

Rapport Online Shop

Franciscus Fekkes
Benjamin Vandersmissen

November 4, 2018

Contents

1	Inleiding	2
2	Planning	2
3	Actors	2
4	Unexpected Changes and exceptions	2
5	Extra Vragen Voor de Use Cases	3
6	Scenario's	3
7	Overige niet-functionele requirements	3
8	Objecten in het systeem	3
9	CRC cards	4
9.1	Scenario A: Display Catalog to user	4
9.2	Scenario B: Add Item to cart	4
9.3	Scenario C: Remove item from cart	4
9.4	Scenario D: Admin changes Order	4
9.5	Scenario E: Place Order and pay	4
10	Object-oriented vs functional decomposition	4
11	Bijlage	5
11.1	Functional Requirements	5
11.2	Non-Functional Requirements	5

1 Inleiding

Dit document beschrijft de voorbereiding voor een online shop. De requirements zijn te vinden in de bijlage. Er is een planning opgesteld, een risicoanalyse gemaakt, er zijn aantal scenario's onderzocht en de basis klassen zijn bedacht.

2 Planning

De PERT chart met de onderdelen van het project is gegeven in figuur 1. De *earliest start date* wordt gezien vanaf week 0. Alle *earliest start dates* en *latest end dates* zijn te zien in tabel 4. Het kritieke pad is aangegeven op figuur 1. De rode nodes liggen op de kritieke paden. Geen van deze nodes mogen een vertraging geven anders gaat dat ergens anders moeten worden ingehaald of het project zal later worden opgeleverd.

Om te zien hoeveel werknemers er nodig zullen zijn om alles in goede banen te leiden is er een Gantt chart opgezet. Deze is te zien op figuur 2. Als er oneindig veel mankracht gebruikt kan worden is alles af binnen 17 weken (gerekend vanaf week 1). Er zijn zeven senioren en vijf junioren nodig om alles af te werken in die korte periode. Aangezien de zeven senioren en vijf junioren niet altijd aan het werk zijn, zullen we beide limiteren tot maximaal drie per week. Daarvoor is een nieuwe Gantt chart voor opgesteld, figuur 3. Het project zal 22 weken duren (geteld vanaf week 1), vijf weken langer maar met veel minder mankracht. Op de grafieken onder de Gantt chart is ook meteen te zien dat alle senioren en junioren constanter aan het werk zijn. Er zijn geen pieken meer waarbij er ineens veel meer mensen nodig zijn.

Voor de risico analyse is het geval van oneindig veel mankracht berekend. De waardes zijn te vinden in tabel 5. De optimistic, likely, pessimistic en estimated times (OT,LT,PT,ET) waren al gegeven. Daaruit is de standaard deviatie S per onderdeel berekend. Dan is voor ieder onderdeel de standaard deviatie (SP) van het kritieke pad berekend naar dat onderdeel. Daarnaast zijn ook de cumulative pessimistic time (cumPT) en de cumulative likely time (cumLT) berekend om de cumulatieve worst-case delay (cumWCD) te bepalen. De laatste geeft aan wat de ergste vertragingen kunnen veroorzaken. In het ergste geval kan het zijn dat we pas 10 weken later kunnen opleveren.

3 Actors

Om een duidelijker overzicht te maken met wie/wat het programma zal moeten intrageren zijn de *primary* en *secondaryactors* bepaald (tbl6). Voor alle actors zijn ook de goals en tasks opgezet. Deze zijn terug te vinden in tabel 7 en 8.

De klant/customer zal een account willen registreren zodat hij een product kan bestellen. Hij zal ook producten willen kopen en/of tijdelijk in een winkelwagentje (cart) bijhouden zodat hij ze later kan kopen. Hij zal ook zijn orders willen bekijken en misschien wil hij ze cancellen.

Een administrator wil statestieken op kunnen halen van alle producten. Hij wil ook categorieën kunnen aanmaken en daar producten aan toevoegen.

De bank moet de betalingen regelen. De betaling zal bevestigd moeten worden en als de klant het wil, zal ze ook gecancelled worden.

De delivery service moet aangeven wanneer de order klaar is om geleverd te worden. Vanaf dan kan de order niet meer ongedaan worden gemaakt.

De mail service zal de conformatie e-mails moeten versturen naar de klant.

4 Unexpected Changes and exceptions

Bepaalde uitzonderingen of veranderingen moeten worden opgevangen om consistentie en robuustheid te kunnen garanderen.

Stel bijvoorbeeld dat de klant meer van een product koopt dan in de stock aanwezig is. Dan moet er een error worden gegeven zodat de order niet door kan gaan. Wat er daarna gebeurt is aan de klant.

Het winkelwagentje zal bijgehouden worden als dat kan omdat ze geen schade aan kan richten. Stel dat er net een betaling is gebeurt wanneer er een uitzondering plaatst vindt, dan moet de bank deze betaling annuleren en de order wordt gecancelled en terug in het winkelwagentje geplaatst. Als de administrator een order wil aanpassen en tijdens het aanpassen komt deze op "ready" te staan, wordt de aanpassing genegeerd.

5 Extra Vragen Voor de Use Cases

Wanneer moet er allemaal een conformatie mail gestuurd worden? Mogen er categorieën verwijderd worden en zo ja, wat als er nog producten in zitten? Hoe worden de meest populaire items bepaald? Wat zit er allemaal in de persoonlijke info, maakt het e-mail adres daar deel vanuit? Welke betalingsvormen zijn er allemaal?

6 Scenario's

Er zijn vijf scenario's bedacht, deze zijn te zien in figuren 4, 5, 6, 7 en 8.

In het eerste scenario plaats een klant een order. Hij gaat de nodige info moeten geven (bv. betalingsmethode, lever adres, etc.) om dan te kunnen betalen. De bank zal dan de betaling moeten bevestigen zodat de mail service een conformatie mail naar de klant kan sturen. Als laatste moet de delivery service de order op "ready" zetten, wanneer dat gebeurd is, is de order afgerond.

Bij het tweede scenario is een klant naar een product aan het zoeken/browsen. Hij kan de catalogus sorteren en producten aan zijn winkelwagentje toevoegen.

Bij het derde scenario wordt er een order gecancelled. Dit kan door zowel de customer als de administrator gedaan worden. De bank zal dan het geld terug moeten storten naar de customer en de mail service zal een conformatie mail sturen naar de klant zodat die op de hoogte wordt gebracht.

Bij het vierde scenario registreert een klant zich. Hij moet zijn persoonlijke informatie meegeven, naam en email. De mail service zal dan een conformatie mail sturen.

In het laatste scenario verandert een administrator een order. De bank moet dan geld terug storten naar de klant als nodig en er moet een conformatie mail naar de klant worden gestuurd.

Er zijn nog drie scenario's verder uitgediept. Twee primaire scenario's waar er niet mis gaat, zie tabellen 1 en 3 en een scenario dat een alternatief pad is van een ander scenario tabel 2.

7 Overige niet-functionele requirements

- Efficiënt en snel zoeken in de database naar producten.
- Snelle sorteer algoritmes voor de categorieën.
- Lichte webpagina's.
- Wisselen van bank toestaan.

De eerste drie zijn er voor user-friendliness, ze zorgen voor een snelle en daarmee aangename shop ervaring. De lichte webpagina's zijn nodig om het op alle platformen bruikbaar te maken, anders kan webbrowsen op bv. een gsm voor stottingen zorgen.

Het wisselen van bank moet, anders kan het zijn dat bepaalde gebruikers niet kunnen betalen omdat zij niet de juiste bank hebben.

8 Objecten in het systeem

Om het de objecten te herkennen in zijn de objecten in onderlijnde, vette letters gezet in de bijlage "Functional Requirements". Voor de objecten zijn er dan candidaat klasse namen opgesteld, tabel 14.

9 CRC cards

Om duidelijker te maken wat objecten zijn en hoe ze moeten intrageren zijn er ook nog CRC cards gemaakt. Deze staan in de bijlage. De uitleg per scenario staat hieronder.

9.1 Scenario A: Display Catalog to user

Zie tabel 9.

Catalog is verantwoordelijk voor het sorteren van de items en het weergeven aan de client, omdat het weergeven voor elke client hetzelfde is en voor alle items ook hetzelfde is. Item heeft basisinfo die word weergegeven door de catalog en ook een hoeveelheid in stock, zodat items out of stock niet in de catalog worden weergegeven. User heeft geen responsibilities hier, maar wel een collaboration met Catalog.

9.2 Scenario B: Add Item to cart

Zie tabel 10.

Elke User heeft een eigen Cart en kan dus ook items toevoegen aan zijn eigen Cart. De Cart zelf bevat Items met elks een eigen hoeveelheid. Het toevoegen van items gebeurt in User omdat elke user een cart geassocieerd heeft met zichzelf en dat dus niet geïmplementeerd kan worden in Item.

9.3 Scenario C: Remove item from cart

Zie tabel 11.

Dit scenario is net zoals scenario B alleen in plaats van een item toe te voegen, wordt er een item verwijderd. Daarom passen we onze CRC cards ook zo gewoon aan en moeten we er niets extra aan aanpassen.

9.4 Scenario D: Admin changes Order

Zie tabel 12. Een Client kan meerdere Orders hebben, maar een Order kan maar exact één Client hebben, daarom is dit een verantwoordelijkheid van Order om dit bij te houden. Een Order kan alleen maar aangepast worden als de order zelf nog niet geleverd is, daarom wordt er een flag bijgehouden voor elke Order. Administrators mogen orders aanpassen en een order wordt geïdentificeerd door zijn ID. Een Order bevat ook verwijzingen naar Items, houdt bij welke items er in zitten en hoeveel.

9.5 Scenario E: Place Order and pay

Zie tabel 13.

Een Order wordt gegenereerd op basis van de Cart of winkelwagentje van een Client. De totaalprijs van de order wordt berekend en toegevoegd aan de Order zodat er betaald kan worden. De betaling gebeurt extern via een bank of online bankieren, als die succesvol is, dan wordt er een flag op de Order aangepast en dan kan de order klaargemaakt worden voor levering.

10 Object-oriented vs functional decomposition

Object-oriented decomposition is beter voor complexe systemen en door encapsulatie is er een zekere robuustheid tegen veranderingen. Object-oriented decomposition is duidelijk een betere keuze hier, omdat er een hoop veranderingen kunnen gebeuren, bijvoorbeeld in orders en hoe die betaald worden. Een andere reden dat object-oriented beter is, is omdat in de requirements er ook al een aantal entiteiten (objecten) gespecificeerd zijn die geïmplementeerd moeten worden. Die kunnen we dus al gebruiken om een object-oriented hiërarchie mee op te stellen. Ook zorgt object-oriented ervoor dat de functionaliteit gegroepeerd staat bij de objecten die ze nodig hebben.

11 Bijlage

11.1 Functional Requirements

De volgende functionele requirements zijn gegeven door de klant.

- The **system** must have following entities in order to describe the current situation of orders and clients:
 - **Items**: items that can be sold. An item has a **name**, **price** and **category** (electronics, home & living, hifi, ...).
 - **Cart**: the current items a client has chosen but not yet ordered.
 - **Order**: a single order made by a client.
 - **Clients**: all registered and unregistered clients that already ordered something.
- Following categories must exist, but it should be possible to add new categories to the store: Phone, Tablet, Computer, Image, Sound, Home & Living, Kitchen, Travel, Fashion, Sport. All items must be displayed in a **catalog**, ordered by category. Every category should be displayed on a separate page. A user can sort the items according to name, popularity or price.
- A user can add/remove items to/from the cart and change the desired quantity.
- A user has the ability to order the items in his cart. The user then has to fill in some standard information and afterwards select a payment method.
- When a **payment** has been received a confirmation mail has to be sent to the user.
- An **administrator** must be able to make changes to placed (but not yet delivered) orders.
- An administrator should be able to get an overview of all open and delivered orders.
- A user is able to log in and get an overview of his personal information, his mailing preferences and his open and delivered orders.
- An overview of some of the most popular items should be displayed on the home page of the shop.
- An administrator must be able to get some statistics and analysis of previous orders. Such as: for every item the total amount sold, find items that are often sold together. Required is the Eclat algorithm for itemset mining. Also recommendations should be given when a user is looking for an item.
- A user should be able to cancel his order as long as it is not yet ready for delivery.

11.2 Non-Functional Requirements

De volgende niet-functionele requirements zijn gegeven door de klant.

- Programming Language: Java
- Web-based UI
- Adaptability: UI should be displayed nice on every device
- Availability
- Robustness
- Easy-to-use
- No inconsistent states should occur

Use Case 1	Customer plaats order
Goal en Context	Customer bevestigt order & betaald
Scope & level	
Preconditions	Er zit minstens 1 item in de order Alles in de order moet in stock zijn
Succes endcondition	Delivery zet order op "ready"
Fail endcondition	Cancel betaling/order Bank accepteert betaling niet
Primary Actors	Customer
Secondary Actors	Bank Delivery
Trigger	Customer bevestigt order
Beschrijving:	
stap	Actie
1	Customer bevestigt order
2	Customer geeft payment info
3	Customer betaalt
4	Stuur betaalt request naar de bank
5	Bank accepteert
6	Stuur conformatie mail
7	Wacht totdat delivery order op "ready" zet
8	Zet order op "ready"
Subvariaties:	
3	Selecteer betaal methode
Alternatieve Paden:	
2,3,4	Customer cancelled betaling
5,6,7	Customer cancelled order
5	Bank accepteert betaling niet

Table 1: Scenario 1: primair scenario waarbij de klant een order gaat betalen.

Use Case 3	Customer Cancels Order
Goal en Context	Order wordt gecancelled
Scope & level	
Preconditions	Order is niet "ready"
Succes endcondition	Order is gecancelled
Fail endcondition	Order is niet gecancelled
Primary Actors	Customer
Secondary Actors	Bank
Trigger	Customer cancels order
Beschrijving:	
stap	Actie
1	Customer cancels order
2	Stuur terugbetalings request naar bank
3	Bank conformeert terugbetaling
4.1	Stuur conformatie mail naar klant
4.2	Steek items terug in stock
Subvariaties:	
—geen—	
Alternatieve Paden:	
—geen—	

Table 2: Scenario 2: secundair scenario waarbij een klant een order cancelled.

Use Case 2	Administrator verandert order
Goal en Context	De administrator wilt een order veranderen
Scope & level	
Preconditions	Order is niet "ready"
Succes endcondition	Order is veranderd & opgeslagen Conformatie mail is verzonden
Fail endcondition	Order is op "ready" gezet
Primary Actors	Administrator
Secondary Actors	Bank
Trigger	Administrator selecteert order

Beschrijving:

stap	Actie
1	Administrator selecteert order
2	Administrator veranderd order
3	Bank geeft geld terug als nodig
4.1	Stuur conformatie mail naar customer
4.2	Steek items terug in stock

Subvariaties:

—geen—

Alternatieve Paden:

2 Order kan hier nog op "ready" worden gezet.

Table 3: Scenario 3: primair scenario waarbij een administrator een order aanpast.

onderdeel	<i>earliest start date</i>	<i>latest end date</i>
Start	0	0
Items	1	2
Clients	1	3
Cart	4	6
Catalog	3	6
Order	3	4
Payment	4	6
Sort Items	7	8
Add Items To Cart	7	7
Place Order	7	8
Pay Order	7	8
GUI: Show item pages	9	11
Admin: Retrieve Orders	9	9
Admin: Change Order	9	11
Show Polular Items	12	10
Admin: Get Stats	12	14
Finishing	15	16

Table 4: Bovenstaande tabel laat de *earliest start dates* en *latest end dates* van ieder onderdeel zien. De *latest end date* eindigen die week. Start bijvoorbeeld begint in week 0 en eindigt ook in week 0. Ze duurt dus 1 week.

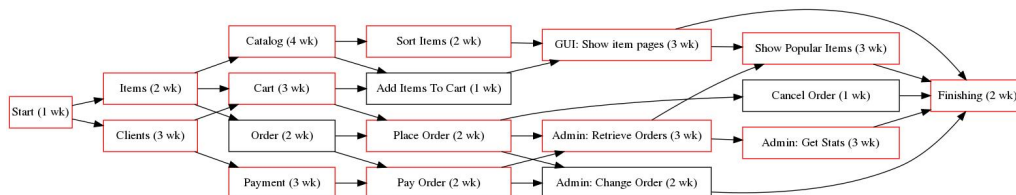


Figure 1: Gegeven PERT chart van de delen van het project. De kritieke paden zijn aangegeven in het rood.

onderdeel	LT	OT	PT	ET	S	SP	cumPT	cumLT	cumWCD
Start	1	1	1	1.00	0.00	0.00	1	1	0
Items	2	1	3	2.00	0.33	0.33	4	3	1
Clients	3	3	4	3.17	0.17	0.17	5	4	1
Catalog	4	3	6	4.17	0.50	0.50	10	7	3
Cart	3	3	5	3.33	0.33	0.37	10	7	3
Order	2	2	2	2.00	0.00	0.33	6	5	1
Payment	3	2	5	3.17	0.50	0.53	10	7	3
Sort Items	2	2	2	2.00	0.00	0.53	12	9	3
Add Items to Cart	1	1	2	1.17	0.17	0.53	12	8	4
Place Order	2	2	2	2.00	0.00	0.37	12	9	3
Pay Order	2	2	4	2.33	0.33	0.53	14	9	5
Gui: Show Item Page	3	3	4	3.17	0.17	0.55	16	12	4
Cancel Order	1	1	2	1.17	0.17	0.41	14	10	4
Admin: Retrieve Orders	3	3	3	3.00	0.00	0.53	17	12	5
Admin: Change Order	2	1	2	1.83	0.17	0.55	16	11	5
Show Popular Items	3	2	6	3.33	0.67	0.87	23	15	8
Admin: Get Stats	3	2	4	3.00	0.33	0.62	21	15	6
Finishing	2	2	4	2.33	0.33	0.93	27	17	10

Table 5: Bovenstaande tabel de waarschijnlijke (likely) tijd (LT), de optimistische tijd (OT), de pessimistische tijd (PT), de geschatte (estimated) tijd (ET) zien. Ze laat ook de standaard deviatie (S) op iedere tijd zien en de grootste standaard deviatie op het kritieke pad naar die node (SP). De cumulatieve pessimistische en waarschijnlijke tijd zijn te zien in cumPT en cumLT en de cumulatieve worst-case delay is te vinden in de kolom cumWCD.

primary actors:
customer
administrator
secondary actors:
Bank
Delivery Service
Mail service

Table 6: Boven is een overzicht van de primary en secondary actors te zien.

actors	goals
Customer	producten kopen producten bijhouden voor later orders bekijken orders cancellen registreer persoonlijke info aanpassen producten zoeken
Administrator	verander orders statistieken ophalen categorieën toevoegen producten toevoegen aan categorieën

Table 7: Boven staat een overzicht van de goals van de primary actors.

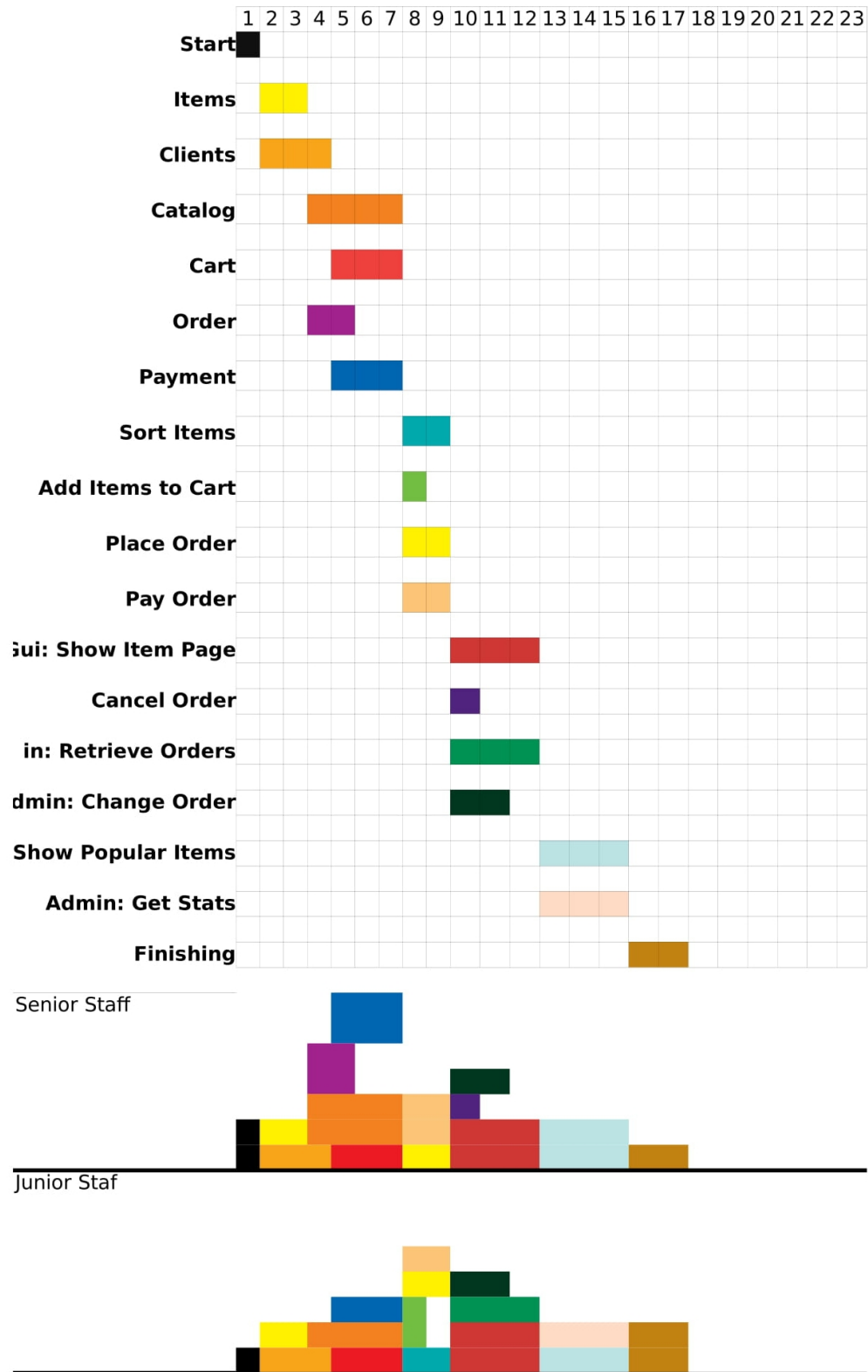


Figure 2: Boven is de Gantt chart te zien waarbij er oneindig veel mankracht gebruikt mag worden. Daaronder is de werkverdeling per week te zien.

actors	tasks
Bank	accept/confirmeer betaling cancel betaling
Delivery	order op "ready" zetten
Mail Service	conformatie e-mails sturen

Table 8: Boven staat een overzicht van de tasks van de secondary actors.



Figure 3: Boven is de Gantt chart te zien waarbij er maximaal drie senioren en drie junioren developers ten alle tijde aanwezig zijn. Daaronder is de werkverdeling per week te zien.

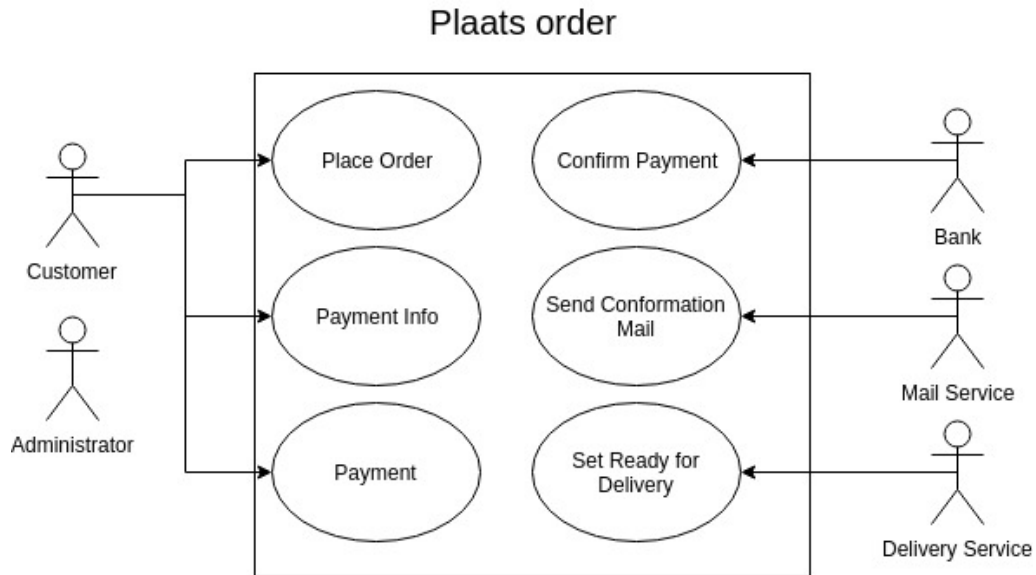


Figure 4: Boven is het eerste scenario te zien waarbij er een order wordt geplaatst en betaald.

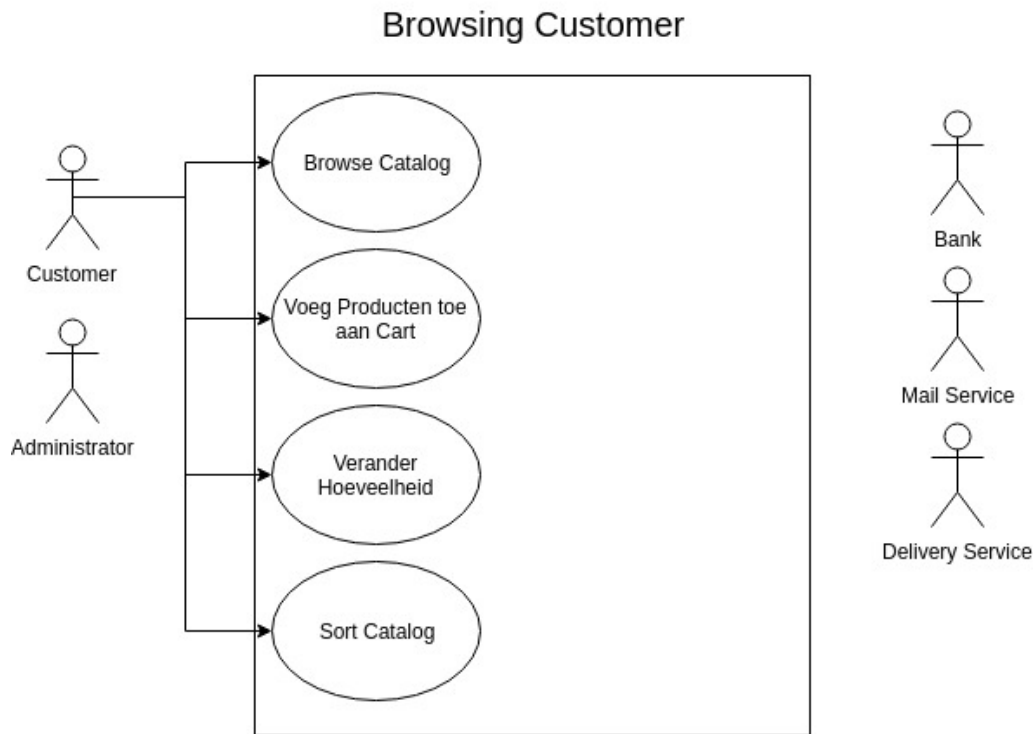


Figure 5: Boven is het tweede scenario te zien waarbij een customer aan het zoeken is naar producten.

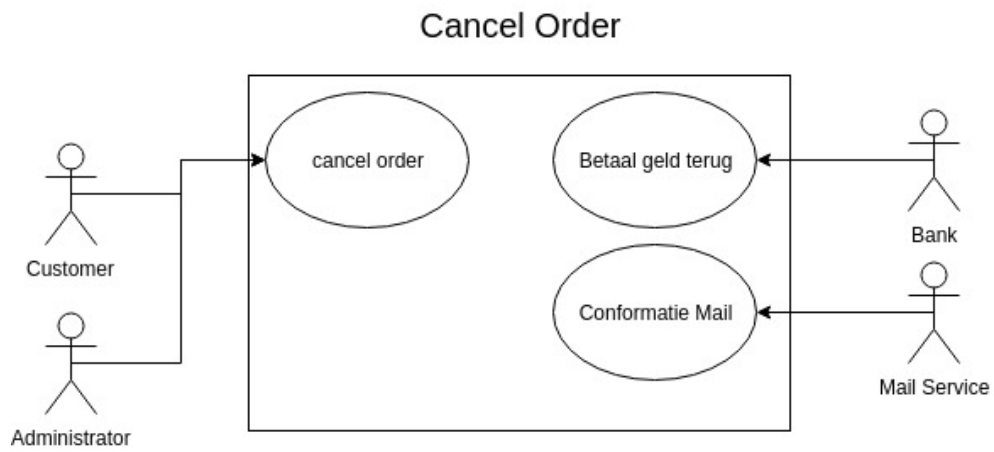


Figure 6: Boven is het derde scenario te zien waarbij de customer of de administrator een order cancelled.

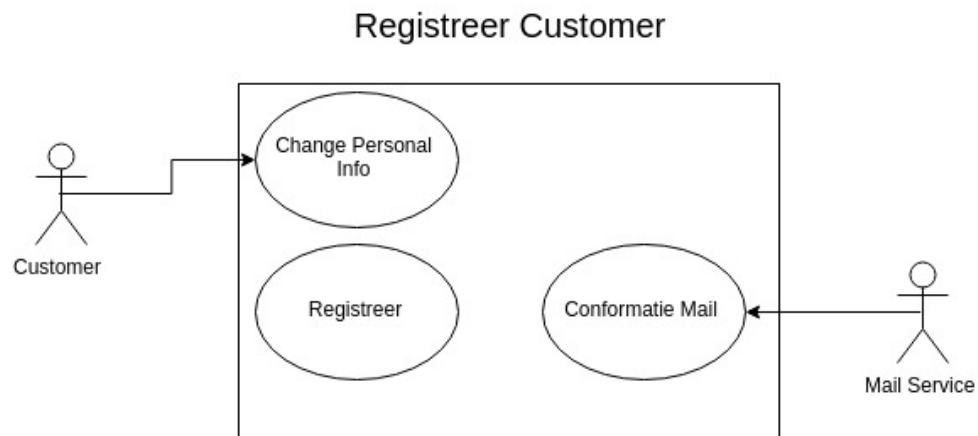


Figure 7: Boven is het vierde scenario te zien. Een customer registreert zichzelf hier.

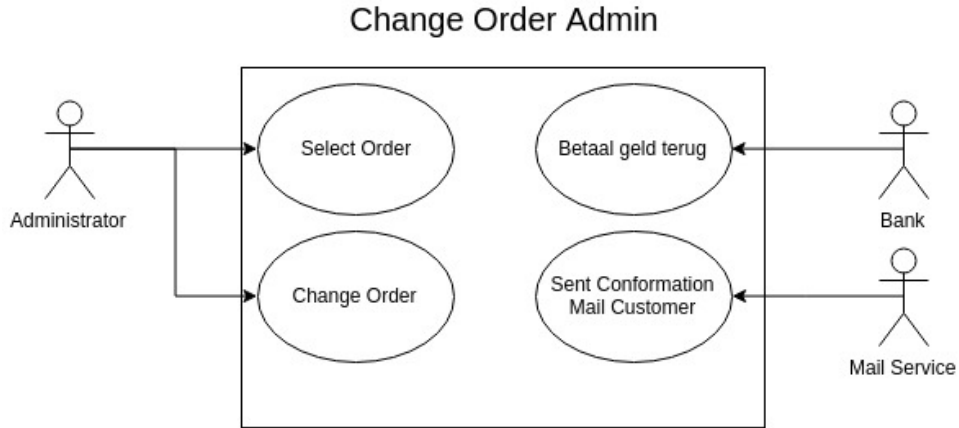


Figure 8: Boven is het vijfde en laatste scenario te zien. Ze laat zien wat er gebeurt als een administrator een order wil veranderen.

Class Catalog		Class User	
Sub/Super-classes		Sub/Super-classes	
/		Sub: Client	
Responsibilities		Responsibilities	
Responsibility	Collaboration	Responsibility	Collaboration
Sort Items By Category	Item		
Display To Catalog	Client		

Class Item		Class Client	
Sub/Super-classes		Sub/Super-classes	
/		Super: User	
Responsibilities		Responsibilities	
Responsibility	Collaboration	Responsibility	Collaboration
Contains amount in stock			
Contains basic product information			

Table 9: Scenario A : Display Catalog To User

Class Cart		Class User	
Sub/Super-classes		Sub/Super-classes	
/		Sub: Client	
Responsibilities		Responsibilities	
Responsibility	Collaboration	Responsibility	Collaboration
Contains items, each with an amount	Item		
Linked to specific Client	Client		

Class Item		Class Client	
Sub/Super-classes		Sub/Super-classes	
/		Super: User	
Responsibilities		Responsibilities	
Responsibility	Collaboration	Responsibility	Collaboration
Contains basic product information	/	Can add an Item to its own Cart	Cart, Item

Table 10: Scenario B : Add Item to Cart

Class Cart		Class User	
Sub/Super-classes		Sub/Super-classes	
/		Sub: Client	
Responsibilities		Responsibilities	
Responsibility	Collaboration	Responsibility	Collaboration
Contains items, each with an amount	Item		
Linked to specific Client	Client		

Class Item		Class Client	
Sub/Super-classes		Sub/Super-classes	
/		Super: User	
Responsibilities		Responsibilities	
Responsibility	Collaboration	Responsibility	Collaboration
Contains basic product information	/	Can remove an Item to its own Cart	Cart, Item

Table 11: Scenario C : Remove Item to Cart

Class Item		Class Client	
Sub/Super-classes		Sub/Super-classes	
/		Super: User	
Responsibilities		Responsibilities	
Responsibility	Collaboration	Responsibility	Collaboration
Contains basic product information	/		

Class User		Class Administrator	
Sub/Super-classes		Sub/Super-classes	
Sub: Administrator, Client		Super: User	
Responsibilities		Responsibilities	
Responsibility	Collaboration	Responsibility	Collaboration
		Can change orders	Order

Class Order	
Sub/Super-classes	
/	
Responsibilities	
Responsibility	Collaboration
Contains items, each with an amount	Item
Linked to specific Client	Client
Has an order ID	/
Has a flag 'delivered'	/

Table 12: Scenario D : Admin changes order

Class Item	
Sub/Super-classes	
/	
Responsibilities	
Responsibility	Collaboration
Contains basic product information	/

Class Order	
Sub/Super-classes	
/	
Responsibilities	
Responsibility	Collaboration
Contains items, each with an amount	Item
Linked to specific Client	Client
Calculates a total price based on Items	Item
Has an order ID	/
Has a flag 'Paid'	/

Class Cart	
Sub/Super-classes	
/	
Responsibilities	
Responsibility	Collaboration
Contains items, each with an amount	Item
Linked to specific Client	Client
Checkout to Order	Item, Client, Order

Class User	Class Client
Sub/Super-classes	Sub/Super-classes
Sub: Client	Super: User
Responsibilities	Responsibilities
Responsibility	Collaboration

Table 13: Scenario E : Place Order and Pay

Nouns & Synonyms	Candidate Class Name
System	- : dont model the system
store, shop	Store (? Model ourselves)
Item	Item (+: core concept)
Cart	Cart (+: core concept)
Order	Order (+: core concept)
Client	Client (+: core concept)
Administrator	Administrator (+: core concept)
Catalog	Catalog (+: core concept)
Category	Category (? item attribute)
Payment	Payment (+: core concept)

Table 14: Objecten die voorkomen in het systeem.