze Thalia-app dient de gebruiker zelfs in dronken status nog een pizza te kunnen bestellen, iets wat bij de app van bijvoorbeeld Domino's niet persé hoeft. Ook is het assortiment bij de Thalia-borrels zeer beperkt en dat van Domino's zeer ruim. We zien dus dat dit in geen enkel geval concurreert met onze eigen app.

We weten van de borrelcommissie van Thalia dat er vraag is naar een vervanging van de pizzalijst. Zij hebben zelfs gezegd dat ze de pizzalijst willen gaan vervangen met onze app. We hebben dus bijna contractair vaststaan dat onze app gebruikt gaat worden.

## 1.3 Specificaties

In dit deel beschrijven we de eigenschappen van onze app wat meer in detail. Dit noemen we de specificaties en we beschrijven het doormiddel van Requirements. Requirements (eisen) bestaan uit “functionele” en “niet-functionele” eisen. Het verschil tussen deze twee wordt beschreven als in dat functionele eisen direct zichtbaar zijn en invloed hebben op het systeem, terwijl een niet-functionele eis niet direct zichtbaar is voor de gebruiker, maar wel degelijk onderdeel is van het systeem.

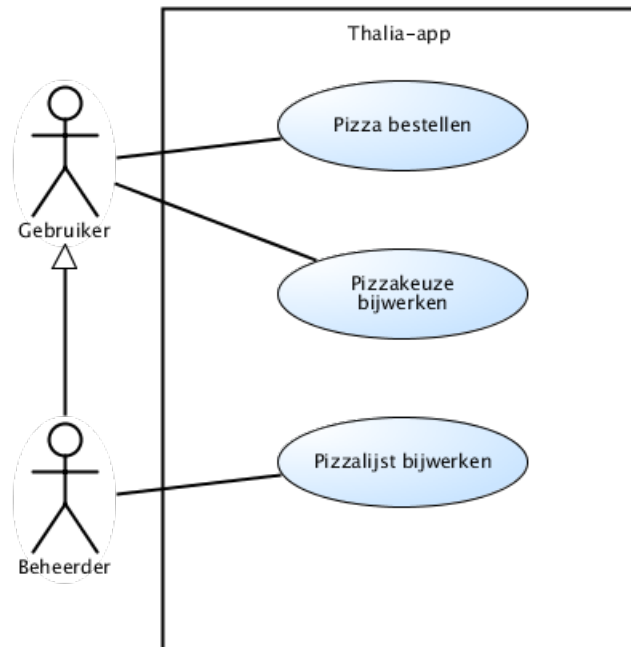
We zullen eerst de functionele eisen van het systeem beschrijven en daarna de niet functionele eisen. Vervolgens beschrijven we welke use case modellen hieruit volgen. Ook hier zullen we alles steeds opdelen in “basis-app” eisen en “optionele” eisen.

### 1.3.1 Functionele eisen

- Gebruikers en beheerders moeten in staat zijn pizza’s te bestellen van de borrelcie-pizzalijst.
- Beheerders moeten in staat zijn pizza’s aan de borrelcie-pizzalijst toe te voegen.

### 1.3.2 Niet-functionele eisen

- Documentation; het systeem moet een log bestand bijhouden. Zo kan er altijd worden nagegaan wie wat besteld heeft.
- Efficiency; het systeem moet efficiënter zijn dan de huidige pizzalijst.
- Extensibility; het systeem moet makkelijk uit te breiden zijn. Voor het geval dat er nieuwe pizza’s op de lijst komen etc.
- Performance; het systeem moet snel reageren. Anders raakt de gebruiker ongeduldig en verlangt deze weer terug naar de oude pizzalijst.
- Security; het systeem moet veilig zijn (wachtwoordbeveiliging). Gebruikers moeten inloggen met hun eigen Thalia-ID. Met dit ID heeft de gebruiker ook toegang tot allerlei andere functionaliteiten op de Thalia-site. Ook staan daar zijn of haar persoonsgegevens, die uiteraard privacygevoelig zijn.



Figuur 1: Integrated use case diagram

De bijbehorende use case survey is als volgt:

Use case naam	Beschrijving	Initiërende actoren
Pizza bestellen	Deze use case stelt de gebruiker/beheerder in staat een pizza van de pizzalijst te bestellen.	gebruiker, beheerder
Pizzakeuze bijwerken	Deze use case stelt de gebruiker/beheerder in staat een gemaakte keuze aan te passen of te verwijderen	gebruiker, beheerder
Pizzalijst bijwerken	Deze use case stelt de beheerder in staat een pizza aan de pizzalijst toe te voegen of te verwijderen	beheerder

Figuur 1 toont de integrated use case diagram. Hier is te zien wat de interactie met het systeem is. Een beheerder is uiteraard ook een gebruiker, hij kan alleen net wat meer. Verder is het allemaal niet zo ingewikkeld, aangezien onze app gefocust is op één ding, pizza's bestellen. Om die reden zullen we op de volgende pagina de use case "Pizza bestellen" uitwerken.

<b>Use Case:</b>	<b>Pizza bestellen (UC_01)</b>
Description	Deze use case stelt de gebruiker/beheerder in staat een pizza van de pizzalijst te bestellen.
Version	1.0
Basic course of events	<ul style="list-style-type: none"> <li>• 1) - Geb./Beh. - geeft aan pizza willen te bestellen.</li> <li>• 2) - Sys. - toont pizzalijst en vraagt gebruiker om zijn/haar keuze te maken.</li> <li>• 3) - Geb./Beh. - geeft aan welke pizza uit de lijst hij/zij wil en geeft aan klaar te zijn met bestellen.</li> <li>• 4) - Sys. - toont gekozen pizza en vraagt om bevestiging.</li> <li>• 5) - Geb./Beh. - gebruiker gaat akkoord.</li> <li>• 6) - Sys. - geeft bevestiging van keuze en toont de gebruiker het nog te betalen bedrag en geeft de gebruiker betaalinformatie.</li> </ul>
Alternate paths	<ul style="list-style-type: none"> <li>• 4A) - Sys. - geeft aan dat de gekozen pizza niet op de lijst voorkomt en vraagt de geb./beh. opnieuw zijn/haar keuze te maken.</li> <li>• “use case gaat terug naar 2”</li> </ul> <p>+ 5A) - Geb./Beh. - gaat niet akkoord.  + 5B) - Sys. - geeft aan keuze te annuleren.  + “use case stopt”</p>
Preconditions	<ul style="list-style-type: none"> <li>• De gebruiker/beheerder heeft nog geen pizza besteld.</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>• De gebruiker/beheerder staat met zijn/haar bestelde pizza op de lijst van mensen die pizza hebben besteld.</li> </ul>
Assumptions	<ul style="list-style-type: none"> <li>• Het systeem bevat een pizzalijst.</li> </ul>
Trigger	De beheerder en gebruiker hebben in de app aangegeven dat ze een pizza willen bestellen.

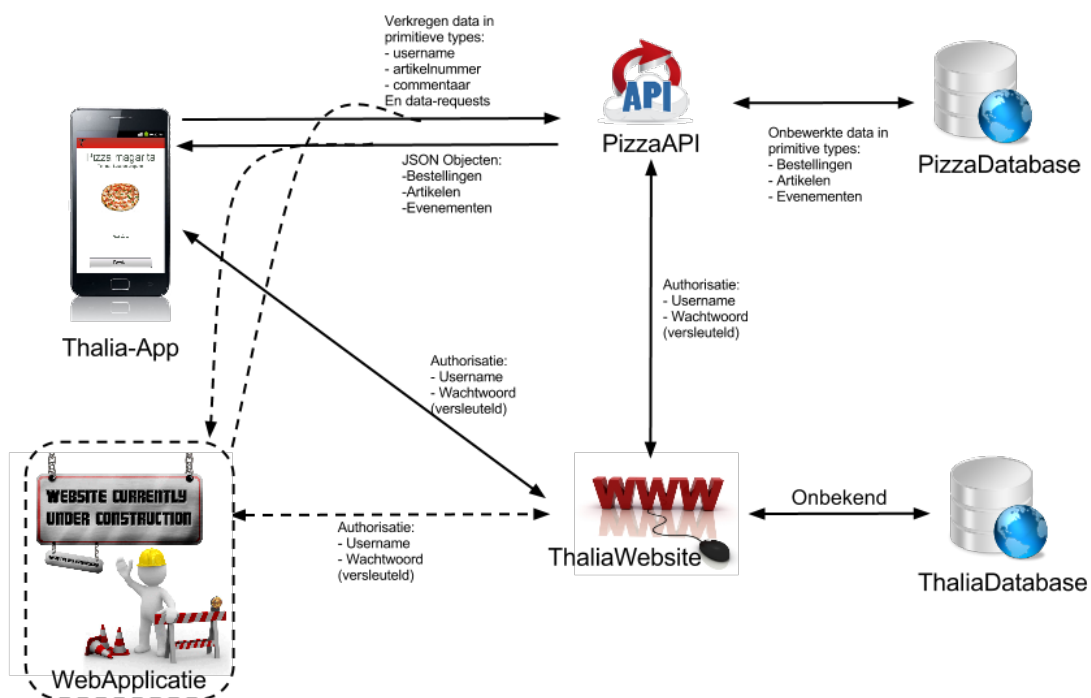
## 2 Ontwerp

### 2.1 Globaal ontwerp

In dit onderdeel zal het ontwerp van de Thalia-app worden uitgelegd. Met ontwerp wordt voornamelijk het 'technisch ontwerp' bedoeld. Dit kan al snel in details treden, deze zullen wij dan zo begrijpelijk mogelijk proberen te verwoorden.

In het hiernavolgend onderdeel zal de systeemarchitectuur in detail worden uitgelegd.

#### 2.1.1 Systeemarchitectuur



Onze systeemarchitectuur bestaat uit vijf basisonderdelen en een webapplicatie die later zal worden toegevoegd. De vijf basisonderdelen zijn:

- De Thalia-app
- De online Thalia-appdatabase
- De pizza-API



- De Thaliawebsite
- De Thaliadatabase

Allereerst zullen we per basissysteemonderdeel een uitleg geven en daarna een uitleg over de datacommunicatie.

### **2.1.2 Thalia-app**

De Thalia-app is een Android-applicatie voor mobiele telefoons met het Android OS. De Applicatie is geschreven in Java met behulp van een Eclipse ontwikkelomgeving en de SDK van Android. Deze App heeft als doel om via de Pizza-API verbinding te maken met de online database en met zowel de website als de database van Thalia.

### **2.1.3 Online database**

Om alle verkregen bestellingen centraal op te slaan hebben wij te maken met een centrale server, die alle bestellingen bijhoudt. Om dit soepel te laten verlopen hebben wij gekozen om hier een database van te maken. Er is gekozen om deze database te implementeren in MySQL aangezien dit tegenwoordig door iedere webhost wordt ondersteund en het best ondersteund wordt in PHP. Er is voor de beheerders de tool phpMyAdmin aanwezig om de pizzalijst te beheren.

### **2.1.4 Pizza-API**

De pizza-API is een Application Programming Interface die de Thalia-App in staat stelt om te communiceren met de online database. De API is geschreven in object-georiënteerd PHP en communiceert met de Thalia-app via een HTTP - en Post-request in samenwerking met Json. De communicatie van de API naar de online database verloopt via een MySQL query. Het verzoeken van data of een verzoek tot het toevoegen van data, verloopt via voorgedefinieerde verzoeken. Om de API flexibel te houden en makkelijk uit te breiden wordt deze object-georiënteerd opgebouwd.

### **2.1.5 Thaliawebsite & database**

De thaliawebsite is in ons systeem een onafhankelijke partij. De Thalia-App zendt via HTTP- en Post-requests de vraag voor authorisatie naar de site en ontvangt een response terug. Deze response is in HTML en kan worden uitgelezen voor bruikbare informatie. De website heeft zelf contact met de Thaliadatabase, dus alle gebruikers die bekend zijn bij Thalia en dus geregistreerd in hun database, kunnen inloggen in de Thalia-App.

### **2.1.6 Communicatie**

De communicatie binnen ons systeem is onder te verdelen in 3 type berichten:

- **Authorisatie:** Dit zijn kleine berichten, met daarin username en een versleuteld wachtwoord, tussen verschillende onderdelen en de Thaliawebsite om te controleren of de gebruiker ook echt een Thaliatid is. Deze berichten worden over het HTTP-protocol verstuurd in een plain text-formaat.
- **Berichten naar de Pizza-API:** Dit zijn ook kleine berichten waarin een onderdeel gegevens kan opvragen of bijdragen. Typische voorbeelden van dit soort berichten zijn: een verzoek naar informatie over een bestelling, een bestelling plaatsen of een verzoek voor alle beschikbare artikelen. Deze berichten worden over het HTTP-protocol verstuurd in een plain text-formaat.
- **Berichten van de Pizza-API:** Als de API een voorgedefinieerd verzoek heeft vervangen zal deze relatief grote berichten terugsturen. Om dit soepel te laten verlopen, worden deze berichten in JSON-formaat teruggestuurd over het HTTP-protocol.

## 2.2 Detailontwerp

Use Case “Pizza bestellen”

- 1) Gebruiker kiest in het hoofdmenu aan pizza te willen bestellen door op de desbetreffende button te klikken.
- 2) Er wordt een nieuwe intent gestart om naar de activity te gaan waarin de gebruiker zijn/haar pizza kan kiezen. De pizzalijst wordt gedownload van de online database en wordt ontvangen in Json en geparsed naar een ArrayList. Vervolgens wordt gecontroleerd of de gebruiker al een pizza heeft besteld, als dat het geval is wordt deze pizza bovenin de pizzalijst gezet en krijgt deze een afbeelding zodat de gebruiker weet dat deze besteld is. Tijdens al deze operaties ziet de gebruiker een voortgangsbalk zodat deze weet dat de app bezig is en voortgang boekt. Na voltooiën van deze operaties krijgt de gebruiker een lijst met afbeeldingen en namen te zien van de te bestellen product en wordt gevraagd een keuze te maken.
- 3) Het door de gebruiker gekozen product wordt als object geserialiseerd en in een bundle gestopt, vervolgens wordt deze meegestuurd met de intent die de volgende activity start. Ook wordt meegegeven of je deze pizza besteld hebt, zodat de volgende activity weet dat daar feedback over gegeven moet worden.
- 4) De activity die de gekozen pizza laat zien aan de gebruiker leest eerst de meegestuurde data van de bundle. Hierna krijgt de gebruiker het door hem/haar gekozen product te zien met bijbehorende feedback.
- 5) Na het drukken op de bestelbutton wordt door middel van een popup gevraagd om zijn/haar bestelling te bevestigen. Bij bevestiging wordt de API aangeroepen om de bestelling op te slaan. Het product weer geserialiseerd meegestuurd naar de volgende activity en krijgt de gebruiker ook hier weer een voortgangsbalk te zien.

6)De app ontvangt een Json object van de API en leest uit of de bestelling geplaatst is. Vervolgens wordt het (gedeserialiseerde) product (incl. afbeelding) nogmaals getoont en geeft betaalinformatie op het scherm weer.

## 2.3 Ontwerpverantwoording

Wat onze app onderscheidt van alle anderen is dat onze app een duidelijk doel heeft en we daar makkelijk op kunnen testen. Daarnaast is onze app volledig gericht op gebruiksvriendelijkheid, dit blijkt ook uit onze ontwerpbeslissingen.

Zoals bijvoorbeeld de keuze voor MySQL als databaseserver. We hadden hiervoor de keuze tussen verschillende databaseservers zoals PostgreSQL, SQLite en natuurlijk MySQL. We hebben hierbij verschillende afwegingen gemaakt. SQLite viel snel af aangezien deze database niet standaard door de meeste webhosts ondersteund wordt en wij daar weinig ervaring mee hebben. De keuze tussen de laatste twee was echter lastiger, hoewel Postgre ook niet standaard ondersteund wordt door de meeste webhost, biedt deze databaseserver een hoop extra mogelijkheden en er was een lid van het ontwikkelteam erg enthousiast over deze databaseserver. MySQL daartegenover wordt door iedere webhost ondersteund en er is binnen ons team ook ervaring mee, echter biedt MySQL totaal geen veiligheid en is de dataintegriteit erg slecht. Uiteindelijk woog de gebruiksvriendelijkheid voor de beheerders hier zwaarst en hebben wij dus gekozen voor MySQL aangezien dit voor de gebruiker het makkelijkst te hosten is.

Onze eerste opzet was een redelijk succes dit bleek ook uit ons eerste Usability test, welke hier te vinden is. Echter gaven gebruikers indirect wel aan dat er wat zaken duidelijker moesten en dat er meer informatie gegeven moest worden als de applicatie bijvoorbeeld aan het laden was.

Daarom hebben we daarna een aantal nieuwe ontwerpbeslissingen genomen, waaronder het ophalen van informatie van de server door android met AsyncTasks in plaats van het gebruiken van een model volgens het in java standaard Observer/Observable principe dat zelf zijn informatie ophaalt. Deze keuze hebben we ook voor de gebruiksvriendelijkheid gemaakt. Want hoewel het voor ons makkelijker was om het Observer/Observable principe te gebruiken en deze code beter her te gebruiken is, bieden AsyncTasks meer mogelijkheden die handig zijn specifiek in het gebruik binnen android. Dit zorgt ervoor dat de gebruiker snellere reacties en meer informatie krijgt als de gebruiker de app gebruikt. De gebruikers bleken hier ook erg tevreden mee te zijn, zoals te zien in de laatste usability-test.

## 3 Evaluatie

Dit onderdeel beschrijft onze laatste usability test. Hiermee hopen wij bevestiging van de gebruiker te krijgen of onze app doet wat de gebruiker van de app verwacht dat hij doet. Ook kunnen we via deze methode een lijst van punten ter verbetering samenstellen.

### 3.1 Testdoelen

Een testonderzoek heeft altijd testdoelen. Als testdoelen hebben we het volgende bedacht. Wij gaan:

- Testen of volgens de gebruiker de app niet traag reageert (Engaging).
- Bepalen of de app sneller is dan het gebruik van de huidige pizzalijst (Efficient).
- Testen of het gebruik van de app voor de gebruiker, die nog nooit een Thalia-pizzalijst in zijn handen heeft gehad, makkelijk is te begrijpen (Easy to learn).
- Bepalen of de gebruiker na het eerste gebruik van de app, de app nog een keer wil gebruiken (Engaging).
- Bepalen of de gebruiker met de app bereikt, wat hij verwacht had te bereiken (Effective).
- Bepalen of de gebruiker vindt dat de app niet te veel crasht (Error tolerant)
- Is het voor de gebruiker duidelijk dat bij een tweede keer bestellen, de eerste keer wordt overgeschreven?

We kiezen voor deze vragen omdat deze vragen precies de vijf E's bevatten (zie wat tussen haakjes staat). Uit onderzoek<sup>1</sup> blijkt dat de vijf E's leiden tot een compleet beeld in een usability-test. We nemen aan dat via dit systeem onze usability-test geen essentiële zaken over het hoofd ziet.

---

<sup>1</sup>Whitney Quesenberry ([www.wqusability.com](http://www.wqusability.com))

## 3.2 Gebruikers(sub)groepen

Via gebruikersgroepen komen wij erachter of de app voldoet voor een bepaalde groep. We hebben de volgende gebruikersgroepen opgesteld.

- **Thaliaan**

- **Thaliaan die vaak op borrels is te vinden:**

- Dit zijn personen die het huidige systeem van de pizzalijst gewend zijn. Deze groep heeft al wel eens een pizza besteld via de pizzalijst. De overgang naar de app, zou voor deze groep, niet heel erg moeilijk moet zijn.

- **Thaliaan die nog niet op een borrel is geweest:**

- Dit zijn personen die het huidige systeem van de pizzalijst niet kennen. Ze hebben dus nog nooit een pizza besteld en kennen het fenomeen ook niet. De kans is heel groot dat deze mensen in de toekomst naar een borrel gaan en een pizza willen bestellen. De app moet dus voor deze mensen makkelijk toegankelijk zijn.

### 3.3 Scenario's

Aangezien onze app niet heel erg ingewikkeld is, zijn er ook niet veel verscheidene scenario's te bedenken. Je zou uiteraard wel naar alle pizza's kunnen vragen, maar in wezen is dat niet verschillend van elkaar. Dus heeft het naar onze mening, niet veel zin om naar meer dan twee scenario's te vragen.

**De volgende scenario's zullen aan iedereen worden opgelegd:**

- Bestel een pizza Margarita en vertel mij aan het eind wat je moet betalen.
- Bestel een pizza Tonno en vertel mij aan het eind wat je moet betalen, bestel daarna een pizza Salami en vertel mij wat je moet betalen. Vertel mij daarna wat je denkt definitief besteld te hebben.

### 3.4 Methoden voor gegevensverzameling

Wij hebben ervoor gekozen om een interview achteraf te gebruiken. Het mooie van een interview achteraf is dat het zeer flexibel is. Men is dus in staat ter plekke extra vragen te bedenken die het gesprek met de gebruiker kunnen bevorderen. Ook ben je, door extra vragen te stellen, in staat meer informatie en meningen van de gebruiker te weten te komen.

Het gevaar is dat je dingen vergeet te vragen en niet consequent bent in je vraagstelling. Zo zouden er verschillende beelden kunnen ontstaan, terwijl de gebruiker misschien dezelfde mening heeft als de andere gebruiker. We zijn van mening dat onze realisatie, ons ervan weerhouden heeft deze fout te maken.

#### **Wat er in ieder geval gevraagd zal worden:**

- Wat had je van te voren verwacht van de app?
- Voldoet de app aan je verwachtingen?
- Wat vond je vervelend?
- Heb je bereikt wat je wilde bereiken?
- Vond je dat de app je genoeg response gaf, zodat je wist waar de app mee bezig was?
- Vond je dat de app snel genoeg reageerde?
- Was het duidelijk waar je moest zijn, toen de app je vertelde hoe je moest betalen?
- Heb je een tip hoe het beter zou kunnen?
- Vind je het een verbetering van de papieren pizzalijst?
- Vond je het een stabiele app, crashte hij niet teveel?

Nogmaals: het interview is flexibel en kan waar dan ook worden uitgebreid. Zolang er maar vragen in deze richting gesteld worden.

Dit heeft ons uiteindelijk ruwe interviewdata opgeleverd (zie bijlage) die wij vervolgens kunnen analyseren.

### 3.5 Procedure

Het werven van proefpersonen:

We zijn van plan mensen in onze omgeving te vragen. We kennen genoeg mensen die bereid zijn om onze app te testen, die ook nog eens voldoen aan al onze gebruikersgroepen. We zullen per gebruikersgroep twee proefpersonen zoeken. Op deze manier hebben we een duplo en op die manier rechtvaardigen we meer onze resultaten.

**Stappenplan voor het testen:**

- Geef de proefpersoon een mobiel en vraag aan hem/haar of hij/zij de app wilt openen.
- Vraag wat hij er tot nu toe van vindt.
- Geef de gebruiker een scenario die hij/zij moet uitvoeren.
- Vraag aan het eind wat de gebruiker ervan vindt (post-eerste indruk).
- Houd het interview.

Dit gehele proces zal worden opgenomen, zodat het aan het eind geanalyseerd kan worden. Ook levert een opname een verantwoording waar altijd naar kan worden terugverwezen.



## 3.6 Resultaten

### 3.6.1 Werving proefpersonen

Omdat we alleen de Thalianen hebben kunnen testen en alleen de Thaliaan zonder admin-rechten (dit in verband met de huidige functionaliteiten van de app), hebben we de volgende personen gevonden:

- **Thaliaan die vaak op borrels is te vinden:**  
We hebben Dré Hendriks en Tom Sanders bereid gevonden om onze app te testen, als zijnde Thaliaan die vaak op borrels is te vinden.
- **Thaliaan die nog niet op een borrel is geweest:**  
We hebben Gerco van Heerdt en Matjah Sonneveld bereid gevonden om onze app te testen, als zijnde Thaliaan die nog nooit op een borrel is geweest.

### 3.6.2 Resultaten interviews

De ruwe interviews zijn in een bijlage in dit document te vinden. De resultaten zullen we hier per testdoel opschrijven.

- **Testen of volgens de gebruiker de app niet traag reageert (Engaging).**  
Iedere proefpersoon vond dat de app snel genoeg reageerde. Op dit onderdeel is niks aan te merken.  
testdoel: gehaald.
- **Bepalen of de app sneller is dan het gebruik van de huidige pizzalijst (Efficient).**  
De proefpersonen Dré en Tom waren van mening, dat deze app zeker sneller is dan de huidige pizzalijst.  
testdoel: gehaald.
- **Testen of het gebruik van de app voor de gebruiker, die nog nooit een Thalia-pizzalijst in zijn handen heeft gehad, makkelijk is te begrijpen (Easy to learn).**  
Proefpersoon Gerco en Matjah vonden alles duidelijk en wisten meteen wat ze moesten doen. Gerco noemde het zelfs een verbetering op de vorige keer (fase 1).  
testdoel: gehaald.
- **Bepalen of de gebruiker na het eerste gebruik van de app, de app nog een keer wil gebruiken (Engaging)**  
Proefpersoon Dré maakte de opmerking dat hij dit wel als borrelcie-app zag zitten en zag zich dus de app nogmaals gebruiken.  
testdoel: gehaald.

- **Bepalen of de gebruiker met de app bereikt, wat hij verwacht had te bereiken (Effective).**  
Alle vier de proefpersonen lieten weten dat de app voldeed aan hun verwachtingen. Proefpersoon Tom vond het wel jammer dat de prijzen niet al in de hoofdlijst stonden.  
testdoel: gehaald.
- **Bepalen of de gebruiker vindt dat de app niet te veel crasht (Error tolerant)**  
Bij onze vier onderzoeken is de app nooit vastgelopen. Deze vraag is dus ook niet gesteld. Proefpersoon Tom constateerde wel dat hij ergens een melding te snel voorbij zag flitsen.  
Hierdoor zien wij dit testdoel als: gehaald.
- **Is het voor de gebruiker duidelijk dat bij een tweede keer bestellen, de eerste keer word overgeschreven?**  
Het was voor alle proefpersonen duidelijk wat er gebeurde bij een tweede keer bestellen. Ze begrepen dus dat hun bestelling werd overschreven.  
testdoel: gehaald.

We realiseren ons dat we geen compleet beeld zullen hebben van wat men van onze app vindt. Toch geeft dit ons voldoende informatie over fouten die nog in onze app zitten en dus genoeg materiaal waar wij mee aan de slag kunnen gaan.

### 3.7 Conclusies

In principe zijn al onze testdoelen gehaald. We hebben alleen wat cosmetisch commentaar gekregen:

- Berichten flitsen te snel voorbij (cosmetische fout); is simpel op te lossen en dat gaan we ook doen.
- Geen prijzen in hoofdlijst (cosmetische fout); we hebben ervoor gekozen dit zo te houden en zullen hier geen actie ondernemen.

We constateren hierbij dat onze app door de gebruikers is geaccepteerd en goedgekeurd. Met alleen nog één kleine cosmetische fout te verbeteren, verklaren wij deze app als voltooid.

## 4 Reflectie

### 4.1 Groep

We zijn als groep zeer tevreden over het project in zijn algemeenheid. Dit komt omdat we bereikt hebben wat we wilde maken. Er is een app gekomen, waarmee je pizza's kan bestellen.

Ook over het proces zijn we tevreden. Er was een duidelijke verdeling tussen Informatiekunde (Mats) en Informatica (Tim & Jip). Dat vertaalde zich in dat Mats voornamelijk het ontwerpdocument, usability tests en presentaties heeft gedaan. Tim & Jip hebben zich voornamelijk met het programmeren en de integratie met de Thaliasite bezig gehouden. We zijn allen van mening dat dit een succesvolle en duidelijke verdeling is geweest. Voor herhaling vatbaar.

Om nog een completer beeld te geven hebben we tijdens een workshop onze plus- en minpunten opgeschreven. De pluspunten waren:

- Usability test was positief
- Het maken van de database was een succes
- Pizza API werkt goed

De minpunten waren:

- We waren niet tevreden over het eerste ontwerpdocument
- De spelling van de documenten viel tegen

Deze punten spreken voor zich. We kunnen in ieder geval erover zeggen dat we ervan geleerd hebben. In de toekomst zullen we de spelling beter delegeren en documenten in hun totaliteit beter controleren.

## 4.2 Reflectie Mats

Als ik de begin- en eindsituatie van R&D bekijk, kan ik toch wel terugkijken op een succesvolle cursus. Ik heb echt het gevoel dat ik iets geleerd heb. Met name het onderdeel dat je kijkt naar wanneer een app nou daadwerkelijk een succes is, vond ik erg interessant. Wat wel jammer is bij deze cursus, is dat een informatiekundige niet volledig mee kan komen met de informaticamensen. Dit komt omdat de kennis van programmeren niet toereikend is. Dit resulteert met name in bijvoorbeeld onbegrip bij presentaties, als er over bepaalde technische implementatie wordt gepraat. Wat ik erg jammer vind. Ook heb ik weinig aan het programmeren kunnen bijdragen.

Het gevolg hiervan is dat er een tweedeling ontstaat in het vak, waardoor informatiekundigen zich alleen nog maar bezig houden met het onderzoeken en het presenteren ervan (in de vorm van presentaties en/of verslagen), terwijl de informaticamensen zich alleen maar met het programmeren bezig houden. Op zich een realistisch beeld van de toekomst (althans dat denk ik), maar misschien komt het dit vak niet geheel ten goede. Ik heb dan ook, daardoor lichtelijk het gevoel dat het vak daar zijn doel mist.

Als ik kijk naar mijn eigen inzet, ben ik zeer tevreden. Ik heb echt het gevoel dat ik heb bijgedragen wat ik bij heb kunnen dragen. Ik heb mooie presentaties in elkaar gezet, waar ik overigens later veel positieve reacties op heb gekregen. Dat doet mij altijd goed. Daarnaast kan ik ook trots terugkijken op een paar, dikke, maar prachtige verslagen. Er komt een stuk professionaliteit bij kijken, wat ik erg leuk vind. Ik heb door deze cursus het gevoel dat ik mijn ideeën en keuzes veel beter kan weergeven en verantwoorden. Dat komt omdat ik nu veel meer achtergrondkennis heb opgedaan over hoe anderen in de geschiedenis dat voor elkaar hebben gekregen.

Een mooie cursus, ben zeer tevreden en kijk met plezier uit naar R&D2!

## 4.3 Reflectie Tim

### Terugblik R&D1

R&D1, is een cursus waar ik al het hele jaar naar uitkeek, dit omdat het natuurlijk geweldig is om een eigen product neer te zetten en met behulp van technieken deze zoveel mogelijk te optimaliseren. Ook de ontwikkelingsfase leek mij, door het samenwerken in een projectgroep, bijzonder leerzaam. Achteraf bleek dit ook zeker waar, maar één punt had ik bij het vooruitzicht over het hoofd gezien en dat was de tijd die in zo'n applicatie gaat zitten. Natuurlijk betekent dat meer werk. Gelukkig betekent meer werk ook meer ervaring en is die tijd dus nuttig besteed.

### Wat heb ik geleerd?

Inhoudelijk heb ik veel geleerd bij R&D1, zoals het werken (via git) met een programmeerpartner op afstand. Ook was programmeren met de google android SDK mij helemaal onbekend. Maar zo heb ik natuurlijk ook geleerd over ontwikkelen, van ontwerpfase tot usability test en doorontwikkeling. Daar had ik niet of nauwelijks kennis van voordat ik aan deze cursus begon.

### Groepswork

Ik vond het erg prettig samenwerken met mijn groepspartners, dit kwam voornamelijk doordat we een duidelijke taakverdeling hadden afgesproken. Hierdoor schoot, ook mede door de grote hoeveelheid ingestoken tijd, het project lekker op.

Een aantal punten van deze cursus die mij sterk bij zullen blijven zijn:

- De liftzin
- "Jour best guess is not good enough" (Jakob Nielsen)
- Zeggen dat je je best gedaan hebt op het product is niet genoeg.

Al met al ben ik erg tevreden over ons product en de ontwikkeling ervan, we hebben grootste plannen met het verder uitwerken van onze applicatie en willen hiervan de standaard-app van Thalia maken.

## 4.4 Reflectie Jip

Ik heb genoten van deze cursus, hoewel ik, na in verschillende softwaredevelopingteams te hebben gezeten, weinig nieuws geleerd wat betreft werkwijze of samenwerken. Echter heb ik in deze cursus kennis gemaakt met het formeel beargumenteren van ontwerpbeslissingen. Dit bracht voor mij wel een geheel nieuwe kant aan het ontwerpen van een stuk software. Ik denk dat door mijn ervaring en een goede samenwerking binnen de groep, we binnen een korte en drukke tijd toch een erg mooi stukje software hebben neergezet dat hoewel niet vernieuwend voor de softwarewereld wel zonder problemen werkt en nog echt gebruikt kan worden ook. Ik denk dat wij als groep een goede dynamiek hebben, niet iedereen is het altijd met elkaar eens, maar dit geeft goede discussie waardoor je soms toch op oplossingen komt die je nooit had verwacht.

Ook de cursus zelf vond ik prettig gegeven, de colleges werden goed gegeven en waren erg informatief. Echter zijn er onderdelen zoals het ontwerpdocument waar meer documentatie bij mag, zodat je als deelnemer ook echt weet wat er van je verwacht wordt en je bijvoorbeeld niet achteraf te horen krijgt dat je een ORM-schema moest aanleveren in plaats van een UML-diagram.

Verder ben ik erg tevreden met deze cursus en met ons resultaat, laten we hopen dat R&D2 even succesvol is!

## Bijlagen

### Verwijzing naar code

De code van de ThaliaApp is hier te vinden. Dit is de gehele Android code waarmee u zelf de app kan compileren.

### Interviews (ruw)

#### Interview Gerco = G

M: "Gerco, je hebt nu tweede versie van onze Thalia-app in handen. Wat verwacht je ervan?"

G: "Ik hoop, voor jullie, dat er wat dingen verbeterd zijn."

M: "Ok, bestel voor mij dan maar een pizza Hawaii"

G: "Ok, even kijken. Ah ik zie nu inderdaad wat ik besteld heb. Ziet er goed uit."

M: "Wijzig je bestelling maar, ik wil er nu extra kaas bij namelijk."

G: "Ok geregeld!"

M: "Ok, wat is je eindconclusie?"

G: "Uhm, het is een hele verbetering. Dingen zijn net wat duidelijker dan eerst, goed gedaan!"

M: "Ok, dankjewel!"

## Interview Matjah = MJ

M: "Matjah, hier heb je onze app."

MJ: "Ziet er goed uit!"

M: "Dankjewel, wat verwacht je ervan?"

MJ: "Ik vermoed dat ik er pizza's mee kan bestellen -lacht-."

M: "Ok bestel maar een pizza Tonno voor me."

MJ: "Waar is de borrelcie?"

M: "Normaal, achter de bar."

MJ: "Ok, duidelijk"

M: "Vind je dat je duidelijk kan terugvinden wat je besteld hebt?"

MJ: "Ja, ziet er keurig uit!"

M: "Ok, dankjewel!"

### **Interview Dré = D**

M: "Dré je hebt onze app in handen."

D: "Ik wil deze kiezen."

M: "Deze?"

D: "Pizza Campagnola, met origano. Met, pizza margarita. Zo, besteld! Moet ik nog iets doen? Wacht, anders wijzig ik mijn pizza wel... met kaas"

M: "Vind je het een verbetering ten opzichte van de vorige keer?"

D: "Ja, dit erg... alles is een stuk duidelijker nu!"

M: "En ten opzichte van de huidige pizzalijst?"

D: "Ja, ik denk dat dit wel iets te bieden heeft aan onze borrelcie!"

M: "Ok, dankjewel!"

### **Interview Tom = T**

T: "Ik uhm, wil een pizza bestellen."

M: "Stel je voor je staat op een borrel en iemand zegt; PIZZA BESTELLEN! Ga je gang!"

T: "Ik ben al ingeloft"

M: "Nou, heel goed!"

T: "Nou, ik heb wel zin in een pizza Margarita, dat neem ik eigenlijk altijd. Ik vind het wel een beetje jammer dat hier geen prijzen bij staan -wijst naar hoofdlijst- maar... als ik op de pizza klik dan zal er vast wel iets verschijnen. Ah, kijk! Tomaat, kaas en oregano! Daar hou ik van -lacht-. Ik wil er graag een beetje Tonno bij! Ik denk dat ik op bestellen moet drukken?! Ja dat weet ik zeker. Ok, uh, ik denk dat ik hem heb besteld... maar, wie zijn de borrelcie? Volgens mij is Tim daar één van! Nou hoofdmenu, enige optie. Uhm, er vloog net een dialoogje voorbij?! Ik heb geen idee wat er gebeurde, maar ik zie een vinkje bij mijn pizza staan, dus neem aan dat het goed is. Moet ik verder nog wat doen?"

M: "Zou je eventueel de behoefte hebben om te veranderen. Voldoet het meteen aan wat je normaal ook zou doen?"

T: "Bij nader inzien, heb ik toch meer zin in een pasta bolognese en ik wil er kaas bij. Ben benieuwt of ik een melding krijg, dat ik mijn bestelling wijzig. Ik zie vervangen staan en er wordt mij gevraagd of ik zeker weet dat ik wil vervangen. Dat wil ik ja, mooi! Ok, ik zie dat mijn oude bestelling is weggegooid."

M: "Vind je het duidelijk?"

T: "Ik had wel een melding verwacht dat ik mijn oude bestelling weg zou gooien, in plaats van dat ik hem alleen maar vervang."

M: "Je gaf net zelf aan dat hij vroeg of je wilde vervangen."

T: "ja, ja dan is het goed"

M: "Beter dan de pizzalijst?"

T: "Ja, ik vind van wel ja. Als er maar duidelijk wordt omgeroepen wanneer de app gebruikt kan worden. Misschien dat er een of andere notificatie binnenkomt als er een pizza wordt besteld, maar uh, dat weet ik niet."

M: "Ja, ok, dankjewel!"