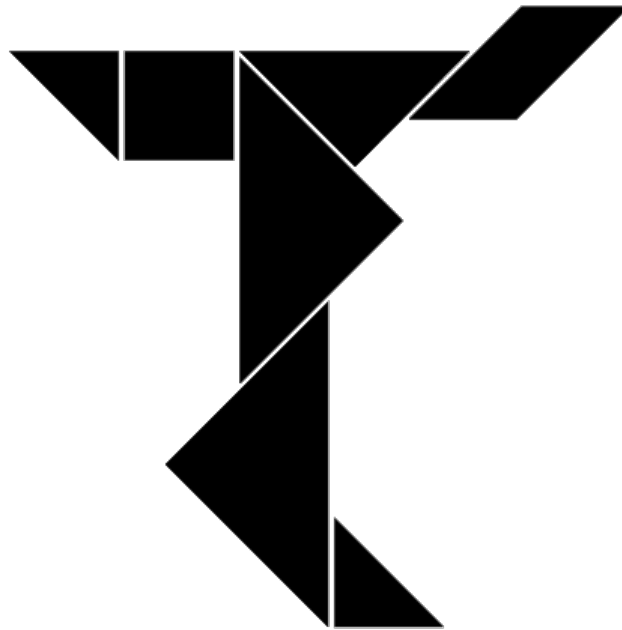


Thalia-app;
Het hoe, waarom en want?!

Appsinth; Jip Dekker, Tim Janssen, Mats Ouborg

18 mei 2012



Inhoudsopgave

1	Inleiding	3
2	Requirements	4
2.1	Functionele eisen	4
2.2	Niet functionele eisen	4
2.3	Use case model, use case survey & use cases	4
2.3.1	Use case model	5
2.3.2	Use case survey	6
2.4	Use cases	7
2.4.1	Pizza bestellen	7
2.4.2	Pizzalijst bijwerken	8
2.4.3	Borrelcie agenda bekijken	9
2.4.4	Borrelcie agenda bijwerken	10
2.4.5	Activiteiten inzien	11
2.4.6	Activiteiten toevoegen	12
2.4.7	Aanmelden activiteiten	13
3	Ontwerp	14
3.1	Globaal ontwerp	14
3.2	Gegevensontwerp & Detail-ontwerp per component	14
3.3	Gebruikersinterface	16
4	Planning	17
4.1	Toelichting	17

1 Inleiding

Dit document bevat het gehele ontwerp proces van de Thalia-app en is gemaakt door AppsinthTM.

Het idee om een Thalia app te maken kwam voornamelijk voort uit het feit dat er op de borrels een probleem werd ondervonden. Als mensen namelijk hun pizza bestelden, ging dat voorheen via een pizzalijst. Het probleem was alleen dat de pizzalijst soms uren rond ging en ons idee is dat dit veel sneller kan. Oplossing: een Thalia-app waarmee pizza's kunnen worden besteld.

Bij deze gedachtenstroom kwamen meerdere functies voor een Thalia-app voorbij, waarvan we in dit document een paar zullen behandelen. De functie om jezelf aan te melden voor een activiteit van Thalia en de functie om de borrelcie-agenda in te kunnen kijken hebben uiteindelijk dit document gehaald, maar we zien dit als iets optioneels. Om duidelijk in dit document onderscheid te maken tussen de basis-app (pizza bestellen) en de optionele dingen (borrelcie-agenda en activiteiten), hebben we deze respectievelijk blauw en groen gekleurd. Als er in dit document dus een blauw of een groen gekleurd onderdeel staat, weet u dus meteen waartoe dit behoort.

Dit document zal het ontwerp van de app beschrijven via 'use cases' en 'klassendiagrammen'. Waarschijnlijk zal het in het begin even wennen zijn deze te lezen, maar wij zijn ervan overtuigd dat met de gegeven toelichting de 'use cases' en 'klassendiagrammen' te begrijpen zijn.

2 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 (denk aan 'het systeem moet 24/7 online zijn').

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 extra' eisen.

2.1 Functionele eisen

Functionele basis-app eisen:

- Gebruiker 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.

Functionele optionele extra eisen:

- Gebruikers en beheerders moeten in staat zijn de borrelcie-agenda in te kunnen zien.
- Beheerders moeten in staat zijn de borrelcie-agenda bij te werken. (Items toevoegen, veranderen en verwijderen.)
- Gebruikers en beheerders moeten in staat zijn de Thalia-activiteitenagenda in te zien.
- Beheerders moeten in staat zijn de Thalia-activiteitenagenda bij te werken. (Items toevoegen, veranderen en verwijderen.)
- Gebruikers en beheerders moeten in staat zijn zich aan te kunnen melden voor een activiteit uit de Thalia-activiteitenagenda.

2.2 Niet-functionele eisen

Niet-functionele basis-app eisen:

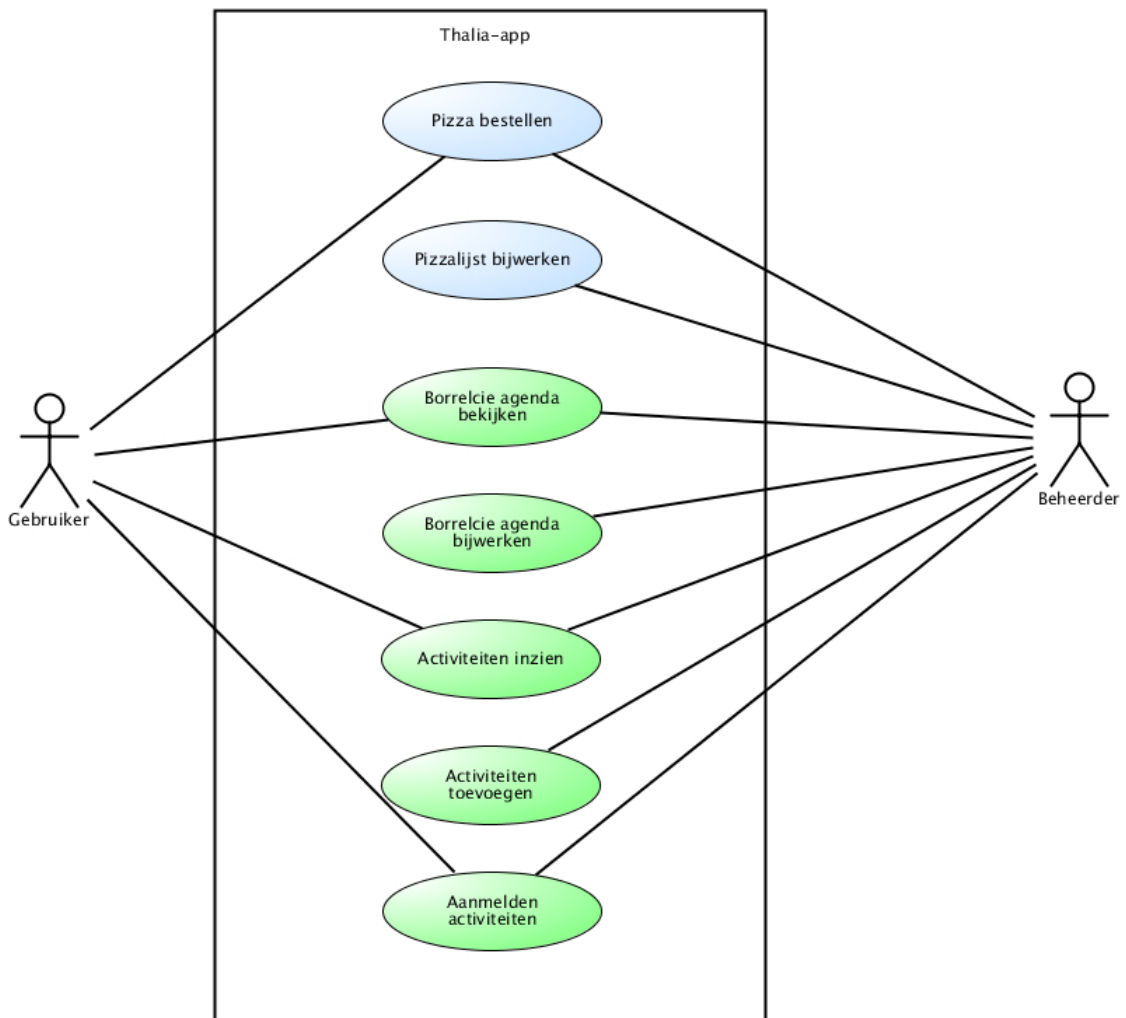
- Het systeem moet 24/7 online zijn.
- Het systeem moet makkelijk in onderhoud zijn.

Niet-functionele optionele extra eisen:

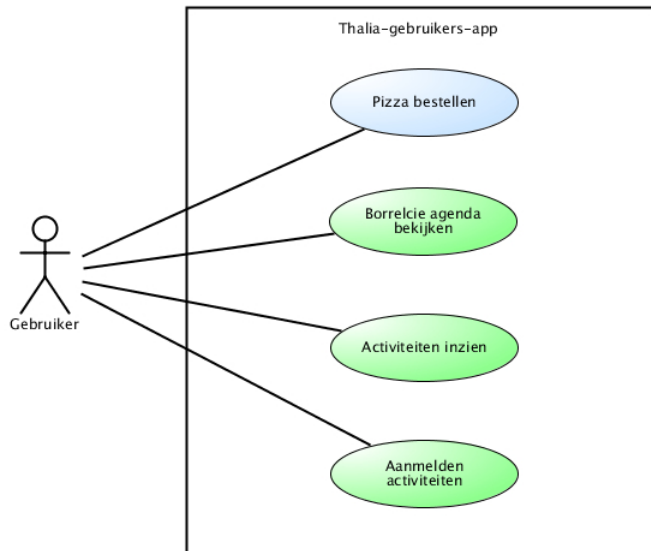
- Het systeem moet 24/7 online zijn.
- Het systeem moet makkelijk in onderhoud zijn.

2.3 Use case model, use case survey & use cases

In dit deel zullen we de use case modellen weergeven en beschrijven, dit zijn figuren die duidelijk maken met welke interacties (communicatie met het systeem) de gebruiker/beheerder te maken krijgt. Om duidelijk het onderscheid te maken tussen onze 'basis-app' en de 'optionele extra's', zijn deze in de use cases respectievelijk aangegeven met blauw en groen.



Figuur 1: Integrated use case diagram

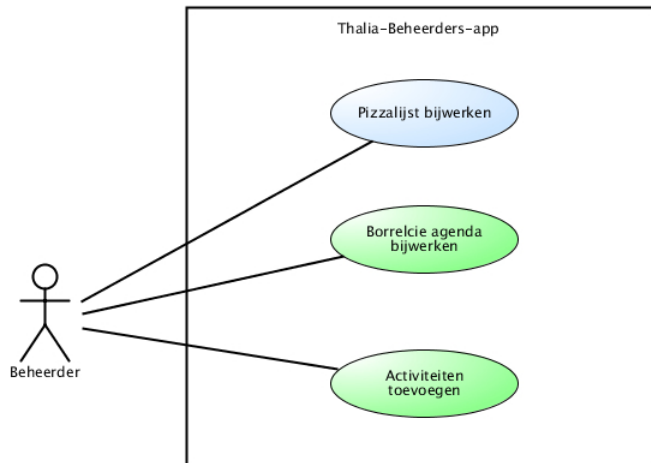


Figuur 2: Integrated use case diagram voor de 'gebruikers-app'

2.3.1 Use case model

Figuur 1 toont het volledige integrated use case diagram van de Thalia-app.

We hebben aangegeven dat we twee verschillende apps gaan maken, namelijk de 'gebruikers-app' (zie figuur 2) en de 'beheerders-app' (zie figuur 3). Dit houdt in dat er eigenlijk twee systemen gebouwd gaan worden. Vandaar dat we onze hierboven getoonde 'Integrated use case diagram' hebben opgesplitst zodat hij toepasbaar is op de 'gebruiker-' en de 'beheerders-app'.



Figuur 3: Integrated use case diagram voor de 'beheerders-app'

2.3.2 Use case survey


Dit is een overzicht van alle 'use cases' die de app beschrijven; gegeven de naam, korte beschrijving van de 'use case' en de persoon (actor) die via de 'use case' communiceert met het systeem.

Verder zullen de individuele use cases verder worden uitgewerkt.

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
Pizzalijst bijwerken	Deze use case stelt de beheerder in staat een pizza aan de pizzalijst toe te voegen of te verwijderen	beheerder
Borrelcie-agenda bekijken	Deze use case stelt de gebruiker/beheerder in staat de borrelcie-agenda te bekijken.	gebruiker, beheerder
Borrelcie-agenda bijwerken	Deze use case stelt de beheerder in staat items aan de borrelcie-agenda toe te voegen en/of te verwijderen.	beheerder
Activiteiten inzien	Deze use case stelt de gebruiker/beheerder in staat om activiteiten van Thalia in te zien.	gebruiker, beheerder
Activiteiten toevoegen	Deze use case stelt de beheerder in staat activiteiten van Thalia toe te voegen.	beheerder
Aanmelden activiteiten	Deze use case stelt de gebruiker/beheerder in staat zich aan te melden voor een activiteit van Thalia.	gebruiker, beheerder

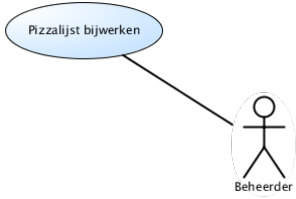
2.4 Use cases

2.4.1 Pizza bestellen

Use Case:	Pizza bestellen (UC_01)
Use case diagram	 <pre> graph TD UC((Pizza bestellen)) --- G[Gebruiker] UC --- B[Beheerder] </pre>
Description	Deze use case stelt de gebruiker/beheerder in staat een pizza van de pizza lijst te bestellen.
Version	1.0
Basic course of events	<ul style="list-style-type: none"> • 1) - Geb./Beh. - geeft aan pizza willen te bestellen. • 2) - Sys. - toont pizzalijst en vraagt gebruiker om zijn/haar keuze te maken. • 3) - Geb./Beh. - geeft aan welke pizza uit de lijst hij/zij wil en geeft aan klaar te zijn met bestellen. • 4) - Sys. - toont gekozen pizza en vraagt om bevestiging. • 5) - Geb./Beh. - gebruiker gaat akkoord. • 6) - Sys. - geeft bevestiging van keuze en toont de gebruiker het nog te betalen bedrag en geeft de gebruiker betaal informatie.
Alternate pats	<ul style="list-style-type: none"> • 4A) - Sys. - geeft aan dat de gekozen pizza niet op de lijst voorkomt en vraagt de geb./beh. opnieuw zijn/haar keuze te maken. • 'use case gaat terug naar 2)' + 5A) - Geb./Beh. - gaat niet akkoord. + 5B) - Sys. - geeft aan keuze te annuleren. + 'use case stopt'
Preconditions	<ul style="list-style-type: none"> • De gebruiker/beheerder heeft nog geen pizza besteld.
Postconditions	<ul style="list-style-type: none"> • De gebruiker/beheerder staat met zijn/haar bestelde pizza op de lijst van mensen die pizza hebben besteld.

Assumptions	<ul style="list-style-type: none">• Het systeem bevat een pizzalijst.
Trigger	De beheerder en gebruiker hebben in de app aangegeven dat ze een pizza willen bestellen.

2.4.2 Pizzalijst bijwerken


Use Case:	Pizzalijst bijwerken (UC_02)
Use case diagram	
Description	Deze use case stelt de beheerder in staat een pizza aan de pizza lijst toe te voegen of te verwijderen.
Version	1.0
Basic course of events	<ul style="list-style-type: none"> • 1) - Beh. - geeft aan de pizzalijst te willen bijwerken. • 2) - Sys. - toont huidige pizzalijst en vraagt de beheerder welk onderdeel hij/zij wil veranderen. • 3) - Beh. - geeft aan welke pizza uit de lijst hij/zij wil bewerken. • 4) - Sys. - toont gekozen pizza en vraagt nieuwe naam. • 5) - Beh. - geeft nieuwe naam op. • 6) - Sys. - vraagt om nieuwe prijs. • 7) - Beh. - geeft nieuwe prijs. • 8) - Sys. - toont pizza met naam en prijs en vraagt om bevestiging. • 9) - Beh. - gaat akkoord. • 10) - Sys. - geeft aan dat pizzalijst is bijgewerkt.
Alternate pats	<ul style="list-style-type: none"> • 4A) - Sys. - geeft aan dat pizza niet in lijst voorkomt en vraagt aan beheerder of hij/zij de pizza aan lijst wilt toevoegen. • 4B.1) - Beh. - geeft aan de pizza niet te willen toevoegen. 'use case gaat verder bij 2' • 4B.2) - Beh. - geeft aan pizza te willen toevoegen aan pizza lijst. • 4C) - Sys. - vraagt beheerder om prijs. • 4D) - Beh. - geeft prijs. • 4E) - Sys. - geeft pizza met naam en prijs weer en vraagt beheerder om bevestiging. • 4F.1) - Beh. - gaat akkoord. • 4F.2) - Beh. - gaat niet akkoord. 'use case stopt' • 4G) - Sys. - geeft aan dat pizza lijst is bijgewerkt. <p>+ 9A) - Beh. - gaat niet akkoord. 'use case stopt'</p>

Preconditions	N.V.T.
Postconditions	N.V.T.
Assumptions	<ul style="list-style-type: none"> • Het systeem bevat een pizzalijst.
Trigger	De beheerder heeft aangegeven dat hij/zij de pizzalijst wil aanpassen.

2.4.3 Borrelcie agenda bekijken

Use Case:	Borrelcie agenda bekijken (UC_03)
Use case diagram	<pre> graph TD Gebruiker((Gebruiker)) --- UC((Borrelcie agenda bekijken)) Beheerder((Beheerder)) --- UC </pre>
Description	Deze use case stelt de gebruiker/beheerder in staat de borrelcie agenda te bekijken.
Version	1.0
Basic course of events	<ul style="list-style-type: none"> • 1) - Geb./Beh. - geeft aan de borrelcie agenda te willen bekijken. • 2) - Sys. - toont de borrelcie agenda.
Alternate pats	<ul style="list-style-type: none"> • N.V.T.
Preconditions	N.V.T.
Postconditions	N.V.T.
Assumptions	<ul style="list-style-type: none"> • Er is een borrelcie agenda aanwezig in het systeem.
Trigger	De beheerder en/of de gebruiker heeft in het systeem aangegeven de borrelcie agenda te willen bekijken.

2.4.4 Borrelcie agenda bijwerken

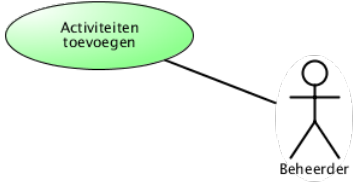
Use Case:	Borrelcie agenda bijwerken (UC_04)
Use case diagram	
Description	Deze use case stelt de beheerder in staat items aan de borrelcie agenda toe te voegen en/of te verwijderen.
Version	1.0
Basic course of events	<ul style="list-style-type: none"> • 1) - Beh. - geeft aan de borrelcie agenda bij te willen werken. • 2) - Sys. - vraagt om item dat moet worden bijgewerkt. • 3) - Beh. - geeft naam op van item dat moet worden bijgewerkt. • 4) - Sys. - toont item en vraagt om nieuwe naam. • 5) - Beh. - geeft nieuwe naam. • 6) - Sys. - vraagt nieuwe datum en tijd. • 7) - Beh. - geeft nieuwe datum en tijd. • 8) - Sys. - vraagt nieuwe beschrijving. • 9) - Beh. - geeft nieuwe beschrijving. • 10) - Sys. - toont bijgewerkte agenda item en vraagt om bevestiging. • 11) - Beh. - geeft bevestiging. • 12) - Sys. - geeft aan dat agenda is bijgewerkt.
Alternate pats	<ul style="list-style-type: none"> • 4A) - Sys. - geeft aan dat item niet bestaat en vraagt beheerder of hij/zij deze wilt toevoegen aan agenda. • 4B.1) - Beh. - geeft aan nieuw item te willen toevoegen. • 4B.2) - Beh. - geeft aan nieuw item niet te willen toevoegen. 'use case stopt' • 4C) - Sys. - vraagt naam nieuw item. • 4D) - Beh. - geeft naam nieuw item. • 4E) - Sys. - vraagt datum en tijd nieuw item. • 4F) - Beh. - geeft datum en tijd nieuw item. • 4G) - Sys. - vraagt beschrijving nieuw item. • 4H) - Beh. - geeft beschrijving nieuw item. • 4I) - Sys. - geeft nieuw item weer en vraagt om bevestiging. • 4J.1) - Beh. - geeft akkoord melding. • 4J.2) - Beh. - gaat niet akkoord. 'use case stopt' • 4K) - Sys. - geeft melding dat item aan agenda is toegevoegd. <p>+ 11A) - Beh. - gaat niet akkoord. 'use case stopt'</p>

Preconditions	N.V.T.
Postconditions	N.V.T.
Assumptions	<ul style="list-style-type: none"> • Er is in het systeem een borrelcie agenda aanwezig.
Trigger	De beheerder heeft in het systeem aangegeven dat hij/zij de borrelcie agenda wil bijwerken.

2.4.5 Activiteiten inzien

Use Case:	Activiteiten inzien (UC_05)
Use case diagram	<pre> graph TD G[Gebruiker] --- UC((Activiteiten inzien)) B[Beheerder] --- UC </pre>
Description	Deze use case stelt de gebruiker/beheerder in staat om activiteiten van Thalia in te zien.
Version	1.0
Basic course of events	<ul style="list-style-type: none"> • 1) - Geb./Beh. - geeft aan de activiteiten van Thalia in te willen zien. • 2) - Sys. - toont lijst met activiteiten van Thalia.
Alternate pats	<ul style="list-style-type: none"> • N.V.T.
Preconditions	N.V.T.
Postconditions	N.V.T.
Assumptions	<ul style="list-style-type: none"> • Er is een Thalia activiteiten lijst in het systeem aanwezig.
Trigger	De beheerder en/of de gebruiker heeft in het systeem aangegeven de activiteiten lijst van Thalia te willen inzien.

2.4.6 Activiteiten toevoegen

Use Case:	Activiteiten toevoegen (UC_06)
Use case diagram	
Description	Deze use case stelt de beheerder in staat activiteiten van Thalia toe te voegen.
Version	1.0
Basic course of events	<ul style="list-style-type: none"> • 1) - Beh. - geeft aan activiteit te willen toevoegen. • 2) - Sys. - systeem vraagt naam nieuwe activiteit. • 3) - Beh. - geeft naam nieuwe activiteit. • 4) - Sys. - vraagt datum en tijd van nieuwe activiteit. • 5) - Beh. - geeft datum en tijd van nieuwe activiteit. • 6) - Sys. - vraagt om beschrijving van activiteit. • 7) - Beh. - geeft beschrijving van nieuwe activiteit. • 8) - Sys. - vraagt deadline datum en tijd voor inschrijvingen. • 10) - Beh. - geeft deadline datum en tijd voor inschrijvingen. • 11) - Sys. - toont nieuwe activiteit en vraagt om akkoord. • 12) - Beh. - geeft bevestiging. • 13) - Sys - geeft aan dat activiteit aan Thalia agenda is toegevoegd.
Alternate pats	<ul style="list-style-type: none"> • 4A) - Sys. - geeft aan dat naam al voorkomt in agenda en vraagt beheerder of hij/zij deze wil bijwerken. • 4B.1) - Beh. - geeft aan dit item te willen bijwerken. • 4B.2) - Beh. - geeft aan dit item niet te willen bijwerken. 'use case stopt' • 4C) - Sys. - vraagt nieuwe naam activiteit. • 4D) - Beh. - geeft nieuwe naam activiteit. • 4E) - Sys. - vraagt nieuwe datum en tijd activiteit. • 4F) - Beh. - geeft nieuwe datum en tijd activiteit. • 4G) - Sys. - vraagt nieuwe beschrijving activiteit. • 4H) - Beh. - geeft nieuwe beschrijving activiteit. • 4I) - Sys. - vraagt nieuwe deadline datum en tijd inschrijvingen. • 4J) - Beh. - geeft nieuwe deadline datum en tijd inschrijvingen. • 4K) - Sys. - toont nieuwe gewijzigde activiteit en vraagt om bevestiging. • 4L.1) - Beh. - geeft bevestiging. • 4L.2) - Beh. - geeft geen bevestiging. 'use case stopt' <p>+ 12A) - Beh. - geeft gaat niet akkoord. 'use case stopt'</p>

Preconditions	N.V.T.
Postconditions	Een nog niet bestaande item moet na toevoeging aan de agenda zijn toegevoegd.
Assumptions	<ul style="list-style-type: none"> • Er is een Thalia activiteiten lijst in het systeem aanwezig.
Trigger	De beheerder heeft in het systeem aangegeven een activiteit in de Thalia activiteiten lijst te willen toevoegen.

2.4.7 Aanmelden activiteiten

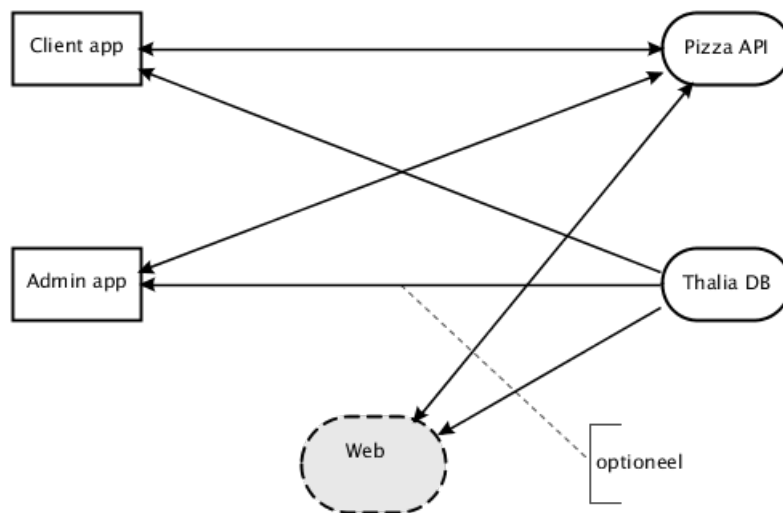
Use Case:	Aanmelden activiteiten (UC_07)
Use case diagram	
Description	Deze use case stelt de gebruiker/beheerder in staat zich aan te melden voor een activiteit van Thalia.
Version	1.0
Basic course of events	<ul style="list-style-type: none"> • 1) - Geb./Beh. - geeft aan zich aan te willen melden voor een activiteit. • 2) - Sys. - geeft lijst van Thalia activiteiten weer. • 3) - Geb./Beh. - geeft aan voor welke activiteit hij/zij zich wilt aanmelden. • 4) - Sys. - toont activiteit en vraagt om bevestiging. • 5) - Geb./Beh. - geeft bevestiging. • 6) - Sys. - geeft 'betaal tegoed', 'datum' en 'tijd' van activiteit, en meld dat aanmelden is gelukt.
Alternate pats	<ul style="list-style-type: none"> • 4A) - Sys. - geeft aan dat activiteit niet bestaat. <i>'use case gaat terug naar 2)'</i> • 5A) - Geb./Beh. - geeft geen bevestiging. <i>'use case stopt'</i>
Preconditions	<ul style="list-style-type: none"> • Gebruiker/beheerder is nog niet aangemeld voor evenement.
Postconditions	<ul style="list-style-type: none"> • Gebruiker/beheerder is aangemeld voor evenement.
Assumptions	<ul style="list-style-type: none"> • Er is een Thalia activiteiten lijst in het systeem aanwezig.
Trigger	De beheerder en/of de gebruiker heeft in het systeem aangegeven zich te willen aanmelden voor een activiteit.

3 Ontwerp

In dit onderdeel zal het ontwerp van de Thalia-app worden uitgelegd. Met ontwerp wordt het voornamelijk 'technisch ontwerp' bedoeld. Hier zal dus al gauw in UML klassendiagrammen worden gepraat, maar we zijn ervan overtuigd dat deze diagrammen met de juiste toelichting begrijpelijk zijn voor niet technisch onderlegde mensen.

3.1 Globaal ontwerp

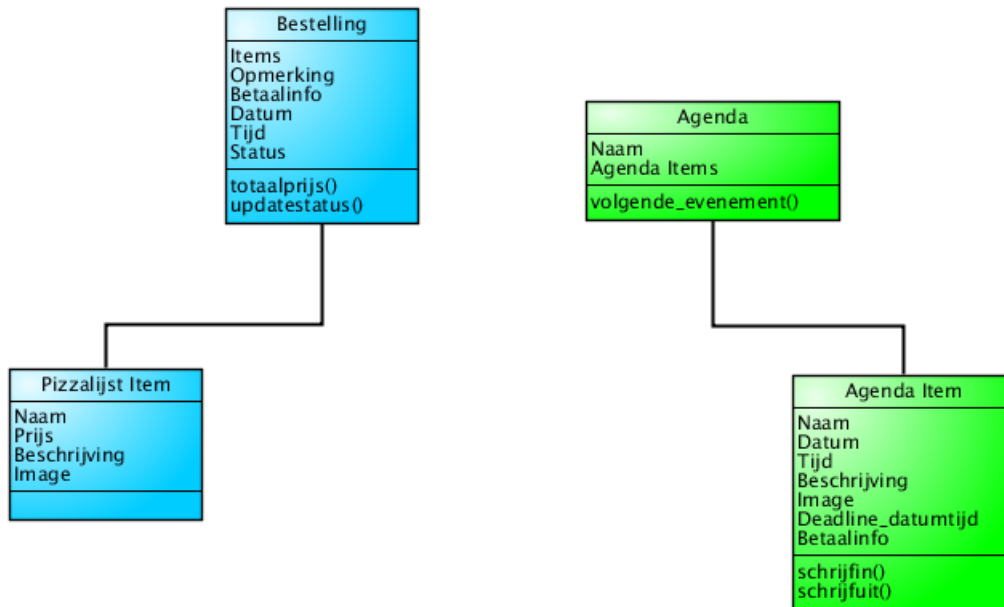
In figuur 4 ziet u het globale overzicht van de architectuur die de Thalia-app werkend gaat maken.



Figuur 4: Globaal ontwerp weergave van de architectuur van de Thalia-app.

Per onderdeel zal hier uitleg worden gegeven:

- Client app: De Client-App is de android-applicatie waarmee de 'gebruiker' zijn/haar acties kan verrichten zoals het bestellen van een pizza.
- Admin app: Admin-App is de android-applicatie waarmee 'beheerders' de administratie van het gehele systeem kunnen bijwerken.
- Pizza API: De Pizza-API is een API die kan communiceren met onze online database.
- Thalia DB: Thalia-DB is de database van Studievereniging Thalia, hierin staan de gegevens van de leden van de vereniging.
- De Web-interface zorgt voor een vervanging van de Client- en Admin-App indien er geen android device aanwezig is.



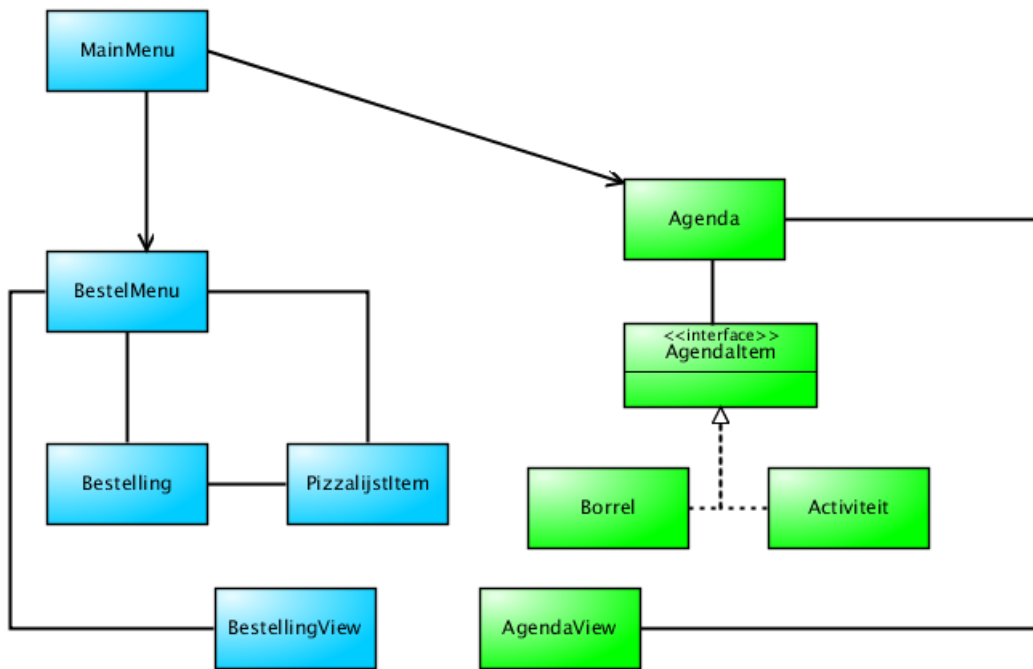
Figuur 5: Een verwacht datamodel.

3.2 Gegevensontwerp & Detailontwerp per component

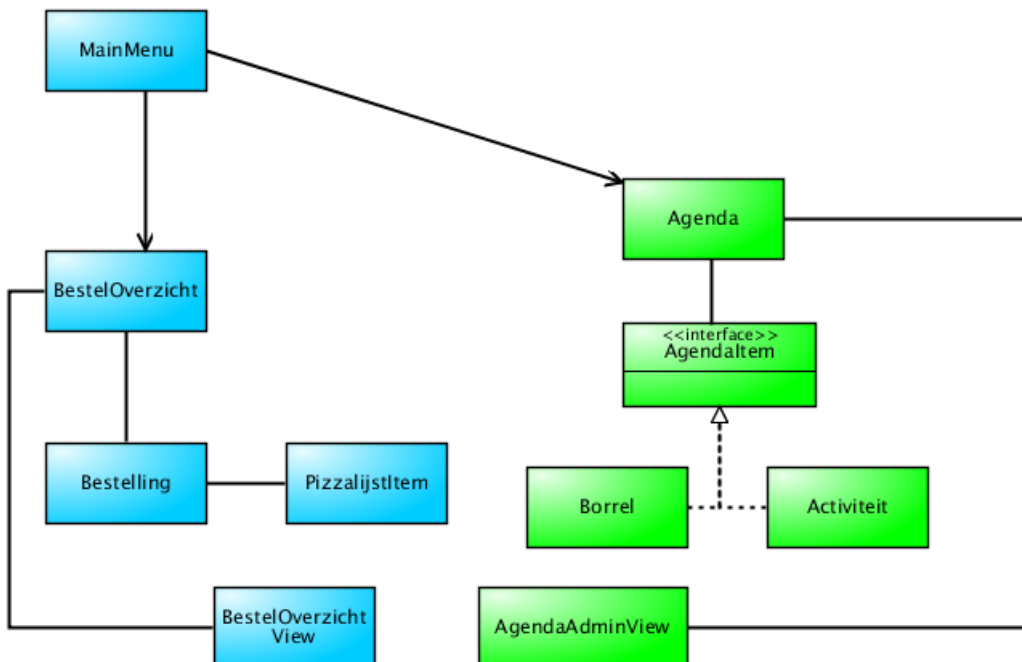
In figuur 5 ziet u het verwachte datamodel, hierin specificeren we de data die we gebruiken voor het bestellen van pizza's en de data die we nodig zouden hebben voor een agenda. Er nog data verwerkt moeten worden om de gebruikers te identificeren, echter is deze nog niet bekend in afwachting van de beheerder van de Thalia-database.

In figuur 6 ziet u een klassediagram van de gebruikersversie van de app, hierin hebben wij pijlen toegevoegd als aanduiding van een overgang der activiteit. Beide views en het 'MainMenu' zijn in dit klassediagram als enkele klassen weergegeven, echter zullen deze mogelijk opgesplitst om het MVC-patroon aan te houden.

In figuur 7 ziet u een klassediagram van de beheerdersversie van de app, hierin hebben wij pijlen toegevoegd als aanduiding van een overgang der activiteit. Het grote verschil met de gebruikersversie is dat deze app een overzicht van bestellingen geeft en daarom een andere weergave gebruikt. De Agenda is ongeveer hetzelfde, echter heeft de gebruiker hier de mogelijkheid details aan te passen. Beide views en het 'MainMenu' zijn in dit klassediagram als enkele klassen weergegeven, echter zullen deze mogelijk opgesplitst om het MVC-patroon aan te houden.



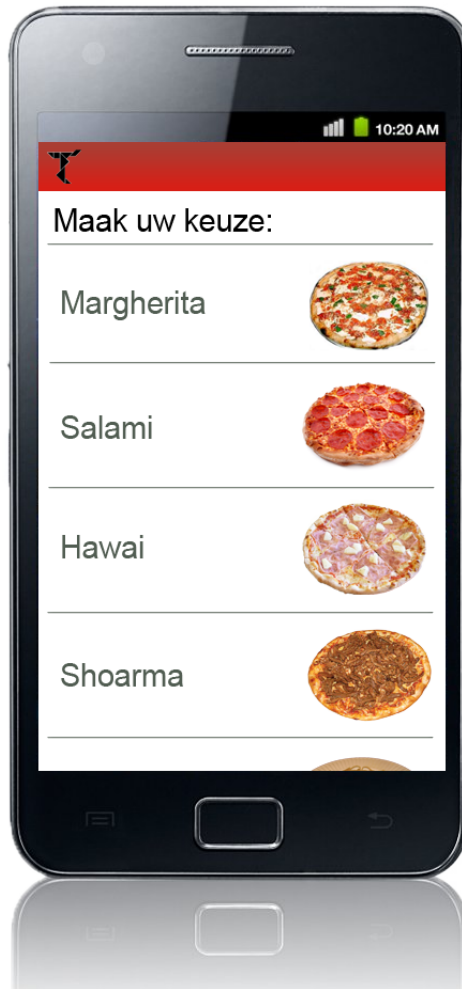
Figuur 6: Gebruikers-app.



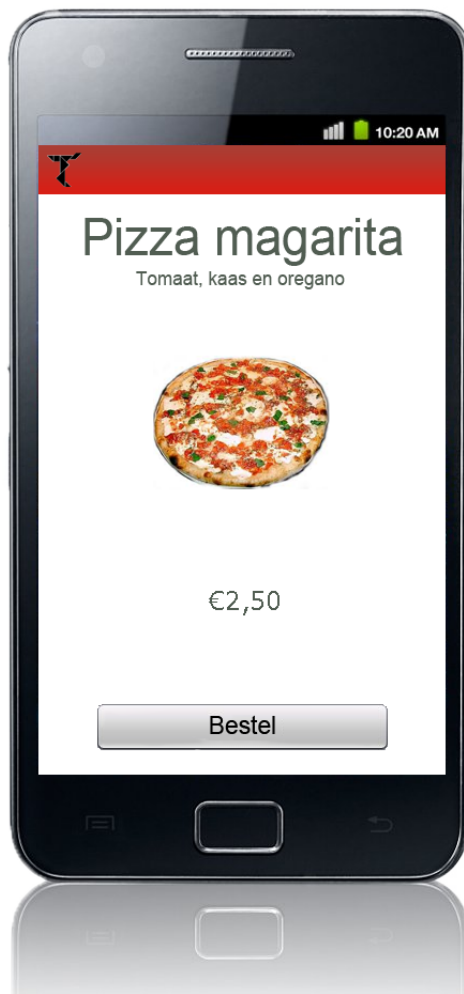
Figuur 7: Beheerders-app.

3.3 Gebruikersinterface

In figuur 8 en figuur 9 ziet u een voorbeeld van hoe wij de Thalia-app ons voorstellen. We gaan voor een clean en duidelijk ontwerp en met name de duidelijkheid staat bij ons hoog in het vaandel. Als u naar figuur 9 kijkt, is waarschijnlijk meteen duidelijk wat er in deze situatie van de gebruiker gevraagd wordt. Dit proberen wij in de gehele app toe te passen.

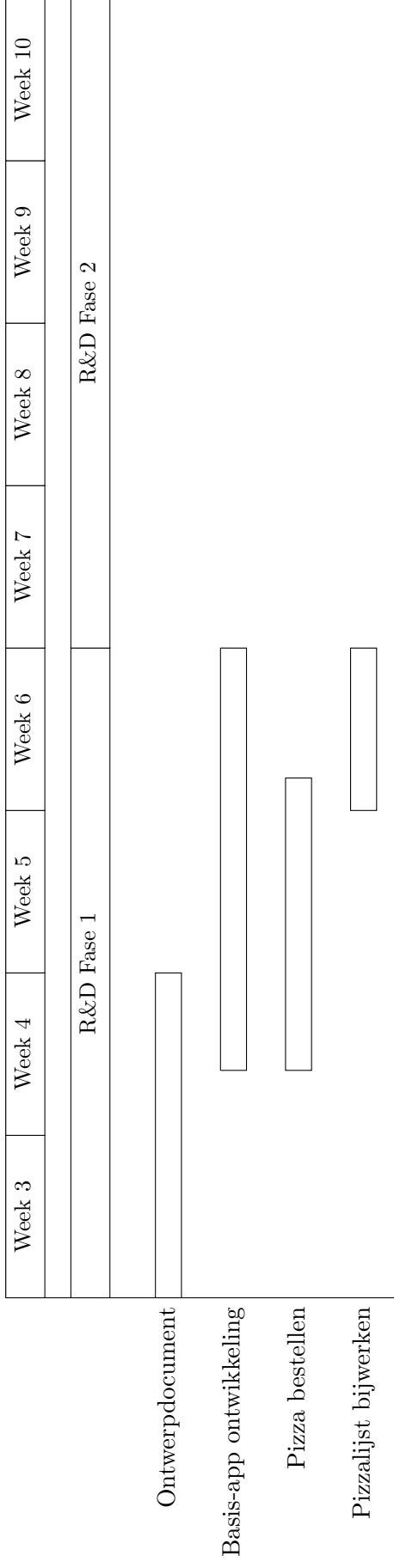


Figuur 8: Voorbeeld van de Thalia-app interface.



Figuur 9: Voorbeeld van de Thalia-app interface.

4 Planning



4.1 Toelichting

Aangezien wij het heel lastig vinden om in deze week al voor fase 2 te plannen, hebben we die open gelaten. Vooral nog zijn alleen de 'basis-app' dingen ingepland. Voor de 'optionele-app' dingen gaan we alleen tijd vrijmaken als die er is, dus als de hier geplande dingen meevallen.